GUIDE TO AVAILABLE MATHEMATICAL SOFTWARE

Ronald F. Boisvert
Sally E. Howe
David K. Kahaner

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Applied Mathematics
Scientific Computing Division
Washington, DC 20234

Issued January 1984

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director
The Guide to Available Mathematical Software (GAMS) documents software that has been made available for use by the staff of the National Bureau of Standards by its Center for Applied Mathematics. Certain software products, both public-domain and commercial, are identified in GAMS in order to adequately document this software. Identification of such products does not imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the identified software is necessarily the best available for staff purposes. Conversely, the omission of any such software does not imply its unsuitability for use. The GAMS staff welcome information about scientific software of possible use to NBS staff.
PREFACE

Recent advances in the mathematical sciences have resulted in the emergence of a vast body of reliable and well-designed computer software for aiding in the solution of scientific problems. This software represents both a substantial investment in time and money for its development and maintenance, and a substantial potential savings in time and money to users. The second edition of the Guide to Available Mathematical Software (GAMS) is part of an on-going effort at the National Bureau of Standards to catalog general-purpose mathematical and statistical software used by NBS staff at both the Gaithersburg and the Boulder sites.

The first edition of GAMS contained information about five subprogram libraries: CMLIB, IMSL, MATHWARE, NAG, and PORT. Documentation for the IMSL, NAG, and PORT libraries has been updated for the second edition to reflect their current versions. Eleven new sublibraries have been added to CMLIB which, like MATHWARE, is a repository for public-domain software. Information about software from ten additional libraries has been added to this edition; these libraries are BMDP, DATAPAC, INVAR, MATLAB, Minitab, PDELIB, PLOD, SLDGL, Spectral, and STATLIB. These collections consist not only of subprogram libraries, but also collections of stand-alone programs and interactive systems. In all, 2767 subprograms, programs, and interactive system commands are cataloged here. The inclusion of so much more software prompted a major revision to the classification scheme used to catalog the software.

As with the first edition, this guide to computer software was produced using computer software. "Modules by Class" and the "Module Dictionary" are the output of computer programs that systematically query a database designed for this project using BCS-RIM, a portable proprietary relational information management system. The document was typeset using the \TeX{} text formatting system.

We appreciate the assistance many people provided in the preparation of this edition of GAMS. We first thank the following NBS staff members for their contributions: Janet Blanchard, James Blue, Paul Boggs, Elsie Clark, Amy Del Giorno, George Dines, Janet Donaldson, Jim Filliben, Karla Hoffman, Ric Jackson, Janice Knapp-Cordes, Martin Knapp-Cordes, John Koontz, Daniel Lozier, Colin MacDonald, Jay Murphy, Dianne O'Leary, George Orwell, Hans Oser, Susan Parker, Bert Rust, John Smith, Irene Stegun, Pete Stewart, Selden Stewart, Linda Sung, Roland Sweet, and Anne Trevey. John Barkley, Richard Mattis, Theresa Rodriguez, and Charles Wilson of the NBS Semiconductor Devices and Circuits Division provided VAX computer facilities. We thank Burton Colvin, Director of the Center for Applied Mathematics (CAM); Glenn Ingram, Associate Director for Computing, CAM; and Francis Sullivan, Chief of the Scientific Computing Division, CAM; for their support of research in mathematical and statistical software. The following software library developers provided information about their libraries: Brian Ford at NAG, Phyllis Fox at Bell Laboratories, Jim Gentle at IMSL, MaryAnn Hill at BMDP, and Barbara Ryan at the Minitab Project. The Committee on Statistical Algorithms, Statistical Computing Section, American Statistical Association, assisted with the statistics component of this work; Committee members are: John Aleong, Ken Berk, J. Philip Miller, John Monahan, Bill Sallas, and Del Scott. Finally, we thank the members of the SLATEC (Sandia–Los Alamos–Air Force Weapons Laboratory Technical Exchange Committee) Common Math Library Subcommittee. This common Department of Energy Fortran subprogram library forms the major component of the NBS Core Math Library (CMLIB).

As software for solving scientific problems evolves, so will software documentation such as GAMS. A number of high-quality software products used by NBS staff are not included in this edition of GAMS; we hope to add them to future editions. The format may continue to evolve as we learn how to make GAMS more useful and as technology provides new tools. We welcome your suggestions for improvements.

Ronald F. Boisvert
Sally E. Howe
David K. Kahaner
Scientific Computing Division
Center for Applied Mathematics
National Bureau of Standards
U.S. Department of Commerce
Washington, D.C. 20234

January 1984
CONTENTS

GAMS HELP Card ................................................................. Inside Front Cover
Preface ................................................................................ iii
How to Use GAMS ................................................................. v
GAMS Classes ........................................................................ A1 - A15
Modules by Class ................................................................. B0 - B190
Legend .................................................................................. B0
A. Arithmetic, Error Analysis .................................................. B1
B. Number Theory ................................................................. B4
C. Elementary and Special Functions ...................................... B5
D. Linear Algebra ................................................................... B18
E. Interpolation ...................................................................... B54
F. Solution of Nonlinear Equations ......................................... B61
G. Optimization ..................................................................... B66
H. Differentiation and Integration ........................................... B75
I. Differential and Integral Equations ...................................... B85
J. Integral Transforms ............................................................ B98
K. Approximation .................................................................... B101
L. Statistics, Probability ........................................................ B110
M. Simulation, Stochastic Modeling ....................................... B173
N. Data Handling .................................................................... B174
O. Symbolic Computation ..................................................... B180
P. Computational Geometry .................................................. B181
Q. Graphics ........................................................................... B183
R. Service Routines .............................................................. B186
S. Software Development Tools ............................................. B190
Module Dictionary ................................................................. C0 - C184
Legend .................................................................................. C0
Library Reference ................................................................. D0 - D21
Legend .................................................................................. D0
Index ..................................................................................... E1 - E19
HOW TO USE GAMS

This document is organized so that an NBS staff member can find software that solves a given mathematical or statistical problem. Briefly, given such a problem, a person should

1. find the classification which most clearly identifies the problem—in GAMS CLASSES or the INDEX,
2. find the list of software that solves problems in that classification—in MODULES BY CLASS, and
3. find further information about that software, such as how to obtain detailed documentation or how to access the software—in the MODULE DICTIONARY and the LIBRARY REFERENCE.

The software cataloged in GAMS is organized into fifteen libraries. A library can be either a collection of subprograms, a collection of programs, or an interactive system. Brief descriptions of these libraries are given in the LIBRARY REFERENCE section. Each of these libraries is available for use at the Gaithersburg NBS site, and many are also available at the Boulder site.

Each library is a collection of modules. Depending upon the type of library, a module can be a single subprogram, a stand-alone program, or a command in an interactive system. In some cases, libraries are partitioned into sublibraries. The modules in a given sublibrary come from a single developer and usually solve a narrow range of problems.

The following describes the five sections of GAMS.

A. GAMS CLASSES

This section contains the classification scheme used to organize the software by mathematical problem. The scheme is tree-structured, with the highest level identified by a single letter and corresponding to very general problems such as J (Integral Transforms). Subclasses are identified by alternating letter-and-number combinations; thus, for example, the subclasses of J1 (Fast Fourier transforms) are labeled J1a, J1b, etc. The longer the class identifier, i.e. the lower the level of the tree, the more detailed the problem specification.

B. MODULES BY CLASS

Information about the software cataloged in each class can be obtained from this section, which is organized in the same way as the GAMS classification scheme. For each class, there may be a discussion of the software in the class and/or a list of software.

The list of software for a given class is organized in alphabetical order by library and module name. The software modules in the class are listed along with brief descriptions of what each module does. These lists generally appear at lower levels of the classification scheme, although software modules which solve broad classes of problems may appear listed at higher levels of the classification scheme. Some classes have no modules.

Discussions are included at the highest levels of the classification scheme (major classes A through S have discussion sections, for example). A discussion addresses the computational issues associated with software in its class and subclasses. The discussions usually include literature references. Most of the references are available at the NBS-Gaithersburg Library (Administration Building, Room E120) and at the Department of Commerce Boulder Laboratories Main Library (Radio Building, Room 1202).

C. MODULE DICTIONARY

Information about individual modules can be obtained from this section, which is organized alphabetically by module name. For each module this information includes a brief description, the library (and sublibrary, if appropriate) to which it belongs, portability information, how to obtain detailed documentation, and how to access the software on the Sperry 1100 at NBS. This information is fully described in the MODULE DICTIONARY Legend (page C9).

D. LIBRARY REFERENCE

This section provides information about the libraries to which modules belong. For each library this information includes type (e.g., subprogram library, program library, or interactive system), version, a description and brief listing of the contents, portability information, library developer, and a reference. For each type of computer at NBS on which the library is supported, information is provided about level of NBS support, how to obtain detailed library documentation, and how to gain access to the library on that computer. This information is fully described in the LIBRARY REFERENCE Legend (page D0).
E. INDEX

This section is a key word and phrase index to the classification scheme. It provides an alternative to GAMS CLASSES as a means of finding a particular problem in the classification scheme.

ON-LINE GAMS

An on-line version of GAMS is available to users of the Sperry 1100 at NBS. It is updated regularly and therefore provides the most current information about available software. This interactive program is executed with the command

```
@XQT NBS*XLIB$GAMS (in EXEC)
```

or

```
CALL GAMS (in CTS).
```

Complete usage instructions can be obtained by typing "?" after the program is entered.
A. ARITHMETIC, ERROR ANALYSIS

A1. Integer
A2. Rational
A3. Real
   A3a. Single precision
   A3b. Double precision
   A3c. Extended precision
   A3d. Extended range
A4. Complex
   A4a. Single precision
   A4b. Double precision
   A4c. Extended precision
   A4d. Extended range
A5. Interval
   A5a. Real
   A5b. Complex
A6. Change of representation
   A6a. Type conversion
   A6b. Base conversion
   A6c. Decomposition, construction
A7. Sequences (e.g., convergence acceleration)

B. NUMBER THEORY

C. ELEMENTARY AND SPECIAL FUNCTIONS (search also class L5)

C1. Integer-valued functions (e.g., floor, ceiling, factorial, binomial coefficient)
C2. Powers, roots, reciprocals
C3. Polynomials
   C3a. Orthogonal
      C3a1. Trigonometric
      C3a2. Chebyshev, Legendre
      C3a3. Laguerre
      C3a4. Hermite
   C3b. Non-orthogonal
C4. Elementary transcendental functions
   C4a. Trigonometric, inverse trigonometric
   C4b. Exponential, logarithmic
   C4c. Hyperbolic, inverse hyperbolic
   C4d. Integrals of elementary transcendental functions
C5. Exponential and logarithmic integrals
C6. Cosine and sine integrals
C7. Gamma
   C7a. Gamma, log gamma, reciprocal gamma
   C7b. Beta, log beta
   C7c. Psi function
   C7d. Polygamma function
   C7e. Incomplete gamma
   C7f. Incomplete beta
C7g. Riemann zeta
C8. Error functions
C8a. Error functions, their inverses, integrals, including the normal distribution function
C8b. Fresnel integrals
C8c. Dawson's integral
C9. Legendre functions
C10. Bessel functions
C10a. J, Y, H(1), H(2)
C10a1. Real argument, integer order
C10a2. Complex argument, integer order
C10a3. Real argument, real order
C10a4. Complex argument, real order
C10a5. Complex argument, complex order
C10b. I, K
C10b1. Real argument, integer order
C10b2. Complex argument, integer order
C10b3. Real argument, real order
C10b4. Complex argument, real order
C10b5. Complex argument, complex order
C10c. Kelvin functions
C10d. Airy and Scorer functions
C10e. Struve, Anger, and Weber functions
C10f. Integrals of Bessel functions
C11. Confluent hypergeometric functions
C12. Coulomb wave functions
C13. Jacobian elliptic functions, theta functions
C14. Elliptic integrals
C15. Weierstrass elliptic functions
C16. Parabolic cylinder functions
C17. Mathieu functions
C18. Spheroidal wave functions
C19. Other special functions

D. LINEAR ALGEBRA
D1. Elementary vector and matrix operations
D1a. Elementary vector operations
D1a1. Set to constant
D1a2. Minimum and maximum components
D1a3. Norm
D1a3a. \( L_1 \) (sum of magnitudes)
D1a3b. \( L_2 \) (Euclidean norm)
D1a3c. \( L_\infty \) (maximum magnitude)
D1a4. Dot product (inner product)
D1a5. Copy or exchange (swap)
D1a6. Multiplication by scalar
D1a7. Triad \((ax+y)\) for vectors \(x\), \(y\) and scalar \(a\)
D1a8. Elementary rotation (Givens transformation)
D1a9. Elementary reflection (Householder transformation)
D1a10. Convolutions
D1b. Elementary matrix operations
D1b1. Set to zero, to identity
D1b2. Norm
D1b3. Transpose
D1b4. Multiplication by vector
D1b5. Addition, subtraction
D1b6. Multiplication
D1b7. Matrix polynomial
D1b8. Copy
D1b9. Storage mode conversion
D1b10. Elementary rotation (Givens transformation)
D1b11. Elementary reflection (Householder transformation)

D2. Solution of systems of linear equations (including inversion, LU and related decompositions)
D2a. Real nonsymmetric matrices
D2a1. General
D2a2. Banded
D2a2a. Tridiagonal
D2a3. Triangular
D2a4. Sparse
D2b. Real symmetric matrices
D2b1. General
D2b1a. Indefinite
D2b1b. Positive definite
D2b2. Positive definite banded
D2b2a. Tridiagonal
D2b4. Sparse
D2c. Complex non-Hermitian matrices
D2c1. General
D2c2. Banded
D2c2a. Tridiagonal
D2c3. Triangular
D2c4. Sparse
D2d. Complex Hermitian matrices
D2d1. General
D2d1a. Indefinite
D2d1b. Positive definite
D2d2. Positive definite banded
D2d2a. Tridiagonal
D2d4. Sparse
D2e. Associated operations (e.g., matrix reorderings)

D3. Determinants
D3a. Real nonsymmetric matrices
D3a1. General
D3a2. Banded
D3a2a. Tridiagonal
D3a3. Triangular
D3a4. Sparse
D3b. Real symmetric matrices
D3b1. General
D3b1a. Indefinite
D3b1b. Positive definite
D3b2. Positive definite banded
D3b2a. Tridiagonal
D3b4. Sparse
D3c. Complex non-Hermitian matrices
D3c1. General
D3c2. Banded
D3c2a. Tridiagonal
D3c3. Triangular
D3c4. Sparse
D3d. Complex Hermitian matrices
  D3d1. General
  D3d1a. Indefinite
  D3d1b. Positive definite
D3d2. Positive definite banded
  D3d2a. Tridiagonal
D3d4. Sparse

D4. Eigenvalues, eigenvectors
  D4a. Ordinary eigenvalue problems (Ax = λx)
    D4a1. Real symmetric
    D4a2. Real nonsymmetric
    D4a3. Complex Hermitian
    D4a4. Complex non-Hermitian
    D4a5. Tridiagonal
    D4a6. Banded
    D4a7. Sparse
    D4b. Generalized eigenvalue problems (e.g., Ax = λBx)
      D4b1. Real symmetric
      D4b2. Real general
      D4b3. Complex Hermitian
      D4b4. Complex general
      D4b5. Banded
      D4c. Associated operations
      D4c1. Transform problem
        D4c1a. Balance matrix
        D4c1b. Reduce to compact form
        D4c1b1. Tridiagonal
        D4c1b2. Hessenberg
        D4c1b3. Other
        D4c1c. Standardize problem
        D4c2. Compute eigenvalues of matrix in compact form
          D4c2a. Tridiagonal
          D4c2b. Hessenberg
          D4c2c. Other
          D4c3. Form eigenvectors from eigenvalues
          D4c4. Back transform eigenvectors
          D4c5. Determine Jordan normal form
  D5. QR decomposition, Gram-Schmidt orthogonalization
  D6. Singular value decomposition
  D7. Update matrix decompositions
    D7a. LU
    D7b. Cholesky
    D7c. QR
    D7d. Singular value
  D8. Other matrix equations (e.g., AX + XB = C)
  D9. Overdetermined or underdetermined systems of equations, singular systems,
      pseudo-inverses (search also classes D5, D6, D9, K1a, L8a)

E. INTERPOLATION
E1. Univariate data (curve fitting)
   E1a. Polynomial splines (piecewise polynomials)
   E1b. Polynomials
   E1c. Other functions (e.g., rational, trigonometric)
E2. Multivariate data (surface fitting)
   E2a. Gridded
   E2b. Scattered
E3. Service routines (e.g., grid generation, evaluation of fitted functions) (search also class N5)

F. SOLUTION OF NONLINEAR EQUATIONS

F1. Single equation
   F1a. Smooth
       F1a1. Polynomial
           F1a1a. Real coefficients
           F1a1b. Complex coefficients
       F1a2. Nonpolynomial
       F1b. General (no smoothness assumed)
F2. System of equations
   F2a. Smooth
   F2b. General (no smoothness assumed)
F3. Service routines (e.g., check user-supplied derivatives)

G. OPTIMIZATION (search also classes K, L8)

G1. Unconstrained
   G1a. Univariate
       G1a1. Smooth function
           G1a1a. User provides no derivatives
           G1a1b. User provides first derivatives
           G1a1c. User provides first and second derivatives
       G1a2. General function (no smoothness assumed)
   G1b. Multivariate
       G1b1. Smooth function
           G1b1a. User provides no derivatives
           G1b1b. User provides first derivatives
           G1b1c. User provides first and second derivatives
       G1b2. General function (no smoothness assumed)
G2. Constrained
   G2a. Linear programming
       G2a1. Dense matrix of constraints
       G2a2. Sparse matrix of constraints
   G2b. Transportation and assignments problem
   G2c. Integer programming
       G2c1. Zero/one
       G2c2. Covering and packing problems
       G2c3. Knapsack problems
       G2c4. Matching problems
   G2c5. Routing, scheduling, location problems
   G2c6. Pure integer programming
   G2c7. Mixed integer programming
   G2d. Network (for network reliability search class M)
       G2d1. Shortest path
<table>
<thead>
<tr>
<th>G2d2.</th>
<th>Minimum spanning tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2d3.</td>
<td>Maximum flow</td>
</tr>
<tr>
<td>G2d3a.</td>
<td>Generalized networks</td>
</tr>
<tr>
<td>G2d3b.</td>
<td>Networks with side constraints</td>
</tr>
<tr>
<td>G2d4.</td>
<td>Test problem generation</td>
</tr>
<tr>
<td>G2e.</td>
<td>Quadratic programming</td>
</tr>
<tr>
<td>G2e1.</td>
<td>Positive definite Hessian (i.e. convex problem)</td>
</tr>
<tr>
<td>G2e2.</td>
<td>Indefinite Hessian</td>
</tr>
<tr>
<td>G2f.</td>
<td>Geometric programming</td>
</tr>
<tr>
<td>G2g.</td>
<td>Dynamic programming</td>
</tr>
<tr>
<td>G2h.</td>
<td>General nonlinear programming</td>
</tr>
<tr>
<td>G2h1.</td>
<td>Simple bounds</td>
</tr>
<tr>
<td>G2h1a.</td>
<td>Smooth function</td>
</tr>
<tr>
<td>G2h1a1.</td>
<td>User provides no derivatives</td>
</tr>
<tr>
<td>G2h1a2.</td>
<td>User provides first derivatives</td>
</tr>
<tr>
<td>G2h1a3.</td>
<td>User provides first and second derivatives</td>
</tr>
<tr>
<td>G2h1b.</td>
<td>General function (no smoothness assumed)</td>
</tr>
<tr>
<td>G2h2.</td>
<td>Linear equality or inequality constraints</td>
</tr>
<tr>
<td>G2h2a.</td>
<td>Smooth function</td>
</tr>
<tr>
<td>G2h2a1.</td>
<td>User provides no derivatives</td>
</tr>
<tr>
<td>G2h2a2.</td>
<td>User provides first derivatives</td>
</tr>
<tr>
<td>G2h2a3.</td>
<td>User provides first and second derivatives</td>
</tr>
<tr>
<td>G2h2b.</td>
<td>General function (no smoothness assumed)</td>
</tr>
<tr>
<td>G2h3.</td>
<td>Nonlinear constraints</td>
</tr>
<tr>
<td>G2h3a.</td>
<td>Equality constraints only</td>
</tr>
<tr>
<td>G2h3a1.</td>
<td>Smooth function and constraints</td>
</tr>
<tr>
<td>G2h3a1a.</td>
<td>User provides no derivatives</td>
</tr>
<tr>
<td>G2h3a1b.</td>
<td>User provides first derivatives of function and constraints</td>
</tr>
<tr>
<td>G2h3a1c.</td>
<td>User provides first and second derivatives of function and constraints</td>
</tr>
<tr>
<td>G2h3a2.</td>
<td>General function and constraints (no smoothness assumed)</td>
</tr>
<tr>
<td>G2h3b.</td>
<td>Equality and inequality constraints</td>
</tr>
<tr>
<td>G2h3b1.</td>
<td>Smooth function and constraints</td>
</tr>
<tr>
<td>G2h3b1a.</td>
<td>User provides no derivatives</td>
</tr>
<tr>
<td>G2h3b1b.</td>
<td>User provides first derivatives of function and constraints</td>
</tr>
<tr>
<td>G2h3b1c.</td>
<td>User provides first and second derivatives of function and constraints</td>
</tr>
<tr>
<td>G2h3b2.</td>
<td>General function and constraints (no smoothness assumed)</td>
</tr>
<tr>
<td>G2i.</td>
<td>Global solution to nonconvex problems</td>
</tr>
<tr>
<td>G3.</td>
<td>Optimal control</td>
</tr>
<tr>
<td>G4.</td>
<td>Service routines</td>
</tr>
<tr>
<td>G4a.</td>
<td>Problem input (e.g., matrix generation)</td>
</tr>
<tr>
<td>G4b.</td>
<td>Problem scaling</td>
</tr>
<tr>
<td>G4c.</td>
<td>Check user-supplied derivatives</td>
</tr>
<tr>
<td>G4d.</td>
<td>Find feasible point</td>
</tr>
<tr>
<td>G4e.</td>
<td>Check for redundancy</td>
</tr>
<tr>
<td>G4f.</td>
<td>Other</td>
</tr>
</tbody>
</table>

### H.

**DIFFERENTIATION, INTEGRATION**

<table>
<thead>
<tr>
<th>H1.</th>
<th>Numerical differentiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2.</td>
<td>Quadrature (numerical evaluation of definite integrals)</td>
</tr>
<tr>
<td>H2a.</td>
<td>One-dimensional integrals</td>
</tr>
<tr>
<td>H2a1.</td>
<td>Finite interval (general integrand)</td>
</tr>
<tr>
<td>H2a1a.</td>
<td>Integrand available via user-defined procedure</td>
</tr>
<tr>
<td>H2a1a1.</td>
<td>Automatic (user need only specify required accuracy)</td>
</tr>
<tr>
<td>H2a1a2.</td>
<td>Nonautomatic</td>
</tr>
</tbody>
</table>
H2a1b. Integrand available only on grid
H2a1b1. Automatic (user need only specify required accuracy)
H2a1b2. Nonautomatic
H2a2. Finite interval (specific or special type integrand including weight functions, oscillating and singular integrands, principal value integrals, splines, etc.)
H2a2a. Integrand available via user-defined procedure
H2a2a1. Automatic (user need only specify required accuracy)
H2a2a2. Nonautomatic
H2a2b. Integrand available only on grid
H2a2b1. Automatic (user need only specify required accuracy)
H2a2b2. Nonautomatic
H2a3. Semi-infinite interval (including $e^{-x}$ weight function)
H2a3a. Integrand available via user-defined procedure
H2a3a1. Automatic (user need only specify required accuracy)
H2a3a2. Nonautomatic
H2a4. Infinite interval (including $e^{-x^2}$ weight function)
H2a4a. Integrand available via user-defined procedure
H2a4a1. Automatic (user need only specify required accuracy)
H2a4a2. Nonautomatic
H2b. Multidimensional integrals
H2b1. One or more hyper-rectangular regions
H2b1a. Integrand available via user-defined procedure
H2b1a1. Automatic (user need only specify required accuracy)
H2b1a2. Nonautomatic
H2b1b. Integrand available only on grid
H2b1b1. Automatic (user need only specify required accuracy)
H2b1b2. Nonautomatic
H2b2. Nonrectangular region, general region
H2b2a. Integrand available via user-defined procedure
H2b2a1. Automatic (user need only specify required accuracy)
H2b2a2. Nonautomatic
H2b2b. Integrand available only on grid
H2b2b1. Automatic (user need only specify required accuracy)
H2b2b2. Nonautomatic
H2c. Service routines (compute weight and nodes for quadrature formulas)

I. DIFFERENTIAL AND INTEGRAL EQUATIONS

II. Ordinary differential equations
IIa. Initial value problems
IIa1. General, nonstiff or mildly stiff
IIa1a. One-step methods (e.g., Runge-Kutta)
IIa1b. Multistep methods (e.g., Adams' predictor-corrector)
IIa1c. Extrapolation methods (e.g., Bulirsch-Stoer)
IIa2. Stiff and mixed algebraic-differential equations
IIb. Multipoint boundary value problems
IIb1. Linear
IIb2. Nonlinear
IIb3. Eigenvalue (e.g., Sturm-Liouville)
IIc. Service routines (e.g., interpolation of solutions, error handling)

II. Partial differential equations
IIa. Initial boundary value problems
IIa1. Parabolic
IIa1a. One spatial dimension
IIa1b. Two or more spatial dimensions
IIa2. Hyperbolic
I2b. Elliptic boundary value problems
I2b1. Linear
I2b1a. Second order
I2b1a1. Poisson (Laplace) or Helmholtz equation
I2b1a1a. Rectangular domain (or topologically rectangular in the coordinate system)
I2b1a1b. Nonrectangular domain
I2b1a2. Other separable problems
I2b1a3. Nonseparable problems
I2b1c. Higher order equations (e.g., biharmonic)
I2b2. Nonlinear
I2b3. Eigenvalue
I2b4. Service routines
I2b4a. Domain triangulation \(\text{search also class P2a2c1}\)
I2b4b. Solution of discretized elliptic equations

J.

INTEGRAL TRANSFORMS

J1. Fast Fourier transforms \(\text{search class L10 for time series analysis}\)
J1a. One-dimensional
J1a1. Real
J1a2. Complex
J1a3. Trigonometric (sine, cosine)
J1b. Multidimensional
J2. Convolutions
J3. Laplace transforms
J4. Hilbert transforms

K.

APPROXIMATION \(\text{search also class L8}\)

K1. Least squares \((L_2)\) approximation
K1a. Linear least squares \(\text{search also classes D5, D6, D9}\)
K1a1. Unconstrained
K1a1a. Univariate data (curve fitting)
K1a1a1. Polynomial splines (piecewise polynomials)
K1a1a2. Polynomials
K1a1a3. Other functions (e.g., rational, trigonometric, user-specified)
K1a1b. Multivariate data (surface fitting)
K1a2. Constrained
K1a2a. Linear constraints
K1a2b. Nonlinear constraints
K1b. Nonlinear least squares
K1b1. Unconstrained
K1b1a. Smooth functions
K1b1a1. User provides no derivatives
K1b1a2. User provides first derivatives
K1b1a3. User provides first and second derivatives
K1b1b. General functions
K1b2. Constrained
K1b2a. Linear constraints
K1b2b. Nonlinear constraints
K2. Minimax \((L_\infty)\) approximation
K3. Least absolute value \((L_1)\) approximation
K4. Other analytic approximations (e.g., Taylor polynomial, Padé)
K5. Smoothing
K6. Service routines (e.g., mesh generation, evaluation of fitted functions) (*search also class N5*)

L. STATISTICS, PROBABILITY

L1. Data summarization
   L1a. One univariate quantitative sample
       L1a1. Ungrouped data
       L1a1a. Location
       L1a1b. Dispersion
       L1a1c. Shape
       L1a1d. Distribution, density
       L1a2. Ungrouped data with missing values
       L1a3. Grouped data
       L1a3a. Location
       L1a3b. Dispersion
       L1a3c. Shape
   L1e. One univariate qualitative (proportional) sample
       L1e1. Two or more univariate samples or one multivariate sample
       L1e1a. Ungrouped data
       L1e1a. Location
       L1e1b. Correlation
       L1e2. Ungrouped data with missing values
       L1e3. Grouped data
       L1f. Two or more multivariate samples

L2. Data manipulation (*search also class N*)
   L2a. Transform (*search also class N6 for sorting, ranking*)
   L2b. Group
   L2c. Sample
   L2d. Subset

L3. Graphics (*search also class Q*)
   L3a. Histograms
   L3b. Distribution functions
   L3c. Scatter diagrams
       L3c1. y vs. x
   L3c2. Symbol plots
   L3c3. Multiple plots
   L3c4. Probability plots
       L3c4a. Beta, binomial
       L3c4b. Cauchy, chi-squared
       L3c4c. Double exponential
       L3c4d. Exponential, extreme value
       L3c4e. F distribution
       L3c4f. Gamma, geometric
       L3c4g. Halfnormal
       L3c4h. Lambda, logistic, lognormal
       L3c4i. Negative binomial, normal
       L3c4j. Pareto, Poisson
   L3c4k. t distribution
   L3c4l. Uniform
   L3c4m. Weibull
   L3c5. Time series plots (y; vs. i, vertical, lag)
   L3d. EDA graphics
GAMS: Classification Scheme

January 1984

L4. Elementary statistical inference, hypothesis testing
   L4a. One univariate quantitative sample
       L4a1. Ungrouped data
           L4a1a. Parameter estimation
           L4a12. Binomial
           L4a15. Extreme value
           L4a114. Normal
           L4a116. Poisson
           L4a121. Uniform
           L4a123. Weibull
           L4a1b. Distribution-free (nonparametric) analysis
           L4a1c. Goodness-of-fit tests
           L4a1d. Tests on sequences of numbers
           L4a1e. Density and distribution function estimation
           L4a1f. Tolerance limits
           L4a2. Ungrouped data with missing values
           L4a3. Grouped data
           L4a3a. Parameter estimation
           L4a3a14. Normal
   L4b. Two or more univariate quantitative samples
       L4b1. Ungrouped data
           L4b1a. Parameter estimation
           L4b114. Normal
           L4b1b. Distribution-free (nonparametric) analysis
           L4b2. Ungrouped data with missing values
           L4b3. Grouped data
   L4c. One univariate qualitative (proportional) sample
   L4d. Two or more univariate samples
   L4e. One multivariate sample
       L4e1. Ungrouped data
           L4e1a. Parameter estimation
           L4e114. Normal
           L4e1b. Distribution-free (nonparametric) analysis
           L4e2. Ungrouped data with missing values
           L4e2a. Parameter estimation
           L4e2b. Distribution-free (nonparametric) analysis
           L4e3. Grouped data
           L4e3a. Parameter estimation
           L4e3a14. Normal
           L4e3b. Distribution-free (nonparametric) analysis
           L4e4. Two or more multivariate samples
           L4e4a. Parameter estimation
           L4e4a14. Normal
   L5. Function evaluation (search also class C)
       Univariate
       L5a1. Cumulative distribution functions, probability density functions
           L5a1b. Beta, binomial
           L5a1c. Cauchy, chi-squared
           L5a1d. Double exponential
           L5a1e. Error function, exponential, extreme value
           L5a1f. F distribution
           L5a1g. Gamma, general, geometric
           L5a1h. Halfnormal, hypergeometric
           L5a1k. Kolmogorov-Smirnov
           L5a1l. Lambda, logistic, lognormal
           L5a1n. Negative binomial, normal
GAMS: Classification Scheme

January 1984

A 11

L5a1p. Pareto, Poisson
L5a1t. t distribution
L5a1u. Uniform
L5a1w. Weibull
L5a2. Inverse cumulative distribution functions, sparsity functions
L5a2b. Beta, binomial
L5a2c. Cauchy, chi-squared
L5a2d. Double exponential
L5a2e. Exponential, extreme value
L5a2f. F distribution
L5a2g. Gamma, general, geometric
L5a2h. Halfnormal
L5a2l. Lambda, logistic, lognormal
L5a2n. Negative binomial, normal, normal scores
L5a2p. Pareto, Poisson
L5a2t. t distribution
L5a2u. Uniform
L5a2w. Weibull
L5b. Multivariate
L5b1. Cumulative distribution functions, probability density functions
L5b1n. Normal

L6. Pseudo-random number generation
L6a. Univariate
L6a2. Beta, binomial, Boolean
L6a3. Cauchy, chi-squared
L6a4. Double exponential
L6a5. Exponential, extreme value
L6a6. F distribution
L6a7. Gamma, general (continuous, discrete) distributions, geometric
L6a8. Halfnormal, hypergeometric
L6a9. Integers
L6a12. Lambda, logical, logistic, lognormal
L6a14. Negative binomial, normal
L6a15. Order statistics
L6a16. Pareto, permutations, Poisson
L6a19. Samples, stable distribution
L6a20. t distribution, time series, triangular
L6a21. Uniform
L6a22. Von Mises
L6a23. Weibull
L6b. Multivariate
L6b3. Contingency table, correlation matrix
L6b13. Multinomial
L6b14. Normal
L6b15. Orthogonal matrix
L6b21. Uniform
L6c. Service routines (e.g., seed)

L7. Experimental design, including analysis of variance
L7a. Univariate
L7a1. One-way analysis of variance
L7a1a. Parametric analysis
L7a1a1. Contrasts, multiple comparisons
L7a1a2. Analysis of variance components
L7a1b. Distribution-free (nonparametric) analysis
L7a2. Balanced multiway design
L7a2a. Complete
A 12

GAM: Classification Scheme

January 1984

L7a2a1. Parametric analysis
L7a2a1a. Two-way
L7a2a1b. Factorial
L7a2a1c. Nested
L7a2a2. Distribution-free (nonparametric) analysis
L7a2b. Incomplete
L7a2b1. Parametric analysis
L7a2b1a. Latin square
L7a2b1b. Lattice designs
L7a2b2. Distribution-free (nonparametric) analysis
L7a3. Analysis of covariance
L7a4. General linear model (unbalanced design)
L7a4a. Parametric analysis
L7a4b. Distribution-free (nonparametric) analysis
L7b. Multivariate

L8. Regression (search also classes G, K)
L8a. Linear least squares ($L_2$) (search also classes D5, D6, D9)
L8a1. Simple (e.g., $y = \beta_0 + \beta_1 x + \epsilon$)
L8a1a. Ordinary
L8a1a1. Unweighted
L8a1a1a. No missing values
L8a1a1b. Missing values
L8a1a2. Weighted
L8a1b. Through the origin
L8a1c. Errors in variables
L8a1d. Calibration (inverse regression)
L8a2. Polynomial (e.g., $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$)
L8a2a. Not using orthogonal polynomials
L8a2a1. Unweighted
L8a2a2. Weighted
L8a2b. Using orthogonal polynomials
L8a2b1. Unweighted
L8a2b2. Weighted
L8a3. Piecewise polynomial (i.e. multiphase or spline)
L8a4. Multiple (e.g., $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$)
L8a4a. Ordinary
L8a4a1. Unweighted
L8a4a1a. No missing values
L8a4a1b. Missing values
L8a4a1c. From correlation data
L8a4a1d. Using principal components
L8a4a1e. Using preference pairs
L8a4a2. Weighted
L8a4b. Errors in variables
L8a4d. Logistic
L8a5. Variable selection
L8a6. Regression design
L8a7. Several multiple regressions
L8a8. Multivariate
L8a9. Diagnostics
L8a10. Hypothesis testing, inference
L8a10a. Lack-of-fit tests
L8a10b. Analysis of residuals
L8a10c. Inference
L8b. Biased (ridge)
L8c. Linear least absolute value ($L_1$)
L8d. Linear minimax \((L_{\infty})\)
L8e. Robust
L8f. EDA
L8g. Nonlinear
L8g1. Unweighted
L8g1a. Derivatives not supplied
L8g1b. Derivatives supplied
L8g2. Weighted
L8g2a. Derivatives not supplied
L8g2b. Derivatives supplied
L8h. Service routines
L9. Categorical data analysis
L9a. 2-by-2 tables
L9b. Two-way tables
L9c. Log-linear model
L9d. EDA (e.g., median polish)
L10. Time series analysis (search also class L9c5 for time series graphics)
L10a. Transformations, transforms (search also class J1)
L10b. Smoothing, filtering
L10c. Autocorrelation analysis
L10d. Complex demodulation
L10e. ARMA and ARIMA modeling and forecasting
L10e1. Model and parameter estimation
L10e2. Forecasting
L10f. Spectral analysis
L10g. Cross-correlation analysis
L10g1. Parameter estimation
L10g2. Forecasting
L11. Correlation analysis
L12. Discriminant analysis
L13. Factor analysis
L13a. Principal components analysis
L14. Cluster analysis
L14a. Unconstrained
L14a1. Nested
L14a1a. Joining (e.g., single link)
L14a1b. Divisive
L14a2. Non-nested
L14b. Constrained
L14b1. One-dimensional
L14b2. Two-dimensional
L14c. Display
L15. Life testing, survival analysis

M. SIMULATION, STOCHASTIC MODELING (search also classes L6, L10)
M1. Simulation
M1a. Discrete
M1b. Continuous (Markov models)
M2. Queueing
M3. Reliability
M3a. Quality control
M3b. Electrical network
M4. Project optimization (e.g., PERT)

N.

DATA HANDLING (search also class LR)

N1. Input, output
N2. Bit manipulation
N3. Character manipulation
N4. Storage management (e.g., stacks, heaps, trees)
N5. Searching
  N5a. Extreme value
  N5b. Insertion position
  N5c. On a key
N6. Sorting
  N6a. Internal
    N6a1. Passive (i.e. construct pointer array, rank)
      N6a1a. Integer
      N6a1b. Real
    N6a1b1. Single precision
    N6a1b2. Double precision
    N6a1c. Character
    N6a2. Active
      N6a2a. Integer
      N6a2b. Real
    N6a2b1. Single precision
    N6a2b2. Double precision
    N6a2c. Character
  N6b. External
N7. Merging
N8. Permuting

O.

SYMBOLIC COMPUTATION

P.

COMPUTATIONAL GEOMETRY (search also classes G, Q)

P1. One dimension
P2. Two dimensions
  P2a. Points, lines
    P2a1. Relationships
      P2a1a. Closest and farthest points
      P2a1b. Intersection
    P2a2. Graph construction
    P2a2a. Convex hull
    P2a2b. Minimum spanning tree
    P2a2c. Region partitioning
    P2a2c1. Triangulation
    P2a2c2. Voronoi diagram
  P2b. Polygons (e.g., intersection, hidden line problems)
    P2b1. Circles
P3. Three dimensions
P3a. Points, lines, planes
P3b. Polytopes
P3c. Spheres
P4. More than three dimensions

Q. GRAPHICS (search also classes L3, P)
Q1. Line printer plotting

R. SERVICE ROUTINES
R1. Machine-dependent constants
R2. Error checking (e.g., check monotonicity)
R3. Error handling
R3a. Set criteria for fatal errors
R3b. Set unit number for error messages
R3c. Other utility programs
R4. Documentation retrieval

S. SOFTWARE DEVELOPMENT TOOLS
S1. Program transformation
S2. Static analysis
S3. Dynamic analysis
A: Arithmetic, Error Analysis

Modern scientific computers provide facilities to perform the basic arithmetic operations such as addition, multiplication, and division on both fixed and floating-point numbers. Programming languages make these facilities directly accessible through their implementations of integer and real arithmetic. Other arithmetic systems are also useful for scientific computation, however. Complex arithmetic is the prime example; it is available in major scientific programming languages, and many software libraries provide software to support it. Other important examples are multiple precision integer and real arithmetic, rational arithmetic, and interval arithmetic.

Software implementing non-standard, though elementary, arithmetic operations on a variety of data types are classified in this chapter. For example, the NAG and PORT libraries have collections of Fortran subprograms which implement basic arithmetic operations for both single and double precision complex numbers using only real arithmetic.

Conversion from one data type to another is another elementary operation for which software exists. The PORT library, for example, provides a complete set of Fortran subprograms for performing standard type conversion, including the construction of machine-base numbers given a base-10 mantissa and exponent, and the inverse operation.

January 1984

A: Arithmetic, error analysis

A1: Integer

IMSL subprogram library

VDCPS Decompose an integer into its prime factors.

A2: Rational

A3: Real

A3a: Single precision

A3b: Double precision

A3c: Extended precision

A3d: Extended range

A4: Complex

A4a: Single precision

CMLIB subprogram library (FNLIB sublibrary)

CARG Argument = \( \theta \), in radians of complex number, \( z = |z|e^{i\theta} \).
NAG subprogram library

A02ABE  Modulus of a complex number. Double precision version is A02ABF.
A02ACE  Quotient of two complex numbers. Double precision version is A02ACF.
C00GBE  Complex conjugate of complex data values, Hermitian sequence. Double precision version is C00GBF.
C00GCE  Complex conjugate of complex data values, general sequence. Double precision version is C00GCF.

A4b :  Double precision

NAG subprogram library

A02ABF  Modulus of a complex number.
A02ACF  Quotient of two complex numbers.
C00GBF  Complex conjugate of complex data values, Hermitian sequence.
C00GCF  Complex conjugate of complex data values, general sequence.

PORT subprogram library

CDADD  Adds two complex double precision numbers. Each is represented by a double precision array of two elements.
CDDIV  Divides two double precision complex numbers. Each is represented by a double precision array of two elements.
CDMUL  Multiplies two double precision complex numbers. Each is represented by a double precision array of two elements.

A4c :  Extended precision

A4d :  Extended range

A5 :  Interval

A5a :  Real

A5b :  Complex

A6 :  Change of representation

A6a :  Type conversion

PORT subprogram library

CNVBDC  Converts values from one vector to another. Backward loop, double precision into complex.
CNVBDI  Converts values from one vector to another. Backward loop, double precision into integer.
CNVBDR  Converts values from one vector to another. Backward loop, double precision into real.
CNVBIC  Converts values from one vector to another. Backward loop, integer into complex.
CNVBID  Converts values from one vector to another. Backward loop, integer to double precision.
CNVBIR  Converts values from one vector to another. Backward loop, integer into real.
CNVBRC  Converts values from one vector to another. Backward loop, real into complex.
CNVBRD  Converts values from one vector to another. Backward loop, real into double precision.
CNVBRI  Converts values from one vector to another. Backward loop, real into integer.
CNVFDC  Converts values from one vector to another. Forward loop, double precision into complex.
CNVFDR  Converts values from one vector to another. Forward loop, double precision into integer.
CNVFIC  Converts values from one vector to another. Forward loop, integer into complex.
CNVFID  Converts values from one vector to another. Forward loop, integer to double precision.
CNVFIR  Converts values from one vector to another. Forward loop, integer to real.
CNVFRC  Converts values from one vector to another. Forward loop, real into complex.
CNVFRI  Converts values from one vector to another. Forward loop, real into integer.

A6b :  Base conversion

PORT subprogram library

VBTO D  Converts a mantissa and exponent into a base 10 floating point number. Double precision version is DVBTOD.

A6c :  Decomposition, construction

PORT subprogram library

UMKFL  Decomposes a non-zero floating point number into a mantissa and an exponent. Double precision version is DUMKFL.
VDTO B  Converts a base-10 mantissa and exponent of a floating point number into a machine-base representation. Double precision version is DVDTOB.

A7 :  Sequences (e.g., convergence acceleration)

NAG subprogram library

C06BAE  Performs Shanks' transformation on a given sequence of real values by means of the $\epsilon$-algorithm of Wynn. A (possibly unreliable) estimate of the absolute error is also given. An erratic, but often powerful method for accelerating sequences. Double precision version is C06BAF.
B: Number Theory

Software classified in this chapter performs such number-theoretic calculations as the decomposition of integers into prime factors.

VDCPS Decompose an integer into its prime factors.
C: Elementary and Special Functions

This chapter contains computer programs for selected mathematical functions. The provision of such software is still a wide open field. Some idea of the difficulties involved is given in the remainder of this introduction. The reader who is interested only in what is currently available should refer immediately to the categories below for further specific information. For the special distribution functions of statistics see also class L5.

Some elementary functions are provided by the Fortran compiler. Any compiler that conforms to the ANSI standard should include single and double precision, and in some cases single precision complex, versions of the square root function, the exponential and logarithmic functions, the circular functions and inverses, and the hyperbolic functions. Some compilers provide more. For example, Sperry ASCII Fortran (FTN) includes double precision complex versions of some of these functions, plus the error, complementary error, gamma, and log gamma functions in single and double precision.

But when the same Fortran program is to be run on computers made by different manufacturers, it is not wise to rely on any nonstandard feature of the language. Therefore portable Fortran programs for mathematical functions, other than the ones required by the Fortran standard (ANSI X3.9 - 1978), are of great interest. The major implementational difficulty is how to achieve efficiently the requisite accuracy for different computer wordlengths.

This can be seen by considering very briefly how a special function is computed. It is always computed from some form of approximation. Usually the approximation derives from some type of truncated infinite expansion. In general, the more accurate the approximation, the more expensive the computation. The additional expense can be quite considerable, especially when the approximation involves expansions in series of functions instead of just a power series. Thus an efficient portable function must be self-adaptive in the sense of being able to select the most suitable approximation from a set of possible ones.

There are algorithmic difficulties as well. Rounding errors occur in actually computing an approximation to a function in finite precision. These errors accumulate so that the final computed function value may be much less accurate than expected. Good algorithms minimize the rounding error, but some error always remains. Even for relatively simple functions it can be quite large, especially when the number of arithmetic operations that must take place is large. In any case the algorithm that is used should be stable in the sense that it should not amplify rounding errors too much as the computation progresses.

The accuracy of functions implemented in the compiler is generally higher than the accuracy of functions in a portable library. This is because it is more convenient and less costly in assembly language than in Fortran to perform the computation in extended precision.

Progress has been made in meeting these difficulties in recent years. There now exist fairly widely accepted Fortran calls that return static information about the number of bits in a computer word, the exponent range, etc. for both single and double precision. This information has been used effectively in some cases to solve the algorithm selection problem. Research work continues on developing better algorithms and computer arithmetic, on improving the design of Fortran subroutines for mathematical functions, and on developing standards for programming languages and function libraries.

This work will lead to mathematical function software that is easier to use, gives fuller coverage of input ranges of interest, and that has more versatile modes of recovery from faults. It will also lead to a much wider selection of mathematical functions in portable libraries.

References


C: Elementary and special functions (search also class L5)

C1: Integer-valued functions (e.g., floor, ceiling, factorial, binomial coefficient)

CMLIB subprogram library (FNLIB sublibrary)

BINOM Binomial \( \frac{n!}{(m! \cdot (n-m)!)} \). Double precision version is DBINOM.

FAC Factorial, \( = n! \). Double precision version is DFAC.

POCH Pochhammer's symbol \( (a)_n = \Gamma(a + n)/\Gamma(a) \). Double precision version is DPOCH.

POCH1 Pochhammer's symbol from first order, \( = (a)_n - 1/n \). Double precision version is DPOCH1.

PORT subprogram library

CEIL Finds the smallest integer greater than or equal to \( x \). Input and output are real. Double precision version is DCEIL.

FLR Finds the largest integer less than or equal to \( x \). Input and output are real. Double precision version is DFLR.

ICEIL Finds the smallest integer greater than or equal to \( x \). Input is real, output is integer.

IDCEIL Finds the smallest integer greater than or equal to \( x \). Input is double precision, output is integer.

IFLR Finds the largest integer less than or equal to \( x \). Input is real, output is integer. Double precision version is IDFLR.

C2: Powers, roots, reciprocals

CMLIB subprogram library (FNLIB sublibrary)

CBRT Cube root of real number. Double precision version is DCBRT.

CCBRT Complex cube root of complex argument.

SQRT Square root.

NAG subprogram library

A02AAE Evaluates the square root of a complex number. Double precision version is A02AAF.

C3: Polynomials

PORT subprogram library

ORTH Evaluates a polynomial expressed as a sum of general orthogonal polynomials. Double precision version is DORTH.

C3a: Orthogonal

C3a1: Trigonometric

PORT subprogram library

TRIGP Evaluates a trigonometric polynomial with given coefficients. Double precision version is DTRIGP.
### C3a2: Chebyshev, Legendre

<table>
<thead>
<tr>
<th>CMLIB subprogram library (FCNPAK sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XSLEGF</strong> Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. Double precision version is XDLEGF.</td>
</tr>
<tr>
<td><strong>XSNRMP</strong> Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. Double precision version is XDNRMP.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CMLIB subprogram library (FNLIB sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CSEVL</strong> Evaluates an n term series of Chebyshev polynomials at a given point. Double precision version is DCSEVL.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CMLIB subprogram library (QUADSP sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>QMOMO</strong> Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. Double precision version is DQMOMO.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IMSL subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RLPOL</strong> Generate orthogonal (Gram) polynomials on a given set of abscissas</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAG subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C06DBE</strong> Sum of a Chebyshev series. Double precision version is C06DBF.</td>
</tr>
<tr>
<td><strong>E02AEE</strong> Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). Double precision version is E02AEF.</td>
</tr>
<tr>
<td><strong>E02AHE</strong> Derivative of fitted polynomial in Chebyshev series form. Double precision version is E02AHF.</td>
</tr>
<tr>
<td><strong>E02AJE</strong> Integral of fitted polynomial in Chebyshev series form. Double precision version is E02AJF.</td>
</tr>
<tr>
<td><strong>E02AKE</strong> Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. Double precision version is E02AKF.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PORT subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TCHBP</strong> Evaluates a polynomial expressed as a sum of Chebyshev polynomials. Double precision version is DTCHBP.</td>
</tr>
</tbody>
</table>

### C3a3: Laguerre

### C3a4: Hermite

### C3b: Non-orthogonal

### C4: Elementary transcendental functions

<table>
<thead>
<tr>
<th>IMSL subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MMLINC</strong> Computes an elementary integral from which inverse circular functions, logarithms or inverse hyperbolic functions may be computed by rational operations.</td>
</tr>
</tbody>
</table>
C4a: Trigonometric, inverse trigonometric

CMLIB subprogram library (FNLIB sublibrary)

CACOS  Arc cosine of complex argument, \( \cos^{-1} z \).
CASIN  Arc sin of complex argument, \( \sin^{-1} z \).
CATAN  Arc tangent of complex argument, \( \tan^{-1} z \).
CATAN2 Quadrant correct arctangent of complex arguments, \( \tan^{-1}(z_1/z_2) \).
CCOT   Cotangent of complex argument, cot \( z \).
COS    Cosine of real argument, \( \cos x \).
COSDG  Computes the cosine of an argument given in degrees. Double precision version is DCOSDG.
COT    Cotangent of real argument, cot \( x \). Double precision version is DCOT.
CTAN   Tangent of complex argument, tan \( z \).
DACOS  Arc cosine of double precision argument, \( \cos^{-1} d \).
DASIN  Arc sine of double precision argument, \( \sin^{-1} d \).
DTAN   Double precision tangent, tan \( d \).
SIN    Sine of real argument, sin \( x \).
SINDG  Computes the Sine of an argument in degrees. Double precision version is DSINDG.

NAG subprogram library

S07AAE Tan \( (x) \). Double precision version is S07AAF.
S09AAE Arcsin \( (x) \). Double precision version is S09AAF.
S09ABE Arccos \( (x) \). Double precision version is S09ABF.

PORT subprogram library

ARCOS Computes arccos \( (x) \), answer in radians. Double precision version is DARCOS.
ARSIN Computes arcsin \( (x) \), answer in radians. Double precision version is DARSIN.
TAN Computes the elementary tangent function. If your Fortran library includes this function, use that instead. Double precision version is DTAN.

C4b: Exponential, logarithmic

CMLIB subprogram library (FNLIB sublibrary)

ALNREL  Ln \( (1 + x) \). Double precision version is DLNREL.
ALOG    Ln \( (x) \).
ALOG10  \( \log_{10} x \).
CEXPRL  Relative error exponential of complex argument from first order, \( (e^z - 1)/z \).
CLNREL  Relative error logarithm of complex argument, \( \ln(1+z) \).
CLOG10  Common logarithm of complex argument, \( \log_{10} z \).
EXP     Exponential function, \( = e^x \).
EXPREL  Relative error exponential from first order, \( = (e^x - 1)/x \). Double precision version is DEXPRL.

PORT subprogram library

CDEXP  Computes \( e^x \) for complex double precision \( z \). Complex double precision numbers represented as a double precision array of two elements.
CDLOG  Computes Ln \( (z) \) for complex double precision \( z \). Complex double precision numbers are represented as
a double precision array of two elements.

### C4c : Hyperbolic, inverse hyperbolic

**CMLIB subprogram library (PNLIB sublibrary)**

- **ACOSH** Hyperbolic cosine cosh x. Double precision version is DACOSH.
- **ASINH** Hyperbolic sine sinh x. Double precision version is DASINH.
- **ATANH** Arc hyperbolic tangent tanh\(^{-1}\)x. Double precision version is DATANH.
- **CACOSH** Arc hyperbolic cosine of complex argument, cosh\(^{-1}\)z.
- **CASINH** Arc hyperbolic sin of complex argument, sinh\(^{-1}\)z.
- **CATANH** Arc hyperbolic tangent of complex argument, tanh\(^{-1}\)z.
- **CCOSH** Hyperbolic cosine of complex argument, cosh z.
- **CSINH** Hyperbolic sine of complex argument, sinh z.
- **CTANH** Hyperbolic tangent of complex argument, tanh z.

**NAG subprogram library**

- **S10AAE** Tanh(x). Double precision version is S10AAF.
- **S10ABE** Sinh(x). Double precision version is S10ABF.
- **S10ACE** Cosh(x). Double precision version is S10ACF.
- **S11AAE** Arctanh(x). Double precision version is S11AAF.
- **S11ABE** Arcsinh(x). Double precision version is S11ABF.
- **S11ACE** Arccosh(x). Double precision version is S11ACF.

**PORT subprogram library**

- **ACOSH** Computes hyperbolic arccosine, arccosh(x). Double precision version is DACOSH.
- **ASINH** Computes hyperbolic arcsine, arcsin(x). Double precision version is DASINH.
- **ATANH** Computes hyperbolic arctangent, arctanh(x). Double precision version is DATANH.
- **COSH** Computes hyperbolic cosine, cosh(x). Double precision version is DCOSH.
- **SINH** Computes hyperbolic sin, sinh(x). Double precision version is DSINH.
- **TANH** Computes hyperbolic tangent, tanh(x). Double precision version is DTANH.

### C4d : Integrals of elementary transcendental functions

### C5 : Exponential and logarithmic integrals

**CMLIB subprogram library (AMOSLIB sublibrary)**

- **EXINT** Computes sequences of exponential integrals \(E_{N+K}(x)\), \(K=0,\ldots,M-1\) or \(e^x\) times same to specified tolerance. Double precision version is DEXINT.

**CMLIB subprogram library (PNLIB sublibrary)**

- **ALI** \(\int_0^x 1/\ln t \, dt\). Double precision version is DLI.
- **E1** Exponential integral, \(\int_x^\infty e^{-t}/t \, dt\). Double precision version is DEI.
- **EI** Exponential integral, \(\int_\infty^x e^{-t}/t \, dt\). Double precision version is DEI.
SPENC  Spence Dilogarithm, \(- \int_0^y \ln|1-y|/y \, dy\). Double precision version is DSPENC.

IMSL subprogram library

MMDEI  Exponential integrals.

MMDEN  Exponential integrals of integer order for real argument \(x\) scaled by \(e^x\).

MATHWARE subprogram library (STEGUN sublibrary)

EXPINT  Computes exponential integral \(E_n(x)\). Change computer by changing one line.

SICIEI  Computes sine, cosine, exponential integral as well as hyperbolic sin, cosine, exponential integral.

NAG subprogram library

S13AAE  Exponential integral, \(E_1(x)\). Double precision version is S13AAF.

C8: Cosine and sine integrals

MATHWARE subprogram library (STEGUN sublibrary)

SICIEI  Computes sine, cosine, exponential integral as well as hyperbolic sin, cosine, exponential integral.

NAG subprogram library

S13ACE  Cosine integral, \(ci(x)\). Double precision version is S13ACF.

S13ADE  Sine integral, \(si(x)\). Double precision version is S13ADF.

C7: Gamma

C7a: Gamma, log gamma, reciprocal gamma

CMLIB subprogram library (AMOSLIB sublibrary)

GAMLN  Computes \(\ln|\Gamma(z)|\) for non-negative \(z\).

CMLIB subprogram library (PNLIB sublibrary)

ALGAMS  \(\ln|\Gamma(z)|\), with sign of \(\Gamma(z)\). Double precision version is DLGAMS.

ALNGAM  \(\ln|\Gamma(z)|\). Double precision version is DLNGAM.

CGAMMA  Gamma function of complex argument, \(\Gamma(z)\).

CGAMR  Reciprocal gamma function of complex argument, \(1/\Gamma(z)\).

CLNGAM  Log gamma of complex argument, \(\ln\Gamma(z)\).

GAMMA  \(\Gamma(z)\). Double precision version is DGAMMA.

GAMR  Reciprocal gamma function, \(= 1/\Gamma(z)\). Double precision version is DGAMR.

POCH  Pochhammer's symbol \((a)_z = \Gamma(a + z)/\Gamma(a)\). Double precision version is DPOCH.

POCH1  Pochhammer's symbol from first order, \(=((a)_z - 1)/z\). Double precision version is DPOCH1.

IMSL subprogram library

ALGAMA  Evaluate \(\ln|\Gamma(z)|\).

GAMMA  Evaluate \(\Gamma(z)\).

NAG subprogram library
S14AAE  Gamma function. Double precision version is S14AAF.
S14ABE  Log gamma function. Double precision version is S14ABF.

**C7b:** Beta, log beta

<table>
<thead>
<tr>
<th>Subprogram Library (FNLIB Sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBETA</td>
</tr>
<tr>
<td>BETA</td>
</tr>
<tr>
<td>CBETA</td>
</tr>
<tr>
<td>CLBETA</td>
</tr>
</tbody>
</table>

**C7c:** Psi function

<table>
<thead>
<tr>
<th>Subprogram Library (FNLIB Sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPSI</td>
</tr>
<tr>
<td>PSI</td>
</tr>
<tr>
<td>MMPSI</td>
</tr>
</tbody>
</table>

**C7d:** Polygamma function

**C7e:** Incomplete gamma

<table>
<thead>
<tr>
<th>Subprogram Library (FNLIB Sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMI</td>
</tr>
<tr>
<td>GAMIC</td>
</tr>
<tr>
<td>GAMIT</td>
</tr>
</tbody>
</table>

**C7f:** Incomplete beta

<table>
<thead>
<tr>
<th>Subprogram Library (FNLIB Sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETAI</td>
</tr>
</tbody>
</table>

**C7g:** Riemann zeta

**C8:** Error functions
C8a: Error functions, their inverses, integrals, including the normal distribution function

**CMLIB subprogram library (FNLIB sublibrary)**

**ERF**  Error function, \( = 2/\sqrt{\pi} \int_0^x \exp(-t^2) \, dt \). Double precision version is DERF.

**ERFC** Complementary error function, \( = 2/\sqrt{\pi} \int_x^\infty \exp(-t^2) \, dt \). Double precision version is DERFC.

**IMSL subprogram library**

**ERF** Evaluate the error function. Note: the Fortran mathematical subroutine libraries may also contain ERF.

**ERFC** Evaluate the complementary error function.

**MDNOR** Normal or Gaussian probability distribution function.

**MERFCI** Inverse complementary error function.

**MERFI** Inverse error function.

**MERRCZ** \( \exp(-z^2) \text{erfc}(-iz) \) for complex \( z \).

**MATHWARE subprogram library (STEGUN sublibrary)**

**ERRINT** Computes error function and complementary error function to maximum machine accuracy. To change computers change one line.

**NAG subprogram library**

**S15ABE** Cumulative normal distribution function, \( P(x) \). Double precision version is S15ABF.

**S15ACE** Complement of cumulative normal distribution function, \( Q(x) \). Double precision version is S15ACF.

**S15ADE** Complement of error function, \( \text{erfc}(x) \). Double precision version is S15ADF.

**S15AEE** Error function, \( \text{erf}(x) \). Double precision version is S15AEF.

C8b: Fresnel integrals

**NAG subprogram library**

**S20ACE** Fresnel integrals, \( S(x) \). Double precision version is S20ACF.

**S20ADE** Fresnel integrals, \( C(x) \). Double precision version is S20ADF.

C8c: Dawson's integral

**CMLIB subprogram library (FNLIB sublibrary)**

**DAWS** Dawson's function, \( F(z) = \exp(-z^2) \int_0^x \exp(t^2) \, dt \). Double precision version is DDAWS.

**IMSL subprogram library**

**MMDAS** Dawson integral.

**NAG subprogram library**

**S15AFE** Dawson's integral. Double precision version is S15AFF.
C9 : Legendre functions

CMLIB subprogram library (FCNPAK sublibrary)

XSLEGF Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. Double precision version is XDLEGF.

XSNRMP Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. Double precision version is XDNRMP.

C10 : Bessel functions

C10a : J, Y, H(1), H(2)

C10a1 : Real argument, integer order

CMLIB subprogram library (FNLIB sublibrary)

BESJ0 J_0(z). Double precision version is DBESJ0.
BESJ1 J_1(z). Double precision version is DBESJ1.
BESY0 Y_0(z). Double precision version is DBESY0.
BESY1 Y_1(z). Double precision version is DBESY1.

IMSL subprogram library

MMBSJ0 J_0(z).
MMBSJ1 J_1(z).
MMBSJN J_n(z) \ n = 0, 1, ..., N

NAG subprogram library

S17ACE Y_0(z). Double precision version is S17ACF.
S17ADE Y_1(z). Double precision version is S17ADF.
S17AEE J_0(z). Double precision version is S17AEF.
S17AFE J_1(z). Double precision version is S17AFF.

PORT subprogram library

BESRJ J_n(z) \ n = 0, 1, ..., N. Double precision version is DBESRJ.

C10a2 : Complex argument, integer order

CMLIB subprogram library (AMOSLIB sublibrary)

CJYHBS Computes J_0, J_1, Y_0, Y_1, H_0, H_1 (Bessel and Struve functions) of complex argument.

IMSL subprogram library

MMBZJN J_n(z), \ n = 0, 1, ..., N.

PORT subprogram library

BESCJ J_n(z), \ n = 0, 1, ..., N. Double precision version is DBESCJ.
C10a3 : Real argument, real order

CMLIB subprogram library (AMOSLIB sublibrary)

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BESJ</td>
<td>$J_{\alpha+n}(x)$, $n = 1, \ldots, N, \alpha, x \geq 0$. Uses internal double precision arithmetic.</td>
</tr>
</tbody>
</table>

IMSL subprogram library

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMBSJR</td>
<td>$J_{n+\alpha}(z)$, $n = 0, 1, \ldots, N, \alpha, z \geq 0$.</td>
</tr>
<tr>
<td>MMBSYN</td>
<td>$Y_{n+\alpha}(z)$, $n = 0, 1, \ldots, N, \alpha, z \geq 0$.</td>
</tr>
</tbody>
</table>

C10a4 : Complex argument, real order

C10a5 : Complex argument, complex order

C10b : I, K

C10b1 : Real argument, integer order

CMLIB subprogram library (FNLIB sublibrary)

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BESI0</td>
<td>$I_0(z)$. Double precision version is DBESI0.</td>
</tr>
<tr>
<td>BESI0E</td>
<td>$\exp(-</td>
</tr>
<tr>
<td>BESI1</td>
<td>$I_1(z)$. Double precision version is DBESI1.</td>
</tr>
<tr>
<td>BESI1E</td>
<td>$\exp(-</td>
</tr>
<tr>
<td>BESK0</td>
<td>$K_0(z)$. Double precision version is DBESK0.</td>
</tr>
<tr>
<td>BESK0E</td>
<td>$\exp(z)K_0(z)$. Double precision version is DBSK0E.</td>
</tr>
<tr>
<td>BESK1</td>
<td>$K_1(z)$. Double precision version is DBESK1.</td>
</tr>
<tr>
<td>BESK1E</td>
<td>$\exp(z)K_1(z)$. Double precision version is DBSK1E.</td>
</tr>
</tbody>
</table>

IMSL subprogram library

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMBSIN</td>
<td>$I_n(z)$, $n = 0, 1, \ldots, N$.</td>
</tr>
<tr>
<td>MMBSK0</td>
<td>$K_0(z)$.</td>
</tr>
<tr>
<td>MMBSK1</td>
<td>$K_1(z)$.</td>
</tr>
</tbody>
</table>

NAG subprogram library

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S18ACE</td>
<td>$K_0(z)$. Double precision version is S18ACF.</td>
</tr>
<tr>
<td>S18ADE</td>
<td>$K_1(z)$. Double precision version is S18ADF.</td>
</tr>
<tr>
<td>S18AEE</td>
<td>$I_0(z)$. Double precision version is S18AEEF.</td>
</tr>
<tr>
<td>S18AFE</td>
<td>$I_1(z)$. Double precision version is S18AFF.</td>
</tr>
<tr>
<td>S18CCE</td>
<td>$\exp(z)K_0(z)$. Double precision version is S18CCF.</td>
</tr>
<tr>
<td>S18CDE</td>
<td>$\exp(z)K_1(z)$. Double precision version is S18CDF.</td>
</tr>
<tr>
<td>S18CEE</td>
<td>$\exp(-</td>
</tr>
<tr>
<td>S18CFE</td>
<td>$\exp(-</td>
</tr>
</tbody>
</table>
PORT subprogram library

BESRI  \( I_n(z), \quad n = 0, 1, \ldots, N. \) Double precision version is DBESRI.

C10b2 : Complex argument, integer order

IMSL subprogram library

MMBZIN  \( I_n(z), \quad n = 0, 1, \ldots, N. \)

PORT subprogram library

BESCI  \( I_n(z), \quad n = 0, 1, \ldots, N. \) Double precision version is DBESCI.

C10b3 : Real argument, real order

CMLIB subprogram library (FNLIB sublibrary)

BESKES  \( \exp(z)K_{n+\alpha}(z), \quad n = 0, 1, \ldots, N \) if \( N \geq 0, -1 < \alpha < 1 \), or \( \exp(z)K_{\alpha-n}(z), \quad n = 0, 1, \ldots, |N| \) if \( N < 0, -1 < \alpha < 1 \). Double precision version is DBSKES.

BESKS  \( K_{n+\alpha}(z), \quad n = 0, 1, \ldots, N \) if \( N \geq 0, -1 < \alpha < 1 \), or \( K_{\alpha-n}(z), \quad n = 0, 1, \ldots, |N| \) if \( N < 0, -1 < \alpha < 1 \). Double precision version is DBESKS.

IMSL subprogram library

MMBSIR  \( I_{n+\alpha}(z), \quad n = 0, 1, \ldots, N, \quad \alpha, z \geq 0. \) Exponential scaling optional.

MMBSKR  \( K_{n+\alpha}(z), \quad n = 0, 1, \ldots, N, \quad \alpha, z \geq 0. \) Exponential scaling optional.

C10b4 : Complex argument, real order

C10b5 : Complex argument, complex order

C10c : Kelvin functions

IMSL subprogram library

MMKEL0  ber_0(z), beio_0(z), ker_0(z), kei_0(z).

MMKEL1  ber_1(z), ber_1(z), ker_1(z), kei_1(z).

MMKELD  ber'_0(z), beio'_0(z), ker'_0(z), kei'_0(z).

C10d : Airy and Scorer functions

CMLIB subprogram library (FNLIB sublibrary)

AI  \( Ai(z). \) Double precision version is DAI.

AIE  \( Ai(z), \) exponentially scaled. Double precision version is DAIE.

BI  \( Bi(z). \) Double precision version is DBI.

BIE  \( Bi(z), \) exponentially scaled. Double precision version is DBIE.
NAG subprogram library

S17AGE \(Ai(z)\). Double precision version is S17AGF.
S17AHE \(Bi(z)\). Double precision version is S17AHF.
S17AJE \(Ai'(z)\). Double precision version is S17AJF.
S17AKE \(Bi'(z)\). Double precision version is S17AKF.

C10e : Struve, Anger, and Weber functions

CMLIB subprogram library (AMOSLIB sublibrary)

CJYHBS Computes \(J_0, J_1, Y_0, Y_1, H_0, H_1\) (Bessel and Struve functions) of complex argument.

C10f : Integrals of Bessel functions

C11 : Confluent hypergeometric functions

CMLIB subprogram library (FNLIB sublibrary)

CHU \(U(a, b, z)\). Double precision version is DCHU.

C12 : Coulomb wave functions

C13 : Jacobian elliptic functions, theta functions

C14 : Elliptic integrals

IMSL subprogram library

MMDELE Complete elliptic integral of the second kind.
MMDELK Complete elliptic integral of the first kind.
MMLIND Complete or incomplete elliptic integral of the second kind.
MMLINF Complete or incomplete elliptic integral of the first kind.
MMLINJ Complete or incomplete elliptic integral of the third kind.

NAG subprogram library

S21BAE Degenerate elliptic integrals of 1st kind, primarily for use in computing nondegenerate elliptic integrals and certain elementary circular and hyperbolic functions. Double precision version is S21BAF.
S21BBE Elliptic integrals of 1st kind (complete or incomplete). Double precision version is S21BBF.
S21BCE Elliptic integrals of 2nd kind (complete or incomplete). Double precision version is S21BCF.
S21BDE Elliptic integrals of 3rd kind (complete or incomplete). Double precision version is S21BDF.

CMLIB subprogram library (FCNPAK sublibrary)

RC Degenerate elliptic integrals of 1st kind, primarily for use in computing nondegenerate elliptic integrals and certain elementary circular and hyperbolic functions.
<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>Elliptic integrals of 1st kind (complete or incomplete).</td>
</tr>
<tr>
<td>RD</td>
<td>Elliptic integrals of 2nd kind (complete or incomplete).</td>
</tr>
<tr>
<td>RJ</td>
<td>Elliptic integrals of 3rd kind (complete or incomplete).</td>
</tr>
</tbody>
</table>

**C15 :** Weierstrass elliptic functions

**IMSL subprogram library**

**MMWPL**  
Weierstrass p-function in the lemniscatic case for complex argument with unit period parallelogram.

**MMWPL1**  
First derivative of the Weierstrass p-function in the lemniscatic case for complex argument with unit period parallelogram.

**MMWPQ**  
Weierstrass p-function in the equianharmonic case for complex argument with unit period parallelogram.

**MMWPQ1**  
First derivative of the Weierstrass p-function in the equianharmonic case for complex argument with unit period parallelogram.

**C16 :** Parabolic cylinder functions

**C17 :** Mathieu functions

**C18 :** Spheroidal wave functions

**C19 :** Other special functions
D: Linear Algebra

The most widely used constructs in mathematical computation are, undoubtedly, matrices and vectors. Consequently, it is not surprising to find a rich selection of software designed to perform operations such as computing solutions of linear systems, eigenvalues, determinants, and inverses. While it may seem that there are many routines that perform essentially the same task, one will find that many of these routines are designed to take advantage of special features of a given problem. Thus, to select the most appropriate routine one should determine as many attributes of one's matrix as possible. Some important attributes are given below.

1. The entries of the matrix are either real or complex numbers.
2. Special symmetry properties are important. If the matrix is real, it is called symmetric when the elements satisfy $a_{ij} = a_{ji}$. If the matrix is complex, it is called Hermitian when the elements $a_{ij}$ of the matrix $A$ satisfy $a_{ij} = \overline{a_{ji}}$, complex conjugate of $a_{ij}$.
3. The location and density of the non-zero entries of the matrix are important attributes. With no other knowledge of the matrix one would say the matrix is dense, i.e. that most of the entries are non-zero. A matrix is called sparse if there are relatively few non-zero entries. The cross-over point between dense and sparse is difficult to determine in practice.
4. Of importance among sparse matrices are band matrices wherein all the non-zero entries are concentrated on a band near the main diagonal of the matrix. Such matrices commonly arise when differential equations are discretized. Tridiagonal matrices are band matrices those in which the band consists only of the main diagonal and the closest lower and upper diagonals.
5. Hermitian and symmetric matrices may also have another important attribute—positive definiteness. A matrix $A$ is positive definite if $x^T A x > 0$ for all $x > 0$. This condition may be difficult to determine in practice without some independent knowledge of the matrix. See one of the references for more information on this attribute.

Most of the software classified in chapter D are Fortran subprograms. The notable exception is MATLAB, an interactive system for matrix analysis. In MATLAB one can solve linear systems of equations, determine eigenvalues, compute inverses, and solve least squares problems using simple interactive commands which operate on vectors and matrices.

January 1984

References


---

**D : Linear Algebra**

**MATLAB**

- **MATLAB** An interactive system for defining and manipulating matrices. It includes solving linear systems, linear least squares, eigenvalue and eigenvector calculation, QR decomposition, singular value decomposition and inverses, as well as other interesting and useful features.

**MINITAB**

- **MINITAB** Minitab's vector and matrix commands include COPY (vectors, vectors to matrices, and conversely), DIAGONAL (create a diagonal matrix or extract the diagonal of a matrix), TRANSPOSE, INVERSE, and EIGEN (calculate eigenvalues and eigenvectors for a symmetric matrix).
D1: Elementary vector and matrix operations

The subprograms in this class provide automatic computation of common elementary vector and matrix operations, such as the dot (inner) product of two vectors or the product of two matrices. These are provided to reduce the time and effort required to code problems. In addition, they can also help in making a program more portable while at the same time achieving high efficiency on a wide variety of machines. For example, the Basic Linear Algebra Subprograms (BLAS) available in CMLIB are widely distributed and are often implemented in machine language to increase their efficiency. Since these operations are often found in the innermost loops of large computations the resulting savings can often be significant.

January 1984

References


D1 : Elementary vector and matrix operations

D1a : Elementary vector operations

D1a1 : Set to constant

**DATAPAC subprogram library**

**DEFINE** Defines a vector of constants by setting all of the elements in the single precision vector X equal to XNEW.

**NAG subprogram library**

**F01CQE** Sets the elements of a vector to zero. Double precision version is F01CQF.

**PORT subprogram library**

**SETC** Set a specified number of values in a complex array equal to a constant.

**SETI** Set a specified number of values in an integer array equal to a constant.

**SETL** Set a specified number of values in a logical array equal to a constant.

**SETR** Set a specified number of values in a real array equal to a constant. Double precision version is SETD.

D1a2 : Minimum and maximum components

**CMLIB subprogram library (BLAS sublibrary)**

**ICAMAX** Find smallest index of maximum magnitude component of a complex vector.

**ISAMAX** Find smallest index of maximum magnitude component of a single precision vector. Double precision version is IDAMAX.

**CMLIB subprogram library (XBLAS sublibrary)**

**ISAMIN** Find the smallest index of the minimum magnitude component of a real vector.

**ISMAX** Find the smallest index of the maximum component of a real vector.

**ISMIN** Find the smallest index of the minimum component of a real vector.

**DATAPAC subprogram library**

**MAX** Computes the sample maximum of the data in the input vector X.
MIN
Computes the sample minimum of the data in the input vector X.

IMSL subprogram library

USMNMX
Determination of the minimum and maximum values of a vector.

PORT subprogram library

EXTRMI
Finds extremal points of an integer function defined on a mesh.

EXTRMR
Finds extremal points of a real function defined on a mesh. Double precision version is EXTRMD.

## D1a3: Norm

### D1a3a: L-1 (sum of magnitudes)

CMLIB subprogram library (BLAS sublibrary)

SASUM
Compute single precision sum of absolute values of components of vector. Double precision version is DASUM.

SCASUM
Compute complex sum of absolute values of components of vector.

IMSL subprogram library

VABSMF
Sum of the absolute values of the elements of a vector or a subset of a vector.

VABSMS
Sum of the absolute values of the elements of a row (or column) of a matrix stored in symmetric storage mode.

### D1a3b: L-2 (Euclidean norm)

CMLIB subprogram library (BLAS sublibrary)

SCNRNM2
Compute the Euclidean length or L2 norm of a complex vector.

SNRM2
Compute the Euclidean length or L2 norm of a single precision vector. Double precision version is DNRM2.

NAG subprogram library

F05ABE
Approximate 2-norm of a vector. Double precision version is F05ABF.

PORT subprogram library

SNRM2
Finds the length (Euclidean norm) of a vector, without underflow or overflow. Double precision version is DNRM2.

### D1a3c: L-infinity (maximum magnitude)

IMSL subprogram library

VABMXF
Maximum absolute value of the elements of a vector or a subset of the elements of a vector.

VABMXS
Maximum absolute value of the elements of a row or column of a matrix stored in symmetric storage mode.
### D1a4 : Dot product (inner product)

**CMLIB subprogram library (BLAS sublibrary)**

- **CDOTC**  Compute complex dot product using conjugated vector components.
- **CDOTU**  Compute complex dot product using unconjugated vector components.
- **DQDOTA** Compute dot product of 2 double precision vectors plus a double precision constant plus extended precision constant.
- **DQDOTI** Compute in extended precision dot product of 2 double precision vectors plus d.p. constant. Result is double precision.
- **DSDOT**  Compute single precision dot product using double precision accumulation.
- **SDOT**   Compute single precision dot product. Double precision version is DDOT.
- **SDSDOT** Compute single precision dot product and add a constant using double precision accumulation.

**CMLIB subprogram library (XBLAS sublibrary)**

- **CDCDOT** Computes complex precision dot product and adds a scalar. Uses double precision accumulation.
- **DCDOT**  Computes a complex precision dot product using double precision accumulation.

**IMSL subprogram library**

- **VIPRFF** Vector inner product of two vectors or subsets of two vectors.
- **VIPRSS** Vector inner product of two vectors each of which is part of some matrix stored in symmetric mode.

**NAG subprogram library**

- **F01DAE** Returns the sum of an initial value and a scalar product, using basic precision arithmetic. Double precision version is F01DAF.
- **F01DBE** Returns the sum of an initial value and a scalar product, using additional precision arithmetic. Double precision version is F01DBF.
- **F01DCE** Computes the value of a complex scalar product and subtracts it from a complex initial value, using basic precision arithmetic. Double precision version is F01DCF.
- **F01DDE** Computes the value of a complex scalar product and subtracts it from a complex initial value, using additional precision arithmetic. Double precision version is F01DDF.
- **F01DEE** Returns the value of the scalar product of two arrays of length N, using basic precision arithmetic. Double precision version is F01DEF.
- **X03AAE** Real inner product added to initial value, basic/additional precision. Double precision version is X03AAF.
- **X03ABE** Complex inner product added to initial value, basic/additional precision. Double precision version is X03ABF.

### D1a5 : Copy or exchange (swap)

**CMLIB subprogram library (BLAS sublibrary)**

- **CCOPY**  Copy a vector X to a vector Y, both complex.
- **CSWAP**  Interchange vectors X and Y, both complex.
- **SCOPY**  Copy a vector X to a vector Y, both single precision. Double precision version is DCOPY.
- **SSWAP**  Interchange vectors X and Y, both single precision. Double precision version is DSWAP.

**CMLIB subprogram library (XBLAS sublibrary)**

- **SCOPYM** Copies negative of array SX into array SY, with corresponding increments INCX and INCY.
DATAPAC subprogram library

COPY Copies the contents of the vector X into vector Y.
MOVE Copies M elements of the vector X (starting with position IX1) into the vector Y (starting with position IY1).

NAG subprogram library

F01CNE Copies a vector of length M into a row of a matrix. Double precision version is F01CNF.
F01CPE Copies the contents of a vector into a second vector. Double precision version is F01CPF.

PORT subprogram library

MOVEBC Move a complex vector using backward DO loop.
MOVEBI Move an integer vector using backward DO loop.
MOVEBL Move a logical vector using backward DO loop.
MOVEBR Move a real vector using backward DO loop. Double precision version is MOVEBD.
MOVEFC Move a complex vector using forward DO loops.
MOVEFD Move a double precision vector using forward DO loop.
MOVEFI Move an integer vector using forward DO loop.
MOVEFL Move a logical vector using forward DO loop.
MOVEFR Move a real vector using forward DO loop. Double precision version is MOVRFD.

**D1a6 : Multiplication by scalar**

CMLIB subprogram library (BLAS sublibrary)

CSCAL Compute complex constant times complex vector.
CSSCAL Compute real constant times complex vector.
SSCAL Compute a constant times a vector, both single precision. Double precision version is DSCAL.

MATHWARE subprogram library (NASHLIB sublibrary)

A11VS Standardizes a complex vector to have maximum component of magnitude equal to one.

**D1a7 : Triad (a x + y for vectors x,y and scalar a)**

CMLIB subprogram library (BLAS sublibrary)

CAXPY Compute a constant times a vector plus a vector, all complex.
SAXPY Compute a constant times a vector plus a vector, all single precision. Double precision version is DAXPY.

**D1a8 : Elementary rotation (Givens transformation)**

CMLIB subprogram library (BLAS sublibrary)

SROT Apply Givens plane rotation to a single precision vector. Double precision version is DROT.
SROTM Apply modified Givens plane rotation to single precision vector. Double precision version is DROTM.
**Dla9 :** Elementary reflection (Householder transformation)

**IMSL subprogram library**

**VHS12** Real Householder transformation computation and applications.

**Dla10 :** Convolutions

**IMSL subprogram library**

**VCONVO** Vector convolution.

**NAG subprogram library**

**C06ACE** Circular convolution of two real vectors of period $2^m$. Double precision version is C06ACF.

**Dlb :** Elementary matrix operations

**Dlb1 :** Set to zero, to identity

**NAG subprogram library**

**F01CAE** Sets elements of an $mn$ matrix $A$ to zero. Double precision version is F01CAF.

**F01CBE** Sets the elements $A(i,j)$ to one if $i=j$ and zero otherwise, where $1 \leq i \leq m$ and $1 \leq j \leq n$. Double precision version is F01CBF.

**Dlb2 :** Norm

**IMSL subprogram library**

**VNRMF1** 1-norm of matrices (full storage mode).

**VNRMF2** Euclidean-norm of matrices (full storage mode).

**VNRMFI** Infinity-norm matrices (full storage mode).

**VNRMS1** 1-norm of matrices (symmetric storage mode).

**VNRMS2** Euclidean-norm of matrices (symmetric storage mode).

**Dlb3 :** Transpose

**IMSL subprogram library**

**VTRAN** Transpose a rectangular matrix.

**NAG subprogram library**

**F01CRE** Re-orders the elements of a vector of length $mn$, containing an $mn$ matrix, $A$, so that the new vector contains the transpose matrix. Double precision version is F01CRF.
**D1b4**: Multiplication by vector

**NAG subprogram library**

F01CSE  Forms the product $c = Ab$ where $b$ is a vector and $A$ is a symmetric matrix whose lower triangle is stored by rows in a one-dimensional array. Double precision version is F01CSF.

**D1b5**: Addition, subtraction

**IMSL subprogram library**

VUABQ  Matrix addition (band + band symmetric matrices).
VUAFB  Matrix addition (full + band matrices).
VUAFFQ  Matrix addition (full + band symmetric matrices).
VUAFS  Matrix addition (full + symmetric matrices).
VUASB  Matrix addition (symmetric + band matrices).
VUASQ  Matrix addition (symmetric + band symmetric matrices).

**NAG subprogram library**

F01CDE  Adds elements of the $mn$ matrices $B$ and $C$ and stores the results in elements of the matrix $A$. Double precision version is F01CDF.
F01CEE  Subtracts elements of the matrix $C$ from elements of the matrix $B$ and stores the results in elements of the matrix $A$. Double precision version is F01CEF.
F01CGE  Adds elements of the matrix $B$ to elements in different positions in the matrix $A$. Double precision version is F01CGF.
F01CHE  Subtracts elements of the matrix $B$ from elements in a different position in the matrix $A$. Double precision version is F01CHF.

**D1b6**: Multiplication

**IMSL subprogram library**

VMULBB  Matrix multiplication (band storage mode).
VMULBF  Matrix multiplication (band by full matrices).
VMULBS  Matrix multiplication (band by symmetric matrices).
VMULFB  Matrix multiplication (full by band matrices).
VMULFF  Matrix multiplication (full storage mode).
VMULFM  Matrix multiplication of the transpose of matrix $a$ by matrix $b$ (full storage mode).
VMULFP  Matrix multiplication of matrix $a$ by the transpose of matrix $b$ (full storage mode).
VMULFQ  Matrix multiplication (full by band symmetric matrices).
VMULFS  Matrix multiplication (full by symmetric matrices).
VMULQB  Matrix multiplication (band symmetric by band matrices).
VMULQF  Matrix multiplication (band symmetric by full matrices).
VMULQQ  Matrix multiplication (band symmetric storage mode).
VMULQS  Matrix multiplication (band symmetric by symmetric matrices).
VMULSB  Matrix multiplication (symmetric by band matrices).
VMULSF  Matrix multiplication (symmetric by full matrices).
VMULSQ  Matrix multiplication (symmetric by band symmetric matrices).
VMULSS  Matrix multiplication (symmetric storage mode).
VTPROF  Transpose product of matrix (full storage mode).
VTPROS  Transpose product of a matrix (symmetric storage mode).

**NAG subprogram library**

**F01CKE**  Returns with the result of multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C. Double precision version is F01CKF.

**F01CLE**  Post-multiplies the matrix B with the transpose of the matrix C and places the result in the matrix A. Double precision version is F01CLF.

**D1b7 :**  Matrix polynomial

**IMSL subprogram library**

**VPOLYF**  Matrix polynomial (full storage mode).

**D1b8 :**  Copy

**NAG subprogram library**

**F01CFE**  Copies elements of the matrix B into different positions in the matrix A. Double precision version is F01CFF.

**F01CME**  Copies elements of one matrix into a second matrix. Double precision version is F01CMF.

**F01CNE**  Copies a vector of length M into a row of a matrix. Double precision version is F01CNF.

**D1b9 :**  Storage mode conversion

**IMSL subprogram library**

**VCVTBF**  Storage mode conversion of matrices (band to full storage mode).
**VCVTCH**  Storage mode conversion of matrices (full complex to Hermitian).
**VCVTFB**  Storage mode conversion of matrices (full to band storage mode).
**VCVTFQ**  Storage mode conversion (full to band symmetric storage mode).
**VCVTFS**  Storage mode conversion of matrices (full to symmetric).
**VCVTHC**  Storage mode conversion of matrices (Hermitian to full complex).
**VCVTQF**  Storage mode conversion (band symmetric to full storage mode).
**VCVTQS**  Storage mode conversion (band symmetric to symmetric storage mode).
**VCVTSF**  Storage mode conversion of matrices (symmetric to full).
**VCVTSQ**  Storage mode conversion (symmetric to band symmetric storage mode).

**D1b10 :**  Elementary rotation (Givens transformation)

**CMLIB subprogram library (BLAS sublibrary)**

**CROTG**  Construct Givens plane rotation of complex matrix.
D2i: Linear equations

SROTG  Construct Givens plane rotation of single precision matrix. Double precision version is DROTG.
SROTMG Construct modified Givens plane rotation of single precision matrix. Double precision version is DROTMG.

CMLIB subprogram library (XBLAS sublibrary)

CSROT Applies Givens plane rotation to complex matrix.

MATHWARE subprogram library (NASLIB sublibrary)

ROT N Plane rotation of two rows of a matrix.

D1b11: Elementary reflection (Householder transformation)

IMSL subprogram library

VHS12 Real Householder transformation computation and applications.
VHSH2C Complex Householder transformation to zero a single element of a matrix.
VHSH2R Real Householder transformation to zero a single element of a matrix.
VHS3R Real Householder transformation to zero two elements of a matrix.

D2: Solutions of systems of linear equations
(including Inversion, LU, and related decompositions)

Software in this class is concerned with the solution if nonsingular systems of linear equations

Ax = b

for an unknown vector x. (Software for solving problems where the solution is overdetermined or underdetermined, i.e., A is not square, can be found in class D9.) Matrix inversion is almost never the correct way to solve this problem. Routines specially designed for solving such systems are much to be preferred, both for speed and accuracy. On reflection one often will discover that the expression A⁻¹b is simply a convenient way of writing "the solution of the system Ax = b" and that one should do just that rather than form an inverse explicitly. One exception is the calculation of a matrix inverse in order to obtain covariances for some statistical applications.

Since operations involving matrices frequently form a major part of the computation time for a given problem, great effort has been expended to isolate the dominant part of a particular computation into a single routine in order to avoid unnecessary repetitions when possible. This has the somewhat annoying consequence that one may be required to call two routines to solve a single problem. However, in some cases this isolation can significantly reduce the computation time. As an example, suppose we want to solve the linear system above for many different vectors b. The procedure for solving would be two steps:

(a) the matrix A is factored as a product of two or three matrices of a special form independent of the vector b, and
(b) this factorization is used to solve for x given b.

If A is of order n, then generally step (a) requires computational time proportional to n^3, whereas step (b) requires only a time proportional to n^2. Hence, if one has many different right side vectors b, it is clearly more efficient to do step (a) only once. This situation occurs, for example, when solving time-dependent partial differential equations by implicit methods.

The Fortran subprogram libraries CMLIB, IMSL, and NAG each have large selections of programs for solving this problem. The principal public-domain package is LINPACK, which was developed at Argonne National Laboratory. Both LINPACK and a set of driver routines, called LINDRIVES, are available in CMLIB. These routines are classified according to the type of matrix A to which they may be applied, so it is important to determine as much about the special structure of a problem as possible in order to choose the best routine.

References


### D2:
Solution of systems of linear equations (including inversion, LU and related decompositions)

### D2a:
Real nonsymmetric matrices

#### D2a1:
General

**CMLIB subprogram library (LINDRIVES sublibrary)**

- **SGEFS** Factors and solves a general NXN single precision system of linear equations. Double precision version is DGEFS.

- **SGEIR** Factors and solves a general single precision system of linear equations and estimates solution accuracy (requires NxN extra storage).

**CMLIB subprogram library (LINPACKS sublibrary)**

- **SGECO** Computes LU factorization of real general matrix and estimates its condition. Double precision version is DGECO.

- **SGEDI** Uses LU factorization of real general matrix to compute its determinant and/or inverse. Double precision version is DGEDI.

- **SGEFA** Computes LU factorization of real general matrix. Double precision version is DGefa.

- **SGESL** Uses LU factorization of real general matrix to solve systems. Double precision version is DGESL.

- **SQRSL** Applies the output of SQRDC to compute coordinate transformations, projections, and least squares solutions (general real matrix). Double precision version is DQRSL.

**IMSL subprogram library**

- **LEQIF** Linear equation solution - full matrices (virtual memory version).

- **LEQOF** Linear equation solution - full matrices (out-of-core version).

- **LEQT1F** Linear equation solution - full storage mode - space economizer solution.

- **LEQT2F** Linear equation solution - full storage mode - high accuracy solution.

- **LIN1F** Inversion of a matrix - full storage mode space economizer solution.

- **LIN2F** Inversion of a matrix - full storage mode high accuracy solution.

- **LIN3F** In place inverse, equation solution, and/or determinant evaluation - full storage mode.

- **LUDATF** L-U decomposition by the Crout algorithm with optional accuracy test.

- **LUELNF** Elimination part of solution of Ax=b (full storage mode).

- **LUREFF** Refinement of solution to linear equations full storage mode.

**MATHWARE subprogram library (NASLIB sublibrary)**

- **A5GE** Gauss elimination with partial pivoting for solution of system of linear equations, Ax=f.

- **A6BS** Back substitution for the solution of a triangular system of linear equations, Rx=f.

**NAG subprogram library**

- **F01AAE** Calculates the approximate inverse of a real matrix by Crout's method with partial pivoting. Double precision version is F01AAF.
Dai space Solves Banded band Factors

PORT high band Factors CMLIB

Computes January

\[ \text{**F01BTE**} \quad \text{Decomposes a real matrix into product of triangular matrices LU by Gaussian elimination with partial pivoting. Block-column method used for efficiency on paged virtual machines. Double precision version is F01BTF.} \]

\[ \text{**F03AFE**} \quad \text{LU-factorisation and determinant, real matrix. Double precision version is F03AFF.} \]

\[ \text{**F04AAE**} \quad \text{Simultaneous linear equations (black box), real matrix, approximate solution, multiple right hand sides. Double precision version is F04AAF.} \]

\[ \text{**F04AEE**} \quad \text{Simultaneous linear equations (black box), real matrix, accurate solution, multiple right hand sides. Double precision version is F04AEF.} \]

\[ \text{**F04AHE**} \quad \text{Calculates the accurate solution of set of real linear equations with multiple right hand sides, } AX=B, \text{ where } A \text{ has been decomposed into triangular matrices using F03AFE. Double precision version is F04AHF.} \]

\[ \text{**F04AJE**} \quad \text{Calculates the approximate solution of set of real linear equations with multiple right hand sides, } AX=B, \text{ where } A \text{ has been decomposed into triangular matrices using F03AFE. Double precision version is F04AJF.} \]

\[ \text{**F04ARE**} \quad \text{Simultaneous linear equations (black box), real matrix, approximate solution, one right hand side. Double precision version is F04ARF.} \]

\[ \text{**F04ATE**} \quad \text{Simultaneous linear equations (black box), real matrix, accurate solution, one right hand side. Double precision version is F04ATF.} \]

\[ \text{**F04AYE**} \quad \text{Calculates the approximate solution of set of real linear equations with multiple right hand sides, } AX=B, \text{ where } A \text{ has been decomposed into triangular matrices using F01BTE. Double precision version is F04AYF.} \]

\[ \text{PORT subprogram library} \]

\[ \text{**LINEQ**} \quad \text{Solves a real system of linear equations, } AX=B, \text{ where } B \text{ is allowed to be a matrix or a vector. Double precision version is DLINEQ.} \]

| D2a2 | Banded |

\[ \text{CMLIB subprogram library (LINDRIVES sublibrary)} \]

\[ \text{**SNBCO**} \quad \text{Factors a real band matrix by Gaussian elimination and estimates condition of the matrix. Double precision version is DNBCO.} \]

\[ \text{**SNBFA**} \quad \text{Factors a single precision band matrix by elimination. Double precision version is DNBFA.} \]

\[ \text{**SNBFS**} \quad \text{Factors and solves a general nonsymmetric single precision banded system of linear equations. Double precision version is DNBSF.} \]

\[ \text{**SNBIR**} \quad \text{Factors and solves a general nonsymmetric single precision banded system of equations and estimates solution accuracy (needs } N \times (2ML+MU) \text{ extra storage).} \]

\[ \text{**SNBSL**} \quad \text{Solves a general nonsymmetric single precision banded system of linear equations using factors computed previously. Double precision version is DNBSL.} \]

\[ \text{CMLIB subprogram library (LINPACKS sublibrary)} \]

\[ \text{**SGBCO**} \quad \text{Computes LU factorization of real band matrix and estimates its condition. Double precision version is DGBCO.} \]

\[ \text{**SGBFA**} \quad \text{Computes LU factorization of real band matrix. Double precision version is DGBFA.} \]

\[ \text{**SGBSL**} \quad \text{Uses LU factorization of real band matrix to solve systems. Double precision version is DGBSL.} \]

\[ \text{IMSL subprogram library} \]

\[ \text{**LEQT1B**} \quad \text{Linear equation solution - band storage mode - space economizer solution.} \]

\[ \text{**LEQT2B**} \quad \text{Linear equation solution - band storage mode - high accuracy solution.} \]
January 1984  GAMS: Modules by Class  B 29

NAG subprogram library

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F01LBE</td>
<td>LU-factorisation, real band matrix. Double precision version is F01LBF.</td>
</tr>
<tr>
<td>F04LDE</td>
<td>Simultaneous linear equations (factorising the matrix of coefficients), real band matrix, approximate solution. Double precision version is F04LDF.</td>
</tr>
</tbody>
</table>

D2a2a : Tridiagonal

CMLIB subprogram library (LINPACKS sublibrary)

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGTSL</td>
<td>Factors a real tridiagonal matrix and simultaneously solves a system. Double precision version is DGTSL.</td>
</tr>
</tbody>
</table>

D2a3 : Triangular

CMLIB subprogram library (LINPACKS sublibrary)

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRCO</td>
<td>Estimates the condition of real triangular matrix. Double precision version is DTRCO.</td>
</tr>
<tr>
<td>STRDI</td>
<td>Computes determinant and/or inverse of real triangular matrix. Double precision version is DTRDI.</td>
</tr>
<tr>
<td>STRSL</td>
<td>Solves systems with real triangular matrix. Double precision version is DTRSL.</td>
</tr>
</tbody>
</table>

D2a4 : Sparse

CMLIB subprogram library (FSHPK sublibrary)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POIS3D</td>
<td>Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations in 3D.</td>
</tr>
<tr>
<td>POISTG</td>
<td>Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations.</td>
</tr>
</tbody>
</table>

CMLIB subprogram library (SLVBLK sublibrary)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLVBLK</td>
<td>Solves Ax=b where A is an almost block diagonal matrix. These arise in finite element or piecewise polynomial approximation.</td>
</tr>
</tbody>
</table>

CMLIB subprogram library (YSMP sublibrary)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDRV</td>
<td>Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting (compressed storage mode).</td>
</tr>
<tr>
<td>NDRV</td>
<td>Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting (uncompressed storage mode).</td>
</tr>
<tr>
<td>TDRV</td>
<td>Solves sparse systems of linear algebraic equations by Gaussian elimination without pivoting.</td>
</tr>
</tbody>
</table>

MATHWARE subprogram library (ITPACK sublibrary)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSI</td>
<td>Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection.</td>
</tr>
<tr>
<td>RSCG</td>
<td>Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive.</td>
</tr>
<tr>
<td>RSSI</td>
<td>Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive.</td>
</tr>
</tbody>
</table>
SOR Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection.

SSORCG Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection.

SSORSI Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration, adaptive parameter selection.

NAG subprogram library

F01BRE Decomposes real sparse matrix. Either forms LU-decomposition of permutation of entire matrix, or permutes matrix to block lower triangular form and then decomposes diagonal blocks. Double precision version is F01BRF.

F01BSE Decomposes a real sparse matrix using the pivotal sequence previously obtained by F01BRE when a matrix of the same sparsity pattern was decomposed. Double precision version is F01BSF.

F04AXE Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, $Ax = b$ or $A^T x = b$, where A has been decomposed by F01BRE or F01BSE. Double precision version is F04AXF.

D2b : Real symmetric matrices

D2b1 : General

D2b1a : Indefinite

CMLIB subprogram library (LINPACKS sublibrary)

SSICO Computes factorization of real symmetric indefinite matrix and estimates its condition. Double precision version is DSICO.

SSIDI Uses factorization of real symmetric indefinite matrix to compute its determinant and/or inverse. Double precision version is DSIDI.

SSIFA Computes factorization of real symmetric indefinite matrix. Double precision version is DSIFA.

SSISL Uses factorization of real symmetric indefinite matrix to solve systems. Double precision version is DSISL.

SSPCO Computes factorization of real symmetric indefinite matrix stored in packed form and estimates its condition. Double precision version is DSPCO.

SSPDI Uses factorization of real symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is DSPDI.

SSPFA Computes factorization of real symmetric indefinite matrix stored in packed form. Double precision version is DSPFA.

SSPSL Uses factorization of real symmetric indefinite matrix stored in packed form to solve systems. Double precision version is DSPSL.

IMSL subprogram library

LEQ1S Linear equation solution - indefinite matrix - symmetric storage mode - space economizer solution.

LEQ2S Linear equation solution - indefinite matrix - symmetric storage mode - high accuracy solution.
January 1984  GAMS: Modules by Class  B 31

D2b1b: Positive definite

CMLIB subprogram library (LINDRIVES sublibrary)

- SPOFS  Factors and solves a symmetric positive definite single precision system of linear equations. Double precision version is DPOFS.

SPOIR  Factors and solves a symmetric positive definite single precision system of equations and estimates solution accuracy (needs NxN extra storage).

CMLIB subprogram library (LINPACKS sublibrary)

SCHDC  Compute Cholesky decomposition of real positive definite matrix with optional pivoting. Double precision version is DCHDC.

SPOCO  Uses Cholesky algorithm to factor real positive definite matrix and estimate its condition. Double precision version is DPOCO.

SPODI  Uses factorization of real positive definite matrix to compute its determinant and/or inverse. Double precision version is DPODI.

SPOFA  Uses Cholesky algorithm to factor real positive definite matrix. Double precision version is DPOFA.

SPOSL  Uses factorization of real positive definite matrix to solve systems. Double precision version is DPOSL.

SPPCO  Uses Cholesky algorithm to factor real positive definite matrix stored in packed form and estimate its condition. Double precision version is DPPOCO.

SPPDI  Uses factorization of real positive definite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is DPPODI.

SPPFA  Uses Cholesky algorithm to factor real positive definite matrix stored in packed form. Double precision version is DPPOFA.

SPPSL  Uses factorization of real positive definite matrix stored in packed form to solve systems. Double precision version is DPPOSL.

IMSL subprogram library

LEQT1P  Linear equation solution - positive definite matrix - symmetric storage mode - space economizer solution.

LEQT2P  Linear equations solution - positive definite matrix - symmetric storage mode - high accuracy solution.

LINVP  Inversion of matrix - positive definite symmetric storage mode - space economizer solution.

LINV2P  Inversion of a matrix - positive definite symmetric storage mode - high accuracy solution.

LINV3P  In place inverse, equation solution, positive definite matrix - symmetric storage mode.

LUDECP  Decomposition of a positive definite matrix symmetric storage mode.

LUELMP  Elimination part of the solution of Ax=b positive definite matrix - symmetric storage mode.

LUREFP  Refinement of solution to linear equations positive definite matrix - symmetric storage mode.

MATHWARE subprogram library (NASHLIB sublibrary)

A24CG  Solution of a consistent system of linear equations with symmetric non negative definite coefficient matrix.

A7CH  Cholesky decomposition of symmetric non-negative definite matrix in compact storage.

A8CS  Cholesky back substitution for the solution of consistent sets of linear equations with symmetric coefficient matrices, compact storage.

A9GJ  Bauer Reinsch inversion of a positive definite symmetric matrix by a modification of Gauss-Jordan method.

NAG subprogram library

- F01ABE  Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky’s method and iterative refinement. (Simplified parameter list). Double precision version is F01ABF.
F01ACE Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. Double precision version is F01ACF.

F01ADE F01ADE calculates the approximate inverse of a real symmetric positive definite matrix by Cholesky's method. Double precision version is F01ADF.

F01BQE Forms the Cholesky decomposition of a real symmetric matrix G whose lower triangle only is stored. If G not positive definite, forms Cholesky decomposition of G+E, E a diagonal matrix. Double precision version is F01BQF.

F01BUE Decomposes a symmetric positive definite matrix into form ULDLT where U is unit upper triangular, L is unit lower triangular, D is diagonal. Precedes F01BVE. Double precision version is F01BUF.

F01BXE Performs the Cholesky factorization UDU of a real symmetric positive definite matrix A. Double precision version is F01BXF.

F03AEE LLT-factorisation and determinant, real symmetric positive-definite matrix. Double precision version is F03AEF.

F04ABE Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, multiple right hand sides. Double precision version is F04ABF.

F04AFE Calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. Double precision version is F04AFF.

F04AGE Calculates the approximate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. Double precision version is F04AGF.

F04AQE Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, Ax = b, where A has been decomposed into LDLT using F01BQE. (Economical storage.). Double precision version is F04AQF.

F04ASE Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, one right hand side. Double precision version is F04ASF.

F04AZE Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, Ax = b, where A has been decomposed into triangular matrices using F01BXE. Double precision version is F04AZF.

D2b2: Positive definite banded

CMLLIB subprogram library (LINPACKS sublibrary)

SPBCO Uses Cholesky algorithm to compute factorization of real positive definite band matrix and estimates its condition. Double precision version is DPBCO.

SPBFA Uses Cholesky algorithm to compute factorization of real positive definite band matrix. Double precision version is DPBFA.

SPBSL Uses factorization of real positive definite band matrix to solve systems. Double precision version is DPBSL.

IMSL subprogram library

LEQ1PB Linear equation solution - positive definite symmetric band matrix - band symmetric storage mode - space economizer solution.

LEQ2PB Linear equation solution - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution.

LIN1PB Inversion of a matrix - positive definite band symmetric matrix - band symmetric storage mode - space economizer solution.

LIN2PB Inversion of matrix - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution.
<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUDAPB</td>
<td>Decomposition of a positive definite band symmetric matrix - band symmetric storage mode.</td>
</tr>
<tr>
<td>LUELPB</td>
<td>Elimination part of solution of ( Ax = b ) positive definite band symmetric matrix band symmetric storage mode.</td>
</tr>
<tr>
<td>LUREPB</td>
<td>Refinement of solution to linear equations positive definite band symmetric matrix band symmetric storage mode.</td>
</tr>
</tbody>
</table>

**NAG subprogram library**

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F01MCE</td>
<td>LDLt-factorisation, real symmetric positive-definite variable-bandwidth matrix. Double precision version is F01MCF.</td>
</tr>
<tr>
<td>F03AGE</td>
<td>LLt-factorisation and determinant, real symmetric positive-definite band matrix. Double precision version is F03AGF.</td>
</tr>
<tr>
<td>F04ACE</td>
<td>Simultaneous linear equations (black box), real symmetric positive-definite band matrix, approximate solution, multiple right hand sides. Double precision version is F04ACF.</td>
</tr>
<tr>
<td>F04ALE</td>
<td>Calculates the approximate solution of a set of real symmetric positive definite band linear equations with multiple right hand sides, ( AX = B ), where ( A ) has been decomposed into triangular matrices using F03AGE. Double precision version is F04ALF.</td>
</tr>
<tr>
<td>F04MCE</td>
<td>Calculates the approximate solution of a system of real linear equations with multiple right hand sides, ( Ax = B ), where ( A ) is a symmetric positive definite variable-bandwidth matrix, which has previously been factorised by F01MCE. Related systems may also be solved. Double precision version is F04MCF.</td>
</tr>
</tbody>
</table>

**CMLIB subprogram library (LINPACKS sublibrary)**

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPTSL</td>
<td>Decomposes real symmetric positive definite tridiagonal matrix and simultaneously solves a system. Double precision version is DPTSL.</td>
</tr>
</tbody>
</table>

**CMLIB subprogram library (YSMP sublibrary)**

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODRV</td>
<td>Computes the minimum degree ordering of equations and unknowns for a system of linear algebraic equations in sparse storage mode.</td>
</tr>
<tr>
<td>SDRV</td>
<td>Solves sparse symmetric systems of linear algebraic equations by Gaussian elimination without pivoting.</td>
</tr>
</tbody>
</table>

**MATHWARE subprogram library (ITPACK sublibrary)**

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSI</td>
<td>Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection.</td>
</tr>
<tr>
<td>RSCG</td>
<td>Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive.</td>
</tr>
<tr>
<td>RSSI</td>
<td>Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive.</td>
</tr>
<tr>
<td>SOR</td>
<td>Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection.</td>
</tr>
<tr>
<td>SSORCG</td>
<td>Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection.</td>
</tr>
<tr>
<td>SSORSI</td>
<td>Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration, adaptive.</td>
</tr>
</tbody>
</table>
adaptive parameter selection.

MATHWARE subprogram library (NASLIB sublibrary)

**A24CG** Solution of a consistent system of linear equations with symmetric non negative definite coefficient matrix.

**D2c** Complex non-Hermitian matrices

**D2c1** General

CMLIB subprogram library (LINDRIVES sublibrary)

• **CGEFS** Factors and solves a general complex system of linear equations.

• **CGEIR** Factors and solves a general complex system of linear equations and provides estimate of accuracy of the solution (needs NXN extra storage).

CMLIB subprogram library (LINPACKC sublibrary)

**CGECO** Compute LU factorization of general complex matrix and estimate its condition.

**CGEDI** Compute determinant and/or inverse of general complex matrix from its LU factors.

**CGEFA** Compute LU factorization of general complex matrix.

**CGESL** Use LU factorization of general complex matrix to solve systems.

**CQRSL** Applies the output of CQRDC to compute coordinate transformations, projections, and least squares solutions (general complex matrix).

IMSL subprogram library

**LEQ2C** Linear equation solution - complex matrix high accuracy solution.

**LEQT1C** Matrix decomposition, linear equation solution - space economizer solution complex matrices.

NAG subprogram library

**F03AHE** LU-factorisation and determinant, complex matrix. Double precision version is F03AHF.

• **F04ADE** Simultaneous linear equations (black box), complex matrix, approximate solution, multiple right hand sides. Double precision version is F04ADF.

**F04AKE** Calculates the approximate solution of a set of complex linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AHE. Double precision version is F04AKF.

PORT subprogram library

**CLINQ** Solves a complex system of linear equations. Coefficient matrix must be input as two real matrices. Double precision version is DCLINQ.

**D2c2** Banded

CMLIB subprogram library (LINDRIVES sublibrary)

**CNBCO** Factors a complex band matrix by Gaussian elimination and estimates its condition number.

**CNBFA** Factors a non-symmetric complex band matrix by elimination.

• **CNBFS** Factors and solves a general complex band matrix system of linear equations.

**CNBIR** Factors and solves a general nonsymmetric complex band system of equations and estimates accuracy.

* Denotes easy-to-use modules
of the solution (requires $N(2ML+MU)$ extra storage).

**CNBSL**  Solves the nonsymmetric complex band system of equations using factors previously computed.

**CMLIB subprogram library (LINPACKC sublibrary)**

**CGBCO**  Compute LU factorization of complex band matrix and estimate its condition.

**CGBFA**  Compute LU factorization of general complex band matrix.

**CGBSL**  Uses LU factorization of complex band matrix to solve systems.

**D2c2a :**  Tridiagonal

**CMLIB subprogram library (LINPACKC sublibrary)**

**CGTSL**  Solves systems with general complex tridiagonal matrix.

**D2c3 :**  Triangular

**CMLIB subprogram library (LINPACKC sublibrary)**

**CTRCO**  Estimates condition of complex triangular matrix.

**CTRDI**  Computes determinant and/or inverse of complex triangular matrix.

**CTRSL**  Solves systems with complex triangular matrix.

**D2c4 :**  Sparse

**D2d :**  Complex Hermitian matrices

**D2d1 :**  General

**D2d1a :**  Indefinite

**CMLIB subprogram library (LINPACKC sublibrary)**

**CHICO**  Computes factorization of complex Hermitian indefinite matrix and estimates its condition.

**CHIDI**  Uses factorization of complex Hermitian indefinite matrix to compute its inertia, determinant, and/or inverse.

**CHIFA**  Computes factorization of complex Hermitian indefinite matrix.

**CHISL**  Uses factorization of complex Hermitian indefinite matrix to solve systems.

**CHPCO**  Computes factorization of complex Hermitian indefinite matrix stored in packed form and estimates its condition.

**CHPDI**  Uses factorization of complex Hermitian indefinite matrix stored in packed form to compute its inertia, determinant, and inverse.

**CHPFA**  Computes factorization of complex Hermitian indefinite matrix stored in packed form.

**CHPSL**  Uses factorization of complex Hermitian indefinite matrix stored in packed form to solve systems.

**CSICO**  Computes factorization of complex symmetric indefinite matrix and estimates its condition.
CSIDI Uses factorization of complex symmetric indefinite matrix to compute its determinant and/or inverse.
CSIFA Computes factorization of complex symmetric indefinite matrix.
CSISL Uses factorization of complex symmetric indefinite matrix to solve systems.
CSPCO Computes factorization of complex symmetric indefinite matrix stored in packed form and computes its condition.
CSPDI Uses factorization of complex symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse.
CSPFA Computes factorization of complex symmetric indefinite matrix stored in packed form.
CSPSL Uses factorization of complex symmetric indefinite matrix stored in packed form to solve systems.

D2d1b : Positive definite

CMLIB subprogram library (LINDRIVES sublibrary)
- CPOFS Factors and solves positive definite symmetric complex system of linear equations.
- CPOIR Solves positive definite Hermitian complex system of linear equations and estimates the accuracy of the solution (requires NxN extra storage).

CMLIB subprogram library (LINPACKC sublibrary)
- CCHDC Compute Cholesky decomposition of complex positive definite matrix with optional pivoting.
- CPOCO Uses Cholesky algorithm to compute factorization of complex positive definite matrix and estimates its condition.
- CPODI Uses factorization of complex positive definite matrix to compute its determinant and/or inverse.
- CPOFA Uses Cholesky algorithm to compute factorization of complex positive definite matrix.
- CPOSL Uses factorization of complex positive definite matrix to solve systems.
- CPPCO Uses Cholesky algorithm to factor complex positive definite matrix stored in packed form.
- CPPDI Uses factorization of complex positive definite matrix stored in packed form to compute determinant and/or inverse.
- CPPFA Uses Cholesky algorithm to factor complex positive definite matrix stored in packed form.
- CPPSL Uses factorization of complex positive definite matrix stored in packed form to solve systems.

NAG subprogram library
- F01BNE Performs the Cholesky decomposition of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. Double precision version is F01BNF.
- F01BPE Determines the inverse of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. Double precision version is F01BPF.
- F04AWE Calculates the approximate solutions of a set of complex linear equations with multiple right hand sides, AX = B, where A is positive definite Hermitian, following the Cholesky decomposition of A by F01BNE. Double precision version is F04AWF.

D2d2 : Positive definite banded

CMLIB subprogram library (LINPACKC sublibrary)
- CPBCO Uses Cholesky algorithm to compute factorization of complex positive definite band matrix and estimates its condition.
- CPBFA Uses Cholesky algorithm to compute factorization of complex positive definite band matrix.
CPBSL  Uses factorization of complex positive definite band matrix to solve systems.

D2d2a :  Tridiagonal

CMLIB subprogram library (LINPACKS sublibrary)

CPTSL  Solves systems with complex positive definite tridiagonal matrix.

D2d4 :  Sparse

D2e :  Associated operations (e.g., matrix reorderings)

D3: Determinants

The determinant of a square matrix is a very useful tool in mathematical theory. However, determinants are very dependent upon the scaling of matrix elements and hence are an unreliable indicator of the numerical difficulties which might arise in trying to solve some problem. Casual users should exercise caution in using determinants in the formulation and solution of problems. Listed below are a number of programs which calculate determinants carefully; they are organized according to matrix type as in class D2.

References

D3 :  Determinants

D3a :  Real nonsymmetric matrices

D3a1 :  General

CMLIB subprogram library (LINPACKS sublibrary)

SGEDI  Uses LU factorisation of real general matrix to compute its determinant and/or inverse. Double precision version is DGEDI.

IMSL subprogram library

LINV3F  In place inverse, equation solution, and/or determinant evaluation - full storage mode.

NAG subprogram library

• F03AAE  Determinant (black box), real matrix. Double precision version is F03AAF.
F03AFE  LU-factorisation and determinant, real matrix. Double precision version is F03AFF.
D3a2: Banded

CMLIB subprogram library (LINDRIVES sublibrary)

SNBDI Computes the determinant of a single precision band matrix using factors previously computed. Double precision version is DNBDI.

CMLIB subprogram library (LINPACKS sublibrary)

SGBDI Uses LU factorization of real band matrix to compute its determinant. (No provision for computing matrix inverse.) Double precision version is DGBDI.

D3a2a: Tridiagonal

D3a3: Triangular

CMLIB subprogram library (LINPACKS sublibrary)

STRDI Computes determinant and/or inverse of real triangular matrix. Double precision version is DTRDI.

D3a4: Sparse

D3b: Real symmetric matrices

D3b1: General

D3b1a: Indefinite

CMLIB subprogram library (LINPACKS sublibrary)

SSIDI Uses factorization of real symmetric indefinite matrix to compute its determinant and/or inverse. Double precision version is DSIDI.

SSPDI Uses factorization of real symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is DSPDI.

D3b1b: Positive definite

CMLIB subprogram library (LINPACKS sublibrary)

SPODI Uses factorization of real positive definite matrix to compute its determinant and/or inverse. Double precision version is DPODI.

SPPDI Uses factorization of real positive definite matrix stored in packed form to compute its determinant and/or inverse. Double precision version is DPPDI.

NAG subprogram library

- F03ABE Determinant (black box), real symmetric positive-definite matrix. Double precision version is F03ABF.
- F03AEF LLt-factorization and determinant, real symmetric positive-definite matrix. Double precision version is F03AEF.

* Denotes easy-to-use modules
D3b2 : Positive definite banded

CMLIB subprogram library (LINPACKS sublibrary)

SPBDI Uses factorization of real positive definite band matrix to compute its determinant. (No provision for matrix inverse.). Double precision version is DPBDI.

NAG subprogram library

• F03ACE Determinant (black box), real symmetric positive-definite band matrix. Double precision version is F03ACF.
F03AGE LLt-factorisation and determinant, real symmetric positive-definite band matrix. Double precision version is F03AGF.

D3b2a : Tridiagonal

D3b4 : Sparse

D3c : Complex non-Hermitian matrices

D3c1 : General

CMLIB subprogram library (LINPACKC sublibrary)

CGEDI Compute determinant and/or inverse of general complex matrix from its LU factors.

NAG subprogram library

• F03ADE Determinant (black box), complex matrix. Double precision version is F03ADF.
F03AHE LU-factorisation and determinant, complex matrix. Double precision version is F03AHF.

D3c2 : Banded

CMLIB subprogram library (LINDRIVES sublibrary)

CNBDI Computes the determinant of a complex band matrix from previously computed factors.

CMLIB subprogram library (LINPACKC sublibrary)

CGBDI Compute determinant of complex band matrix from its LU factors. (No provision for computing inverse directly.).

D3c2a : Tridiagonal

D3c3 : Triangular

CMLIB subprogram library (LINPACKC sublibrary)

CTRDI Computes determinant and/or inverse of complex triangular matrix.

• Denotes easy-to-use modules
The calculation of the eigenvalues and/or eigenvectors of a matrix has received much attention. It was for this class of problems that the first modern mathematical software package, EISPACK, was produced, and most of the routines in this class originated in that project. If you want to compute the eigenvalues and/or eigenvectors for one particular matrix,
you may want to use one of the drivers which invoke a series of EISPACK routines to perform this task. The IMSL and NAG libraries each have such a set of routines. Individual EISPACK routines, as well as a collection of public-domain drivers, called LICEPACK, are available in CMLIB. The EISPACK Guide (see reference below) has a flowchart that is designed to lead one through the maze of routines.

January 1984

References

D4 : Eigenvalues, eigenvectors

D4a : Ordinary eigenvalue problems \((Ax = \lambda x)\)

D4a1 : Real symmetric

CMLIB subprogram library (EISPACK sublibrary)

RS
Computes eigenvalues and, optionally, eigenvectors of a real symmetric matrix.

RSP
Compute eigenvalues and, optionally, eigenvectors of a real symmetric matrix packed into a one dimensional array.

CMLIB subprogram library (LICEPACK sublibrary)

• SSIEV
Computes the eigenvalues and, optionally, the eigenvectors of a real symmetric matrix.

• SSPEV
Computes eigenvalues and, optionally eigenvectors of real symmetric matrix stored in packed form.

IMSL subprogram library

EIGRS
Eigenvalues and (optionally) eigenvectors of a real symmetric matrix.

MATHWARE subprogram library (NASLIB sublibrary)

A13ESV
Solves eigenproblem for real symmetric matrix via Singular Value Decomposition.

A14JE
Jacobi algorithm for the eigenvalues and eigenvectors of a real symmetric matrix.

NAG subprogram library

• F02AAE
Real symmetric matrix, (black box), all eigenvalues. Double precision version is F02AAF.

• F02ABE
Real symmetric matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02ABF.

• F02BBE
Real symmetric matrix, (black box), selected eigenvalues and eigenvectors. Double precision version is F02BBF.

D4a2 : Real nonsymmetric

CMLIB subprogram library (EISPACK sublibrary)

RG
Computes eigenvalues and, optionally, eigenvectors of a real general matrix.

CMLIB subprogram library (LICEPACK sublibrary)

- Denotes easy-to-use modules.
- **SGEEV** Computes the eigenvalues and, optionally, the eigenvectors of a general real matrix.

  *IMSL subprogram library*

- **EIGRF** Eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode.

  *NAG subprogram library*

- **F02AFE** Real matrix, (black box), all eigenvalues. Double precision version is F02AFF.
- **F02AGE** Real matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02AGF.
- **F02BCE** Real matrix, (black box), selected eigenvalues and eigenvectors. Double precision version is F02BCF.

  *PORT subprogram library*

- **EIGEN** Finds all eigenvalues and eigenvectors of a real matrix. Output consists of pairs of real arrays. Double precision version is DEIGEN.

### D4n3 : Complex Hermitian

- **CMLIB subprogram library (EISPACK sublibrary)**

- **CH** Computes the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix.

  *CMLIB subprogram library (LICEPACK sublibrary)*

- **CHIEV** Computes the eigenvalues and, optionally, the eigenvectors of a complex hermitian matrix.

  *IMSL subprogram library*

- **EIGCH** Eigenvalues and (optionally) eigenvectors of a complex Hermitian matrix.

  *NAG subprogram library*

- **F02AWE** Complex Hermitian matrix, (black box), all eigenvalues. Double precision version is F02AWF.
- **F02AXE** Complex Hermitian matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02AXF.

### D4n4 : Complex non-Hermitian

- **CMLIB subprogram library (EISPACK sublibrary)**

- **CG** Computes the eigenvalues and, optionally, the eigenvectors of a complex general matrix.

  *CMLIB subprogram library (LICEPACK sublibrary)*

- **CGEEV** Computes the eigenvalues and, optionally, the eigenvectors of a general complex matrix.

  *IMSL subprogram library*

- **EIGCC** Eigenvalues and (optionally) eigenvectors of a complex general matrix.

  *NAG subprogram library*

- **F02AJE** Complex matrix, (black box), all eigenvalues. Double precision version is F02AJF.
- **F02AKE** Complex matrix, (black box), all eigenvalues and eigenvectors. Double precision version is F02AKF.
- **F02BDE** Complex matrix, (black box), selected eigenvalues and eigenvectors. Double precision version is F02BDF.

* Denotes easy-to-use modules
D4a5: Tridiagonal

CMLIB subprogram library (EISPACK sublibrary)

BISECT Compute eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.

IMTQL1 Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method.

IMTQL2 Computes eigenvalues and eigenvectors of symmetric tridiagonal matrix using implicit QL method.

IMTQLV Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. Eigenvectors may be computed later.

RATQR Computes largest or smallest eigenvalues of symmetric tridiagonal matrix using rational QR method with Newton correction.

RST Computes eigenvalues and, optionally, eigenvectors of a real symmetric tridiagonal matrix.

RT Compute eigenvalues and eigenvectors of a special real tridiagonal matrix.

TQL1 Compute eigenvalue of symmetric tridiagonal matrix by QL method.

TQL2 Compute eigenvalues and eigenvectors of symmetric tridiagonal matrix.

TQLRAT Computes eigenvalues of symmetric tridiagonal matrix using a rational variant of the QL method.

TRIDIB Computes eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.

TSTURM Computes eigenvalues of symmetric tridiagonal matrix in given interval and eigenvectors by Sturm sequencing.

IMSL subprogram library

EQRT1S Smallest or largest m eigenvalues of a symmetric tridiagonal matrix.

EQRT2S Eigenvalues and (optionally) eigenvectors of a symmetric tridiagonal matrix using the QL method.

EQRT3S The smallest (or largest) eigenvalues of a tridiagonal matrix in algebraic value whose sum exceeds a given value.

NAG subprogram library

F02AVE Real symmetric tridiagonal matrix, all eigenvalues, QL algorithm. Double precision version is F02AVF.

F02BEE Real symmetric tridiagonal matrix, selected eigenvalues and eigenvectors, bisection and inverse iteration. Double precision version is F02BEF.

F02BFE Real symmetric tridiagonal matrix, selected eigenvalues, bisection. Double precision version is F02BFF.

D4a6: Banded

CMLIB subprogram library (EISPACK sublibrary)

BQR Computes some of the eigenvalues of a real symmetric band matrix using the QR method with shifts of origin.

RSB Computes eigenvalues and, optionally, eigenvectors of real symmetric band matrix.

IMSL subprogram library

EIGBS Find some eigenvalues and (optionally) eigenvectors of a real symmetric band matrix.

D4a7: Sparse

D4b: Generalized eigenvalue problems (e.g., Ax = λBx)
D4b1: Real symmetric

CMLIB subprogram library (EISPACK sublibrary)

RSG Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: \( Ax = \lambda Bx. \)

RSGAB Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: \( ABx = \lambda x. \)

RSGBA Computes eigenvalues and, optionally, eigenvectors of real symmetric generalized eigenproblem: \( BAx = \lambda x. \)

IMSL subprogram library

EIGZS Eigenvalues and (optionally) eigenvectors of the system \( Ax = \lambda Bx \) where \( A \) and \( B \) are real symmetric matrices and \( B \) is positive definite.

MATHWARE subprogram library (NASHLIB sublibrary)

A15GSE Solution of the generalized symmetric eigenvalue problem by two applications of the Jacobi algorithm.

A25RQM Rayleigh quotient minimization by conjugate gradients.

NAG subprogram library

- F02ADE Generalised real symmetric eigenproblem \( Ax = kBx \) with positive-definite \( B \) (black box), all eigenvalues. Double precision version is F02ADF.
- F02AEE Generalised real symmetric eigenproblem \( Ax = kBx \) with positive-definite \( B \) (black box), all eigenvalues and eigenvectors. Double precision version is F02AEF.

D4b2: Real general

CMLIB subprogram library (EISPACK sublibrary)

RGG Computes eigenvalues and eigenvectors for real generalized eigenproblem: \( Ax = \lambda Bx. \)

IMSL subprogram library

EIGZF Eigenvalues and (optionally) eigenvectors of the system \( Ax = \lambda Bx \) where \( A \) and \( B \) are real matrices.

MATHWARE subprogram library (NASHLIB sublibrary)

A10GII Solves generalized eigenvalue problem by inverse iteration via Gauss elimination.

NAG subprogram library

- F02BJE Generalised eigenproblem \( Ax = kBx \), QZ algorithm (black box), real matrices, all eigenvalues and (optionally) eigenvectors. Double precision version is F02BJF.

D4b3: Complex Hermitian

D4b4: Complex general

IMSL subprogram library

EIGZC Eigenvalues and (optionally) eigenvectors of the system \( Ax = \lambda Bx \) where \( A \) and \( B \) are complex matrices.

NAG subprogram library
### D4b5: Banded

**NAG subprogram library**

**F02SDE** Generalised real eigenproblem $Ax=kBx$, where $A$ and $B$ are band matrices, eigenvector by inverse iteration. Double precision version is F02SDF.

### D4c: Associated operations

**MATHWARE subprogram library (NASHLIB sublibrary)**

**A12CVR** Residuals for a complex eigenvalue $e+if$ and eigenvector $x+iy$ of the matrix $A+iZ$.

### D4c1: Transform problem

#### D4c1a: Balance matrix

**CMLIB subprogram library (EISPACK sublibrary)**

**BALANC** Balances a general real matrix and isolates eigenvalues whenever possible.

**CBAL** Balances a complex general matrix and isolates eigenvalues whenever possible.

**IMSL subprogram library**

**EBALAC** Balance a complex general matrix and isolate eigenvalues whenever possible.

**EBALAF** Balance a real matrix.

**NAG subprogram library**

**F01ATE** Balances a real unsymmetric matrix. Double precision version is F01ATF.

**F01AVE** Balances a complex matrix. Double precision version is F01AVF.

### D4c1b: Reduce to compact form

#### D4c1b1: Tridiagonal

**CMLIB subprogram library (EISPACK sublibrary)**

**BANDR** Reduces real symmetric band matrix to symmetric tridiagonal matrix and, optionally, accumulates orthogonal similarity transformations.

**HTRID3** Reduces complex Hermitian (packed) matrix to real symmetric tridiagonal matrix by unitary similarity transformations.

**HTRIDI** Reduces complex Hermitian matrix to real symmetric tridiagonal matrix using unitary similarity transformations.

**TRED1** Reduce real symmetric matrix to symmetric tridiagonal matrix using orthogonal similarity transfor-
<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td>TRED2</td>
<td>Reduce real symmetric matrix to symmetric tridiagonal matrix using and accumulating orthogonal transformations.</td>
</tr>
<tr>
<td>TRED3</td>
<td>Reduce real symmetric matrix stored in packed form to symmetric tridiagonal matrix using orthogonal transformations.</td>
</tr>
<tr>
<td><strong>NAG subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td>F01AGE</td>
<td>Gives the Householder reduction of a real symmetric matrix to tridiagonal form for use in F02BEF, F02AVF and F02BFF. Double precision version is F01AGF.</td>
</tr>
<tr>
<td>F01AJE</td>
<td>Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02AME. Double precision version is F01AJF.</td>
</tr>
<tr>
<td>F01AYE</td>
<td>Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02BEE, F02AVE, and F02BFE. More economical in storage than similar F01AGE. Double precision version is F01AYF.</td>
</tr>
<tr>
<td>F01BCE</td>
<td>Gives the Householder reduction of a complex Hermitian matrix to tridiagonal form for use in F01AVE or F02AYE. Double precision version is F01BCF.</td>
</tr>
<tr>
<td>F01BWE</td>
<td>Reduces a symmetric band matrix to tridiagonal form. This routine normally used with F02AVE to find all eigenvalues of A. For selected values, use F02BME. Double precision version is F01BWF.</td>
</tr>
</tbody>
</table>

### D4c1b2: Hessenberg

<table>
<thead>
<tr>
<th>Subprogram Library (EISPACK sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMHES</td>
</tr>
<tr>
<td>CORTH</td>
</tr>
<tr>
<td>ELMHES</td>
</tr>
<tr>
<td>ORTHES</td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td>EHESSC</td>
</tr>
<tr>
<td>EHESSF</td>
</tr>
<tr>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td>F01AKE</td>
</tr>
<tr>
<td>F01AME</td>
</tr>
</tbody>
</table>

### D4c1b3: Other

<table>
<thead>
<tr>
<th>Subprogram Library (EISPACK sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZHES</td>
</tr>
</tbody>
</table>
form using orthogonal transformations. Usually followed by QZIT, QZVAL, QZVEC.

**QZIT**
The second step of the QZ algorithm for generalized eigenproblems. Accepts an upper Hessenberg and an upper triangular matrix and reduces the former to quasi-triangular form while preserving the form of the latter. Usually preceeded by QZHES and followed by QZVAL and QZVEC.

**D4c1e :** Standardize problem

**CMLIB subprogram library (EISPACK sublibrary)**
- **FIGI** Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix.
- **FIGI2** Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix.
- **REDUC** Reduces generalized symmetric eigenproblem $Ax = \lambda Bx$, to standard symmetric eigenproblem, using Cholesky factorization.
- **REDUC2** Reduces certain generalized symmetric eigenproblems to standard symmetric eigenproblem, using Cholesky factorization.

**NAG subprogram library**
- **F01AEE** Reduces the generalized eigenproblem $Ax = \lambda Bx$, where A is real symmetric and B is real symmetric positive definite, to the standard symmetric eigenproblem. Double precision version is F01AEF.
- **F01BDE** Reduces eigenproblems $ABx = \lambda x$, $x^TBA = \lambda x^T$, $BAy = \lambda y$, $y^T AB = \lambda y^T$ to standard symmetric eigenproblem $Qz = \lambda z$. Double precision version is F01BDF.
- **F01BVE** Transforms the generalized symmetric eigenproblem $Ax = \lambda Bx$ to equivalent standard eigenproblem $Cy = \lambda y$; A, B, C symmetric band matrices, B positive definite + decomposed $Cy = \lambda y$. Double precision version is F01BVF.

**D4c2 :** Compute eigenvalues of matrix in compact form

**D4c2a :** Tridiagonal

**CMLIB subprogram library (EISPACK sublibrary)**
- **BISECT** Compute eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
- **IMTQL1** Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method.
- **IMTQL2** Computes eigenvalues and eigenvectors of symmetric tridiagonal matrix using implicit QL method.
- **IMTQLV** Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. Eigenvectors may be computed later.
- **RATQR** Computes largest or smallest eigenvalues of symmetric tridiagonal matrix using rational QR method with Newton correction.
- **TQL1** Compute eigenvalue of symmetric tridiagonal matrix by QL method.
- **TQL2** Compute eigenvalues and eigenvectors of symmetric tridiagonal matrix.
- **TQLRAT** Computes eigenvalues of symmetric tridiagonal matrix using a rational variant of the QL method.
- **TRIDIB** Computes eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
- **TSTURM** Computes eigenvalues of symmetric tridiagonal matrix in given interval and eigenvectors by Sturm sequencing.

**NAG subprogram library**
- **F02AME** Real symmetric matrix, all eigenvalues and eigenvectors, after reduction to tridiagonal form by F01AJE,
QL algorithm. Double precision version is F02AMF.

**F02AYE** Complex Hermitian matrix, all eigenvalues and eigenvectors, after reduction to real tridiagonal form by F01BCE, QL algorithm. Double precision version is F02AYF.

**D4c2b :** Hessenberg

**CMLIB subprogram library (EISPACK sublibrary)**

**CINVIT** Computes eigenvectors of a complex upper Hessenberg matrix associated with specified eigenvalues using inverse iteration.

**COMLR** Computes eigenvalues of a complex upper Hessenberg matrix using the modified LR method.

**COMLR2** Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix using modified LR method.

**COMQR** Computes eigenvalues of complex upper Hessenberg matrix using the QR method.

**COMQR2** Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix.

**HQR** Computes eigenvalues of a real upper Hessenberg matrix using the QR method.

**HQR2** Computes eigenvalues and eigenvectors of real upper Hessenberg matrix using QR method.

**INVIT** Computes eigenvectors of upper Hessenberg (real) matrix associated with specified eigenvalues by inverse iteration.

**NAG subprogram library**

**F02ANE** Complex upper Hessenberg matrix, all eigenvalues, LR algorithm. Double precision version is F02ANF.

**F02APE** Real upper Hessenberg matrix, all eigenvalues, QR algorithm. Double precision version is F02APF.

**F02AQE** Real matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AKE and F01APE, QR algorithm. Double precision version is F02AQF.

**F02ARE** Complex matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AME, LR algorithm. Double precision version is F02ARF.

**D4c2c :** Other

**CMLIB subprogram library (EISPACK sublibrary)**

**QZVAL** The third step of the QZ algorithm for generalized eigenproblems. Accepts a pair of real matrices, one in quasi-triangular form and the other in upper triangular form and computes the eigenvalues of the associated eigenproblem. Usually preceded by QZHES, QZIT, and followed by QZVEC.

**D4c3 :** Form eigenvectors from eigenvalues

**CMLIB subprogram library (EISPACK sublibrary)**

**BANDV** Forms eigenvectors of real symmetric band matrix associated with a set of ordered approximate eigenvalues by inverse iteration.

**QZVEC** The optional fourth step of the QZ algorithm for generalized eigenproblems. Accepts a matrix in quasi-triangular form and another in upper triangular form and computes the eigenvectors of the triangular problem and transforms them back to the original coordinates. Usually preceded by QZHES, QZIT, QZVAL.

**TINVIT** Eigenvectors of symmetric tridiagonal matrix corresponding to some specified eigenvalues, using inverse iteration.
NAG subprogram library

F02BKE  Compute selected eigenvectors of a real upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. Double precision version is F02BKF.

F02BLE  Compute selected eigenvectors of a complex upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. Double precision version is F02BLF.

D4c4  Back transform eigenvectors

CMLIB subprogram library (EISPACK sublibrary)

BAKVEC  Forms eigenvectors of certain real non-symmetric tridiagonal matrix from symmetric tridiagonal matrix output from FIGI.

BALBAK  Forms eigenvectors of real general matrix from eigenvectors of matrix output from BALANC.

CBABK2  Forms eigenvectors of complex general matrix from eigenvectors of matrix output from CBAL.

COMBAK  Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from COMHES.

CORTB  Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from CORTH.

ELMBAK  Forms eigenvectors of real general matrix from eigenvectors of upper Hessenberg matrix output from ELMHES.

ELTRAN  Accumulates the stabilized elementary similarity transformations used in the reduction of a real general matrix to upper Hessenberg form by ELMHES.

HTRIB3  Computes eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from HTRID3.

HTRIBK  Forms eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from HTRIDI.

ORTBAK  Forms eigenvectors of general real matrix from eigenvectors of upper Hessenberg matrix output from ORTHES.

ORTRAN  Accumulates orthogonal similarity transformations in reduction of real general matrix by ORTHES.

REBAK  Forms eigenvectors of generalized symmetric eigensystem from eigenvectors of derived matrix output from REDUC or REDUC2.

REBAKB  Forms eigenvectors of generalized symmetric eigensystem from eigenvectors of derived matrix output from REDUC2.

TRBAK1  Forms the eigenvectors of real symmetric matrix from eigenvectors of symmetric tridiagonal matrix formed by TRED1.

TRBAK3  Forms eigenvectors of real symmetric matrix from the eigenvectors of symmetric tridiagonal matrix formed by TRED3.

NAG subprogram library

F01AFE  Derives eigenvectors of several generalized eigenproblems from the corresponding eigenvectors of the related standard symmetric eigenproblem. Double precision version is F01AFF.

F01AHE  Derives eigenvectors of a real symmetric matrix from the eigenvectors of the tridiagonal form produced by F01AGE. Double precision version is F01AHF.

F01ALE  Transforms eigenvectors of a Hessenberg matrix to those of a real unsymmetric matrix from which the Hessenberg matrix has previously been derived. Double precision version is F01ALF.

F01ANE  Transforms eigenvectors of a complex upper Hessenberg matrix to those of a complex unsymmetric matrix from which the Hessenberg matrix has previously been derived. Double precision version is F01ANF.

F01APE  Forms the matrix of accumulated transformations from information left by F01AKE. Double precision
version is F01APF.

F01AUE  Transforms eigenvectors of a balanced matrix to those of the original real unsymmetric matrix. Double precision version is F01AUF.

F01AWE  Transforms eigenvectors of a balanced matrix to those of the original complex matrix. Double precision version is F01AWF.

F01AZE  Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form produced by F01AYE. Double precision version is F01AZF.

F01BEE  Derives eigenvectors of the problems $y^T A B = \lambda y^T$ and $B A y = \lambda y$ from corresponding eigenvectors of derived standard symmetric eigenproblem. Double precision version is F01BEF.

D4c5 : Determine Jordan normal form

D5 : QR decomposition, Gram-Schmidt orthogonalisation

CMLib subprogram library (LINPACKC sublibrary)

CQRDC  Computes QR decomposition of general complex matrix.

CMLib subprogram library (LINPACKS sublibrary)

SQRDC  Computes QR decomposition of real general matrix. Double precision version is DQRDC.

CMLib subprogram library (SQRLSS sublibrary)

* SQRANK  For solving linear systems in least squares sense. Computes the QR decomposition of matrix using LINPACK subroutines. Double precision version is DQRANK.

MATHWARE subprogram library (NASHLIB sublibrary)

A3GR  Given's reduction of a real rectangular matrix.

NAG subprogram library

F01AXE  Reduces an $m \times n$ real matrix, $m \geq n$, to upper triangular form for use in F04AME and F04ANE. Uses Householder transformations with column pivoting. Double precision version is F01AXF.

F01QAE  QR-factorisation, real $m \times n$ matrix ($m \geq n$). Double precision version is F01QAF.

F01QBE  RQ-factorisation, real $m \times n$ matrix ($m \leq n$). Double precision version is F01QBF.

F02WDE  S.V.D. of a real $m \times n$ matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation ($m \geq n$). Double precision version is F02WDF.

F05AAE  Schmidt orthogonalisation of $n$ vectors of order $m$. Double precision version is F05AAF.

D6 : Singular value decomposition

CMLib subprogram library (EISPACK sublibrary)

MINFIT  Compute Singular Value Decomposition of rectangular real matrix and solve related Linear Least Squares problem.

SVD  Compute Singular Value Decomposition of arbitrary real rectangular matrix.

CMLib subprogram library (LINPACKC sublibrary)

CSVDC  Computes Singular Value Decomposition of general complex matrix.
SSVDC Computes Singular Value Decomposition of real general matrix. Double precision version is DSVDC.

**IMSL subprogram library**

**LSVDB** Singular value decomposition of a bidiagonal matrix.

**LSVDF** Singular value decomposition of a real matrix.

**MATHWARE subprogram library (NASHLIB sublibrary)**

**A1SVD** Singular Value Decomposition by means of orthogonalizing plane rotations.

**NAG subprogram library**

**F01LZE** Reduction by similarity transformations, real matrix to bidiagonal form. Double precision version is F01LZF.

**F02SZE** Singular value decomposition of a real bidiagonal matrix. Double precision version is F02SZF.

**F02WAE** Singular value decomposition of a real \( m \times n \) matrix, singular values and right singular vectors, \((m \geq n)\). Double precision version is F02WAF.

**F02WBE** Singular value decomposition of a real \( m \times n \) matrix, singular values and right singular vectors, \((m < n)\). Double precision version is F02WBF.

**F02WCE** Singular value decomposition of a real \( m \times n \) matrix, singular values and left and right singular vectors. Double precision version is F02WCF.

**F02WDE** S.V.D. of a real \( m \times n \) matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation \((m \geq n)\). Double precision version is F02WDF.

<table>
<thead>
<tr>
<th>D7</th>
<th>Update matrix decompositions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7a</td>
<td>LU</td>
</tr>
<tr>
<td>D7b</td>
<td>Cholesky</td>
</tr>
</tbody>
</table>

**CMLIB subprogram library (LINPACKC sublibrary)**

**CCHDD** Downdates Cholesky factorization of positive definite complex matrix.

**CCHEX** Updates Cholesky factorization of positive definite complex matrix.

**CCHUD** Updates Cholesky factorization of positive definite matrix.

**CMLIB subprogram library (LINPACKS sublibrary)**

**SCHDD** Downdates Cholesky factorization of real positive definite matrix. Double precision version is DCHDD.

**SCHEX** Updates Cholesky factorization of real positive definite matrix. Double precision version is DCHEX.

**SCHUD** Updates Cholesky factorization of real positive definite matrix. Double precision version is DCHUD.

| D7c | QR |
| D7d | Singular value |

| D8 | Other matrix equations (e.g., \( AX+XB=C \)) |
D9: Overdetermined and underdetermined systems of equations, singular systems, pseudo-inverses

Systems of linear algebraic equations possess either a single (unique) solution, an infinite number of solutions, or no solutions. Those systems possessing a unique solution are called non-singular. A singular system with no solutions is called inconsistent. Consistent singular systems always have an infinite number of solutions. In practical problems overdetermined systems (more rows than columns) are likely to be inconsistent, while underdetermined systems (fewer rows than columns) are likely to possess an infinite number of solutions.

Although there are no solutions to an inconsistent system, it is often useful to find the best approximation to the solution, in some sense. That is, given a system \(Ax = b\), we wish to find a vector \(x\) which minimizes \(||Ax - b||\) in some vector norm. The usual case is the \(L_2\) norm, in which case \(x\) is a solution in the least squares sense. For consistent systems with an infinite number of solutions one needs some extra condition in order to specify which solution is desired. Often there is some objective function which one wants to minimize over all solutions \(x\). One example of an objective function is \(|x|\), in which case one determines a "minimum norm solution".

Additional computational difficulty occurs when the problem is "rank deficient". In the case of overdetermined systems this means that the columns of \(A\) are linearly dependent, while in the case of underdetermined systems it means that the rows of \(A\) are linearly dependent. Rank deficient overdetermined systems occur, for example, in regression models with redundant (or nearly redundant) variables. Some software only applies to the full rank case, and hence one should be careful in selecting software when one suspects rank deficiency.

References


<table>
<thead>
<tr>
<th>D9 : Over or underdetermined or singular systems, pseudo-inverses (search also classes D5, D6, K1a, L8a)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library (EISPACK sublibrary)</strong></td>
</tr>
<tr>
<td>MINFIT</td>
</tr>
</tbody>
</table>

| **CMLIB subprogram library (LINPACKC sublibrary)**           |
| CQRSL              | Applies the output of CQRDC to compute coordinate transformations, projections, and least squares solutions (general complex matrix). |

| **CMLIB subprogram library (LINPACK8 sublibrary)**           |
| SQRSL              | Applies the output of SQRDC to compute coordinate transformations, projections, and least squares solutions (general real matrix). Double precision version is DQRSL. |

| **CMLIB subprogram library (SGLSS sublibrary)**              |
| LLSIA              | Computes least squares solution to \(AX=B\) with \(A\) an \(m\) by \(n\) matrix with \(m \geq n\) Flexible version of SGLSS. |
| SGLSS              | Solves linear least squares problems. Emphasis is put on detecting possible rank deficiency. Performs QR factorization using Householder transformations. Easy-to-use driver for LLSIA and ULSIA. |
| ULSIA              | Finds the minimal length solution of the underdetermined system of equations \(AX=B\) where \(A\) is an \(m\) by \(n\) matrix with \(m \leq n\) Flexible version of SGLSS. |

* Denotes easy-to-use modules
CMLIB subprogram library (SQRLLS sublibrary)

- SQRLLS For solving linear systems in least squares sense. Finds solution and residual after matrix factored by SQRANK. Double precision version is DQRLSS.

CMLIB subprogram library (SUDSSODS sublibrary)

SODS Solves an overdetermined system of linear equations. For full rank matrices the unique least squares solution is provided. The least squares solution of minimal length can be obtained in the rank deficient case.

SUDS Solves underdetermined systems of linear equations. For full rank matrices the minimum norm solution is returned, as well as an orthonormal basis for the null space of the matrix. If the system of equations is inconsistent only the least squares solution of minimal length is computed.

IMSL subprogram library

LGINF Generalized inverse of a real matrix.

LLBQF Solution of linear least squares problem high accuracy solution.

LLSQF Solution of a linear least squares problem.

OFIMA3 Least squares solution to the matrix equation AT = B.

MATHWARE subprogram library (NASHLIB sublibrary)

A2LSVD Least squares solution of rectangular linear system by Singular Value Decomposition.

NAG subprogram library

E02GAE L1-approximation by general linear function. Double precision version is E02GAF.

E02GBE L1-approximation by general linear function subject to linear inequality constraints. Double precision version is E02GBF.

E02GCE Calculates an L-infinity solution to an over-determined system of linear equations. Double precision version is E02GCF.

F01BLE Calculates the rank and pseudo-inverse of an mn real matrix A, m.GE.n, rank(A).LE.n, using Householder's factorisation. Double precision version is F01BLF.

- F04AME Least-squares, m real equations in n unknowns, rank=n, m>=n, accurate solution (black box). Double precision version is F04AMF.

- F04ANE Least-squares, m real equations in n unknowns, rank=n, m>=n, approximate solution (after factorisation by F01AXE). Double precision version is F04ANF.

- F04JAE Least-squares, m real equations in n unknowns, rank<n, m=n, minimal least-squares solution. Double precision version is F04JAF.

- F04JDE Least-squares, m real equations in n unknowns, rank<n, m<n, minimal least-squares solution. Double precision version is F04JDF.

- F04JGE Least-squares, m real equations in n unknowns, rank=n, m>n, least-squares solution if rank=n, otherwise minimal least-squares solution. Double precision version is F04JGF.

PORT subprogram library

CLST2 Finds the least squares solution of a complex linear algebraic system of equations AX=B. B may be a matrix. Uses real arithmetic. Double precision version is DCLST2.

LSTSQ Finds the least squares solution of a system of linear equations, AX=B. B may be a matrix. Double precision version is DLSTSQ.
E: Interpolation

By interpolation we mean the determination of a function which "passes through" given data values. The data may be one-dimensional (curve fitting), or higher dimensional (surface fitting). More generally, interpolation can also refer to requirements that derivative data match (this is sometimes referred to as osculatory interpolation). The purpose of interpolation is to find a functional form which can replace data which is known exactly. This may be design data, for example, in which case one needs a visually pleasing curve which represents the profile of a given object. Interpolation should not be used if the data have noise. In that case approximation techniques such as least squares (see classes K1 or 1.3) are more appropriate.

January 1984

E1: Univariate data (curve fitting)

In this case the problem is to find a function \( g(x) \) such that

\[
g(x_i) = y_i; \quad i = 1, \ldots, n
\]

where the points \((x_i, y_i)\) are given.

Polynomials have traditionally been used as interpolating functions. Unfortunately, serious computational problems can arise in determining the coefficients of interpolating polynomials. Furthermore polynomial interpolants often wiggle unphysically between data points. Alternative formulations can sometimes help but the more recent trend is to use a piecewise polynomial interpolant, most often a cubic spline.

A piecewise polynomial is a function which is composed of different polynomials joined together at points (called knots). A piecewise polynomial function made up of polynomials of degree \( n \) is called a spline if the function has \( n-1 \) continuous derivatives at the knots. Thus, a cubic spline is made up of cubic polynomials joined together so that the entire function has continuous first and second derivatives. Such functions are awkward to deal with analytically because a single neat formula is not available to represent them, but they are perfect for computations where the process of going from \( z \) to \( g(z) \) can be thought of as a "black box".

When users ask for a spline interpolation to data they normally must input two extra conditions in addition to the data points. That is, the interpolant has two extra degrees of freedom. Often these are used to specify the derivatives of the interpolant at the endpoints, but there are many other choices. Some programs don't have these options which give them the appearance of being easier to use. Rather, the extra degrees of freedom have been fixed internally and are not always consistent with the physical model. With just a little care, however, piecewise polynomials can make excellent interpolants and produce results that usually are more acceptable than polynomials.

The cubic Hermite are another heavily used set of interpolating functions. These functions are also piecewise cubic polynomials like cubic splines, but they are only joined together with one continuous derivative overall. They are easy to compute and are more flexible than splines. For example, if the data are monotonic it is possible to obtain a Hermite cubic which not only interpolates the data but is also monotonic between the data values. Splines often fail to behave this way.

Two common methods used to represent piecewise polynomial functions are the pp and B representations. In the pp representation one simply stores a list of coefficients of the polynomials describing the pp function on each interval. In the B representation the coefficients of an expansion in terms of a particular piecewise polynomial basis (the so-called B-splines) are stored; this is analogous to writing a polynomial as a linear sum of Legendre polynomials. The pp representation requires more storage that the B representation, but is simpler to evaluate. On the other hand, the B-representation is simpler to compute.

Subprograms for piecewise polynomial interpolation often come in pairs. One routine sets up the interpolant from the given data and is called just once. The second evaluates the interpolant or its derivatives at any point by using the results of the first subprogram and may be called many times.

January 1984

References


### E1 : Univariate data (curve fitting)

#### E1a : Polynomial splines (piecewise polynomials)

<table>
<thead>
<tr>
<th>CMLIB subprogram library (BSPLINE sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BINT4</strong> Computes B-spline which interpolates given X,Y data with various end conditions. The “B” representation is used. Double precision version is DBINT4.</td>
</tr>
<tr>
<td><strong>BINTK</strong> Produces B-spline coefficients of k-th order B-spline with given knots and with values at given points. Double precision version is DBINTK.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CMLIB subprogram library (PCHIP sublibrary)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PCHIC</strong> Determines a piecewise monotone, piecewise cubic Hermite interpolant to given data. User control is available over boundary conditions and/or treatment of points where monotonicity switches direction.</td>
</tr>
<tr>
<td><strong>PCHIM</strong> Determines a monotone piecewise cubic Hermite interpolant to given data. Default boundary values are provided which are compatable with monotonicity. The interpolant will have an extremum at each point where monotonicity switches direction.</td>
</tr>
<tr>
<td><strong>PCHSP</strong> Determines the cubic spline interpolant to given data. User has control over boundary conditions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IMSL subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ICSCCU</strong> Cubic spline interpolation (easy-to-use version).</td>
</tr>
<tr>
<td><strong>ICSICU</strong> Interpolatory approximation by cubic splines with arbitrary second derivative end conditions.</td>
</tr>
<tr>
<td><strong>ICSPLN</strong> Cubic spline interpolation with periodic end conditions.</td>
</tr>
<tr>
<td><strong>IQHSCU</strong> Visually pleasing interpolant of one dimensional data via piecewise cubic Hermite function.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAG subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E01BAE</strong> Interpolating functions cubic spline interpolant. Double precision version is E01BAF.</td>
</tr>
<tr>
<td><strong>E02BAE</strong> Least-squares curve fit by cubic splines (including interpolation). Double precision version is E02BAF.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PORT subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CSPFI</strong> Fits a cubic spline function to n input data pairs (x,y) with various endpoint conditions. Double precision version is DCSPFI.</td>
</tr>
<tr>
<td><strong>CSPIN</strong> Interpolates at requested points in given input data using a spline approximation—not a least squares fit. Double precision version is DCSPIN.</td>
</tr>
</tbody>
</table>

### E1b : Polynomials

<table>
<thead>
<tr>
<th>NAG subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E01AAE</strong> Interpolated values, one variable, data at unequally spaced points, Aitken’s technique. Double precision version is E01AAF.</td>
</tr>
<tr>
<td><strong>E01ABE</strong> Interpolated values, one variable, data at equally spaced points, Everett’s formula. Double precision version is E01ABF.</td>
</tr>
</tbody>
</table>
E01AEE  Interpolating functions, polynomial interpolant, data may include derivative values. Double precision version is E01AEF.

E02AFE  Least-squares curve fit by polynomials, special data points (including interpolation). Double precision version is E02AFF.

E1c : Other functions (e.g., rational, trigonometric)

NAG subprogram library

E01RAE  Produces, from a set of function values and corresponding abscissae, the coefficients of an interpolating rational function expressed in continued fraction form. Double precision version is E01RAF.

E2 : Multivariate data (surface fitting)

A typical multivariate interpolation problem is to find a function \( g(x, y) \) such that
\[
g(x_i, y_i) = z_i \quad i = 1, \ldots, n
\]
where the points \((x_i, y_i, z_i)\) are given. This is a two-dimensional interpolation problem. It corresponds to the determination of a surface passing through a given set of points in three-space. Higher-dimensional problems also occur, although it is difficult to visualize the interpolant in this case.

The difficulty of multidimensional interpolation depends upon the regularity of the data. If data is provided at all points on a rectangular grid, then the problem is relatively easy, since essentially one-dimensional methods may be applied. Most of the software for this case is based upon tensor products of one-dimensional piecewise polynomials. A number of libraries now provide software for two-dimensional gridded data interpolation, and several also handle the three-dimensional case. The more difficult case is when the data is not on a regular grid; this is the so-called scattered data interpolation problem. One of the difficulties here is that data may be dense in some areas and very sparse (or non-existent) in others; no numerical method can be expected to reproduce the underlying function in the absence of data. One should avoid using software for this problem without carefully examining the result. Much less software is available for this case, and what is available is restricted to two dimensions.

Numerical methods for scattered data interpolation can be classified as either global or local, although several hybrids have also been proposed. In a global method one writes the interpolant as a linear combination of functions each of which is non-zero at a single interpolation point and zero at all others. Such interpolants are easy to construct, expensive to evaluate, and sometimes lead to visually displeasing surfaces. Local methods often begin by triangularizing the data (determining a triangular grid with vertices corresponding to interpolation points). A separate polynomial function is then determined on each triangle which interpolates the local data. Although the resulting interpolant is easy to evaluate, determining it is more complex, especially if continuity restrictions are imposed.

References

B2INK Computes parameters of a piecewise-polynomial that interpolates a given set of two-dimensional gridded data. (Use B2VAL to evaluate function.) Double precision version is DB2INK.

B3INK Computes parameters of a piecewise-polynomial that interpolates a given set of three-dimensional gridded data. (Use B3VAL to evaluate function.) Double precision version is DB3INK.

IMSL subprogram library

IBCCCU Bicubic spline two-dimensional coefficient calculator.

IBCIEU Bicubic spline two-dimensional interpolator.

NAG subprogram library

E01ACE Interpolated values, two variables, data on rectangular grid, fitting bicubic spline. Double precision version is E01ACF.

E2b : Scattered

CMLIB subprogram library (LOTPS sublibrary)

LOTPS Passes smooth function thru points (XI(K),YI(K),FI(K),I=1..NPI) and returns an array of interpolated values on user specified grid.

IMSL subprogram library

IQHSCV Smooth surface fitting with irregularly distributed data points.

E3 : Service routines (e.g., grid generation, evaluation of fitted functions) (search also class N5)

CMLIB subprogram library (BSPLINE sublibrary)

BFQAD Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. Double precision version is DBFQAD.

BSPDR Constructs divided difference table from "B" representation of B-spline for a derivative calculation. Double precision version is DBSPDR.

BSPEV Calculates the value of a spline and its derivatives at X from its "B" representation. Double precision version is DBSPEV.

BSPPP Converts from "B" representation of B-spline to piecewise polynomial representation. Double precision version is DBSSPP.

BSPVD Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. Double precision version is DBSPVD.

BSPVN Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order. Double precision version is DBSPVN.

BSQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Double precision version is DBSQAD.

BVALU Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its "B" representation. Double precision version is DBVALU.

INTRV Computes the index into a knot or breakpoint sequence corresponding to a given point X. Double precision version is DINTRV.

PFQAD Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in piecewise polynomial representation. Double precision version is DPFQAD.

PPQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. Double precision version is DPPQAD.
PPVAL Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its piecewise polynomial representation. Double precision version is DPPVAL.

**CMLIB subprogram library (PCHIP sublibrary)**

CHFDV Evaluates a cubic polynomial and its first derivative at an array of points. The polynomial must be given in Hermite form.

CHFEV Evaluates a cubic polynomial given in Hermite form at an array of points.

PCHFD Evaluates a piecewise cubic Hermite function and its first derivative at an array of points.

PCHFE Evaluates a piecewise cubic Hermite function at an array of points.

PCHIA Evaluates the definite integral of a piecewise cubic Hermite function over an arbitrary interval.

PCHID Evaluates the definite integral of a piecewise cubic Hermite function over an interval whose endpoints are data points.

PCHMC Checks a cubic Hermite function for monotonicity.

**CMLIB subprogram library (TENSORBS sublibrary)**

B2VAL Evaluates the two-dimensional interpolating function computed by B2INK or one of its partial derivatives. Double precision version is DB2VAL.

B3VAL Evaluates the three-dimensional interpolating function computed by B3INK or one of its partial derivatives. Double precision version is DB3VAL.

**IMSL subprogram library**

DBCEVL Bicubic spline mixed partial derivative evaluator.

DCSEVU Cubic spline first and second derivative evaluator.

DCSQDU Cubic spline quadrature.

IBCEVL Evaluation of a bicubic spline.

ICSEVU Evaluation of a cubic spline.

**NAG subprogram library**

E01RBE Evaluates continued fractions of the form produced by NAG library routine E01RAE. Double precision version is E01RBF.

E02AEE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). Double precision version is E02AEF.

E02AHE Derivative of fitted polynomial in Chebyshev series form. Double precision version is E02AHF.

E02AKE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. Double precision version is E02AKF.

E02BBE Evaluation of fitted functions, cubic spline as E02BAE, function only. Double precision version is E02BBF.

E02BCE Evaluation of fitted functions, cubic spline as E02BAE, function and derivatives. Double precision version is E02BCF.

E02BDE Evaluation of fitted functions, cubic spline as E02BAE, definite integral. Double precision version is E02BDF.

E02CBE Evaluation of fitted functions, polynomial in two variables as E02CAE. Double precision version is E02CBF.

E02DBE Evaluation of fitted functions, bicubic spline as E02DAE. Double precision version is E02DBF.

E02ZAE Sort 2-d data into panels for fitting or evaluating bicubic splines. Double precision version is E02ZAF.

**PORT subprogram library**

BSPL1 Evaluates, at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives. Double precision version is DBSPL1.
<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSPLD</td>
<td>Evaluates at a given set of points in a specified mesh interval, basis splines and their derivatives.</td>
<td>Double</td>
</tr>
<tr>
<td>BSPLI</td>
<td>Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points.</td>
<td>Double</td>
</tr>
<tr>
<td>BSPLN</td>
<td>Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. Double precision version is DBSPLN.</td>
<td>Double</td>
</tr>
<tr>
<td>CSPFE</td>
<td>Evaluates a cubic spline function which has already been fit to n input data pairs (x,y) by CSPFI.</td>
<td>Double</td>
</tr>
<tr>
<td>DLUMD</td>
<td>Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points.</td>
<td></td>
</tr>
<tr>
<td>EEBSF</td>
<td>Estimates the error in a given B-spline fit to a function, f, by refining the mesh. Double precision version is DEEBSF.</td>
<td>Double</td>
</tr>
<tr>
<td>EEBSI</td>
<td>Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. Double precision version is DEEBSI.</td>
<td>Double</td>
</tr>
<tr>
<td>EESFF</td>
<td>Finds the maximum absolute error in a given B-spline fit to a function, f. Double precision version is DEESFF.</td>
<td>Double</td>
</tr>
<tr>
<td>EESFI</td>
<td>Finds the maximum absolute error in a given B-spline fit to a function, f, on a set of user selected intervals. Double precision version is DEESFI.</td>
<td>Double</td>
</tr>
<tr>
<td>IDMNPB</td>
<td>Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.</td>
<td></td>
</tr>
<tr>
<td>ILUMB</td>
<td>Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. Double precision version is IDLUMB.</td>
<td></td>
</tr>
<tr>
<td>ILUMD</td>
<td>Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. Double precision version is IDLUMD.</td>
<td></td>
</tr>
<tr>
<td>IMNPB</td>
<td>Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.</td>
<td></td>
</tr>
<tr>
<td>IPUMB</td>
<td>Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. Double precision version is IDPUMB.</td>
<td></td>
</tr>
<tr>
<td>IPUMD</td>
<td>Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. Double precision version is IDPUMD.</td>
<td></td>
</tr>
<tr>
<td>IUMB</td>
<td>Given interval endpoints, this generates a uniform mesh for B-spline use. Double precision version is IDUMB.</td>
<td></td>
</tr>
<tr>
<td>IUMD</td>
<td>Given interval endpoints, this generates a uniform mesh. Double precision version is IDUMD.</td>
<td></td>
</tr>
<tr>
<td>LUMB</td>
<td>Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. Double precision version is DLUMB.</td>
<td></td>
</tr>
<tr>
<td>LUMD</td>
<td>Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. Double precision version is IDLUMD.</td>
<td></td>
</tr>
<tr>
<td>MNPB</td>
<td>Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. Double precision version is DMNPB.</td>
<td></td>
</tr>
<tr>
<td>PUMB</td>
<td>Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. Double precision version is DPUMB.</td>
<td></td>
</tr>
<tr>
<td>PUMD</td>
<td>Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Double precision version is DPUMD.</td>
<td></td>
</tr>
<tr>
<td>SPLN1</td>
<td>Evaluates a function and derivatives described previously by an expansion in terms of B-splines. Double precision version is DSPLN1.</td>
<td></td>
</tr>
<tr>
<td>SPLN2</td>
<td>Evaluates a function described by a previously determined expansion in B-splines. More flexible than SPLN1. Double precision version is DSPLN2.</td>
<td></td>
</tr>
<tr>
<td>SPLND</td>
<td>Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. Double precision version is DSPLND.</td>
<td></td>
</tr>
<tr>
<td>SPLNE</td>
<td>Evaluates at a set of points, a function described by a previously determined expansion in terms of</td>
<td></td>
</tr>
</tbody>
</table>
B-splines. Double precision version is DSPLNE.

**SPLNI**
Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Double precision version is DSPLNI.

**UMB**
Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. Double precision version is DUMB.

**UMD**
Given interval endpoints, this generates a uniform mesh of distinct points. Double precision version is DUMD.
F: Solution of Nonlinear Equations

This chapter covers the solution of a system of \( m \) nonlinear equations for \( m \) unknowns. Most routines for this problem find a single solution of the given system based upon a starting guess provided by the user. Convergence cannot usually be guaranteed unless the starting guess is reasonably good. A special case is \( m = 1 \), a single nonlinear equation. Globally convergent algorithms are available for this case, provided one knows an interval in which the solution lies. Algorithms for the case of a single polynomial equations are even more specialized, and a number of programs are available which reliably compute all the zeros of a polynomial of moderate order.

January 1984

F : Solution of nonlinear equations

F1: Single equation

Solving one nonlinear equation,

\[ f(x) = 0, \]

for one root is easy if the user has a good estimate of the answer or can bound the interval where the root is to be found.

Usually, the user provides a guess at the root, a Fortran SUBROUTINE or FUNCTION to calculate \( f \) for any given \( z \), and a specification of the accuracy desired. Some programs stop when \( |f(x)| \) is small enough, and others stop when \( z \) is believed to be close to a root. Either tolerance might be absolute or relative, depending on the program. The most reliable programs require the user to provide an interval \((a, b)\) in which a root lies. By evaluating \( f \) at a sequence of points in the interval, ever smaller sub-intervals are produced which contain the root. Some such programs require \( f(a) \geq 0 \) and \( f(b) \leq 0 \) or the reverse, and then can guarantee finding a root. Programs which use derivatives of \( f \), or estimates of the derivatives, are less reliable, but may be somewhat faster than the programs which shrink the interval surrounding the root. Polynomial equations should ordinarily be solved by a special purpose program.

In Fortran subprograms that allow the user to specify a name for the FUNCTION subprogram which evaluates the function to be "zeroed" the user should be sure that this name appears

(1) in an EXTERNAL statement in the main program,
(2) in the subroutine CALL, and
(3) as the name of a FUNCTION.

January 1984

References


F1 : Single equation

F1a : Smooth

F1a1: Polynomial

Programs for solving polynomial equations,

\[ a_0 + a_1 z + a_2 z^2 + \ldots + a_n z^n = 0, \]

are quite reliable. They find all \( n \) roots of the polynomial. The difficulty is that high-degree polynomials are inherently unstable. Small errors in calculating the \( a \)'s can give large errors in the roots. Fortunately, many problems which require solving a polynomial equation, have formulations which are preferable. The casual user should consult an expert before attempting to solve such a problem with \( n \) larger than five.

January 1984

References

### Flal: Polynomial

<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPQR79</td>
<td>Computes all the zeros of a general real polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial.</td>
</tr>
<tr>
<td>RPZERO</td>
<td>Computes all the zeros of a polynomial with real coefficients. Error bounds are also computed. Uses Newton's Method for systems.</td>
</tr>
<tr>
<td>ZPOLR</td>
<td>Zeros of a polynomial with real coefficients (Laguerre).</td>
</tr>
<tr>
<td>ZQADR</td>
<td>Zeros of a quadratic with real coefficients.</td>
</tr>
<tr>
<td>ZRPOLY</td>
<td>Zeros of a polynomial with real coefficients (Jenkins-Traub).</td>
</tr>
</tbody>
</table>

### Flala: Real coefficients

<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C02AEE</td>
<td>All zeros of polynomial, Grant &amp; Hitchin’s method, real coefficients. Double precision version is C02AEF.</td>
</tr>
<tr>
<td>RPOLY</td>
<td>Finds zeros of a polynomial with real coefficients. Output zeros are in a pair of arrays, for real and imaginary part.</td>
</tr>
</tbody>
</table>

### Flalb: Complex coefficients

<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPQR79</td>
<td>Computes all the zeros of a general complex polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial.</td>
</tr>
<tr>
<td>CPZERO</td>
<td>Computes all the zeros of a polynomial with complex coefficients. Error bounds are also obtained. Uses Newton’s Method for systems.</td>
</tr>
<tr>
<td>ZCPOLY</td>
<td>Zeros of a polynomial with complex coefficients (Jenkins-Traub).</td>
</tr>
<tr>
<td>ZQADC</td>
<td>Zeros of a quadratic with complex coefficients.</td>
</tr>
<tr>
<td>C02ADE</td>
<td>All zeros of polynomial, Grant &amp; Hitchin’s method, complex coefficients. Double precision version is C02ADF.</td>
</tr>
<tr>
<td>CPOLY</td>
<td>Finds the zeros of a polynomial with complex coefficients. Uses two real arrays to represent complex</td>
</tr>
</tbody>
</table>
numbers, inconvenient. Double precision version is DCPOLY.

### F1a2: Nonpolynomial

**IMSL subprogram library**
- **ZANLYT**: Zeros of an analytic complex function using the Muller method with deflation.
- **ZFALSE**: Zero of a function given an interval containing the zero.
- **ZREAL1**: The real zeros of a real function - to be used when initial guesses are poor.
- **ZREAL2**: The real zeros of a real function - to be used when initial guesses are good.

**NAG subprogram library**
- **C05AJE**: Zero of continuous function of one variable, from a given starting value, continuation method. Double precision version is C05AJF.
- **C05AXE**: Zero of continuous function of one variable, from a given starting value, continuation method (reverse communication). Double precision version is C05AXF.

### F1b: General (no smoothness assumed)

**CMLIB subprogram library (ZEROIN sublibrary)**
- **ZEROIN**: Finds a zero of a user defined function on an interval given the endpoints A and B such that \( F(A) \cdot F(B) < 0 \).

**IMSL subprogram library**
- **ZBRENT**: Zero of a function which changes sign in a given interval (Brent algorithm).

**MATHWARE subprogram library (NASHLIB sublibrary)**
- **A18RF**: Root finding or minimization by bisection and false position.

**NAG subprogram library**
- **C05ADE**: Zero of continuous function of one variable, in a given interval, Bus & Dekker algorithm. Double precision version is C05ADF.
- **C05AGE**: Zero of continuous function of one variable, from a given starting value, search for interval, Bus & Dekker algorithm. Double precision version is C05AGF.
- **C05AVE**: Zero of continuous function of one variable, search for interval containing zero (reverse communication). Double precision version is C05AVF.
- **C05AZE**: Zero of continuous function of one variable, in a given interval, Bus & Dekker algorithm (reverse communication). Double precision version is C05AZF.

**PORT subprogram library**
- **ZERO**: Finds a single real root of a function within an interval specified by the user. Double precision version is DZERO.

### F2: System of equations

Solving a system of nonlinear equations,

\[
F(x) = 0,
\]
where \( \mathbf{x} \) is a vector and \( F \) is a vector-valued function, both of length \( n \), is not easy unless the user can provide a good estimate of the root.

Usually the user provides this estimate, a Fortran SUBROUTINE subprogram to calculate \( F \) for any given \( \mathbf{x} \), and a specification of the accuracy required. Some programs stop when \( \| F(\mathbf{x}) \| \) is small enough, and others stop when \( \mathbf{x} \) is believed to be close to a root. Some programs interpret \( \| F(\mathbf{x}) \| \) as the largest component of \( F \), others as the square root of the sum of squares of the components. Tolerances might be absolute or relative, depending on the program.

Programs to solve systems of nonlinear equations usually approximate numerically the “Jacobian matrix”, a matrix of partial derivatives of the components of \( F \) with respect to the components of \( \mathbf{x} \). Some programs allow the user to provide this matrix, and therefore can be more reliable. The Jacobian matrix is used in a matrix version of Newton’s method. A sequence of \( \mathbf{x} \)'s is constructed which may converge to a root. Programs vary as to when they quit if the process does not seem to be converging.

In Fortran subprograms that allow the user to specify a name for the FUNCTION subprogram which evaluates the function to be “zeroed” the user should be sure that this name appears

1. in an EXTERNAL statement in the main program,
2. in the subroutine CALL, and
3. as the name of a FUNCTION.

### References


<table>
<thead>
<tr>
<th>F2</th>
<th>System of equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2a</td>
<td>Smooth</td>
</tr>
</tbody>
</table>

| CMLIB subprogram library (SNL61E sublibrary) |
|-----------------|------------------------------------------|
| SNSQ | Finds a zero of a system of \( N \) nonlinear equations in \( N \) variables by a modification of the Powell hybrid method. Flexible usage. |
| • SNSQE | Finds a zero of a system of \( N \) nonlinear equations in \( N \) variables by a modification of Powell’s hybrid method. An easy to use driver for SNSQ. |
| • SOS | Finds a zero of a system of \( N \) nonlinear equations in \( N \) unknowns using Brown’s method. |

<table>
<thead>
<tr>
<th>IMSL subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZSCNT</td>
</tr>
<tr>
<td>ZSPOW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAG subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td>• C05NBE</td>
</tr>
<tr>
<td>C05NCE</td>
</tr>
</tbody>
</table>
• C05PBE  Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. Double precision version is C05PBF.

C05PCE  Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. (Comprehensive version of C05PBE.). Double precision version is C05PCF.

PORT subprogram library

• ZONE  Finds a solution of a system of non-linear equations. Double precision version is DZONE.
ZONEJ  Finds a solution of a system of non-linear equations. User must provide a SUBROUTINE to compute the Jacobian matrix. Double precision version is DZONEJ.

F2b :  General (no smoothness assumed)

F3 :  Service routines (e.g., check user-supplied derivatives)

IMSL subprogram library

ZSRCH  Generate points in an n dimensional space.

NAG subprogram library

C05ZAE  Checks the user-provided Jacobian prior to use in C05PBE or C05PCE. Double precision version is C05ZAF.
E04HCE  Check user's routine calculating first derivatives of function. Double precision version is E04HCF.
E04HDE  Check user's routine calculating second derivatives of function. Double precision version is E04HDF.
G: Optimization

The general problem addressed in this chapter is that of minimizing or maximizing a function $f(x)$ of one or more variables $x = (x_1, x_2, \ldots, x_n)$. In addition, the values of $x$ which are allowed might be required to satisfy an expression such as $c(x) = 0$, or $d(x) \geq 0$, or that the $x_i$ can assume only integer values. Such expressions are referred to as constraints, and a number of constraints are referred to collectively as the constraint set $C$ (e.g. $C = \{c_i(x) \geq 0, d_j(x) \geq 0, \text{ for } i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}$).

Alternative solution methods are chosen depending upon the functional form of $f$ and the constraint set $C$. For example, if $f$ and each constraint of $C$ is linear, then linear programming techniques can be applied. Problems with tens of thousands of variables and thousands of constraints can be solved using such techniques. If, in addition, the constraint set can be formulated as a network (arcs representing variables and nodes representing constraints) then network algorithms can be used to solve problems having millions of arcs and thousands of nodes.

Once the problem becomes nonlinear, that is, either the objective function or one or more of the constraints are nonlinear, then computational effort increases and choosing among algorithms becomes more difficult. Different algorithms are designed for solving various classes of nonlinear programming problems, such as unconstrained optimization problems, problems with inequality constraints, problems with equality constraints, and problems with both types of constraints. Within each of these categories, different algorithms make specific assumptions about the problem structure. For example, in unconstrained optimization, some procedures assume that the objective function is differentiable and use gradient values, whereas other algorithms do not make this assumption and rely primarily on function evaluations. (Algorithms requiring derivatives often provide the option of finite difference derivatives.) For problems with equality constraints, some algorithms can only handle linear constraints, while others can handle nonlinear constraints as well.

In general, one should use algorithms which exploit the structure of the problem. Thus, one should use a network code over a general linear programming code when feasible, especially for problems having all integer data. Problems having only linear constraints should be solved by an algorithm which fully exploits this structure. Problems for which one can provide derivatives should be solved by algorithms that capitalize on this. Similarly, it is much more efficient to use an algorithm that exploits least-squares structure when it exists. Other types of problems for which specific techniques have been developed include least absolute value, quadratic objective function with linear constraints, geometric programming problems, linear-fractional objective function with linear constraints, fixed point problems, shortest path, longest path, and minimum spanning tree problems, and a variety of integer programming problems where the linear constraint set has a specific structure.

Finally, inherent structure in the underlying application can sometimes be exploited to create a more tractable mathematical model. Nevertheless, there are a variety of special-purpose algorithms that can be constructed for giving good, fast, approximate solutions with known error bounds for some intractable problems.

References

G1a1 : Smooth function

G1a1a : User provides no derivatives

IMSL subprogram library

ZXLSF One-dimensional minimization of a smooth function using safeguarded quadratic interpolation.

NAG subprogram library

E04ABE Minimum, function of one variable using function values only. Double precision version is E04ABF.

G1a1b : User provides first derivatives

NAG subprogram library

E04BBE Minimum, function of one variable, using first derivative. Double precision version is E04BBF.

G1a1c : User provides first and second derivatives

G1a2 : General function (no smoothness assumed)

IMSL subprogram library

ZXGSN One-dimensional unimodal function minimization using the golden section search method.

ZXGSP One-dimensional unimodal function minimization using the golden section search method - data parameters specified.

MATHWARE subprogram library (NASHLIB sublibrary)

A16GS Grid or equal interval search along a line.

A17LS Success failure linear search with parabolic inverse interpolation.

A18RF Root finding or minimization by bisection and false position.

PORT subprogram library

FMIN Finds an approximate local minimum of a univariate user defined EXTERNAL function, f. Double precision version is DFMIN.

G1b : Multivariate

G1b1 : Smooth function

G1b1a : User provides no derivatives

CMLIB subprogram library (NL2SN sublibrary)

SMSNO Minimize a general unconstrained objective function using finite difference gradients and secant Hessian approximations. Double precision version is DSMSNO.
**IMSL subprogram library**

ZXCGR  A conjugate gradient algorithm for finding the minimum of a function of n variables.

ZZMIN  Minimum of a function of n variables using a quasi-Newton method.

**NAG subprogram library**

- **E04CGE**  Unconstrained minimum, function of several variables (easy-to-use), using function values only, quasi-Newton algorithm. Double precision version is E04CGF.

- **E04JBE**  Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. Double precision version is E04JBF.

---

**G1b1b : User provides first derivatives**

**CMLIB subprogram library (NL2SN sublibrary)**

SUMSL  Minimizes a general unconstrained objective function using analytic gradient and a Hessian approximation from a secant update. Double precision version is DSUMSL.

**MATHWARE subprogram library (NASHLIB sublibrary)**

A21VM  Variable metric minimisation method.

A22CGM  Function minimization by conjugate gradients.

**NAG subprogram library**

- **E04DBE**  Unconstrained minimum, function of several variables (comprehensive), using first derivatives, conjugate direction algorithm. Double precision version is E04DBF.

- **E04DEE**  Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, quasi-Newton algorithm. Double precision version is E04DEF.

- **E04DFE**  Unconstrained minimum, function of several variables (easy-to-use), using first derivatives, modified Newton algorithm. Double precision version is E04DFF.

---

**G1b1c : User provides first and second derivatives**

**CMLIB subprogram library (NL2SN sublibrary)**

HUMSL  Minimizes a general unconstrained objective function using (analytic) gradient and Hessian provided by the user. Double precision version is DHUMSL.

**NAG subprogram library**

- **E04EBE**  Unconstrained minimum, function of several variables (easy-to-use), using first and second derivatives, modified Newton algorithm. Double precision version is E04EBF.

---

**G1b2 : General function (no smoothness assumed)**

**MATHWARE subprogram library (NASHLIB sublibrary)**

A19NM  Nelder Mead simplex function minimisation. A short routine, but often quite successful.

**NAG subprogram library**

- **E04CCE**  Unconstrained minimum, function of several variables (comprehensive), using function values only, simplex algorithm. Double precision version is E04CCF.
GAMS: Modules by Class

January 1984

G2 : Constrained

G2a : Linear programming

G2a1 : Dense matrix of constraints

IMSL subprogram library
- ZX0LP Solve the linear programming problem (phase one or phase two) via the revised simplex algorithm.
- ZX3LP Solve the linear programming problem via the revised simplex algorithm easy to use version.
- ZX4LP Solve the linear programming problem via the revised simplex algorithm (alternate easy to use version).

NAG subprogram library
- H01ABE Linear programming, simplex algorithm, one iteration. Double precision version is H01ABF.
- H01ADE Linear programming, revised simplex method. Double precision version is H01ADF.
- H01BAE Linear programming, numerically stable form of simplex method. Double precision version is H01BAF.

G2a2 : Sparse matrix of constraints

CMLIB subprogram library (SPLP sublibrary)
- SPLP Solves linear optimization problems, that is, it minimizes the linear function $\text{COSTS}^T x$ subject to $Ax = w$, where the entries of the vectors $x$ and $w$ may have simple upper or lower bounds. Uses a sparse storage mode for the matrix $A$ and out-of-core scratch storage.

G2b : Transportation and assignments problem

NAG subprogram library
- H03ABE Solves the classical Transportation ("Hitchcock") problem. Double precision version is H03ABF.

G2c : Integer programming

G2c1 : Zero/one

G2c2 : Covering and packing problems

G2c3 : Knapsack problems

G2c4 : Matching problems

G2c5 : Routing, scheduling, location problems

• Denotes easy-to-use modules
G2c6: Pure integer programming

NAG subprogram library
H02BAE Integer linear programming, Gomory's method with Wilson's cuts. Double precision version is H02BAF.

G2c7: Mixed integer programming

G2d: Network (for network reliability search class M)

G2d1: Shortest path

G2d2: Minimum spanning tree

G2d3: Maximum flow

G2d3a: Generalized networks

G2d3b: Networks with side constraints

G2d4: Test problem generation

G2e: Quadratic programming

G2e1: Positive definite Hessian (i.e. convex problem)

NAG subprogram library
H02AAE Quadratic programming, Beale's method. Double precision version is H02AAF.

G2e2: Indefinite Hessian

G2f: Geometric programming

G2g: Dynamic programming

G2h: General nonlinear programming
<table>
<thead>
<tr>
<th>G2h1</th>
<th>Simple bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2h1a</td>
<td>Smooth function</td>
</tr>
<tr>
<td>G2h1a1</td>
<td>User provides no derivatives</td>
</tr>
<tr>
<td></td>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td>ZXMWD</td>
<td>Global minimum (with constraints) of a function of n variables.</td>
</tr>
<tr>
<td></td>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td>E04JAE</td>
<td>Minimum, function of several variables, simple bounds (easy-to-use), using function values only, quasi-Newton algorithm. Double precision version is E04JAF.</td>
</tr>
<tr>
<td>E04JBE</td>
<td>Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. Double precision version is E04JBF.</td>
</tr>
<tr>
<td>G2h1a2</td>
<td>User provides first derivatives</td>
</tr>
<tr>
<td></td>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td>E04KAE</td>
<td>Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, quasi-Newton algorithm. Double precision version is E04KAF.</td>
</tr>
<tr>
<td>E04KBE</td>
<td>Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, quasi-Newton algorithm. Double precision version is E04KBF.</td>
</tr>
<tr>
<td>E04KCE</td>
<td>Minimum, function of several variables, simple bounds (easy-to-use), using first derivatives, modified Newton algorithm. Double precision version is E04KCF.</td>
</tr>
<tr>
<td>E04KDE</td>
<td>Minimum, function of several variables, simple bounds (comprehensive), using first derivatives, modified Newton algorithm. Double precision version is E04KDF.</td>
</tr>
<tr>
<td>G2h1a3</td>
<td>User provides first and second derivatives</td>
</tr>
<tr>
<td></td>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td>E04LAE</td>
<td>Minimum, function of several variables, simple bounds (easy-to-use), using first and second derivatives, modified Newton algorithm. Double precision version is E04LAF.</td>
</tr>
<tr>
<td>E04LBE</td>
<td>Minimum, function of several variables, simple bounds (comprehensive), using first and second derivatives, modified Newton algorithm. Double precision version is E04LBF.</td>
</tr>
<tr>
<td>G2h1b</td>
<td>General function (no smoothness assumed)</td>
</tr>
<tr>
<td>G2h2</td>
<td>Linear equality or inequality constraints</td>
</tr>
<tr>
<td>G2h2a</td>
<td>Smooth function</td>
</tr>
</tbody>
</table>

* Denotes easy-to-use modules
G2h2a1 : User provides no derivatives

G2h2a2 : User provides first derivatives

G2h2a3 : User provides first and second derivatives

G2h2b : General function (no smoothness assumed)

G2h3 : Nonlinear constraints

G2h3a : Equality constraints only

G2h3a1 : Smooth function and constraints

G2h3a1a : User provides no derivatives

G2h3a1b : User provides first derivatives of function and constraints

G2h3a1c : User provides first and second derivatives of function and constraints

G2h3a2 : General function and constraints (no smoothness assumed)

G2h3b : Equality and inequality constraints

G2h3b1 : Smooth function and constraints

G2h3b1a : User provides no derivatives

NAG subprogram library

E04UAE Minimum, function of n variables, non-linear constraints, function and constraint values only, sequential augmented Lagrangian quasi-Newton method. Double precision version is E04UAF.

G2h3b1b : User provides first derivatives of function and constraints

NAG subprogram library

E04VAE Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential augmented Lagrangian quasi-Newton method. Double precision version is E04VAF.

E04VBE Minimum, function of several variables, general non-linear constraints, using first derivatives, sequential
augmented Lagrangian modified Newton method. Double precision version is E04VBF.

**G2h3b1e**: User provides first and second derivatives of function and constraints

**NAG subprogram library**

**E04WAE** Minimum, function of \( n \) variables, non-linear constraints, first and second derivatives, sequential augmented Lagrangian modified Newton method. Double precision version is E04WAF.

**G2h3b2**: General function and constraints (no smoothness assumed)

**G2i**: Global solution to nonconvex problems

**G3**: Optimal control

**G4**: Service routines

**G4a**: Problem input (e.g., matrix generation)

**G4b**: Problem scaling

**G4c**: Check user-supplied derivatives

**NAG subprogram library**

**E04HCE** Check user’s routine calculating first derivatives of function. Double precision version is E04HCF.

**E04HDE** Check user’s routine calculating second derivatives of function. Double precision version is E04HDF.

**F04YAE** Check user’s routine calculating Jacobian matrix of first derivatives. Double precision version is E04YAF.

**E04YBE** Check user’s routine calculating second derivative term in Hessian matrix of sum of squares. Double precision version is E04YBF.

**E04ZAE** Check user’s routines calculating first derivatives of function and constraints. Double precision version is E04ZAF.

**E04ZBE** Check user’s routines calculating second derivatives of function and constraints. Double precision version is E04ZBF.

**G4d**: Find feasible point

**IMSL subprogram library**

**ZSRCH** Generate points in an \( n \) dimensional space.

**NAG subprogram library**

**H01AFE** Find feasible point or vertex which satisfies linear constraints. Double precision version is H01AFF.
Check for redundancy

Other

NAG subprogram library

E04HBE Finite-difference intervals for estimating first derivatives. Double precision version is E04HBF.
H: Differentiation and Integration

This chapter contains programs for performing the fundamental operations of calculus, the computation of derivatives and the evaluation of definite integrals.

H1: Numerical differentiation

This class contains programs which approximate derivatives of a function (given analytically or by data values) at one or a few points. It does not include programs which first produce a fit if the fitted function can be used independently. Programs which do this are classified in chapters E and K.

Computing derivatives numerically often leads to serious rounding errors. If the function to be differentiated is available as a subprogram rather than as data values, adaptive procedures can be used to locate the best mesh to use in some interval. If the function can be extended to the complex plane then the most effective method is to represent the derivative by the Cauchy integral formula and use the trapezoidal quadrature rule to evaluate it.

References


H1: Numerical differentiation

**IMSL subprogram library**

**DRVTE** Calculate first, second, or third derivative of a user-supplied function.

**MATHWARE subprogram library (OLIVER sublibrary)**

**DIFFERENTI** Calculates 1st, 2nd, or 3rd derivative of given function F at fixed point x to specified accuracy, program picks evaluation points.

**NAG subprogram library**

**D04AAE** Numerical differentiation of a function of one real variable, derivatives up to order 14. Double precision version is D04AAF.

**PORT subprogram library**

**CSPDI** Finds a numerical approximation to the first derivative at requested points in given input data by using spline interpolation. Double precision version is DCSPDI.

H2: Quadrature (numerical evaluation of definite integrals)

This category includes routines which evaluate the definite integral of a function which is either given explicitly or by means of a table of data. Programs can either be automatic or non-automatic.

Most routines for the integration of an explicit function require that the integrand be in the form of a Fortran function subprogram. In this case the exact function name must appear

(1) in an EXTERNAL statement in the user's main program,
(2) in the CALL statement which invokes the quadrature subroutine, and
(3) as a Fortran FUNCTION itself.

The one exception to this is the very easy to use module Q1DA, which assumes that the function is called F.
Users must always be careful that their integrand function be defined wherever it might be evaluated. This depends on the specific routine but many programs always evaluate the integrand at the endpoints (in one dimension). If the problem is singular (e.g., $1/\sqrt{x}$ at 0.0) the user must arbitrarily set the integrand value at the singularity and prevent a divide by zero from occurring. Similarly, if the integrand has an apparent singularity (e.g., $\sin x/x$ at 0.0) the user must set the integrand value properly at this point (for this example $f(0) = 1$ is appropriate). A more correct solution is to use a program which does not perform endpoint evaluation. Sometimes the documentation omits this information. If internal singularities occur this same type of difficulty will arise. If the location of any singularities are known, it is highly recommended that the range of integration be split into pieces with the singularities only at the endpoints. This can be done automatically by some programs, for example, Q1DAX or QAGP by simply telling the integrator the locations of any problem points.

For "one shot" problems adaptive methods are very efficient in terms of human time. Adaption means that the integrand evaluation points are selected by the program and will vary from problem to problem. Automatic non-adaptive programs are also common. Some of these use a sequence of meshes, increasingly fine, and combine the results using an extrapolation algorithm. For the most part these techniques are not as efficient as the adaptive ones.

All automatic programs incur a penalty in terms of overhead, or surcharge. Non-automatic routines depend on a quadrature evaluation rule, a formula. Gauss quadrature rules are very efficient in terms of integrand evaluations but do not easily lend themselves to estimating errors. These formulas involve irrational points and weights which must be computed and tabulated in advance to be efficient. However they can save substantial computing time if the user is prepared to spend some time thinking about his problem and experimenting with different numbers of points. A number of routines which can generate and retrieve these formulas are listed in H2c. Programs which evaluate the formulas are listed in H2a1a2, H2a2a2, H2a3a2, H2b2a2 and H2b2b2.

Kronrod quadrature is a relatively new development. This involves the use of a pair of formulas, an $n$-point Gauss and a $2n + 1$ point Kronrod of which $n$ points are the original Gauss points. This allows for an accurate error estimate at modest cost. Many new programs use these formulas. See [3] for details.

There are several programs for the integration of tabular, or gridded, data. The underlying idea is that some function is "fit" to the data and the function is then integrated. Some programs accept the user's data directly, such as CSPQU. Others require that the user input the form of the function, usually either a spline or a polynomial, such as PCHIA or E02AJE. In the second case most users will have to precede the call to the integrator by a call to a "fitter", either an interpolator from Chapter E or an approximant from Chapter K.

Programs also exist for multiple integration. For the integration over hyperrectangles, one can achieve 5-6 figure accuracy for smooth functions and 2-3 figure accuracy for non-smooth functions up to dimension 15. For integration over a bounded irregular region, one embeds this region inside a hyperrectangle and defines the integrand to be zero outside the region. Is such cases a Monte Carlo type program, such as D01GDE, is recommended. One can also integrate over hyperspheres up to dimension 4 and over simplexes.

Two dimensional adaptive quadrature programs are just beginning to come into use. The overhead surcharge factor is even higher and there is much more scope for creativity by the problem originator. It has been common practice to use a pair of automatic one-dimensional programs to calculate iteratively a 2-D integral. In such cases it is suggested that the inner integral be evaluated more accurately to avoid instability. A COMMON block is also required to pass the second independent variable to the inner integrand which must be a function of only one argument.

January 1984

References

Module 77

GAMS: Modules by Class

**H2a1a:** Integrand available via user-defined procedure

**H2a1al:** Automatic (user need only specify required accuracy)

**CMLIB subprogram library (Q1DA sublibrary)**

- **Q1DA**
  Automatic evaluation of a user-defined function of one variable. Special features include randomization and singularity weakening.

- **Q1DAX**
  Flexible subroutine for the automatic evaluation of definite integrals of a user-defined function of one variable. Special features include randomization, singularity weakening, restarting, specification of an initial mesh (optional), and output of smallest and largest integrand values.

- **Q1DB**
  Automatic evaluation of a user-defined function of one variable. Integrand must be a Fortran Function but user may select name. Special features include randomization and singularity weakening. Intermediate in usage difficulty between Q1DA and Q1DAX.

**CMLIB subprogram library (QUADSP sublibrary)**

- **QAG**
  Automatic adaptive integrator, will handle many non-smooth integrands using Gauss Kronrod formulas. Double precision version is DQAG.

- **QAGE**
  Automatic adaptive integrator, can handle most non-smooth functions also provides more information than QAG. Double precision version is DQAGE.

- **QAGS**
  Automatic adaptive integrator, will handle most non-smooth integrands including those with endpoint singularities, uses extrapolation. Double precision version is DQAGS.

- **QAGSE**
  Automatic adaptive integrator, can handle integrands with endpoint singularities provides more information than QAGS. Double precision version is DQAGSE.

- **QNG**
  Automatic non-adaptive integrator for smooth functions, using Gauss Kronrod Patterson formulas. Double precision version is DQNG.

**IMSL subprogram library**

- **DCADRE**
  Numerical integration of a function using cautious adaptive Romberg extrapolation.

**NAG subprogram library**

- **D01AHE**
  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for well-behaved integrands. Double precision version is D01AHF.

- **D01AJE**
  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval allowing for badly-behaved integrands. Double precision version is D01AJF.

- **D01ARE**
  Computes definite integral over a finite range to a specified relative or absolute accuracy, using Patterson's method. Double precision version is D01ARF.

- **D01BDE**
  Quadrature for one-dimensional integrals, non-adaptive integration over a finite interval. Double precision version is D01BDF.

**PORT subprogram library**

- **ODEQ**
  Finds the integral of a set of functions over the same interval by using the differential equation solver ODES1. For smooth functions. Double precision version is DODEQ.

- **QUAD**
  Finds the integral of a general user defined EXTERNAL function by an adaptive technique to given absolute accuracy. Double precision version is DQUAD.

- **RQUAD**
  Finds the integral of a general user defined EXTERNAL function by an adaptive technique. Combined absolute and relative error control. Double precision version is DRQUAD.

* Denotes easy-to-use modules
**H2a1a2**: Nonautomatic

**CMLIB subprogram library (QUADSP sublibrary)**

- **QK15**: Evaluates integral of given function on an interval with a 15 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK15.
- **QK21**: Evaluates integral of given function on an interval with a 21 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK21.
- **QK31**: Evaluates integral of given function on an interval with a 31 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK31.
- **QK41**: Evaluates integral of given function on an interval with a 41 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK41.
- **QK51**: Evaluates integral of given function on an interval with a 51 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK51.
- **QK61**: Evaluates integral of given function on an interval with a 61 point Gauss Kronrod formula and returns error estimate. Double precision version is DQK61.

**NAG subprogram library**

- **D01BAE**: Quadrature for one-dimensional integrals, Gaussian rule-evaluation. Double precision version is D01BAF.

**H2a1b**: Integrand available only on grid

**H2a1b1**: Automatic (user need only specify required accuracy)

**H2a1b2**: Nonautomatic

**NAG subprogram library**

- **D01GAE**: Quadrature for one-dimensional integrals, integration of a function defined by data values only. Double precision version is D01GAF.

**PORT subprogram library**

- **CSPQU**: Finds the integral of a function defined by pairs \((x,y)\) of input points. The x's can be unequally spaced. Uses spline interpolation. Double precision version is DCSPQU.

**H2a2**: Finite interval (special integrand, e.g. weight functions, oscillating, singular, principal value, splines, etc.)

**H2a2a**: Integrand available via user-defined procedure

**H2a2a1**: Automatic (user need only specify required accuracy)

**CMLIB subprogram library (BSPLINE sublibrary)**

- **BFQAD**: Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. Double precision version is DBFQAD.
- **PFQAD**: Computes integral on \((X1,X2)\) of product of function and the ID-th derivative of B-spline which is in
piecewise polynomial representation. Double precision version is DPFQAD.

**CMLIB subprogram library (QUADSP sublibrary)**

- **QAGP**  Automatic adaptive integrator, allows user to specify location of singularities or difficulties of integrand, uses extrapolation. Double precision version is DQAGP.
- **QAGPE** Automatic adaptive integrator for function with user specified endpoint singularities, provides more information that QAGP. Double precision version is DQAGPE.
- **QAWC**  Cauchy principal value integrator, using adaptive Clenshaw Curtis method (real Hilbert transform). Double precision version is DQAWC.
- **QAWCE** Cauchy Principal value integrator, provides more information than QAWC (real Hilbert transform). Double precision version is DQAWCE.
- **QAWO**  Automatic adaptive integrator for integrands with oscillatory sin or cosine factor. Double precision version is DQAWO.
- **QAWOE** Automatic integrator for integrands with explicit oscillatory sin or cosine factor, provides more information than QAWO. Double precision version is DQAWOE.
- **QAWS**  Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities. Double precision version is DQAWS.
- **QAWSE** Automatic integrator for integrands with explicit algebraic and/or logarithmic endpoint singularities, more information than QAWS. Double precision version is DQAWSE.
- **QMOMO** Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. Double precision version is DQMOMO.

**NAG subprogram library**

- **D01AKE**  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, method suitable for oscillating functions. Double precision version is D01AKF.
- **D01ALE**  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, allowing for singularities at user-specified points. Double precision version is D01ALF.
- **D01ANE**  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function \( \cos(\omega x) \) or \( \sin(\omega x) \). Double precision version is D01ANF.
- **D01APE**  Adaptive integration of a function of one variable over a finite interval with weight function with algebraico-logarithmic endpoint singularities. Double precision version is D01APF.
- **D01AQE**  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function \( 1/(x-c) \) (Hilbert transform). Double precision version is D01AQF.

**PORT subprogram library**

- **BQUAD**  Adaptively integrates functions which have discontinuities in their derivatives. User can specify these points. Double precision version is DBQUAD.

---

**Nonautomatic**  

**CMLIB subprogram library (QUADSP sublibrary)**

- **QC25C**  Uses 25 point Clenshaw-Curtis formula to estimate integral of \( F \times W \) where \( W=1/(X-C) \). Double precision version is DQC25C.
- **QC25F**  Clenshaw-Curtis integration rule for function with \( \cos \) or \( \sin \) factor, also uses Gauss Kronrod formula. Double precision version is DQC25F.
- **QC25S**  Estimates integral of function with algebraico-logarithmic singularities with 25 point Clenshaw-Curtis formula and gives error estimate. Double precision version is DQC25S.
- **QK15W**  Evaluates integral of given function times arbitrary weight function on interval with 15 point Gauss Kronrod formula and gives error estimate. Double precision version is DQK15W.

---

* Denotes easy-to-use modules
NAG subprogram library

D01FBE Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

H2a2b: Integrand available only on grid

H2a2b1: Automatic (user need only specify required accuracy) or exact

CMLIB subprogram library (BSPLINE sublibrary)

BSQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Double precision version is DBSQAD.

PPQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. Double precision version is DPPQAD.

CMLIB subprogram library (PCHIP sublibrary)

PCHIA Evaluates the definite integral of a piecewise cubic Hermite function over an arbitrary interval.
PCHID Evaluates the definite integral of a piecewise cubic Hermite function over an interval whose endpoints are data points.

IMSL subprogram library

DCSQDU Cubic spline quadrature.

NAG subprogram library

E02AJE Integral of fitted polynomial in Chebyshev series form. Double precision version is E02AJF.
E02BDE Definite integral of a cubic previously computed by E02BAE in its "B" representation. Double precision version is E02BDF.

PORT subprogram library

BSPLI Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. Double precision version is DBSPLI.
SPLNI Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Double precision version is DSPLNI.

H2a2b2: Nonautomatic

H2a3: Semi-infinite interval (integrand can include $e^{-x}$ weight function)

H2a3a: Integrand available via user-defined procedure

H2a3a1: Automatic (user need only specify required accuracy)

CMLIB subprogram library (QUADSP sublibrary)

QAGI Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. Double precision version is DQAGI.
QAGIE  Automatic integrator for semi-infinite or infinite intervals and general integrands, provides more information than QAGI. Double precision version is DQAGIE.

QAWF  Automatic integrator for Fourier integrals on \((a, \infty)\) with factors \(\sin(\omega x)\) or \(\cos(\omega x)\), by integrating between zeros. Double precision version is DQAWF.

QAWFE  Automatic integrator for Fourier integrals, with \(\sin(\omega x)\) factor on \((a, \infty)\), provides more information than QAWF. Double precision version is DQAWFE.

**NAG subprogram library**

D01AME  Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. Double precision version is D01AMF.

<table>
<thead>
<tr>
<th>H2a3a2 : Nonautomatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLIB subprogram library (QUADSP sublibrary)</td>
</tr>
<tr>
<td>QK15I  Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error estimate. Double precision version is DQK15I.</td>
</tr>
</tbody>
</table>

**NAG subprogram library**

D01BAE  Quadrature for one-dimensional integrals, Gaussian rule-evaluation. Double precision version is D01BAF.

D01FBE  Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

<table>
<thead>
<tr>
<th>H2a4 : Infinite interval (integrand can include (e^{-x^2}) weight function)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>H2a4a : Integrand available via user-defined procedure</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>H2a4a1 : Automatic (user need only specify required accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLIB subprogram library (QUADSP sublibrary)</td>
</tr>
<tr>
<td>QAGI  Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. Double precision version is DQAGI.</td>
</tr>
</tbody>
</table>

QAGIE  Automatic integrator for semi-infinite or infinite intervals and general integrands, provides more information than QAGI. Double precision version is DQAGIE.

**NAG subprogram library**

D01AME  Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. Double precision version is D01AMF.

<table>
<thead>
<tr>
<th>H2a4a2 : Nonautomatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLIB subprogram library (QUADSP sublibrary)</td>
</tr>
<tr>
<td>QK15I  Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error estimate. Double precision version is DQK15I.</td>
</tr>
</tbody>
</table>

* Denotes easy-to-use modules
NAG subprogram library

D01BAE Quadrature for one-dimensional integrals, Gaussian rule-evaluation. Double precision version is D01BAF.

D01FBE Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

H2b: Multidimensional integrals

H2b1: One or more hyper-rectangular regions

H2b1a: Integrand available via user-defined procedure

H2b1a1: Automatic (user need only specify required accuracy)

IMSL subprogram library

DMLIN Numerical integration of a function of several variables over a hyper-rectangle (Gaussian method).

NAG subprogram library

D01FAE Quadrature for multi-dimensional integrals over a hyper-rectangle, Monte Carlo method. Double precision version is D01FAF.

D01FCE Quadrature for multi-dimensional integrals over a hyper-rectangle, adaptive method. Double precision version is D01FCF.

D01GBE Calculates an approximation to the integral of a function over a hyper-rectangular region, using a Monte-Carlo method. An approximate relative estimate is also returned. Suitable for low accuracy work. Double precision version is D01GBF.

H2b1a2: Nonautomatic

NAG subprogram library

D01FBE Quadrature for 1 to 20 dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation, extensive selection. Double precision version is D01FBF.

D01FDE Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. Double precision version is D01FDF.

D01GCE Calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov-Conroy number theoretic method. Returns a simple error estimate by repeating the computation with different (randomized) sets of points. Double precision version is D01GCF.

H2b1b: Integrand available only on grid

H2b1b1: Automatic (user need only specify required accuracy)
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Library</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2b1b2</td>
<td>Nonautomatic</td>
<td>IMSL subprogram library</td>
<td>Bicubic spline quadrature.</td>
</tr>
<tr>
<td>H2b2</td>
<td>Nonrectangular region, general region</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2b2a</td>
<td>Integrand available via user-defined procedure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2b2a1</td>
<td>Automatic (user need only specify required accuracy)</td>
<td>IMSL subprogram library</td>
<td>Numerical integration of a function of two variables.</td>
</tr>
<tr>
<td>D01DAE</td>
<td>Quadrature for two-dimensional integrals over a finite region. Double precision version is D01DAF.</td>
<td>NAG subprogram library</td>
<td></td>
</tr>
<tr>
<td>D01JAE</td>
<td>Attempts to evaluate an integral over an n-dimensional sphere (n=2,3,4), to a user specified absolute or relative accuracy, by means of a modified Sag-Szekeres method. Can handle singularities on the surface or at the center of the sphere. Returns an error estimate. Double precision version is D01JAF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2b2a2</td>
<td>Nonautomatic</td>
<td>NAG subprogram library</td>
<td></td>
</tr>
<tr>
<td>D01FDE</td>
<td>Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. Double precision version is D01FDF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D01PAE</td>
<td>Returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation. Double precision version is D01PAF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2b2b</td>
<td>Integrand available only on grid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2b2b1</td>
<td>Automatic (user need only specify required accuracy)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2b2b2</td>
<td>Nonautomatic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2c</td>
<td>Service routines (compute weight and nodes for quadrature formulas)</td>
<td>NAG subprogram library</td>
<td></td>
</tr>
<tr>
<td>D01BBE</td>
<td>Weights and abscissae for Gaussian quadrature rules, restricted choice of rule, using pre-computed weights and abscissae. Double precision version is D01BBF.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D01BCE</td>
<td>Weights and abscissae for Gaussian quadrature rules, more general choice of rule calculating the weights</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
and abscissae. Double precision version is D01BCF.

**PORT subprogram library**

**GAUSQ** Finds the abscissae and weights for Gauss quadrature on the interval (a,b) for a general weight function with known moments. Double precision version is DGAUSQ.

**GQ0IN** Finds the abscissae and weights for Gauss-Laguerre quadrature on the interval (0,+infinity). Double precision version is DGQ0IN.

**GQM11** Finds the abscissae and weights for Gauss-Legendre quadrature on the interval (-1,1). Double precision version is DGQM11.
I: Differential and Integral Equations

Differential and integral equations are the basis of most mathematical models of continuous processes, and hence the solution of these equations is a very important problem found in many applications. This chapter is divided into three parts which reflect the most general types of such equations—ordinary differential equations (ODEs), partial differential equations (PDEs), and integral equations.

I1: Ordinary differential equations

Physical laws are often posed in the form of systems of ordinary differential equations. Most programs for solving ODEs operate only on systems of first order equations of the form

\[ y' = f(t, y) \]

or

\[ Ay' = f(t, y) \]

where \( y \) and \( f \) are “vectors” and A is a square matrix. Most higher order systems can be reduced to one of these forms by a change of variable.

An initial value problem is a system of ODEs and a vector of numbers which specify the solution \( y \) at a specific (initial) time. The problem is to determine the solution at subsequent times. An initial value problem is stiff if the physical system contains time constants varying over several decades and the solution is desired in a range where the fastest components have died out. A boundary value problem is similar except that the “data” is not all given at a single point, e.g., some components may be specified at \( t_1 \) and others at \( t_2 \) etc.

Programs for solving initial or boundary value problems usually approximate the solution at a discrete set of points which are chosen dynamically by the integrator. These points may not correspond to the user’s output points, rather they are selected to get to the end as efficiently as possible. This requires an interpolation procedure (invisible to the user). The solution methods are often implicit and hence ultimately require the solution of systems of linear algebraic equations. Boundary value problems are more difficult and the programs are less reliable. Current techniques include shooting (solving a sequence of initial value problems), collocation (forcing an approximate solution to satisfy the ODEs at selected points), integral equations, invariant imbedding and finite differences.

Most programs require an error tolerance to be specified by the user. If the program terminates normally the implication is that the solution is given to within that tolerance. In practice this is often true. However, the input tolerance is usually used only to control local errors. There is no attempt to control accumulation of error over a long sequence of integration steps.

In Fortran subprograms that allow the user to specify the name of a subroutine to evaluate the differential equations be sure that this name appears

1. in an EXTERNAL statement in the main program,
2. in the CALL statement which invokes the ODE solver, and
3. as the name of a SUBROUTINE.

Another frequent programming error is to confuse the array \( Y \) which is input to the main ODE program with the array of the same name in the user-supplied subroutine to evaluate the derivatives.

References


II: Ordinary differential equations

IIa: Initial value problems

PLOD interactive system

*PLOD* An easy to use interactive system for the solution of initial value problems for ordinary differential equations. Requires a Tektronix or Hewlett-Packard graphics terminal. The user can change initial conditions, interval, parameters etc., and examine various plots on the terminal. Little programming needed.

IIal: General, nonstiff or mildly stiff

IIala: One-step methods (e.g., Runge-Kutta)

CMLIB subprogram library (DEPAC sublibrary)

DERKF Solves a system of first order ordinary differential equations with arbitrary initial conditions by a Runge-Kutta method.

IMSL subprogram library

DVERK Differential equation solver - Runge-Kutta-Verner fifth and sixth order method.

NAG subprogram library

*D02BAE* Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range. Double precision version is D02BAF.

*D02BBE* Initial value problems for system of ordinary differential equations, (simple driver) Runge-Kutta-Merson method, over a range with intermediate output. Double precision version is D02BBF.

*D02BDE* Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, over a range with global error estimate and stiffness check. Double precision version is D02BDF.

*D02BGE* Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a component of the solution attains a given value. Double precision version is D02BGF.

*D02BHE* Initial value problems for system of O.D.E.s, (simple driver) Runge-Kutta-Merson method, until a function of the solution is zero. Double precision version is D02BHF.

*D02PAE* Initial value problems for system of O.D.E.s, integrating over a range (facilities for error-control and interrupts) Runge-Kutta-Merson method. Double precision version is D02PAF.

*D02YAE* Initial value problems for system of ordinary differential equations, integration over one step by Runge-Kutta-Merson method. Double precision version is D02YAF.
IIa1b : Multistep methods (e.g., Adams' predictor-corrector)

CMLIB subprogram library (CDRIV sublibrary)
- CDRIV1 Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff formulas, Easy to use.
- CDRIV3 Numerical integration of complex initial value problems for ODEs, Gear and Adams formulas, Implicit eqs., Sparse Jacobians, root finding.

CMLIB subprogram library (DEABM sublibrary)
- DEABM Solves a system of first order ordinary differential equations with arbitrary initial conditions by a predictor-corrector method.

CMLIB subprogram library (SDASSL sublibrary)
- SDASSL Solves the system of differential/algebraic equations of the form \( g(t, y, y') = 0 \), with given initial values. Double precision version is DDASSL.

CMLIB subprogram library (SDRIV sublibrary)
- SDRIV1 Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear Stiff Formulas, Easy to Use. Double precision version is DDRIV1.

IMSL subprogram library
- DGEAR Differential equation solver - variable order Adams predictor corrector method or Gear's method.

NAG subprogram library
- D02CAE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range. Double precision version is D02CAF.
- D02CBE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range with intermediate output. Double precision version is D02CBF.
- D02CGE Initial value problems for system of O.D.E.s, (simple driver) variable order and step Adams method, until a component of the solution attains a given value. Double precision version is D02CGF.
- D02CHE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, until a function of the solution is zero. Double precision version is D02CHF.
- D02QAE Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Adams method. Double precision version is D02QAF.

SLDCL subprogram library
- SLGA1 Solves by difference methods an initial value problem for a system of ordinary differential equations. The stepsize and order of the method are chosen automatically to maintain prescribed local discretization error bounds.
- SLGA2 Solves by difference methods a mixed implicit/algebraic initial value problem for a system of ordinary differential equations. Stepsize and order are chosen automatically to maintain estimated local discretization error within prescribed bounds.
- SLGA3 Solves by difference methods an initial value problem for an explicit system of ordinary differential equations. Chooses stepsize and order of method to maintain estimate of local discretization error within prescribed bounds. Also provides values of first derivative at output points.

* Denotes easy-to-use modules
### SLGA4

Solves by difference methods a mixed algebraic/implicit initial value problem for a first order system of ordinary differential equations. Chooses stepsize and order to maintain estimate of local discretization error within prescribed bounds. Also provides derivative at output points.

### IIa1c: Extrapolation methods (e.g., Bulirsch-Stoer)

**IMSL subprogram library**

**DREBS**

Differential equation solver extrapolation method.

**PORT subprogram library**

- **ODES**
  
  Solves an initial value problem for a system of ordinary differential equations. Easy to use. Double precision version is DODES.

- **ODES1**
  
  Solves an initial value problem for a system of ordinary differential equations. Allows great flexibility and user control. Double precision version is DODES1.

### IIa2: Stiff and mixed algebraic-differential equations

**CMLIB subprogram library (CDRIV sublibrary)**

- **CDRIV1**
  
  Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff formulas, Easy to use.

- **CDRIV2**
  
  Numerical integration of complex initial value problems for ordinary differential equations, Gear stiff and Adams formulas, root finding.

- **CDRIV3**
  
  Numerical integration of complex initial value problems for ODEs, Gear and Adams formulas, Implicit eqs., Sparse Jacobians, root finding.

**CMLIB subprogram library (DEPAC sublibrary)**

- **DEBDF**
  
  Solves a system of first order stiff ordinary differential equations with arbitrary initial conditions by Gear's method.

**CMLIB subprogram library (SDRIV sublibrary)**

- **SDRIV1**
  
  Numerical Integration, Initial Value Problems, Ordinary Differential Eqs., Gear Stiff Formulas, Easy to Use. Double precision version is DDRIV1.

- **SDRIV2**
  

- **SDRIV3**
  

**IMSL subprogram library**

- **DGEAR**
  
  Differential equation solver - variable order Adams predictor corrector method or Gear's method.

**NAG subprogram library**

- **D02EAE**
  
  Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Gear method for stiff systems, over a range. Double precision version is D02EAF.

- **D02EBE**
  
  Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, over a range with intermediate output. Double precision version is D02EBF.

- **D02EGE**
  
  Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until the solution attains a given value. Double precision version is D02EGF.

- **D02EHE**
  
  Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for

* Denotes easy-to-use modules
stiff systems, until a function of the solution is zero. Double precision version is D02EHF.

**D02QBE** Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order-step Gear method for stiff systems. Double precision version is D02QBF.

### I1b : Multipoint boundary value problems

#### I1b1 : Linear

<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library (BVSUP sublibrary)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>BVSUP</strong></td>
<td>Solves a system of linear two-point boundary value problems using superposition, orthogonalization, and variable step integration.</td>
</tr>
<tr>
<td><strong>NAG subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>D02GBE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), general linear problem. Double precision version is D02GBF.</td>
</tr>
<tr>
<td><strong>D02JAE</strong></td>
<td>Solves a regular linear two point boundary value problem for a single n-th order ordinary differential equation by a Chebyshev series using collocation and least squares. Double precision version is D02JAF.</td>
</tr>
<tr>
<td><strong>D02JBE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, collocation and least-squares, system of 1st order linear equations. Double precision version is D02JBF.</td>
</tr>
<tr>
<td><strong>D02TGE</strong></td>
<td>Boundary-value problems for system of ordinary differential equations, collocation and least-squares, system of n-th order linear equations. Double precision version is D02TGF.</td>
</tr>
</tbody>
</table>

#### I1b2 : Nonlinear

<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DTPTB</strong></td>
<td>Solve a system of ordinary differential equations with boundary conditions at two points, using a multiple shooting method.</td>
</tr>
<tr>
<td><strong>DVCPR</strong></td>
<td>Solve a system of ordinary differential equations with boundary conditions at two points, using a variable order, variable step size finite difference method with deferred corrections.</td>
</tr>
<tr>
<td><strong>NAG subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>D02AGE</strong></td>
<td>Solves two point boundary value problem for a system of ODEs using initial value techniques. Double precision version is D02AGF.</td>
</tr>
<tr>
<td><strong>D02GAE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), simple non-linear problem. Double precision version is D02GAF.</td>
</tr>
<tr>
<td><strong>D02HAE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, shooting and matching technique, boundary values to be determined. Double precision version is D02HAF.</td>
</tr>
<tr>
<td><strong>D02HBE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined. Double precision version is D02HBF.</td>
</tr>
<tr>
<td><strong>D02RAE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction, general non-linear problem, continuation facility. Double precision version is D02RAF.</td>
</tr>
<tr>
<td><strong>D02SAE</strong></td>
<td>Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined, subject to extra algebraic equations. Double precision version is D02SAF.</td>
</tr>
<tr>
<td><strong>SLGRO</strong></td>
<td>Solves by difference methods on a non-equidistant grid a second-order implicit system of ordinary differential equations with implicit boundary conditions prescribed. An estimate of the discretization</td>
</tr>
</tbody>
</table>

* Denote easy-to-use modules
error is provided.

**SLGR1**  Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with prescribed implicit boundary conditions. An estimate of the discretization error is provided.

**SLGR2**  Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a given relative accuracy gridpoints and order of method are chosen automatically to minimize number of gridpoints.

**SLGR3**  Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a prescribed relative accuracy the order of the method is chosen to minimize the number of gridpoints.

**SLGR4**  Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a prescribed relative accuracy the gridpoints and order of method are chosen to minimize the number of gridpoints.

**SLGR5**  Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a given relative accuracy the gridpoints and order of method are chosen to minimize number of gridpts. Suited for boundary layer problems.

<table>
<thead>
<tr>
<th>slab3</th>
<th>Eigenvalue (e.g., Sturm-Liouville)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NAG subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>D02AGE</strong></td>
<td>Solves two point boundary value problem for a system of ODEs using initial value techniques. Double precision version is D02AGF.</td>
</tr>
<tr>
<td><strong>D02KAE</strong></td>
<td>Second-order Sturm-Liouville problems, regular system, finite range, eigenvalue only. Double precision version is D02KAF.</td>
</tr>
<tr>
<td><strong>D02KDE</strong></td>
<td>Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue only. Double precision version is D02KDF.</td>
</tr>
<tr>
<td><strong>D02KEE</strong></td>
<td>Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue and eigenfunction. Double precision version is D02KEF.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>slabc</th>
<th>Service routines (e.g., interpolation of solutions, error handling)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NAG subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>D02XAE</strong></td>
<td>Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, all components. Double precision version is D02XAF.</td>
</tr>
<tr>
<td><strong>D02XBE</strong></td>
<td>Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, one component. Double precision version is D02XBF.</td>
</tr>
<tr>
<td><strong>D02XGE</strong></td>
<td>Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, all components. Double precision version is D02XGF.</td>
</tr>
<tr>
<td><strong>D02XHE</strong></td>
<td>Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, one component. Double precision version is D02XHF.</td>
</tr>
</tbody>
</table>

| **PORT subprogram library** | |
| **ODESE** | Standard error subprogram for the routine ODES1. Double precision version is DODESE. |
| **ODESH** | Default HANDLE routine for ODES. Used to access the results at the end of each integration time step. Double precision version is DODESH. |
Partial differential equations are an important tool for researchers modelling continuous processes in all areas of science. Such complex models are rarely solvable using analytic techniques; instead, they provide some of the most challenging problems in all of scientific computing. Successful numerical methods must forge an effective synthesis of techniques from such diverse areas as approximation theory, numerical quadrature, and the numerical solution of linear and nonlinear algebraic equations.

The most common problems are "second order" and may be classified as either elliptic, parabolic or hyperbolic, although the most complex systems are combinations of these. A solution is required on some one, two or three dimensional domain, which may be bounded (with irregular boundaries adding geometrical headaches) or not.

Elliptic equations model steady-state phenomena, with the solution determined by conditions specified on the boundaries of the domain (Laplace's equation, \( u_{xx} + u_{yy} = 0 \), is the prototype). Parabolic problems add the element of time, with the solution at future times dependent upon the given solution at some initial time (the heat equation, \( u_t = u_{xx} \), is the prototype). Hyperbolic problems are also of the initial-boundary value problem type, but are characterized by the finite propagation speed of data (the wave equation, \( u_{tt} = u_{xx} \), is the prototype).

Because of the great diversity of problem characteristics, most programs for solving partial differential equations have been aimed at a specific problem in a specific applications area, with little possibility of easy extension to other problems. Thus, the state of general purpose software for partial differential equations is still in its infancy. The exceptions are in the areas of separable elliptic equations on rectangular domains and parabolic system solvers in one space dimension.

Most software begins with some finite dimensional approximation of the spatial part of the differential equation and boundary conditions. Two basic techniques are used—finite differences and finite elements. In finite differences, derivatives are directly approximated by difference quotients, leading to a system of algebraic equations whose solution yields values of the unknown quantities at a finite set of points. In finite elements, the solution is represented as a finite sum of known functions, each of which is zero on most of the domain. Variational techniques are then used to obtain a system of algebraic equations which determine the unknown coefficients of this finite sum. Finite elements have been tremendously successful in such application areas as structural engineering where their ability to easily conform to complex geometries is essential. In other areas such as fluid dynamics, finite differences remain popular due to their inherent simplicity. The computational complexity of the problem increases nonlinearly with dimension, and storage and computation times may be prohibitive for all but the coarsest of approximations in three dimensions.

The most widely distributed package of subroutines for elliptic problems is undoubtedly the FISHPAK collection (available in CMLIB). These routines provide efficient and reliable solutions to separable elliptic problems. These are single linear elliptic equations which are defined as the sum of two one-dimensional equations, one depending only upon \( x \) and one depending only upon \( y \). The domain must be rectangular (in Cartesian, polar, or surface spherical coordinates) with simple boundary conditions. In this case very fast techniques related to the numerical separation of variables may be applied. The SLDGL library contains several subprograms which discretize and solve general nonlinear systems of elliptic equations defined on rectangular domains. A single routine for problems on non-rectangular domains is also available. Each of these programs is based on finite differences. Laplace's equation is easily reformulated so that integral equation techniques may be applied. This is especially attractive for problems on complicated domains. The NAG library has a subroutine based on such techniques.

Most successful general purpose software packages for parabolic equations use the so-called method of lines. Here, the spatial approximation is used to generate a system of ordinary differential equations which is solved by existing general purpose software for that problem. Each of the IMSL, NAG, PDELIB, and SLDGL libraries have subprograms based on this technique.

References

Wiley & Sons, New York.

<table>
<thead>
<tr>
<th>I2</th>
<th>Partial differential equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2a</td>
<td>Initial boundary value problems</td>
</tr>
<tr>
<td>I2a1</td>
<td>Parabolic</td>
</tr>
<tr>
<td>I2a1a</td>
<td>One spatial dimension</td>
</tr>
</tbody>
</table>

**IMSL subprogram library**

| DPDES | Solve a system of partial differential equations of the form $U_T = FCN(X,T, U,UX, UXX)$ using the method of lines with cubic Hermite polynomials. |

**NAG subprogram library**

| D03PAE | P.D.E.s, parabolic, one space variable, method of lines, single equation. Double precision version is D03PAF. |
| D03PBE | P.D.E.s, parabolic, one space variable, method of lines, simple system. Double precision version is D03PBF. |
| D03PGE | P.D.E.s, parabolic, one space variable, method of lines, general system. Double precision version is D03PGF. |

**PDELIB subprogram library (PDELIB sublibrary)**

| MOL1D | Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available. |
| PDECOL | Solves general nonlinear systems of initial-boundary-value problems in one space dimension with general boundary conditions. Spatial derivatives may be of at most second order. Uses method of lines based on collocation of B-spline basis functions. |

**SLDGL subprogram library**

| SL1P1 | Solves a fully implicit difference scheme for a one-dimensional system of parabolic differential equations with general boundary conditions on a specified equidistant spatial grid using a specified order spatial method. Order and stepsize in time and error estimate are computed. |
| SL1P2 | Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. User specifies equidistant grid and order in space, stepsize and order in time and error estimate computed. |
| SL1P3 | Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions. For a given relative accuracy an equidistant spatial grid and optimal order are computed as well as stepsize and order in time. |
| SL1P4 | Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. Given a relative accuracy an equidistant spatial grid and optimal order are determined. |
| SL1P5 | Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions and non-equidistant spatial grid and spatial order. Stepsize and order in time are computed as well as an estimate of global error. |
Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. A non-equidistant spatial grid and spatial order are specified and the stepsize and order in time and error estimate are computed.

Solves a fully implicit difference scheme for a system of one-dimensional parabolic equations with specified initial and boundary conditions. For a specified relative accuracy a non-equidistant spatial grid and optimal order are computed as well as an estimate of global error.

Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. For a given relative accuracy a non-equidistant spatial grid and order and error estimate are computed.

PDELIB subprogram library (PDELIB sublibrary)

PDETWO Solves general nonlinear systems of initial-boundary-value problems in two spatial dimensions with quasi-linear boundary conditions. Uses the method of lines based upon finite differences on a user-specified rectangular mesh.

SLDCL subprogram library

SL2P1 Solves a fully implicit difference scheme for an implicit system of two- dimensional parabolic equations on a rectangle with specified initial and boundary conditions. User provides non-equidistant spatial grid and spatial orders and stepsize and order in time and error estimate are computed.

SL2P2 Solves a fully implicit difference scheme for an implicit system of two- dimensional parabolic equations. Initial conditions are computed from an elliptic boundary value problem. User specifies non-equidistant spatial grid and spatial orders and stepsize and order in time and error estimate are computed.

SL2P3 Solves a fully implicit difference scheme for an implicit system of parabolic equations with given initial and boundary conditions. For a given error tolerance an optimal combination of spatial gridpoint distribution and order are computed as well as an estimate of the global error.

SL2P4 Solves a fully implicit difference scheme for an implicit system of two- dimensional parabolic equations with the initial conditions determined as the solution of an elliptic boundary value problem. For a given tolerance optimal spatial gridpoint distribution and order are computed.

SL3P1 Solves a fully implicit difference scheme for a three-dimensional system of parabolic equations with specified initial and boundary conditions on a parallelepiped. The user provides a non-equidistant spatial grid and spatial orders and stepsize and order in time and an error estimate are computed.

PDELIB subprogram library (PDELIB sublibrary)

IMSL subprogram library

DPDES Solve a system of partial differential equations of the form $UT = FCN(X,T, U, UX, UXX)$ using the method of lines with cubic Hermite polynomials.

MOL1D Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available.

I2b : Elliptic boundary value problems
I2b1 : Linear

I2b1a : Second order

I2b1a1 : Poisson (Laplace) or Helmholtz equation

I2b1a1a : Rectangular domain (or topologically rectangular in the coordinate system)

CMLIB subprogram library (FSHPK sublibrary)

HSTCRT Solves the Helmholtz or Poisson equations in two dimensions in Cartesian coordinates on a staggered grid.
HSTCSP Solves a modified Helmholtz equation in spherical coordinates with axisymmetry using a staggered grid.
HSTCYL Solves a modified Helmholtz equation in cylindrical coordinates on a staggered grid.
HSTPLR Solves the Helmholtz or Poisson equation in polar coordinates on a staggered grid.
HSTSSP Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere using a staggered grid.
HW3CRT Solves the Helmholtz or Poisson equation in three dimensions using Cartesian coordinates.
HWSCRT Solves the Helmholtz or Poisson equation in two dimensions in Cartesian coordinates.
HWSCSP Solves a modified Helmholtz equation in spherical coordinates with axisymmetry.
HWSCYL Solves a modified Helmholtz equation in cylindrical coordinates.
HWSPLR Solves the Helmholtz or Poisson equation in polar coordinates.
HWSSSP Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere.

I2b1a1b : Nonrectangular domain

NAG subprogram library

D03EAE Partial differential equations, elliptic, Laplace’s equation in 2-d for an arbitrary domain. Double precision version is D03EAF.

I2b1a2 : Other separable problems

CMLIB subprogram library (FSHPK sublibrary)

SEPELI Solves separable elliptic boundary value problems on a rectangle.
SEPX4 Solves separable elliptic boundary value problems on a rectangle with constant coefficients in one direction.

I2b1a3 : Nonseparable problems

I2b1c : Higher order equations (e.g., biharmonic)
I2b2 : Nonlinear

SLDGL subroutine library

SL2E2 Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are automatically determined.

SL2E4 Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed.

SL2E5 Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the non-equidistant grid and order of the method are determined automatically. Discretization error estimated.

SL2E6 Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem with general boundary conditions on a rectangle. For a prescribed equidistant grid and relative accuracy the routine automatically determines the order of the method. Error estimate provided.

SL2E7 Solves a fully implicit difference scheme for two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided. Line iteration used.

SL2E8 Solves a fully implicit difference scheme for a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are determined. Approximation solved by line iteration.

SL2EB1 Solves a fully implicit difference scheme for a system of two-dimensional elliptic equations on a general region with Dirichlet or Neumann boundary conditions specified. An non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided.

SL3E1 Solves a fully implicit difference scheme for a system of three-dimensional elliptic equations on a parallelepiped with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed.

I2b3 : Eigenvalue

I2b4 : Service routines

I2b4a : Domain triangulation (search also class P2a2c1)

NAG subroutine library

D03MAE Triangulation of a plane region. Double precision version is D03MAF.

I2b4b : Solution of discretized elliptic equations

CMLIB subroutine library (FSHPK sublibrary)

BLKTRI Solves block tridiagonal systems of linear algebraic equations arising from the discretization of separable elliptic partial differential equations.

CBLKTR Solves certain complex block tridiag. systems of lin. eqns. arising from the discretization of sparable elliptic partial differential equations.

CMGNBN Solves certain complex block tridiag. systems of lin. eqns. arising from Helmholtz or Poisson eqn in 2 dim. Cartesian coordinates.
GENBUN  Solves certain block tridiagonal systems of lin. equations arising from Helmholtz or Poisson equation in two Cartesian coordinates.

POIS3D  Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations in 3D.

POISTG  Solves block tridiagonal linear systems of algebraic equations arising from the discretization of separable elliptic partial differential equations.

**MATHWARE subprogram library (ITPACK sublibrary)**


JSI  Iterative solution of large sparse systems of linear equations. Jacobi method, Chebyshev acceleration, adaptive parameter selection.

RSCG  Iterative solution of large sparse systems of linear equations. Reduced system method, conjugate gradient acceleration, adaptive.

RSSI  Iterative solution of large sparse systems of linear equations. Reduced system method, Chebyshev acceleration, adaptive.

SOR  Iterative solution of large sparse systems of linear equations. SOR method, adaptive parameter selection.

SSORCG  Iterative solution of large sparse systems of linear equations. SSOR method, conjugate gradient acceleration, adaptive parameter selection.

SSORSI  Iterative solution of large sparse systems of linear equations. SSOR method, Chebyshev acceleration, adaptive parameter selection.

**NAG subprogram library**

D03EBE  P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, iterate to convergence. Double precision version is D03EBF.

D03ECE  P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, iterate to convergence. Double precision version is D03ECF.

D03UAE  P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, one iteration. Double precision version is D03UAF.

D03UBE  P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, one iteration. Double precision version is D03UBF.

**13: Integral equations**

Integral equations may be "first" or "second" kind depending on whether the unknown function appears only inside or both inside and outside the integral. The latter are much easier to solve, both computationally and mathematically. First kind equations are often very sensitive numerically and special care must be exercised or the results will be meaningless. Easier problems correspond to those with more singular kernels; very smooth kernels are particularly difficult to handle.

January 1984

**References**


## I3 : Integral equations

### NAG subprogram library

<table>
<thead>
<tr>
<th>Subprogram</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D05AAE</td>
<td>Linear non-singular Fredholm integral equation, 2nd kind, split kernel. Double precision version is D05AAF.</td>
</tr>
<tr>
<td>D05ABE</td>
<td>Linear non-singular Fredholm integral equation, 2nd kind, smooth kernel. Double precision version is D05ABF.</td>
</tr>
</tbody>
</table>
J: Integral Transforms

Programs in this chapter compute integral transforms, the most common of which is the Fourier transform. FFT programs allow transformation to/from spectral space in an amount of computer time proportional to \( n \ln(n) \) where \( n \) is the number of data points. The programs are most efficient when \( n \) is highly composite, a power of two being the most propitious choice. It may take 200-300% more computing for an \( n = 127 \) transform than an \( n = 128 \), although most programs will work correctly for any \( n \). On the other hand, adding zero value data points can introduce spurious effects in the spectrum. The program selected must not only satisfy the physical model requirements (pure cosine transform, complex transform, etc.) but should also be appropriate for the intended computer. FFT programs can be made extra efficient by taking advantage of special machine hardware (e.g. vectorization capability) or by writing segments in assembly language. Even in Fortran, implementations on the same machine can easily differ 30% in running time. Note that while there are a few multidimensional FFT programs, most one dimensional modules are easily used in a multidimensional setting by writing a simple driver.

January 1984

References


| J : | Integral transforms |
| J1 : | Fast Fourier transforms (search class L10 for time series analysis) |
| J1a : | One-dimensional |
| J1a1 : | Real |

CMLIB subprogram library (FFTPKG sublibrary)
- FFTTB Computes real periodic sequence from real Fourier coefficients. Performs Fourier synthesis.
- FFTTF Computes Fourier coefficients of real periodic sequence (fast). Performs Fourier analysis.

IMSL subprogram library
- FFTRC Compute the fast Fourier transform of a real valued sequence.

NAG subprogram library
- C06EAE Discrete Fourier transform, FFT algorithm, no extra workspace, real data values. Double precision version is C06EAF.
- C06FAE Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, real data values. Double precision version is C06FAF.

PORT subprogram library
- FFTR Mixed radix fast Fourier transform to find the transform of 2N real data points. Double precision version is DFFTR.
- FFTRI Finds the inverse Fourier transform using Fourier coefficients assumed to arise from real data in the time domain. Double precision version is DFFTRI.
- RLTR An auxiliary routine for use together with FFT to transform 2N real data points. Uses less storage than FFTR. Double precision version is DRLTR.

* Denotes easy-to-use modules
J1a2: Complex

CMLIB subprogram library (FFTPKG sublibrary)

CFFTB  Backward complex discrete (fast) Fourier transform. Performs Fourier synthesis.
CFFTF  Forward complex discrete (fast) Fourier transform. Performs Fourier analysis.

IMSL subprogram library

FFT2C  Computes the fast Fourier transform of a complex valued sequence of length equal to a power two.
FFTCC  Compute the fast Fourier transform of a complex valued sequence.

NAG subprogram library

C06EBE  Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (Hermitian sequence). Double precision version is C06EBF.
C06ECE  Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (general sequence). Double precision version is C06ECF.
C06FBE  Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (Hermitian sequence). Double precision version is C06FBF.
C06FCE  Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (general sequence). Double precision version is C06FCF.

PORT subprogram library

FFT  Compute FFT of complex data sequence (forward or inverse) any number of points. Useful for multivariate transforms. Uses only real arithmetic. Double precision version is DFFT.
*FFTCC  Mixed radix fast Fourier transform of complex data. Two arrays used for complex data. Double precision version is DFFTC.
*FFTCCI  Finds the inverse fast Fourier transform, given the Fourier coefficients in the frequency domain. Double precision version is DFFTCCI.

J1a3: Trigonometric (sine, cosine)

CMLIB subprogram library (FFTPKG sublibrary)

COSQB  Fast Fourier transform of quarter wave data. Computes a sequence from cosine series representation. Fourier synthesis.
COSQF  Computes fast Fourier transform of quarter wave data. Fourier analysis. Computes coefficients in cosine series with odd wave numbers.
COST  Computes discrete (fast) cosine transform of even sequence X(I).
SINQB  Computes (fast) Fourier transform of quarter wave data. Backward (fast) sine transform. Performs Fourier synthesis.
SINQF  Computes (fast) Fourier transform of quarter wave data. Forward (fast) sine transform. Performs Fourier analysis.
SINT  Computes (fast) Fourier sine transform of an odd sequence X(I).

IMSL subprogram library

FFTSC  Compute the sine and cosine transforms of a real valued sequence.

• Denotes easy-to-use modules
### J1b: Multidimensional

**IMSL subprogram library**

**FFT3D**  Compute the fast Fourier transform of a complex valued 1, 2 or 3 dimensional array.

**NAG subprogram library**

**C06ADE**  Discrete Fourier transform, FFT algorithm, complex data values within a multi-variable transform. Double precision version is C06ADF.

**PORT subprogram library**

**FFT**  Compute FFT of complex data sequence (forward or inverse) any number of points. Useful for multivariate transforms. Uses only real arithmetic. Double precision version is DFFT.

### J2: Convolutions

**IMSL subprogram library**

**VCONVO**  Vector convolution.

**NAG subprogram library**

**C06ACE**  Circular convolution of two real vectors of period $2^m$. Double precision version is C06ACF.

### J3: Laplace transforms

**IMSL subprogram library**

**FLINV**  Inverse Laplace transform of a user supplied complex function.

### J4: Hilbert transforms

**CMLIB subprogram library (QUADSP sublibrary)**

- **QAWC**  Cauchy principal value integrator, using adaptive Clenshaw-Curtis method (real Hilbert transform). Double precision version is DQAWC.

- **QAWCE**  Cauchy Principal value integrator, provides more information than QAWC (real Hilbert transform). Double precision version is DQAWCE.

- **QC25C**  Uses 25 point Clenshaw-Curtis formula to estimate integral of $F$ times $W$ where $W=1/(X-C)$. Double precision version is DQC25C.

**NAG subprogram library**

**D01AQE**  Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $1/(x-c)$ (Hilbert transform). Double precision version is D01AQF.

---

* Denotes easy-to-use modules
K: Approximation

Any computer calculation which tries to model a continuous process involves some sort of approximation. In this chapter we consider only those programs which have curve or surface fitting as their ultimate goal. Computer-aided design is an important application of this software. Here one would like a simple mathematical function which represents the shape of an object under study. Such a function might be used, for example, by a numerically controlled milling machine. Another example is in computer graphics, where one wants a visually pleasing curve which approximates given data. Function approximation is a third example. Here one has a complicated mathematical function which must be replaced by a simpler one providing an approximation with a guaranteed maximum error.

The nature of an approximating curve or surface is determined by two fundamental choices: the choices of "norm" and "form". The "norm" is the means for measuring the distance of an approximating function \( g(x) \) from the data \((x_i, y_i), i = 1, \ldots, n\). In most cases we want to determine the free parameters of the approximating function so that this distance is minimized. Three important choices are:

(a) Least squares: minimize \( \sum_{i=1}^{n} (y_i - g(x_i))^2 \)
(b) Least absolute value: minimize \( \sum_{i=1}^{n} |y_i - g(x_i)| \)
(c) Minimax: minimize \( \max_{1 \leq i \leq n} |y_i - g(x_i)| \)

Interpolation is a special case of approximation. Here there are at least as many free parameters in \( g(x) \) as there are data points, and we require that \( y_i - g(x_i) = 0 \) for \( i = 1, \ldots, n \), that is, \( g(x) \) passes through the data points. Interpolation is useful when fitting a curve or surface to data which is known exactly, while least squares is more appropriate for data with inherent error. Software for interpolation is classified in chapter E.

The classical "form" for approximating functions is the polynomial. Unfortunately, polynomials are inadequate for many applications: low degree polynomials lack flexibility, while high degree polynomials can fluctuate wildly, and hence lack physical interpretation. An important breakthrough of the 1960's and 70's was piecewise polynomials (splines). These functions consist of low degree polynomial pieces joined smoothly at a set of breakpoints or knots. Piecewise polynomials are both flexible and stable, and have been used with enormous success by engineers and scientists in areas such as the aerospace and automobile industries for years. Easy to use software for computations with piecewise polynomials is now abundant.

Both polynomials and piecewise polynomials are examples of linear approximating functions. That is, they can be written in the form

\[
g(x) = a_1 f_1(x) + a_2 f_2(x) + \ldots + a_n f_n(x)
\]

where the "basis" functions \( f_1, f_2, \ldots, f_n \) are fixed and the coefficients \( a_1, a_2, \ldots, a_n \) are determined from the data. (Spline functions can be written in this form using the so-called B-representation.) Nonlinear approximating functions such as rational functions (the quotient of two polynomials) or sums of exponentials are also often useful, though more difficult to handle computationally.

A wide variety of norms and fitting functions are represented in existing software for approximation. A wealth of software is available for the general least squares approximation problem. Least absolute value and minimax computations are much more difficult and relatively less software is available for these.

Most software for approximation utilizes polynomial or piecewise polynomial fitting functions, although some is also available for trigonometric and rational functions. The ultimate flexibility is offered by routines that work with user-defined fitting functions. In many cases two subprograms are needed to solve an approximation problem: one to determine the coefficients of the approximating function from the data and the second to evaluate the fitted function at one or more points chosen by the user. Occasionally subroutines for evaluating derivatives or definite integrals of the approximating function are also provided.

References

K1: Least squares approximation

In least squares curve fitting a set of \( m \) data values \( (x_i, y_i), \ i = 1, \ldots, m \) are approximated by a function \( g(x) \) which depends on \( n \) unknown parameters \( a_1, a_2, \ldots, a_n \). The set of parameter values is found which minimizes

\[
\sum_{i=1}^{m} w_i(y_i - g(x_i))^2
\]

where the \( w_i \) are suitably chosen positive weights. In statistics this is known as the regression problem. Software which performs least squares approximation with statistical analysis of the results is found in class L8. This class also overlaps with chapter G (Optimization) where software for "minimizing sums of squares" can be found.

There is an important distinction between linear and nonlinear least squares approximation. In the former the unknown parameters appear as linear coefficients of the known fitting functions, whereas in the latter they may appear in any position. For example, in the fitting function

\[
y = ae^{bx} + ce^{dx}
\]

\( a \) and \( c \) are linear parameters while \( b \) and \( d \) enter nonlinearly. This concept of linearity refers only to the unknown parameters and not to the fitting functions. These functions can be highly nonlinear functions of the \( x \) values, as in the example above.

It is also possible to impose side conditions or constraints on least squares fits. For example, one might force the fitted function to pass through a data point, or to have non-negative slope at a given point.

Linear least squares approximation problems have an equivalent matrix formulation, and hence algorithms for these problems are special cases of those for solving overdetermined systems of linear equations (described in class D9). Each of the CMLIB, IMSL, NAG, and PORT libraries have subroutines for linear least squares curve fitting using piecewise polynomials (splines). The NAG library provides subroutines using polynomial basis functions, while the IMSL library provides a subroutine based on user-defined fitting functions. Several subroutines are also available for the linearly constrained case.

In the nonlinear case iterative methods must be employed, and these are special cases of the more general optimization routines described in chapter G. Typically, such algorithms require derivatives of the function with respect to the parameters evaluated at the data points and the current parameter estimates. Some modules allow the user the option of either providing these derivatives or having them estimated by finite differences. Each of the CMLIB, IMSL, MATHWARE and NAG libraries have subprograms for nonlinear least squares problems. In addition, an interactive system called INVAR is available on the Sperry 1100 system, with versions supporting graphical output on line printers or Tektronix terminals (using DISSPLA).

References


K1: Least squares (L2) approximation

K1a: Linear least squares (search also classes D5, D6, D9)

K1a1: Unconstrained

K1a1a: Univariate data (curve fitting)
### K1a1a1: Polynomial splines (piecewise polynomials)

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMLIB subprogram library</td>
<td>(FC sublibrary)</td>
</tr>
<tr>
<td>FC</td>
<td>Fits piecewise polynomial to discrete data with equality and inequality constraints.</td>
</tr>
<tr>
<td>IMSL subprogram library</td>
<td></td>
</tr>
<tr>
<td>ICSFKU</td>
<td>Least squares approximation by cubic splines fixed knots.</td>
</tr>
<tr>
<td>ICSSCU</td>
<td>Cubic spline data smoother.</td>
</tr>
<tr>
<td>ICSSCV</td>
<td>Cubic spline data smoother (easy-to-use version).</td>
</tr>
<tr>
<td>ICSVKU</td>
<td>Least squares approximation by cubic splines - variable knots.</td>
</tr>
<tr>
<td>NAG subprogram library</td>
<td></td>
</tr>
<tr>
<td>E02BAE</td>
<td>Least-squares curve fit by cubic splines (including interpolation). Double precision version is E02BAF.</td>
</tr>
<tr>
<td>PORT subprogram library</td>
<td></td>
</tr>
<tr>
<td>DL2SF</td>
<td>Fits discrete data with a B-spline of order K, by least squares. Double precision version is DDL2SF.</td>
</tr>
<tr>
<td>DL2SW</td>
<td>Fits discrete data with a B-spline of order k, by weighted least squares. Double precision version is DDL2SW.</td>
</tr>
<tr>
<td>L2SFF</td>
<td>Obtains a weighted least square expansion of a known function in terms of B-splines of order K, at given mesh points. Double precision version is DL2SFF.</td>
</tr>
<tr>
<td>L2SFH</td>
<td>Obtains a weighted least square expansion of a known function in and its derivatives in terms of B-splines of order K at given mesh points. Double precision version is DL2SFH.</td>
</tr>
</tbody>
</table>

### K1a1a2: Polynomials

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAG subprogram library</td>
<td></td>
</tr>
<tr>
<td>E02ADE</td>
<td>Least-squares curve fit by polynomials, arbitrary data points. Double precision version is E02ADF.</td>
</tr>
<tr>
<td>E02AFE</td>
<td>Least-squares curve fit by polynomials, special data points (including interpolation). Double precision version is E02AFF.</td>
</tr>
</tbody>
</table>

### K1a1a3: Other functions (e.g., rational, trigonometric, user-specified)

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSL subprogram library</td>
<td></td>
</tr>
<tr>
<td>IFLSQ</td>
<td>Least squares approximation with user supplied functions.</td>
</tr>
</tbody>
</table>

### K1a1b: Multivariate data (surface fitting)

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAG subprogram library</td>
<td></td>
</tr>
<tr>
<td>E02CAE</td>
<td>Least-squares surface fit by polynomials, for data on lines. Double precision version is E02CAF.</td>
</tr>
<tr>
<td>E02DAE</td>
<td>Least-squares surface fit by bicubic splines. Double precision version is E02DAF.</td>
</tr>
</tbody>
</table>

### K1a2: Constrained

* Denotes easy-to-use modules.
K1a2a : Linear constraints

CMLIB subprogram library (FC sublibrary)

FC Fits piecewise polynomial to discrete data with equality and inequality constraints.
LPDP Solves least projected distance problem.
LSEI Solves linearly constrained least squares problem with equality and inequality constraints. Covariance matrix opt output.
WNLS Solves linearly constrained non-negative least squares problem.

NAG subprogram library

E02AGE Least-squares curve fit by polynomials, arbitrary data points, values and derivatives may be constrained. Double precision version is E02AGF.

K1a2b : Nonlinear constraints

K1b : Nonlinear least squares

K1b1 : Unconstrained

K1b1a : Smooth functions

K1b1a1 : User provides no derivatives

CMLIB subprogram library (NL2SN sublibrary)

NL2SN Minimizes a nonlinear sum of squares using residual values only. Double precision version is DNL2SN.

CMLIB subprogram library (SNLS1E sublibrary)

SNLS1 Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. Flexible usage, including various options for providing Jacobian. Covariance matrix is available via the subroutine SCOV.
SNLS1E Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. An easy to use driver for SNLS1. The covariance matrix is available by calling the subroutine SCOV.

IMSL subprogram library

ZXSSQ Minimum of the sum of squares of m functions in n variables using a finite difference Levenberg-Marquardt algorithm.

INVAR1 Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only.
INVAR2 Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA graphics.

NAG subprogram library

• Denotes easy-to-use modules
<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E04FCE</td>
<td>Unconstrained minimum, sum of squares, n variables (comprehensive), using function values only, combined Gauss-Newton and modified Newton algorithm. Double precision version is E04FCF.</td>
</tr>
<tr>
<td>E04FDE</td>
<td>Unconstrained minimum, sum of squares, n variables (easy-to-use), function values only, Gauss-Newton and modified Newton algorithm. Double precision version is E04PDF.</td>
</tr>
<tr>
<td>K1b1a2</td>
<td>User provides first derivatives</td>
</tr>
<tr>
<td>NL2S1</td>
<td>Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Double precision version is DNL2S1.</td>
</tr>
<tr>
<td>SNLS1</td>
<td>Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. Flexible usage, including various options for providing Jacobian. Covariance matrix is available via the subroutine SCOV.</td>
</tr>
<tr>
<td>SNLS1E</td>
<td>Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. An easy to use driver for SNLS1. The covariance matrix is available by calling the subroutine SCOV.</td>
</tr>
<tr>
<td>INVAR1</td>
<td>Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only.</td>
</tr>
<tr>
<td>INVAR2</td>
<td>Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA graphics.</td>
</tr>
<tr>
<td>A23MRT</td>
<td>Modified Marquardt procedure for minimizing a nonlinear sum of squares function.</td>
</tr>
<tr>
<td>E04GBE</td>
<td>Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Double precision version is E04GBF.</td>
</tr>
<tr>
<td>E04GCE</td>
<td>Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Double precision version is E04GCF.</td>
</tr>
<tr>
<td>E04GDE</td>
<td>Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04GDF.</td>
</tr>
<tr>
<td>E04GEE</td>
<td>Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04GEF.</td>
</tr>
<tr>
<td>K1b1a3</td>
<td>User provides first and second derivatives</td>
</tr>
<tr>
<td>E04HEE</td>
<td>Unconstrained minimum, sum of squares, n variables (comprehensive), using second derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04HEF.</td>
</tr>
<tr>
<td>E04HFE</td>
<td>Unconstrained minimum, sum of squares, n variables (easy-to-use), using second derivatives, Gauss-Newton and modified Newton algorithm. Double precision version is E04HFF.</td>
</tr>
</tbody>
</table>
K1b1b: General functions

K1b2: Constrained

K1b2a: Linear constraints

K1b2b: Nonlinear constraints

K2: Minimax ($L_\infty$) approximation

- **IMSL subprogram library**
  - IRATCU: Rational weighted Chebyshev approximation of a continuous function.
  - RLLMV: Perform linear regression using the minimax criterion.

- **NAG subprogram library**
  - E02ACE: Minimax curve fit by polynomials. Double precision version is E02ACF.
  - E02GCE: Calculates an $L_\infty$ solution to an over-determined system of linear equations. Double precision version is E02GCF.

- **PORT subprogram library**
  - BURAM: Finds the best uniform rational approximation to a given function on a specified mesh. Double precision version is DBURAM.
  - BURM1: Finds the best uniform rational approximation to a given function on a specified mesh, starting from a given initial approximation. Double precision version is DBURM1.

K3: Least absolute value ($L_1$) approximation

- **IMSL subprogram library**
  - RLLAV: Perform linear regression using the least absolute values criterion.

- **NAG subprogram library**
  - E02GAE: L1-approximation by general linear function. Double precision version is E02GAF.
  - E02GBE: L1-approximation by general linear function subject to linear inequality constraints. Double precision version is E02GBF.

K4: Other analytic approximations (e.g., Taylor polynomial, Padé)

- **NAG subprogram library**
  - E02RAE: Padé-approximants. Double precision version is E02RAF.

K5: Smoothing

- **IMSL subprogram library**
ICSMOU One-dimensional data smoothing by error detection.
ICSSCU Cubic spline data smoother.
ICSSCV Cubic spline data smoother (easy-to-use version).

STATLIB subprogram library

MOVAVG Computes a k-term symmetric moving average of a series.

K6: Service routines (e.g., mesh generation, evaluation of fitted functions) (search also class N5)

CMLIB subprogram library (BSPLINE sublibrary)

BFQAD Integrates function times derivative of B-spline from X1 to X2. The B-spline is in “B” representation. Double precision version is DBFQAD.
BSPEV Calculates the value of a spline and its derivatives at X from its “B” representation. Double precision version is DBSPEV.
BSPPP Converts from “B” representation of B-spline to piecewise polynomial representation. Double precision version is DBSPPP.
BSPVD Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. Double precision version is DBSPVD.
BSPVN Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order. Double precision version is DBSPVN.
BSQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in “B” representation. Double precision version is DBSQAD.
BVALU Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its “B” representation. Double precision version is DBVALU.
INTRV Computes the index into a knot or breakpoint sequence corresponding to a given point X. Double precision version is DINTRV.
PFQAD Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in piecewise polynomial representation. Double precision version is DPQFQAD.
PPQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in piecewise polynomial representation. Double precision version is DPPQAD.
PPVAL Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its piecewise polynomial representation. Double precision version is DPPVAL.

IMSL subprogram library

DBCEVL Bicubic spline mixed partial derivative evaluator.
DCSEVU Cubic spline first and second derivative evaluator.
DCSQDU Cubic spline quadrature.
IBCEVL Evaluation of a bicubic spline.
ICSEVU Evaluation of a cubic spline.

NAG subprogram library

E02AEE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). Double precision version is E02AEF.
E02AHE Derivative of fitted polynomial in Chebyshev series form. Double precision version is E02AHF.
E02AJE Integral of fitted polynomial in Chebyshev series form. Double precision version is E02AJF.
E02AKE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. Double precision version is E02AKF.

* Denotes easy-to-use modules
E02BBE Evaluation of fitted functions, cubic spline as E02BAE, function only. Double precision version is E02BBF.
E02BCE Evaluation of fitted functions, cubic spline as E02BAE, function and derivatives. Double precision version is E02BCF.
E02BDE Evaluation of the definite integral of a cubic spline which was obtained by using E02BAE. Double precision version is E02BDF.
E02CBE Evaluation of fitted functions, polynomial in two variables as E02CAE. Double precision version is E02CBF.
E02DBE Evaluation of fitted functions, bicubic spline as E02DAE. Double precision version is E02DBF.
E02RBE Evaluation of fitted functions, rational function as E02RAE. Double precision version is E02RBF.
E02ZAE Sort 2-d data into panels for fitting or evaluating bicubic splines. Double precision version is E02ZAF.

PORT subprogram library

BSPL1 Evaluates, at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives. Double precision version is DBSPL1.
BSPLD Evaluates at a given set of points in a specified mesh interval, basis splines and their derivatives. Double precision version is DBSPLD.
BSPLI Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. Double precision version is DBSPLI.
BSPLN Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. Double precision version is DBSPLN.
CSPFE Evaluates a cubic spline function which has already been fit to n input data pairs (x,y) by CSPFI. Double precision version is DCSPFE.
DLUMD Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points.
EEBSF Estimates the error in a given B-spline fit to a function, f, by refining the mesh. Double precision version is DEEBSF.
EEBSI Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. Double precision version is DEEBSI.
EESFF Finds the maximum absolute error in a given B-spline fit to a function, f. Double precision version is DEESSF.
EESFI Finds the maximum absolute error in a given B-spline fit to a function, f, on a set of user selected intervals. Double precision version is DEESFI.
IDMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.
ILUMB Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. Double precision version is IDLUMB.
ILUMD Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. Double precision version is IDLUMD.
IMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval.
IPUMB Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. Double precision version is IDPUMB.
IPUMD Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. Double precision version is IDPUMD.
IUMB Given interval endpoints, this generates a uniform mesh for B-spline use. Double precision version is IDUMB.
IUMD Given interval endpoints, this generates a uniform mesh. Double precision version is IDUMD.
LUMB Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. Double precision version is DLUMB.
LUMD: Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. Double precision version is IDUMD.

MNPB: Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. Double precision version is DMNPB.

PUMB: Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. Double precision version is DPUMB.

PUMD: Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Double precision version is DPUMD.

SPLN1: Evaluates a function and derivatives described previously by an expansion in terms of B-splines. Double precision version is DSPLN1.

SPLN2: Evaluates a function described by a previously determined expansion in B-splines. More flexible than SPLN1. Double precision version is DSPLN2.

SPLND: Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. Double precision version is DSPLND.

SPLNE: Evaluates at a set of points, a function described by a previously determined expansion in terms of B-splines. Double precision version is DSPLNE.

SPLNI: Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Double precision version is DSPLNI.

UMB: Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. Double precision version is DUMB.

UMD: Given interval endpoints, this generates a uniform mesh of distinct points. Double precision version is DUMD.
L: Statistics, Probability

The software cataloged in this chapter covers a wide spectrum of statistical computations—from elementary summary statistics to sophisticated analyses such as regression. Subroutine libraries, program libraries, and interactive systems are each represented. Output ranges from a single number to encyclopedic printouts to graphical displays. The following choices should be considered when selecting statistical software.

1. **Accuracy/precision.** Accuracy of numerical results is crucial. Two different formulas for computing a variance, for example, though mathematically equivalent, may produce different results on computers. Users should select software with the appropriate precision, and that software should report precision problems.

2. **Documentation.** Computational formulas and procedures should be fully documented.

3. **Computer.** Because of physical size, libraries are not necessarily available on all machine types (large mainframes, minicomputers, and microcomputers). Proprietary libraries are commonly leased only for specific machines.

4. **Size of data set.** Whether or not a data set of a particular size can fit on a given computer must be considered when selecting statistical software. Some software is available for analyzing large data sets which cannot be stored in direct-access memory. Programs in a program library and interactive systems may or may not have data set size constraints.

5. **Data entry.** Since some libraries are unavailable on small computers, and data are sometimes difficult (bearing on impossible) to transport from small to large computers, trouble may arise in bringing software and data together.

6. **Analyzer's time.** A user of statistical software needs to know the language used to communicate with the software. This language usually is Fortran in the case of subroutine libraries, while each stand-alone program library and each interactive system has its own command language. In order to save time, therefore, users are often inclined to use familiar software. Users should not be reluctant to use unfamiliar software, however, because a working knowledge of any software language can generally be quickly acquired. On the other hand, the user is strongly warned against applying the "try everything" approach to the statistical analysis of data, which may be a temptation given the ease of software use.

The analysis capabilities of any one statistical software library are not entirely duplicated in another, although there is overlap. While the subroutines from several libraries can usually easily be incorporated in a single program, it is more difficult to use several stand-alone programs or interactive systems for one analysis.

7. **Amount of output.** Some statistical software produces encyclopedic printed output—this is especially true of programs run in batch mode. Software designed for interactive use commonly provides relatively terse output, since a screen image provides a natural bound on the amount of output useable at any one time. The form of printed output from subroutines can vary from none (results are returned via the call sequence) to encyclopedic.

8. **Storage/time trade-off.** Some statistical computations make efficient use of time, others of storage, and commonly not of both simultaneously. This trade-off should be considered when selecting software for an analysis involving a data set of a given size and frequency of execution.

9. **Graphics.** Researchers are strongly encouraged to plot their data—a picture may reveal useful information not otherwise available through numerical summaries and statistical analyses.

The libraries providing statistical software in this edition of GAMS are:

Subroutine libraries: CMLIB (several sublibraries), DATAPAC, IMSL, NAG, STATLIB

Program libraries: BMDP, INVAR

Interactive systems: Minitab, Spectrían

Consult the Library Reference for further information about these libraries.

Each user-callable subroutine in a subroutine library, and each program in a program library, is an entity which is cataloged at one or more classes in the classification scheme, and is an entry in the Module Dictionary. Interactive systems have been handled differently. For the Minitab system, some of the 150 commands are quite elementary, and those commands are grouped under the name "Minitab." For example, there is a "Minitab" entry under class L2a for data transformation which lists approximately two dozen Minitab commands. The whole Spectrían system is classified at L10f for spectral analysis.

The following software is scheduled for inclusion in future editions of GAMS:

- DATAPLOT (Filliben, 1980), an interactive system for statistical data analysis and Tektronix graphics
- GLIM (Baker and Nelder, 1978), an interactive system for analysis of variance and other statistical analyses
- OMNITAB (Hogben and Peavy, 1971, 1977), an interactive system for statistical and other types of data analyses
- ROSEPACK (Coleman et al., 1980), an interactive system for robust regression
- SPSS (Nie et al., 1975; Hull and Nie, 1981), a program for a wide variety of statistical analyses.
REFERENCES


L: Statistics, probability

L1: Data Summarization

Data summarization is often the first step in the analysis of a data set. It attempts to provide insight into the data through one or a few values which characterize, for example, location or correlation. Classical summary statistics satisfy some optimality criteria when the data are, say, normally distributed. Nonparametric or distribution-free statistics often provide useful information when the distribution of the data is not known.

The software in this chapter is purely for summarizing data, not for elementary statistical analyses such as inference or hypothesis testing (that is in chapter L4), nor for more advanced statistical analyses such as in chapters L7 through L15.

Chapter L1 is first organized by the type of data: one univariate quantitative sample (L1a), one univariate qualitative (proportional) sample (L1c), two or more univariate samples or one multivariate sample (L1e), and two or more multivariate samples (L1i). The labeling of these subchapters parallels that of chapter L4; there is no need for a class for two or more univariate quantitative samples here since software for this task is the same as the software for one multivariate sample, but there is a need to distinguish in chapter L4. There is currently no software in GAMS for summarizing qualitative data; see class L4c for software for summarization and inference. There is also currently no software classified in L1i.

Software is classified both at leaves of the tree and at higher levels; the former compute individual summary statistics while the latter compute a variety of such statistics.

Consult the references listed in the introduction to chapter L for more information about data summarization.

January 1984

L1: Data summarization
### Lia:
One univariate quantitative sample

### Llal:
Ungrouped data

#### BMDP program library

**P1D**
Prints univariate statistics (mean, standard deviation, std. error of mean, coefficient of variation, extreme values, extreme z-scores, range) for each variable. Options: statistics for each level of each grouping variable, sorting, printing all cases OR only cases with values missing or values outside specified limits.

**P2D**
For each variable, prints frequency and percent for each distinct value; mean, median, mode, standard deviation, std. errors of mean and median, skewness, kurtosis, half interquartile range; histogram, and stem-and-leaf plot. Options: initially round or truncate, three robust location estimates.

#### DATAPAC subprogram library

**LOC**
Computes 4 estimates (midrange, mean, midmean, and median) of the data in the input vector X.

#### IMSL subprogram library

**BDLTV**
Produce letter value summary.

**BEIUGR**
Estimation of basic statistical parameters using ungrouped data.

**NMTIE**
Tie statistics, given a sample of observations.

#### MINITAB interactive system

**MINITAB**
Minitab’s vector summarization commands include COUNT, SUM, MEAN, MAX, MIN, MEDIAN, N, NMISS (number of missing values), STDEV, SSQ (sum of squares), DESCRIBE (N, MEAN, MEDIAN, STDEV, MAX, MIN, 5% trimmed mean, quartiles) for columns or rows (use prefix R, e.g., RMEAN) of data in the Minitab worksheet.

#### NAG subprogram library

**G01AAE**
Simple descriptive statistics, one variable, from raw data. Double precision version is G01AAF.

#### STATLIB subprogram library

**STATS**
Computes 53 descriptive statistics for a single random sample.

**STATSS**
Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation and computed results returned to the user.

**STATSW**
Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation.

#### DATAPAC subprogram library

**MEAN**
Computes the sample mean of the data in the input vector X.

**MEDIAN**
Computes the sample median of the data in the input vector X.

**MIDM**
Computes the sample midmean, i.e. the sample 25% (on each side) trimmed mean of the data in the input vector X.

**TRIM**
Computes the sample trimmed mean of the data in the input vector X.

**WIND**
Computes the sample Windsorized mean of the data in the input vector X.
Llalb : Dispersion

**DATAPAC subprogram library**

- MIDR: Computes the sample midrange of the data in the input vector X.
- RANGE: Computes the sample range of the data in the input vector X.
- RELSD: Computes the sample relative standard deviation of the data in the input vector X.
- SCALE: Computes 4 estimates of the scale (variation, scatter, dispersion) of the data in the input vector X.
- SD: Computes the sample standard deviation (with denominator N-1) of the data in the input vector X.
- VAR: Computes the sample variance (with denominator N-1) of the data in the input vector X.

Llalc : Shape

**DATAPAC subprogram library**

- STMOM3: Computes the sample standardized third central moment of the data in the input vector X.
- STMOM4: Computes the sample standardized fourth central moment of the data in the input vector X.

Llald : Distribution, density

**BMDP program library**

- P4D: Counts frequency of each number, letter, or symbol in single-column fields (A1 format). Options: input case label variables in A4 format, diagnostic printing useful in preliminary data screening. Specified characters may be replaced by blanks or symbols.

**DATAPAC subprogram library**

- COUNT: Computes the number of observations between XMIN and XMAX (inclusively) in the input vector X.
- FREQ: Computes the sample frequency and sample cumulative frequency for the data in the input vector X.
- PROPOR: Computes the sample proportion which is the proportion of data between XMIN and XMAX (inclusively) in the input vector X.
- SAMPP: Computes the sample 100P percent point (where P is between 0.0 and 1.0, exclusively) of the data in the input vector X.

L1a2 : Ungrouped data with missing values

**BMDP program library**

- P1D: Prints univariate statistics (mean, standard deviation, std. error of mean, coefficient of variation, extreme values, extreme z-scores, range) for each variable. Options: statistics for each level of each grouping variable, sorting, printing all cases OR only cases with values missing or values outside specified limits.

- P2D: For each variable, prints frequency and percent for each distinct value; mean, median, mode, standard deviation, std. errors of mean and median, skewness, kurtosis, half interquartile range; histogram, and stem-and-leaf plot. Options: initially round or truncate, three robust location estimates.

**IMSL subprogram library**

- BESTAT: Computations of basic univariate statistics from data possibly containing missing values, with weighting on option.
MINITAB interactive system

MINITAB
Minitab's vector summarization commands include COUNT, SUM, MEAN, MAX, MIN, MEDIAN, N, NMISS (number of missing values), STDEV, SSQ (sum of squares), DESCRIBE (N, MEAN, MEDIAN, STDEV, MAX, MIN, 5% trimmed mean, quartiles) for columns or rows (use prefix R, e.g., RMEAN) of data in the Minitab worksheet.

L1a3 : Grouped data

IMSL subprogram library

BEGRPS
Moments estimation for grouped data with and without Sheppards corrections.

BEIGRP
Estimation of basic statistical parameters using grouped data.

NAG subprogram library

G01ADE
Simple descriptive statistics, one variable, from frequency table. Double precision version is G01ADF.

L1a3a : Location

L1a3b : Dispersion

L1a3c : Shape

L1e : One univariate qualitative (proportional) sample

L1e : Two or more univariate samples or one multivariate sample

L1e1 : Ungrouped data

IMSL subprogram library

BECOR

BECORI
Estimates of means, standard deviations, and correlation coefficients (in-core version).

BECOVM
Means and variance-covariance matrix.

BECVL
Variances and covariances of linear functions (out-of-core version).

BECVLI
Variances and covariances of linear functions (in-core version).

MINITAB interactive system

TABLE
Produces and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarising (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output.

NAG subprogram library

G01ABE
Simple descriptive statistics, two variables, from raw data. Double precision version is G01ABF.

G02BAE
Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X.
Double precision version is G02BAF.

**G02BDE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array \( X \). Double precision version is G02BDF.

**G02BGE** Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array \( X \). Double precision version is G02BGF.

**G02BKE** Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array \( X \). Double precision version is G02BKF.

**L1a :** Location

**L1b :** Correlation

**DATAPAC subprogram library**

**CORR** Computes the sample correlation coefficient between the sets of data in the input vectors \( X \) and \( Y \).

**SPCORR** Computes the Spearman rank correlation coefficient between the two sets of data in the input vectors \( X \) and \( Y \).

**MINITAB interactive system**

**CORRELATION** Calculates the Pearson product moment correlation coefficient between two or more pairs of vectors, handles missing values, and optionally saves results.

**NAG subprogram library**

**G02BNE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array \( X \), overwriting \( X \) with the ranks of the observations. Double precision version is G02BNF.

**G02BPE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array \( X \) containing missing values, overwriting \( X \) with the ranks of the observations. Double precision version is G02BPF.

**G02BQE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array \( X \), preserving \( X \). Double precision version is G02BQF.

**G02BRE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array \( X \) containing missing values, preserving \( X \). Double precision version is G02BRF.

**G02BSE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array \( X \) containing missing values, preserving \( X \). Double precision version is G02BSF.

**L1c :** Ungrouped data with missing values

**BMDP program library**

**P8D** Four methods to compute covariance and correlation matrices when data contain missing values or values out of range. Options: weights, summary statistics, save results, pairwise t-tests based on the pattern of incomplete data.

**PAM** Describes pattern of invalid values (missing or out of range) for multivariate data. Options: weights, grouping, estimates covariance and correlation matrices by one of three methods (including maximum likelihood), replace invalid values using means or one of several regression procedures, plots, save results.

**IMSL subprogram library**
BECOVW  Means and variance-covariance or correlation matrix from data possibly containing missing observations, with weighting on option.

BEMMI  Estimates of means, std. devs., correlation coefficients, and coefficients of skewness and kurtosis from a data matrix containing missing observations (in-core version).

BEMMO  Estimates of means, std. devs., correlation coefficients, and coefficients of skewness and kurtosis from a data matrix containing missing observations (out-of-core version).

MINITAB interactive system

CORRELATION  Calculates the Pearson product moment correlation coefficient between two or more pairs of vectors, handles missing values, and optionally saves results.

TABLE  Produces and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarizing (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output.

NAG subprogram library

G02BBE  Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. Double precision version is G02BBF.

G02BCE  Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. Double precision version is G02BCF.

G02BEE  Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. Double precision version is G02BEF.

G02BFE  Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in array X containing missing values. Double precision version is G02BFF.

G02BHE  Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BHF.

G02BJE  Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BJF.

G02BLE  Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BLF.

G02BME  Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. Double precision version is G02BMF.

L1e3 : Grouped data

L1f :  Two or more multivariate samples

L2: Data Manipulation

This chapter contains software for transforming data (L2a), grouping or tallying data (L2b), sampling data (L2c), and selecting a subset of a data set (L2d). The transformation software handles univariate data (e.g., commonly used transformations), multivariate data, and transformations for use in nonparametric (distribution-free) analyses. Software
January 1984

GAMS: Modules by Class B 117

for transforming time series is classified at L10a. Software for sorting and ranking is classified at N6. Data creation commands are currently classified at L2.

Consult the references listed in the introduction to chapter L for more information about data manipulation.

January 1984

<table>
<thead>
<tr>
<th>L2</th>
<th>Data manipulation (search also class N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOIN</td>
<td>Merges constants and/or vectors into vectors.</td>
</tr>
<tr>
<td>SET</td>
<td>Create a constant vector or a vector of integers in increments of 1 or more or with other patterns.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L2a</th>
<th>Transform (search also class L10a for time series transformations and class N6 for sorting, ranking)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1S</td>
<td>At each pass through the data, computes univariate statistics (choose means, standard deviations, geometric means, harmonic means, extreme values), and transforms or edits the data using statistics computed in the previous pass. Options: printing, save results.</td>
</tr>
<tr>
<td>DISCR2</td>
<td>Discretizes the data in the vector X into NUMCLA classes.</td>
</tr>
<tr>
<td>DISCR3</td>
<td>Discretizes the data in the vector X into NUMCLA classes.</td>
</tr>
<tr>
<td>DISCRE</td>
<td>Discretizes the data of the vector X according to class width.</td>
</tr>
<tr>
<td>REPLAC</td>
<td>Replaces (with the value XNEW) all observations in the vector X which are inside the interval [XMIN, XMAX].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IMSL subprogram library</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDTRGI</td>
</tr>
<tr>
<td>BDTRGO</td>
</tr>
<tr>
<td>RLGQMI</td>
</tr>
<tr>
<td>RLGQMO</td>
</tr>
<tr>
<td>RLPOL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MINITAB interactive system</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTER</td>
</tr>
<tr>
<td>DIFFERENCES</td>
</tr>
<tr>
<td>NSCORES</td>
</tr>
<tr>
<td>WALSH</td>
</tr>
</tbody>
</table>
**L2b:** Group

**IMSL subprogram library**

- **BDCOU1** Tally of observations into a one-way frequency table.
- **BDCOU2** Tally of observations into a two-way frequency table.
- **BDTAB** Computations of frequencies of multivariate data.
- **BDTWT** Computations of a two-way frequency table.
- **GTDDU** D-square tally.
- **GTPL** Poker test tally of hand types and statistics.
- **GTPR** Tally of coordinates of pairs (or lagged pairs) of random numbers.
- **GTRTN** Tally of number of runs up and down.
- **GTTRT** Tally for triplets test.

**NAG subprogram library**

- **G01AEE** Frequency table from raw data. Double precision version is G01AEF.

**L2c:** Sample

**IMSL library**

- **SSPAND** Simple random sampling with proportion data-inferences regarding the population proportion and total.
- **SSPBLK** Stratified random sampling with proportion data - inferences regarding the population proportion and total.
- **SSRBLK** Stratified random sampling with continuous data-inferences regarding the population proportion and total using ratio or regression estimation.
- **SSSBLK** Stratified random sampling with continuous data - inferences regarding the population mean and total using ratio or regression estimation.
- **SSSAND** Simple random sampling with continuous data - inferences regarding the population mean and total.
- **SSSCAN** Single stage cluster sampling with continuous data - inferences regarding the population mean and total.
- **SSSEST** Two-stage sampling with continuous data and equisized primary units - inferences regarding the population mean and total.

**L2d:** Subset

**DATAPAC library**

- **DELETE** Deletes all observations in the vector X which are inside the interval [XMIN, XMAX].
- **RETAIN** Retains all observations in the vector X which are inside the interval [XMIN, XMAX].
- **SUBSE1** Carry over into Y all observations of vector X for which the corresponding elements in vector D are in the interval [DMIN,DMAX].
- **SUBSE2** Carry over into Y all observations of vector X for which the corresponding elements in vector D1 are in the inclusive interval [D1MIN,D1MAX] and also for which the corresponding elements in D2 are in the interval [D2MIN,D2MAX].
- **SUBSET** Retain all observations in vector X for which the corresponding elements in vector D are in the interval...
[DMIN, DMAX].

**Minitab Interactive System**

**Minitab**  Minitab’s subsetting commands include PICK, CHOOSE, and OMIT for selecting or deleting entries in a vector in a Minitab worksheet.

**NAG Subprogram Library**

**G02CEE**  Service routines for multiple linear regression, select elements from vectors and matrices. Double precision version is G02CEF.

### L3: Graphics

The usefulness of graphical analysis of data cannot be overemphasized. Important but perhaps subtle information about data not apparent in numerical output can often be easily discovered through inspection of graphical displays.

This edition of GAMS only includes software for line-printer plots. High-resolution graphics software will appear in this chapter and/or in chapter Q (general-purpose graphics) in future editions.

**References**


### L3 : Graphics (search also class Q)

**L3a : Histograms**

**Datapac Subprogram Library**

**HIST**  Produces 2 histograms (with differing class widths) of the data in the input vector X.

**PLOTU**  Produces 4 plots: data plot (X(I) versus I), autoregression plot (X(I) versus X(I-1)), histogram, and normal probability plot.

**IMSL Subprogram Library**

**USHHST**  Print a horizontal histogram.

**USHST**  Print a vertical histogram.

**USHST2**  Print a vertical histogram, plotting two frequencies with one bar of the histogram.

**Minitab Interactive System**

**HISTOGRAM**  Prints a histogram of the values in each of one or more vectors, with optional user-specification of the first midpoint and the interval width.

**NAG Subprogram Library**

**G01AJE**  Prints a histogram on a character printing device, with user control over size, positioning, and the range of data values included. Double precision version is G01AJF.

**Statlib Subprogram Library**

**HISTO**  Produces a histogram and summary statistics.

**HISTOC**  Produces a histogram and summary statistics, with user control of the number of cells, and of the upper and lower histogram boundaries.
**L3b: Distribution functions**

**BMDP program library**

**P5D**
Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data — either separately or combined in one plot. Plot options.

**IMSL subprogram library**

**USPC**
Print a sample pdf, a theoretical pdf and confidence band information; plot these on option.

**USPDF**
Plot of two sample probability distribution functions against their spectra.

**USPLO**
Printer plot of up to ten functions.

**USPLOD**
Printer plot of up to ten functions.

**L3c: Scatter diagrams**

**BMDP program library**

**P6D**
Bivariate (scatter) plots. Options: several variables, or subsets of one variable (symbols identify group membership), on the same plot; prints correlation and linear regression statistics (line is marked on plot frame); user control for plot size, scales, and symbols.

**L3c1: Y vs. X**

**DATAPAC subprogram library**

**PLOT**
Yields a one-page printer plot of Y(I) versus X(I).

**PLOT6**
Yields a one-page printer plot of Y(I) versus X(I) for specified axis limits.

**ILOTS**
Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data.

**PLOTST**
Yields a narrow-width (71-character) of Y(I) versus X(I) for a subset of the data.

**PLOTT**
Yields a narrow-width (71-character) plot of Y(I) versus X(I).

**MINITAB interactive system**

**PLOT**
Prints a scatter diagram, with optional scale specification.

**NAG subprogram library**

**G01AGE**
Line printer scatter plot of two variables. Double precision version is G01AGF.

**STATLIB subprogram library**

**PLT**
Displays a 50x100 character line printer scatter plot.

**PLTH**
Displays a 50x50 character line printer scatter plot.

**PLTHL**
Displays a 50x50 character line printer scatter plot with user control of plot limits.

**PLTL**
Displays a 50x100 character line printer scatter plot with user control of plot limits.

**L3c2: Symbol plots**

**DATAPAC subprogram library**

**PLOT10**
Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data, with special plot characters,
and with specified axis limits and labels.

PLOT7  Yields a one-page printer plot of \( Y(I) \) versus \( X(I) \) with special plot characters and for specified axis limits.

PLOTS8  Yields a one-page printer plot of \( Y(I) \) versus \( X(I) \) with special plot characters for a subset of the data with specified axis limits.

PLOT9  Yields a one-page printer plot of \( Y(I) \) versus \( X(I) \) with special plot characters and for specified axis limits and axis labels.

PLOT C  Yields a one-page printer plot of \( Y(I) \) versus \( X(I) \) with special plotting characters.

PLOTCT  Yields a narrow-width (71-character) plot of \( Y(I) \) versus \( X(I) \) with special plotting characters.

PLOTC  Yields a one-page printer plot of \( Y(I) \) versus \( X(I) \) with special characters for specified axis limits and axis labels.

PLOTSCT  Yields a narrow-width (71-character) plot of \( Y(I) \) versus \( X(I) \) with special plot characters and a subset of the data.

PLTSCT  Yields a narrow-width (71-character) plot of \( Y(I) \) versus \( X(I) \) with special plot characters and a subset of the data.

**MINITAB interactive system**

LLOT  Prints a letter plot with symbols corresponding to numerical "tag" values. Scale specification is optional.

TLOT  Prints pseudo three-dimensional plot of \( y \) versus \( x \) versus \( z \), with symbols indicating the values of \( z \), and with optional scale specification.

**STATLIB subprogram library**

SPLT  Displays a \( 50 \times 100 \) character line printer scatter plot with user control of the plotting symbol used for each point.

SPLTH  Displays a \( 50 \times 50 \) character line printer scatter plot with user control of the plotting symbol used for each point.

SPLTHL  Displays a \( 50 \times 50 \) character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point.

SPLTL  Displays a \( 50 \times 100 \) character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point.

**Multiple plots**

**MINITAB interactive system**

MPLOT  Prints multiple scatter diagrams on the same axis.

**STATLIB subprogram library**

MPLT  Displays a \( 50 \times 100 \) character line printer plot of several dependent variables vs. a common independent variable.

MPLTH  Displays a \( 50 \times 50 \) character line printer plot of several dependent variables vs. a common independent variable.

MPLTHL  Displays a \( 50 \times 50 \) character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.

MPLTL  Displays a \( 50 \times 100 \) character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.

**Probability plots**

**IMSL subprogram library**

USPRP  Probability plot.
L3c4b: Beta, binomial

L3c4c: Cauchy, chi-squared

DATAPAC subprogram library

CAUPLT Generates a Cauchy probability plot with median = 0 and 75% point = 1.

CHSPLT Generates a chi-squared probability plot with integer degrees of freedom parameter value = NU.

L3c4d: Double exponential

DATAPAC subprogram library

DEXPLT Generates a double exponential (Laplace) probability plot with mean = 0 and standard deviation = sqrt(2).

L3c4e: Exponential, extreme value

DATAPAC subprogram library

EV1PLT Generates an extreme value type 1 probability plot with mean = Euler's number = 0.57721566 and standard deviation = pi/sqrt(6).

EV2PLT Generates an extreme value type 2 probability plot with tail length parameter = GAMMA.

EXPPLT Generates an exponential probability plot with mean = 1 and standard deviation = 1.

L3c4f: F distribution

L3c4g: Gamma, geometric

DATAPAC subprogram library

GAMPLT Generates a gamma probability plot with tail length parameter = GAMMA, mean = GAMMA, and standard deviation = sqrt(GAMMA).

GEOPLT Generates a geometric probability plot with parameter P.

L3c4h: Halfnormal

BMDP program library

P5D Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data - either separately or combined in one plot. Plot options.

DATAPAC subprogram library

HFNPLT Generates a halfnormal probability plot with mean = sqrt(2/pi) and standard deviation = 1.
<table>
<thead>
<tr>
<th>L3c41 :</th>
<th>Lambda, logistic, lognormal</th>
</tr>
</thead>
</table>

**DATAPAC subprogram library**

- `LAMPLT` Generates a (Tukey) lambda distribution probability plot with tail length parameter `ALAMBA`.
- `LGNPLT` Generates a lognormal probability plot with mean $= \sqrt{e}$.
- `LOGPLT` Generates a logistic probability plot with mean $= 0$ and standard deviation $= \pi/\sqrt{3}$.

<table>
<thead>
<tr>
<th>L3c4n :</th>
<th>Negative binomial, normal</th>
</tr>
</thead>
</table>

**BMDP program library**

- `P5D` Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data – either separately or combined in one plot. Plot options.

**DATAPAC subprogram library**

- `NORPLT` Generates a normal (Gaussian) probability plot with mean $= 0$ and standard deviation $= 1$.
- `PLOTU` Produces 4 plots: data plot $(X(I)$ versus $I$), autoregression plot $(X(I)$ versus $X(I-1))$, histogram, and normal probability plot.

**NAG subprogram library**

- `G01AHE` Line printer scatter plot of one variable against normal scores. Double precision version is `G01AHF`.

<table>
<thead>
<tr>
<th>L3c4p :</th>
<th>Pareto, Poisson</th>
</tr>
</thead>
</table>

**DATAPAC subprogram library**

- `PARPLT` Generates a Pareto probability plot with tail length parameter `GAMMA`.
- `POIPLT` Generates a Poisson probability plot with tail length parameter `ALAMBA`, mean $= ALAMBA$ and standard deviation $= \sqrt{ALAMBA}$.

<table>
<thead>
<tr>
<th>L3c4t :</th>
<th>t distribution</th>
</tr>
</thead>
</table>

**DATAPAC subprogram library**

- `TPLT` Generates a Student's t probability plot with degrees of freedom parameter `NU`.

<table>
<thead>
<tr>
<th>L3c4u :</th>
<th>Uniform</th>
</tr>
</thead>
</table>

**DATAPAC subprogram library**

- `UNIPLT` Generates a uniform probability plot on the unit interval $(0,1)$ with mean $= 0.5$ and standard deviation $= \sqrt{(1/12)}$.

<table>
<thead>
<tr>
<th>L3c4w :</th>
<th>Weibull</th>
</tr>
</thead>
</table>

**DATAPAC subprogram library**
**WEIPLT** Generates a Weibull probability plot with tail length parameter GAMMA.

**L3c5** Time series plots (X(i) vs. i, vertical, lag)

**DATAPAC subprogram library**

- **PLOTU** Produces 4 plots: data plot (X(I) versus I), autoregression plot (X(I) versus X(I-1)), histogram, and normal probability plot.
- **PLOTX** Yields a one-page printer plot of X(I) versus I.
- **PLOTXT** Yields a narrow-width (71-character) plot of X(I) versus I.
- **PLOTXX** Yields a one-page printer plot of X(I) versus X(I-1) for testing autocorrelation.
- **PLTXXT** Yields a narrow-width (71-character) plot of X(I) versus X(I-1) for testing autocorrelation.

**MINITAB interactive system**

- **TSPlot** Prints a scatter diagram of a time series, optionally using symbols modulo the period. Handles missing values.

**STATLIB subprogram library**

- **VPLT** Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).
- **VPLT2** Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis).
- **VPLT2L** Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.
- **VPLTB** Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).
- **VPLTBL** Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.
- **VPLTL** Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

**L3d** EDA graphics

**IMSL subprogram library**

- **USBOX** Print a boxplot (k samples).
- **USSLF** Print a stem-and-leaf display.

**MINITAB interactive system**

- **BOXPLOT** Prints boxplots - median, hinges, inner and outer fences - for one or more levels. Options: form of plots, notches (confidence interval for population medians).
- **CPlot** Prints a scatter diagram which condenses as many as 10 lines of plot into one line and trims extreme x- and y-values. Option: form of output.
- **CTABLE** Prints a coded two-way table, each cell of which is coded with one character for features MAXIMUM, MINIMUM, or EXTREME, and codes for values between the hinges, between hinges and inner fences, between inner and outer fences, and beyond the outer fences.
- **LVALS** Prints letter-value display - median, hinges, eighths, etc., and optionally saves results.
- **ROOTOGRAM** Prints a suspended rootogram, i.e. a histogram which has been fit with a Gaussian distribution.
The distinction between this chapter and chapter L1 (for data summarization) is that while the same statistics are often computed both for summarization and for inference and testing, this chapter has the additional feature that assumptions are made about the distribution of the data, and either those assumptions are tested or inferences are made based on the assumptions. Software for more sophisticated statistical analyses are classified in chapters L7 through L15. The organization of this chapter parallels that of chapter L1. Thus, the software is first distinguished by the number of populations and the dimension of the data (e.g., one univariate quantitative sample (L4a)).

Chapter L4ala contains software which does both sampling and inference – it may be appropriate for analysis of large data sets. Software for analyzing two or more univariate samples, some of which may be quantitative and other qualitative, is classified at L4d. Software for correlation analysis is classified both at L4e (one multivariate sample) and L11 (correlation analysis). Software is classified both at leaves of the tree and at higher levels in this chapter; the latter perform a variety of testing and inference calculations.

Two general maximum likelihood parameter estimation programs are P3R and PAR from BMDP. Consult the references listed in the introduction to chapter L for more information about elementary inference and hypothesis testing.

January 1984

L4 : Elementary statistical inference, hypothesis testing

L4a : One univariate quantitative sample

BMDP program library

P3D One-sample t-test to test if one group mean is zero (e.g., matched pairs); two-sample t test with and without equal variances assumption, Levene's test for equal variances, histograms. Options: trimmed t test, Hotelling's T-squared and Mahalanobis' D-squared, within-group correlations, data listing.

L4a1 : Ungrouped data

L4ala : Parameter estimation

DATAPAC subprogram library

TAIL Performs a symmetric distribution tail length analysis on the data in the input vector X.

IMSL library

SSRAND Simple random sampling with continuous data - inferences regarding the population mean and total using ratio or regression estimation.

SSRBLK Stratified random sampling with continuous data-inferences regarding the population mean and total using ratio or regression estimation.

SSSAND Simple random sampling with continuous data - inferences regarding the population mean and total.

SSSBLK Stratified random sampling with continuous data - inferences regarding the population mean and total.

SSSCAN Single stage cluster sampling with continuous data - inferences regarding the population mean and total.
Two-stage sampling with continuous data and equisized primary units - inferences regarding the population mean and total.

**BINOMIAL**

**IMSL subprogram library**

**BELBIN** Interval estimate of the parameter p of the binomial distribution.

**EXTREME VALUE**

**DATAPAC subprogram library**

**EXTREM** Performs an extreme value analysis on the data in the input vector X.

**NORMAL**

**DATAPAC subprogram library**

**NOROUT** Performs a normal outlier analysis on the data in the input vector X.

**IMSL subprogram library**

**BEMNON** Location (mean) inferences using a sample from a normal population with known variance.

**BEMSON** Mean and variance inferences using a sample from a normal population.

**BENSON** Variance inferences using a sample from a normal population with known mean.

**GTNOR** Test for normality of random deviates.

**MINITAB interactive system**

**TINTERVAL** Calculates a t-confidence interval with specified percent confidence.

**TTEST** Performs one- or two-sided t-tests.

**ZINTERVAL** Calculates a z-confidence interval with specified percent confidence and standard deviation.

**ZTEST** Performs a one- or two-sided z-test for a specified standard deviation.

**POISSON**

**IMSL subprogram library**

**BELPOS** Interval estimate of the parameter lambda of the Poisson distribution.

**UNIFORM**

**IMSL subprogram library**

**GTMNT** Moments and standardized moments of uniform random numbers.
January 1984

GAMS: Modules by Class

**L4a1a23**: Weibull

**DATAPAC subprogram library**

**WEIB** Performs a Weibull distribution analysis on the data in the input vector X.

**L4a1b**: Distribution-free (nonparametric) analysis

**BMDP program library**

**P3S** Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskall-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall’s coefficient of concordance, Kendall and Spearman rank-correlation coefficients.

**IMSL subprogram library**

**NBSIGN** Sign test (for percentiles).

**MINITAB interactive system**

**WINTERVAL** Calculates a one-sample Wilcoxon rank estimate and confidence interval for the center of a symmetric distribution.

**WTTEST** Performs one-sample one- or two-sided Wilcoxon signed-rank tests.

**L4a1c**: Goodness-of-fit tests

**IMSL subprogram library**

**GFIT** Chi-squared goodness of fit test.

**GTCN** Sample size or number of class intervals determination for chi-squared test applications.

**GTD2T** The d-square test.

**GTPOK** The poker test.

**NKS1** Kolmogorov-Smirnov one-sample test.

**NAG subprogram library**

**G08CAE** Kolmogorov-Smirnov one-sample distribution test. Double precision version is G08CAF.

**L4a1d**: Tests on sequences of numbers

**DATAPAC subprogram library**

**RUNS** Performs a run analysis of the data in the input vector X.

**IMSL subprogram library**

**GTPST** Pairs test or Goods serial test.

**GTRN** Runs test.

**GTTT** Triplets test.

**NBCYC** Noethers test for cyclical trend.

**NBSDL** Cox and Stuart sign test for trends in dispersion and location.

**MINITAB interactive system**
**RUNS**
Performs a two-sided runs test.

**L4alc : Density and distribution function estimation**

**IMSL subprogram library**

**NDKER** Nonparametric probability density function (one dimensional) estimation by the kernel method.

**NDMPLE** Nonparametric probability density function (one dimensional) estimation by the penalized likelihood method.

**USPC** Print a sample pdf, a theoretical pdf and confidence band information; plot these on option.

**L4alf : Tolerance limits**

**DATAPAC subprogram library**

**TOL** Computes normal and distribution-free tolerance limits for the data in the input vector X.

**L4a2 : Ungrouped data with missing values**

**IMSL subprogram library**

**BESTA2** Computations of confidence intervals and other basic statistics using output from IMSL routine BESTAT.

**MINITAB interactive system**

**TINTERVAL** Calculates a t-confidence interval with specified percent confidence.

**TTEST** Performs one- or two-sided t-tests.

**ZINTERVAL** Calculates a z-confidence interval with specified percent confidence and standard deviation.

**ZTEST** Performs a one- or two-sided z-test for a specified standard deviation.

**L4a3 : Grouped data**

**L4a3a : Parameter estimation**

**L4a3a14 : Normal**

**IMSL subprogram library**

**OTMLNR** Maximum likelihood estimation from grouped and/or censored normal data.

**L4b : Two or more univariate quantitative samples**

**BMDP program library**

**P3D** One-sample t-test to test if one group mean is zero (e.g., matched pairs); two-sample t test with and without equal variances assumption, Levene's test for equal variances, histograms. Options: trimmed t
test, Hotelling's T-squared and Mahalanobis' D-squared, within-group correlations, data listing.

L4b1  : Ungrouped data
L4bla : Parameter estimation
L4bla14: Normal

**IMSL subprogram library**

BECTR  Tetrachoric correlation coefficient estimation.
BEPAT  Mean and variance inferences using samples from each of two normal populations with unequal variances.
BEPET  Mean and variance inferences using samples from each of two normal populations with equal variances.

**MINITAB interactive system**

TWOSAMPLE Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.
TWOT   Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.

L4b1b  : Distribution-free (nonparametric) analysis

**BMDP program library**

P3S    Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskall-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients.

**DATAPAC subprogram library**

SPCORR Computes the Spearman rank correlation coefficient between the two sets of data in the input vectors X and Y.

**IMSL subprogram library**

NAK1   Kruskal-Wallis test for identical populations.
NBQT   Cochran q test.
NHINC  Includance test.
NKS2   Kolmogorov-Smirnov two-sample test.
NMCC   Calculate and test the significance of the Kendall coefficient of concordance.
NMKN   Kendall's test for correlation (rank correlation coefficient).
NMKSF  Frequency distribution of K and the probability of equalling or exceeding K, where K, the total score from the Kendall rank correlation coefficient calculations, and N, the sample size, are given.
NMKTS  K-sample trends test against ordered alternatives.
NRBHA  Bhapkar v test.
NRWMD  Wilcoxon signed rank test.
NRWRST  Wilcoxon's rank-sum test.
Minitab Interactive System

Kruskal-Wallis Test

Perform Kruskal-Wallis test, based on ranks, of the null hypothesis that there is no difference among K population locations against the alternative of at least one difference. (This is a K-sample generalization of the Mann-Whitney-Wilcoxon test and is a nonparametric alternative to one-way ANOVA.)

Mann-Whitney Test

Performs one- or two-sided two-sample rank test (a.k.a. Wilcoxon rank test) for the difference between two population medians, and calculates the corresponding point and confidence interval estimates.

NAG Subprogram Library

G08AAE Sign test on two paired samples. Double precision version is G08AAF.
G08ABE Wilcoxon matched pairs signed ranks test on two paired samples. Double precision version is G08ABF.
G08ACE Median test on two samples of unequal size. Double precision version is G08ACF.
G08ADE Mann-Whitney U-test on two samples of unequal size. Double precision version is G08ADF.
G08BAE Mood's and David's tests on two samples of unequal size. Double precision version is G08BAF.
G08DAE Kendall's coefficient of concordance. Double precision version is G08DAF.

Minitab Interactive System

Two-Sample Test

Performs a one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.

Grouped Data

One Univariate Qualitative (Proportional) Sample

Simple Random Sampling with Proportion Data - Inferences Regarding the Population Proportion and Total.

Stratified Random Sampling with Proportion Data - Inferences Regarding the Population Proportion and Total.

Two or More Univariate Samples

Biserial and Point-Biserial Correlation Coefficients for a Qualitatively Dichotomized Variable and a Numerically Measurable and Classified Variable.

Biserial Correlation Coefficient for a Qualitatively Dichotomized Variable and a Numerically or Qualitatively Classified Variable.
L4e: One multivariate sample

L4e1: Ungrouped data

L4e1a: Parameter estimation

L4e1a14: Normal

STATLIB subprogram library

COREL: Performs correlation analysis of a multivariate random sample.

CORELS: Performs correlation analysis of a multivariate random sample with computed results returned to the user.

L4e1b: Distribution-free (nonparametric) analysis

L4e2: Ungrouped data with missing values

L4e2a: Parameter estimation

L4e2b: Distribution-free (nonparametric) analysis

L4e3: Grouped data

L4e3a: Parameter estimation

L4e3a14: Normal

IMSL subprogram library

CBNRHO: Estimation of the bivariate normal correlation coefficient using a contingency table.

L4e3b: Distribution-free (nonparametric) analysis

L4e4: Two or more multivariate samples

L4e4a: Parameter estimation
Evaluation of probability density and distribution functions and their inverses comprises a major component of special function evaluation. A discussion of the computational aspects of the evaluation of these functions can be found in the introduction to chapter C (Special Functions).

This chapter first distinguishes between univariate and multivariate probability functions, and then between evaluating distribution and density functions and evaluating percent point functions (the inverses of distribution functions) and sparsity functions (the inverses of density functions). The subchapters are then further broken down by name of distribution (in alphabetical order). No distinction is made in the classification scheme between density and distribution functions because there are at present few enough codes in each class.

The cataloged software provides capabilities for the evaluation of a wide variety of univariate distribution and density functions and their inverses; there is little software classified for multivariate functions.
and 75\% point = 1.

**CAUPDF** Computes the probability density function value for the Cauchy distribution with median \(= 0\) and 75\% point \(= 1\).

**CHSCDF** Computes the cumulative distribution function value for the chi-squared distribution with degrees of freedom parameter \(= \text{NU}\).

### IMSL subprogram library

**MDCH** Chi-squared probability distribution function.

**MDCHN** Non-central chi-squared probability distribution function.

### NAG subprogram library

**G01BCE** Chi-square distribution. Double precision version is G01BCF.

### L\text{5}al\text{d}:

**Double exponential**

### DATAPAC subprogram library

**DEXCDF** Computes the cumulative distribution function value for the double exponential (Laplace) distribution with mean \(= 0\).

**DEXPDF** Computes the probability density function value for the double exponential (Laplace) distribution with mean \(= 0\).

### L\text{5}ale:

**Error function, exponential, extreme value**

### CMLIB subprogram library (FNLIB sublibrary)

**ERF** Error function, \(= (2 / \text{square root of pi}) \times \text{the integral from 0 to x of } e^{-(t^2)}dt\). Double precision version is DERF.

**ERFC** Complementary error function, \(= (2 / \text{square root of pi}) \times \text{the integral from x to infinity of } e^{-(t^2)}dt\). Double precision version is DERFC.

### DATAPAC subprogram library

**EV1CDF** Computes the cumulative distribution function value for the extreme value type 1 distribution.

**EV2CDF** Computes the cumulative distribution function value for the extreme value type 2 distribution with tail length parameter \(= \text{GAMMA}\).

**EXPCDF** Computes the cumulative distribution function value for the exponential distribution with mean \(= 1\) and standard deviation \(= 1\).

**EXPPDF** Computes the probability density function value for the exponential distribution with mean \(= 1\) and standard deviation \(= 1\).

### IMSL subprogram library

**ERF** Evaluate the error function. Note: the Fortran mathematical subroutine libraries may also contain ERF.

**ERFC** Evaluate the complemented error function.

### MATHWARE subprogram library (STEGUN sublibrary)

**ERRINT** Computes error function and complemented error function to maximum machine accuracy. To change computers change one line.

### NAG subprogram library
S15ADE  Complement of error function, erfc(x). Double precision version is S15ADF.
S15AEE  Error function, erf(x). Double precision version is S15AEF.

**L5alf : F distribution**

**DATAPAC subprogram library**

FCDF  Computes the cumulative distribution function value for the F-distribution with degrees of freedom parameters NU1 and NU2.

**IMSL subprogram library**

MDFD  F probability distribution function.
MDFDRE F probability distribution function (integer or fractional degrees of freedom).

**NAG subprogram library**

G01BBE F (variance ratio) distribution. Double precision version is G01BBF.

**L5alg : Gamma, general, geometric**

**DATAPAC subprogram library**

GAMCDF  Computes the cumulative distribution function value for the gamma distribution with tail length parameter = GAMMA.
GEOCDF  Computes the geometric cumulative distribution function value at the value X with parameter = P.

**IMSL subprogram library**

MDGAM  Gamma probability distribution function.
MDGC  General cumulative probability distribution function, given ordinates of the density.
NDEST  Evaluate probability density function at specified points.

**L5ahl : Halfnormal, hypergeometric**

**DATAPAC subprogram library**

HFNCDF  Computes the cumulative distribution function value for the halfnormal distribution with mean = sqrt(2/pi) and standard deviation = 1.

**IMSL subprogram library**

MDHYP  Hypergeometric probability distribution function.

**L5alk : Kolmogorov-Smirnov**

**IMSL subprogram library**

MDSMR  Kolmogorov-Smirnov statistics asymptotic probability distribution function.
<table>
<thead>
<tr>
<th>L5a11 : Lambda, logistic, lognormal</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATAPAC subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LAMCDF</strong> computes the cumulative distribution function value for the (Tukey) lambda distribution with tail length parameter = ALAMBA.</td>
<td></td>
</tr>
<tr>
<td><strong>LAMPDF</strong> computes the probability density function value for the (Tukey) lambda distribution with tail length parameter ALAMBA.</td>
<td></td>
</tr>
<tr>
<td><strong>LGNCDF</strong> computes the cumulative distribution function value for the lognormal distribution with mean sqrt(e).</td>
<td></td>
</tr>
<tr>
<td><strong>LOGCDF</strong> computes the cumulative distribution function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).</td>
<td></td>
</tr>
<tr>
<td><strong>LOGPDF</strong> computes the probability density function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L5a1n : Negative binomial, normal</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATAPAC subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>NBCDF</strong> computes the cumulative distribution function value at X for the negative binomial distribution with parameters P and N.</td>
<td></td>
</tr>
<tr>
<td><strong>NORCDF</strong> computes the cumulative distribution function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.</td>
<td></td>
</tr>
<tr>
<td><strong>NORPDF</strong> computes the probability density function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.</td>
<td></td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MDNOR</strong> normal or Gaussian probability distribution function.</td>
<td></td>
</tr>
<tr>
<td><strong>MSMRAT</strong> ratio of the ordinate to the upper tail area of the standardized normal (Gaussian) distribution.</td>
<td></td>
</tr>
<tr>
<td><strong>NAC subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>S15ABE</strong> cumulative normal distribution function, p(x). Double precision version is S15ABF.</td>
<td></td>
</tr>
<tr>
<td><strong>S15ACE</strong> complement of cumulative normal distribution function, q(x). Double precision version is S15ACF.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L5a1p : Pareto, Poisson</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATAPAC subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>PARCDF</strong> computes the cumulative distribution function value for the Pareto distribution with tail length parameter GAMMA.</td>
<td></td>
</tr>
<tr>
<td><strong>POICDF</strong> computes the cumulative distribution function value at X for the Poisson distribution with tail length parameter ALAMBA.</td>
<td></td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MDTPS</strong> cumulative probability and, optionally, individual terms of the Poisson probability distribution function.</td>
<td></td>
</tr>
<tr>
<td><strong>MINITAB interactive system</strong></td>
<td></td>
</tr>
<tr>
<td><strong>POISSON</strong> prints table of Poisson probabilities and cumulative distribution function.</td>
<td></td>
</tr>
</tbody>
</table>
**t distribution**

**DATAPAC subprogram library**

**TCDF** Computes the cumulative distribution function value for Student's t distribution with degrees of freedom parameter NU.

**IMSL subprogram library**

**MDTD** Students t probability distribution function.

**MDTN** Non-central t probability distribution function.

**MDTNF** Integral related to calculation of noncentral t and bivariate normal probability distribution functions.

**NAG subprogram library**

**G01BAE** Student's t distribution. Double precision version is G01BAF.

**Uniform**

**DATAPAC subprogram library**

**UNICDF** Computes the cumulative distribution function value for the uniform (rectangular) distribution on the unit interval (0,1).

**UNIPDF** Computes the probability density function value for the uniform (rectangular) distribution on the unit interval (0,1).

**Weibull**

**DATAPAC subprogram library**

**WEICDF** Computes the cumulative distribution function value for the Weibull distribution with tail length parameter GAMMA.

**Inverse cumulative distribution functions, sparsity functions**

**Beta, binomial**

**DATAPAC subprogram library**

**BINPPF** Computes the percent point function value at P for the binomial distribution with parameters PPAR and N.

**IMSL subprogram library**

**MDBETI** Inverse beta probability distribution function.

**NAG subprogram library**

**G01CDE** Inverse beta distribution of first kind. Double precision version is G01CDF.
### L5a2c: Cauchy, chi-squared

**DATAPAC subprogram library**

- **CAUPPF**: Computes the percent point function value for the Cauchy distribution with median \( = 0 \) and 75% point \( = 1 \).
- **CAUSF**: Computes the sparsity function value for the Cauchy distribution with median \( = 0 \) and 75% point \( = 1 \).
- **CHSPPF**: Computes the percent point function value for the chi-squared distribution with integer degrees of freedom parameter \( = \text{NU} \).

**IMSL subprogram library**

- **MDCHI**: Inverse chi-squared probability distribution function.

**NAG subprogram library**

- **G01CCE**: Inverse chi-square distribution. Double precision version is G01CCF.

### L5a2d: Double exponential

**DATAPAC subprogram library**

- **DEXPPF**: Computes the percent point function value for the double exponential (Laplace) distribution with mean \( = 0 \).
- **DEXSF**: Computes the sparsity function value for the double exponential (Laplace) distribution with mean \( = 0 \) and standard deviation \( = \sqrt{2} \).

### L5a2e: Exponential, extreme value

**DATAPAC subprogram library**

- **EV1PPF**: Computes the percent point function value for the extreme value type 1 distribution with mean \( = \text{Euler's number} = 0.57721566 \).
- **EV2PPF**: Computes the percent point function value for the extreme value type 2 distribution with tail length parameter \( = \text{GAMMA} \).
- **EXPPPF**: Computes the percent point function value for the exponential distribution with mean \( = 1 \) and standard deviation \( = 1 \).
- **EXPSF**: Computes the sparsity function value for the exponential distribution with mean \( = 1 \) and standard deviation \( = 1 \).

**IMSL subprogram library**

- **MERFCI**: Inverse complemented error function.
- **MERFI**: Inverse error function.

### L5a2f: F distribution

**IMSL subprogram library**

- **MDFI**: Inverse F probability distribution function.

**NAG subprogram library**
G01CBE  Inverse F (variance ratio) distribution. Double precision version is G01CBF.

**L5a2g : Gamma, general, geometric**

**DATAPAC subprogram library**

GAMPPF  Computes the percent point function value for the gamma distribution with mean = GAMMA and standard deviation = sqrt(GAMMA).

GEOPPF  Computes the percent point function value for the geometric distribution with parameter PPAR.

**IMSL subprogram library**

MDGCI  Inverse of a general cumulative probability distribution function, given ordinates of the density.

**L5a2h : Halfnormal**

**DATAPAC subprogram library**

HFNPPF  Computes the percent point function value for the halfnormal distribution with mean = sqrt(2/pi) and standard deviation = 1.

**L5a2l : Lambda, logistic, lognormal**

**DATAPAC subprogram library**

LAMPPF  Computes the percent point function value for the (Tukey) lambda distribution with tail length parameter ALAMBA.

LAMSF  Computes the sparsity function value for the (Tukey) lambda distribution with tail length parameter ALAMBA.

LGNPPF  Computes the percent point function value for the lognormal distribution with mean = sqrt(e).

LOGPPF  Computes the percent point function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).

LOGSF  Computes the sparsity function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).

**L5a2n : Negative binomial, normal, normal scores**

**DATAPAC subprogram library**

NBPPF  Computes the percent point function value at P for the negative binomial distribution with parameters PPAR and N.

NORPPF  Computes the percent point function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.

NORSF  Computes the sparsity function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1.

**IMSL subprogram library**

MDNRIS  Inverse standard normal (Gaussian) probability distribution function.

MSEN0  Expected values of normal order statistics.
**GAMS: Modules by Class**

**NAG subprogram library**

**G01CEE** Inverse normal distribution. Double precision version is G01CEF.

**G01DAE** Calculation of normal scores. Double precision version is G01DAF.

---

**Datapac subprogram library**

**L5a2p** Pareto, Poisson

**PARPPF** Computes the percent point function value for the Pareto distribution with tail length parameter GAMMA.

**POIPPF** Computes the percent point function value at P for the Poisson distribution with mean = ALAMBA and standard deviation = sqrt(ALAMBA).

---

**L5a2t** t distribution

**Datapac subprogram library**

**TPPF** Computes the percent point function value for the Student’s t distribution with degrees of freedom parameter NU.

---

**IMSL subprogram library**

**MDSTI** Inverse of a modification of Student’s t probability distribution function.

---

**NAG subprogram library**

**G01CAE** Inverse Student’s t distribution. Double precision version is G01CAF.

---

**L5a2u** Uniform

**Datapac subprogram library**

**UNIPPF** Computes the percent point function value for the uniform (rectangular) distribution on the unit interval (0,1).

**UNISF** Computes the sparsity function value for the uniform (rectangular) distribution on the unit interval (0,1).

---

**L5a2w** Weibull

**Datapac subprogram library**

**WEIPPF** Computes the percent point function value for the Weibull distribution with tail length parameter GAMMA.

---

**L5b** Multivariate

---

**L5b1** Cumulative distribution functions, probability density functions

---
L6: Random numbers

L6b1n : Normal

IMSL subprogram library

MDBNOR  Bivariate normal probability distribution function.
MDTNF   Integral related to calculation of noncentral t and bivariate normal probability distribution functions.

L6: Pseudo-Random Number Generation

The generation of pseudo-random numbers according to certain statistical distributions is an area of statistical computing which has important applications in the design of experiments, the development and analysis of statistical methods, the analysis of algorithms, and simulation studies.

Generation of uniform pseudo-random numbers is central, primarily because pseudo-random numbers from other distributions can often be obtained by transformation of uniform numbers. Efficient methods have been developed for such commonly used distributions as the normal and its relatives, while special methods have been developed for some less common distributions; approximation methods are also used.

Quality and portability are important considerations when selecting a pseudo-random number generator. Some generators have been judged much better than others at simulating the desired distribution in a given computer/compiler environment. Some generators produce acceptable results in each of several environments, while others produce exactly the same results in different environments. While the latter may be desirable when simulation studies are to be performed in several environments, a price is paid in execution time.

The software cataloged here provides capabilities for a wide variety of univariate distributions (classified in alphabetical order), and several multivariate distributions.

January 1984

References


L6 : Pseudo-random number generation

L6a : Univariate

L6a2 : Beta, binomial, Boolean

DATAPAC subprogram library

BETRAN  Generates a random sample of size N from the beta distribution with parameters ALPHA and BETA.
BINRAN  Generates a random sample of size N from the binomial distribution with parameters P and NPAR.

IMSL subprogram library

GGBN   Binomial random deviate generator.
GGBTR  Beta random deviate generator.

MINITAB interactive system

BRANDOM Generates K pseudo-random numbers from binomial distribution (number of successes in n Bernoulli trials with probability p of success).
BTRIALS Generates pseudo-random sequence of K 0's and 1's, with the probability p of a 1.

NAG subprogram library

G05DLE Pseudo-random real numbers, Beta distribution of the first kind. Double precision version is G05DLF.
G05DME  Pseudo-random real numbers, Beta distribution of the second kind. Double precision version is G05DMF.
G05DZE  Pseudo-random logical value. Double precision version is G05DZF.
G05EDE  Set up reference vector for generating pseudo-random integers, binomial distribution. Double precision version is G05EDF.

**L6a3 :** Cauchy, chi-squared

**DATAPAC subprogram library**

**CAURAN** Generates a random sample of size N from the Cauchy distribution with median = 0 and 75% point = 1.

**CHSRAN** Generates a random sample of size N from the chi-squared distribution with integer degrees of freedom parameter = NU.

**IMSL subprogram library**

**GGCAY** Cauchy random deviate generator.

**GGCHS** Chi-squared random deviate generator.

**NAG subprogram library**

**G05DFE** Pseudo-random real numbers, Cauchy distribution. Double precision version is G05DFF.

**G05DHE** Pseudo-random real numbers, chi-square distribution. Double precision version is G05DHF.

**L6a4 :** Discrete, double exponential

**DATAPAC subprogram library**

**DEXRAN** Generates a random sample of size N from the double exponential (Laplace) distribution with mean = 0 and standard deviation = sqrt(2).

**MINITAB interactive system**

**DRANDOM** Generates K pseudo-random numbers from a user-specified discrete distribution.

**L6a5 :** Exponential, extreme value

**DATAPAC subprogram library**

**EV1RAN** Generates a random sample of size N from the extreme value type 1 distribution with mean = Euler's number = 0.57721566.

**EV2RAN** Generates a random sample of size N from the extreme value type 2 distribution with tail length parameter = GAMMA.

**EXPRAN** Generates a random sample of size N from the exponential distribution with mean = 1 and standard deviation = 1.

**IMSL subprogram library**

**GGEXN** Exponential random deviate generator.

**GGEXT** Random deviate generator for a mixture of two exponentials.

**NAG subprogram library**
<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>G05DBE</strong></td>
<td>Pseudo-random real numbers, exponential distribution. Double precision version is G05DBF.</td>
</tr>
<tr>
<td><strong>L6a6</strong></td>
<td>F distribution</td>
</tr>
<tr>
<td><strong>FRAN</strong></td>
<td>Generates a random sample of size N from the F-distribution with degrees of freedom parameters ( \nu_1 ) and ( \nu_2 ).</td>
</tr>
<tr>
<td><strong>G05DKE</strong></td>
<td>Pseudo-random real numbers, Snedecor’s F-distribution. Double precision version is G05DKF.</td>
</tr>
<tr>
<td><strong>L6a7</strong></td>
<td>Gamma, general (continuous, discrete) distributions, geometric</td>
</tr>
<tr>
<td><strong>GAMRAN</strong></td>
<td>Generates a random sample of size N from the gamma distribution with tail length parameter ( \text{Gamma} ), mean ( \text{Gamma} ) and standard deviation ( \text{sqrt(Gamma)} ).</td>
</tr>
<tr>
<td><strong>GEORAN</strong></td>
<td>Generates a random sample of size N from the geometric distribution with parameter P.</td>
</tr>
<tr>
<td><strong>GGAMR</strong></td>
<td>One parameter gamma random deviate generator, and usable as basis for 2 parameter gamma, exponential, chi-squared, chi, beta, t and F deviate generator.</td>
</tr>
<tr>
<td><strong>GGDA</strong></td>
<td>General discrete distribution random deviate generator using alias method.</td>
</tr>
<tr>
<td><strong>GGDT</strong></td>
<td>General discrete distribution random deviate generator using table lookup method.</td>
</tr>
<tr>
<td><strong>GGEOT</strong></td>
<td>Geometric random deviate generator.</td>
</tr>
<tr>
<td><strong>GGVCR</strong></td>
<td>General continuous distribution random deviate generator.</td>
</tr>
<tr>
<td><strong>G05DGE</strong></td>
<td>Pseudo-random real numbers, gamma distribution with parameters (g,h). Double precision version is G05DGF.</td>
</tr>
<tr>
<td><strong>G05EXE</strong></td>
<td>Set up reference vector from supplied cumulative distribution function or probability distribution function. Double precision version is G05EXF.</td>
</tr>
<tr>
<td><strong>L6a8</strong></td>
<td>Halfnormal, hypergeometric</td>
</tr>
<tr>
<td><strong>HFNran</strong></td>
<td>Generates a random sample of size N from the halfnormal distribution with mean ( \text{sqrt(2/pi)} ) and standard deviation ( 1 ).</td>
</tr>
<tr>
<td><strong>GGHPR</strong></td>
<td>Hypergeometric random deviate generator.</td>
</tr>
<tr>
<td><strong>G05EFE</strong></td>
<td>Set up reference vector for generating pseudo-random integers, hypergeometric distribution. Double precision version is G05EFF.</td>
</tr>
</tbody>
</table>
### L6a9: Integers

**MINITAB interactive system**

**IRANDOM** Generates $K$ pseudo-random integers in a specified interval.

**NAG subprogram library**

**G05DYE** Pseudo-random integer from uniform distribution. Double precision version is G05DYF.

**G05EBE** Set up reference vector for generating pseudo-random integers, uniform distribution. Double precision version is G05EBF.

**G05EYE** Pseudo-random integer from reference vector. Double precision version is G05EYF.

### L6a12: Lambda, logical, logistic, lognormal

**DATAPAC subprogram library**

**LAMRAN** Generates a random sample of size $N$ from the (Tukey) lambda distribution with tail length parameter ALAMBA.

**LGNRAN** Generates a random sample of size $N$ from the lognormal distribution with mean $= \sqrt{e}$.

**LOGRAN** Generates a random sample of size $N$ from the logistic distribution with mean $= 0$ and standard deviation $= \pi/\sqrt{3}$.

**IMSL subprogram library**

**GGNLG** Log-normal random deviate generator.

**NAG subprogram library**

**G05DCE** Pseudo-random real numbers, logistic distribution. Double precision version is G05DCF.

**G05DEE** Pseudo-random real numbers, lognormal distribution. Double precision version is G05DEF.

**G05DZE** Pseudo-random logical value. Double precision version is G05DZF.

### L6a14: Negative binomial, normal

**CMLIB subprogram library (FNLIB sublibrary)**

**RGAUSS** Normal random number.

**CMLIB subprogram library (RV sublibrary)**

**RNOR** Generates quasi normal random numbers with zero mean and unit standard deviation.

**DATAPAC subprogram library**

**NBRAN** Generates a random sample of size $N$ from the negative binomial distribution with parameters $P$ and NPAR.

**NORRAN** Generates a random sample of size $N$ from the normal (Gaussian) distribution with mean $= 0$ and standard deviation $= 1$.

**IMSL subprogram library**

**GBBNR** Negative binomial random deviate generator.

**GGNML** Normal or Gaussian random deviate generator.

**GGNO** Generate set of order statistics from normal distribution.
Normal random deviate generator via the polar method.

Normal random deviate generator. Function form of GGNML.

MINITAB interactive system

NRANDOM Generates K pseudo-random numbers from the normal distribution with specified mean and standard deviation.

NAG subprogram library

G05DDE Pseudo-random real numbers, normal distribution (a,b). Double precision version is G05DDF.
G05EEE Set up reference vector for generating pseudo-random integers, negative binomial distribution. Double precision version is G05EEF.

STATLIB subprogram library

NRAND Generates a vector of normally distributed pseudo-random numbers.

L6a15: Order statistics

IMSL subprogram library

GGNO Generate set of order statistics from normal distribution.
GGUO Generate set of order statistics from uniform (0,1) distribution.

L6a16: Pareto, permutations, Poisson

DATAPAC subprogram library

PARRAN Generates a random sample of size N from the Pareto distribution with tail length parameter GAMMA.
POIRAN Generates a random sample of size N from the Poisson distribution with mean = ALAMBA and standard deviation = sqrt(ALAMBA).
RANPER Generates a random permutation of size N of the values 1.0, 2.0, 3.0, ..., N-1, N.

IMSL subprogram library

GGNPP Nonhomogeneous Poisson process generator with rate function lambda(t) - fixed interval, fixed number, or one at a time.
GGPER Generate a random permutation of the integers 1 to k.
GGPON Poisson random deviate generator where the Poisson parameter changes frequently.
GGPOS Poisson random deviate generator where the Poisson parameter does not change often.

MINITAB interactive system

PRANDOM Generates K pseudo-random numbers from the Poisson distribution with specified population mean K.

NAG subprogram library

G05ECE Set up reference vector for generating pseudo-random integers, Poisson distribution. Double precision version is G05ECF.
G05EHE Performs a pseudo-random permutation of a vector of integers. Double precision version is G05EHF.
**L0a19 :** Samples, stable distribution

**IMSL subprogram library**

**GGSRS** Generate a simple random sample from a finite population.

**GGSTA** Stable distribution random deviate generator.

**MINITAB interactive system**

**SAMPLE** Randomly selects without replacement values from one or more vectors, optionally carrying along other vectors.

**NAG subprogram library**

**G05EJE** Selects a pseudo-random sample from an integer vector. Double precision version is G05EJF.

---

**L0a20 :** t distribution, time series, triangular

**DATAPAC subprogram library**

**TRAN** Generates a random sample of size N from the Student's t distribution with degrees of freedom parameter NU.

**IMSL subprogram library**

**FTGEN** Generation of a time series from a given ARIMA (Box-Jenkins) stochastic model.

**GGTRA** Triangular distribution random deviate generator.

**NAG subprogram library**

**G05DJE** Pseudo-random real numbers, Student's t distribution. Double precision version is G05DJF.

**G05EGE** Set up reference vector for univariate ARMA time series model. Double precision version is G05EGF.

**G05EWE** Generate next term from ARMA time series using vector from G05EGE. Double precision version is G05EWF.

---

**L0a21 :** Uniform

**CMLIB subprogram library (PNLIB sublibrary)**

**RAND** Uniform random number on [0,1].

**RUNIF** Sequence of uniform random numbers on [0,1].

**CMLIB subprogram library (RV sublibrary)**

**UNI** Generates uniformly distributed random numbers on the interval [0,1). UNI's main advantages are a long cycle and a high degree of reproducibility on other machines (it runs on any machine with at least 16 bit integer arithmetic).

**DATAPAC subprogram library**

**UNIRAN** Generates a random sample of size N from the uniform (rectangular) distribution on the unit interval (0,1).

**IMSL subprogram library**

**GGUBFS** Basic uniform (0,1) random number generator. Function form of GGUBS.

**GGUBS** Basic uniform (0,1) pseudo-random number generator.

**GGUBT** Uniform (0,1) pseudo-random number generator using alternate multiplier.
GGUD  Discrete uniform random number generator.
GGUO  Generate set of order statistics from uniform (0,1) distribution.
GGUW  Uniform (0,1) random number generator with shuffling.

MINITAB interactive system
URANDOM Generates K pseudo-random numbers from the uniform (0,1) distribution.

NAG subprogram library
G05CAE  Pseudo-random real numbers, uniform distribution over (0,0,1.0). Double precision version is G05CAF.
G05DAE  Pseudo-random real numbers, uniform distribution over (a,b). Double precision version is G05DAF.
G05DYE  Pseudo-random integer from uniform distribution. Double precision version is G05DYF.
G05EBE  Set up reference vector for generating pseudo-random integers, uniform distribution. Double precision version is G05EBF.

PORT subprogram library
RANBYT  Returns the real random variate generated by UNI, together with its bit pattern presented in four 8-bit bytes.
UNI    Returns a single real random variate from the uniform [0,1) distribution.

IMSL subprogram library
GGVMS  Von Mises random deviate generator.

DATAPAC subprogram library
WEIRAN  Generates a random sample of size N from the Weibull distribution with tail length parameter GAMMA.

IMSL subprogram library
GGWIB  Weibull random deviate generator.

NAG subprogram library
G05DPE  Pseudo-random real numbers, Weibull distribution. Double precision version is G05DPF.

IMSL subprogram library
ZSRCH  Generate points in an n dimensional space.

IMSL subprogram library
GGCOR  Generate a random orthogonal matrix and a random correlation matrix.

GGTAB  Generate a random contingency table with given row and column totals.

**L6b13 :** Multinomial

- **IMSL subprogram library**
  - GGMTN  Multinomial random deviate generator.

**L6b14 :** Normal

- **IMSL subprogram library**
  - GGNSM  Multivariate normal random deviate generator with given covariance matrix.

- **NAG subprogram library**
  - G05EZE  Returns a pseudo-random multivariate normal vector taken from a distribution described by a reference vector set up by G05EAEB. Double precision version is G05EZF.

**L6b15 :** Orthogonal matrix

- **IMSL subprogram library**
  - GGCOR  Generate a random orthogonal matrix and a random correlation matrix.

**L6b21 :** Uniform

- **IMSL subprogram library**
  - GGSPH  Generation of uniform random deviates from the surface of the unit sphere in 3 or 4 space.

**L6c :** Service routines (e.g., seed)

- **NAG subprogram library**
  - G05CBE  Initialise random number generating routines, to give a repeatable sequence. Double precision version is G05CBF.
  - G05CCE  Initialise random number generating routines, to give nonrepeatable sequence. Double precision version is G05CCF.
  - G05CFE  Save state of random number generating routines. Double precision version is G05CFF.
  - G05CGE  Restore state of random number generating routines. Double precision version is G05CGF.

- **PORT subprogram library**
  - RANSET  Initializes the uniform random number generator, UNI, to other than the default initial values.

**L7: Experimental Design, Including Analysis of Variance**
Computer implementation of analyses of experimental designs is straightforward, and the output is commonly clear, when the number of observations is constant across all treatment level combinations. Care should be exercised when the number of observations varies across cells; some output values may be computed differently than expected, and, for the analysis, ordinary interpretation of classical tests may not apply.

Most statistical analyses of experimental designs assume independent, normally distributed errors with zero mean and constant variance. The appropriateness of assumptions such as these should be evaluated through statistical and graphical analysis. Transformation of the data may make assumptions more reasonable. Less restrictive assumptions are needed for nonparametric (distribution-free) analyses.

Software in this chapter is first distinguished by data type - univariate and multivariate (though there are currently no multivariate codes classified). The next level distinguishes among forms of experimental designs. Software is classified both at the leaves of the tree and at higher levels in this chapter; the latter performs more than one of the tasks in the subchapters.

MATLAB, an interactive system, has analysis of variance capabilities (see the Library Reference).

References
### L7a1a : Parametric analysis

**BMDP program library**

**PIV**
Performs one-way ANOVA or ANCOVA with standard results. For ANCOVA, tests 1) equality of slopes, 2) zero slope, and 3) equality of adjusted cell means; plots the covariate for each group. Tests equality of pairs of means (or adjusted means). Options: linear contrasts, within-group correlations and statistics.

**IMSL subprogram library**

**ACRDAN**
Analysis of one-way classification design data.

**MINITAB interactive system**

**AOVONEWAY**
Performs a one-way analysis of variance and prints standard results.

**ONEWAY**
Performs one-way analysis of variance, prints standard results, and optionally saves results.

**STATLIB subprogram library**

**G04AEE**
One-way analysis of variance, subgroups of unequal size. Double precision version is G04AEF.

**ONEWY**
Performs one-way analysis of variance for two or more random samples.

**ONEWYS**
Performs one-way analysis of variance for two or more random samples, with computed results returned to the user.

### L7a1a1 : Contrasts, multiple comparisons

**IMSL subprogram library**

**ACTRST**
Contrast estimates and sums of squares.

**ASNKMC**
Student-Newman-Keuls multiple comparison test.

### L7a1a2 : Analysis of variance components

**IMSL subprogram library**

**AGVACL**
One or two-sided interval estimate of a variance component.

### L7a1b : Distribution-free (nonparametric) analysis

**IMSL subprogram library**

**NAWRPE**
Wilsons ANOVA (1, 2, or 3 way designs) with equal replication.

**NAWRPU**
Wilsons ANOVA (1, 2, or 3 way designs) with unequal replication.

**STATLIB subprogram library**

**G08AFE**
Kruskal-Wallis 1-way analysis of variance on k samples of unequal size. Double precision version is G08AFF.

### L7a2 : Balanced multiway design
### BMDP program library

**P8V**  
ANOVA for complete designs with equal cell sizes — nested, crossed, partially nested, partially crossed designs for fixed-effect models, mixed models (including repeated measures), and random-effect models, with parameter estimation and printing.

**P9D**  
Provides descriptive statistics (means, std. devs., frequencies, one-way ANOVA table) of groups (cells) for data classified into cells using one or more grouping variables. Options: miniplots of cell means (eight per page), plot frames are defined by combinations of levels of grouping variables.

### IMLSL subprogram library

**AGBACP**  
Analysis of balanced complete experimental design structure data.

**AGXPM**  
Expected mean squares for balanced complete design models.

**AORDR**  
Reordering of the data obtained from any balanced complete experimental design.

### BMDP program library

**P7D**  
Side-by-side histograms for each cell in one-way or two-way ANOVA, within-group summary statistics and ANOVA table (with equality of variance test and tests that do not assume equal variances). Options: trimmed mean analysis, ANOVA diagnostics, tests of pairwise mean comparisons, correlations, Winsorized means.

### IMLSL subprogram library

**ARCBAN**  
Analysis of two-way classification design data.

### MINITAB interactive system

**TWOWAYAOV**  
Performs two-way analysis of variance for balanced data (equal number of observations, one or more, in each cell) and prints standard results. Options: fit additive model, save results.

### NAG subprogram library

**G04AFE**  
Two-way analysis of variance, cross-classification, subgroups of equal size. Double precision version is G04AFF.

### IMLSL subprogram library

**AFACN**  
Full factorial plan analysis.

**AFACT**  
Full factorial plan analysis - easy to use version.
ANESTE  Analysis of completely nested design data with equal numbers in the subclasses.
ANESTU  Analysis of completely nested design data with unequal numbers in the subclasses.

**L7a2a2 : Distribution-free (nonparametric) analysis**

BMDP program library
P3S  Computes and prints results from one or more of the following: sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskall-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients.

IMSL subprogram library
NAFRE  Friedman's test for randomized complete block designs.
NAWNRP  Wilsons ANOVA (2 or 3 way designs) without replicates.
NAWRPE  Wilsons ANOVA (1, 2, or 3 way designs) with equal replication.

NAG subprogram library
G08AEE  Friedman 2-way analysis of variance on k matched samples. Double precision version is G08AEF.

**L7a2b : Incomplete**

**L7a2b1 : Parametric analysis**

IMSL subprogram library
ABIBN  Analysis of balanced incomplete block and balanced lattice designs.

**L7a2b1a : Latin square**

IMSL subprogram library
ALSQAN  Analysis of Latin square design data.

NAG subprogram library
G04ADE  Three-way analysis of variance, Latin square design. Double precision version is G04ADF.

**L7a2b1b : Lattice designs**

**L7a2b2 : Distribution-free (nonparametric) analysis**

**L7a3 : Analysis of covariance**

BMDP program library
P1V  Performs one-way ANOVA or ANCOVA with standard results. For ANCOVA, tests 1) equality of slopes,
2) zero slope, and 3) equality of adjusted cell means; plots the covariate for each group. Tests equality of pairs of means (or adjusted means). Options: linear contrasts, within-group correlations and statistics.

**IMSL subprogram library**

**ANCOV1** Covariance analysis for one-way classification design data.

**BMDP program library**

**P3V** Uses maximum likelihood (ML) and restricted ML approaches to balanced and unbalanced fixed and random coefficient models of quite arbitrary form (including having covariates), with parameter estimation, hypothesis testing, and printing. Weights optional.

**L7a4:** General linear model (unbalanced design)

**AGLMD** General linear model analysis.

**AMEANS** Preparation of a set of unbalanced data for analysis by the method of unweighted means.

**NAG subprogram library**

**G04AGE** Two-way analysis of variance, hierarchial classification, subgroups of unequal size. Double precision version is G04AGF.

**L7a4a:** Parametric analysis

**IMSL subprogram library**

**AGLMD** General linear model analysis.

**AMEANS** Preparation of a set of unbalanced data for analysis by the method of unweighted means.

**NAG subprogram library**

**G04AGE** Two-way analysis of variance, hierarchial classification, subgroups of unequal size. Double precision version is G04AGF.

**L7a4b:** Distribution-free (nonparametric) analysis

**NAWRPU** Wilsons ANOVA (1, 2, or 3 way designs) with unequal replication.

**L7b:** Multivariate

**L8: Regression**

Regression analysis is concerned with estimating the relationship between one “dependent” variable and one or more “independent” variables. The quality of the estimate is commonly judged by evaluating a function of the residuals (the differences between the observations of the dependent variable and their estimates). Least squares regression provides the best linear unbiased estimates of the regression parameters when the errors are independent random variables with zero mean and constant variance. Techniques such as $L_1$, $L_\infty$, robust, or EDA (exploratory data analysis) regression may be more appropriate when these distributional assumptions are not valid; other alternatives are appropriate when the dependent variables are also subject to error, when the errors are autocorrelated, and so on.

Numerical problems arise in multiple regression when the design matrix is ill-conditioned; orthogonal polynomials provide one solution to this problem in the case of polynomial regression, while elimination of independent variables and ridge regression provide possible solutions in the general case. Variable selection techniques are useful in selecting one or more “best” regression equations given a number of candidate independent variables.

Regression software comes in the form of subroutines (IMSL and NAG have both building blocks and full regression subroutines. STATLIB subroutines provide comprehensive printing of output, CMLIB packages provides several special-purpose regression capabilities), programs (nine in BMDP, two in INVAR), and interactive system commands (four in
Minitab).

The software in this chapter is first organized by the type of regression (e.g., linear least squares regression (L8a), EDA regression (L8f), nonlinear regression (L8g)). The bulk of the software is in chapter L8a, and is distinguished either by the nature of the independent variable data or by the purpose (e.g., simple (L8a1), multiple (L8a4), regression diagnostics (L8a9)). In addition to doing regression, much of the software in these sections also provides hypothesis testing, inference, and analysis of residuals, so that chapters L8a9 and L8a10 (for hypothesis testing and inference) contain software that provides only diagnostics or testing and inference.

Future editions of GAMS will include software for robust regression and for total least squares (errors in variables) regression, and addition nonlinear regression codes.

References


<table>
<thead>
<tr>
<th>L8</th>
<th>Regression (search also classes G, K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L8a</td>
<td>Linear least squares (L-2) (search also classes D5, D6, D9)</td>
</tr>
</tbody>
</table>

**MINITAB interactive system**

**REGRESS** Performs simple or multiple linear regression, prints standard results. Options: amount of output, save results, weights, handle missing values, through the origin, compute and save regression diagnostics, lack of fit tests.

<table>
<thead>
<tr>
<th>L8a1</th>
<th>Simple</th>
</tr>
</thead>
<tbody>
<tr>
<td>L8a1a</td>
<td>Ordinary</td>
</tr>
<tr>
<td>L8a1al</td>
<td>Unweighted</td>
</tr>
<tr>
<td>L8a1ala</td>
<td>No missing values</td>
</tr>
</tbody>
</table>

**IMSL subprogram library**

**RLONE** Analysis of a simple linear regression model.

**NAG subprogram library**

**G02CAE** Simple linear regression with constant term, no missing values. Double precision version is G02CAF.

| L8a1alb | Missing values |

**IMSL subprogram library**
BEMIRI  Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (in-core version).

BEMIRO  Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (out-of-core version).

**NAG subprogram library**

G02CCE  Simple linear regression with constant term, missing values. Double precision version is G02CCF.

<table>
<thead>
<tr>
<th>L8a1a2 :</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>L8a1b :</td>
<td>Through the origin</td>
</tr>
<tr>
<td>L8a1c :</td>
<td>Errors in variables</td>
</tr>
<tr>
<td>L8a1d :</td>
<td>Calibration (inverse regression)</td>
</tr>
<tr>
<td>L8a2 :</td>
<td>Polynomial</td>
</tr>
<tr>
<td>L8a2a :</td>
<td>Not using orthogonal polynomials</td>
</tr>
<tr>
<td>L8a2a1 :</td>
<td>Unweighted</td>
</tr>
<tr>
<td>L8a2a2 :</td>
<td>Weighted</td>
</tr>
</tbody>
</table>

**NAG subprogram library**

G02CBE  Simple linear regression without constant term, no missing values. Double precision version is G02CBF.

G02CDE  Simple linear regression without constant term, missing values. Double precision version is G02CDF.

**CMLIB subprogram library (SLRPACK sublibrary)**

RGM  Computes estimates of simple linear regression parameters for a geometric mean regression.

RYORK  Estimates simple linear regression coefficients when both variables are subject to errors which are not necessarily homogeneous in variance.

**IMSL subprogram library**

RLINCF  Response control using a fitted simple linear regression model.

**DATAPAC subprogram library**

POLY  Computes a least squares polynomial fit (of degree = IDEG) of the response variable data in the vector Y as a function of vector X and with optional weights.
L8a2b: Using orthogonal polynomials

BMDP program library

P5R Least squares fit of a polynomial in one independent variable to the dependent variable. Prints standard results and goodness-of-fit statistics for each polynomial degree. Computations use orthogonal polynomials. Options: weights, additional printing, and three plots.

IMSL subprogram library

• RLFOR Fit a univariate curvilinear regression model using orthogonal polynomials with optional weighting and prediction analysis - easy-to-use version.

RLPOL Generate orthogonal polynomials with the associated constants AA and BB.

L8a2b1: Unweighted

IMSL subprogram library

RLDCVA Variance estimates for decoded orthogonal polynomial regression coefficients.

RLFOTH Fit a univariate curvilinear regression model using orthogonal polynomials.

STATLIB subprogram library

PFIT Performs linear least squares regression analysis of a polynomial model.

PFITS Performs linear least squares regression analysis of a polynomial model, with computed results returned to the user.

L8a2b2: Weighted

IMSL subprogram library

RLDCVA Variance estimates for decoded orthogonal polynomial regression coefficients.

RLFOTH Fit a univariate curvilinear regression model using orthogonal polynomials with weighting.

STATLIB subprogram library

PFITW Performs weighted linear least squares regression analysis of a polynomial model.

PFITWS Performs weighted linear least squares regression analysis of a polynomial model with computed results returned to the user.

L8a3: Piecewise polynomial (i.e. multiphase or spline)

CMLIB subprogram library (FC sublibrary)

FC Fits piecewise polynomial to discrete data with equality and inequality constraints.

L8a4: Multiple

BMDP program library

P1R Performs multiple linear regression and prints standard results. Options: weights, form of input, regression on subsets or groups and test of equality of regression lines, intercept term present or absent,
more printing, five plots, save predicted values and residuals.

**P2R**

Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving.

**L8a4a** : Ordinary

**L8a4a1** : Unweighted

**L8a4a1a** : No missing values

**STATLIB subprogram library**

**FIT**

Performs linear least squares regression analysis of a general linear model.

**FITS**

Performs linear least squares regression analysis of a general linear model with computed results returned to the user.

**L8a4a1b** : Missing values

**L8a4a1c** : From correlation data

**IMSL subprogram library**

**RLMUL**

Multiple linear regression analysis.

**NAG subprogram library**

**G02CGE**

Performs a multiple linear regression on the set of variables whose means, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients are given. Double precision version is G02CGF.

**G02CHE**

Performs a multiple linear regression with no constant on the set of variables whose sums of squares and cross-products about zero and correlation-like coefficients are given. Double precision version is G02CHF.

**L8a4a1d** : Using principal components

**BMDP program library**

**P4R**

Regression analysis for a dependent variable on a set of principal components computed from the independent variables in a stepwise manner determined either by magnitude of eigenvalue or correlations between dependent variable and components, with printing. Options: form of input, more printing, four plots.

**L8a4a1e** : Using preference pairs

**BMDP program library**

**P9M**

Scoring based on preference pairs - for each observation construct score as linear combination of
variables with coefficients based on expert preference, in stepwise manner. Options: printing, plots, compare results when analysis is repeated for different judges.

STATLIB subprogram library

FITW  Performs weighted linear least squares regression analysis of a general linear model.
FITWS Performs weighted linear least squares regression analysis of a general linear model with computed results returned to the user.

BMDP program library

PLR  Stepwise logistic regression for binary dependent variable and categorical (design variables are formed) and continuous independent variables, using either maximum likelihood or approximate asymptotic estimates for stepping. Three options for generating design variables, plots, SYSTEM WARNING - MAX PAGES interactive stepping.

BMDP program library

P2R  Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving.

P9R  Estimates regression equations for "best" (by R-squared, adjusted R-squared, or Mallows' C(p) criterion) subset of predictor variables by Furnival-Wilson algorithm. Options: weights, form of input, Durbin-Watson statistic. Cook's distance and several types of residuals may be printed, plotted, or saved.

IMSL subprogram library

RLEAP Leaps and bounds algorithm for determining a number of best regression subsets from a full regression model. USLEAP is a special purpose output routine designed to be used only in conjunction with RLEAP.

RLSEP Selection of a regression model using a forward stepwise algorithm, and computation of the usual analysis of variance table entries - easy-to-use version.

RLSTP Regression model selection using a forward stepwise algorithm with results available after each step.

MINITAB interactive system

STEPWISE Performs stepwise linear regression using forward selection, backward elimination, conventional stepwise, or user intervention. Options available through subcommands: F-to-enter and F-to-remove, force and remove sets of variables, print next "best" (by the F-statistic) K alternatives.
Regression design

**IMSL subprogram library**

**RLCOMP**  Generation of an orthogonal central composite design.

**L8a7 :** Several multiple regressions

**BMDP program library**

**P1R** Performs multiple linear regression and prints standard results. Options: weights, form of input, regression on subsets or groups and test of equality of regression lines, intercept term present or absent, more printing, five plots, save predicted values and residuals.

**P8R** Computes the partial correlations of a set of variables after removing the linear effects of a second set of variables. Can be used for regression, especially if multiple dependent variables are present. Prints standard results. Options: weights, form of input, additional printing and plots.

**NAG subprogram library**

**G02CJE** Performs one or more multiple linear regressions, regressing each of a set of dependent variables separately on the same set of independent variables. Input is raw data; output includes, for each dependent variable, estimates of regression coefficients and an estimate of the variance of residuals. Double precision version is G02CJF.

**L8a8 :** Multivariate

**IMSL subprogram library**

**OFIMA3** Least squares solution to the matrix equation $AT = B$.

**L8a9 :** Diagnostics

**BMDP program library**

**P2R** Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving.

**P9R** Estimates regression equations for "best" (by R-squared, adjusted R-squared, or Mallows' C(p) criterion) subset of predictor variables by Furnival-Wilson algorithm. Options: weights, form of input, Durbin-Watson statistic. Cook's distance and several types of residuals may be printed, plotted, or saved.

**MINITAB interactive system**

**REGRESS** Performs simple or multiple linear regression, prints standard results. Options: amount of output, save results, weights, handle missing values, through the origin, compute and save regression diagnostics, lack of fit tests.

**L8a10 :** Hypothesis testing, inference
### L8a10a: Lack-of-fit tests

**IMSL subprogram library**

- **RLFITI**: Pure replication error degrees of freedom and sum of squares - in-core version.
- **RLFITO**: Pure replication error degrees of freedom and sum of squares - out-of-core version.

### L8a10b: Analysis of residuals

**IMSL subprogram library**

- **RLRES**: Perform a residual analysis for a fitted regression model.

### L8a10c: Inference

**IMSL subprogram library**

- **RLINPF**: Inverse prediction using a fitted simple linear regression model.
- **RLOPDC**: Response prediction using an orthogonal polynomial regression model.
- **RLPRDI**: Confidence intervals for the true response and for the average of a set of future observations on the response - in-core version.
- **RLPRDO**: Confidence intervals for the true response and for the average of a set of future observations on the response - out-of-core version.

### L8b: Biased (ridge)

**BMDP program library**

- **P3R**: Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots.
- **PAR**: Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridging, four plots, fitting models defined by differential equations.

### L8c: Linear least absolute value (L-1)

**IMSL subprogram library**

- **RLLMV**: Perform linear regression using the minimax criterion.

### L8d: Linear minimax (L-infinity)

**IMSL subprogram library**

- **RLLAV**: Perform linear regression using the least absolute values criterion.
### Robust

<table>
<thead>
<tr>
<th><strong>EDA</strong></th>
<th><strong>RLINE</strong></th>
<th>Fits straight line to x-y data by resistant line procedure – partitions data by x-value into three groups and uses an iterative procedure to find the line that makes the median residual in the left and the right partitions equal.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>RSMOOTH</strong></td>
<td>Computes resistant smoother by 4253H, twice (or 3RSSH, twice), i.e. successive application of running medians and Hanning (running weighted averages), and save results.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Nonlinear</strong></th>
<th><strong>INVAR program library</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INVAR1</strong></td>
<td>Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only.</td>
</tr>
<tr>
<td><strong>INVAR2</strong></td>
<td>Interactive program for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA graphics.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Unweighted</strong></th>
<th><strong>Derivatives not supplied</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L8g1a</strong></td>
<td><strong>BMDP program library</strong></td>
</tr>
<tr>
<td></td>
<td><strong>PAR</strong></td>
</tr>
<tr>
<td></td>
<td><strong>CMLIB subprogram library (NL2SN sublibrary)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>NL2SN</strong></td>
</tr>
<tr>
<td></td>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RSMITZ</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ZXSSQ</strong></td>
</tr>
<tr>
<td></td>
<td><strong>STATLIB subprogram library</strong></td>
</tr>
<tr>
<td></td>
<td><strong>NFIT</strong></td>
</tr>
<tr>
<td></td>
<td><strong>NFITS</strong></td>
</tr>
<tr>
<td>L8g1b</td>
<td>Derivatives supplied</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td><strong>BMDP program library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>P3R</strong></td>
<td>Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots.</td>
</tr>
<tr>
<td><strong>CMLIB subprogram library (NL2SN sublibrary)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>NL2S1</strong></td>
<td>Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Double precision version is DNL2S1.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L8g2</th>
<th>Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BMDP program library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>L8g2a</strong></td>
<td>Derivatives not supplied</td>
</tr>
<tr>
<td><strong>BMDP program library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>PAR</strong></td>
<td>Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridging, four plots, fitting models defined by differential equations.</td>
</tr>
<tr>
<td><strong>CMLIB subprogram library (NL2SN sublibrary)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>NL2SN</strong></td>
<td>Minimizes a nonlinear sum of squares using residual values only. Double precision version is DNL2SN.</td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ZXSSQ</strong></td>
<td>Minimum of the sum of squares of m functions in n variables using a finite difference Levenberg-Marquardt algorithm.</td>
</tr>
<tr>
<td><strong>STATLIB subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>NFITW</strong></td>
<td>Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm.</td>
</tr>
<tr>
<td><strong>NFITWS</strong></td>
<td>Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L8g2b</th>
<th>Derivatives supplied</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BMDP program library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>P3R</strong></td>
<td>Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots.</td>
</tr>
<tr>
<td><strong>CMLIB subprogram library (NL2SN sublibrary)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>NL2S1</strong></td>
<td>Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Double precision version is DNL2S1.</td>
</tr>
</tbody>
</table>
Categorical data analysis is concerned with the analysis of counts of observations assigned to categories (commonly displayed as contingency tables).

While special purpose software is available for analysing two-by-two and two-way tables, software for the log-linear model allows analysis of higher dimensional tables, modeling, parameter estimation, analysis of residuals, and analysis of special types of tables (e.g., those with structural zeros). Some EDA techniques use the median instead of the mean as a measure of central tendency. Graphical displays of parameter estimates and of residuals are useful in analyzing the quality of a fitted model.

Software is classified both at the leaves of the tree and at higher levels in this chapter; the latter perform a variety of categorical data analysis computations. Software for summarizing data in tabular form is also classified in class L2b.

References

January 1984

GAMS: Modules by Class

L9a: 2-by-2 tables

IMSL subprogram library

NHEXT Fishers exact method for 2-by-2 tables.

L9b: Two-way tables

IMSL subprogram library

BDTWT Computations of a two-way frequency table.
CBNRHO Estimation of the bivariate normal correlation coefficient using a contingency table.
CTPR Compute exact probabilities for contingency tables.
CTRBYC Analysis of a contingency table.

MINITAB interactive system

CHISQUARE Performs chi-square test for association (non-independence) on a two-way table and prints standard results.

NAG subprogram library

G01AFE Two-way contingency table analysis. Double precision version is G01AFF.

L9c: Log-linear model

IMSL subprogram library

CTLLF Log-linear fit of contingency table.

L9d: EDA (e.g., median polish)

IMSL subprogram library

BEMDP Median polish of a two-way table.

MINITAB interactive system

MPOLISH Uses median polish to fit an additive model to a two-way layout which may be unbalanced and may have empty cells. Options: fit columns first, number of iterations, save results.

L10: Time Series Analysis

Two widely used approaches to time series analysis are Box-Jenkins methods and spectral analysis. The Box-Jenkins approach seeks to build models with autoregressive terms (to model the "memory" of the underlying system) and moving average terms (to model the random effects). The method seems most appropriate when the source of the time series data is a structured system subject to random forcings. Business and economic time series have been successfully analyzed using this approach, which is currently also used in many engineering applications.

Spectral analysis, which had its origin in the work of Fourier, seeks to replace a given time series with its representation in terms of frequencies. Assuming that the time series can be modelled as a superposition of oscillations with the appropriate frequencies and amplitudes, spectral analysis seeks to isolate those characteristics. The method is most appropriate for series with strong deterministic components as well as noise. It is commonly used in physical science and engineering.
applications, and also has been used successfully in biology and the social sciences.

The two methods are not mutually exclusive, and in fact share a number of techniques (e.g., autocorrelation analysis). Subchapters in this chapter include time series data manipulation (e.g., filtering (L10b)), autocorrelation analysis (L10c), Box-Jenkins ARMA and ARIMA techniques (L10e), and spectral analysis (L10f). Cross-correlation analysis (L10g) deals with more than one time series.

The software in this chapter includes subroutines from CMLIB (MAXENTROPY sublibrary), DATAPAC, IMSL, NAG, and STATLIB, programs from BMDP, interactive system commands from Minitab, and the interactive system SPECAN (with high resolution graphics capabilities) from the Spectral library.

<table>
<thead>
<tr>
<th>L10 :</th>
<th>Time series analysis <em>(search also class L3c5 for time series graphics)</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>L10a :</td>
<td>Transformations, transforms <em>(search also class J1)</em></td>
</tr>
<tr>
<td></td>
<td><strong>DATAPAC subprogram library</strong></td>
</tr>
<tr>
<td>FOURIE</td>
<td>Performs a Fourier analysis of the data in the input vector X.</td>
</tr>
<tr>
<td></td>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td>FTRDIF</td>
<td>Transformations, differences and seasonal differences of a time series for model identification.</td>
</tr>
<tr>
<td></td>
<td><strong>MINITAB interactive system</strong></td>
</tr>
<tr>
<td>DIFFERENCES</td>
<td>Computes differences between observations at a specified lag in a time series.</td>
</tr>
<tr>
<td>LAG</td>
<td>Computes lagged observations in a time series.</td>
</tr>
<tr>
<td>L10b :</td>
<td>Smoothing, filtering</td>
</tr>
<tr>
<td></td>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td>FTKALM</td>
<td>Kalman filtering.</td>
</tr>
<tr>
<td>FTRDIF</td>
<td>Transformations, differences and seasonal differences of a time series for model identification.</td>
</tr>
<tr>
<td>ICSMOU</td>
<td>One-dimensional data smoothing by error detection.</td>
</tr>
<tr>
<td></td>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td>G13AAE</td>
<td>Carries out non-seasonal and seasonal differencing on a time series. Information which allows the original series to be reconstituted from the differenced series is also produced. This information is required in time series forecasting. Double precision version is G13AAF.</td>
</tr>
<tr>
<td>G13BAE</td>
<td>Filters a time series by an ARIMA model. Double precision version is G13BAF.</td>
</tr>
<tr>
<td></td>
<td><strong>STATLIB subprogram library</strong></td>
</tr>
<tr>
<td>MOVAVG</td>
<td>Computes a k-term symmetric moving average of a series.</td>
</tr>
<tr>
<td>TSDIFF</td>
<td>Performs a user-controlled differencing operation on a series.</td>
</tr>
</tbody>
</table>

**References**

L10c: Autocorrelation analysis

DATAPAC subprogram library

AUTOCO Computes the sample autocorrelation coefficient of the data in the input vector X.

TIME Performs a time series analysis on the data in the input vector X.

IMSL subprogram library

FTAUTO Mean, variance, autocovariances, autocorrelations, and partial autocorrelations for a stationary time series.

MINITAB interactive system

ACF Computes and graphs the autocorrelations of a time series, and optionally saves results.
PACF Computes and graphs partial autocorrelations of a time series and optionally saves results.

NAG subprogram library

G13ABE Computes the sample autocorrelation function of a time series. It also computes the sample mean, the sample variance and a statistic which may be used to test the hypothesis that the true autocorrelation function is zero. Double precision version is G13ABF.

G13ACE Calculates partial autocorrelation coefficients given a set of autocorrelation coefficients. It also calculates the predictor error variance ratios for increasing order of finite lag autoregressive predictor, and the autoregressive parameters associated with the predictor of maximum order. Double precision version is G13ACF.

STATLIB subprogram library

ACORR Performs autocorrelation analysis of a series.
ACORRD Performs autocorrelation analysis of a series differenced with a user-controlled differencing operation.
ACORRSS Performs autocorrelation analysis of a series, with computed results returned to the user.

L10d: Complex demodulation

DATAPAC subprogram library

DEMOD Performs a complex demodulation on the data in the input vector X at the input demodulation frequency \( F \).

L10e: ARMA and ARIMA modeling and forecasting

BMDP program library

P2T Interactive or batch Box-Jenkins time series analysis for univariate time domain models (including ARIMA, regression, intervention, and transfer function models) - model identification, parameter estimation, testing, forecasting. Options: print, plot, differencing and filtering, save results.

IMSL subprogram library

FTCP Non-seasonal ARIMA (Box-Jenkins) stochastic model analysis for a single time series with full parameter iteration and maximum likelihood estimation.

MINITAB interactive system

ARIMA Fits non-seasonal and seasonal models to a time series with \( p \) the order of the AR part, \( d \) the number of differences, \( q \) the order of the MA part and with optional seasonality with period \( S \), AR order \( P \),
number of differences D, and MA order Q. Options: starting values, forecasting, save results.

### L10e1: Model and parameter estimation

**IMSL subprogram library**
- **FTARPS** Preliminary estimation of the autoregressive parameters in an ARIMA stochastic model.
- **FTMA** Preliminary estimation of the moving average parameters in an ARIMA stochastic model.
- **FTML** Maximum likelihood estimation of autoregressive and moving average parameters in an ARIMA (Box-Jenkins) stochastic model.

**NAG subprogram library**
- **G13AHE** Calculates preliminary estimates of the parameters of an autoregressive moving-average (ARMA) model from an autocorrelation function. Double precision version is G13ADF.
- **G13AEE** Iteratively fits seasonal autoregressive-integrated moving-average (ARIMA) model to observed time series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGE or G13AHE in forecasting. Double precision version is G13AFF.
- **G13AFE** Easy-to-use version of G13AEE. Iteratively fits seasonal ARIMA model to observed series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGE or G13AHE in forecasting. Double precision version is G13AFF.

**STATLIB subprogram library**
- **ARIMAE** Performs least squares estimation of the parameters in an ARIMA (Box-Jenkins) model using an adaptation of Pack's code.

### L10e2: Forecasting

**IMSL subprogram library**
- **FTCAST** Time series forecasts and probability limits using an ARIMA (Box-Jenkins) model.
- **FTWEIN** Wiener forecast for a stationary stochastic process.

**NAG subprogram library**
- **G13AGE** Accepts new observation of fully specified (by G13AEE or G13AFE) time series and updates "state set" (from G13AEE or G13AFE) information for use in further forecasting. Returns residuals corresponding to the new observations, which may be used in checking that new observations conform to previously fitted model. Double precision version is G13AGF.
- **G13AHE** Produces forecasts of a time series, given model already fitted (by G13AEE or G13AFE). Original observations are not required, since the subroutine uses state set produced originally by G13AEE or G13AFE or updated by G13AGE. Standard errors of the forecasts are also provided. Double precision version is G13AHF.
- **G13AJE** Applies a fully specified seasonal ARIMA model to an observed time series, generates the state set for forecasting and (optionally) derives a specified number of forecasts together with their standard deviations. Double precision version is G13AJF.

**STATLIB subprogram library**
- **ARIMAF** Performs minimum mean square error forecasts for a given (fitted) ARIMA (Box-Jenkins) model, using an adaptation of Pack's code.
January 1984

GAMS: Modules by Class

BMDP program library

P1T Interactive or batch spectral analysis of one or two time series, with estimates of spectral density and coherence between variables. Options: print, plot (variable vs. time, lagged plots, complex demodulation, periodogram), handle missing values, remove seasonal means and linear trend, filtering, save results.

CMLIB subprogram library (MAXENTROPY sublibrary)

BURG Computes the coefficients of a finite length causal forward or backward prediction filter and uses both the forward and backward predictions in a symmetric manner to generate the maximum entropy spectrum by means of a Toeplitz recursion.

IMSL subprogram library

FTFPS Fast Fourier transform estimates of power spectra and cross spectra of time series.

FTFREQ Single or multichannel time series analysis in the time and frequency domains.

NAG subprogram library

G13CAE Calculates the smoothed sample spectrum of a univariate time series using one of four windows - rectangular, Bartlett, Tukey, or Parzen window. Double precision version is G13CAF.

G13CBE Calculates the smoothed sample spectrum of a univariate time series using spectral smoothing by the trapezium frequency (Danielli) window. Double precision version is G13CBF.

G13CCE Calculates the smoothed sample cross spectrum of a bivariate time series using one of four lag windows - rectangular, Bartlett, Tukey, or Parzen window. Double precision version is G13CCF.

G13CDE Calculates the smoothed sample cross spectrum of a bivariate time series using spectral smoothing by the trapezium frequency (Danielli) window. Double precision version is G13CDF.

G13CEE For a bivariate time series, calculates the cross amplitude spectrum and squared coherency, together with lower and upper bounds from the univariate and bivariate (cross) spectra. Double precision version is G13CEF.

G13CFF For a bivariate time series, calculates the gain and phase together with lower and upper bounds from the univariate and bivariate spectra. Double precision version is G13CFF.

G13CGF For a bivariate time series, calculates the noise spectrum together with multiplying factors for the bounds and the impulse response function and its standard error, from the univariate and bivariate spectra. Double precision version is G13CGF.

SPECTRLAN program library

SPECAN An interactive spectral analysis package for time series data. Produces periodograms, cumulative periodograms, continuous Fourier power spectra, cumulative power spectra, Fourier amplitude spectra, maximum entropy spectra, and integrated maximum entropy spectra. With DISSPLA graphics.

STATLIB subprogram library

ACSPEC Computes the series autospectrum from the Fourier transform of the user-supplied autocorrelation function, with user-supplied lag window truncation values.

ASPEC Computes the series autospectrum from the Fourier transform of the autocorrelation function, using the Jenkins and Watts window closing technique.

ASPECS Computes the series autospectrum from the Fourier transform of the autocorrelation function, with user-supplied lag window truncation values.

IASPEC Computes the integrated sample periodogram of a series (not recommended for long series because of the algorithm used).
Cross-correlation analysis

Parameter estimation

**IMSL subprogram library**

- **FTCROS** Means, variances, cross-covariances, and cross-correlations for two mutually stationary n channel time series.
- **FTCRXY** Cross-covariance of two mutually stationary time series.
- **FTFPS** Fast Fourier transform estimates of power spectra and cross spectra of time series.
- **FTTR** Parameter estimates for a univariate transfer function model.

**MINITAB interactive system**

- **CCF** Computes and graphs cross-correlations between two time series.

**NAG subprogram library**

- **G13BCE** Calculates cross correlations between two time series. Double precision version is G13BCF.

**STATLIB subprogram library**

- **CCORRS** Perform cross-correlation analysis between a pair of series.
- **CCSPEC** Computes phase and squared coherence spectra from the Fourier transform of the user-supplied correlation functions, with user-supplied lag window truncation values.
- **CSPEC** Computes phase and squared coherence spectra from the Fourier transform of the correlation functions, using the Jenkins and Watts window closing techniques.
- **CSPECS** Computes phase and squared coherence spectra from the Fourier transform of the correlation functions, with user-supplied lag window truncation values.
- **ICSPEC** Displays plots of the integrated sample phase and co-spectra for a pair of series (not recommended for long series because of the algorithm used).

**Forecasting**

**IMSL subprogram library**

- **FTWENM** Multichannel Wiener forecast.
- **FTWENX** Maximum likelihood parameter estimates for a multichannel, single output time series model.

### L11: Correlation Analysis

Correlation analysis deals with the relationships among one or more variables or sets of variables. Parametric analyses make assumptions about the distributions of the variables, while distribution-free (nonparametric) analyses make fewer or less restrictive assumptions.

Chapter L11 contains software for more sophisticated correlation analyses than the software in chapters L1e (for calculating parametric and nonparametric correlations) and chapters L4b and L4d (for calculation, elementary inference, and hypothesis testing of correlation statistics).

Consult the references listed in the introduction to chapter L for more information about correlation analysis.
L11: Correlation analysis

BMDP program library

P6M Computes canonical correlation analysis for two sets of variables and Bartlett’s test for the significance of the remaining eigenvalues, with printing. Options: weights, form of input, additional printing and plotting, save results.

P6R Computes the partial correlations of a set of variables after removing the linear effects of a second set of variables. Can be used for regression, especially if multiple dependent variables are present. Prints standard results. Options: weights, form of input, additional printing and plots.

STATLIB subprogram library

COREL Performs correlation analysis of a multivariate random sample.

CORELS Performs correlation analysis of a multivariate random sample with computed results returned to the user.

L12: Discriminant Analysis

Discriminant analysis is concerned with distinguishing among two or more populations. A “training” data set may be used to develop a discriminant function (commonly linear in the observational data) which is then used with data whose population membership is known. Classical discriminant analysis assumes that each population is multivariate normal with known variance-covariance structure, and substantial statistical inference is possible in this case. Stepwise techniques are commonly used to select a discriminant function. Both statistical and graphical techniques are useful in evaluating results.

This chapter contains one BMDP program and two IMSL subroutines; additional discriminant analysis software exists, especially of the distribution-free (nonparametric) form appropriate when the classical assumptions are inappropriate, and that software will be included in future editions.

January 1984

References


L12: Discriminant analysis

BMDP program library

P7M Stepwise forward or backward discriminant analysis (including jackknifed classifications, percent correctly classified, Mahalanobis’ distances, canonical variable coefficients, eigenvalues, scores, and plot of first two canonical variables). Options: interactive stepping, save results.

IMSL subprogram library

ODFISH Linear discriminant analysis method of Fisher for reducing the number of variables.

ODNORM Multivariate normal linear discriminant analysis among several known groups.

L13: Factor Analysis

Factor analysis is a statistical procedure used to analyze multivariate data. Factor analysis itself is commonly used in the social sciences. Principal components analysis, the most widely used special case of factor analysis, is commonly used with high dimensional data, and yields lower dimensional data which approximate the original data. Software for performing regression analysis using principal components as independent variables is classified at class L8a4a1d.
References

## L14: Cluster Analysis

Cluster analysis is helpful in discovering structure in a large set of multivariate observations. Observations are clustered into groups such that observations in the same cluster are similar according to some criterion.

Starting from a matrix whose entries are measures of similarity or dissimilarity (e.g., Euclidean distance) between pairs of observations, one form of cluster analysis merges observations or clusters into larger clusters (using some measure such as the average distance between observations in two clusters) until some stopping criterion is met. Other types of cluster analysis divide the observations into clusters, produce overlapping clusters, and have constraints.

Future editions of GAMS will contain additional cluster analysis software.

January 1984
References

L14 : Cluster analysis

L14a : Unconstrained

L14a1 : Nested

L14a1a : Joining (e.g., single link)

BMDP program library

P1M Stepwise cluster analysis of variables using one of four measures of similarity, three criteria for combining clusters, with printing of a summary table of clusters, shaded distance measure display, and a tree showing cluster formation. Options: form of input, additional printing and display.

P2M Stepwise cluster analysis of cases (observations) using one of four distance measures (including Euclidean and one for data that are frequency counts) and three linkage algorithms (single, centroid, k nearest neighbors), with a summary table of clusters and a cluster tree. Options: weights, standardized data.

IMSL subprogram library

OCDIS Pairwise Euclidean distances between the columns of a matrix.

OCLINK Perform a single-linkage or complete-linkage hierarchical cluster analysis given a similarity matrix.

L14a1b : Divisive

L14a2 : Non-nested

BMDP program library

P3M Forms blocks (submatrices of the data matrix) where a subset of the cases (for a subset of the variables) cluster together, with printing of the blocks and tree diagrams for cases and for variables - appropriate for categorical data with few levels.

PKM By k-means procedure, partitions a set of cases (observations) into k clusters - beginning with user-specified initial clusters or one cluster, proceeding in divisive stepwise manner, then doing iterative reallocation - prints cluster profile and plot. Options: weights, standardize data (four ways), save results.

L14b : Constrained

L14b1 : One-dimensional

L14b2 : Two-dimensional
L15: Life Testing, Survival Analysis

Two BMDP programs and one IMSL subroutine provide software for analysis of survival data. Other software available in the biomedical and the physical sciences communities will be included in future editions.

\[ \text{January 1984} \]

BMDP program library

- **P1L** Estimates survival (time-to-response) distribution of patients who have been observed over varying periods of time by product-limit (Kaplan-Meier) or actuarial life table (Cutler-Ederer) method. Options: three forms of input, Mantel-Cox and Breslow test of equality of survival curves, five plots.

- **P2L** Analyzes survival data with covariates using Cox proportional hazard regression model. Options: two forms of input, stepwise selection of covariates, time-dependent covariates, stratification, significance tests, three plots, print survival functions and residuals.

IMSL subroutine library

- **USTREE** Print a binary tree (which may represent the output of a clustering algorithm in chapter O).

- **CLIPE** Life table analysis.
M: Simulation, Stochastic Modelling

In order to build realistic mathematical models one must often account for the possibility of randomness. This is because elements of the phenomena under study cannot often be predicted in advance, but rather exhibit a probabilistic behavior which should be accounted for in the model. Probability models can be used to address a variety of questions such as “How may cabs should be on the street?”, “How much time should pass before scheduling maintenance?”, “How many beds should a hospital provide?”, or “When should I replace my machines?”. These questions share the common characteristic noted above—certain elements on which the answers depend can only be predicted in a statistical sense.

Software which can be used to build and to study stochastic models is classified in this chapter. A variety of such software exists, and will be cataloged in future editions of GAMS.

References

N: Data Handling

Software for data handling provides the basic tools needed to manipulate data structures. A data structure is a method for organizing data, together with a set of admissible operations on the data. While the most familiar data structures include real numbers, arrays, and sequential files, many more exotic data types have proven useful. Examples of these are character strings, hash tables, lists, stacks, trees, heaps, queues, and many types of files. The most common types of operations on complex data structures are input, output, insertion, deletion, searching, and sorting.

The input/output software cataloged here provides capabilities for the input and output of data types not usually available as standard options in programming languages. These capabilities include free-format input and easy-to-read output of vectors and matrices.

The data structures handled by software cataloged here are sequences and stacks. A sequence is a simple list, usually stored as an array. A stack is a list in which items are added or deleted (and retrieved) on a last-in first-out basis. The subroutines for stack management are from the PORT library; while these routines are principally used by other PORT routines (and are invisible to the user), they may be useful in their own right.

Software for sorting is described in the text for class N6 below.

REFERENCES


N1: Input, output

DATAPAC subprogram library

READ Performs a format-free read.
READG Performs a format-free read of data from input unit = IRD.
SKIPR Reads through (skips over) NLHEAD lines from input unit = 5.
WRITE Writes out the contents of the vector X in an orderly and neat fashion.

IMSL subprogram library

USCWV Print a complex vector.
USWBM Print a matrix stored in band storage mode.
USWBS Print a matrix stored in band symmetric storage mode.
USWCH Print a complex matrix stored in Hermitian storage mode.
USWCM Print a complex matrix stored in full storage mode.
USWFM Print a matrix stored in full storage mode.
USWFV Print a vector.
USWSM Print a matrix stored in symmetric storage mode.

N2: Bit manipulation

IMSL subprogram library

GTPBC Count of the number of zero bits in a given subset of a real word.
**N3**: Character manipulation

**BMDP program library**

**P4D** Counts frequency of each number, letter, or symbol in single-column fields (A1 format). Options: input case label variables in A4 format, diagnostic printing useful in preliminary data screening. Specified characters may be replaced by blanks or symbols.

**N4**: Storage management (e.g., stacks, heaps, trees)

**IMSL subprogram library**

**USTREE** Print a binary tree (which may represent the output of a clustering algorithm in chapter 0).

**PORT subprogram library**

**ISTKGT** Allocates (gets) an array from the storage stack for PORT library programs.
**ISTKIN** Initialize the length of the dynamic storage stack for PORT library programs.
**ISTKMD** Changes size of last stack allocation for PORT library programs.
**ISTKQU** Returns the number of available items that remain in the stack for PORT library programs.
**ISTKRL** Releases the last storage allocations requested for PORT library programs.
**ISTKST** Returns information on the status of the stack for PORT library programs.

**N5**: Searching

**N5a**: Extreme value

**PORT subprogram library**

**EXTRMI** Finds extremal points of an integer function defined on a mesh.
**EXTRMR** Finds extremal points of a real function defined on a mesh. Double precision version is EXTRMD.
**INTRVI** Finds the interval in an integer array to which an integer element belongs.

**N5b**: Insertion position

**PORT subprogram library**

**INTRVR** Finds the interval in a real array to which a real element belongs. Double precision version is INTRVD.

**N5c**: On a key

**N6: Sorting**

Sorting is the rearrangement of units of data (called records) so that a particular data field (the key) is in ascending or descending order. Although more common in business processing, this problem has also seen many scientific applications since Von Neumann coded sorting algorithms on the EDVAC in 1945. Techniques for sorting can be generally classified into two types—internal or external—depending upon whether all the data fits in memory or not.
Internal sorting programs usually accept a one-dimensional array of keys and one or more "parallel" data arrays. The output of the program is either a set of rearranged data arrays ("active sorting") or a permutation array P ("passive sorting"). In the latter case the data is not reordered, but instead one may find the i-th element in sorted order in the P(i)th position. Good general purpose algorithms are those that minimize data comparisons and exchanges and do not have large memory requirements. Such programs usually run in time proportional to n log n, where n is the number of records. C. A. R. Hoare's QUICKSORT is generally considered the best. These considerations may change for machines with special architecture, however.

External sorting methods are usually combinations of internal sorting and external merges, and here one is more interested in minimizing the amount of time spent in input or output of data. Most large computer manufacturers (those supporting COBOL, for instance), provide sorting programs as part of the operating system, although not all will interface with Fortran.

References

VSAR  Sorting of matrices (with options).

MINITAB interactive system

RANK  Ranks the values in a vector. Ties are assigned the average rank.

NAG subprogram library

M01AAE  Passively sort a real vector into ascending order. Double precision version is M01AAF.
M01ABE  Passively sort a real vector into descending order. Double precision version is M01ABF.

PORT subprogram library

SRTPAR  Passively sorts real data into ascending order. Double precision version is SRTPAD.
SRTPDR  Passively sorts real data into descending order. Double precision version is SRTPDD.

N6a1b2  :  Double precision

N6a1c  :  Character

PORT subprogram library

SRTPAH  Passively sorts Hollerith data into ascending order.
SRTPDH  Passively sorts Hollerith data into descending order.

N6a2  :  Active

N6a2a  :  Integer

CMLIB subprogram library (SSORT sublibrary)

ISORT  Sorts an integer array in either increasing or decreasing order. Optionally another integer array can be carried along.

NAG subprogram library

M01AGE  Actively sort the rows of an integer matrix into ascending order of an index column. (Identical to M01AGF.). Double precision version is M01AGF.
M01AHE  Actively sort the rows of an integer matrix into descending order of an index column. (Identical to M01AHF.). Double precision version is M01AHF.
M01ALE  Actively sort an integer vector into ascending order of an index column and provide an index to the original order. (Identical to M01ALF.). Double precision version is M01ALF.
M01AME  Actively sort an integer vector into descending order of an index column and provide an index to the original order. (Identical to M01AMF.). Double precision version is M01AMF.
M01AQE  Actively sort an integer vector into ascending order (Singleton's implementation of Quicksort). (Identical to M01AQF.). Double precision version is M01AQF.
M01ARE  Actively sort an integer vector into descending order (Singleton's implementation of Quicksort). (Identical to M01ARF.). Double precision version is M01ARF.

PORT subprogram library

SRTAI  Actively sorts integer data into ascending order.
<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library (SSORT sublibrary)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SSORT</strong></td>
<td>Sorts an array (X) (of (N) real numbers) into increasing or decreasing order. An optional array (Y) is carried along with (X).</td>
</tr>
<tr>
<td><strong>DATAPAC subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SORT</strong></td>
<td>Sorts (in ascending order) the (N) elements of the vector (X) and puts the resulting (N) sorted values into the vector (Y).</td>
</tr>
<tr>
<td><strong>SORTC</strong></td>
<td>Sorts (in ascending order) the (N) elements of the vector (X) and rearranges the elements of the vector (Y).</td>
</tr>
<tr>
<td><strong>SORTP</strong></td>
<td>Sorts (in ascending order) the (N) elements of the vector (X), puts the resulting (N) sorted values into the vector (Y), and puts the position (in the original vector (X)) of each of the sorted values into the single precision vector (XPOS).</td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>VSAR</strong></td>
<td>Sorting of matrices (with options).</td>
</tr>
<tr>
<td><strong>VSORA</strong></td>
<td>Sorting of columns of a real matrix into ascending order of keys in rows.</td>
</tr>
<tr>
<td><strong>VSRTA</strong></td>
<td>Sorting of arrays by algebraic value.</td>
</tr>
<tr>
<td><strong>VSRTM</strong></td>
<td>Sorting of arrays by absolute value.</td>
</tr>
<tr>
<td><strong>VSRTP</strong></td>
<td>Sorting of arrays by absolute value permutations returned.</td>
</tr>
<tr>
<td><strong>VSRTR</strong></td>
<td>Sorting of arrays by algebraic value permutations returned.</td>
</tr>
<tr>
<td><strong>MINITAB interactive system</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ORDER</strong></td>
<td>Sorts in ascending order the values in each of one or more vectors.</td>
</tr>
<tr>
<td><strong>SORT</strong></td>
<td>Sorts a vector in ascending order and optionally carries along other vectors.</td>
</tr>
<tr>
<td><strong>NAG subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>M01AEE</strong></td>
<td>Actively sort the rows of a real matrix into ascending order of an index column. Double precision version is M01AEF.</td>
</tr>
<tr>
<td><strong>M01AFE</strong></td>
<td>Actively sort the rows of a real matrix into descending order of an index column. Double precision version is M01AFF.</td>
</tr>
<tr>
<td><strong>M01AJE</strong></td>
<td>Actively sort a real vector into ascending order and provide an index to the original order. Double precision version is M01JF.</td>
</tr>
<tr>
<td><strong>M01AKE</strong></td>
<td>Actively sort a real vector into descending order and provide an index to the original order. Double precision version is M01KF.</td>
</tr>
<tr>
<td><strong>M01ANE</strong></td>
<td>Actively sort a real vector into ascending order (Singleton’s implementation of Quicksort). Double precision version is M01ANF.</td>
</tr>
<tr>
<td><strong>M01APE</strong></td>
<td>Actively sort a real vector into descending order (Singleton’s implementation of Quicksort). Double precision version is M01APF.</td>
</tr>
<tr>
<td><strong>PORT subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SRTAR</strong></td>
<td>Actively sorts real data into ascending order. Double precision version is SRTAD.</td>
</tr>
</tbody>
</table>
SRTDR

Actively sorts real data into descending order. Double precision version is SRTDD.

N6a2b2

Double precision

IMSL subprogram library

VSODA

Actively sorts real data into descending order. Double precision version is SRTDD.

N6a2c

Character

CMLIB subprogram library (SSORT sublibrary)

CSORT

Sorts a character array in either increasing or decreasing order. Optionally another character array can be carried along.

NAG subprogram library

M01BAE

Actively sort a character vector into reverse alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BAF.). Double precision version is M01BBF.

M01BBE

Actively sort a character vector into alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BBF.). Double precision version is M01BBF.

M01BCE

Actively sort the columns of a character matrix into reverse alphanumeric order of an index column. (Identical to M01BCF.). Double precision version is M01BCF.

M01BDE

Actively sort the columns of a character matrix into alphanumeric order of an index column. (Identical to M01BDF.). Double precision version is M01BDF.

PORT subprogram library

SRTAH

Actively sorts Hollerith data into ascending order.

SRTDH

Actively sorts Hollerith data into descending order.

N6b

External

N7

Merging

N8

Permuting

IMSL subprogram library

VSRTU

Interchange the rows or columns of a matrix using a permutation vector such as the one obtained from IMSL routines VSRTP or VSRTR.

PORT subprogram library

SRTRH

Rearranges Hollerith data according to permutation stored in IP.

SRTRI

Rearranges integer data according to permutation stored in IP.

SRTRR

Rearranges real data according to permutation stored in IP. Double precision version is SRTRD.
O: Symbolic Computation

This chapter is designed to contain software for the manipulation of mathematical expressions in their natural, symbolic form rather than just the manipulation of numbers. Such programs have been in use for some time to perform tedious, but important, symbolic computations in a number of fields such as celestial mechanics and quantum electrodynamics. Many standard operations of algebra and the calculus, including polynomial and rational arithmetic, differentiation, integration, and taking limits can routinely be done in such systems.

Unfortunately, no software for symbolic computations is cataloged in the current edition of GAMS. Information about existing software can be found in the references.

References

P: Computational Geometry

Computational geometry is the design and analysis of algorithms for geometric computations. One example of a computational geometry problem is finding the convex hull of a set of n points in the Euclidean plane, i.e., the smallest convex set containing the points. While naive algorithms involve examining all pairs of points and require \( O(n^2) \) operations, more efficient algorithms require \( O(n \log n) \) operations, and the newest algorithms require, on average, \( O(n) \) operations when the spatial distribution of the points is random. Since each point must be examined at least once, \( O(n) \) is obviously the lower bound on the number of operations. The convex hull of a set of points can be used to efficiently find the circle of smallest radius containing all of the points; the center of the circle is the location which minimizes the maximum distance to any of the given points and would "be a suitable location for an emergency service facility if it were desired to minimize the worst-case response time" (Shamos, 1977). Other areas of computational geometry applications include statistics and graphics.

A substantial amount of computational geometry software has been written, and we hope to include more in future editions of GAMS.

REFERENCES


P : Computational geometry (search also classes G, Q)

P1 : One dimension

P2 : Two dimensions

P2a : Points, lines

P2a1 : Relationships

P2a1a : Closest and farthest points

P2a1b : Intersection

P2a2 : Graph construction

P2a2a : Convex hull

P2a2b : Minimum spanning tree

P2a2c : Region partitioning
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2a2c1</td>
<td>Triangulation</td>
</tr>
<tr>
<td></td>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td>D03MAE</td>
<td>Triangulation of a plane region. Double precision version is D03MAF.</td>
</tr>
<tr>
<td>P2a2c2</td>
<td>Voronoi diagram</td>
</tr>
<tr>
<td>P2b</td>
<td>Polygons (e.g., intersection, hidden line problems)</td>
</tr>
<tr>
<td>P2c</td>
<td>Circles</td>
</tr>
<tr>
<td>P3</td>
<td>Three dimensions</td>
</tr>
<tr>
<td>P3a</td>
<td>Points, lines, planes</td>
</tr>
<tr>
<td>P3b</td>
<td>Polytopes</td>
</tr>
<tr>
<td>P3c</td>
<td>Spheres</td>
</tr>
<tr>
<td>P4</td>
<td>More than three dimensions</td>
</tr>
</tbody>
</table>
Q: Graphics

The simplest type of graphics output device imaginable—the line printer—is the only type of device supported by the software currently cataloged in this chapter. For statistical data analysis, this type of software is often sufficient. More sophisticated graphics software can be used to produce graphs and diagrams on graphics hardware devices such as drum or flatbed plotters, electrostatic printer/plotters, or CRT screens. This software typically has many internal routines written for specific hardware and thus is often difficult to transport from one machine to another.

Users may have software control of graphics devices at a number of different levels. For example, one Fortran subroutine call might simply draw a line between two specified points. Another might plot a smooth curve through a given set of points. Yet another might draw a two-dimensional perspective plot of a surface in three dimensions represented by a table of function values, complete with the title and axis labels.

The most frequently used line-printer plots are histograms for univariate data and scatter diagrams for bivariate data. Options for these plots include the axis limits, special plot characters, printing multiple plots on one set of axes, selection of a subset of the data, and the size of the plot. Software for line-printer plots of time series, pseudo-three-dimensional plots, a plot of a binary tree, and EDA (experimental data analysis) graphics are also available.

Four presentation-quality graphics software products are available to NBS staff but are not cataloged in this edition of GAMS. They are DATAPLOT (Filliben, 1980), a Tektronix-based interactive system for graphical data analysis, DISSPLA (ISSCO, 1981), a library of Fortran subroutines allowing the user extensive software control, including color, PLOTIO (Tektronix, 1977), Tektronix's basic graphics Fortran subroutine package, and SURFACE II (Sampson, 1978), a high-level control language for plotting surfaces and contour maps and which handles data on irregular grids. DISSPLA and PLOTIO are available on both the Sperry 1100 in Gaithersburg and the Cyber 750 in Boulder, while DATAPLOT is only available on the Sperry 1100 and SURFACE II is only available on the Cyber 750.

January 1984

REFERENCES


Q: Graphics (search also classes L9, P)

Q1: Line printer plotting

DATAPAC subprogram library

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIST</td>
<td>Produces 2 histograms (with differing class widths) of the data in the input vector X.</td>
</tr>
<tr>
<td>PLOT</td>
<td>Yields a one-page printer plot of Y(I) versus X(I).</td>
</tr>
<tr>
<td>PLOT10</td>
<td>Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data, with special plot characters, and with specified axis limits and labels.</td>
</tr>
<tr>
<td>PLOT6</td>
<td>Yields a one-page printer plot of Y(I) versus X(I) for specified axis limits.</td>
</tr>
<tr>
<td>PLOT7</td>
<td>Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits.</td>
</tr>
<tr>
<td>PLOT8</td>
<td>Yields a one-page printer plot of Y(I) versus X(I) with special plot characters for a subset of the data with specified axis limits.</td>
</tr>
<tr>
<td>PLOT9</td>
<td>Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits and axis labels.</td>
</tr>
<tr>
<td>PLOT1C</td>
<td>Yields a one-page printer plot of Y(I) versus X(I) with special plotting characters.</td>
</tr>
<tr>
<td>PLOTICT</td>
<td>Yields a narrow-width (71-character) plot of Y(I) versus X(I) with special plotting characters.</td>
</tr>
</tbody>
</table>
PLOTS Yields a one-page printer plot of $Y(I)$ versus $X(I)$ for a subset of the data.
PLOTSC Yields a one-page printer plot of $Y(I)$ versus $X(I)$ with special characters for a subset of the data.
PLOTST Yields a narrow-width (71-character) of $Y(I)$ versus $X(I)$ for a subset of the data.
PLOTT Yields a narrow-width (71-character) plot of $Y(I)$ versus $X(I)$.
PLOTX Yields a one-page printer plot of $X(I)$ versus $I$.
PLOTXT Yields a narrow-width (71-character) plot of $X(I)$ versus $I$.
PLTSC Yields a narrow-width (71-character) plot of $Y(I)$ versus $X(I)$ with special plot characters and a subset of the data.

**IMSL subprogram library**

USHHST Print a horizontal histogram.
USHST Print a vertical histogram.
USPLO Printer plot of up to ten functions.
USPLOD Printer plot of up to ten functions.
USTREE Print a binary tree (which may represent the output of a clustering algorithm in chapter O).

**MINITAB interactive system**

HISTOGRAM Prints a histogram of the values in each of one or more vectors, with optional user-specification of the first midpoint and the interval width.

LQUERY Prints a letter plot with symbols corresponding to numerical "tag" values. Scale specification is optional.
MQUERY Prints multiple scatter diagrams on the same axis.
PQUERY Prints a scatter diagram, with optional scale specification.
TQUERY Prints pseudo three-dimensional plot of $y$ versus $x$ versus $z$, with symbols indicating the values of $z$, and with optional scale specification.
TSP_QUERY Prints a scatter diagram of a time series, optionally using symbols modulo the period. Handles missing values.

**NAG subprogram library**

G01AGE Line printer scatter plot of two variables. Double precision version is G01AGF.

**STATLIB subprogram library**

MPLT Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable.
MPLTH Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable.
MPLTHL Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.
MPLTL Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits.
PLT Displays a 50x100 character line printer scatter plot.
PLTH Displays a 50x50 character line printer scatter plot.
PLTHL Displays a 50x50 character line printer scatter plot with user control of plot limits.
PLTL Displays a 50x100 character line printer scatter plot with user control of plot limits.
SPLT Displays a 50x100 character line printer scatter plot with user control of the plotting symbol used for each point.
SPLTH Displays a 50x50 character line printer scatter plot with user control of the plotting symbol used for each point.
SPLTHL Displays a 50x50 character line printer scatter plot with user control of the plot limits and of the
plotting symbol used for each point.

**SPLTL** Displays a 50x100 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point.

**VPLT** Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).

**VPLT2** Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis).

**VPLT2L** Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

**VPLTB** Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis).

**VPLTBL** Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.

**VPLTL** Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits.
**R: Service Routines**

This chapter contains subprograms which perform fairly low-level utility functions such as error checking, error handling and retrieval of information about machine characteristics. These routines are generally not of immediate interest to routine users of software libraries. However, they are quite important for those writing software intended to be portable to and reliable on a wide variety of machines.

January 1984

---

**R1: Machine-dependent constants**

Modules classified in this section are useful for increasing the portability of programs. In Fortran, some degree of portability can be obtained by using a subset of Fortran acceptable to all compilers and by isolating all machine-dependent parameters so they are easily changed when moving to a new environment. The latter can be done in two distinct ways. The most direct is to place machine specific information in a visible place in your program and document clearly which machine parameters you are using and how they should be changed to move to another computer, or perhaps to another precision. The advantage of this is that your program becomes fully self-contained. An alternate approach is to use widely available subroutines which return information about the current machine environment. Examples and the routines I1MACH, R1MACH and D1MACH in the CMLIB and PORT libraries, as well as the routines in chapters X01 and X02 of the NAG library. These allow you to write very portable code. When machine information is required it is obtained "automatically" by evaluation of these functions within your program.

The decision as to which of these approaches to use depends mostly on the volume of software that might have to be transported. The first seems quite simple for only one, or at most a few, routines. The second is very useful if a substantial amount of software is being moved, since all the changes are localized in the machine-constant routines. Even when moving a small block of code the latter is often better since it is easy to forget to make small but necessary changes in lower level routines.

January 1984

---

**References**


---

**R1: Machine-dependent constants**

<table>
<thead>
<tr>
<th>Subprogram Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library</strong></td>
<td>(MACHCONST sublibrary)</td>
</tr>
<tr>
<td>I1MACH</td>
<td>Provides integer machine dependent information, e.g. largest integer.</td>
</tr>
<tr>
<td>R1MACH</td>
<td>Provides single precision machine dependent information, e.g. R1MACH(4) returns machine epsilon. Double precision version is D1MACH.</td>
</tr>
<tr>
<td>UGETIO</td>
<td>To retrieve current values and to set new values for input and output unit identifiers.</td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
<td></td>
</tr>
<tr>
<td>X01AAE</td>
<td>Pi. Double precision version is X01AAF.</td>
</tr>
<tr>
<td>X01ABE</td>
<td>Euler's constant, gamma. Double precision version is X01ABF.</td>
</tr>
<tr>
<td>X02AAE</td>
<td>Smallest possible e such that 1.0+e &gt; 1.0. Double precision version is X02AAF.</td>
</tr>
<tr>
<td>X02ABF</td>
<td>Smallest representable positive real number. Double precision version is X02ABF.</td>
</tr>
</tbody>
</table>
X02ACE  Largest representable positive real number. Double precision version is X02ACF.
X02ADE  Ratio of X02ABE to X02AAE. Double precision version is X02ADF.
X02AEE  Largest negative permissible argument for exp. Double precision version is X02AEF.
X02AFE  Returns the value of the largest positive argument permitted for EXP. Double precision version is X02AFF.
X02AGE  Smallest representable positive real number with representable inverse. Double precision version is X02AGF.
X02AHE  Returns the value of the largest positive argument permitted for SIN and COS. Double precision version is X02AHG.
X02BAE  Base of floating-point arithmetic. Double precision version is X02BAG.
X02BBE  Largest representable integer. Double precision version is X02BBF.
X02BCE  Largest positive integer power to which 2.0 can be raised without overflow. Double precision version is X02BCF.
X02BDE  Largest negative integer power to which 2.0 can be raised without underflow. Double precision version is X02BDF.
X02BEE  Maximum number of decimal digits that can be represented. Double precision version is X02BFE.
X02CAE  Estimate of active-set size (on machines withpaged virtual store). Double precision version is X02CAF.
X02DAE  Switch for taking precautions to avoid underflow. Double precision version is X02DAF.

PORT subprogram library

I1MACH  Provides the integer constants required to adapt PORT library programs to individual computers.
R1MACH  Provides the single precision machine-dependent constants required to adapt PORT library programs to individual computers. Double precision version is D1MACH.

R2 : Error checking (e.g., check monotonicity)

PORT subprogram library

MONOI   Test if an integer vector is monotone increasing or decreasing.
MONOR   Test if a real vector is monotone increasing or decreasing. Double precision version is MONOD.
SMONOI  Test if an integer vector is strictly monotone increasing or decreasing.
SMONOR  Test if a real vector is strictly monotone increasing or decreasing. Double precision version is SMONOD.

R3: Error handling

Most debugging aids for the casual programmer are very system dependent. Within large collections of programs such as the commercial libraries certain conventions have been established for reporting errors that occur. For the most part library routines do this by calling a subroutine which performs various actions depending on the severity of the error. Library users do not often need to know about these programs. Programmers who are developing a "package" for use by others may wish to take advantage of some of these utilities, however. One such error handler is XERROR, a public-domain package available in (and used by) CMLIB.

In some cases it is necessary for casual users to change some of the defaults in use by the error handler. For example, every error may cause the program to abort. This is easily changed, and the way that this is done depends upon the particular library in use. For detailed information about the error handlers in the IMSL, NAG, and PORT libraries, one should consult the appropriate library reference manual (see the Library Reference section of GAMS).
### R3: Error handling

<table>
<thead>
<tr>
<th><strong>R3a</strong>: Set criteria for fatal errors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library</strong> (XERROR sublibrary)</td>
</tr>
<tr>
<td><strong>XSETF</strong> Set KONTRL for XERROR, default is =2.</td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td><strong>UERSET</strong> Set message level for IMSL routine UERTST.</td>
</tr>
<tr>
<td><strong>PORT subprogram library</strong></td>
</tr>
<tr>
<td><strong>ENTSRC</strong> Saves current recovery mode status and sets a new one for PORT library programs.</td>
</tr>
<tr>
<td><strong>RETSRC</strong> Test and reset error recovery mode for PORT library programs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>R3b</strong>: Set unit number for error messages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library</strong> (XERROR sublibrary)</td>
</tr>
<tr>
<td><strong>XSETUA</strong> Set up to 5 output unit numbers.</td>
</tr>
<tr>
<td><strong>XSETUN</strong> Set one output unit number.</td>
</tr>
<tr>
<td><strong>IMSL subprogram library</strong></td>
</tr>
<tr>
<td><strong>UGETIO</strong> To retrieve current values and to set new values for input and output unit identifiers.</td>
</tr>
<tr>
<td><strong>NAG subprogram library</strong></td>
</tr>
<tr>
<td><strong>X04AAE</strong> Return or set unit number for error messages for Nag library programs. Double precision version is X04AAF.</td>
</tr>
<tr>
<td><strong>X04ABE</strong> Return or set unit number for advisory messages for NAG library programs. Double precision version is X04ABF.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>R3c</strong>: Other utility programs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CMLIB subprogram library</strong> (XERROR sublibrary)</td>
</tr>
<tr>
<td><strong>NUMXER</strong> Get most current message number.</td>
</tr>
<tr>
<td><strong>XERABT</strong> Terminate run and print traceback. (Requires system dependent programming to execute properly, else just STOPs.).</td>
</tr>
<tr>
<td><strong>XERCLR</strong> Clear current message number.</td>
</tr>
<tr>
<td><strong>XERCTL</strong> Perform special error processing of one message.</td>
</tr>
<tr>
<td><strong>XERDMP</strong> Print error summary and clear tables.</td>
</tr>
<tr>
<td><strong>XERMAX</strong> Set limit of MAX times each message can be printed.</td>
</tr>
<tr>
<td><strong>XERROR</strong> Process a message.</td>
</tr>
<tr>
<td><strong>XERRWV</strong> Process a message with numeric values.</td>
</tr>
<tr>
<td><strong>XGETF</strong> Get current value of KONTRL.</td>
</tr>
<tr>
<td><strong>XGETUA</strong> Get current output unit numbers.</td>
</tr>
<tr>
<td><strong>XGETUN</strong> Get current output unit number.</td>
</tr>
</tbody>
</table>

* Denotes easy-to-use modules
IMSL subprogram library

UERTST Print a message reflecting an error condition.

NAG subprogram library

P01AAE Return value of error indicator, or terminate with an error message. Used exclusively by NAG library programs. Double precision version is P01AAF.

PORT subprogram library

ENTER Save current error recovery mode and storage allocation status for PORT library programs.

EPRINT Print the current error message if the program is in the error state for PORT library programs.

EROFF Turns off the error state for PORT library programs.

LEAVE Restores prior error recovery mode and reset the stack for PORT library programs.

NERROR Provides the current error number for PORT library programs.

SETERR Sets the error indicator and depending on options prints a message and provides a dump for PORT library programs.

R4: Documentation retrieval

This section contains modules whose sole purpose is to provide information about programming conventions within a library or package. For example the UHELP routines give information about conventions in the IMSL library.

January 1984

R4: Documentation retrieval

IMSL subprogram library

UHELP Display methods of obtaining info on IMSL conventions regarding various subjects provide means for individual sites to supply users with site specific info.

UHELP1 Write information regarding IMSL conventions and notation to an output file.

UHELP2 Write information regarding IMSL input and output conventions.

UHELP3 Write information regarding IMSL error detecting facilities.

UHELP4 Write information regarding matrix/vector storage modes used in IMSL subroutines.
S: Software Development Tools

This chapter is designed to contain software tools which ease the process of program development and maintenance. The types of jobs performed by such tools are program transformation (e.g., convert to double precision), static analysis (e.g., flow analysis, interface analysis), and dynamic analysis (e.g., tracing, timing, assertion checking).

Unfortunately, no software development tools are cataloged in the current edition of GAMS. Information about existing software can be found in the reference.

References


S  :  Software development tools

S1  :  Program transformation

S2  :  Static analysis

S3  :  Dynamic analysis
The "module" is the smallest unit cataloged in GAMS. A module may be a Fortran subprogram, a stand-alone program, or a command in an interactive system. In this section we present a summary description of each module in alphabetical order. The following legend is an explanation of the information included in the summaries.

**LEGEND**

Entries in the Module Dictionary take the form

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Form</th>
<th>Class(es)</th>
<th>Usage</th>
<th>On-line doc</th>
<th>Tests</th>
<th>Access</th>
<th>See also</th>
</tr>
</thead>
</table>

where

**Name** is the module name.

**Description** briefly describes the purpose of the module.

**Form** indicates (a) whether the module is portable or proprietary, (b) whether the module is a subprogram, a stand-alone program, or a command in an interactive system, and (c) the name of the library (and sublibrary, if appropriate) containing the module. The use of proprietary software is governed by a licensing agreement, while the use of software designated as portable is unrestricted. (See the legend of the Library Reference, page D0, for further discussion of portability issues.) If the module is a subprogram then the language in which it is coded and the arithmetic precision (single or double) of the computed results is also given.

**Class(es)** lists up to three GAMS classifications which best describe the purpose of the module.

**Usage** indicates how to use the module. This is the call sequence for subprograms and the syntax of the command for interactive systems.

**On-line doc** indicates how to obtain detailed on-line documentation using the Sperry 1100 system at NBS. This usually takes the form "CTS command (EXEC command)."

**Tests** occurs (optionally) only for subprograms. It gives the name of the file element on the Sperry 1100 system at NBS which contains a main program which calls the module. This may be either an example program or a program used to test the integrity of the subprogram.

**Access** indicates how to gain access to the module on the Sperry 1100 system at NBS. For subprograms this is the name of a relocatable library, for stand-alone programs it is the command used to execute the program, and for interactive system commands it is the command used to execute the system.

**See also** (optional) gives the names of other modules in the same library which are almost always used in conjunction with this module.
A02AAE Evaluates the square root of a complex number. Proprietary single precision Fortran subprogram in NAG library. Double precision version is A02AAF. | Class(es): C2 | Usage: CALL A02AAE(XR,XI,YR,YI) | On-line doc: CALL GAMSDOC A02AAE (or @PRT NAG+DOC.A02AAE) | Access: LIB NBS+NAG

A02AAF Evaluates the square root of a complex number. Proprietary double precision Fortran subprogram in NAG library. Single precision version is A02AAE. | Class(es): C2 | Usage: CALL A02AAF(XR,XI,YR,YI) | On-line doc: CALL GAMSDOC A02AAF (or @PRT NAG+DOC.A02AAF) | Access: LIB NBS+NAG

A02ABE Modulus of a complex number. Proprietary single precision Fortran subprogram in NAG library. Double precision version is A02ABF. | Class(es): A4a | Usage: D = A02ABE(XR, XI) | On-line doc: CALL GAMSDOC A02ABE (or @PRT NAG+DOC.A02ABE) | Access: LIB NBS+NAG

A02ABF Modulus of a complex number. Proprietary double precision Fortran subprogram in NAG library. | Class(es): A4b | Usage: D = A02ABF(XR, XI) | On-line doc: CALL GAMSDOC A02ABF (or @PRT NAG+DOC.A02ABF) | Access: LIB NBS+NAG

A02ACE Quotient of two complex numbers. Proprietary single precision Fortran subprogram in NAG library. Double precision version is A02ACF. | Class(es): A4a | Usage: CALL A02ACE(XR, XI, YR, YI, ZR, ZI) | On-line doc: CALL GAMSDOC A02ACE (or @PRT NAG+DOC.A02ACE) | Access: LIB NBS+NAG

A02ACF Quotient of two complex numbers. Proprietary double precision Fortran subprogram in NAG library. | Class(es): A4b | Usage: CALL A02ACF(XR, XI, YR, YI, ZR, ZI) | On-line doc: CALL GAMSDOC A02ACF (or @PRT NAG+DOC.A02ACF) | Access: LIB NBS+NAG


A11VS Standardizes a complex vector to have maximum component of magnitude equal to one. Portable single precision Fortran subprogram in MATHWARE library. | Class(es): D1a6 | Usage: CALL A11VS(N,X,Y,NORM) | On-line doc: @PRT,S MATHWARE+NASHLIB.A11VS | Tests: MATHWARE+NASHLIB.A11-12 | Access: See individual sublibrary documentation


<table>
<thead>
<tr>
<th>Function</th>
<th>Library</th>
<th>Description</th>
<th>Class(es)</th>
<th>Usage</th>
<th>Tests</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOSH</td>
<td>GAMS</td>
<td>Hyperbolic cosine. Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DACOSH.</td>
<td>Class(es): C4c</td>
<td>Usage: R=ACOSH(X)</td>
<td>On-line doc: CALL GAMSDOC ACOSH (or OPRT...</td>
<td></td>
</tr>
</tbody>
</table>
CMLIB+DOC.SUMMARY/FNLIB | Access: LIB NBS+CMLIB

ACOSH Computes hyperbolic arcsine, arccosh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DACOSH. | Class(es): C4c | Usage: X = ACOSH(X) | On-line doc: CALL GAMSDOC ACOSH (or @PRT PORT+DOC.ACOSH) | Access: LIB NBS+PORT

ACCRAN Analysis of one-way classification design data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1a | Usage: CALL ACCRAN(Y,NT,N,TM,WTV,S,GM,NDF,IER) | On-line doc: CALL GAMSDOC ACCRAN (or @PRT IMSL+DOC.ACCRAN) | Access: LIB NBS+IMSL

ACSPEC Computes the series autospectrum from the Fourier transform of the user-supplied autocorrelation function, with user-supplied lag window truncation values. | Portable single precision Fortran program in STATLIB library. | Class(es): L10f | Usage: CALL ACSPEC (RHO, NC, NW, LAGS, Y, N, SCRAT, NS) | On-line doc: CALL GAMSDOC ACSPEC (or @PRT STATLIB+DOC.ACSPEC) | Tests: STATLIB+TEST.DEMO4 | Access: LIB NBS+STATLIB

ACTRST Contrast estimates and sums of squares. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a1a | Usage: CALL ACTRST(T,NR,N,ID,P,IP,Q,SQ) | On-line doc: CALL GAMSDOC ACTRST (or @PRT IMSL+DOC.ACTRST) | Access: LIB NBS+IMSL

AFACN Full factorial plan analysis. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2ab | Usage: CALL AFACN(IOPT, NF, NL, Y, SS, NDF, IER) | On-line doc: CALL GAMSDOC AFACN (or @PRT IMSL+DOC.AFACN) | Access: LIB NBS+IMSL

AFAPT Full factorial plan analysis - easy to use version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2ab | Usage: CALL AFAPT(NF, NL, IJOB, Y, GMEAN, YMEANS, INDEX, STAT, IS, IER) | On-line doc: CALL GAMSDOC AFAPT (or @PRT IMSL+DOC.AFAPT) | Access: LIB NBS+IMSL

AGBACP Analysis of balanced complete experimental design structure data. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1 | Usage: CALL AGBACP(NF, NL, I1A, Y, JW, LST, LOC, SS, NDF, IER) | On-line doc: CALL GAMSDOC AGBACP (or @PRT IMSL+DOC.AGBACP) | Access: LIB NBS+IMSL


AGVACL One or two-sided interval estimate of a variance component. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a2 | Usage: CALL AGVACL(V, FDF, SC, IOP, STAT, IER) | On-line doc: CALL GAMSDOC AGVACL (or @PRT IMSL+DOC.AGVACL) | Access: LIB NBS+IMSL

AGXPM Expected mean squares for balanced complete design models. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L7a2a1 | Usage: CALL AGXPM(IOPT, NF, M, IA, NL, INL, CMS, IORD, IEMS, STAT, IS, ERTM, IE, IER) | On-line doc: CALL GAMSDOC AGXPM (or @PRT IMSL+DOC.AGXPM) | Access: LIB NBS+IMSL

AI Airy function. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DAI. | Class(es): C10d | Usage: R=AI(X) | On-line doc: CALL GAMSDOC AI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

AIE Exponentially scaled Airy function. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DAIE. | Class(es): C10d | Usage: R=AIE(X) | On-line doc: CALL GAMSDOC AIE (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

ALBETA Log Beta. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DLBETA. | Class(es): C7b | Usage: R=ALBETA(A,B) | On-line doc: CALL GAMSDOC ALBETA (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

ALGAMA Evaluate the log [base e] of the absolute value of the gamma function. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): C7a | Usage: X = ALGAMA(Y) | On-line doc: CALL GAMSDOC ALGAMA (or @PRT IMSL+DOC.ALGAMA) | Access: LIB NBS+IMSL

ALGAMS G = \ln \abs{\Gamma(x)}, Log abs gamma with sign of gamma S = \sign \Gamma(x). | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DLGAMS. | Class(es): C7a | Usage: CALL ALGAMS(X,G,S) | On-line doc: CALL GAMSDOC ALGAMS (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

AIL Integral over the range 0 to X of (1/\ln t) dt. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DLI. | Class(es): C5 | Usage: R=ALI(X) | On-line doc: CALL GAMSDOC ALI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

ALNGAM Ln absolute value of Gamma(x). | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DLNGAM. | Class(es): C7a | Usage: R=ALNGAM(X) | On-line doc: CALL GAMSDOC ALNGAM (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

ALNREL Ln(1 + x). | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DLNREL. | Class(es): C4b | Usage: R=ALNREL(X) | On-line doc: CALL GAMSDOC ALNREL (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

ALOG Ln(x). | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. | Class(es): C4b | Usage: R=ALOG(X) | On-line doc: CALL GAMSDOC ALOG (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB
ALOG10  Common logarithm Log to the base 10 of x. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library.
Class(es): C|B | Usage: R=ALOG10(X) | On-line doc: CALL GAMSDOC ALOG10 (or @FRT CMLIB+DOC.SUMMARY/FNLIB)
Access: LIB C|B+CMLIB
ALSQAN  Analysis of Latin square design data. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a2b1a | Usage: CALL ALSQAN(Y;NR,NT,IND,EM,GMS,NDP,IER) | On-line doc: CALL GAMSDOC ALSQAN (or @FRT IMSL+DOC.ALSQAN)
Access: LIB NBS+IMSL
AMEANS  Preparation of a set of unbalanced data for analysis by the method of unweighted means. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a4a | Usage: CALL AMEANS(Y,N,K,YM,HN,SS,IER) | On-line doc: CALL GAMSDOC AMEANS (or @FRT IMSL+DOC.AMEANS)
Access: LIB NBS+IMSL
ANCOV1  Covariance analysis for one-way classification design data. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a3 | Usage: CALL ANCOV1(XY,NOP,IXY,YXY,ITM,SXY,VARB,VART,SS,NDP,WK,IER) | On-line doc: CALL GAMSDOC ANCOV1 (or @FRT IMSL+DOC.ANCOV1)
Access: LIB NBS+IMSL
ANESTE  Analysis of completely nested design data with equal numbers in the subclasses. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a2a1c | Usage: CALL ANESTE(NF,NL,Y,S,NDR,IER) | On-line doc: CALL GAMSDOC ANESTE (or @FRT IMSL+DOC.ANESTE)
Access: LIB NBS+IMSL
ANESTU  Analysis of completely nested design data with unequal numbers in the subclasses. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a2a1c | Usage: CALL ANESTU(NF,NL,Y,GMS,NDP,EMS,TK,WK,IER) | On-line doc: CALL GAMSDOC ANESTU (or @FRT IMSL+DOC.ANESTU)
Access: LIB NBS+IMSL
AORDR  Reordering of the data obtained from a balanced complete experimental design. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a2a1 | Usage: CALL AORDR(NF,NL,JP,Y,JD,YT,WK,IER) | On-line doc: CALL GAMSDOC AORDR (or @FRT IMSL+DOC.AORDR)
Access: LIB NBS+IMSL
AOVONEWAY  Performs a one-way analysis of variance and prints standard results. | Command in MINITAB Proprietary interactive system.
Class(es): L7a1a | Usage: AOVONEWAY on data in C,. . . ,C | On-line doc: HELP AOVONEWAY (in MINITAB) | Tests: MINITAB+TESTSOURCE | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)
ARCBAN  Analysis of two-way classification design data. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a2a1a | Usage: CALL ARCBAN(Y,NR,NE,NT,EM,GMS,NDP,IER) | On-line doc: CALL GAMSDOC ARCBAN (or @FRT IMSL+DOC.ARCBAN)
Access: LIB NBS+IMSL
ARCS  Computes arcsin(x), answer in radians. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DARCOS.
Class(es): C4a | Usage: X = ARCS(X) | On-line doc: CALL GAMSDOC ARCS (or @FRT PORT+DOC.ARCOS)
Access: LIB NBS+PORT
ARIMA  Fits non-seasonal and seasonal models to a time series with p the order of the AR part, d the number of differences, q the order of the MA part and with optional seasonality with period S, AR order P, number of differences D, and MA order Q. Options: starting values, forecasting, save results. | Command in MINITAB Proprietary interactive system.
Class(es): L10e1 | Usage: ARIMA P=K, D=K, Q=K | For residuals S=K, S=Q | For data in C (put residuals in C (put estimated parameters in C))
subcommands CONSTANT or NOCONSTANT | STARTING values in C | FORECAST [forecast origin = K] up to K leads ahead (store forecasts in C [confidence limits in C or C]) | On-line doc: HELP ARIMA (in MINITAB) | Tests: MINITAB+TESTSOURCE | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)
ARIMAE  Performs least squares estimation of the parameters in an ARIMA (Box-Jenkins) model using an adaptation of Pack's code. | Portable single precision Fortran subroutine in STATLIB library.
Class(es): L10e2 | Usage: CALL ARIMAE(Y, N, NAR, NDIFAC, NMA, ND, IOD, INC, NP, IOPA, COEF, RES, NRES, STOPS, STOPCR, MIT, SCRAT, NS) | On-line doc: CALL GAMSDOC ARIMAE (or @FRT STATLIB+DOC.ARIMAE)
Tests: STATLIB+TESTDEMO3 | Access: LIB NBS+STATLIB
ARIMAF  Performs minimum mean square error forecasts for a given (fitted) ARIMA (Box-Jenkins) model, using an adaptation of Pack's code.
Class(es): L10e2 | Usage: CALL ARIMAF(Y, N, NAR, NDIFAC, NMA, ND, IOD, INC, NP, IOPA, COEF, ICI, NP, NTO, ITO, FOR, FORLCL, FORUCL, FORUCL, IRDM, ICDIM, NU, YN, SCRAT, NS) | On-line doc: CALL GAMSDOC ARIMAF (or @FRT STATLIB+DOC.ARIMAF)
Tests: STATLIB+TESTDEMO3 | Access: LIB NBS+STATLIB
ARINS  Computes arcsin(x), answer in radians. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DARSIN.
Class(es): C4a | Usage: X = ARINS(X) | On-line doc: CALL GAMSDOC ARINS (or @FRT PORT+DOC.ARINS)
Access: LIB NBS+PORT
ASINH  Hyperbolic sine. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DASINH.
Class(es): C4e | Usage: R = ASINH(X) | On-line doc: CALL GAMSDOC ASINH (or @FRT CMLIB+DOC.SUMMARY/FNLIB)
Access: LIB NBS+CMLIB
ASINH  Computes hyperbolic arcsine, arcsin(x). | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DASINH.
Class(es): C4e | Usage: X = ASINH(X) | On-line doc: CALL GAMSDOC ASINH (or @FRT CMLIB+DOC.SUMMARY/FNLIB)
Access: LIB NBS+CMLIB
ASINHC  Student-Newman-Keule multiple comparison test. | Proprietary single precision Fortran subroutine in IMSL library.
Class(es): L7a11 | Usage: CALL ASINMC(Y,M,SY,NDP,ALPHA,IR,JW,IER) | On-line doc: CALL GAMSDOC ASINMC (or @FRT IMSL+DOC.ASINMC)
Access: LIB NBS+IMSL
ASPEC  Computes the series autospectrum from the Fourier transform of the autocorrelation function, using the Jenkins and Watts window closing technique. | Portable single precision Fortran subroutine in STATLIB library.
Class(es): 1.0f | Usage: CALL ASPEC(Y, N) | On-line doc: CALL GAMSDOC ASPEC (or @FRT STATLIB+DOC.ASPEC)
Tests: STATLIB+TESTDEMO4 | Access: LIB NBS+STATLIB
ASPECS Computes the series autospectrum from the Fourier transform of the autocorrelation function, with user-supplied lag window truncation values. Portable single precision Fortran subprogram in STATLIB library. Class(es): L10f Usage: CALL ASPECS (Y, N, NW, LAGS, SCRAT, NS) On-line doc: CALL GAMSDOC ASPECS (or @PRT STATLIB+DOC.ASPECS) Tests: STATLIB+TEST.DEM04 Access: LIB NBS*STATLIB

ATANH Arc hyperbolic tangent. Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DATANH. Class(es): C4c Usage: R=ATANH(X) On-line doc: CALL GAMSDOC ATANH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB

ATANH Computes hyperbolic arctangent, arctanh(x). Proprietary single precision Fortran subprogram in PORT library. Double precision version is DATANH. Class(es): C4c Usage: X = ATANH (X) On-line doc: CALL GAMSDOC ATANH (or @PRT PORT+DOC.ATANH) Access: LIB NBS*PORT

AUTOCO Computes the sample autocorrelation coefficient of the data in the input vector X. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L10c Usage: CALL AUTOCO(X,N,IWRITE,XAUTOC) On-line doc: CALL GAMSDOC AUTOCO (or @PRT DATAPAC+DOC.AUTOCO) Access: LIB NBS*DATAPAC
B


B2VAL Evaluates the two-dimensional interpolating function computed by B2INK or one of its partial derivatives. Portable single precision Fortran subprogram in TENSORBS subroutine of CMLIB library. Double precision version is DB2VAL. Class(es): E3 Usage: CALL B2VAL(X,Y,F,DFX,DFY,KX,KY,TF,BCOEF,WORK,IFLAG) On-line doc: CALL GAMSDOC B2VAL (or @PRT CMLIB+DOC.B2VAL/TENSORBS) Tests: CMLIB+TEST-SOURCE.@Q1/TENSORBS, CMLIB+TEST-SOURCE.@Q2/TENSORBS Access: LIB NBS+CMLIB

B3INK Computes parameters of a piecewise-polynomial that interpolates a given set of three-dimensional gridded data. (Use B3VAL to evaluate function.) Portable single precision Fortran subprogram in TENSORBS subroutine of CMLIB library. Double precision version is DB3INK. Class(es): E2a Usage: CALL B3INK(X,Y,Z,F,FCN,LDF,KX,KY,KZ,TF,BCOEF,WORK,IFLAG) On-line doc: CALL GAMSDOC B3INK (or @PRT CMLIB+DOC.B3INK/TENSORBS) Tests: CMLIB+TEST-SOURCE.@Q1/TENSORBS, CMLIB+TEST-SOURCE.@Q2/TENSORBS Access: LIB NBS+CMLIB

B3VAL Evaluates the three-dimensional interpolating function computed by B3INK or one of its partial derivatives. Portable single precision Fortran subprogram in TENSORBS subroutine of CMLIB library. Double precision version is DB3VAL. Class(es): E3 Usage: CALL B3VAL(X,Y,Z,F,DFX,DFY,DFZ,KX,KY,KZ,TF,BCOEF,WORK,IFLAG) On-line doc: CALL GAMSDOC B3VAL (or @PRT CMLIB+DOC.B3VAL/TENSORBS) Tests: CMLIB+TEST-SOURCE.@Q1/TENSORBS, CMLIB+TEST-SOURCE.@Q2/TENSORBS Access: LIB NBS+CMLIB

BAKVEC Forms eigenvectors of certain real non-symmetric tridiagonal matrix from symmetric tridiagonal matrix output from FIGI. Portable single precision Fortran subprogram in EISPACK subroutine of CMLIB library. Class(es): D4c4 Usage: CALL BAKVEC(NM,N,T,E,M,Z,IERR) On-line doc: CALL GAMSDOC BAKVEC (or @PRT CMLIB+DOC.BAKVEC/EISPACK) Access: LIB NBS+CMLIB See also: FIGI

BALANC Balances a general real matrix and isolates eigenvalues whenever possible. Portable single precision Fortran subprogram in EISPACK subroutine of CMLIB library. Class(es): D4c1a Usage: CALL BALANC(NM,N,A,LOW,IGH,SCALE) On-line doc: CALL GAMSDOC BALANC (or @PRT CMLIB+DOC.BALANC/EISPACK) Access: LIB NBS+CMLIB

BALBAK Forms eigenvectors of real general matrix from eigenvectors of matrix output from BAKVEC. Portable single precision Fortran subprogram in EISPACK subroutine of CMLIB library. Class(es): D4c4 Usage: CALL BALBAK(NM,N,LOW,IGH,SCALE,M,Z) On-line doc: CALL GAMSDOC BALBAK (or @PRT CMLIB+DOC.BALBAK/EISPACK) Access: LIB NBS+CMLIB See also: BALANC

BANDR Reduces real symmetric band matrix to symmetric tridiagonal matrix and, optionally, accumulates orthogonal similarity transformations. Portable single precision Fortran subprogram in EISPACK subroutine of CMLIB library. Class(es): D4c1b1 Usage: CALL BANDR(NM,N,MB,JA,A,D,E,Z,ITAB,ITOT) On-line doc: CALL GAMSDOC BANDR (or @PRT CMLIB+DOC.BANDR/EISPACK) Access: LIB NBS+CMLIB


BDCOU1 Tally of observations into a one-way frequency table. Proprietary single precision Fortran subprogram in IMSL library. Class(es): L2b Usage: CALL BDCOU1(X,N,K,DIV,BU,BL,TAB,IER) On-line doc: CALL GAMSDOC BDCOU1 (or @PRT IMSL+DOC.BDCOU1) Access: LIB NBS+IMSL

BDCOU2 Tally of observations into a two-way frequency table. Proprietary single precision Fortran subprogram in IMSL library. Class(es): L2b Usage: CALL BDCOU2(X,Y,N,K1,K2,DIVX,DIVY,XU,YU,YL,IT,TAB,IER) On-line doc: CALL GAMSDOC BDCOU2 (or @PRT IMSL+DOC.BDCOU2) Access: LIB NBS+IMSL

BDLTV Produce letter value summary. Proprietary single precision Fortran subprogram in IMSL library. Class(es): L1a1 Usage: CALL BDLTV(X,N,NUM,SUMRY,IER) On-line doc: CALL GAMSDOC BDLTV (or @PRT IMSL+DOC.BDLTV) Access: LIB NBS+IMSL

BDTAB Computations of frequencies of multivariate data. Proprietary single precision Fortran subprogram in IMSL library. Class(es): L2b Usage: CALL BDTab(X,M,KMAX,NOPT,ICNT,K,ITAB,VECV,IVC,VARV,IV,IDIST,WK,IER) On-line doc: CALL GAMSDOC BDTab (or @PRT IMSL+DOC.BDTab) Access: LIB NBS+IMSL


BDTWI Computations of a two-way frequency table. Proprietary single precision Fortran subprogram in IMSL library. Class(es): L2b Usage: CALL BDTW1(N1,N2,K,ITAB,VECV,IVC,VARV,IV,IM,IRTOT,ICTOT,IALTOT,CHISQ,
P, IER) | On-line doc: CALL GAMSDOC BDTWT (or @PRIMSL+DOC.BDTWT) | Access: LIB NBS+IMSL


**BECOVW** Means and variance-covariance or correlation matrix from data possibly containing missing observations, with weighting on option. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1l2 | Usage: CALL BECOVW (X,IX,W,NBR,TEMP,XM,VCV,IER) | On-line doc: CALL GAMSDOC BECOVW (or @PRIMSL+DOC.BECOVW) | Access: LIB NBS+IMSL


**BECRPS** Moments estimation for grouped data with and without Shepards corrections. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1l3 | Usage: CALL BECRPS (N,C,CI,U,UC,IER) | On-line doc: CALL GAMSDOC BECRPS (or @PRIMSL+DOC.BECRPS) | Access: LIB NBS+IMSL

**BEIGRP** Estimation of basic statistical parameters using grouped data. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1l3 | Usage: CALL BEIGRP (F,Y,K,YLM,WID,IOPT,STAT,IER) | On-line doc: CALL GAMSDOC BEIGRP (or @PRIMSL+DOC.BEIGRP) | Access: LIB NBS+IMSL

**BEIUGR** Estimation of basic statistical parameters using ungrouped data. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1l1 | Usage: CALL BEIUGR (Y,N,IOPT,STAT,IER) | On-line doc: CALL GAMSDOC BEIUGR (or @PRIMSL+DOC.BEIUGR) | Access: LIB NBS+IMSL

**BELBIN** Interval estimate of the parameter p of the binomial distribution. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4l1a2 | Usage: CALL BELBIN (NTRIAL,N,ALPHA,PHAT,FLOWER,UPPER,IER) | On-line doc: CALL GAMSDOC BELBIN (or @PRIMSL+DOC.BELBIN) | Access: LIB NBS+IMSL

**BELPOS** Interval estimate of the parameter lambda of the Poisson distribution. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4l1a16 | Usage: CALL BELPOS (NPOIS,NN,ALPHA,RLAMHT,RLAMLR,RLAMUP,IER) | On-line doc: CALL GAMSDOC BELPOS (or @PRIMSL+DOC.BELPOS) | Access: LIB NBS+IMSL

**BEMDP** Median polish of a two-way table. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L9d | Usage: CALL BEMDP (TAB,IR,IC,IRTAB,MAXIT,WK,IER) | On-line doc: CALL GAMSDOC BEMDP (or @PRIMSL+DOC.BEMDP) | Access: LIB NBS+IMSL

**BEMIRI** Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (in-core version). Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1a1b | Usage: CALL BEMIRI (X,N,M,I,X,MEAN,B,A,S,IBAS,INCID,IER) | On-line doc: CALL GAMSDOC BEMIRI (or @PRIMSL+DOC.BEMIRI) | Access: LIB NBS+IMSL

**BEMIRI** Estimates means, simple regression coefficients, their intercepts, standard errors of the regression coefficients, and standard deviations for arrays which contain missing values (out-of-core version). Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1a1b | Usage: CALL BEMIRI (X,N,M,I,X,MEAN,B,A,S,IBAS,INCID,IER) | On-line doc: CALL GAMSDOC BEMIRI (or @PRIMSL+DOC.BEMIRI) | Access: LIB NBS+IMSL


**BEMNON** Location (mean) inferences using a sample from a normal population with known variance. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4l1a14 | Usage: CALL BEMNON (Y,N,IOPT,CRIT,M,YMN,STAT,IER) | On-line doc: CALL
GAMS: Module Dictionary
January 1984

GAMSDOC BEMON (or @PRT IMSL+DOC.BEMON) | Access: LIB NBS+IMSL

BEMSON Mean and variance inferences using a sample from a normal population. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a14 | Usage: CALL BEMSON (Y,N,IOP,CRT,M,PAR,STAT,NDf,IER) | On-line doc: CALL GAMSDOC BEMSON (or @PRT IMSL+DOC.BEMSON) | Access: LIB NBS+IMSL

BENSON Variance inferences using a sample from a normal population with known mean. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1a14 | Usage: CALL BENSON (Y,N,IOP,CRT,M,VAR,STAT,NDf,IER) | On-line doc: CALL GAMSDOC BENSON (or @PRT IMSL+DOC.BENSON) | Access: LIB NBS+IMSL

BEPAT Mean and variance inferences using samples from each of two normal populations with unequal variances. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1a14 | Usage: CALL BEPAT (Y,N,IOP,CRT,STAT,NDf,IER) | On-line doc: CALL GAMSDOC BEPAT (or @PRT IMSL+DOC.BEPAT) | Access: LIB NBS+IMSL

BEPET Mean and variance inferences using samples from each of two normal populations with equal variances. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4b1a14 | Usage: CALL BEPET (Y,N,IOP,CRT,STAT,NDf,IER) | On-line doc: CALL GAMSDOC BEPET (or @PRT IMSL+DOC.BEPET) | Access: LIB NBS+IMSL

BESCI Modified Bessel functions, I, of complex argument and integer order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBESCI. | Class(es): C10b2 | Usage: CALL BESCI (XR, XI, NB, BR, BI) | On-line doc: CALL GAMSDOC BESCI (or @PRT PORT+DOC.BESCI) | Access: LIB NBS+PORT

BESCJ Bessel functions, J, of complex argument and integer order. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DBESCI. | Class(es): C10a2 | Usage: CALL BESJCJ (XR, XI, NB, BR, BI) | On-line doc: CALL GAMSDOC BESCJ (or @PRT PORT+DOC.BESCI) | Access: LIB NBS+PORT

BESIOE Modified hyperbolic Bessel functions of special integer order scaled by an exponential: \( e^{**abs(x)} \cdot I_1 \) sub 0 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESIOE. | Class(es): C10b1 | Usage: \( R = BESIOE(X) \) | On-line doc: CALL GAMSDOC BESIOE (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESKOE Modified hyperbolic Bessel functions of special integer order scaled by an exponential: \( e^{**abs(x)} \cdot K_1 \) sub 0 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESKOE. | Class(es): C10b1 | Usage: \( R = BESKOE(X) \) | On-line doc: CALL GAMSDOC BESKOE (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESJ Computes an N member sequence of J Bessel fcs, \( J_{x+1}(\alpha) \equiv x \) \( \alpha > 0 \). | Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. | Class(es): C10a8 | Usage: CALL BESJ(X,ALPHA,N,Y2) | On-line doc: CALL GAMSDOC BESJ (or @PRT CMLIB+DOC.BESJ/AMOSLIB) | Tests: CMLIB+TESTSOURCE.BESJ/AMOSLIB | Access: LIB NBS+CMLIB

BESJO Bessel function of the first kind, zero order J 0 sub 0 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESJO. | Class(es): C10a1 | Usage: \( R = BESJO(X) \) | On-line doc: CALL GAMSDOC BESJO (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESKJ Bessel function of the first kind, zero order K 1 sub 1 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESKJ. | Class(es): C10a1 | Usage: \( R = BESKJ(X) \) | On-line doc: CALL GAMSDOC BESKJ (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESK0 Modified (hyberbolic) Bessel function of special integer order Third kind, order 0. K sub 0 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESK0. | Class(es): C10b1 | Usage: \( R = BESK0(X) \) | On-line doc: CALL GAMSDOC BESK0 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESK0E Modified (hyberbolic) Bessel function of special integer order scaled by an exponential. Third kind order 0. \( e^{**x} \cdot K_0 \) sub 0 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESK0E. | Class(es): C10b1 | Usage: \( R = BESK0E(X) \) | On-line doc: CALL GAMSDOC BESK0E (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESK1 Modified (hyperbolic) Bessel function of special integer order; third kind, order one K sub 1 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESK1. | Class(es): C10b1 | Usage: \( R = BESK1(X) \) | On-line doc: CALL GAMSDOC BESK1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESK1E Modified (hyperbolic) Bessel function of special integer order scaled by an exponential. Third kind, order one. \( e^{**x} \cdot K_1 \) sub 1 (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESK1E. | Class(es): C10b1 | Usage: \( R = BESK1E(X) \) | On-line doc: CALL GAMSDOC BESK1E (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESKES Seq of Bessel function scaled by exponential. Abs(n) values are computed for \( n > 0 \), \( i = 0,1,...,n-1 \) for \( n < 0 \), \( i = 0,1,...,n+1 \) \( e^{**x} \cdot K_1 \) sub \( v+1 \) (x). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DBESKES.
is DBSKES. | Class(es): C10b3 | Usage: CALL DBSKES(XNU, X, N, BK) | On-line doc: CALL GAMSDOC DBSKES (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBSKES Sequences of Bessel Functions. Abs(n) values are computed. For n > 0, i = 0, 1...n-1 for n < 0, i = 0, -1...-n+1 K sub v+1 (x). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBSKES. | Class(es): C10b3 | Usage: CALL DBSKES(XNU,X,N,BK) | On-line doc: CALL GAMSDOC DBSKES (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBSREB Biserial and point-biserial correlation coefficients for a qualitatively dichotomized variable and a numerically measurable and classified variable. | Proprietary single precision Fortran program in IMSL library. | Class(es): L4d | Usage: CALL DBSREB (N,A,IA,STAT,IER) | On-line doc: CALL GAMSDOC DBSREB (or @PRT IMSL+DOC.BERSR) | Access: LIB NBS+MSL

DBSRI Modified Bessel functions, I, of real argument and integer order. | Proprietary single precision Fortran program in PORT library. Double precision version is DBSRI. | Class(es): C10bl | Usage: CALL DBSRI (X, NB, B) | On-line doc: CALL GAMSDOC DBSRI (or @PRT PORT+DOC.BERSI) | Access: LIB NBS+PORT

DBSRJ Bessel functions, J, of real argument and integer order. | Proprietary single precision Fortran program in PORT library. Double precision version is DBSRJ. | Class(es): C10a1 | Usage: CALL DBSRJ (X, NB, B) | On-line doc: CALL GAMSDOC DBSRJ (or @PRT PORT+DOC.BERSI) | Access: LIB NBS+PORT

DBSRN Biserial correlation coefficient for a qualitatively dichotomized variable and a numerically or qualitatively classified variable. | Proprietary single precision Fortran program in IMSL library. | Class(es): L4d | Usage: CALL DBSRN (N,A,IA,STAT,IER) | On-line doc: CALL GAMSDOC DBSRN (or @PRT IMSL+DOC.BERSN) | Access: LIB NBS+MSL

BESTA2 Computations of confidence intervals and other basic statistics using output from IMSL routine BESTAT. | Proprietary single precision Fortran program in IMSL library. | Class(es): L4a2 | Usage: CALL BESTA2 (NVAR, 10PT, CLM, CLV, STATS, IS, RNG, CV, CIM, CIV, IER) | On-line doc: CALL GAMSDOC BESTA2 (or @PRT IMSL+DOC.BESTA2) | Access: LIB NBS+MSL | See also: BESTAT

BESTAT Computations of basic univariate statistics from data possibly containing missing values, with weighting on option. | Proprietary single precision Fortran program in IMSL library. | Class(es): L1a2 | Usage: CALL BESTAT (X, IX, WT, NBR, XMISS, STATS, IS, WK, IER) | On-line doc: CALL GAMSDOC BESTAT (or @PRT IMSL+DOC.BESTAT) | Access: LIB NBS+MSL

BESY0 Modified (hyperbolic) Bessel function of special integer order. Second kind, order 0. Y sub y 0 (x). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBESY0. | Class(es): C10a1 | Usage: R = BESY0(X) | On-line doc: CALL GAMSDOC BESY0 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BESY1 Modified (hyperbolic) Bessel function of special integer order. Second kind, order 1. Y sub y 1(x). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBESY1. | Class(es): C10a1 | Usage: R = BESY1(X) | On-line doc: CALL GAMSDOC BESY1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BETA Beta(a,b) = (Gamma(a) * Gamma(b)) / Gamma(a + b). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBETA. | Class(es): C7b | Usage: R = BETA(A,B) | On-line doc: CALL GAMSDOC BETA (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BETAI Incomplete Beta. I sub y sub a (x, b) = B sub y sub x (a, b) / B(a,b). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBETAI. | Class(es): C7b | Usage: R = BETAI(X,A,B) | On-line doc: CALL GAMSDOC BETAI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BETRAN Generates a random sample of size N from the beta distribution with parameters ALPHA and BETA. | Portable single precision Fortran program in DATAPAC library. | Class(es): L8a2 | Usage: CALL BETRAN(N,ALPHA,BETA,ISTART,X) | On-line doc: CALL GAMSDOC BETRAN (or @PRT DATAPAC+DOC.BETRAN) | Access: LIB NBS+DATAPAC

BFQAD Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation. | Portable single precision Fortran program in SPLINE sublibrary of CMLIB library. Double precision version is DBFQAD. | Class(es): H2a2al E3 K6 | Usage: CALL BFQAD(T,T,BCOEF,N,K,ID,X1,X2,TOL,QUAD,IERR,WORK) | On-line doc: CALL GAMSDOC BFQAD (or @PRT CMLIB+DOC.BFQAD/SPLINE and CMLIB+SUMMARY/SPLINE) | Tests: CMLIB+TEST-SOURCE.$F/BPNSPLINE | Access: LIB NBS+CMLIB

BI Bairy function Bi(x). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBI. | Class(es): C10d | Usage: R = BI(X) | On-line doc: CALL GAMSDOC BI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BIE Exponentially scaled Airy function. Bairy Bi(x), x <= 0; exp(-2/3 * x^2/3) * Bi(x), x >= 0. | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBIE. | Class(es): C10d | Usage: R = BIE(X) | On-line doc: CALL GAMSDOC BIE (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BINCDF Computes the cumulative distribution function value at X for the binomial distribution with parameters P and N. | Portable single precision Fortran program in DATAPAC library. | Class(es): L8a1b | Usage: CALL BINCDF(X,P,N,CDF) | On-line doc: CALL GAMSDOC BINCDF (or @PRT DATAPAC+DOC.BINCDF) | Access: LIB NBS+DATAPAC

BINOM Binomial n! / (m! * (n-m)!). | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DBINOM. | Class(es): C1 | Usage: R = BINOM(N, M) | On-line doc: CALL GAMSDOC BINOM (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

BINOMIAL Prints table of binomial probabilities and cumulative distribution function, and optionally saves results. | Command in MINITAB

BINPPF Computes the percent point function value at P for the binomial distribution with parameters PPAR and N. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L6a2b | Usage: CALL BINPPF(P,PPAR,N,PPF) | On-line doc: CALL GAMSDOC BINPPF (or &PRT DATAPAC*DOC.BINPPF) | Access: LIB NBS*DATAPAC

BINRAN Generates a random sample of size N from the binomial distribution with parameters P and NPAR. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L6a2 | Usage: CALL BINRAN(N,P,NPAR,ISTART,X) | On-line doc: CALL GAMSDOC BINRAN (or &PRT DATAPAC*DOC.BINRAN) | Access: LIB NBS*DATAPAC


BQUA D Adaptively integrates functions which have discontinuities in their derivatives. User can specify these points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DQUAD. Class(es): H2a2x1 | Usage: CALL BQUAD (F,N,X,EPS,ANS,REST) | On-line doc: CALL GAMSDOC BQUAD (or &PRT PORT*DOC.BQUAD) | Access: LIB NBS*PORT


BSPLD Evaluates at a given set of points in a specified mesh interval, basis splines and their derivatives. Proprietary single precision Fortran subroutine in PORT library. Double precision version is DBSPLD. Class(es): E3 K6 Usage: CALL BSPLD (K,T,N,X,N,X,LEFT,MD,BX) On-line doc: CALL GAMSDOC BSPLD (or @PRT PORT*DOC.BSPLD) Access: Lib NBS*PORT

BSPLI Obtains the integrals of basis splines, from the left-most mesh point to a specified set of points. Proprietary single precision Fortran subroutine in PORT library. Double precision version is DBSPLI. Class(es): H2a2a1 E3 K6 Usage: CALL BSPLI (K,T,N,X,N,X,LEFT,BX) On-line doc: CALL GAMSDOC BSPLI (or @PRT PORT*DOC.BSPLI) Access: Lib NBS*PORT

BSPLN Evaluates at a given set of points in a specified mesh interval, all the basis splines which are nonzero in that interval. Proprietary single precision Fortran subroutine in PORT library. Double precision version is DBSPLN. Class(es): E3 K6 Usage: CALL BSPLN (K,T,N,X,N,X,LEFT,BX) On-line doc: CALL GAMSDOC BSPLN (or @PRT PORT*DOC.BSPLN) Access: Lib NBS*PORT

BSPPP Converts from the "B" representation of B-spline to piecewise polynomial representation. Portable single precision Fortran subroutine in BSPLINE (or @PRT PORT*DOC.BSPLD) Access: Lib NBS*PORT

BSPPV Calculates value and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X. Portable single precision Fortran subroutine in BSPLINE (or @PRT PORT*DOC.BSPLD) Access: Lib NBS*PORT

BSPVN Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Portable single precision Fortran subroutine in BSPLINE (or @PRT PORT*DOC.BSPLD) Access: Lib NBS*PORT

BSQAD Computes the integral of a B-spline from X1 to X2. The B-spline must be in "B" representation. Portable single precision Fortran subroutine in BSPLINE (or @PRT PORT*DOC.BSPLD) Access: Lib NBS*PORT

BTRIALS Generates pseudo-random sequence of K 0's and 1's, with the probability p of a 1. Command in MINITAB Proprietary interactive system. Class(es): L8a2 Usage: BTRIALS K Bernoulli trials with p = K, put into C On-line doc: HELP BTRIALS (in Minintab) Tests: MINITAB*TEST-SOURCE Access: @XQT NBS**MINITAB.MINITAB (or CALL MINITAB in CTS)

BURAM Finds the best uniform rational approximation to a given function on a specified mesh. Proprietary single precision Fortran subroutine in PORT library. Double precision version is DBURAM. Class(es): K2 Usage: CALL BURAM (NPTS, MESH, FN, M, N, P, Q, DELK) On-line doc: CALL GAMSDOC BURAM (or @PRT PORT*DOC.BURAM) Access: Lib NBS*PORT See also: TCHBP

BURG Computes the coefficients of a finite length causal forward or backward prediction filter and uses both the forward and backward predictions in a symmetric manner to generate the maximum entropy spectrum by means of a Toeplitz recursion. Portable single precision Fortran subroutine in MAXENTROPY (or @PRT PORT*DOC.BURG) Access: Lib NBS*CMLIB

BURN1 Finds the best uniform rational approximation to a given function on a specified mesh, starting from a given initial approximation. Proprietary single precision Fortran subroutine in PORT library. Double precision version is DBURN1. Class(es): K2 Usage: CALL BURN1 (NPTS, MESH, FN, MAXITR, ITOL, M, N, P, Q, DELK) On-line doc: CALL GAMSDOC BURN1 (or @PRT PORT*DOC.BURN1) Access: Lib NBS*PORT See also: TCHBP

BVALU Calculates (at X) the value of the IDERIV-th derivative of the B-spline from its "B" representation. Portable single precision Fortran subroutine in BSPLINE (or @PRT PORT*DOC.BVALU) Access: Lib NBS*PORT

BVSUP Solves a system of linear two-point boundary value problems using superposition, orthogonalization, and variable step integration. Portable single precision Fortran subroutine in BVSUP (or @PRT PORT*DOC.BVSUP) Access: Lib NBS*CMLIB
C02ADE  All zeros of polynomial, Grant and Hitchin's method, complex coefficients.  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C02ADF.  | Class(es): F1ab  | Usage: CALL C02ADE (AR, AC, N, REZ, IMZ, TOL, IFAIL)  | On-line doc: CALL GAMSDOC C02ADE (or @PRT NAG*DOC.C02ADE)  | Access: Lib NBS*NAG

C02ADF  All zeros of polynomial, Grant and Hitchin's method, complex coefficients.  | Proprietary double precision Fortran subroutine in NAG library.  | Class(es): F1ab  | Usage: CALL C02ADF (AR, AC, N, REZ, IMZ, TOL, IFAIL)  | On-line doc: CALL GAMSDOC C02ADF (or @PRT NAG*DOC.C02ADF)  | Access: Lib NBS*NAG

C02AEE  Zero of continuous function of one variable, real coefficients.  | Proprietary single precision Fortran subroutine in NAG library.  | Class(es): F1a  | Usage: CALL C02AEE (A, N, REZ, IMZ, TOL, IFAIL)  | On-line doc: CALL GAMSDOC C02AEE (or @PRT NAG*DOC.C02AEE)  | Access: Lib NBS*NAG

C02AEP  All zeros of polynomial, Grant and Hitchin's method, real coefficients.  | Proprietary single precision Fortran subroutine in NAG library.  | Class(es): F1a  | Usage: CALL C02AEP (A, N, REZ, IMZ, TOL, IFAIL)  | On-line doc: CALL GAMSDOC C02AEP (or @PRT NAG*DOC.C02AEP)  | Access: Lib NBS*NAG

C05ADE  Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm.  | Proprietary single precision Fortran subroutine in NAG library.  | Class(es): F1b  | Usage: CALL C05ADE (A, B, EPS, ETA, F, X, IFAIL)  | On-line doc: CALL GAMSDOC C05ADE (or @PRT NAG*DOC.C05ADE)  | Access: Lib NBS*NAG

C05ADF  Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm.  | Proprietary double precision Fortran subroutine in NAG library. Single precision version is C05AEF.  | Class(es): F1b  | Usage: CALL C05ADF (A, B, EPS, ETA, F, X, IFAIL)  | On-line doc: CALL GAMSDOC C05ADF (or @PRT NAG*DOC.C05ADF)  | Access: Lib NBS*NAG

C05AEG  Zero of continuous function of one variable, from a given starting value, search for interval, Bus and Dekker algorithm.  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C05AGF.  | Class(es): F1b  | Usage: CALL C05AEG (X, H, EPS, ETA, F, A, B, IFAIL)  | On-line doc: CALL GAMSDOC C05AEG (or @PRT NAG*DOC.C05AEG)  | Access: Lib NBS*NAG

C05AGF  Zero of continuous function of one variable, from a given starting value, search for interval, Bus and Dekker algorithm.  | Proprietary double precision Fortran subroutine in NAG library. Single precision version is C05AGE.  | Class(es): F1b  | Usage: CALL C05AGF (X, H, EPS, ETA, F, A, B, IFAIL)  | On-line doc: CALL GAMSDOC C05AGF (or @PRT NAG*DOC.C05AGF)  | Access: Lib NBS*NAG

C05AJE  Zero of continuous function of one variable, from a given starting value, continuation method.  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C05AIF.  | Class(es): F1a  | Usage: CALL C05AJE (X, EPS, ETA, F, NFMAX, IFAIL)  | On-line doc: CALL GAMSDOC C05AJE (or @PRT NAG*DOC.C05AJE)  | Access: Lib NBS*NAG

C05AJF  Zero of continuous function of one variable, from a given starting value, continuation method.  | Proprietary double precision Fortran subroutine in NAG library. Single precision version is C05AJE.  | Class(es): F1a  | Usage: CALL C05AJF (X, EPS, ETA, F, NFMAX, IFAIL)  | On-line doc: CALL GAMSDOC C05AJF (or @PRT NAG*DOC.C05AJF)  | Access: Lib NBS*NAG

C05AVE  Zero of continuous function of one variable, search for interval containing zero(reverse communication).  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C05AVF.  | Class(es): F1b  | Usage: CALL C05AVE (X, FX, H, BOUNDL, BOUNDU, Y, C, IND, IFAIL)  | On-line doc: CALL GAMSDOC C05AVE (or @PRT NAG*DOC.C05AVE)  | Access: Lib NBS*NAG

C05AVF  Zero of continuous function of one variable, search for interval containing zero(reverse communication).  | Proprietary double precision Fortran subroutine in NAG library. Single precision version is C05AVE.  | Class(es): F1b  | Usage: CALL C05AVF (X, FX, H, BOUNDL, BOUNDU, Y, C, IND, IFAIL)  | On-line doc: CALL GAMSDOC C05AVF (or @PRT NAG*DOC.C05AVF)  | Access: Lib NBS*NAG

C05AXE  Zero of continuous function of one variable, from a given starting value, continuation method (reverse communication).  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C05AXF.  | Class(es): F1a  | Usage: CALL C05AXE (X, FX, TOL, IR, SCALE, C, IND, IFAIL)  | On-line doc: CALL GAMSDOC C05AXE (or @PRT NAG*DOC.C05AXE)  | Access: Lib NBS*NAG

C05AXF  Zero of continuous function of one variable, from a given starting value, continuation method (reverse communication).  | Proprietary double precision Fortran subroutine in NAG library. Single precision version is C05AXE.  | Class(es): F1a  | Usage: CALL C05AXF (X, FX, TOL, IR, SCALE, C, IND, IFAIL)  | On-line doc: CALL GAMSDOC C05AXF (or @PRT NAG*DOC.C05AXF)  | Access: Lib NBS*NAG

C05AZE  Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm (reverse communication).  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C05AZF.  | Class(es): F1b  | Usage: CALL C05AZE (X, Y, FX, TOLX, IR, C, IND, IFAIL)  | On-line doc: CALL GAMSDOC C05AZE (or @PRT NAG*DOC.C05AZE)  | Access: Lib NBS*NAG

C05AZF  Zero of continuous function of one variable, in a given interval, Bus and Dekker algorithm (reverse communication).  | Proprietary double precision Fortran subroutine in NAG library. Single precision version is C05AZE.  | Class(es): F1b  | Usage: CALL C05AZF (X, Y, FX, TOLX, IR, C, IND, IFAIL)  | On-line doc: CALL GAMSDOC C05AZF (or @PRT NAG*DOC.C05AZF)  | Access: Lib NBS*NAG

C05NBE  Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of the Powell hybrid method. Derivatives of the function are not required.  | Proprietary single precision Fortran subroutine in NAG library. Double precision version is C05NBF.  | Class(es): F2s  | Usage: CALL C05NBE (FCN, N, X, FVEC, XTOL, WA, LWA, IFAIL)  | On-line doc: CALL GAMSDOC C05NBE (or @PRT NAG*DOC.C05NBE)  | Access: Lib NBS*NAG

C05NBG  Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of the Powell hybrid method. Derivatives of the function are not required.  | Proprietary double precision Fortran subroutine in NAG library. Single precision version...
is C05NBE.  | Class(es): F2a | Usage: CALL C05NBF (FCN,N,X,FVEC,XTOL,WA,I,FAIL) | On-line doc: CALL GAMSDOC C05NBF (or @PRT NAG+DOC.C05NBF) | Access: LIB NBS+NAG

C06NCE  Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. Derivatives of the functions are not required. (Comprehensive version of C05NBE.)  Proprietary single precision Fortran subprogram in NAG library. | Class(es): F2a | Usage: CALL C06NCE (FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSPCN,DIAG,MODE, FACTOR,NPRINT, NFEV,FJAC,LDFJAC,R,LR,QT,W,FAIL) | On-line doc: CALL GAMSDOC C06NCE (or @PRT NAG+DOC.C06NCE) | Access: LIB NBS+NAG

C06NCF  Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. Derivatives of the functions are not required. (Comprehensive version of C05NBF.)  Proprietary double precision Fortran subprogram in NAG library. | Class(es): F2a | Usage: CALL C06NCF (FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSPCN,DIAG,MODE, FACTOR,NPRINT, NFEV,FJAC,LDFJAC,R,LR,QT,W,FAIL) | On-line doc: CALL GAMSDOC C06NCF (or @PRT NAG+DOC.C06NCF) | Access: LIB NBS+NAG

C06PBE  Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06PBF. | Class(es): F2a | Usage: CALL C06PBE (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,WA,I,FAIL) | On-line doc: CALL GAMSDOC C06PBE (or @PRT NAG+DOC.C06PBE) | Access: LIB NBS+NAG

C06PBF  Easy-to-use routine to find a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06PBE. | Class(es): F2a | Usage: CALL C06PBF (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,WA,I,FAIL) | On-line doc: CALL GAMSDOC C06PBF (or @PRT NAG+DOC.C06PBF) | Access: LIB NBS+NAG

C06PCF  Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. (Comprehensive version of C05PCF.)  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06PCF. | Class(es): F2a | Usage: CALL C06PCF (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,DIAG,MODE, FACTOR,NPRINT, NFEV,FJAC,LDFJAC,R,LR,QT,W,FAIL) | On-line doc: CALL GAMSDOC C06PCF (or @PRT NAG+DOC.C06PCF) | Access: LIB NBS+NAG

C06PCF  Finds a zero of a system of N nonlinear functions in N variables by a modification of Powell's hybrid method. The user must provide the Jacobian. (Comprehensive version of C05PCF.)  Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06PCF. | Class(es): F2a | Usage: CALL C06PCF (FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,DIAG,MODE, FACTOR,NPRINT, NFEV,FJAC,LDFJAC,R,LR,QT,W,FAIL) | On-line doc: CALL GAMSDOC C06PCF (or @PRT NAG+DOC.C06PCF) | Access: LIB NBS+NAG

C06ZAE  Checks the user-provided Jacobian prior to use in C06PBE or C06PCF. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ZAF. | Class(es): F3 | Usage: CALL C06ZAE (M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR) | On-line doc: CALL GAMSDOC C06ZAE (or @PRT NAG+DOC.C06ZAE) | Access: LIB NBS+NAG

C06ZAF  Checks the user-provided Jacobian prior to use in C06PBF or C06PCF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06ZAE. | Class(es): F3 | Usage: CALL C06ZAF (M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR) | On-line doc: CALL GAMSDOC C06ZAF (or @PRT NAG+DOC.C06ZAF) | Access: LIB NBS+NAG

C06ACE  Circular convolution of two real vectors of period 2**m. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ACF. | Class(es): D1a10 J2 | Usage: CALL C06ACE (A, B, C, D, N1, M, ND2, SCALE, ABTRAN) | On-line doc: CALL GAMSDOC C06ACE (or @PRT NAG+DOC.C06ACE) | Access: LIB NBS+NAG

C06ACF  Circular convolution of two real vectors of period 2**m. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06ACE. | Class(es): D1a10 J2 | Usage: CALL C06ACF (A, B, C, D, N1, M, ND2, SCALE, ABTRAN) | On-line doc: CALL GAMSDOC C06ACF (or @PRT NAG+DOC.C06ACF) | Access: LIB NBS+NAG

C06ADE  Discrete Fourier transform, FFT algorithm, complex data values within a multi-variable transform. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ADF. | Class(es): J1b | Usage: CALL C06ADE (A, B, N, NV, KS, IW, NIW, W1, NW1, W2, NW2, IFAIL) | On-line doc: CALL GAMSDOC C06ADE (or @PRT NAG+DOC.C06ADE) | Access: LIB NBS+NAG

C06ADF  Discrete Fourier transform, FFT algorithm, complex data values within a multi-variable transform. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06ADE. | Class(es): J1b | Usage: CALL C06ADF (A, B, N, NV, KS, IW, NIW, W1, W1, W2, NW2, IFAIL) | On-line doc: CALL GAMSDOC C06ADF (or @PRT NAG+DOC.C06ADF) | Access: LIB NBS+NAG

C06BAE  Performs Shanks' transformation on a given sequence of real values by means of the Epsilon Algorithm of Wynn. A (possibly unreliable) estimate of the absolute error is also given. An erratic, but often powerful method for accelerating sequences. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06BAF. | Class(es): A7 | Usage: CALL C06BAE (SEQN, NCALL, RESULT, ABSERR, WORK, IWORK, IFAIL) | On-line doc: CALL GAMSDOC C06BAE (or @PRT NAG+DOC.C06BAE) | Access: LIB NBS+NAG

C06BAF  Performs Shanks' transformation on a given sequence of real values by means of the Epsilon Algorithm of Wynn. A (possibly unreliable) estimate of the absolute error is also given. An erratic, but often powerful method for accelerating sequences. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06BAE. | Class(es): A7 | Usage: CALL C06BAF (SEQN, NCALL, RESULT, ABSERR, WORK, IWORK, IFAIL) | On-line doc: CALL GAMSDOC C06BAF (or @PRT NAG+DOC.C06BAF) | Access: LIB NBS+NAG

C06DBE  Sum of a Chebyshev series. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06DBF.
C06DBF  Sum of a Chebyshev series.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06DBE.  
  Class(es): C3a2  |  Usage: D = C06DBE (X, C, N, S)  |  On-line doc: CALL GAMSDOC C06DBE (or @PRT NAG+DOC.C06DBE)  |  Access: LIB NBS+NAG

C06EAE  Discrete Fourier transform, FFT algorithm, no extra workspace, real data values.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06EAF.  
  Class(es): J1a1  |  Usage: CALL C06EAF (X, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06EAE (or @PRT NAG+DOC.C06EAE)  |  Access: LIB NBS+NAG

C06EAF  Discrete Fourier transform, FFT algorithm, no extra workspace, real data values.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is C06EAE.  
  Class(es): J1a1  |  Usage: CALL C06EAF (X, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06EAE (or @PRT NAG+DOC.C06EAE)  |  Access: LIB NBS+NAG

C06EBF  Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (Hermitean sequence).  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06EBF.  
  Class(es): J1a2  |  Usage: CALL C06EBF (X, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06EBF (or @PRT NAG+DOC.C06EBF)  |  Access: LIB NBS+NAG

C06ECF  Discrete Fourier transform, FFT algorithm, no extra workspace, complex data values (general sequence).  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06ECF.  
  Class(es): J1a2  |  Usage: CALL C06ECF (X, Y, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06ECF (or @PRT NAG+DOC.C06ECF)  |  Access: LIB NBS+NAG

C06EAF  Discrete Fourier transform, FFT algorithm, no extra workspace, greater speed, real data values.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06EAF.  
  Class(es): J1a1  |  Usage: CALL C06EAF (X, N, WORK, IFAIL)  |  On-line doc: CALL GAMSDOC C06EAF (or @PRT NAG+DOC.C06EAF)  |  Access: LIB NBS+NAG

C06FFB  Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (Hermitean sequence).  |  Proprietary double precision Fortran subprogram in NAG library. Double precision version is C06FFB.  
  Class(es): J1a2  |  Usage: CALL C06FFB (X, N, WORK, IFAIL)  |  On-line doc: CALL GAMSDOC C06FFB (or @PRT NAG+DOC.C06FFB)  |  Access: LIB NBS+NAG

C06FCE  Discrete Fourier transform, FFT algorithm, extra workspace for greater speed, complex data values (general sequence).  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06FCE.  
  Class(es): J1a2  |  Usage: CALL C06FCE (X, Y, N, WORK, IFAIL)  |  On-line doc: CALL GAMSDOC C06FCE (or @PRT NAG+DOC.C06FCE)  |  Access: LIB NBS+NAG

C06EGB  Complex conjugate of complex data values, Hermitean sequence.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06EGB.  
  Class(es): A4a  |  Usage: CALL C06EGB (X, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06EGB (or @PRT NAG+DOC.C06EGB)  |  Access: LIB NBS+NAG

C06GCB  Complex conjugate of complex data values, Hermitean sequence.  |  Proprietary double precision Fortran subprogram in NAG library.  
  Class(es): A4b  |  Usage: CALL C06GCB (X, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06GCB (or @PRT NAG+DOC.C06GCB)  |  Access: LIB NBS+NAG

C06GCG  Complex conjugate of complex data values, general sequence.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is C06GCG.  
  Class(es): A4a  |  Usage: CALL C06GCG (Y, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06GCG (or @PRT NAG+DOC.C06GCG)  |  Access: LIB NBS+NAG

C06GCF  Complex conjugate of complex data values, general sequence.  |  Proprietary double precision Fortran subprogram in NAG library.  
  Class(es): A4b  |  Usage: CALL C06GCF (Y, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06GCF (or @PRT NAG+DOC.C06GCF)  |  Access: LIB NBS+NAG

C06GBF  Complex conjugate of complex data values, Hermitean sequence.  |  Proprietary double precision Fortran subprogram in NAG library.  
  Class(es): A4b  |  Usage: CALL C06GBF (X, N, IFAIL)  |  On-line doc: CALL GAMSDOC C06GBF (or @PRT NAG+DOC.C06GBF)  |  Access: LIB NBS+NAG

CACOSH  Arc hyperbolic cosine of complex argument, cosh**-1 (x).  |  Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library.  
  Class(es): Cte  |  Usage: C=CACOSH(Z)  |  On-line doc: CALL GAMSDOC CACOSH (or @PRT CMLIB+DOC.SUMMARY/FNLIB)  |  Access: LIB NBS+CMLIB
CMLIB*DOC.SUMMARY/FNLIB | Access: LIB NBS+CMLIB

CARG Argument = theta, in radians of complex number, z = |e^{i*theta} | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): A4a | Usage: R=CARG(Z) | On-line doc: CALL GAMSDOC CARG (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CATAN2 Quadrant correct arctangent of complex arguments, tan^-1 (z1/z2). | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: C=CATAN2(Z1,Z2) | On-line doc: CALL GAMSDOC CATAN2 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CAUCDF Computes the cumulative distribution function value for the Cauchy distribution with median = 0 and 75%point = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5ac1 | Usage: CALL CAUCDF(X,CDF) | On-line doc: CALL GAMSDOC CAUCDF (or @PRT DATAPAC+DOC.CAUCDF) | Access: LIB NBS+DATAPAC

CAUPDF Computes the probability density function value for the Cauchy distribution with median = 0 and 75%point = 1. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5ac1 | Usage: CALL CAUPDF(X,PDF) | On-line doc: CALL GAMSDOC CAUPDF (or @PRT DATAPAC+DOC.CAUPDF) | Access: LIB NBS+DATAPAC

CAUPLT Generates a Cauchy probability plot with median = 0 and 75%point = 1. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3cc4 | Usage: CALL CAUPLT(X,N) | On-line doc: CALL GAMSDOC CAUPLT (or @PRT DATAPAC+DOC.CAUPLT) | Access: LIB NBS+DATAPAC

CAUPPF Computes the percent point function value for the Cauchy distribution with median = 0 and 75%point = 1. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5ac2 | Usage: CALL CAUPPF(P,PPF) | On-line doc: CALL GAMSDOC CAUPPF (or @PRT DATAPAC+DOC.CAUPPF) | Access: LIB NBS+DATAPAC

CAURAN Generates a random sample of size N from the Cauchy distribution with median = 0 and 75%point = 1. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a3 | Usage: CALL CAURAN(N,ISTART,X) | On-line doc: CALL GAMSDOC CAURAN (or @PRT DATAPAC+DOC.CAURAN) | Access: LIB NBS+DATAPAC

CAUSF Computes the sparsity function value for the Cauchy distribution with median = 0 and 75%point = 1. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5ac2 | Usage: CALL CAUSF(P,SP) | On-line doc: CALL GAMSDOC CAUSF (or @PRT DATAPAC+DOC.CAUSF) | Access: LIB NBS+DATAPAC


CBAK2K Forms eigenvectors of complex general matrix from eigenvectors of matrix output from CBAL. Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL CBAK2K(NM,N,LOW,IGH,SCALE,MZR,ZI) | On-line doc: CALL GAMSDOC CBAK2 (or @PRT CMLIB+DOC.CBAK2/EISPACK) | Access: LIB NBS+CMLIB | See also: CBAL

CBAL Balances a complex general matrix and isolates eigenvalues whenever possible. Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c1a | Usage: CALL CBAL(NM,N,AR,AL,LOW,IGH,SCALE) | On-line doc: CALL GAMSDOC CBAL (or @PRT CMLIB+DOC.CBAL/EISPACK) | Access: LIB NBS+CMLIB

CBETA Beta function of complex arguments, = (Gamma(z1)*Gamma(z2))/Gamma(z1+z2). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C7b | Usage: C=CBETA(Z1,Z2) | On-line doc: CALL GAMSDOC CBETA (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CBNRHO Estimation of the bivariate normal correlation coefficient using a contingency table. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L0b L443a | Usage: CALL CBNRHO(AP,IA,IB,IRCY,EPS,RHO,VAR,JER) | On-line doc: CALL GAMSDOC CBNRHO (or @PRT IMSL+DOC.CBNRHO) | Access: LIB NBS+IMSL

CBRT Cube root of real number. Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double
GAMS: Module Dictionary
January 1984

precise version is DCBRT. | Class(es): C2 | Usage: C=CBRT(X) | On-line doc: CALL GAMSDOC CBRT (or @PRT CLMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS*CLMLIB

CCBRT Complex cube root of complex argument. | Class(es): C2 | Usage: R=CBRT(X) | On-line doc: CALL GAMSDOC CBRT (or @PRT CLMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS*CLMLIB


CCORRS Perform cross-correlation analysis between a pair of series. | Portable single precision Fortran subroutine in STATLIB library. | Class(es): L10gl | Usage: CALL CCORRS(Y1, Y2, N, NC, RHO, IRHO) | On-line doc: CALL GAMSDOC CCORRS (or @PRT STATLIB*DOC.CCORRS) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB

CCOSH Hyperbolic cosine of complex argument, cosh z. | Portable single precision Fortran subroutine in FNLIB sublibrary of CLMLIB library. | Class(es): C4c | Usage: C=CCOSH(Z) | On-line doc: CALL GAMSDOC CCOSH (or @PRT CLMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS*CLMLIB

CCOT Cotangent of complex argument, cot z. | Portable single precision Fortran subroutine in FNLIB sublibrary of CLMLIB library. | Class(es): C4a | Usage: C=CCOT(Z) | On-line doc: CALL GAMSDOC CCOT (or @PRT CLMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS*CLMLIB

CCSPEC Computes phase and squared coherence spectra from the Fourier transform of the user-supplied correlation functions, with user-supplied lag window truncation values. | Portable single precision Fortran subroutine in STATLIB library. | Class(es): L10gl | Usage: CALL CCSPEC(RH1, RH2, NC, RHOC, NCC, NW, LAGS, Y1, Y2, N, SCRAT, NS) | On-line doc: CALL GAMSDOC CCSPEC (or @PRT STATLIB*DOC.CSPEC) | Tests: STATLIB*TEST.DEMO4 | Access: LIB NBS*STATLIB

CDADD Adds two complex double precision numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): A6b | Usage: CALL CDADD(A,B,C) | On-line doc: CALL GAMSDOC CDADD (or @PRT PORT+DOC.CDADD) | Access: LIB NBS*PORT

CDCDOT Computes complex precision dot product and adds a scalar. Uses double precision accumulation. | Portable single precision Fortran subroutine in XBLAS sublibrary of CLMLIB library. | Class(es): D1a4 | Usage: C = CCDDOT(N,CB,CX,INCX,CY,INCY) | On-line doc: CALL GAMSDOC CDCDOT (or @PRT CLMLIB+DOC.CDCDOT/XBLAS) | Access: LIB NBS*CLMLIB

CDDIV Divides two complex double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): A4b | Usage: CALL CDIV(A,B,C) | On-line doc: CALL GAMSDOC CDIV (or @PRT PORT+DOC.CDIV) | Access: LIB NBS*PORT

CDIV Divides two complex double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): A4b | Usage: CALL CDIV(A,B,C) | On-line doc: CALL GAMSDOC CDIV (or @PRT PORT+DOC.CDIV) | Access: LIB NBS*PORT

CDEXP Computes exp (s) for complex double precision s. Complex double precision numbers represented as a double precision array of two elements. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): C4b | Usage: CALL CDEXP (X,EXP) | On-line doc: CALL GAMSDOC CDEXP (or @PRT PORT+DOC.CDEXP) | Access: LIB NBS*PORT

CDLOG Computes ln (s) for complex double precision s. Complex double precision numbers represented as a double precision array of two elements. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): C4b | Usage: CALL CDLOG (X,LOG) | On-line doc: CALL GAMSDOC CDLOG (or @PRT PORT+DOC.CDLOG) | Access: LIB NBS*PORT
CDMUL Multiplies two double precision complex numbers. Each is represented by a double precision array of two elements. | Class(es): Ab | Usage: CALL CDMUL (A,B,C) | On-line doc: CALL GAMSDOC CDMUL (or @PORT PORT+DOCD.UDMUL) | Access: LIB NBS+PORT


CDSUB Subtracts two double precision complex numbers. Each is represented by a double precision array of two elements. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): Ab | Usage: CALL CDSUB (A,B,C) | On-line doc: CALL GAMSDOC CDSUB (or @PORT PORT+DOCD.CDSUB) | Access: LIB NBS+PORT

CEIL Finds the smallest integer greater than or equal to x. Input and output are real. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DCEIL. | Class(es): C1 | Usage: X = CEIL (X) | On-line doc: CALL GAMSDOC CEIL (or @PORT PORT+DOCD.CEIL) | Access: LIB NBS+PORT

CENTER Centers data to mean 0, standard deviation 1. Optionally can select location and scale or minimum and maximum. | Command in MINITAB Proprietary interactive system. | Class(es): L2 | Usage: CENTer C,...,C put into C,...,C | [subcommands LOCATION [K,...,K]; SCALE [K,...,K]; MINMAX [K,...,K]]; On-line doc: HELP CENTER (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @QTX NBS+MINITAB,MINITAB (or CALL MINITAB in CTS)

CEXPRL Relative error exponential from first order, (\((e^{**})-1)/x\). | Portable single precision Fortran subprogram in FNLIB subrubby of CMLIB library. | Class(es): C4b | Usage: C=CEXPRL(L) | On-line doc: CALL GAMSDOC CEXPRL (or @PORT CMLIB+DOCD.SUMMARY/FNLB) | Access: LIB NBS+CMLIB


CFPTI Initialize array WSAVE for forward/backward FFT. | Portable single precision Fortran subprogram in FFTPKG subrubby of CMLIB library. | Class(es): -0- | Usage: CALL CFPTI(N,WSAVE) | On-line doc: CALL GAMSDOC CFPTI (or @PORT CMLIB+DOCD.CFPTI/FFTPKG) | Tests: CMLIB+TEST-SOURCE.\$Q/FFTPKG | Access: LIB NBS+CMLIB | See also: CFFTB,CFFT1

CG Computes the eigenvalues and, optionally, the eigenvectors of a complex general matrix. | Portable single precision Fortran subprogram in EISPACK subrubby of CMLIB library. | Class(es): D6x | Usage: CALL CG(NM,N,AR,AL,WR,WI,MA,MZ,ZI,FV1,FV2,FV3,IEERR) | On-line doc: CALL GAMSDOC CG (or @PORT CMLIB+DOCD.CG/EISPACK) | Access: LIB NBS+CMLIB


CGAMR Reciprocal gamma function of complex argument, 1/Gamma(z). | Portable single precision Fortran subprogram in FNLIB subrubby of CMLIB library. | Class(es): C7a | Usage: C=CGAMR(Z) | On-line doc: CALL GAMSDOC CGAMR (or @PORT CMLIB+DOCD.SUMMARY/FNLB) | Access: LIB NBS+CMLIB

CGBDI Compute determinant of complex band matrix from its LU factors. (No provision for computing inverse directly.) Portable single precision Fortran subroutine in LINPACKC library of CMLIB library. Class(es): D2c2 Usage: CALL CGBDI(ABD,LDA,N,ML,MU,IPVT,DET) On-line doc: CALL GAMSDOC CGBDI (or @VRT CMLIB+DOC.CGBDI/LINPACKC) Tests: CMLIB*TEST-SOURCE.$F1/LINPACKC, CMLIB*TEST-SOURCE.$F2/LINPACKC Access: LIB NBS+CMLIB See also: CGCDO CGBFA


CGEDI Compute determinant and/or inverse of general complex matrix from its LU factors. Portable single precision Fortran subroutine in LINPACKC library of CMLIB library. Class(es): D2c1 D3c1 Usage: CALL CGEDI(A,LDA,N,IPVT,DET,WORK,JOB) On-line doc: CALL GAMSDOC CGEDI (or @VRT CMLIB+DOC.CGEDI/LINPACKC) Tests: CMLIB*TEST-SOURCE.$F1/LINPACKC, CMLIB*TEST-SOURCE.$F2/LINPACKC Access: LIB NBS+CMLIB See also: CGEFA CGEVO


CH Computes the eigenvalues and, optionally, eigenvectors of a complex Hermitian matrix. Portable single precision Fortran subroutine in EISPACK library of CMLIB library. Class(es): D4a4 Usage: CALL CH(N,M,N,AR,AI,W,MACT,Z,RI,FV1,FV2,FM1,IERR) On-line doc: CALL GAMSDOC CH (or @VRT CMLIB+DOC.CH/EISPACK) Access: LIB NBS+CMLIB

CHFDV Evaluates a cubic polynomial and its first derivative at an array of points. The polynomial must be given in Hermite form. Portable single precision Fortran subroutine in PCHIP library of CMLIB library. Class(es): E3 Usage: CALL CHFDV(X1,X2,F1,F2,D1,D2,NE,DE,F,E,NEXT,EERR) On-line doc: CALL GAMSDOC CHFDV (or @VRT CMLIB+DOC.CHFDV/PCHIP) Tests: CMLIB*TEST-SOURCE.$Q/PCHIP Access: LIB NBS+CMLIB

CHFEV Evaluates a cubic polynomial given in Hermite form at an array of points. Portable single precision Fortran subroutine in PCHIP library of CMLIB library. Class(es): E3 Usage: CALL CHFEV(X1,X2,F1,F2,D1,D2,NE,DE,F,E,NEXT,EERR) On-line doc: CALL GAMSDOC CHFEV (or @VRT CMLIB+DOC.CHFEV/PCHIP) Tests: CMLIB*TEST-SOURCE.$Q/PCHIP Access: LIB NBS+CMLIB
GAMS: Module Dictionary


CHIDI Uses factorization of complex Hermitian indefinite matrix to compute its inertia determinant, and/or inverse. | Portable single precision Fortran program in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a D3d1a | Usage: CALL CHIDI(A,LDA,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC CHIDI (or @PRT CMLIB+DOC.CHIDI/LINPACKC) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB | See also: CHICO CHIFA


CHKDER Checks gradients of M nonlinear functions in N variables evaluated at a point X for consistency with the functions themselves. A companion subroutine to subprograms SNSL1E and SNL1S. This subroutine can also be used to check the coding of the Jacobian matrix calculation. | Portable single precision Fortran program in SNSL1E sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL CHKDER(M,N,X,FVEC,FJAC,LDJAC,X,FVEC,MODE,ERR) | On-line doc: CALL GAMSDOC CHKDER (or @PRT CMLIB+DOC.CHKDER/SNSL1E) | Tests: CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB


CHPSL Uses factorization of complex Hermitian indefinite matrix stored in packed form to solve systems. | Portable single precision Fortran program in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1a | Usage: CALL CHPSL(AP,N,KPVT,B) | On-line doc: CALL GAMSDOC CHPSL (or @PRT CMLIB+DOC.CHPSL/LINPACKC) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB | See also: CHPCO CHPSL

CHSCDF Computes the cumulative distribution function value for the chi-squared distribution with degrees of freedom parameter = NU. | Portable single precision Fortran program in DATAPAC sublibrary of CMLIB library. | Class(es): L5a1c | Usage: CALL CHSCDF(X,NU,CDF) | On-line doc: CALL GAMSDOC CHSCDF (or @PRT DATAPAC+DOC.CHSCDF) | Access: LIB NBS+DATAPAC

CHSPLT Generates a chi-squared probability plot with integer degrees of freedom parameter value = NU. | Portable single precision Fortran program in DATAPAC library. | Class(es): L3c4c | Usage: CALL CHSPLT(X,NU) | On-line doc: CALL GAMSDOC CHSPLT (or @PRT DATAPAC+DOC.CHSPLT) | Access: LIB NBS+DATAPAC

CHSSPF Computes the percent point function value for the chi-squared distribution with integer degrees of freedom parameter = NU. | Portable single precision Fortran program in DATAPAC library. | Class(es): L5a2c | Usage: CALL CHSSPF(P,NU,PPF) | On-line doc: CALL GAMSDOC CHSSPF (or @PRT DATAPAC+DOC.CHSSPF) | Access: LIB NBS+DATAPAC

CHSRAN Generates a random sample of size N from the chi-squared distribution with integer degrees of freedom parameter = NU. | Portable single precision Fortran program in DATAPAC library. | Class(es): L6a3 | Usage: CALL CHSRAN(N,NU,ISTART,X) | On-line doc: CALL GAMSDOC CHSRAN (or @PRT DATAPAC+DOC.CHSRAN) | Access: LIB NBS+DATAPAC

January 1984
CHU Confluent hypergeometric function, \( U(a,b,x) \). | Portable single precision Fortran subprogram in FNLIB sublibary of CMLIB library. | Double precision version is DCHU. | Class(es): C11 | Usage: \( R=CHU(A,B,X) \) | On-line doc: CALL GAMSDOC CHU (or PORT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CJYHBS Computes J sub 0, J sub 1, Y sub 0, Y sub 1, H sub 0, H sub 1 (Iessel and Struve functions) of complex argument. | Portable single precision Fortran subprogram in AMOSLIB sublibary of CMLIB library. | Class(es): C10a4 C10c | Usage: CALL CJYHBS(Z,KODE,CJ0,CJ1,CY0,CY1,CH0,CH1) | On-line doc: CALL GAMSDOC CJYHBS (or PORT CMLIB+DOC.CJYHBS/AMOSLIB) | Access: LIB NBS+CMLIB

CLBETA Log Beta of complex arguments, in B(z1,z2). | Portable single precision Fortran subprogram in FNLIB sublibary of CMLIB library. | Class(es): C7b | Usage: C=CLBETA(Z1,Z2) | On-line doc: CALL GAMSDOC CLBETA (or PORT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CLNREL Relative error logarithm of complex argument, \( \text{ln}(1+z) \). | Portable single precision Fortran subprogram in FNLIB sublibary of CMLIB library. | Class(es): C4b | Usage: C=CLNREL(Z) | On-line doc: CALL GAMSDOC CLNREL (or PORT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CNBSL Solves the nonsymmetric complex band system of equations using factors previously computed. | Portable single precision Fortran subprogram in LINDRIVES sublibary of CMLIB library. | Class(es): D2c2 | Usage: CALL CNBSL (ABE,LD,A,N,ML,MU,IPVT,B,JOE) | On-


C NVFDI Converts values from one vector to another. Forward loop, double precision into integer. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFDI (N,A,B) | On-line doc: CALL GAMSDOC CNVFDI (or ©PRT PORT+DOC.CNVFDI) | Access: LIB NBS+PORT

C NVFDR Converts values from one vector to another. Forward loop, double precision into real. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFDR (N,A,B) | On-line doc: CALL GAMSDOC CNVFDR (or ©PRT PORT+DOC.CNVFDR) | Access: LIB NBS+PORT

C NVFIC Converts values from one vector to another. Forward loop, integer into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFIC (N,A,B) | On-line doc: CALL GAMSDOC CNVFIC (or ©PRT PORT+DOC.CNVFIC) | Access: LIB NBS+PORT

C NVFID Converts values from one vector to another. Forward loop, integer to double precision. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFID (N,A,B) | On-line doc: CALL GAMSDOC CNVFID (or ©PRT PORT+DOC.CNVFID) | Access: LIB NBS+PORT

C NVFIR Converts values from one vector to another. Forward loop, integer to real. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFIR (N,A,B) | On-line doc: CALL GAMSDOC CNVFIR (or ©PRT PORT+DOC.CNVFIR) | Access: LIB NBS+PORT

C NVFRC Converts values from one vector to another. Forward loop, real into complex. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFRC (N,A,B) | On-line doc: CALL GAMSDOC CNVFRC (or ©PRT PORT+DOC.CNVFRC) | Access: LIB NBS+PORT

C NVFRD Converts values from one vector to another. Forward loop, real into double precision. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFRD (N,A,B) | On-line doc: CALL GAMSDOC CNVFRD (or ©PRT PORT+DOC.CNVFRD) | Access: LIB NBS+PORT

C NVFRI Converts values from one vector to another. Forward loop, real into integer. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): A6a | Usage: CALL CNVFRI (N,A,B) | On-line doc: CALL GAMSDOC CNVFRI (or ©PRT
CODE Codes the elements of the input vector X between 1.0 for minimum, 2.0 for next larger, etc. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): N3ib | Usage: CALL CODE(X,Y) | On-line doc: CALL GAMSDOC CODE (or @PRT DATAPAC+DOC.CODE) | Access: LIB NBS+DATAPAC

COMBAK Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from COMHES. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4c1t Usage: CALL COMBAK(NM,LOW,IGH,AR,AI,INT,M,ZR,ZI) | On-line doc: CALL GAMSDOC COMBAK (or @PRT CMLIB+DOC.COMBAK/EISPACK) | Access: LIB NBS+CMLIB


COMLR Computes eigenvalues of a complex upper Hessenberg matrix using the modified LR method. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4cb2 Usage: CALL COMLR(NM,LOW,IGH,HR,HI,WR,WI,IERR) | On-line doc: CALL GAMSDOC COMLR (or @PRT CMLIB+DOC.COMLR/EISPACK) | Access: LIB NBS+CMLIB

COMLR2 Computes eigenvalues and eigenvectors of complex upper Hessenberg matrix using modified LR method. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4cb2 Usage: CALL COMLR2(NM,LOW,IGH,INT,HR,HI,WR,WI,ZR,ZI,IERR) | On-line doc: CALL GAMSDOC COMLR2 (or @PRT CMLIB+DOC.COMLR2/EISPACK) | Access: LIB NBS+CMLIB

COMQR Computes eigenvalues of complex upper Hessenberg matrix using the QR method. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4cb2 Usage: CALL COMQR(NM,N,LOW,IGH,HR,HI,WR,WI,IERR) | On-line doc: CALL GAMSDOC COMQR (or @PRT CMLIB+DOC.COMQR/EISPACK) | Access: LIB NBS+CMLIB


COPY Copies the contents of the vector X into vector Y. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): D1a5 Usage: CALL COPY(X,Y) | On-line doc: CALL GAMSDOC COPY (or @PRT DATAPAC+DOC.COPY) | Access: LIB NBS+DATAPAC


CORELS Performs correlation analysis of a multivariate random sample with computed results returned to the user. | Portable single precision Fortran subroutine in STATLIB library. | Class(es): L4el14 L11 Usage: CALL CORELS(YM, M, X, Y, SCRT, NS, NPRT, R, IR, PR, IPR) | On-line doc: CALL GAMSDOC CORELS (or @PRT STATLIB+DOC.CORELS) Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

CORR Computes the correlation coefficient between the sets of data in the input vectors X and Y. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L1e1b Usage: CALL CORR(X,Y,N,IWRITE,C) | On-line doc: CALL GAMSDOC CORR (or @PRT DATAPAC+DOC.CORR) | Access: LIB NBS+DATAPAC

CORREL Correlates the Pearson product moment correlation coefficient between two or more pairs of vectors, handles missing values, and optionally saves results. | Portable single precision Fortran subroutine in MINITAB Proprietary interactive system. | Class(es): L1e1b L1e2 Usage: CALL CORRELATION between columns in C... | On-line doc: CALL GAMSDOC CORRELATION (or @PRT MINITAB+DOC.CORRELATION) Tests: MINITAB+TESTSOURCE | Access: QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

CORTB Forms eigenvectors of complex general matrix from eigenvectors of upper Hessenberg matrix output from CORTH. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 Usage: CALL CORTB(NM,LOW,IGH,AR,AI,ORTR,ORTI,M,ZR,ZI) | On-line doc: CALL GAMSDOC CORTB (or @PRT CMLIB+DOC.CORTB/EISPACK) | Access: LIB NBS+CMLIB See also: CORTH

CORTH Reduces complex general matrix to complex upper Hessenberg using unitary similarity transformations. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4cb2 Usage: CALL CORTH(NM,N,LOW,IGH,AR,AI,ORTR,ORTI) | On-line doc: CALL GAMSDOC CORTH (or @PRT CMLIB+DOC.CORTH/EISPACK) | Access: LIB NBS+CMLIB See also: CORTH

COS Cosine of real argument, cos x. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. | Class(es): C4a Usage: R=COS(X) | On-line doc: CALL GAMSDOC COS (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

COSDG Computes the cosine of an angle given in degrees. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. | Class(es): C4a Usage: R=COSDG(X) | On-line doc: CALL GAMSDOC COSDG (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

COSH Computes hyperbolic cosine, cosh(x). | Portable single precision Fortran subroutine in PORT library. | Double precision version is DCOSH. | Class(es): C4c Usage: X = COSH(X) | On-line doc: CALL GAMSDOC COSH (or @PRT PORT+DOC.COSH) | Access: LIB NBS+PORT

COSQB Fast Fourier transform of quarter wave data. Computes a sequence from cosine series representation. Fourier synthesis. | Portable


COSQI Initialize array WSAVE for SUBROUTINES COSQF and COSQB. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL COSQI(N,WSAVE) | On-line doc: CALL GAMSDOC COSQI (or @VRT CMLIB+DOC.COSQI/FFTPKG) | Tests: CMLIB+TEST-SOURCE.$Q/FFTPKG | Access: LIB NBS+CMLIB | See also: COSQB,COSQI

COST Computes discrete (fast) cosine transform of even sequence X[]. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): J1a3 | Usage: CALL COST(N,X,WSAVE) | On-line doc: CALL GAMSDOC COST (or @VRT CMLIB+DOC.COST/FFTPKG) | Tests: CMLIB+TEST-SOURCE.$Q/FFTPKG | Access: LIB NBS+CMLIB | See also: COSTI

COSTI Initialize array WSAVE for SUBROUTINE COST. | Portable single precision Fortran subprogram in FFTPKG sublibrary of CMLIB library. | Class(es): -0- | Usage: CALL COSTI(N,WSAVE) | On-line doc: CALL GAMSDOC COSTI (or @VRT CMLIB+DOC.COSTI/FFTPKG) | Tests: CMLIB+TEST-SOURCE.$Q/FFTPKG | Access: LIB NBS+CMLIB | See also: COST

COT Cotangent of real argument, cot x. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Double precision version is Dcot. | Class(es): C4a | Usage: R=COT(X) | On-line doc: CALL GAMSDOC COT (or @VRT CMLIB+DOC.COT/FFTPKG) | Tests: CMLIB+TEST-SOURCE.$Q/FFTPKG | Access: LIB NBS+CMLIB

COUNT Computes the number of observations between XMIN and XMAX (inclusively) in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1d | Usage: CALL COUNT(X,N,XMIN,XMAX,WRITE,XCOUNT) | On-line doc: CALL GAMSDOC COUNT (or @VRT DATAPAC+DOC.COUNT) | Access: LIB NBS+DATAPAC


CPBDI Uses factorization of complex positive definite band matrix to compute determinant. (No provision for computing inverse.) | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d2 | Usage: CALL CPBDI(ABD,LDA,N,M,DET) | On-line doc: CALL GAMSDOC CPBDI (or @VRT CMLIB+DOC.CPBDI/LINPACKC) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB | See also: CPBCO CPBFA


CPLOT Prints a scatter diagram which condenses as many as 10 lines of plot into one line and trims extreme x- and y-values. Option: form of output. | Command in MINITAB Proprietary interactive system. | Class(es): L3d | Usage: CPLOT y in C vs x in C [, subcommand LINES = K; CHARACTERS = K; XBOUNDS are from K to K; YBOUNDS are from K to K] | On-line doc: HELP CPLOT (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)


CPODI Uses factorization of complex positive definite matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b D3d1b | Usage: CALL CPODI(A, LDA, N, DET, JOB) | On-line doc: CALL GAMSDOC CPODI (or @VRT CMLIB+DOC.CPODI/LINPACKC) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB | See also: CPCO CPOFA


CPLY Finds the zeros of a polynomial with complex coefficients. Uses two real arrays to represent complex numbers, inconvenient. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DCPOLY. | Class(es): F3b1b | Usage: CALL CPLY(DEGREE,OPR,OPZ,ZEROI,ZEROI) | On-line doc: CALL GAMSDOC CPOLY (or @PRT PORT*DOC.CPPLY) | Access: LIB NBS+PORT


CPPDI Uses factorization of complex positive definite matrix stored in packed form to compute determinant and/or inverse. | Portable single precision Fortran subroutine in LINPACKC sublibrary of CMLIB library. | Class(es): D2d1b | Usage: CALL CPPDI(AP,N,DDET,JOBD) | On-line doc: CALL GAMSDOC CPPDI (or @PRT CMLIB*DOC.CPPLC/LINPACKC) | Tests: CMLIB*TEST-SOURCE.$F1/LINPACKC, CMLIB*TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB


CPRQT Computes all the zeros of a general complex polynomial using eigenvalue methods, requiring NxN storage for Nth degree polynomial. | Portable single precision Fortran subroutine in CPRQT70 sublibrary of CMLIB library. | Class(es): F3l1b | Usage: CALL CPRQT70(NDP,E0FF,ROOT1,IERR,WORK) | On-line doc: CALL GAMSDOC CPRQT70 (or @PRT CMLIB*DOC.CPQT70/LINPACKC) | Tests: CMLIB*TEST-SOURCE.$F1/LINPACKC | Access: LIB NBS+CMLIB

CPSI Psi (digamma) of complex argument, Psi(z)=Gamma'(z)/Gamma(z). | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. | Class(es): C7c | Usage: C=PSSI(Z) | On-line doc: CALL GAMSDOC CPSI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CSEVL  Evaluates an n term series of Chebyshev polynomials at a given point.  Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Double precision version is DCSEVL.  Class(es): C3a2 | Usage: R = CSEVL(X,CS,N) | On-line doc: CALL GAMSDOC CSEVL (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB | See also: INTS


CSIDI  Uses factorization of complex symmetric indefinite matrix to compute its determinant and/or inverse.  Portable single precision Fortran subroutine in LINPACKC sublibrary of CMLIB library.  Class(es): D2d1a D3d1a | Usage: CALL CSIDI(A,LDA,N,KPVT,DET,N,DET,IERT,WORK,JOB) | On-line doc: CALL GAMSDOC CSIDI (or @PRT CMLIB+DOC.CSIDI/LINPACKC) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB | See also: CSICO CSIFA


CSINH  Hyperbolic sine of complex argument, sinh z.  Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library.  Class(es): C4c | Usage: C = CSINH(Z) | On-line doc: CALL GAMSDOC CSINH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


CSORT  Sorts a character array in either increasing or decreasing order. Optionally another character array can be carried along.  Portable single precision Fortran subroutine in SORT sublibrary of CMLIB library.  Class(es): N6a2c | Usage: CALL CSORT(X,Y,N,KFLAG,WORK) | On-line doc: CALL GAMSDOC CSORT (or @PRT CMLIB+DOC.CSORT/SORT) | Tests: CMLIB+TEST-SOURCE.$Q/SSORT | Access: LIB NBS+CMLIB


CSPDI  Uses factorization of complex symmetric indefinite matrix stored in packed form to compute its determinants and/or inverse.  Portable single precision Fortran subroutine in LINPACKC sublibrary of CMLIB library.  Class(es): D2d1a D3d1a | Usage: CALL CSPDI(2,P,N,KPVT,DET,DET,INFO,N,INFO,WORK,JOB,2) | On-line doc: CALL GAMSDOC CSPDI (or @PRT CMLIB+DOC.CSPDI/LINPACKC) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC | Access: LIB NBS+CMLIB | See also: CSPCO CSPFA

CSPDI  Finds a numerical approximation to the first derivative at requested points in given input data by using spline interpolation.  Proprietary single precision Fortran subroutine in PORT library. Double precision version is DCSPDI.  Class(es): H1 | Usage: CALL CSPDI(X,Y,N,XX,Y,YPP,NN) | On-line doc: CALL GAMSDOC CSPDI (or @PRT PORT+DOC.CSPDI) | Access: LIB NBS+PORT

CSPFE  Evaluates a cubic spline function which has already been fit to n input data pairs (x,y) by CSPFI.  Proprietary single precision Fortran subroutine in PORT library. Double precision version is DCSPFE.  Class(es): E3 K6 | Usage: CALL CSPFE(X,Y,YP,YPP,X,YPP,NN) | On-line doc: CALL GAMSDOC CSPFE (or @PRT PORT+DOC.CSPFE) | Access: LIB NBS+PORT

CSPFI  Fits a cubic spline function to n input data pairs (x,y) with various endpoint conditions.  Proprietary single precision Fortran subroutine in PORT library. Double precision version is DCSPFI.  Class(es): E1a | Usage: CALL CSPFI(X,Y,N,B,YP,YPP) | On-line doc: CALL GAMSDOC CSPFI (or @PRT PORT+DOC.CSPFI) | Access: LIB NBS+PORT | See also: CSPFE

CSPIN  Interpolates at requested points in given input data using a spline approximation—not a least squares fit.  Proprietary single precision Fortran subroutine in PORT library. Double precision version is DCSPIN.  Class(es): E1a | Usage: CALL CSPIN(X,Y,N,XX,Y,YN) | On-line doc: CALL GAMSDOC CSPIN (or @PRT PORT+DOC.CSPIN) | Access: LIB NBS+PORT
CSPQU  Finds the integral of a function defined by pairs (x,y) of input points. The x's can be unequally spaced. Uses spline interpolation.  |  Proprietary single precision Fortran program in PORT library. Double precision version is DCSPQU.  |  Class(es): H2ab12 |  Usage: CALL CSPQU(X,Y,N,XLOW, XHIGH,ANS)  |  On-line doc: CALL GAMSDOC CSPQU (or @PRT PORT+DOC.CSPQU)  |  Access: Lib NBS+PORT

CSPSL  Uses factorization of complex symmetric indefinite matrix stored in packed form to solve systems.  |  Portable single precision Fortran program in LINPACKC sublibrary of CMLIB library.  |  Class(es): D2d1a |  Usage: CALL CSPSL(AP,N,KPVT,B)  |  On-line doc: CALL GAMSDOC CSPSL (or @PRT CMLIB+DOC.CSPSL/LINPACKC)  |  Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC  |  Access: Lib NBS+CMLIB  |  See also: CSPCO CSPFA


CTABLE  Prints a coded two-way table, each cell of which is coded with one character for features MAXIMUM, MINIMUM, or EXTREME, and codes for values between the hinges, between hinges and inner fences, between inner and outer fences, and beyond the outer fences.  |  Command in MINITAB Proprietary interactive system.  |  Class(es): L3d |  Usage: CTAble data in C, row levels in C, columns levels in C [or subcommand MAXIMUM or MINIMUM or EXTREME];  |  On-line doc: HELP CTAble (in Minitab);  |  Tests: MINITAB+TEST-SOURCE.  |  Access: @QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

CTAN  Tangent of complex argument, tan z.  |  Portable single precision Fortran program in FNLIB sublibrary of CMLIB library.  |  Class(es): C4a |  Usage: C=CTAN(Z)  |  On-line doc: CALL GAMSDOC CTAN (or @PRT CMLIB+DOC.SUMMARY/FNLIB)  |  Access: Lib NBS+CMLIB

CTANH  Hyperbolic tangent of complex argument, tanh z.  |  Portable single precision Fortran program in FNLIB sublibrary of CMLIB library.  |  Class(es): C4c |  Usage: C=CTANH(Z)  |  On-line doc: CALL GAMSDOC CTANH (or @PRT CMLIB+DOC.SUMMARY/FNLIB)  |  Access: Lib NBS+CMLIB


CTRBYC  Analysis of a contingency table.  |  Proprietary single precision Fortran program in IMSL library.  |  Class(es): L9b |  Usage: CALL CTRBYC(A,IRC,JRC,IR,IC,STAT,IER)  |  On-line doc: CALL GAMSDOC CTRBYC (or @PRT IMSL+DOC.CTRBYC)  |  Access: Lib NBS+IMSL


CTRDI  Computes determinant and/or inverse of complex triangular matrix.  |  Portable single precision Fortran program in LINPACKC sublibrary of CMLIB library.  |  Class(es): D2c3 D3c |  Usage: CALL CTRDI(T,LDT,N,DET,JOB,INFO)  |  On-line doc: CALL GAMSDOC CTRDI (or @PRT CMLIB+DOC.CTRDI/LINPACKC)  |  Tests: CMLIB+TEST-SOURCE.$F1/LINPACKC, CMLIB+TEST-SOURCE.$F2/LINPACKC  |  Access: Lib NBS+CMLIB  |  See also: CTRCO

D

**D01AHE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for well-behaved integrands. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AHF. | Class(es): H2a1a1 | Usage: D = D01AHE (A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL) | On-line doc: CALL GAMSDOC D01AHE (or @PRT NAG*DOC.D01AHE) | Access: LIB NBS*NAG

**D01AHF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for well-behaved integrands. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AHE. | Class(es): H2a1a1 | Usage: D = D01AHF (A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL) | On-line doc: CALL GAMSDOC D01AHF (or @PRT NAG*DOC.D01AHF) | Access: LIB NBS*NAG

**D01AJE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval allowing for badly-behaved integrands. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AJF. | Class(es): H2a1a1 | Usage: CALL D01AJE (A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01AJE (or @PRT NAG*DOC.D01AJE) | Access: LIB NBS*NAG

**D01AHE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for badly-behaved integrands. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AHF. | Class(es): H2a1a1 | Usage: CALL D01AHE (A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL) | On-line doc: CALL GAMSDOC D01AHE (or @PRT NAG*DOC.D01AHE) | Access: LIB NBS*NAG

**D01AHF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for badly-behaved integrands. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AHE. | Class(es): H2a1a1 | Usage: D = D01AHF (A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL) | On-line doc: CALL GAMSDOC D01AHF (or @PRT NAG*DOC.D01AHF) | Access: LIB NBS*NAG

**D01AJE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval allowing for badly-behaved integrands. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AJF. | Class(es): H2a1a1 | Usage: CALL D01AJE (A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01AJE (or @PRT NAG*DOC.D01AJE) | Access: LIB NBS*NAG

**D01AJF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval suitable for badly-behaved integrands. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AJE. | Class(es): H2a2a1 | Usage: CALL D01AJF (A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01AJF (or @PRT NAG*DOC.D01AJF) | Access: LIB NBS*NAG

**D01AKE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, method suitable for oscillating functions. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AKF. | Class(es): H2a2a1 | Usage: CALL D01AKE (A, B, EPSABS, EPSREL, RESULT, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01AKE (or @PRT NAG*DOC.D01AKE) | Access: LIB NBS*NAG

**D01AKF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, method suitable for oscillating functions. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AKE. | Class(es): H2a2a1 | Usage: CALL D01AKF (A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWi, LIFAIL) | On-line doc: CALL GAMSDOC D01AKF (or @PRT NAG*DOC.D01AKF) | Access: LIB NBS*NAG

**D01ALE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, allowing for singularities at user-specified points. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01ALF. | Class(es): H2a2a1 | Usage: CALL D01ALE (A, B, NPTS, POINTS, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01ALE (or @PRT NAG*DOC.D01ALE) | Access: LIB NBS*NAG

**D01ALF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, allowing for singularities at user-specified points. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01ALE. | Class(es): H2a2a1 | Usage: CALL D01ALF (A, B, NPTS, POINTS, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01ALF (or @PRT NAG*DOC.D01ALF) | Access: LIB NBS*NAG

**D01AME** Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AMF. | Class(es): H2a2a1 H2a4a1 | Usage: CALL D01AME (F, BOUND, INF, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01AME (or @PRT NAG*DOC.D01AME) | Access: LIB NBS*NAG

**D01AMF** Quadrature for one-dimensional integrals, adaptive integration of a function over an infinite or semi-infinite interval. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AME. | Class(es): H2a2a1 H2a4a1 | Usage: CALL D01AMF (F, BOUND, INF, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, LIFAIL) | On-line doc: CALL GAMSDOC D01AMF (or @PRT NAG*DOC.D01AMF) | Access: LIB NBS*NAG

**D01ANE** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $\cos(wx)$ or $\sin(wx)$. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01ANF. | Class(es): H2a2a1 | Usage: CALL D01ANE (G, A, B, OMEGA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI IFAIL) | On-line doc: CALL GAMSDOC D01ANE (or @PRT NAG*DOC.D01ANE) | Access: LIB NBS*NAG

**D01ANF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $\cos(wx)$ or $\sin(wx)$. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01ANE. | Class(es): H2a2a1 | Usage: CALL D01ANF (G, A, B, OMEGA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI IFAIL) | On-line doc: CALL GAMSDOC D01ANF (or @PRT NAG*DOC.D01ANF) | Access: LIB NBS*NAG

**D01APE** Adaptive integration of a function of one variable over a finite interval with weight function with algebraic-logarithmic endpoint singularities. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01APF. | Class(es): H2a2a1 | Usage: CALL D01APE (G, A, B, ALFA, BETA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI, IFAIL) | On-line doc: CALL GAMSDOC D01APE (or @PRT NAG*DOC.D01APE) | Access: LIB NBS*NAG

**D01APP** Adaptive integration of a function of one variable over a finite interval with weight function with algebraic-logarithmic endpoint singularities. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01APE. | Class(es): H2a2a1 | Usage: CALL D01APP (G, A, B, ALFA, BETA, KEY, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI IFAIL) | On-line doc: CALL GAMSDOC D01APP (or @PRT NAG*DOC.D01APP) | Access: LIB NBS*NAG

**D01AQF** Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $\frac{1}{(x-c)}$ (Hilbert transform). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01AQF. | Class(es): H2a2a1 | Usage: CALL D01AQF (G, A, B, C, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IWI IFAIL) | On-line doc: CALL GAMSDOC D01AQF (or @PRT NAG*DOC.D01AQF) | Access: LIB NBS*NAG
D01AQF Quadrature for one-dimensional integrals, adaptive integration of a function over a finite interval, weight function $1/(x-c)$ (Hilbert transform). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01AQE. | Class(es): H2a1a1 J4 | Usage: CALL D01AQF (G, A, B, C, EPSABS, EPSREL, RESULT, ABERR, W, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC D01AQF (or @PRT NAG+DOC.D01AQF) | Access: Lib NBS+NAG

D01ARE Computes definite integral over a finite range to a specified relative or absolute accuracy, using Patterson's method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01ARF. | Class(es): H2a1a1 | Usage: CALL D01ARE (A, B, F, FUN, RELACC, ABSCACC, MAXRUL, IPARM, ACC, ANS, ALPHA, IFAIL) | On-line doc: CALL GAMSDOC D01ARE (or @PRT NAG+DOC.D01ARE) | Access: Lib NBS+NAG

D01ARF Computes definite integral over a finite range to a specified relative or absolute accuracy, using Patterson's method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01ARE. | Class(es): H2a1a1 | Usage: CALL D01ARF (A, B, F, FUN, RELACC, ABSCACC, MAXRUL, IPARM, ACC, ANS, ALPHA, IFAIL) | On-line doc: CALL GAMSDOC D01ARF (or @PRT NAG+DOC.D01ARF) | Access: Lib NBS+NAG

D01BAE Quadrature for one-dimensional integrals, Gaussian rule-evaluation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BAF. | Class(es): H2a1a2 | Usage: CALL D01BAE (D01XXX, A, B, N, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01BAE (or @PRT NAG+DOC.D01BAE) | Access: Lib NBS+NAG

D01BAF Quadrature for one-dimensional integrals, Gaussian rule-evaluation. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BAE. | Class(es): H2a1a2 | Usage: CALL D01BAF (D01XXX, A, B, N, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01BAF (or @PRT NAG+DOC.D01BAF) | Access: Lib NBS+NAG

D01BBE Weights and abscissae for Gaussian quadrature rules, restricted choice of rule, using pre-computed weights and abscissae. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BBF. | Class(es): H2c | Usage: CALL D01BBE (D01XXX, A, B, ITYPE, N, WEIGHT, ABSCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BBE (or @PRT NAG+DOC.D01BBE) | Access: Lib NBS+NAG

D01BBF Weights and abscissae for Gaussian quadrature rules, restricted choice of rule, using pre-computed weights and abscissae. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BBE. | Class(es): H2c | Usage: CALL D01BBF (D01XXX, A, B, ITYPE, N, WEIGHT, ABSCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BBF (or @PRT NAG+DOC.D01BBF) | Access: Lib NBS+NAG

D01BCE Weights and abscissae for Gaussian quadrature rules, more general choice of rule calculating the weights and abscissae. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BCF. | Class(es): H2c | Usage: CALL D01BCE (ITYPE, A, B, C, D, N, WEIGHT, ABSCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BCE (or @PRT NAG+DOC.D01BCE) | Access: Lib NBS+NAG

D01BCF Weights and abscissae for Gaussian quadrature rules, more general choice of rule calculating the weights and abscissae. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BCE. | Class(es): H2c | Usage: CALL D01BCF (ITYPE, A, B, C, D, N, WEIGHT, ABSCIS, IFAIL) | On-line doc: CALL GAMSDOC D01BCF (or @PRT NAG+DOC.D01BCF) | Access: Lib NBS+NAG

D01BDE Quadrature for one-dimensional integrals, non-adaptive integration over a finite interval. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01BDF. | Class(es): H2a1a1 | Usage: CALL D01BDE (F, A, B, EPSABS, EPSREL, RESULT, ABERR) | On-line doc: CALL GAMSDOC D01BDE (or @PRT NAG+DOC.D01BDE) | Access: Lib NBS+NAG

D01BDF Quadrature for one-dimensional integrals, non-adaptive integration over a finite interval. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01BDE. | Class(es): H2a1a1 | Usage: CALL D01BDF (F, A, B, EPSABS, EPSREL, RESULT, ABERR) | On-line doc: CALL GAMSDOC D01BDF (or @PRT NAG+DOC.D01BDF) | Access: Lib NBS+NAG

D01DAE Quadrature for two-dimensional integrals over a finite region. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01DAF. | Class(es): H2b2a1 | Usage: CALL D01DAE (YA, YB, PHI1, PHI2, ABSCACC, ANS NPTS, IFAIL) | On-line doc: CALL GAMSDOC D01DAE (or @PRT NAG+DOC.D01DAE) | Access: Lib NBS+NAG

D01DAF Quadrature for two-dimensional integrals over a finite region. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01DAE. | Class(es): H2b2a1 | Usage: CALL D01DAF (YA, YB, PHI1, PHI2, ABSCACC, ANS NPTS, IFAIL) | On-line doc: CALL GAMSDOC D01DAF (or @PRT NAG+DOC.D01DAF) | Access: Lib NBS+NAG

D01FAE Quadrature for multi-dimensional integrals over a hyper-rectangle, Monte Carlo method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FAF. | Class(es): H2b2a1 | Usage: CALL D01FAE (NDIM, AA, BB, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FAE (or @PRT NAG+DOC.D01FAE) | Access: Lib NBS+NAG

D01FAP Quadrature for multi-dimensional integrals over a hyper-rectangle, Monte Carlo method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FAE. | Class(es): H2b2a1 | Usage: CALL D01FAP (NDIM, AA, BB, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FAP (or @PRT NAG+DOC.D01FAP) | Access: Lib NBS+NAG

D01FBE Quadrature for multi-dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FBB. | Class(es): H2b2a2 | Usage: D = D01FBE (NDIM, NPTVEC, LWA, WEIGHT, ABSCIS, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01FBE (or @PRT NAG+DOC.D01FBE) | Access: Lib NBS+NAG
D01FBF Quadrature for multi-dimensional integrals over a hyper-rectangle, Gaussian rule-evaluation. Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FBE. [Class(es): H2b1a2 | Usage: D = D01FBF (NDIM, NPTVEC, LWA, WEIGHT, ABCSIS, FUN, IFAIL) | On-line doc: CALL GAMSDOC D01FBF (or @PRT NAG *DOC.D01FBF) | Access: LIB NASA NAG]

D01FCE Quadrature for multi-dimensional integrals over a hyper-rectangle, adaptive method. Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FCF. [Class(es): H2b1a1 | Usage: CALL D01FCE (NDIM, A, B, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FCE (or @PRT NAG *DOC.D01FCE) | Access: LIB NASA NAG]

D01FCF Quadrature for multi-dimensional integrals over a hyper-rectangle, adaptive method. Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FCF. [Class(es): H2b1a1 | Usage: CALL D01FCF (NDIM, A, B, MINPTS, MAXPTS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINVAL, IFAIL) | On-line doc: CALL GAMSDOC D01FCF (or @PRT NAG *DOC.D01FCF) | Access: LIB NASA NAG]

D01FDE Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. Proprietary single precision Fortran subprogram in NAG library. Double precision version is D01FDF. [Class(es): H2b1a2 | Usage: CALL D01FDE (NDIM, FUNCTN, SIGMA, REGION, LIMIT, RO, U, RESULT, NCALS, IFAIL) | On-line doc: CALL GAMSDOC D01FDE (or @PRT NAG *DOC.D01FDE) | Access: LIB NASA NAG]

D01FDF Calculates an approximation to a definite integral in up to 30 dimensions, using the method of Sag and Szekeres. The region of integration is an n-sphere, or by built-in transformation via the unit n-cube, any product region. Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01FDF. [Class(es): H2b1a2 | Usage: CALL D01FDF (NDIM, FUNCTN, SIGMA, REGION, LIMIT, RO, U, RESULT, NCALS, IFAIL) | On-line doc: CALL GAMSDOC D01FDF (or @PRT NAG *DOC.D01FDF) | Access: LIB NASA NAG]

D01GAE Quadrature for one-dimensional integrals, integration of a function defined by data values only. Proprietary single precision Fortran program in NAG library. Double precision version is D01GAF. [Class(es): H2a1b2 | Usage: CALL D01GAE (X, Y, N, ANS, ER, IFAIL) | On-line doc: CALL GAMSDOC D01GAE (or @PRT NAG *DOC.D01GAE) | Access: LIB NASA NAG]

D01GAF Quadrature for one-dimensional integrals, integration of a function defined by data values only. Proprietary double precision Fortran program in NAG library. Single precision version is D01GAE. [Class(es): H2a1b2 | Usage: CALL D01GAF (X, Y, N, ANS, ER, IFAIL) | On-line doc: CALL GAMSDOC D01GAF (or @PRT NAG *DOC.D01GAF) | Access: LIB NASA NAG]

D01GBE Calculates an approximation to the integral of a function over a hyper-rectangular region, using a Monte-Carlo method. An approximate relative estimate is also returned. Suitable for low accuracy work. Proprietary single precision Fortran program in NAG library. Double precision version is D01GBF. [Class(es): H2b1a1 | Usage: CALL D01GBE (NDIM, A, B, MINCLS, MAXCLS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINEST, IFAIL) | On-line doc: CALL GAMSDOC D01GBE (or @PRT NAG *DOC.D01GBE) | Access: LIB NASA NAG]

D01GBF Calculates an approximation to the integral of a function over a hyper-rectangular region, using a Monte-Carlo method. An approximate relative estimate is also returned. Suitable for low accuracy work. Proprietary double precision Fortran program in NAG library. Single precision version is D01GBE. [Class(es): H2b1a1 | Usage: CALL D01GBF (NDIM, A, B, MINCLS, MAXCLS, FUNCTN, EPS, ACC, LENWRK, WRKSTR, FINEST, IFAIL) | On-line doc: CALL GAMSDOC D01GBF (or @PRT NAG *DOC.D01GBF) | Access: LIB NASA NAG]

D01GCE Calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov-Conroy number theoretic method. Returns a simple error estimate by repeating the computation with a different (randomised) set of points. Proprietary single precision Fortran program in NAG library. Double precision version is D01GCF. [Class(es): H2b1a2 | Usage: CALL D01GCE (NDIM, FUNCTN, REGION, NPTS, VK, VRAND, ITRANS, RES, ERR, IFAIL) | On-line doc: CALL GAMSDOC D01GCE (or @PRT NAG *DOC.D01GCE) | Access: LIB NASA NAG | See also: D01GYE, D01GZE]

D01GCF Calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov-Conroy number theoretic method. Returns a simple error estimate by repeating the computation with a different (randomised) set of points. Proprietary double precision Fortran program in NAG library. Single precision version is D01GCE. [Class(es): H2b1a2 | Usage: CALL D01GCF (NDIM, FUNCTN, REGION, NPTS, VK, VRAND, ITRANS, RES, ERR, IFAIL) | On-line doc: CALL GAMSDOC D01GCF (or @PRT NAG *DOC.D01GCF) | Access: LIB NASA NAG | See also: D01GYF, D01GZF]

D01JAE Attempts to evaluate an integral over an n-dimensional sphere (n=2,3,4), to a user specified absolute or relative accuracy, by means of a modified Sag-Szekeres method. Can handle singularities on the surface or at the center of the sphere. Returns an error estimate. Proprietary single precision Fortran program in NAG library. Double precision version is D01JAF. [Class(es): H2b2b1 | Usage: CALL D01JAE (P, NDIM, RADIUS, EPSA, EPSR, METHOD, ICOORD, RESULT, ESTERR, NEVALS, IFAIL) | On-line doc: CALL GAMSDOC D01JAE (or @PRT NAG *DOC.D01JAE) | Access: LIB NASA NAG]

D01JAF Attempts to evaluate an integral over an n-dimensional sphere (n=2,3,4), to a user specified absolute or relative accuracy, by means of a modified Sag-Szekeres method. Can handle singularities on the surface or at the center of the sphere. Returns an error estimate. Proprietary double precision Fortran program in NAG library. Single precision version is D01JAE. [Class(es): H2b2b1 | Usage: CALL D01JAF (P, NDIM, RADIUS, EPSA, EPSR, METHOD, ICOORD, RESULT, ESTERR, NEVALS, IFAIL) | On-line doc: CALL GAMSDOC D01JAF (or @PRT NAG *DOC.D01JAF) | Access: LIB NASA NAG]

D01PAE Returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation. Proprietary single precision Fortran program in NAG library. Double precision version is D01PAF. [Class(es): H2b2b2 | Usage: CALL D01PAE (NDIM, VERTEX, IV1, IV2, FUNCTN, MINORD, MAXORD, FINVLS, ESTERR, IFAIL) | On-line doc: CALL GAMSDOC D01PAE (or @PRT NAG *DOC.D01PAE) | Access: LIB NASA NAG]
DO1PAF Returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation. Proprietary double precision Fortran subprogram in NAG library. Single precision version is D01PAE. Class(es): H28b2 | Usage: CALL DO1PAF(NDIM, VERTEX, I1, I2, FUNCTN, MINORD, MAXORD, FINVLS, ESTERR, IFAIL) | On-line doc: CALL GAMSDOC DO1PAF (or OPRT NAG*DOC.DO1PAF) | Access: LIB NBS*NAG


DO2AGF Solves two point boundary value problem for a system of ODEs using initial value techniques. Proprietary single precision Fortran subprogram in NAG library. Single precision version is D02AGE. Class(es): libb2 libb3 | Usage: CALL DO2AGF (H, ERROR, PARERR, PARAM, C, N, Ni, M, AUX, BCAUX, RAUX, PRSOL, MAT, COPY, WSPEC, WSPA, WSPAC, IFAIL) | On-line doc: CALL GAMSDOC DO2AGF (or OPRT NAG*DOC.DO2AGF) | Access: LIB NBS*NAG


DO2CAE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range. Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CAF. Class(es): Ilia1b | Usage: CALL D02CAE (X, XEND, N, Y, TOL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CAE (or OPRT NAG*DOC.DO2CAE) | Access: LIB NBS*NAG


DO2CBE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range with
January 1984

GAMS: Module Dictionary C 31

intermediate output. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CBF. | Class(es): 11a | Usage: CALL D02CBE (X, XEND, N, Y, TOL, IRELAB, FCN, OUTPUT, W, IFAIL) | On-line doc: CALL GAMSDOC D02CBE (or @PRT NAG*DOC.D02CBE) | Access: LIB NBS*NAG

D02CBF Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, over a range with intermediate output. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CBF. | Class(es): 11a | Usage: CALL D02CBF (X, XEND, N, Y, TOL, IRELAB, FCN, OUTPUT, W, IFAIL) | On-line doc: CALL GAMSDOC D02CBF (or @PRT NAG*DOC.D02CBF) | Access: LIB NBS*NAG

D02CGE Initial value problems for system of O.D.E.s, (simple driver) variable order and step Adams method, until a component of the solution attains a given value. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CGF. | Class(es): 11a | Usage: CALL D02CGE (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CGE (or @PRT NAG*DOC.D02CGE) | Access: LIB NBS*NAG

D02CGF Initial value problems for system of O.D.E.s, (simple driver) variable order and step Adams method, until a component of the solution attains a given value. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CGE. | Class(es): 11a | Usage: CALL D02CGF (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IFAIL) | On-line doc: CALL GAMSDOC D02CGF (or @PRT NAG*DOC.D02CGF) | Access: LIB NBS*NAG

D02CHE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, until a function of the solution is zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02CHF. | Class(es): 11a | Usage: CALL D02CHE (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, G, W , IFAIL) | On-line doc: CALL GAMSDOC D02CHE (or @PRT NAG*DOC.D02CHE) | Access: LIB NBS*NAG

D02CHF Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Adams method, until a function of the solution is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02CHE. | Class(es): 11a | Usage: CALL D02CHF (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, G, W , IFAIL) | On-line doc: CALL GAMSDOC D02CHF (or @PRT NAG*DOC.D02CHF) | Access: LIB NBS*NAG

D02EAE Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Gear method for stiff systems, over a range. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EAF. | Class(es): 11a | Usage: CALL D02EAE (X, XEND, N, Y, TOL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EAE (or @PRT NAG*DOC.D02EAF) | Access: LIB NBS*NAG

D02EAF Initial value problems for system of O.D.E.s, (simple driver) variable-order variable-step Gear method for stiff systems, over a range. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EAE. | Class(es): 11a | Usage: CALL D02EAF (X, XEND, N, Y, TOL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EAF (or @PRT NAG*DOC.D02EAF) | Access: LIB NBS*NAG

D02EBE Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, over a range with intermediate output. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EBF. | Class(es): 11a | Usage: CALL D02EBE (X, XEND, N, Y, TOL, IRELAB, FCN, MPED, PEDERV, OUTPUT, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EBE (or @PRT NAG*DOC.D02EBE) | Access: LIB NBS*NAG

D02EBF Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, over a range with intermediate output. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EBE. | Class(es): 11a | Usage: CALL D02EBF (X, XEND, N, Y, TOL, IRELAB, FCN, MPED, PEDERV, OUTPUT, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EBF (or @PRT NAG*DOC.D02EBF) | Access: LIB NBS*NAG

D02EGE Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until the solution attains a given value. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EGF. | Class(es): 11a | Usage: CALL D02EGE (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EGE (or @PRT NAG*DOC.D02EGE) | Access: LIB NBS*NAG

D02EGF Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until the solution attains a given value. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EGE. | Class(es): 11a | Usage: CALL D02EGF (X, XEND, N, Y, TOL, HMAX, M, VAL, FCN, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EGF (or @PRT NAG*DOC.D02EGF) | Access: LIB NBS*NAG

D02EHE Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until a function of the solution is zero. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02EHF. | Class(es): 11a | Usage: CALL D02EHE (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, MPED, PEDERV, G, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EHE (or @PRT NAG*DOC.D02EHE) | Access: LIB NBS*NAG

D02EHF Initial value problems for system of O.D.E.s, (simple driver) variable order and step Gear method for stiff systems, until a function of the solution is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02EHE. | Class(es): 11a | Usage: CALL D02EHF (X, XEND, N, Y, TOL, IRELAB, HMAX, FCN, MPED, PEDERV, G, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02EHF (or @PRT NAG*DOC.D02EHF) | Access: LIB NBS*NAG

D02GAE Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), simple non-linear problem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02GAF. | Class(es): 11b | Usage: CALL D02GAE (U, V, N, A, B, TOL, FCN, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GAE (or @PRT NAG*DOC.D02GAE) | Access: LIB NBS*NAG

**D02GBE** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), general linear problem.

| Usage: Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02GBF. | Class(es): Ilb1 | Usage: CALL D02GBE (A, B, N, TOL, FCNF, FCNG, C, D, GAM, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GBE (or @PRT NAG+DOC.D02GBE) | Access: LIB NBS+NAG |

**D02GBF** Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction (Pereyra), general linear problem.

| Usage: Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02GBE. | Class(es): Ilb1 | Usage: CALL D02GBF (A, B, N, TOL, FCNF, FCNG, C, D, GAM, MNP, X, Y, NP, W, LW, IW, LIW, IFAIL) | On-line doc: CALL GAMSDOC D02GBF (or @PRT NAG+DOC.D02GBF) | Access: LIB NBS+NAG |

**D02HAE** Boundary-value problems for system of O.D.E.s, shooting and matching technique, boundary values to be determined.

| Usage: Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02HAF. | Class(es): Ilb2 | Usage: CALL D02HAE (U, V, N, A, B, TOL, FCN, SOLN, M1, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HAE (or @PRT NAG+DOC.D02HAE) | Access: LIB NBS+NAG |

**D02HAF** Boundary-value problems for system of O.D.E.s, shooting and matching technique, boundary values to be determined.

| Usage: Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02HAE. | Class(es): Ilb2 | Usage: CALL D02HAF (U, V, N, A, B, TOL, FCN, SOLN, M1, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HAF (or @PRT NAG+DOC.D02HAF) | Access: LIB NBS+NAG |

**D02HBE** Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined.

| Usage: Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02HBF. | Class(es): Ilb2 | Usage: CALL D02HBE (P, N1, PE, E, N, SOLN, M1, FCN, BC, RANGE, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HBE (or @PRT NAG+DOC.D02HBE) | Access: LIB NBS+NAG |

**D02HBF** Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined.

| Usage: Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02HBE. | Class(es): Ilb2 | Usage: CALL D02HBF (P, N1, PE, E, N, SOLN, M1, FCN, BC, RANGE, W, IW, IFAIL) | On-line doc: CALL GAMSDOC D02HBF (or @PRT NAG+DOC.D02HBF) | Access: LIB NBS+NAG |

**D02JAE** Solves a regular linear two point boundary value problem for a single n-th order ordinary differential equation by a Chebyshev series using collocation and least-squares.

| Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02JAF. | Class(es): Ilb1 | Usage: CALL D02JAE (N, CF, BC, X0, X1, K1, KP, C, W, LW, IW, IFAIL) | On-line doc: CALL GAMSDOC D02JAE (or @PRT NAG+DOC.D02JAE) | Access: LIB NBS+NAG |

**D02JAF** Solves a regular linear two point boundary value problem for a single n-th order ordinary differential equation by a Chebyshev series using collocation and least-squares.

| Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02JAE. | Class(es): Ilb1 | Usage: CALL D02JAF (N, CF, BC, X0, X1, K1, KP, C, W, LW, IW, IFAIL) | On-line doc: CALL GAMSDOC D02JAF (or @PRT NAG+DOC.D02JAF) | Access: LIB NBS+NAG |

**D02JBE** Boundary-value problems for system of O.D.E.s, collocation and least-squares, system of 1st order linear equations.

| Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02JBF. | Class(es): Ilb1 | Usage: CALL D02JBE (N, CF, BC, X0, X1, K1, KP, C, IC, W, LW, IW, IFAIL) | On-line doc: CALL GAMSDOC D02JBE (or @PRT NAG+DOC.D02JBE) | Access: LIB NBS+NAG |

**D02JBF** Boundary-value problems for system of O.D.E.s, collocation and least-squares, system of 1st order linear equations.

| Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02JBE. | Class(es): Ilb1 | Usage: CALL D02JBF (N, CF, BC, X0, X1, K1, KP, C, IC, W, LW, IW, IFAIL) | On-line doc: CALL GAMSDOC D02JBF (or @PRT NAG+DOC.D02JBF) | Access: LIB NBS+NAG |

**D02KAE** Second-order Sturm-Liouville problems, regular system, finite range, eigenvalue only.

| Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02KAF. | Class(es): Ilb3 | Usage: CALL D02KAE (X, XR, COEFFN, BCOND, K, TOL, ELAM, DELAM, MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KAE (or @PRT NAG+DOC.D02KAE) | Access: LIB NBS+NAG |

**D02KAF** Second-order Sturm-Liouville problems, regular system, finite range, eigenvalue only.

| Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02KAE. | Class(es): Ilb3 | Usage: CALL D02KAF (X, XR, COEFFN, BCOND, K, TOL, ELAM, DELAM, MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KAF (or @PRT NAG+DOC.D02KAF) | Access: LIB NBS+NAG |

**D02KDE** Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue only.

| Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02KDF. | Class(es): Ilb3 | Usage: CALL D02KDE (XPOINT, M, COEFFN, BDYVAL, K, TOL, ELAM, HMAX, MAXIT, MAXFUN MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KDE (or @PRT NAG+DOC.D02KDE) | Access: LIB NBS+NAG |

**D02KDF** Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue only.

| Proprietary double precision Fortran subroutine in NAG library. Single precision version is D02KDE. | Class(es): Ilb3 | Usage: CALL D02KDF (XPOINT, M, COEFFN, BDYVAL, K, TOL, ELAM, HMAX, MAXIT, MAXFUN MONIT, IFAIL) | On-line doc: CALL GAMSDOC D02KDF (or @PRT NAG+DOC.D02KDF) | Access: LIB NBS+NAG |

**D02KEE** Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue and eigenfunction.

| Proprietary single precision Fortran subroutine in NAG library. Double precision version is D02KEF. | Class(es): Ilb3 | Usage: CALL D02KEE (XPOINT, M, MATCH, COEFFN, BDYVAL, K, TOL, ELAM, HMAX, MAXIT, MAXFUN, MONIT, REPORT, IFAIL) | On-line doc: CALL GAMSDOC D02KEE (or @PRT NAG+DOC.D02KEE) | Access: LIB NBS+NAG |
GAMS: Module Dictionary

C 33

D02KEF Second-order Sturm-Liouville problems, regular/singular system, finite/infinite range, eigenvalue and eigenfunction. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02KEE. | Class(es): I1b5 | Usage: CALL D02KEF (XPONINT, M, MATCH, COEFFN, BDYVAL, K, TOL, ELAM, DELAM, HMAX, MAXIT, MAXFUN, MONIT, REPORT, IFAIL) | Online doc: CALL GAMSDOC D02KEF (or @VRT NAG+DOC.D02KEF) | Access: LIB NBS+NAG

D02PAE Initial value problems for system of O.D.E.s, integrating over a range (facilities for error-control and interrupts) Runge-Kutta-Merson method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02PAF. | Class(es): I1a1 | Usage: CALL D02PAE (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02PAE (or @VRT NAG+DOC.D02PAE) | Access: LIB NBS+NAG | See also: D02XAE D02XBE

D02PAF Initial value problems for system of O.D.E.s, integrating over a range (facilities for error-control and interrupts) Runge-Kutta-Merson method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02PAE. | Class(es): I1a1 | Usage: CALL D02PAF (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02PAF (or @VRT NAG+DOC.D02PAF) | Access: LIB NBS+NAG | See also: D02XAF D02XBF

D02QAE Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Adams method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02QAF. | Class(es): I1a1 | Usage: CALL D02QAE (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02QAE (or @VRT NAG+DOC.D02QAE) | Access: LIB NBS+NAG | See also: D02QAF D02QBF

D02QAF Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Adams method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02QAE. | Class(es): I1a1 | Usage: CALL D02QAF (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, W, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02QAF (or @VRT NAG+DOC.D02QAF) | Access: LIB NBS+NAG | See also: D02QGF D02XHP

D02QBE Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Gear method for stiff systems. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02QBE. | Class(es): I1a2 | Usage: CALL D02QBE (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, MPED, PEDERV, PW, W, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02QBE (or @VRT NAG+DOC.D02QBE) | Access: LIB NBS+NAG | See also: D02QBE D02RRAE

D02QBF Initial value problems for system of O.D.E.s, integrating over a range (error-control and interrupts) variable-order -step Gear method for stiff systems. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02QBE. | Class(es): I1a2 | Usage: CALL D02QBF (X, XEND, N, Y, CIN, TOL, FCN, COMM, CONST, COUT, PW, W, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02QBF (or @VRT NAG+DOC.D02QBF) | Access: LIB NBS+NAG | See also: D02QGF D02XHP

D02RAE Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction, general non-linear problem, continuation facility. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02RAF. | Class(es): I1b2 | Usage: CALL D02RAE (N, MNP, NP, NUMBEG, NUMMIX, TOL, INIT, X, Y, IY, ABT, FCN, G, IJAC, JACOBF, JACOGB, DELEPS, JACEPS, JACGEP, WORK, LWORK, LW, LWORK, IWORK, LWORK, IFAIL) | Online doc: CALL GAMSDOC D02RAE (or @VRT NAG+DOC.D02RAE) | Access: LIB NBS+NAG

D02RAF Boundary-value problems for system of O.D.E.s, finite difference technique with deferred correction, general non-linear problem, continuation facility. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02RAE. | Class(es): I1b2 | Usage: CALL D02RAF (N, MNP, NP, NUMBEG, NUMMIX, TOL, INIT, X, Y, IY, ABT, FCN, G, IJAC, JACOBF, JACOGB, DELEPS, JACEPS, JACGEP, WORK, LWORK, LW, LWORK, IWORK, IFAIL) | Online doc: CALL GAMSDOC D02RAF (or @VRT NAG+DOC.D02RAF) | Access: LIB NBS+NAG

D02SAE Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined, subject to extra algebraic equations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02SAF. | Class(es): I1b2 | Usage: CALL D02SAE (P, M, N, N1, PE, PF, E, DP, NPOINT, WP, IWP, ICOUNT, RANGE, BC, FCN EQN, CONST, YMAX, MONIT, PRSOL, W, IW1, IW2, IFAIL) | Online doc: CALL GAMSDOC D02SAE (or @VRT NAG+DOC.D02SAE) | Access: LIB NBS+NAG

D02SAE Boundary-value problems for system of O.D.E.s, shooting and matching technique, general parameters to be determined, subject to extra algebraic equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02SASAE. | Class(es): I1b2 | Usage: CALL D02SAE (P, M, N, N1, PE, PF, E, DP, NPOINT, WP, IWP, ICOUNT, RANGE, BC, FCN EQN, CONST, YMAX, MONIT, PRSOL, W, IW1, IW2, IFAIL) | Online doc: CALL GAMSDOC D02SAE (or @VRT NAG+DOC.D02SAE) | Access: LIB NBS+NAG

D02TGE Boundary-value problems for system of ordinary differential equations, collocation and least-squares, system of n-th order linear equations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02TG2F. | Class(es): I1b1 | Usage: CALL D02TGE (N, M, L, X0, X1, K1, KP, C, IC, COEFF, BDYC, W, LW, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02TGE (or @VRT NAG+DOC.D02TGE) | Access: LIB NBS+NAG

D02TG2F Boundary-value problems for system of ordinary differential equations, collocation and least-squares, system of n-th order linear equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02TGE. | Class(es): I1b1 | Usage: CALL D02TG2F (N, M, L, X0, X1, K1, KP, C, IC, COEFF, BDYC, W, LW, IW, IW1, IFAIL) | Online doc: CALL GAMSDOC D02TG2F (or @VRT NAG+DOC.D02TG2F) | Access: LIB NBS+NAG

D02XAE Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, all components. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XAF. | Class(es): I1c | Usage: CALL D02XAE (XSOL, X, COUT, N, Y, W, IW, IW1, SOL, IFAIL) | Online doc: CALL GAMSDOC D02XAE (or @VRT NAG+DOC.D02XAE) | Access: LIB NBS+NAG | See also: D02PAE
D03EAF Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAF, all components.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XAE. | Class(es): 11c | Usage: CALL D03EAF (XSOL, X, Y, W, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D03EAF (or OPRT NAG+DOC.D03EAF) | Access: LIB NBS+NAG | See also: D02PAF

D02XBE Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAE, one component.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XBE. | Class(es): 11c | Usage: CALL D02XBE (XSOL, X, COUT, N, Y, W, I, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XBE (or OPRT NAG+DOC.D02XBE) | Access: LIB NBS+NAG | See also: D02PAE

D02XBF Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02PAF, one component.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XBE. | Class(es): 11c | Usage: CALL D02XBF (XSOL, X, COUT, N, Y, W, I, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XBF (or OPRT NAG+DOC.D02XBF) | Access: LIB NBS+NAG | See also: D02PAF

D02XGE Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, all components.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XGF. | Class(es): 11c | Usage: CALL D02XGE (XSOL, X, COUT, N, Y, W, I, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XGE (or OPRT NAG+DOC.D02XGE) | Access: LIB NBS+NAG | See also: D02QAE D02QBE

D02XGF Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAF or D02QBF, all components.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XGE. | Class(es): 11c | Usage: CALL D02XGF (XSOL, X, COUT, N, Y, W, I, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XGF (or OPRT NAG+DOC.D02XGF) | Access: LIB NBS+NAG | See also: D02QAF D02QBF

D02XHE Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAE or D02QBE, one component.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02XHE. | Class(es): 11c | Usage: CALL D02XHE (XSOL, X, COUT, N, W, I, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XHE (or OPRT NAG+DOC.D02XHE) | Access: LIB NBS+NAG | See also: D02QAE D02QBE

D02XHF Initial value problems for system of ordinary differential equations, interpolation of solutions produced by D02QAF or D02QBF, one component.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02XHE. | Class(es): 11c | Usage: CALL D02XHF (XSOL, X, COUT, N, W, I, M, SOL, IFAIL) | On-line doc: CALL GAMSDOC D02XHF (or OPRT NAG+DOC.D02XHF) | Access: LIB NBS+NAG | See also: D02QAF D02QBF

D02YAE Initial value problems for system of ordinary differential equations, integration over one step by Runge-Kutta-Merson method.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D02YAF. | Class(es): 11a | Usage: CALL D02YAE (X, H, N, Y, FCN, W, I1, I2) | On-line doc: CALL GAMSDOC D02YAE (or OPRT NAG+DOC.D02YAE) | Access: LIB NBS+NAG

D02YAF Initial value problems for system of ordinary differential equations, integration over one step by Runge-Kutta-Merson method.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D02YAE. | Class(es): 11a | Usage: CALL D02YAF (X, H, N, Y, FCN, W, I1, I2) | On-line doc: CALL GAMSDOC D02YAF (or OPRT NAG+DOC.D02YAF) | Access: LIB NBS+NAG

D03EAE Partial differential equations, elliptic, Laplace's equation in 2-d for an arbitrary domain.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03EAF. | Class(es): 12b | Usage: CALL D03EAE (STAGE1, EXT, DORM, N, P, Q, X, Y, N1P1, PHI, PHID, ALPHA, C, IC, NP4, ICINT, NP1, IFAIL) | On-line doc: CALL GAMSDOC D03EAE (or OPRT NAG+DOC.D03EAE) | Access: LIB NBS+NAG

D03EAF Partial differential equations, elliptic, Laplace's equation in 2-d for an arbitrary domain.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03EAE. | Class(es): 12b | Usage: CALL D03EAF (STAGE1, EXT, DORM, N, P, Q, X, Y, N1P1, PHI, PHID, ALPHA, C, IC, NP4, ICINT, NP1, IFAIL) | On-line doc: CALL GAMSDOC D03EAF (or OPRT NAG+DOC.D03EAF) | Access: LIB NBS+NAG

D03EBE P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 6-point 2-d molecule, iterate to convergence.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03EBE. | Class(es): 12b | Usage: CALL D03EBE (N1, N2, N3, N4, M, A, B, C, D, E, Q, T, APARAM, ITMAX, ITCON, ITCOUNT, ITUSED, NDIR, ION, N, YN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, WRKSP2, WRKSP3, IFAIL) | On-line doc: CALL GAMSDOC D03EBE (or OPRT NAG+DOC.D03EBE) | Access: LIB NBS+NAG

D03EBF P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 6-point 2-d molecule, iterate to convergence.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03EBE. | Class(es): 12b | Usage: CALL D03EBF (N1, N2, N3, N4, M, A, B, C, D, E, Q, T, APARAM, ITMAX, ITCON, ITCOUNT, ITUSED, NDIR, ION, N, YN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, WRKSP2, WRKSP3, IFAIL) | On-line doc: CALL GAMSDOC D03EBF (or OPRT NAG+DOC.D03EBF) | Access: LIB NBS+NAG

D03ECE P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, iterate to convergence.
| Proprietary single precision Fortran subprogram in NAG library. Double precision version is D03ECE. | Class(es): 12b | Usage: CALL D03ECE (N1, N2, N3, N4, M, A, B, C, D, E, F, G, T, APARAM, ITMAX, ITCON, ITCOUNT, ITUSED, NDIR, ION, N, YN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, | On-line doc: CALL GAMSDOC D03ECE (or OPRT NAG+DOC.D03ECE) | Access: LIB NBS+NAG

D03ECF P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, iterate to convergence.
| Proprietary double precision Fortran subprogram in NAG library. Single precision version is D03ECF. | Class(es): 12b | Usage: CALL D03ECF (N1, N2, N3, N4, M, A, B, C, D, E, F, G, T, APARAM, ITMAX, ITCON, ITCOUNT, ITUSED, NDIR, ION, N, YN, CONRES, CONCHN, RESIDS, CHNGS, WRKSP1, | On-line doc: CALL GAMSDOC D03ECF (or OPRT NAG+DOC.D03ECF) | Access: LIB NBS+NAG
GAMS: Module Dictionary

January 1984

D03MAE Triangulation of a plane region. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is D03MAP. | Class(es): 12b4a P2a2c1 | Usage: CALL D03MAE (H, M, N, NB, NPTS, PLACES, INDEX, IDIM, IN, DIST, LD, IFAIL) | On-line doc: CALL GAMSDOC D03MAE (or ®PRT NAG*DOC.D03MAE) | Access: LIB NBS*NAG

D03MAP Triangulation of a plane region. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is D03MAE. | Class(es): 12b4a P2a2c1 | Usage: CALL D03MAP (H, M, N, NB, NPTS, PLACES, INDEX, IDIM, IN, DIST, LD, IFAIL) | On-line doc: CALL GAMSDOC D03MAP (or ®PRT NAG*DOC.D03MAP) | Access: LIB NBS*NAG

D03PAE P.D.E.s, parabolic, one space variable, method of lines, single equation. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is D03PAF. | Class(es): 12a1a | Usage: CALL D03PAE (M, A, B, TS, TOUT, U, NPTS, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PAE (or ®PRT NAG*DOC.D03PAE) | Access: LIB NBS*NAG

D03PAF P.D.E.s, parabolic, one space variable, method of lines, single equation. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is D03PAE. | Class(es): 12a1a | Usage: CALL D03PAF (M, A, B, TS, TOUT, U, NPTS, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PAF (or ®PRT NAG*DOC.D03PAF) | Access: LIB NBS*NAG

D03PBE P.D.E.s, parabolic, one space variable, method of lines, simple system. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is D03PBF. | Class(es): 12a1a | Usage: CALL D03PBE (NPDE, M, PDEF, BNDY, A, B, TS, TOUT, U, NPTS, IMESH, X, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PBE (or ®PRT NAG*DOC.D03PBE) | Access: LIB NBS*NAG

D03PBF P.D.E.s, parabolic, one space variable, method of lines, simple system. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is D03PBE. | Class(es): 12a1a | Usage: CALL D03PBF (NPDE, M, PDEF, BNDY, A, B, TS, TOUT, U, NPTS, IMESH, X, ACC, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PBF (or ®PRT NAG*DOC.D03PBF) | Access: LIB NBS*NAG

D03PGF P.D.E.s, parabolic, one space variable, method of lines, general system. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is D03PGF. | Class(es): 12a1a | Usage: CALL D03PGF (NPDE, M, PDEF, BNDY, TS, TOUT, U, I, NPTS, X, RELERR, ABERR, INORM, MONTR, IMON, IBAND, WORK, IWK, IND, IFAIL) | On-line doc: CALL GAMSDOC D03PGF (or ®PRT NAG*DOC.D03PGF) | Access: LIB NBS*NAG

D03UGA P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, one iteration. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is D03UGAF. | Class(es): 12b4b | Usage: CALL D03UGA (N1, N2, N1M, A, B, C, D, E, APARAM, IT, R, W RKSP1, WRKSP2, IFAIL) | On-line doc: CALL GAMSDOC D03UGA (or ®PRT NAG*DOC.D03UGA) | Access: LIB NBS*NAG

D03UGAF P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 5-point 2-d molecule, one iteration. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is D03UGA. | Class(es): 12b4b | Usage: CALL D03UGAF (N1, N2, N1M, A, B, C, D, E, APARAM, IT, R, W RKSP1, WRKSP2, IFAIL) | On-line doc: CALL GAMSDOC D03UGAF (or ®PRT NAG*DOC.D03UGAF) | Access: LIB NBS*NAG

D03UBA P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, one iteration. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is D03UBAF. | Class(es): 12b4b | Usage: CALL D03UBA (N1, N2, N3, N1M, N2M, A, B, C, D, E, F, G, APARAM, IT, R, W RKSP1, WRKSP2, IFAIL) | On-line doc: CALL GAMSDOC D03UBA (or ®PRT NAG*DOC.D03UBA) | Access: LIB NBS*NAG

D03UBAF P.D.E.s, elliptic, solution of finite difference equations by strongly implicit procedure, for 7-point 3-d molecule, one iteration. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is D03UBA. | Class(es): 12b4b | Usage: CALL D03UBAF (N1, N2, N3, N1M, N2M, A, B, C, D, E, F, G, APARAM, IT, R, W RKSP1, WRKSP2, IFAIL) | On-line doc: CALL GAMSDOC D03UBAF (or ®PRT NAG*DOC.D03UBAF) | Access: LIB NBS*NAG


library. Double precision version is D05ABF. | Class(es): 18 | Usage: CALL D05ABE (K, G, LAMBDA, A, B, ODOREV, EV, N, CM, F1, WK, NMAX, NT2P1, F, C, IFAIL) | On-line doc: CALL GAMSDOC D05ABE (or @PRT NAG+DOC.D05ABE) | Access: LIB NBS+NAG

D05ABF Linear non-singular Fredholm integral equation, 2nd kind, smooth kernel. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is D05ABF. | Class(es): 18 | Usage: CALL D05ABE (K, G, LAMBDA, A, B, ODOREV, EV, N, CM, F1, WK, NMAX, NT2P1, F, C, IFAIL) | On-line doc: CALL GAMSDOC D05ABF (or @PRT NAG+DOC.D05ABF) | Access: LIB NBS+NAG

D1MACH Provides machine dependent information, e.g. D1MACH(4) returns double precision machine epsilon. | Portable double precision Fortran subprogram in MACHCONST sublibrary of CMLIB library. Single precision version is R1MACH. | Class(es): R1 | Usage: D1MACH(I) | On-line doc: CALL GAMSDOC D1MACH (or @PRT CMLIB+DOC.D1MACH/MACHCONST) | Tests: CMLIB+TEST-SOURCE.8Q/MACHCONST | Access: LIB NBS+CMLIB

D1MACH Provides machine dependent double precision constants for the PORT library programs. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is R1MACH. | Class(es): R1 | Usage: D1MACH(I) | On-line doc: CALL GAMSDOC D1MACH (or @PRT PORT+DOC.D1MACH) | Access: LIB NBS+PORT

DACOS Arc cosine of double precision argument, cos**-1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: D1DACOS(D) | On-line doc: CALL GAMSDOC DACOS (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DACOSH Arc hyperbolic cosine of double precision argument, cosh**-1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ACOSH. | Class(es): C4c | Usage: D1DACOSH(D) | On-line doc: CALL GAMSDOC DACOSH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DACOSH Computes hyperbolic arc cosine, arcosh. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ACOSH. | Class(es): C4c | Usage: D = DACOSH(X) | On-line doc: CALL GAMSDOC DACOSH (or @PRT PORT+DOC.DACOSH) | Access: LIB NBS+PORT

DARI Airy function of double precision argument, Ai(d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is AI. | Class(es): C1od | Usage: D1DAI(D) | On-line doc: CALL GAMSDOC DARI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DAIE Airy function of double precision argument, -Ai(d), d <= 0; exp(2/3 * d**2(3/2)) * Ai(d), d > 0. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is AIE. | Class(es): C1od | Usage: D1DAIE(D) | On-line doc: CALL GAMSDOC DAIE (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DARCO Computes arcsec(x), answer in radians. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ARCOS. | Class(es): C4a | Usage: D = DARCO(X) | On-line doc: CALL GAMSDOC DARCO (or @PRT PORT+DOC.DARCO) | Access: LIB NBS+PORT

DARCS Computes arcsec(x), answer in radians. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ARCOS. | Class(es): C4a | Usage: D = DARCS(X) | On-line doc: CALL GAMSDOC DARCS (or @PRT PORT+DOC.DARCS) | Access: LIB NBS+PORT

DARSI Computes arcsin(x), answer in radians. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ARSIN. | Class(es): C4a | Usage: D = DARSI(X) | On-line doc: CALL GAMSDOC DARSI (or @PRT PORT+DOC.DARSI) | Access: LIB NBS+PORT

DASIN Arc sine of double precision argument, sin**-1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: D1DASIN(D) | On-line doc: CALL GAMSDOC DASIN (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DASINH Arc hyperbolic sine of double precision argument, sinh**-1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ASINH. | Class(es): C4c | Usage: D1DASINH(D) | On-line doc: CALL GAMSDOC DASINH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DASINH Computes hyperbolic arcsine, arcsinh(x). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ASINH. | Class(es): C4c | Usage: D = DASINH(X) | On-line doc: CALL GAMSDOC DASINH (or @PRT PORT+DOC.DASINH) | Access: LIB NBS+CMLIB


DATANH Arc hyperbolic tangent of double precision argument, tanh**-1 (d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ATANH. | Class(es): C4c | Usage: D1DATANH(D) | On-line doc: CALL GAMSDOC DATANH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DATANH Computes hyperbolic arctangent, arctanh(x). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ATANH. | Class(es): C4c | Usage: D = DATANH(X) | On-line doc: CALL GAMSDOC DATANH (or @PRT PORT+DOC.DATANH) | Access: LIB NBS+PORT

DAWS Dawson's function, F(x) = e**x(e**x-1) * integral from x to 0 of e**((x**2)2)dt. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DDAWS. | Class(es): C8c | Usage: R = DAWS(X) | On-line doc: CALL GAMSDOC DAWS (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DAXPY Compute a constant times a vector plus a vector, all double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SAXPY. | Class(es): D1a7 | Usage: CALL DAXPY(N,DA,DX,INCX,DY,INCY) |
On-line doc: CALL GAMSDOC DAXPY (or @PRT CMLIB+DOC.DAXPY/BLAS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

DB21NK Computes parameters of a piecewise-polynomial that interpolates a given set of two-dimensional gridded data. (Double precision version of DB21NK.) | Portable double precision Fortran program in DTENSORBS sublibrary of CMLIB library. Single precision version is DB21NK. | Class(es): E2a | Usage: CALL DB21NK(X,N,X,Y,FCN,DTENSORBS,LIB NBS+CMLIB) | On-line doc: CALL GAMSDOC DB21NK (or @PRT CMLIB+DOC.DB21NK/DTENSORBS) | Tests: CMLIB+TEST-SOURCE.$Q/DTENSORBS | Access: LIB NBS+CMLIB | See also: DB21VAL

DB2VAL Evaluates the two-dimensional interpolating function computed by DB21NK or one of its partial derivatives. (Double precision version of DB2VAL.) | Portable double precision Fortran program in DTENSORBS sublibrary of CMLIB library. Single precision version is DB2VAL. | Class(es): E3 | Usage: CALL DB2VAL(X,Y,VAL,IDX,IDX,TX,TY,XX,YY,XX,YY,WK,IFLAG) | On-line doc: CALL GAMSDOC DB2VAL (or @PRT CMLIB+DOC.DB2VAL/DTENSORBS) | Tests: CMLIB+TEST-SOURCE.$Q/DTENSORBS | Access: LIB NBS+CMLIB

DB31NK Computes parameters of a piecewise-polynomial that interpolates a given set of three-dimensional gridded data. (Double precision version of DB31NK.) | Portable double precision Fortran program in DTENSORBS sublibrary of CMLIB library. Single precision version is DB31NK. | Class(es): E2a | Usage: CALL DB31NK(X,N,X,Y,FCN,DTENSORBS,LIB NBS+CMLIB) | On-line doc: CALL GAMSDOC DB31NK (or @PRT CMLIB+DOC.DB31NK/DTENSORBS) | Tests: CMLIB+TEST-SOURCE.$Q/DTENSORBS | Access: LIB NBS+CMLIB

DB3VAL Evaluates the three-dimensional interpolating function computed by DB31NK or one of its partial derivatives. (Double precision version of DB3VAL.) | Portable double precision Fortran program in DTENSORBS sublibrary of CMLIB library. Single precision version is DB3VAL. | Class(es): E3 | Usage: CALL DB3VAL(X,Y,VAL,IDX,IDX,TX,TY,XX,YY,XX,YY,WK,IFLAG) | On-line doc: CALL GAMSDOC DB3VAL (or @PRT CMLIB+DOC.DB3VAL/DTENSORBS) | Tests: CMLIB+TEST-SOURCE.$Q/DTENSORBS | Access: LIB NBS+CMLIB


DBCQDU Bicubic spline quadrature. | Proprietary single precision Fortran program in IMSL library. | Class(es): H2b1b2 | Usage: CALL DBCQDU(F,D,F,NX,DX,NNY,YN,NNZ,ZN,BR,WER) | On-line doc: CALL GAMSDOC DBCQDU (or @PRT IMSL+DOC.DBCQDU) | Access: LIB NBS+IMSL

DBESCJ Bessel functions, J, of complex argument and integer order. | Proprietary double precision Fortran program in PORT library. Single precision version is DBESCJ. | Class(es): Cl0a2 | Usage: CALL DBESCJ(XR,XI,NS,NS,NS,NS,BR,BI) | On-line doc: CALL GAMSDOC DBESCJ (or @PRT PORT+DOC.DBESCJ) | Access: LIB NBS+PORT

DBESCJ Bessel functions, J, of complex argument and integer order. | Proprietary double precision Fortran program in PORT library. Single precision version is DBESCJ. | Class(es): Cl0a2 | Usage: CALL DBESCJ(XR,XI,NS,NS,NS,NS,BR,BI) | On-line doc: CALL GAMSDOC DBESCJ (or @PRT PORT+DOC.DBESCJ) | Access: LIB NBS+PORT

DBES0 Bessel function, first kind, order zero, 1 sub 0 (d). Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBES0. | Class(es): Cl0b1 | Usage: D1=DBES0(D) | On-line doc: CALL GAMSDOC DBES0 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBES1 Bessel function, first kind, order one, 1 sub 1 (d). Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBES1. | Class(es): Cl0b1 | Usage: D1=DBES1(D) | On-line doc: CALL GAMSDOC DBES1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBESJ Bessel function, first kind, order zero, J sub 0 (d). Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBESJ. | Class(es): Cl0a1 | Usage: D1=DBESJ(D) | On-line doc: CALL GAMSDOC DBESJ (or @PRT PORT+DOC.DBESCJ) | Access: LIB NBS+PORT

DBESJ Bessel function, first kind, order one, J sub 1 (d). Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBESJ. | Class(es): Cl0a1 | Usage: D1=DBES1(D) | On-line doc: CALL GAMSDOC DBES1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBESK Bessel function, modified, third kind order zero, K sub 0 (d). Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBESK. | Class(es): Cl0b1 | Usage: D1=DBESK0(D) | On-line doc: CALL GAMSDOC DBESK0 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBESK Bessel function, modified, third kind order one, K sub 1 (d). Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBESK. | Class(es): Cl0b1 | Usage: D1=DBESK1(D) | On-line doc: CALL GAMSDOC DBESK1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBESK Compute sequence of Bessel functions, modified, of third kind, K sub v+i (d), for n values of i going up or down from zero. Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DBESK. | Class(es): Cl0b1 | Usage: CALL DBESK(DXNU,D,N,DBK) | On-line doc: CALL GAMSDOC DBESK (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBESR Bessel functions, l, of real argument and integer order. | Proprietary double precision Fortran program in PORT library. Single precision version is DBESR. | Class(es): Cl0b1 | Usage: CALL DBESR(X,N,B) | On-line doc: CALL GAMSDOC DBESR (or @PRT PORT+DOC.DBESR) | Access: LIB NBS+PORT
DBESRJ  Bessel functions, J, of real argument and integer order.  | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BESRJ.  | Class(es): C10a1  | Usage: CALL DBESRJ(X, NB, B)  | On-line doc: CALL GAMSDOC DBESRJ (or @PRT PORT*DOC.DBESRJ)  | Access: LIB NBS*PORT

DBESY0  Bessel function, second kind order zero, Y sub 0 (d).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESY0.  | Class(es): C10a1  | Usage: D=DBESY0(D)  | On-line doc: CALL GAMSDOC DBESY0 (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBESY1  Bessel function, second kind order one, Y sub 1 (d).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESY1.  | Class(es): C10a1  | Usage: D=DBESY1(D)  | On-line doc: CALL GAMSDOC DBESY1 (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBETA  Beta function of double precision arguments, B(a,b) = (Gamma(a) * Gamma(b)) / Gamma(a+b).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BETA.  | Class(es): C7b  | Usage: D=DBETA(DA,DB)  | On-line doc: CALL GAMSDOC DBETA (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBFQAD  Integrates function times derivative of B-spline from X1 to X2. The B-spline is in "B" representation.  | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BQFAD.  | Class(es): H2a2a1 E3 K6  | Usage: CALL DBFQAD(F,T,BCOEF,N,K,ID,X1,X2,TOI,QUAD,JERR,WORK)  | On-line doc: CALL GAMSDOC DBFQAD (or @PRT CMLIB*DOC.DBFQAD/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE)  | Tests: CMLIB*TEST-SOURCE.$F/DBSPLINE  | Access: LIB NBS*CMLIB

DBI  Double precision Bairy function of a real argument, Bi(d1).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BI.  | Class(es): C10d  | Usage: D = DBI(D1)  | On-line doc: CALL GAMSDOC DBI (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBIE  Exponentially scaled double precision Bairy function, -Bi(d1) for d1 < 0, = exp(-2/5 * d1**2/5) * Bi(d1) for d1 > 0.  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BIE.  | Class(es): C10d  | Usage: D = DBIE(D1)  | On-line doc: CALL GAMSDOC DBIE (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBINOM  Double precision Binomial function, ml / (ml * (n-m)!).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BINOM.  | Class(es): C1  | Usage: D = DBINOM(N,M)  | On-line doc: CALL GAMSDOC DBINOM (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB


DBINTK  Produces B-spline coefficients of k-th order B-spline with given knots and with values at given points.  | Portable double precision Fortran subprogram in DBSPLINE sublibrary of CMLIB library. Single precision version is BINTK.  | Class(es): E1a  | Usage: CALL DBINTK(X,Y,T,N,K,BCOEF,Q,WORK)  | On-line doc: CALL GAMSDOC DBINTK (or @PRT CMLIB*DOC.DBINTK/DBSPLINE and CMLIB*DOC.SUMMARY/BSPLINE)  | Tests: CMLIB*TEST-SOURCE.$F/DBSPLINE  | Access: LIB NBS*CMLIB  | See also: DBVALU for evaluation. See package documentation for other facilities.

DBLIN  Numerical integration of a function of two variables.  | Proprietary single precision Fortran program in IMSL library.  | Class(es): H2b2a1  | Usage: R = DUBLIN(F,X,B,X,Y,B,Y,AERR,ERROR,IER)  | On-line doc: CALL GAMSDOC DUBLIN (or @PRT IMSL*DOC.DBLIN)  | Access: LIB NBS*IMSL

DBQUAD  Adaptively integrates functions which have discontinuities in their derivatives. User can specify these points.  | Proprietary double precision Fortran subprogram in PORT library. Single precision version is BQUAD.  | Class(es): H2a2a1  | Usage: CALL DBQUAD(F,N,X,EP,S,AN,ERREST)  | On-line doc: CALL GAMSDOC DBQUAD (or @PRT PORT*DOC.DBQUAD)  | Access: LIB NBS*PORT

DBS0E  Modified (hyperbolic) Bessel function, dbie precision, of special integer order scaled by an exponential, First kind, order zero, e**-d1/ * 1 sub 0 (d1).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESI0E.  | Class(es): C10b1  | Usage: D = DBS0E(D1)  | On-line doc: CALL GAMSDOC DBS0E (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBS1E  Modified (hyperbolic) Bessel function, dbie precision, of special integer order scaled by an exponential, First kind, order one, e**-d1/ * 1 sub 1 (d1).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESI1E.  | Class(es): C10b1  | Usage: D = DBS1E(D1)  | On-line doc: CALL GAMSDOC DBS1E (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBS0E  Modified (hyperbolic) Bessel function, dbie precision, of special integer order scaled by an exponential, Third kind, order zero, e**d1 * K sub 0 (d1).  | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is BESK0E.  | Class(es): C10b1  | Usage: D = DBS0E(D1)  | On-line doc: CALL GAMSDOC DBS0E (or @PRT CMLIB*DOC.SUMMARY/FNLIB)  | Access: LIB NBS*CMLIB

DBSK1E  Modified (hyperbolic) Bessel function, dbie precision, of special integer order scaled by an exponential, Third kind, order one, e**d1 *
January 1984  GAMS: Module Dictionary  C 39

K sub 1 (d1).  | Portable double precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Single precision version is BESK1E.  | Class(es): C10b1 | Usage: D = DBSK1E (D1) | On-line doc: CALL GAMSDOC DBSK1E (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DBSKES Sequences of Bessel functions scaled by an exponential. Modified 3rd kind. e**x * K sub v+i (d), /n/ vals. computed. where i=0,1,..n for n>0, i=0,1,..n-1 for n=0.  | Portable double precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Single precision version is BESKES.  | Class(es): C10b3 | Usage: CALL DBSKES(NU,D,N,DBK) | On-line doc: CALL GAMSDOC DBSKES (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


DBSPLI Evaluates at a given set of points in a specified mesh interval, basis splines together with selected orders of derivatives.  | Proprietary double precision Fortran subroutine in FORT library. Single precision version is BSPLI.  | Class(es): E3 K6 | Usage: CALL DBSPLI(I,K,T,N,X,?LEFT,?D,?NID,?BX) | On-line doc: CALL GAMSDOC DBSPLI (or @PRT PORT+DOC.DBSPLI) | Access: LIB NBS+PORT


DBBSVD Calculates values and derivatives of order less than NDERIV of all B-spline basis functions which do not vanish at X.  | Portable double precision Fortran subroutine in DSBPLINE sublibrary of CMLIB library. Single precision version is BSVD.  | Class(es): E3 K6 | Usage: CALL DBBSVD(T,K,NDERIV,X,?LEFT,LDVNIK,VNIKX,WORK) | On-line doc: CALL GAMSDOC DBBSVD (or @PRT CMLIB+DOC.DBSPLINE and CMLIB+DOC.SUMMARY/FNLIB) | Tests: CMLIB+TEST-SOURCE.$F/DBSPLINE | Access: LIB NBS+CMLIB


DBBSVQ Converts from B-spline to piecewise polynomial representation.  | Portable double precision Fortran subroutine in DSBPLINE sublibrary of CMLIB library. Single precision version is BSPP.  | Class(es): E3 K6 | Usage: CALL DBBSVQ(T,?LEFT,?NIVKX,?IWORK) | On-line doc: CALL GAMSDOC DBBSVQ (or @PRT CMLIB+DOC.DBSPLINE and CMLIB+DOC.SUMMARY/FNLIB) | Tests: CMLIB+TEST-SOURCE.$F/DBSPLINE | Access: LIB NBS+CMLIB

DBBSVPN Calculates the value of all (possibly) nonzero B-spline basis functions at X of a given order.  | Portable double precision Fortran subroutine in DSBPLINE sublibrary of CMLIB library. Single precision version is BSVPN.  | Class(es): E3 K6 | Usage: CALL DBBSVPN(T,JHIGH,K,INDEX,X,?LEFT,?NIVKX,WORK,?IWORK) | On-line doc: CALL GAMSDOC DBBSVPN (or @PRT CMLIB+DOC.DBSPLINE and CMLIB+DOC.SUMMARY/FNLIB) | Tests: CMLIB+TEST-SOURCE.$F/DBSPLINE | Access: LIB NBS+CMLIB

DBBSQP Computes the integral of a B-spline from X1 to X2. The B-spline must be in B-spline representation.  | Portable double precision Fortran subroutine in DSBPLINE sublibrary of CMLIB library. Single precision version is BSQAD.  | Class(es): H2a2a1 E3 K6 | Usage: CALL DBBSQP(T,COEF,N,K,X1,X2,BQUAD,WORK) | On-line doc: CALL GAMSDOC DBBSQP (or @PRT CMLIB+DOC.DBSQAD and CMLIB+DOC.SUMMARY/FNLIB) | Tests: CMLIB+TEST-SOURCE.$F/DBSPLINE | Access: LIB NBS+CMLIB

DBBURAM Finds the best uniform rational approximation to a given function on a specified mesh.  | Proprietary double precision Fortran subroutine in FORT library. Single precision version is BURAM.  | Class(es): K2 | Usage: CALL BURAM (NPTS, MESH, FN, M, N, P, Q, DELK) | On-line doc: CALL GAMSDOC BURAM (or @PRT PORT+DOC.BURAM) | Access: LIB NBS+PORT | See also: DTCHBP

DBBURM Finds best uniform rational approximation to a given function on a specified mesh, starting from a given initial approximation.  | Proprietary double precision Fortran subroutine in FORT library. Single precision version is BURM1.  | Class(es): K2 | Usage: CALL BURM1 (NPTS, MESH, FN, MAXITR, ITOL, M, N, P, Q, DELK) | On-line doc: CALL GAMSDOC BURM1 (or @PRT PORT+DOC.BURM1) | Access: LIB NBS+PORT | See also: DTCHBP


GAMSDOC DCDRE (or @PRT IMSL+DOC.DCDRE) | Access: LIB NBS+IMSL

DCBRT Double precision cube root of a double precision argument. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is CBRT. | Class(es): C2 | Usage: D = DCBRT (D1) | On-line doc: CALL GAMSDOC DCBRT (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


DCEIL Finds the smallest integer greater than or equal to x. Input and output are double precision. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CEIL. | Class(es): C1 | Usage: D = DCEIL (X) | On-line doc: CALL GAMSDOC DCEIL (or @PRT PORT+DOC.DCEIL) | Access: LIB NBS+PORT


DCHU Double precision confluent hypergeometric, U(α,β,d,b,d). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is CHU. | Class(es): C11 | Usage: D1 = DCHU (DA,DA,B,D) | On-line doc: CALL GAMSDOC DCHU (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


DCLST2 Finds the least squares solution of a complex linear algebraic system (double precision) of equations A·X = B. B may be a matrix. | Portable double precision Fortran subprogram in PORT library. Single precision version is CLST2. | Class(es): D0 | Usage: CALL DCLST2 (MDM,N,DM,N,N,R,A,AL,AR,BL,BL,NB,XR,XI) | On-line doc: CALL GAMSDOC DCLST2 (or @PRT PORT+DOC.DCLST2) | Access: LIB NBS+PORT

DCOPY Copies a vector X to a vector Y, both double precision. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SCOPY. | Class(es): D1a6 | Usage: CALL DCOPY (N,DX,INCX,DY,INCY) | On-line doc: CALL GAMSDOC DCOPY (or @PRT CMLIB+DOC.DCOPY/BLAS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

DCOSD Double precision cosine of degree in degrees, cos(d1). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is COSD. | Class(es): C4a | Usage: D = DCOSD (D1) | On-line doc: CALL GAMSDOC DCOSD (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DCOSH Computes hyperbolic cosine, cosh(x). | Proprietary double precision Fortran subprogram in PORT library. Single precision version is COSH. | Class(es): C4c | Usage: D = DCOSH (X) | On-line doc: CALL GAMSDOC DCOSH (or @PRT PORT+DOC.DCOSH) | Access: LIB NBS+PORT

DCOT Double precision cotangent, cot(d1). | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is COT. | Class(es): C4a | Usage: D = DCOT (D1) | On-line doc: CALL GAMSDOC DCOT (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DCPOLY Finds zeros of polynomial with double precision complex coefficients. Uses real double precision arrays to represent complex numbers. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is CPOLY. | Class(es): Flab | Usage: CALL DCPOLY (DEGREE,OPR,OP1,OP2,OR0,OR1,OR2) | On-line doc: CALL GAMSDOC DCPOLY (or @PRT PORT+DOC.DCPOLY) | Access: LIB NBS+PORT

DCSEVL Evaluates an n term series of Chebyshev polynomials at a given point. | Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is CSEVL. | Class(es): C3a2 | Usage: D = DCSEVL (X,D,X,D,C,N) | On-line doc: CALL GAMSDOC DCSEVL (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB | See also: NITSBS

DCSEVU Cubic spline first and second derivative evaluator. | Proprietary single precision Fortran subprogram in IMSL library. |
GAMS: Module Dictionary

January 1984

Class(es): E3 K6  |  Usage: CALL DGSEVP(U, X, Y, C, U, DS, M1, DD, M2, IER)  |  On-line doc: CALL GAMSDOC DGSEVP (or @PRT IMSL+DOC.DGSEVP)  |  Access: LIB NBS+IMSL

DCSPD1  Finds a numerical approximation to the first derivative at requested points in given input data by using spline interpolation.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is CSPD1.  
|  Class(es): H1  |  Usage: CALL DCSPD1 (X, Y, N, XX, YY, YPP, NN)  |  On-line doc: CALL GAMSDOC DCSPD1 (or @PRT PORT+DOC.DCSPD1)  |  Access: LIB NBS+PORT

DCSPFE  Evaluates a cubic spline function which has already been fit to n data pairs (x, y) by DCSPFI.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is CSPFE.  
|  Class(es): E3 K6  |  Usage: CALL DCSPFE (X, Y, YP, NP, N, XX, YY, NN)  |  On-line doc: CALL GAMSDOC DCSPFE (or @PRT PORT+DOC.DCSPFE)  |  Access: LIB NBS+PORT

DCSPFI  Fits a cubic spline function to n input data pairs (x, y), Produces and interpolatory, least squares fits.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is CSPFI.  
|  Class(es): E3a  |  Usage: CALL DCSPFI(X, Y, N, B, YP, YPP)  |  On-line doc: CALL GAMSDOC DCSPFI (or @PRT PORT+DOC.DCSPFI)  |  Access: LIB NBS+PORT

DCSPIN  Interpolates at requested points in given input data using a spline approximation. Interpolation, least squares fits.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is CSPIN.  
|  Class(es): E3a  |  Usage: CALL DCSPIN (X, Y, N, XX, YY, NN)  |  On-line doc: CALL GAMSDOC DCSPIN (or @PRT PORT+DOC.DCSPIN)  |  Access: LIB NBS+PORT

DCSQQU  Finds the integral of a function defined by pairs (x, y) of input points.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is CSPQU.  
|  Class(es): H2a1b2  |  Usage: CALL DCSQQU (X, Y, N, XLOW, XHIGH, ANS)  |  On-line doc: CALL GAMSDOC DCSQQU (or @PRT PORT+DOC.DCSQQU)  |  Access: LIB NBS+PORT

DCSQDU  Cubic spline quadrature.  
|  Proprietary single precision Fortran program in IMSL library.  
|  Class(es): H2a2a1 E3 K6  |  Usage: CALL DCSQDU (X, Y, N, X, C, A, B, Q, IER)  |  On-line doc: CALL GAMSDOC DCSQDU (or @PRT IMSL+DOC.DCSQDU)  |  Access: LIB NBS+IMSL

DDASSL  Solves the system of differential/algebraic equations of the form \( g(t, y) = 0 \), with given initial values.  
|  Portable double precision Fortran program in DDASSL sublibrary of CMLIB library. Single precision version is SDASSL.  
|  Class(es): I1a1b  |  Usage: CALL DDASSL (RES, NEQ, T, YPRIME, TOUT, INFO, RTOL, ATOL, IDID, RWORK, LRW, IWORK, LIW, RPAR, JAC)  |  On-line doc: CALL GAMSDOC DDASSL (or @PRT PORT+DOC.DDASSL)  |  Tests: CMLIB+TEST-SOURCE.DDASSL/SDASSL  |  Access: LIB NBS+CMLIB

DDAWS  Double precision Dawson function \( e^{**(-d1**2)} \) * the integral from 0 to d1 of \( e^{**(t**2)}dt \).  
|  Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is DAWLS.  
|  Class(es): C8c  |  Usage: D = DDAWS (D1)  |  On-line doc: CALL GAMSDOC DDAWS (or @PRT PORT+DOC.DDAWS/FNLIB)  |  Access: LIB NBS+CMLIB

DDL2S  Fits discrete data with a B-spline of order K, by least squares.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is DL2SF.  
|  Class(es): K1a1a1  |  Usage: CALL DDL2SF (X, Y, N, K, T, N, A)  |  On-line doc: CALL GAMSDOC DDL2SF (or @PRT PORT+DOC.DDL2SF)  |  Access: LIB NBS+PORT  |  Tests: also: DSNLPL DSNLPD DSNPNI DSNPLN

DDL2SW  Fits discrete data with a B-spline of order k, by weighted least squares.  
|  Proprietary double precision Fortran program in PORT library. Single precision version is DL2SW.  
|  Class(es): K1a1a1  |  Usage: CALL DDL2SW (X, Y, N, K, T, N, A)  |  On-line doc: CALL GAMSDOC DDL2SW (or @PRT PORT+DOC.DDL2SW)  |  Access: LIB NBS+PORT  |  Tests: also: DSNLPL DSNLPD DSNPNI DSNPLN

DDOT  Compute double precision dot product.  
|  Portable double precision Fortran program in BLAS sublibrary of CMLIB library. Single precision version is DDOT.  
|  Class(es): D1a4  |  Usage: CALL DDOT (N, DX, INCX, DY, INCY)  |  On-line doc: CALL GAMSDOC DDOT (or @PRT CMLIB+DOC.DDOT/BLAS)  |  Tests: CMLIB+TEST-SOURCE.DQ/BLAS  |  Access: LIB NBS+CMLIB

DDRIV1  Numerical Integration, Initial Value Problems, Ordinary Differential Eqns., Gear Stiff Formulas, Easy to Use.  
|  Portable double precision Fortran program in DDRIV sublibrary of CMLIB library. Single precision version is SDRIV1.  
|  Class(es): IIa2 IIa1b  |  Usage: CALL DDRIV1 (N, T, Y, TOUT, MSTATE, EPS, WORK, LENW)  |  On-line doc: CALL GAMSDOC DDRIV1 (or @PRT CMLIB+DOC.SUMMARY/DDRIV)  |  Tests: CMLIB+TEST-SOURCE.DF/DDRIV, CMLIB+TEST-SOURCE.DQ/DDRIV  |  Access: LIB NBS+CMLIB

|  Portable double precision Fortran program in DDRIV sublibrary of CMLIB library. Single precision version is SDRIV2.  

|  Portable double precision Fortran program in DDRIV sublibrary of CMLIB library. Single precision version is SDRIV3.  

DE1  Double precision exponential integral, the integral from d1 to infinity of \( e^{**(-t/t)} \) dt.  
|  Portable double precision Fortran program in FNLIB sublibrary of CMLIB library. Single precision version is E1.  
|  Class(es): C5  |  Usage: D = DE1 (D1)  |  On-line doc: CALL GAMSDOC DE1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB)  |  Access: LIB NBS+CMLIB

DEABM  Solves a system of first order ordinary differential equations with arbitrary initial conditions by a predictor-corrector method.  
|  Portable single precision Fortran program in DEPAC sublibrary of CMLIB library.  
|  Class(es): IIa1b  |  Usage: CALL
GAMS® Module Dictionary

January 1984

DEABM (F, NEQ, T, YOUT, INFO, RTOL, ATOL, IDID, RWORK, LRW, IWORK, LIW, RPAR, IPAR) | Online doc: CALL GAMSDOC DEABM (or @PRT CMLIB+DOC.DEABM/DEPAC) | Tests: CMLIB+TEST-SOURCE.DEABM/DEPAC | Access: Lib NBS+CMLIB

DEBDF Solves a system of first order stiff ordinary differential equations with arbitrary initial conditions by Gear's method. Portable single precision Fortran subprogram in DEPAC sublibrary of CMLIB library. | Class(es): 11a2 | Usage: CALL DEBDF (F, NEQ, T, YOUT, INFO, RTOL, ATOL, IDID, RWORK, LRW, IWORK, LIW, RPAR, IPAR, IAC) | Online doc: CALL GAMSDOC DEBDF (or @PRT CMLIB+DOC.DEBDF/DEPAC) | Tests: CMLIB+TEST-SOURCE.DEBDF/DEPAC | Access: Lib NBS+CMLIB

DEEB8I Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. Proprietary double precision Fortran subprogram in PORT library. Single precision version is EEB8F. | Class(es): E3 K6 | Usage: D = DEEB8F (K, T1, N1, A1, T2, N2, A2) | Online doc: CALL GAMSDOC DEEB8I (or @PRT PORT+DOC.DEEB8I) | Access: Lib NBS+PORT

DEEBSF Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. Proprietary double precision Fortran subprogram in PORT library. Single precision version is EEBSI. | Class(es): E3 K6 | Usage: X = EEBSI (K, T1, N1, A1, T2, N2, A2, X, XEST1, XEST2) | Online doc: CALL GAMSDOC DEEBSF (or @PRT PORT+DOC.DEEBSF) | Access: Lib NBS+PORT

DEEESF Finds the maximum absolute error in a given B-spline fit to a function, f. Proprietary double precision Fortran subprogram in PORT library. Single precision version is EESFF. | Class(es): E3 K6 | Usage: D = DEEESF (K, T, N, A, F) | Online doc: CALL GAMSDOC DEEESF (or @PRT PORT+DOC.DEEESF) | Access: Lib NBS+PORT

DEEESI Finds the maximum absolute error in a given B-spline fit to a function, f, on a set of user selected intervals. Proprietary double precision Fortran subprogram in PORT library. Single precision version is EESFI. | Class(es): E3 K6 | Usage: D = DEEESI (K, T, N, A, X, XEST) | Online doc: CALL GAMSDOC DEEESI (or @PRT PORT+DOC.DEEESI) | Access: Lib NBS+PORT

DEFINE Defines a vector of constants by setting all of the elements in the single precision vector X equal to XNEW. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a1 | Usage: CALL DEFINE (X, N, XNEW) | Online doc: CALL GAMSDOC DEFINE (or @PRT DATAPAC+DOC.DEFINE) | Access: Lib NBS+DATAPAC

DEI Double precision exponential integral, the integral from -d1 to infinity of e**t / t dt. Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is EI. | Class(es): C5 | Usage: D = DEI (D1) | Online doc: CALL GAMSDOC DEI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: Lib NBS+CMLIB

DEIGEN Finds all eigenvalues and eigenvectors of a real matrix. Output consists of pairs of real arrays. Proprietary double precision Fortran subprogram in PORT library. Single precision version is EIGEN. | Class(es): D4a2 | Usage: CALL DEIGEN (NM, N, A, WR, WI, Z) | Online doc: CALL GAMSDOC DEIGEN (or @PRT PORT+DOC.DEIGEN) | Access: Lib NBS+PORT

DELETE Deletes all observations in the vector X which are inside the interval [XMIN, XMAX]. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d | Usage: CALL DELETE (X, N, XMIN, XMAX, NEWN) | Online doc: CALL GAMSDOC DELETE (or @PRT DATAPAC+DOC.DELETE) | Access: Lib NBS+DATAPAC

DEMOD Performs a complex demodulation on the data in the input vector X at the input demodulation frequency = F. | Class(es): L10d | Usage: CALL DEMOD (X, N, F) | Online doc: CALL GAMSDOC DEMOD (or @PRT DATAPAC+DOC.DEMOD) | Access: Lib NBS+DATAPAC

DERF Double precision error function, (2 / sqrt(pi)) * the integral from 0 to d1 of e**(-x**2 / 2) dt. Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ERF. | Class(es): C8 L8a1 | Usage: D = DERF (D1) | Online doc: CALL GAMSDOC DERF (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: Lib NBS+CMLIB

DERC Double precision complementary error function, (2 / sqrt(pi)) * the integral from d1 to infinity of e**(-x**2 / 2) dt. Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is ERF. | Class(es): C8 L8a1 | Usage: D = DERC (D1) | Online doc: CALL GAMSDOC DERC (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: Lib NBS+CMLIB

DERKF Solves a system of first order ordinary differential equations with arbitrary initial conditions by a Runge-Kutta method. Portable single precision Fortran subprogram in DEPAC sublibrary of CMLIB library. | Class(es): I1a1 | Usage: CALL DERKF (F, NEQ, T, YOUT, INFO, RTOL, ATOL, IDID, RWORK, LRW, IWORK, LIW, RPAR, IPAR) | Online doc: CALL GAMSDOC DERKF (or @PRT CMLIB+DOC.DERKF/DEPAC) | Tests: CMLIB+TEST-SOURCE.DERKF/DEPAC | Access: Lib NBS+CMLIB

DEXCDF Computes the cumulative distribution function value for the double exponential (Laplace) distribution with mean = 0. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a1d | Usage: CALL DEXCDF (X, CDF) | Online doc: CALL GAMSDOC DEXCDF (or @PRT DATAPAC+DOC.DEXCDF) | Access: Lib NBS+DATAPAC

DEXINT Computes sequences of exponential integrals E(N+K,X) K=0,...,M-1 or EXP(X) times same to specified tolerance. Portable double precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. Single precision version is EXINT. | Class(es): C5 | Usage: CALL DEXINT (X, N, KODE, TOL, EN, IERR) | Online doc: CALL GAMSDOC DEXINT (or @PRT CMLIB+DOC.DEXINT/AMOSLIB) | Tests: CMLIB+TEST-SOURCE.DEXINT/AMOSLIB | Access: Lib NBS+CMLIB

DEXPDF Computes the probability density function value for the double exponential (Laplace) distribution with mean = 0. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a1d | Usage: CALL DEXPDF (X, PDF) | Online doc: CALL GAMSDOC DEXPDF (or @PRT DATAPAC+DOC.DEXPDF) | Access: Lib NBS+DATAPAC

DEXPLT Generates a double exponential (Laplace) probability plot with mean = 0 and standard deviation = sqrt(2). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a1d | Usage: CALL DEXPLT (X, N) | Online doc: CALL GAMSDOC DEXPLT (or @PRT DATAPAC+DOC.DEXPLT) | Access: Lib NBS+DATAPAC
DEXPF \text{ Computes the percent point function value for the double exponential (Laplace) distribution with mean = 0. Portal double precision Fortran subprogram in DATAPAC library. Class(es): L5a2d Usage: CALL DEXPF(P,PPF) On-line doc: CALL GAMSDOC DEXPF (or @VRT DATAPAC+DOC.DEXPF) Access: LIB NBS*DATAPAC}

DEXPR \text{ Double precision relative error exponential from first order, ((e**d1) - 1) / x. Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is EXPRRL. Class(es): C4b Usage: D = DEXPR(D) On-line doc: CALL GAMSDOC DEXPR (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB}

DEXRA \text{ Generates a random sample of size N from the double exponential (Laplace) distribution with mean = 0 and standard deviation = sqrt(2). Portal single precision Fortran subprogram in DATAPAC library. Class(es): L6a4 Usage: CALL DEXRA(N,ISTART,X) On-line doc: CALL GAMSDOC DEXRA (or @VRT DATAPAC+DOC.DEXRA) Access: LIB NBS*DATAPAC}

DEXSF \text{ Computes the sparsity function value for the double exponential (Laplace) distribution with mean = 0 and standard deviation = sqrt(2). Portal single precision Fortran subprogram in DATAPAC library. Class(es): L5a2d Usage: CALL DEXSF(P,SF) On-line doc: CALL GAMSDOC DEXSF (or @VRT DATAPAC+DOC.DEXSF) Access: LIB NBS*DATAPAC}

DFAC \text{ Double precision factorial, ni. Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is FACT. Class(es): Cl Usage: D = DFAC(N) On-line doc: CALL GAMSDOC DFAC (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB}

DFFT \text{ Computes FFT of complex data sequence (forward or reverse) any number of points. Useful for multivariate transforms. Uses no complex arithmetic. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFT. Class(es): J1a2 11b Usage: CALL DFPT(A,B,NTOT,N,NSPAN,ISN) On-line doc: CALL GAMSDOC DFFTT (or @VRT PORT+DOC.DFFTT) Access: LIB NBS*PORT See also: DFFTC}

DFFTC \text{ Mixed radix fast Fourier transform of complex data. Two arrays used for complex data. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTC. Class(es): J1a2 Usage: CALL DFFTC(N,AR,AL) On-line doc: CALL GAMSDOC DFFTC (or @VRT PORT+DOC.DFFTC) Access: LIB NBS*PORT See also: DFFTCI}

DFFTCI \text{ Finds the inverse fast Fourier transform, given the Fourier coefficients in the frequency domain. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTCI. Class(es): J1a2 Usage: CALL DFFTCI(N,FR,FI) On-line doc: CALL GAMSDOC DFFTCI (or @VRT PORT+DOC.DFFTCI) Access: LIB NBS*PORT See also: DFFTC}

DFFTR \text{ Mixed radix fast Fourier transform to find the transform of 2N real data points. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTR. Class(es): J1a1 Usage: CALL DFFTR(NNP2,A,B) On-line doc: CALL GAMSDOC DFFTR (or @VRT PORT+DOC.DFFTR) Access: LIB NBS*PORT See also: DFFTRI}

DFFTRI \text{ Finds the inverse Fourier transform using Fourier coefficients assumed to arise from real data in the time domain. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FFTRI. Class(es): J1a1 Usage: CALL DFFTRI(NN,FR,FI) On-line doc: CALL GAMSDOC DFFTRI (or @VRT PORT+DOC.DFFTRI) Access: LIB NBS*PORT See also: DFFTR}

DFLR \text{ Finds the largest integer less than or equal to x. Input and output are double precision. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FLR. Class(es): Cl Usage: D = DFLR(X) On-line doc: CALL GAMSDOC DFLR (or @VRT PORT+DOC.DFLR) Access: LIB NBS*PORT}

DFMIN \text{ Finds an approximate local minimum of a univariate user defined EXTERNAL function, f. Proprietary double precision Fortran subprogram in PORT library. Single precision version is FMIN. Class(es): G1a2 Usage: D = DFMIN(F,X,A,B,T) On-line doc: CALL GAMSDOC DFMN (or @VRT PORT+DOC.DFMN) Access: LIB NBS*PORT}

DGAM1 \text{ Double precision incomplete gamma fn., the integral from 0 to d of (t**da-1) * e**-t dt. Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMI. Class(es): C7e Usage: D1 = DGAM1(DA,D) On-line doc: CALL GAMSDOC DGAMI (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB}

DGAMIC \text{ Double precision complementary incomplete gamma function, the integral from d to infinity of (t**da-1) * e**-t dt. Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMIC. Class(es): C7e Usage: D1 = DGAMIC(DA,D) On-line doc: CALL GAMSDOC DGAMIC (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB}

DGAMIT \text{ Double precision Tricomi's incomplete gamma function, d**-da * incomplete gamma (da,d) / gamma (da). Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMIT. Class(es): C7e Usage: D1 = DGAMIT(DA,D) On-line doc: CALL GAMSDOC DGAMIT (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB}

DGAMMA \text{ Double precision gamma function. Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMMA. Class(es): C7a Usage: D1 = DGAMMA(D) On-line doc: CALL GAMSDOC DGAMMA (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB See also: DGAML M}

DGAMMR \text{ Double precision reciprocal gamma function 1 / Gamma(d). Portal double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is GAMMR. Class(es): C7a Usage: D1 = DGAMMR(D) On-line doc: CALL GAMSDOC DGAMMR (or @VRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS*CMLIB}

DGASAQ \text{ Finds the absissae and weights for Gauss Legendre quadrature on the interval (a, b). Proprietary double precision Fortran subprogram in PORT library. Single precision version is GASAQ. Class(es): H2z Usage: CALL DGASAQ(N,A,B,C,NU,X,W) On-line doc: CALL GAMSDOC DGASAQ (or @VRT PORT+DOC.DGASAQ) Access: LIB NBS*PORT}

DGBCO \text{ Computes LU factorization of general double precision band matrix and estimates its condition. Portal double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SCBO. Class(es):


DGQ01N Finds the abscissae and weights for Gauss Laguerre quadrature on the interval (0, +infinity). | Proprietary double precision Fortran subroutine in PORT library. Single precision version is DGQ01N. | Class(es): H2c | Usage: CALL DGQ01N(N,X,W) | On-line doc: CALL GAMSDOC DGQ01N (or OPRT PORT+DOCD.DGQ01N) | Access: LIB NBS+PORT


DHUMSL Minimizes a general unconstrained objective function using (analytic) gradient and Hessian provided by the user. (Double precision version of HUMSL.) | Portable double precision Fortran subroutine in DLNLSN library. Single precision version is HUMSL. | Class(es): G1b1c | Usage: CALL DHUMSL(N,D,X,CALCF,CALCGH,IV,LIV,LY,V,UIPARM,UFPARM,UPFARM) | On-line doc: CALL GAMSDOC DHUMSL (or OPRT CMLIB+DOCD.DHUMSL/DLNLSN) | Tests: CMLIB+TEST-SOURCE.$F2/DLNLSN, CMLIB+TEST-SOURCE.$F2/DLNLSN | Access: LIB NBS+CMLIB

DIFFERENCES Computes differences between observations at a specified lag in a time series. | Command in MINITAB Proprietary interactive dictionary.
DINTRY Computes the index into a knot or breakpoint sequence corresponding to a given point X. Portable double precision Fortran subprogram in DBSPLINE library of CMLIB library. Single precision version is INTRV. Class(es): E3 K6 Portable: Usage: CALL DINTRY(KT,LXT,X,ILD,LFLAG) On-line doc: CALL GAMSDOC DINTRY (or @PRT CMLIB.DOCD.BSPLINE) Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

DISCR2 Discretizes the data in the vector X into NUMCLA classes. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L2a Usage: CALL DISCR2(X,N,NUMCLA,Y) On-line doc: CALL GAMSDOC DISCR2 (or @PRT DATAPAC+DOC.DISCR2) Access: LIB NBS+DATAPAC

DISCRE Discretizes the data of the vector X according to class width. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L2a Usage: CALL DISCRE(X,N,XMIN,XDEL,XMAX,Y) On-line doc: CALL GAMSDOC DISCRE (or @PRT DATAPAC+DOC.DISCRE) Access: LIB NBS+DATAPAC

DL2SF Fits discrete data with a B-spline of order K, by least squares. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DDL2SF. Class(es): Klalal Usage: CALL DL2SF(X,Y,NXY,K,T,NT,A) On-line doc: CALL GAMSDOC DL2SF (or @PRT PORT+DOC.DL2SF) Access: LIB NBS+PORT See also: DSPLNE DSPLNI DSPLN1 DSPLN2 DSPLND

DL2SFF Obtains a weighted least square expansion of a known function in terms of B-splines of order K, at given mesh points. Proprietary double precision Fortran subprogram in PORT library. Single precision version is L2SFF. Class(es): Klalal Usage: CALL DL2SFF(FW,K,T,NT,A) On-line doc: CALL GAMSDOC DL2SFF (or @PRT PORT+DOC.DL2SFF) Access: LIB NBS+PORT See also: DSPLNE DSPLND DSPLNI DSPLN1 DSPLN2 DEEBSF DEESBF DEE5SF DEES5F

DL2SFH Obtains a weighted least square expansion of a known function in and its derivatives in terms of B-splines of order K at given mesh points. Proprietary double precision Fortran subprogram in PORT library. Single precision version is L2SFH. Class(es): Klalal Usage: CALL DL2SFH(FW,MD,K,T,NT,A) On-line doc: CALL GAMSDOC DL2SFH (or @PRT PORT+DOC.DL2SFH) Access: LIB NBS+PORT See also: DSPLNE DSPLND DSPLNI DSPLN1 DSPLN2

DL2SW Fits discrete data with a B-spline of order K, by least squares. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DDL2SW. Class(es): Klalal Usage: CALL DL2SW(X,Y,NXY,K,T,NT,A) On-line doc: CALL GAMSDOC DL2SW (or @PRT PORT+DOC.DL2SW) Access: LIB NBS+PORT See also: SPLNE SPLND SPLNI SPLN1 SPLN2

DLBETA Double precision Log Beta function, in B(a,b). Portable double precision Fortran subprogram in FNLIB library of CMLIB library. Single precision version is ALBETA. Class(es): C7b Usage: D = DLBETA(A,B) On-line doc: CALL GAMSDOC DLBETA (or @PRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS+CMLIB

DLGAMS Double precision Log ab gamma with sign of gamma, G = \ln ab Gamma(d) , S = sign Gamma(d). Portable double precision Fortran subprogram in FNLIB library of CMLIB library. Single precision version is ALGAMS. Class(es): C7a Usage: CALL DLGAMS (D,G,S) On-line doc: CALL GAMSDOC DLGAMS (or @PRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS+CMLIB

DLI Double precision logarithmic integral. Portable double precision Fortran subprogram in FNLIB library of CMLIB library. Single precision version is ALI. Class(es): C5 Usage: D1 = DLI(D) On-line doc: CALL GAMSDOC DLI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS+CMLIB

DLINEQ Solves a real system of linear equations, AX=B, where B is allowed to be a matrix or a vector. Proprietary double precision Fortran subprogram in PORT library. Single precision version is LINEQ. Class(es): D2a Usage: CALL DLINEQ (N,A,B,NB,X) On-line doc: CALL GAMSDOC DLINEQ (or @PRT PORT+DOC.DLINEQ) Access: LIB NBS+PORT

DLNGAM Double precision Log ab gamma, ln of absolute value of Gamma(d). Portable double precision Fortran subprogram in FNLIB library of CMLIB library. Single precision version is ALNGAM. Class(es): C7a Usage: CALL DLNGAM (D) On-line doc: CALL GAMSDOC DLNGAM (or @PRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS+CMLIB

DNLREL Double precision relative error logarithm, ln(1 + d). Portable double precision Fortran subprogram in FNLIB library of CMLIB library. Single precision version is ALNREL. Class(es): C4b Usage: D1 = DNLREL (D) On-line doc: CALL GAMSDOC DNLREL (or @PRT CMLIB+DOC.SUMMARY/FNLIB) Access: LIB NBS+CMLIB

DLSTSQ Finds the least squares solution of a system of linear equations, AX=B. May be a matrix. Proprietary double precision Fortran subprogram in PORT library. Single precision version is LSTSQ. Class(es): D9 Usage: CALL DLSTSQ (MDIM,NDIM,M,N,A,B,ND)
C 46
GAMS\$ Module Dictionary
January 1984

| On-line doc: CALL GAMSDOC DLTSQ (or \$PORT PORT+DOC.DLTSQ) | Access: LIB NBS+PORT |

DLUMB Given a basic mesh, this subdivides each interval uniformly for B-spline use. Multiplicities are allowed. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is LUMB. | Class(es): E3 K6 | Usage: CALL DLUMB (X,NXB,N,K,X,NX) | On-line doc: CALL GAMSDOC DLUMB (or \$PORT PORT+DOC.DLUMB) | Access: LIB NBS+PORT |

DLUMD Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): E3 K6 | Usage: CALL DLUMD (X,NXB,N,X,NX) | On-line doc: CALL GAMSDOC DLUMD (or \$PORT PORT+DOC.DLUMD) | Access: LIB NBS+PORT |

DMLIN Numerical integration of a function of several variables over a hyper-rectangle (Gaussian method). | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): H2b1a1 | Usage: R = DMLIN(F,A,B,N,MAXFCN,AERR,RERR,IER) | On-line doc: CALL GAMSDOC DMLIN (or \$PORT IMSL+DOC.DMLIN) | Access: LIB NBS+IMSL |

DMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is MNPB. | Class(es): E3 K8 | Usage: CALL DMNPB (X,N,X,N,K,T,NT) | On-line doc: CALL GAMSDOC DMNPB (or \$PORT PORT+DOC.DMNPB) | Access: LIB NBS+PORT |


DNL2S1 Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. (Double precision version of NLS21.) | Portable double precision Fortran subroutine in DNL2S2N sublibrary of CMLIB library. Single precision version is NLS21. | Class(es): L8g1b L8g2b K1b1a2 | Usage: CALL DNLS2S1(N,P,X,CALCR,CALCI,V,LIV,LV,V,UIPARM,UPARM,UPPAR) | On-line doc: CALL GAMSDOC DNLS2S1 (or \$PORT CMLIB+DOC.DNLS2S1/DNL2S2N) | Tests: CMLIB+TEST-SOURCE.*F1/DNL2S2N, CMLIB+TEST-SOURCE.*Q1/DNL2S2N | Access: LIB NBS+CMLIB |

DNL2SN Minimizes a nonlinear sum of squares using residual values only. (Double precision version of NLS2SN.) | Portable double precision Fortran subroutine in DNL2SN sublibrary of CMLIB library. Single precision version is NL2SN. | Class(es): L8g1a L8g2a K1b1a1 | Usage: CALL DNLS2SN(N,P,X,CALCR,CALCI,V,LIV,LV,V,UIPARM,UPARM,UPPAR) | On-line doc: CALL GAMSDOC DNLS2SN (or \$PORT CMLIB+DOC.DNLS2SN/DNL2SN) | Tests: CMLIB+TEST-SOURCE.*F1/DNL2SN, CMLIB+TEST-SOURCE.*Q1/DNL2SN | Access: LIB NBS+CMLIB |


DNRM2 Finds the length (Euclidean norm) of a vector, without underflow or overflow. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is SNRMR2. | Class(es): D1a3b | Usage: D = DNRM2(N,INCX) | On-line doc: CALL GAMSDOC DNRM2 (or \$PORT PORT+DOC.DNRM2) | Access: LIB NBS+PORT |

DODEQ Finds the integral of a set of functions over the same interval by using the differential equation solver ODES1. For smooth functions. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is ODEQ. | Class(es): H2a1a1 | Usage: CALL DODEQ (N, F, A, B, EPS, ANS) | On-line doc: CALL GAMSDOC DODEQ (or \$PORT PORT+DOC.DOSEQ) | Access: LIB NBS+PORT |


DODES Solves an initial value problem for a system of ordinary differential equations. Allows greater flexibility and user control. | Proprietary
double precision Fortran subroutine in PORT library. Single precision version is ODES1. | Class(es): 11a1c | Usage: CALL DODES1 (F,X,NX,TSTART,TSTOP,DT,ERROR,ERRPAR,HANDLE,GLMAX,ERPUTS) | On-line doc: CALL GAMSDOC DODES1 (or @PRT PORT+DOC.DODES1) | Access: LIB NBS+PORT | See also: DODES DODESE


DODESH Default HANDLE routine for ODES. Used to access the results at the end of each integration time step. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is ODES1. | Class(es): 11c | Usage: D = DORTEP (T0,X0,T1,X1,NX,DT,TSTOP,E) | On-line doc: CALL GAMSDOC DODESH (or @PRT PORT+DOC.DODESH) | Access: LIB NBS+PORT | See also: DODES

DORTPH Evaluates a polynomial expressed as a sum of general orthogonal polynomials. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is ORTPH. | Class(es): C3 | Usage: | On-line doc: CALL GAMSDOC DORTPH (or @PRT PORT+DOC.DORTPH) | Access: LIB NBS+PORT


DPBSL Uses LU factorization of double precision positive definite band matrix to solve systems. | Portable double precision Fortran subroutine in LINPACKD sublibrary of CMLIB library. Single precision version is SPBSL. | Class(es): D2b2 | Usage: CALL DPBSL(ABD,LDA,N,M,B) | On-line doc: CALL GAMSDOC DPBSL (or @PRT CMLIB+DOC.DPBSL/LINPACKD) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKD, CMLIB+TEST-SOURCE.$F2/LINPACKD | Access: LIB NBS+CMLIB | See also: DPBFA


DPFQAD Integrates function times derivative of B-spline from X1 to X2. The B-spline is in piecewise polynomial representation. | Portable double precision Fortran subroutine in DBSPLINE sublibrary of CMLIB library. Single precision version is PFQAD. | Class(es): H9a2a1 E3 K6 | Usage: CALL DPFQAD(F,LDC,XI,LXI,K,ID,X1,X2,TOL,QUAD,IERR) | On-line doc: CALL GAMSDOC DPFQAD (or @PRT CMLIB+DOC.DPFQAD/DBSPLINE and CMLIB+DOC.SUMMARY/DBSPLINE) | Tests: CMLIB+TEST-SOURCE.$F1/DBSPLINE | Access: LIB NBS+CMLIB

DPOCH Double precision Pochhammer's generalized symbol, = Gamma(a+d) / Gamma(a). | Portable double precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Single precision version is PPOCH. | Class(es): C1 C7a | Usage: D1 = DPOCH (A,D) | On-line doc: CALL GAMSDOC DPOCH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

DPOCH1 Double precision Pochhammer's symbol from first order, = (a) sub d -1 / d. | Portable double precision Fortran subroutine in FNLIB sublibrary of CMLIB library. Single precision version is PPOCH1. | Class(es): C1 C7a | Usage: D1 = DPOCH1 (A,D) | On-line doc: CALL GAMSDOC DPOCH1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB


DPDODI Use factorization of double precision positive definite matrix to compute determinant and/or inverse. | Portable double precision Fortran subroutine in LINPACKD sublibrary of CMLIB library. Single precision version is SPDODI. | Class(es): D3b1b | Usage: CALL DPDODI(A,LDA,N,DET,JOB) | On-line doc: CALL GAMSDOC DPDODI (or @PRT CMLIB+DOC.DPDODI/LINPACKD) | Tests: CMLIB+TEST-SOURCE.$F1/LINPACKD, CMLIB+TEST-SOURCE.$F2/LINPACKD | Access: LIB NBS+CMLIB | See also: DPOCO

DQAGI Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAGI. | Class(es): H2a5a1, H2a4a1 | Usage: CALL DQAGI(F,BOUND,INF, EPSABS, EPSREL, RESULT, ABERR, NEVAL, IER, LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAGI (or @PRT CMLIB+DOC.DQAGI/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAGI/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAGIE Automatic integrator for semi-infinite or infinite integrals and general intervals, provides more information than DQAGI. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAGIE. | Class(es): H2a5a1, H2a4a1 | Usage: CALL DQAGIE(F, BOUND, INF, EPSABS, EPSREL, LIMIT, RESULT, ABERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, IORD, LAST) | On-line doc: CALL GAMSDOC DQAGIE (or @PRT CMLIB+DOC.DQAGIE/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAGIE/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAGP Automatic adaptive integrator, allows user to specify location of singularities or difficulties of integrand, uses extrapolation. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAGP. | Class(es): H2a5a1 | Usage: CALL DQAGP(F, A, B, NPTS2, POINTS, EPSABS, EPSREL, RESULT, ABERR, NEVAL, IER, LENW, LENW, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAGP (or @PRT CMLIB+DOC.DQAGP/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAGP/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAGPE Automatic adaptive integrator for function with user specified endpoint singularities, provides more information than DQAGP. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAGPE. | Class(es): H2a5a1 | Usage: CALL DQAGPE(F, A, B, NPTS2, POINTS, EPSABS, EPSREL, LIMIT, RESULT, ABERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, PTS, IORD, LEVEL, NDIN, LAST) | On-line doc: CALL GAMSDOC DQAGPE (or @PRT CMLIB+DOC.DQAGPE/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAGPE/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAGS Automatic adaptive integrator, will handle most non-smooth integrands including those with endpoint singularities, uses extrapolation. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAGS. | Class(es): H2a5a1 | Usage: CALL DQAGS(F, A, B, EPSABS, EPSREL, RESULT, ABERR, NEVAL, IER, LIMIT, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAGS (or @PRT CMLIB+DOC.DQAGS/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAGS/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAGSE Automatic adaptive integrator, can handle integrands with endpoint singularities provides more information than DQAGS. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAGSE. | Class(es): H2a5a1 | Usage: CALL DQAGSE(F, A, B, EPSABS, EPSREL, LIMIT, RESULT, ABERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, IORD, LAST) | On-line doc: CALL GAMSDOC DQAGSE (or @PRT CMLIB+DOC.DQAGSE/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAGSE/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB


DQAWCE Cauchy Principal value integrator, provides more information than DQAWC (real Hilbert transform). | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAWCE. | Class(es): H2a2a1 J4 | Usage: CALL DQAWCE(F, A, B, EPSABS, EPSREL, LIMIT, RESULT, ABERR, NEVAL, IER, ALIST, BLIST, RLIST, ELIST, IORD, LAST) | On-line doc: CALL GAMSDOC DQAWCE (or @PRT CMLIB+DOC.DQAWCE/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWCE/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAWF Automatic integrator for Fourier integrals on (a, infinity) with factors SIN(Omega*x), COS(Omega*x) by integrating between zeros. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAWF. | Class(es): H2a2a1 | Usage: CALL DQAWF(F, A, OMEGA, INTEGER, EPSABS, RESULT, ABERR, NEVAL, IER, LIMIT, LAST, LENW, MAXPI, LENW, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAWF (or @PRT CMLIB+DOC.DQAWF/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWF/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAWF Integral for Fourier integrals with SIN(Omega*x) factor on (A, INFINITY), provides more information than DQAWF. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAWFE. | Class(es): H2a2a1 | Usage: CALL DQAWFE(F, A, OMEGA, INTEGER, EPSABS, LIMIT, LIMIT, LIMIT, MAXPI, RESULT, ABERR, NEVAL, IER, RLIST, ELIST, IORD, LAST, LENW, MAXPI, LENW, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAWFE (or @PRT CMLIB+DOC.DQAWFE/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWFE/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAWO Automatic adaptive integrator for integrands with oscillatory sin or cosine factor. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAWO. | Class(es): H2a2a1 | Usage: CALL DQAWO(F, A, OMEGA, INTEGER, EPSABS, EPSREL, RESULT, ABERR, NEVAL, IER, LENW, MAXPI, LENW, LAST, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAWO (or @PRT CMLIB+DOC.DQAWO/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWO/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB

DQAWOE Automatic integrator for integrands with explicit oscillatory sin or cosine factor, provides more information than DQAWO. | Portable double precision Fortran subroutine in QUADDP library. Single precision version is QAWOE. | Class(es): H2a2a1 | Usage: CALL DQAWOEF(F, A, OMEGA, INTEGER, EPSABS, EPSREL, LIMIT, ICALL, MAXPI, RESULT, ABERR, LIMIT, IWORK, WORK) | On-line doc: CALL GAMSDOC DQAWOEF (or @PRT CMLIB+DOC.DQAWOEF/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWOEF/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: Lib NBS+CMLIB
NEVAL,IER,ALIST,BLIST,RLIST,ELIST,ORDER,NNLOG,MOMCOM,CHEBMO) | On-line doc: CALL GAMSDOC DQAWOE (or @RPT CMLIB+DOC.DQAWOE/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.$Q/QUADDP | Access: LIB NBS+CMLIB

DQAWS Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities. Portable double precision Fortran subroutine in QUADDP sublibrary of CMLIB library. Single precision version is QAWS. | Class(es): H2a2a2 | Usage: CALL DQAWS(F,A,ALPHA,BETA,INTEGR,EPSCS,EPSCS,RESULT,ABSERR,NEVAL,IER,LIMIT,LENW,LAST,3WORK,WORK) | On-line doc: CALL GAMSDOC DQAWS (or @RPT CMLIB+DOC.DQAWS/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWS/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: LIB NBS+CMLIB

DQAWSF Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities, more information than DQAWS. Portable double precision Fortran subroutine in QUADDP sublibrary of CMLIB library. Single precision version is QAWSF. | Class(es): H2a2a2 | Usage: CALL DQAWS(F,A,ALPHA,BETA,INTEGR,EPSCS,EPSCS,RESULT,ABSERR,NEVAL,IER,ALIST,BLIST,RLIST,ELIST,ORDER,LAST) | On-line doc: CALL GAMSDOC DQAWS (or @RPT CMLIB+DOC.DQAWS/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.DQAWS/QUADDP, CMLIB+TEST-SOURCE.$Q/QUADDP | Access: LIB NBS+CMLIB


DQK15L Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error estimate. Portable double precision Fortran subroutine in QUADDP sublibrary of CMLIB library. Single precision version is QK15L. | Class(es): H2a2a2 | Usage: CALL DQK15L(F,P1,P2,P3,P4,KP,A,B,RESULT,ABSERR,RESABS,RESASC) | On-line doc: CALL GAMSDOC DQK15L (or @RPT CMLIB+DOC.DQK15L/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.$Q/QUADDP | Access: LIB NBS+CMLIB


DQK41 Evaluates integral of given function on an interval with a 41 point Gauss Kronrod formula and returns error estimate. |


**DQMOMO** Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QMOMO. | Class(es): H2a1a3 | Usage: CALL DQMOMO(alfa,beta,ri,rg,rh,integer) | On-line doc: CALL GAMSDOC DQMOMO (or @PRT CMLIB+DOC.DQMOMO/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.$Q/QUADDP | Access: LIB NBS+CMLIB

**DQNG** Automatic non-adaptive integrator for smooth functions, using Gauss Kronrod Patterson formulas. | Portable double precision Fortran subprogram in QUADDP sublibrary of CMLIB library. Single precision version is QNG. | Class(es): H2a1a1 | Usage: CALL DQNG(F,A,B,EPSABS,EPSREL,RESULT,ABSE,VA,IER) | On-line doc: CALL GAMSDOC DQNG (or @PRT CMLIB+DOC.DQNG/QUADDP and CMLIB+DOC.SUMMARY/QUADDP) | Tests: CMLIB+TEST-SOURCE.$Q/QUADDP | Access: LIB NBS+CMLIB


**DQUD** Finds the integral of a general user defined EXTERNAL function by an adaptive technique to given absolute accuracy. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is QUAD. | Class(es): H2a1a1 | Usage: CALL DQUD (F,A,B,EPS,ANS,ERREST) | On-line doc: CALL GAMSDOC DQUD (or @PRT PORT+DOC.DQUD) | Access: LIB NBS+PORT

**DRANDOM** Generates K pseudo-random numbers from a user-specified discrete distribution. | Command in MINITAB Proprietary interactive system. | Class(es): L0a1 | Usage: DRANDOM K,observations,values in C probabilities in C put into O | On-line doc: HELP DRANDOM (in MINITAB) | Tests: MINITAB+TEST-SOURCE. | Access: @QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)


**DLR** An auxiliary routine for use together with FFT to transform 2N real data points. Uses less storage than FFTR. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is RLTR. | Class(es): H1a1 | Usage: CALL DLR (A,B,N,ISN) | On-line doc: CALL GAMSDOC DLR (or @PRT PORT+DOC.DLR) | Access: LIB NBS+PORT | See also: DFFFR

**DROT** Apply Givens plane rotation to double precision vector. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SROT. | Class(es): D1A8 | Usage: CALL DROT(N,D,A,INCX,DY,INCY,D,DZ) | On-line doc: CALL GAMSDOC DROT (or @PRT CMLIB+DOC.DROT/BLAS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

**DROTG** Construct Givens plane rotation of double precision matrix. | Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is SROTG. | Class(es): D1b10 | Usage: CALL DROTG(DA,DB,DC,DS) | On-line doc: CALL
GAMSDOC DROTG (or QRT CMLIB+DOC.DROTG/BLAS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB


DRPOLY Finds zeros of a polynomial with real coefficients. Output zeros are in a pair of arrays, for real and imaginary part. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is RPLOY. | Class(es): Flala | Usage: CALL DRPOLY (DEGREE,COEFF,ZERO,ZEROI) | On-line doc: CALL GAMSDOC DRPOLY (or QRT PORT+DOC.DRPOLY) | Access: LIB NBS+PORT

DRQUAD Finds the integral of a general user defined EXTERNAL function by an adaptive technique. Combined absolute and relative error control. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is RQUAD. | Class(es): H2a | Usage: CALL DRQUAD (F,A,B,EPSCS,EPSFN,ANS,ERREST) | On-line doc: CALL GAMSDOC DRQUAD (or QRT PORT+DOC.DRQUAD) | Access: LIB NBS+PORT


DSPDI Use factorization of double precision symmetric indefinite matrix stored in packed form to compute determinant and/or inverse. Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSPDI. Class(es): D2b1a D3b1a Usage: CALL DSPDI(AP,N,KPVT,DET,INERT,WORK,JOB) On-line doc: CALL GAMSDOC DSPDI (or @PRT CMLIB+DOC.DSPDI/LINPACKD) Tests: CMLIB+TEST-SOURCE.$F1/LINPACKD, CMLIB+TEST-SOURCE.$F2/LINPACKD Access: LIB NBS+CMLIB See also: DSPCO DSPFA

DSPFNC Double precision Spence Dilogarithm, - the integral from 0 to d of (ln(abs.val. of 1-y) / y)dy. Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Single precision version is SPENC. Class(es): C5 Usage: D1 = DSPFNC(D) On-line doc: CALL GAMSDOC DSPFNC (or @PRT CMLIB+DOC.SUMM/I/FNLIB) Access: LIB NBS+CMLIB


DSPLN1 Evaluates a function and derivatives described previously by an expansion in terms of B-splines. Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLN1. Class(es): E3 K6 Usage: CALL DSPLN1(K,T,N,A,X,ID,NID,FX) On-line doc: CALL GAMSDOC DSPLN1 (or @PRT PORT+DOC.DSPLN1) Access: LIB NBS+PORT


DSPLND Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLN1D. Class(es): E3 K6 Usage: CALL DSPLND(K,T,N,A,X,ID,MD,FX) On-line doc: CALL GAMSDOC DSPLND (or @PRT PORT+DOC.DSPLND) Access: LIB NBS+PORT

DSPLNE Evaluates at a set of points, a function described by a previously determined expansion in terms of B-splines. Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLNE. Class(es): E3 K6 Usage: CALL DSPLNE(K,T,N,A,X,ID,FX) On-line doc: CALL GAMSDOC DSPLNE (or @PRT PORT+DOC.DSPLNE) Access: LIB NBS+PORT

DSPLNI Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Proprietary double precision Fortran subprogram in PORT library. Single precision version is SPLNI. Class(es): H32a2a1 E3 K6 Usage: CALL DSPLNI(K,T,N,A,X,FIX) On-line doc: CALL GAMSDOC DSPLNI (or @PRT PORT+DOC.DSPLNI) Access: LIB NBS+PORT

DSPSL Use factorization of double precision symmetric indefinite matrix stored in packed form to solve systems. Portable double precision Fortran subprogram in LINPACKD sublibrary of CMLIB library. Single precision version is SSPSL. Class(es): D2b1a Usage: CALL DSPSLS(AP,N,KPVT,B) On-line doc: CALL GAMSDOC DSPSLS (or @PRT CMLIB+DOC.DPSLS/LINPACKD) Tests: CMLIB+TEST-SOURCE.$F1/LINPACKD, CMLIB+TEST-SOURCE.$F2/LINPACKD Access: LIB NBS+CMLIB See also: DSPCO DSPFA


DTAN Double precision tangent of d. Portable double precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Class(es): C4c Usage: D1 = DTAN (D) On-line doc: CALL GAMSDOC DTAN (or @PRT CMLIB+DOC.SUMM/I/FNLIB) Access: LIB NBS+CMLIB

DTANH Computes the elementary tangent function. If your Fortran library includes this function, use that instead. Proprietary double precision Fortran subprogram in PORT library. Single precision version is TANH. Class(es): C4c Usage: D = DTANH (X) On-line doc: CALL GAMSDOC DTANH (or @PRT PORT+DOC.DTANH) Access: LIB NBS+PORT

DTCHBP Evaluates a polynomial expressed as a sum of Chebyshev polynomials. Proprietary double precision Fortran subprogram in PORT library. Single precision version is TCHBP. Class(es): C3a2 Usage: D = DTCHBP (N, ALPHA, X, X0, X1) On-line doc: CALL GAMSDOC DTCHBP (or @PRT PORT+DOC.DTCHBP) Access: LIB NBS+PORT

DTPTB Solve a system of ordinary differential equations with boundary conditions at two points, using multiple shooting method. Proprietary single precision Fortran subprogram in IMSL library. Class(es): I1L2 Usage: CALL DTPTB
(N,FCN1,FCN,FCNB,XX,A,XX,NITR,X,XAMY,Y,ITOL,WTOL,WORK,IER) | On-line doc: CALL GAMSDOC DTPTB (or @PORT IMSL+DOC.DTPTB) | Access: LIB NBS*IMSL


DUMB Given interval endpoints, this generates a uniform mesh, with needed multiplications for B-spline use. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is UMB. | Class(es): E8 K8 | Usage: CALL DUMB (A,B,NAB,K,N,XX) | On-line doc: CALL GAMSDOC DUMB (or @PORT PORT+DOC.DUMB) | Access: LIB NBS*PORT

DUMD Given interval endpoints, this generates a uniform mesh of distinct points. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is UMD. | Class(es): E8 K8 | Usage: CALL DUMD (A,B,NAB,K,X) | On-line doc: CALL GAMSDOC DUMD (or @PORT PORT+DOC.DUMD) | Access: LIB NBS*PORT

DUMKFL Decomposes a non-zero floating point number into mantissa and an exponent. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is UMKF. | Class(es): A6c | Usage: CALL DUMKFL (F, E, M) | On-line doc: CALL GAMSDOC DUMKFL (or @PORT PORT+DOC.DUMKFL) | Access: LIB NBS*PORT

DVBTOD Converts a mantissa and exponent into a base 10 floating point number. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is VBTO. | Class(es): A6b | Usage: CALL DVBTOD (E, M, E10, M10) | On-line doc: CALL GAMSDOC DVBTOD (or @PORT PORT+DOC.DVBTOD) | Access: LIB NBS*PORT

DVCRP Solve a system of ordinary differential equations with boundary conditions at two points, using a variable order, variable step size finite difference method with deferred corrections. | Proprietary single precision Fortran subroutine in IMSLIB library. | Class(es): I1b2 | Usage: CALL DVCRP (N,FCN,FCN1,FCN2,XX,A,XX,NGMAX,NGRID,IP,IR,TOL,X,Y,ABT,PAR,WORK,IPWORK,IER) | On-line doc: CALL GAMSDOC DVCRP (or @PORT IMSL+DOC.DVCRP) | Access: LIB NBS*IMSL

DVDTOB Converts a base-10 mantissa and exponent of a floating point number into a machine-base representation. | Proprietary double precision Fortran subroutine in PORT library. | Class(es): A6c | Usage: CALL DVDTOB (E10, M10, E, M) | On-line doc: CALL GAMSDOC DVDTOB (or @PORT PORT+DOC.DVDTOB) | Access: LIB NBS*PORT


DZERO Finds a single real root of a function within an interval specified by the user. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is ZERO. | Class(es): F1b | Usage: D = DZERO (F,A,B,T) | On-line doc: CALL GAMSDOC DZERO (or @PORT PORT+DOC.DZERO) | Access: LIB NBS*PORT


DZONEJ Finds a solution of a system of non-linear equations. User must provide a SUBROUTINE to compute the Jacobian matrix. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is ZONEJ. | Class(es): F2a | Usage: CALL DZONEJ (FUNC, D2JAC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC DZONEJ (or @PORT PORT+DOC.DZONEJ) | Access: LIB NBS*PORT
E01AAE  Interpolated values, one variable, data at unequally spaced points, Aitken's technique. | Proprietary single precision Fortran
   subprogram in NAG library. Double precision version is E01AAF. [Class(es): Elb] Usage: CALL E01AAE (A, B, C, N1, N2, N, X) | On-line doc: CALL GAMSDOC E01AAE (or @PRT NAG*DOC.E01AAE) | Access: LIB NBS*NAG

E01AAF  Interpolated values, one variable, data at unequally spaced points, Aitken's technique. | Proprietary double precision Fortran
   subprogram in NAG library. Single precision version is E01AAE. [Class(es): Elb] Usage: CALL E01AAF (A, B, C, N1, N2, N, X) | On-line doc: CALL GAMSDOC E01AAF (or @PRT NAG*DOC.E01AAF) | Access: LIB NBS*NAG

E01ABE  Interpolated values, one variable, data at unequally spaced points, Everett's formula. | Proprietary single precision Fortran subprogram in
   NAG library. Double precision version is E01ABF. [Class(es): Elb] Usage: CALL E01ABE (N, P, A, G, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC E01ABE (or @PRT NAG*DOC.E01ABE) | Access: LIB NBS*NAG

E01ABF  Interpolated values, one variable, data at unequally spaced points, Everett's formula. | Proprietary double precision Fortran subprogram in
   NAG library. Single precision version is E01ABE. [Class(es): Elb] Usage: CALL E01ABF (N, P, A, G, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC E01ABF (or @PRT NAG*DOC.E01ABF) | Access: LIB NBS*NAG

E01ACE  Interpolated values, two variables, data on rectangular grid, fitting bicubic spline. | Proprietary single precision Fortran subprogram in
   NAG library. Double precision version is E01ACF. [Class(es): E2a] Usage: CALL E01ACE (A, B, X, Y, F, VAL, VAL, IFAIL, XX, WORK, AM, D, IGI, M1, N1) | On-line doc: CALL GAMSDOC E01ACE (or @PRT NAG*DOC.E01ACE) | Access: LIB NBS*NAG

E01ACF  Interpolated values, two variables, data on rectangular grid, fitting bicubic spline. | Proprietary double precision Fortran subprogram in
   NAG library. Single precision version is E01ACE. [Class(es): E2a] Usage: CALL E01ACF (A, B, X, Y, F, VAL, VAL, IFAIL, XX, WORK, AM, D, IGI, M1, N1) | On-line doc: CALL GAMSDOC E01ACF (or @PRT NAG*DOC.E01ACF) | Access: LIB NBS*NAG

E01AEE  Interpolating functions, polynomial interpolant, data may include derivative values. | Proprietary single precision Fortran subprogram in
   NAG library. Double precision version is E01AEE. [Class(es): Elb] Usage: CALL E01AEE (M, XMIN, XMAX, X, Y, IP, N, ITMIN, ITMAX, A, WRK, LWRK, IWRK, LWRK, IFAIL) | On-line doc: CALL GAMSDOC E01AEE (or @PRT NAG*DOC.E01AEE) | Access: LIB NBS*NAG | See also: E02AEE

E01AEP  Interpolating functions, polynomial interpolant, data may include derivative values. | Proprietary double precision Fortran subprogram in
   NAG library. Single precision version is E01AEE. [Class(es): Elb] Usage: CALL E01AEP (M, XMIN, XMAX, X, Y, IP, N, ITMIN, ITMAX, A, WRK, LWRK, IWRK, LWRK, IFAIL) | On-line doc: CALL GAMSDOC E01AEP (or @PRT NAG*DOC.E01AEP) | Access: LIB NBS*NAG | See also: E02AEP

E01BAE  Interpolating functions cubic spline interpolant. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01BAE. [Class(es): E1a] Usage: CALL E01BAE (M, X, Y, K, C, LCK, WRK, LWRK, IFAIL) | On-line doc: CALL GAMSDOC E01BAE (or @PRT NAG*DOC.E01BAE) | Access: LIB NBS*NAG | See also: E02BBA E02BCE E02BDE

E01BAF  Interpolating functions cubic spline interpolant. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01BAE. [Class(es): E1a] Usage: CALL E01BAF (M, X, Y, K, C, LCK, WRK, LWRK, IFAIL) | On-line doc: CALL GAMSDOC E01BAF (or @PRT NAG*DOC.E01BAF) | Access: LIB NBS*NAG | See also: E02BBF E02BCF E02BDF

E01RAE  Produces, from a set of functions and corresponding abscissae, the coefficients of an interpolating rational function expressed in continued fraction form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E01RAF. [Class(es): Etc] Usage: CALL E01RAE (N,X,F,M,A,U,IW,IFAIL) | On-line doc: CALL GAMSDOC E01RAE (or @PRT NAG*DOC.E01RAE) | Access: LIB NBS*NAG | See also: E01RAE

E01RAF  Produces, from a set of functions and corresponding abscissae, the coefficients of an interpolating rational function expressed in continued fraction form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E01RAE. [Class(es): Etc] Usage: CALL E01RAF (N,X,F,M,A,U,IW,IFAIL) | On-line doc: CALL GAMSDOC E01RAF (or @PRT NAG*DOC.E01RAF) | Access: LIB NBS*NAG | See also: E01RAF

E01RBF  Evaluates continued fractions of the form produced by NAG library routine E01RAE. | Proprietary single precision Fortran subprogram in
   NAG library. Double precision version is E01RBF. [Class(es): E3] Usage: CALL E01RBE(M,A,U,X,F,IFAIL) | On-line doc: CALL GAMSDOC E01RBE (or @PRT NAG*DOC.E01RBE) | Access: LIB NBS*NAG

E01RBE  Evaluates continued fractions of the form produced by NAG library routine E01RAE. | Proprietary single precision Fortran

E01RBF  Evaluates continued fractions of the form produced by NAG library routine E01RAF. | Proprietary double precision Fortran

E01RBD  Minimax curve fit by polynomials. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02ACF. [Class(es): K2] Usage: CALL E02ACF (X, Y, N, A, M1, REF) | On-line doc: CALL GAMSDOC E02ACF (or @PRT NAG*DOC.E02ACF) | Access: LIB NBS*NAG

E01RBF  Minimax curve fit by polynomials. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02ACF. [Class(es): K2] Usage: CALL E02ACF (X, Y, N, A, M1, REF) | On-line doc: CALL GAMSDOC E02ACF (or @PRT NAG*DOC.E02ACF) | Access: LIB NBS*NAG

E02ADE  Least-squares curve fit by polynomials, arbitrary data points. | Proprietary single precision Fortran subprogram in NAG library.
   Double precision version is E02ADF. [Class(es): K1a1a2] Usage: CALL E02ADE (M, KPLUS1, NROWS, X, Y, W, WORK1, WORK2, A, S, IFAIL) | On-line doc: CALL GAMSDOC E02ADE (or @PRT NAG*DOC.E02ADE) | Access: LIB NBS*NAG | See also: E02AEE
E02ADF  Least-squares curve fit by polynomials, arbitrary data points. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02ADE. | Class(es): K1a1a2 | Usage: CALL E02ADF (M, KPLUS1, NROWS, X, Y, W, WORK1, WORK2, A, S, IFAIL) | On-line doc: CALL GAMSDOC E02ADF (or PRT NAG*DOC.E02ADF) | Access: LIB NBS*NAG | See also: E02AEF

E02AEE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AEF. | Class(es): C3a2 E8 K8 | Usage: CALL E02AEE (NPLUS1, A, XCAP, P, IFAIL) | On-line doc: CALL GAMSDOC E02AEE (or PRT NAG*DOC.E02AEE) | Access: LIB NBS*NAG

E02AEF Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form (simplified parameter list). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AEE. | Class(es): C3a2 E8 K8 | Usage: CALL E02AEF (NPLUS1, A, XCAP, P, IFAIL) | On-line doc: CALL GAMSDOC E02AEF (or PRT NAG*DOC.E02AEF) | Access: LIB NBS*NAG

E02AFE Least-squares curve fit by polynomials, special data points (including interpolation). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AFF. | Class(es): K1a1a2 E1b | Usage: CALL E02AFF (NPULS, F, A, IFAIL) | On-line doc: CALL GAMSDOC E02AFF (or PRT NAG*DOC.E02AFF) | Access: LIB NBS*NAG | See also: E02AEF

E02AFF Least-squares curve fit by polynomials, special data points (including interpolation). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AGF. | Class(es): K1a2a | Usage: CALL E02AGF (M, KPLUS1, NROWS, XMIN, XMAX, X, Y, W, MF, XF, YF, LYF, IP, A, S, NP1, WRK, LWKR, ILWKR, IFAIL) | On-line doc: CALL GAMSDOC E02AGF (or PRT NAG*DOC.E02AGF) | Access: LIB NBS*NAG | See also: E02AEF

E02AGF Least-squares curve fit by polynomials, arbitrary data points, values and derivatives may be constrained. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is E02AGE. | Class(es): K1a2a | Usage: CALL E02AGE (M, KPLUS1, NROWS, XMIN, XMAX, X, Y, W, MF, XF, YF, LYF, IP, A, S, NP1, WRK, LWKR, ILWKR, IFAIL) | On-line doc: CALL GAMSDOC E02AGE (or PRT NAG*DOC.E02AGE) | Access: LIB NBS*NAG | See also: E02AEF

E02AHF Derivative of fitted polynomial in Chebyshev series form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AHE. | Class(es): C3a E8 K6 | Usage: CALL E02AHE (NP1, XMIN, XMAX, A, IA1, LA, PATM1, ADIF, IADIF1, IADIF, IFAIL) | On-line doc: CALL GAMSDOC E02AHE (or PRT NAG*DOC.E02AHE) | Access: LIB NBS*NAG

E02AHE Derivative of fitted polynomial in Chebyshev series form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AHF. | Class(es): C3a E8 K6 | Usage: CALL E02AHF (NP1, XMIN, XMAX, A, IA1, LA, PATM1, ADIF, IADIF1, IADIF, IFAIL) | On-line doc: CALL GAMSDOC E02AHF (or PRT NAG*DOC.E02AHF) | Access: LIB NBS*NAG

E02AJE Integral of fitted polynomial in Chebyshev series form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AJF. | Class(es): H2a2a1 C3a2 K8 | Usage: CALL E02AJE (NP1, XMIN, XMAX, A, IA1, LA, QATM1, AINT, IAIN1, IAIN, IFAIL) | On-line doc: CALL GAMSDOC E02AJE (or PRT NAG*DOC.E02AJE) | Access: LIB NBS*NAG

E02AJF Integral of fitted polynomial in Chebyshev series form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AJE. | Class(es): H2a2a1 C3a2 K8 | Usage: CALL E02AJF (NP1, XMIN, XMAX, A, IA1, LA, QATM1, AINT, IAIN1, IAIN, IFAIL) | On-line doc: CALL GAMSDOC E02AJF (or PRT NAG*DOC.E02AJF) | Access: LIB NBS*NAG

E02AKE Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02AKF. | Class(es): C3a2 E3 K6 | Usage: CALL E02AKE (NP1, XMIN, XMAX, A, IA1, LA, X, RESULT, IFAIL) | On-line doc: CALL GAMSDOC E02AKE (or PRT NAG*DOC.E02AKE) | Access: LIB NBS*NAG

E02AKF Evaluation of fitted functions, polynomial in one variable, from Chebyshev series form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02AKE. | Class(es): C3a2 E3 K6 | Usage: CALL E02AKF (NP1, XMIN, XMAX, A, IA1, LA, X, RESULT, IFAIL) | On-line doc: CALL GAMSDOC E02AKF (or PRT NAG*DOC.E02AKF) | Access: LIB NBS*NAG

E02BAE Least-squares curve fit by cubic splines (including interpolation). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BAF. | Class(es): K1a1a E1a | Usage: CALL E02BAE (M, NCPAT, X, Y, W, K, WORK1, WORK2, C, SS, IFAIL) | On-line doc: CALL GAMSDOC E02BAE (or PRT NAG*DOC.E02BAE) | Access: LIB NBS*NAG | See also: E02BBD E02BCE E02BDE

E02BAF Least-squares curve fit by cubic splines (including interpolation). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BAE. | Class(es): K1a1a E1a | Usage: CALL E02BAF (M, NCPAT, X, Y, W, K, WORK1, WORK2, C, SS, IFAIL) | On-line doc: CALL GAMSDOC E02BAF (or PRT NAG*DOC.E02BAF) | Access: LIB NBS*NAG | See also: E02BBD E02BCE E02BDE

E02BBE Evaluation of fitted functions, cubic spline as E02BAE, function only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BBF. | Class(es): E3 K6 | Usage: CALL E02BBE (NCPAT, K, X, S, IFAIL) | On-line doc: CALL GAMSDOC E02BBE (or PRT NAG*DOC.E02BBE) | Access: LIB NBS*NAG

E02BBF Evaluation of fitted functions, cubic spline as E02BAF, function only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BBD. | Class(es): E3 K6 | Usage: CALL E02BBF (NCAPT, K, X, S, IFAIL) | On-line doc: CALL GAMSDOC E02BBF (or PRT NAG*DOC.E02BBF) | Access: LIB NBS*NAG

E02BCE Evaluation of fitted functions, cubic spline as E02BAE, function and derivatives. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BDE. | Class(es): E3 K6 | Usage: CALL E02BCE (NCAPT, K, X, S, IFAIL) | On-line doc: CALL GAMSDOC E02BCE (or PRT NAG*DOC.E02BCE) | Access: LIB NBS*NAG
in NAG library. Double precision version is E02BCF.  |  Class(es): E3 K6  | Usage: CALL E02BCE (NCPAT, K, C, X, LEFT, S, IFAIL)  | On-line doc: CALL GAMSDOC E02BCE (or @PRT NAG*DOC.E02BCE)  | Access: LIB NBS*NAG

E02BCF Evaluation of fitted functions, cubic spline as E02BAF, function and derivatives.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BCE.  |  Class(es): E3 K6  | Usage: CALL E02BCE (NCPAT, K, C, X, LEFT, S, IFAIL)  | On-line doc: CALL GAMSDOC E02BCE (or @PRT NAG*DOC.E02BCE)  | Access: LIB NBS*NAG

E02BDE Evaluation of fitted functions, cubic spline as E02BAE, definite integral.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02BDF.  |  Class(es): H2a2a1 E3 K6  | Usage: CALL E02BDE (NCPAT, K, C, DEINT, IFAIL)  | On-line doc: CALL GAMSDOC E02BDE (or @PRT NAG*DOC.E02BDE)  | Access: LIB NBS*NAG

E02BDF Evaluation of fitted functions, cubic spline as E02BAF, definite integral.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02BDE.  |  Class(es): H2a2a1 E3 K6  | Usage: CALL E02BDF (NCPAT, K, C, DEINT, IFAIL)  | On-line doc: CALL GAMSDOC E02BDF (or @PRT NAG*DOC.E02BDF)  | Access: LIB NBS*NAG

E02CAE Least-squares surface fit by polynomials, for data on lines.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02CAF.  |  Class(es): K1a1b  | Usage: CALL E02CAE (M, N, K, L, X, Y, F, W, NX, A, NA, XMIN, XMAX, NUX, INUXP1, NUY, INUYP1, WORK, NWORK, IFAIL)  | On-line doc: CALL GAMSDOC E02CAE (or @PRT NAG*DOC.E02CAE)  | Access: LIB NBS*NAG

E02CAF Least-squares surface fit by polynomials, for data on lines.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02CAE.  |  Class(es): K1a1b  | Usage: CALL E02CAF (M, N, K, L, X, Y, F, W, NX, A, NA, XMIN, XMAX, NUX, INUXP1, NUY, INUYP1, WORK, NWORK, IFAIL)  | On-line doc: CALL GAMSDOC E02CAF (or @PRT NAG*DOC.E02CAF)  | Access: LIB NBS*NAG

E02CBE Evaluation of fitted functions, polynomial in two variables as E02CAE.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02CBF.  |  Class(es): E3 K6  | Usage: CALL E02CBF (MFIRST, MLAST, K, L, X, YMIN, YMAG, NNAG, IFAIL)  | On-line doc: CALL GAMSDOC E02CBF (or @PRT NAG*DOC.E02CBF)  | Access: LIB NBS*NAG

E02CBF Evaluation of fitted functions, polynomial in two variables as E02CAF.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02CBE.  |  Class(es): E3 K6  | Usage: CALL E02CBF (MFIRST, MLAST, K, L, X, YMIN, YMAG, NNAG, IFAIL)  | On-line doc: CALL GAMSDOC E02CBF (or @PRT NAG*DOC.E02CBF)  | Access: LIB NBS*NAG

E02DAE Least-squares surface fit by bicubic splines.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02DAF.  |  Class(es): K1a1b  | Usage: CALL E02DAE (M, PX, PY, X, Y, F, W, LAMDA, MU, POINT, NPOINT, DL, C, NC, WS, NWS EPS, SIGMA, RANK, IFAIL)  | On-line doc: CALL GAMSDOC E02DAE (or @PRT NAG*DOC.E02DAE)  | Access: LIB NBS*NAG | See also: E02DDE

E02DAF Least-squares surface fit by bicubic splines.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02DAE.  |  Class(es): K1a1b  | Usage: CALL E02DAF (M, PX, PY, X, Y, F, W, LAMDA, MU, POINT, NPOINT, DL, C, NC, WS, NWS EPS, SIGMA, RANK, IFAIL)  | On-line doc: CALL GAMSDOC E02DAF (or @PRT NAG*DOC.E02DAF)  | Access: LIB NBS*NAG | See also: E02DBF

E02DBE Evaluation of fitted functions, bicubic spline as E02DAE.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02DBF.  |  Class(es): E3 K6  | Usage: CALL E02DBE (M, PX, PY, X, Y, FF, W, LAMDA, MU, POINT, NPOINT, C, NC, IFAIL)  | On-line doc: CALL GAMSDOC E02DBE (or @PRT NAG*DOC.E02DBE)  | Access: LIB NBS*NAG

E02DBF Evaluation of fitted functions, bicubic spline as E02DAF.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02DBE.  |  Class(es): E3 K6  | Usage: CALL E02DBF (M, PX, PY, X, Y, FF, W, LAMDA, MU, POINT, NPOINT, C, NC, IFAIL)  | On-line doc: CALL GAMSDOC E02DBF (or @PRT NAG*DOC.E02DBF)  | Access: LIB NBS*NAG

E02GAF L1-approximation by general linear function.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02GAF.  |  Class(es): D9 K3  | Usage: CALL E02GAE (M, A, LA, B, NPLUS2, TOLER, X, RESID, IRANK, ITER, IWORK, IFAIL)  | On-line doc: CALL GAMSDOC E02GAE (or @PRT NAG*DOC.E02GAE)  | Access: LIB NBS*NAG

E02GAE L1-approximation by general linear function.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02GAF.  |  Class(es): D9 K3  | Usage: CALL E02GAF (M, A, LA, B, NPLUS2, TOLER, X, RESID, IRANK, ITER, IWORK, IFAIL)  | On-line doc: CALL GAMSDOC E02GAF (or @PRT NAG*DOC.E02GAF)  | Access: LIB NBS*NAG

E02GBE L1-approximation by general linear function subject to linear inequality constraints.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02GBF.  |  Class(es): D9 K3  | Usage: CALL E02GBE (M, N, MPL, E, IE, F, X, MXS, MONIT, IPRINT, K, ELIN, INDX, W, W1W, IFAIL)  | On-line doc: CALL GAMSDOC E02GBE (or @PRT NAG*DOC.E02GBE)  | Access: LIB NBS*NAG

E02GBF L1-approximation by general linear function subject to linear inequality constraints.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02GBE.  |  Class(es): D9 K3  | Usage: CALL E02GBF (M, N, MPL, E, IE, F, X, MXS, MONIT, IPRINT, K, ELIN, INDX, W, W1W, IFAIL)  | On-line doc: CALL GAMSDOC E02GBF (or @PRT NAG*DOC.E02GBF)  | Access: LIB NBS*NAG

E02GCE Calculates an L-infinity solution to an over-determined system of linear equations.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02GCF.  |  Class(es): D9 K2  | Usage: CALL E02GCE (M, N, MDIM, NDIM, A, B, TOL, RELERR, X, RESMAX, IRANK, ITER, IFAIL)  | On-line doc: CALL GAMSDOC E02GCE (or @PRT NAG*DOC.E02GCE)  | Access: LIB NBS*NAG
E02GCF  Calculates an L-infinity solution to an over-determined system of linear equations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02GCE. | Class(es): D9 K2 | Usage: CALL E02GCF (M, N, MDIM, NDIM, A, B, TOL, RELERR, X, RESMAX, IRANK, ITER, IFAIL) | On-line doc: CALL GAMSDOC E02GCF (or @PRT NAG+DOC.E02GCF) | Access: LIB NBS+NAG

E02RAE  Pade-approximants. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02RAF. | Class(es): K4 | Usage: CALL E02RAE (IA, IB, C, IC, A, B, W, JW, IFAIL) | On-line doc: CALL GAMSDOC E02RAE (or @PRT NAG+DOC.E02RAE) | Access: LIB NBS+NAG | See also: E02RBE

E02RAF  Pade-approximants. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02RAE. | Class(es): K4 | Usage: CALL E02RAF (IA, IB, C, IC, A, B, W, JW, IFAIL) | On-line doc: CALL GAMSDOC E02RAF (or @PRT NAG+DOC.E02RAF) | Access: LIB NBS+NAG | See also: E02RBE

E02RBE  Evaluation of fitted functions, rational function as E02RAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02RFC. | Class(es): K8 | Usage: CALL E02RBE (A, IA, IB, IB, X, X, ANS, IFAIL) | On-line doc: CALL GAMSDOC E02RBE (or @PRT NAG+DOC.E02RBE) | Access: LIB NBS+NAG

E02RBF  Evaluation of fitted functions, rational function as E02RAF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02RBE. | Class(es): K6 | Usage: CALL E02RBF (A, IA, IB, IB, X, X, ANS, IFAIL) | On-line doc: CALL GAMSDOC E02RBF (or @PRT NAG+DOC.E02RBF) | Access: LIB NBS+NAG

E02ZAE  Sort 2-d data into panels for fitting or evaluating bicubic splines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E02ZAF. | Class(es): E3 K6 | Usage: CALL E02ZAE (FG, FG, LAMDA, MU, M, X, X, Y, POINT, NPINT, ADDRESS, NADRES, IFAIL) | On-line doc: CALL GAMSDOC E02ZAE (or @PRT NAG+DOC.E02ZAE) | Access: LIB NBS+NAG

E02ZAP  Sort 2-d data into panels for fitting or evaluating bicubic splines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E02ZAF. | Class(es): E3 K6 | Usage: CALL E02ZAF (FG, FG, LAMDA, MU, M, X, X, Y, POINT, NPINT, ADDRESS, IFAIL) | On-line doc: CALL GAMSDOC E02ZAF (or @PRT NAG+DOC.E02ZAF) | Access: LIB NBS+NAG

E04ABE  Minimum, function of one variable using function values only. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04ABF. | Class(es): G1a1a | Usage: CALL E04ABE (FUNCT, E1, E2, A, B, MAXCAL, X, F, IFAIL) | On-line doc: CALL GAMSDOC E04ABE (or @PRT NAG+DOC.E04ABE) | Access: LIB NBS+NAG

E04ABF  Minimum, function of one variable using function values only. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04ABE. | Class(es): G1a1a | Usage: CALL E04ABF (FUNCT, E1, E2, A, B, MAXCAL, X, F, IFAIL) | On-line doc: CALL GAMSDOC E04ABF (or @PRT NAG+DOC.E04ABF) | Access: LIB NBS+NAG

E04BBE  Minimum, function of one variable, using first derivative. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04BBF. | Class(es): G1a1b | Usage: CALL E04BBE (FUNCT, E1, E2, A, B, MAXCAL, X, F, G, IFAIL) | On-line doc: CALL GAMSDOC E04BBE (or @PRT NAG+DOC.E04BBE) | Access: LIB NBS+NAG

E04BBF  Minimum, function of one variable, using first derivative. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04BBE. | Class(es): G1a1b | Usage: CALL E04BBF (FUNCT, E1, E2, A, B, MAXCAL, X, F, G, IFAIL) | On-line doc: CALL GAMSDOC E04BBF (or @PRT NAG+DOC.E04BBF) | Access: LIB NBS+NAG

E04CCF  Unconstrained minimum, function of several variables (comprehensive), using function values only, simplex algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04CCF. | Class(es): G1b2 | Usage: CALL E04CCF(N, X, F, TOL, LW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04CCF (or @PRT NAG+DOC.E04CCF) | Access: LIB NBS+NAG

E04DCE  Unconstrained minimum, function of several variables (comprehensive), using function values only, simplex algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04DCE. | Class(es): G1b2 | Usage: CALL E04DCE(N, X, F, LW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04DCE (or @PRT NAG+DOC.E04DCE) | Access: LIB NBS+NAG

E04CGF  Unconstrained minimum, function of several variables (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04CGF. | Class(es): G1b1a | Usage: CALL E04CGF(N, X, F, LW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04CGF (or @PRT NAG+DOC.E04CGF) | Access: LIB NBS+NAG

E04CFF  Unconstrained minimum, function of several variables (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04CFF. | Class(es): G1b1a | Usage: CALL E04CFF(N, X, F, LW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04CFF (or @PRT NAG+DOC.E04CFF) | Access: LIB NBS+NAG

E04DBE  Unconstrained minimum, function of several variables (comprehensive), using first derivatives, conjugate direction algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04DBF. | Class(es): G1b1b | Usage: CALL E04DBE(N, X, F, G, XTOL, FEST, DUM, W, FUNCTION, MONIT, MAXCAL, IFAIL) | On-line doc: CALL GAMSDOC E04DBE (or @PRT NAG+DOC.E04DBE) | Access: LIB NBS+NAG

E04DBF  Unconstrained minimum, function of several variables (comprehensive), using first derivatives, conjugate direction algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04DBE. | Class(es): G1b1b | Usage: CALL E04DBF(N, X, F, G, XTOL, FEST, DUM, W, FUNCTION, MONIT, MAXCAL, IFAIL) | On-line doc: CALL GAMSDOC E04DBF (or @PRT NAG+DOC.E04DBF) | Access: LIB NBS+NAG


E04ECE Unconstrained minimum, sum of squares, n variables (comprehensive), using function values only, combined Gauss-Newton and modified Newton algorithm. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04ECF. Class(es): K1b1a1 | Usage: CALL E04ECE (M, N, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOLE, STEPMAX, X, FSUMSQ, FVEC, FJAC, LI, S, V, IW, NITER, NF, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04ECE (or @PRN NAG*DOC.E04ECE) | Access: LIB NBS*NAG

E04ECC Unconstrained minimum, sum of squares, n variables (comprehensive), using function values only, combined Gauss-Newton and modified Newton algorithm. Proprietary double precision Fortran subroutine in NAG library. Single precision version is E04ECC. Class(es): K1b1a1 | Usage: CALL E04ECC (M, N, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOLE, STEPMAX, X, FSUMSQ, FVEC, FJAC, LI, S, V, IW, NITER, NF, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04ECC (or @PRN NAG*DOC.E04ECC) | Access: LIB NBS*NAG

E04EDE Unconstrained minimum, sum of squares, n variables (easy-to-use), function values only, Gauss-Newton and modified Newton algorithm. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04EDF. Class(es): K1b1 | Usage: CALL E04EDE (M, N, X, FSUMSQ, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04EDE (or @PRN NAG*DOC.E04EDF) | Access: LIB NBS*NAG

E04EDF Unconstrained minimum, sum of squares, n variables (easy-to-use), function values only, Gauss-Newton and modified Newton algorithm. Proprietary double precision Fortran subroutine in NAG library. Single precision version is E04EDF. Class(es): K1b1 | Usage: CALL E04EDF (M, N, X, FSUMSQ, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04EDF (or @PRN NAG*DOC.E04EDF) | Access: LIB NBS*NAG


E04EFC Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Proprietary double precision Fortran subroutine in NAG library. Single precision version is E04EFC. Class(es): K1b1a2 | Usage: CALL E04EFC (M, N, LSQFUN, LSQMON, IPRINT, MAXCAL, ETA, XTOLE, STEPMAX, X, FSUMSQ, FVEC, FJAC, LI, S, V, IW, NITER, NF, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04EFC (or @PRN NAG*DOC.E04EFC) | Access: LIB NBS*NAG

E04ECE Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and modified Newton algorithm. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04EDF. Class(es): K1b1 | Usage: CALL E04ECE (M, N, X, FSUMSQ, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04ECE (or @PRN NAG*DOC.E04ECE) | Access: LIB NBS*NAG

E04ECC Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and modified Newton algorithm. Proprietary double precision Fortran subroutine in NAG library. Single precision version is E04ECC. Class(es): K1b1 | Usage: CALL E04ECC (M, N, X, FSUMSQ, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04ECC (or @PRN NAG*DOC.E04ECC) | Access: LIB NBS*NAG

E04ECE Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04ECF. Class(es): K1b1 | Usage: CALL E04ECE (M, N, X, FSUMSQ, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04ECE (or @PRN NAG*DOC.E04ECF) | Access: LIB NBS*NAG

E04ECC Unconstrained minimum, sum of squares, n variables (easy-to-use), using first derivatives, Gauss-Newton and quasi-Newton algorithm. Proprietary double precision Fortran subroutine in NAG library. Single precision version is E04ECC. Class(es): K1b1 | Usage: CALL E04ECC (M, N, X, FSUMSQ, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04ECC (or @PRN NAG*DOC.E04ECC) | Access: LIB NBS*NAG

E04EDE Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and modified Newton algorithm. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04EDF. Class(es): K1b1a2 | Usage: CALL E04EDE (M, N, LSQMON, IPRINT, MAXCAL, ETA, XTOLE, STEPMAX, X, FSUMSQ, FVEC, FJAC, LI, S, V, IW, NITER, NF, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04EDF (or @PRN NAG*DOC.E04EDF) | Access: LIB NBS*NAG

E04EDF Unconstrained minimum, sum of squares, n variables (comprehensive), using first derivatives, Gauss-Newton and modified Newton algorithm. Proprietary double precision Fortran subroutine in NAG library. Single precision version is E04EDF. Class(es): K1b1a2 | Usage: CALL E04EDF (M, N, LSQMON, IPRINT, MAXCAL, ETA, XTOLE, STEPMAX, X, FSUMSQ, FVEC, FJAC, LI, S, V, IW, NITER, NF, IW, LI, W, L, IFAIL) | On-line doc: CALL GAMSDOC E04EDF (or @PRN NAG*DOC.E04EDF) | Access: LIB NBS*NAG


E04HFE Unconstrained minimum, sum of squares, n variables (comprehensive), using second derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HFF. | Class(es): | K1b1a3 | Usage: CALL E04HFE (M, N, LSQFUN, LSQHES, LSQMON, IPRINT, MAXCAL, ETA, XTOL, STEPMS, X, FSUMSQ, FVEC, FJAC, LI, SJ, V, LV, NITER, NF, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HFE (or OPTION NAG+DOC.E04HFE) | Access: LIB NBS+NAG


E04HFE Unconstrained minimum, sum of squares, n variables (easy-to-use), using second derivatives, Gauss-Newton and modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04HFF. | Class(es): | K1b1a3 | Usage: CALL E04HFE (M, N, X, FSUMSQ, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04HFE (or OPTION NAG+DOC.E04HFE) | Access: LIB NBS+NAG


E04JAE Minimum, function of several variables, simple bounds (easy-to-use), using function values only, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04JAF. | Class(es): | G2h1a1 | Usage: CALL E04JAE (N, IBOUND, BL, BU, X, F, IW, LIW, W, LW, IFAIL) | On-line doc: CALL GAMSDOC E04JAE (or OPTION NAG+DOC.E04JAE) | Access: LIB NBS+NAG


E04JBE Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04JBF. | Class(es): | G2h1a1 | Usage: CALL E04JBE (N, FUNCT, MONIT, IPRINT, LOCPL, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPFX, FEST, DELTA, IBOUND, BU, X, HESL, HI, HESD, ISTATE, F, G, IW, LIW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04JBE (or OPTION NAG+DOC.E04JBE) | Access: LIB NBS+NAG

E04JBF Minimum, function of several variables, simple bounds (comprehensive), using function values only, quasi-Newton algorithm. |
Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04JBF. | Class(es): G2h1a1 G1b1a |
Usage: CALL E04JBF (N, FUNCT, MONIT, IPRINT, LOCSCH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, DELTA, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, XF, IFAIL) | On-line doc: CALL GAMSDOC E04JBF (or @PRT NAG+DOC.E04JBF) | Access: LIB NBS*NAG

E04KAE Minimum function of several variables, simple bounds (easy-to-use), using first derivatives, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KAF. | Class(es): G2h1a2 |
Usage: CALL E04KAE (N, IBOUND, BL, BU, X, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KAE (or @PRT NAG+DOC.E04KAE) | Access: LIB NBS*NAG

E04KAF Minimum function of several variables, simple bounds (easy-to-use), using first derivatives, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KAE. | Class(es): G2h1a2 |
Usage: CALL E04KAF (N, IBOUND, BL, BU, X, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KAF (or @PRT NAG+DOC.E04KAF) | Access: LIB NBS*NAG

E04KBE Minimum function of several variables, simple bounds (comprehensive), using first derivatives, quasi-Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KBE. | Class(es): G2h1a2 |
Usage: CALL E04KBE (N, FUNCT, MONIT, IPRINT, LOCSCH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KBE (or @PRT NAG+DOC.E04KBE) | Access: LIB NBS*NAG

E04KBF Minimum function of several variables, simple bounds (comprehensive), using first derivatives, quasi-Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KBF. | Class(es): G2h1a2 |
Usage: CALL E04KBF (N, FUNCT, MONIT, IPRINT, LOCSCH, INTYPE, MINLIN, MAXCAL, ETA, XTOL, STEPMX, FEST, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KBF (or @PRT NAG+DOC.E04KBF) | Access: LIB NBS*NAG

E04KCE Minimum function of several variables, simple bounds (easy-to-use), using first derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KCF. | Class(es): G2h1a2 |
Usage: CALL E04KCE (N, IBOUND, BL, BU, X, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KCE (or @PRT NAG+DOC.E04KCE) | Access: LIB NBS*NAG

E04KCF Minimum function of several variables, simple bounds (easy-to-use), using first derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KCF. | Class(es): G2h1a2 |
Usage: CALL E04KCF (N, IBOUND, BL, BU, X, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KCF (or @PRT NAG+DOC.E04KCF) | Access: LIB NBS*NAG

E04KDE Minimum function of several variables, simple bounds (comprehensive), using first derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04KDF. | Class(es): G2h1a2 |
Usage: CALL E04KDE (N, FUNCT, MONIT, IPRINT, MAXCAL, ETA, XTOL, DELTA, STEP MX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KDE (or @PRT NAG+DOC.E04KDE) | Access: LIB NBS*NAG

E04KDF Minimum function of several variables, simple bounds (comprehensive), using first derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04KDF. | Class(es): G2h1a2 |
Usage: CALL E04KDF (N, FUNCT, MONIT, IPRINT, MAXCAL, ETA, XTOL, DELTA, STEP MX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04KDF (or @PRT NAG+DOC.E04KDF) | Access: LIB NBS*NAG

E04LAE Minimum function of several variables, simple bounds (easy-to-use), using first and second derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04LAF. | Class(es): G2h1a3 |
Usage: CALL E04LAE (N, IBOUND, BL, BU, X, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LAE (or @PRT NAG+DOC.E04LAE) | Access: LIB NBS*NAG

E04LAF Minimum function of several variables, simple bounds (easy-to-use), using first and second derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04LAE. | Class(es): G2h1a3 |
Usage: CALL E04LAF (N, IBOUND, BL, BU, X, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LAF (or @PRT NAG+DOC.E04LAF) | Access: LIB NBS*NAG

E04LBE Minimum function of several variables, simple bounds (comprehensive), using first and second derivatives, modified Newton algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04LBF. | Class(es): G2h1a3 |
Usage: CALL E04LBE (N, FUNCT, HESS, MONIT, IPRINT, MAXCAL, ETA, XTOL, STEP MX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LBE (or @PRT NAG+DOC.E04LBE) | Access: LIB NBS*NAG

E04LBF Minimum function of several variables, simple bounds (comprehensive), using first and second derivatives, modified Newton algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is E04LBE. | Class(es): G2h1a3 |
Usage: CALL E04LBF (N, FUNCT, HESS, MONIT, IPRINT, MAXCAL, ETA, XTOL, STEP MX, IBOUND, BL, BU, X, HESL, LH, HESD, ISTATE, F, G, IW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04LBF (or @PRT NAG+DOC.E04LBF) | Access: LIB NBS*NAG

E04UAE Minimum function of n variables, non-linear constraints, function and constraint values only, sequential augmented Lagrangian quasi-Newton method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04UAF. | Class(es): G2h3b1a |
Usage: CALL E04UAE (N, MEQ, MINEQ, MRNGE, M, MONAUX, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, GIW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04UAE (or @PRT NAG+DOC.E04UAE) | Access: LIB NBS*NAG

E04UAF Minimum function of n variables, non-linear constraints, function and constraint values only, sequential augmented Lagrangian quasi-Newton method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is E04UAF. | Class(es): G2h3b1a |
Usage: CALL E04UAF (N, MEQ, MINEQ, MRNGE, M, MONAUX, IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND, XL, XU, LAMSET, X, RHO, RLAM, F, GIW, LIW, LW, LW, IFAIL) | On-line doc: CALL GAMSDOC E04UAF (or @PRT NAG+DOC.E04UAF) | Access: LIB NBS*NAG


E04YBE Check user's routine calculating second derivative term in Hessian matrix of sum of squares. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04YBF. Class(es): G4c Usage: CALL E04YBE (M, N, LSQFUN, LSQHES, X, FVEC, FJAC, LI, LB, LIW, LW, IFAIL) On-line doc: CALL GAMSDOC E04YBE (or @PRT NAG*DOC.E04YBE) Access: LIB NBS*NAG


E04ZBE Check user's routines calculating second derivatives of function and constraints. Proprietary single precision Fortran subroutine in NAG library. Double precision version is E04ZBF. Class(es): G4c Usage: CALL E04ZBE (N, M, FUNCT, HESS, CON, CHESS, X, W, LW, IFAIL) On-line doc: CALL GAMSDOC E04ZBE (or @PRT NAG*DOC.E04ZBE) Access: LIB NBS*NAG

January 1984

GAMS: Module Dictionary

C 83

E1 Exponential integral, the integral from x to infinity of (e^{x-t}/t) dt. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DE1. | Class(es): C8 | Usage: R = E1(X) | On-line doc: CALL GAMSDOC E1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

EBALAC Balance a complex general matrix and isolate eigenvalues whenever possible. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4cl1a | Usage: CALL EBALAC (AR,A1,N1,A,K,L,D) | On-line doc: CALL GAMSDOC EBALAC (or @PRT IMSL+DOC.EBALAC) | Access: LIB NBS+IMSL


EESBF Estimates the error in a given B-spline fit to a function, f, by refining the mesh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEESBF. | Class(es): E3 K6 | Usage: X = EESBF (K,T,N1,A1,T2,N2,A2) | On-line doc: CALL GAMSDOC EESBF (or @PRT PORT+DOC.EESBF) | Access: LIB NBS+PORT

EEEBSI Estimates the error in a given B-spline fit to a function f by refining the mesh intervals selected by user. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEEBSI. | Class(es): E3 K6 | Usage: X = EEEBSI (K,T,N,A,F) | On-line doc: CALL GAMSDOC EEEBSI (or @PRT PORT+DOC.EEEBSI) | Access: LIB NBS+PORT

EESFF Finds the maximum absolute error in a given B-spline fit to a function, f. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEESFF. | Class(es): E3 K6 | Usage: X = EESFF (K,T,N,A,F) | On-line doc: CALL GAMSDOC EESFF (or @PRT PORT+DOC.EESFF) | Access: LIB NBS+PORT

EESF1 Finds the maximum absolute error in a given B-spline fit to a function, f, on a set of user selected intervals. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEESF1. | Class(es): E3 K6 | Usage: X = EESF1 (K,T,N,A,F,NX,NEST) | On-line doc: CALL GAMSDOC EESF1 (or @PRT PORT+DOC.EESF1) | Access: LIB NBS+PORT


EHOUSS Reduction of a symmetric matrix to symmetric tridiagonal form. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4clb1 | Usage: CALL EHOUSS (AR,A1,N,D,E2) | On-line doc: CALL GAMSDOC EHOUSS (or @PRT IMSL+DOC.EHOUSS) | Access: LIB NBS+IMSL

EIGBS Find some eigenvalues and (optionally) eigenvectors of a real symmetric band matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a6 | Usage: CALL EIGBS (A,N1A,JOB,NC,M,D,Z1,Z2,WORK,IER) | On-line doc: CALL GAMSDOC EIGBS (or @PRT IMSL+DOC.EIGBS) | Access: LIB NBS+CMLIB


EIGCH Eigenvalues and (optionally) eigenvectors of a complex Hermitian matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a3 | Usage: CALL EIGCH (A,N1JOB,N,D,Z1,WK,IER) | On-line doc: CALL GAMSDOC EIGCH (or @PRT IMSL+DOC.EIGCH) | Access: LIB NBS+IMSL

EIGEN Finds all eigenvalues and eigenvectors of a real matrix. Output consists of pairs of real arrays. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DEIGEN. | Class(es): D4a2 | Usage: CALL EIGEN (NM,N1A,WR,W1,Z) | On-line doc: CALL GAMSDOC EIGEN (or @PRT PORT+DOC.EIGEN) | Access: LIB NBS+PORT

EIGFR Eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a2 | Usage: CALL EIGFR (A,N1A,JOB,W,Z1,WK,IER) | On-line doc: CALL GAMSDOC EIGFR (or @PRT IMSL+DOC.EIGFR) | Access: LIB NBS+IMSL

EIGRS Eigenvalues and (optionally) eigenvectors of a real symmetric matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4a1 | Usage: CALL EIGRS (A,N1JOB,N,D,Z1,WK,IER) | On-line doc: CALL GAMSDOC EIGRS (or @PRT IMSL+DOC.EIGRS) | Access: LIB NBS+IMSL

EIGZC Eigenvalues and (optionally) eigenvectors of the system A*x=lambda B*x where A and B are complex matrices. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D4b4 | Usage: CALL EIGZC
EIGZF  Eigenvalues and (optionally) eigenvectors of the system \( A \times x = \lambda B \times x \) where \( A \) and \( B \) are real matrices.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): D4b2  | Usage: CALL EIGZF (A,IB,N,JOB,EIGA,EIGB,Z,IZ,WK,INFER,IER)  | On-line doc: CALL GAMSDOC EIGZC (or @PRT IMSL+DOC.EIGZC)  | Access: LIB NBS+IMSL

EIGZS  Eigenvalues and (optionally) eigenvectors of the system \( A \times x = \lambda B \times x \) where \( A \) and \( B \) are real symmetric matrices and \( B \) is positive definite.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): D4b1  | Usage: CALL EIGZS (A,B,N,JOB,D,Z,IZ,WK,IER)  | On-line doc: CALL GAMSDOC EIGZS (or @PRT IMSL+DOC.EIGZS)  | Access: LIB NBS+IMSL

ELMBAK  Forms eigenvectors of real general matrix from eigenvectors of upper Hessenberg matrix output from ELMHES.  | Portable single precision Fortran subroutine in EISPACK subsystem of CMLIB library.  | Class(es): D4c4  | Usage: CALL ELMBAK(NM,LOW,IGH,A,INT,M,Z)  | On-line doc: CALL GAMSDOC ELMBAK (or @PRT CMLIB+DOC.ELMBAK/EISPACK)  | Access: LIB NBS+CMLIB  | See also: ELMHES

ELMHES  Reduces real general matrix to upper Hessenberg form using stabilized elementary similarity transformations.  | Portable single precision Fortran subroutine in EISPACK subsystem of CMLIB library.  | Class(es): D4c1b2  | Usage: CALL ELMHES(NM,N,LOW,IGH,A,INT)  | On-line doc: CALL GAMSDOC ELMHES (or @PRT CMLIB+DOC.ELMHES/EISPACK)  | Access: LIB NBS+CMLIB

ELTRAN  Accumulates the stabilized elementary similarity transformations used in the reduction of a real general matrix to upper Hessenberg form by ELMHES.  | Portable single precision Fortran subroutine in EISPACK subsystem of CMLIB library.  | Class(es): D4c1  | Usage: CALL ELTRAN(NM,N,LOW,IGH,A,INT,Z)  | On-line doc: CALL GAMSDOC ELTRAN (or @PRT CMLIB+DOC.ELTRAN/EISPACK)  | Access: LIB NBS+CMLIB  | See also: ELMHES

ENTER  Save current error recovery mode and storage allocation status for PORT library programs.  | Proprietary single precision Fortran subroutine in PORT library.  | Class(es): R3c  | Usage: CALL ENTER (IRNEW)  | On-line doc: CALL GAMSDOC ENTER (or @PRT PORT+DOC.ENTER)  | Access: LIB NBS+PORT

ENTSRC  Saves current recovery mode status and sets a new one for PORT library programs.  | Proprietary single precision Fortran subroutine in PORT library.  | Class(es): R3a  | Usage: CALL ENTSRC (IROLD,IRNEW)  | On-line doc: CALL GAMSDOC ENTSRC (or @PRT PORT+DOC.ENTSRC)  | Access: LIB NBS+PORT

EPRINT  Print the current error message if the program is in the error state for PORT library programs.  | Proprietary single precision Fortran subroutine in PORT library.  | Class(es): R3c  | Usage: CALL EPRINT  | On-line doc: CALL GAMSDOC EPRINT (or @PRT PORT+DOC.EPRINT)  | Access: LIB NBS+PORT

EQRT1S  Smallest or largest \( m \) eigenvalues of a symmetric tridiagonal matrix.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): D4a5  | Usage: CALL EQRT1S (D,E2,N,M,ISW,IER)  | On-line doc: CALL GAMSDOC EQRT1S (or @PRT IMSL+DOC.EQRT1S)  | Access: LIB NBS+IMSL

EQRT2S  Eigenvalues and (optionally) eigenvectors of a symmetric tridiagonal matrix using the QL method.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): D4a5  | Usage: CALL EQRT2S (D,E,N,Z,IZ,IER)  | On-line doc: CALL GAMSDOC EQRT2S (or @PRT IMSL+DOC.EQRT2S)  | Access: LIB NBS+IMSL

EQRT3S  The smallest (or largest) eigenvalues of a tridiagonal matrix in algebraic value whose sum exceeds a given value.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): D4a5  | Usage: CALL EQRT3S (D,E2,N,VALUE,M,ISW,IFER,IER)  | On-line doc: CALL GAMSDOC EQRT3S (or @PRT IMSL+DOC.EQRT3S)  | Access: LIB NBS+IMSL

ERF  Error function, \( = \sqrt{2 \over \pi} \times \) the integral from 0 to \( x \) of \( \exp(-t^2)dt \).  | Portable single precision Fortran subroutine in FNLIB subsystem of CMLIB library.  | Double precision version is DERF.  | Class(es): C8a L5a1e  | Usage: R = ERF (X)  | On-line doc: CALL GAMSDOC ERF (or @PRT CMLIB+DOC.SUMMARY/FNLIB)  | Access: LIB NBS+CMLIB

ERF  Evaluate the error function.  | Note: The Fortran mathematical subroutine libraries may also contain ERF.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): C8a L5a1e  | Usage: X = ERF (Y)  | On-line doc: CALL GAMSDOC ERF (or @PRT IMSL+DOC.ERF)  | Access: LIB NBS+IMSL

ERFC  Complementary error function, \( = \sqrt{2 \over \pi} \times \) the integral from \( x \) to infinity of \( \exp(-t^2)dt \).  | Portable single precision Fortran subroutine in FNLIB subsystem of CMLIB library.  | Double precision version is DERFC.  | Class(es): C8a L5a1e  | Usage: R = ERFC (X)  | On-line doc: CALL GAMSDOC ERFC (or @PRT CMLIB+DOC.SUMMARY/FNLIB)  | Access: LIB NBS+CMLIB

ERFC  Evaluate the complementary error function.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): C8a L5a1e  | Usage: X = ERFC (Y)  | On-line doc: CALL GAMSDOC ERFC (or @PRT IMSL+DOC.ERFC)  | Access: LIB NBS+CMLIB

ERRINT  Computes error function and complementary error function to maximum machine accuracy. To change computers change one line.  | Portable double precision Fortran subroutine in STEGUN subsystem of MATHWARE library.  | Class(es): C8a L5a1e  | Usage: CALL ERRINT(X,ERF,ERFC)  | On-line doc: @PRT,S MATHWARE*STEGUN.ERRINT/DOC  | Access: See individual subroutine documentation

ERROFF  Turns off the error state for PORT library programs.  | Proprietary single precision Fortran subroutine in PORT library.  | Class(es): R3c  | Usage: CALL ERROFF  | On-line doc: CALL GAMSDOC ERROFF (or @PRT PORT+DOC.ERROFF)  | Access: LIB NBS+PORT

EV1CDF  Computes the cumulative distribution function value for the extreme value type 1 distribution.  | Portable single precision Fortran subroutine in DATAPAC library.  | Class(es): L5a1e  | Usage: CALL EV1CDF(X,CDP)  | On-line doc: CALL GAMSDOC EV1CDF (or @PRT DATAPAC+DOC.EV1CDF)  | Access: LIB NBS+DATAPAC
EV1PLT Generates an extreme value type 1 probability plot with mean = Euler's number = 0.57721566 and standard deviation = \( \pi/\sqrt{6} \).
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8ae | Usage: CALL EV1PLT(X,N) | On-line doc: CALL GAMSDOC EV1PLT (or ®RT DATAPAC®DOC.EV1PLT) | Access: LIB NBS®DATAPAC

EV1PPF Computes the percent point function value for the extreme value type 1 distribution with mean = Euler's number = 0.57721566.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8ae | Usage: CALL EV1PPF(P,PPP) | On-line doc: CALL GAMSDOC EV1PPF (or ®RT DATAPAC®DOC.EV1PPF) | Access: LIB NBS®DATAPAC

EV1RAN Generates a random sample of size N from the extreme value type 1 distribution with mean = Euler's number = 0.57721566.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8ae | Usage: CALL EV1RAN(N,ISTART,X) | On-line doc: CALL GAMSDOC EV1RAN (or ®RT DATAPAC®DOC.EV1RAN) | Access: LIB NBS®DATAPAC

EV2CDF Computes the cumulative distribution function value for the extreme value type 2 distribution with tail length parameter = GAMMA.

EV2PLT Generates an extreme value type 2 probability plot with tail length parameter = GAMMA.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8ae | Usage: CALL EV2PLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC EV2PLT (or ®RT DATAPAC®DOC.EV2PLT) | Access: LIB NBS®DATAPAC

EV2PPF Computes the percent point function value for the extreme value type 2 distribution with tail length parameter = GAMMA.

EV2RAN Generates a random sample of size N from the extreme value type 2 distribution with tail length parameter = GAMMA.
| Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8ae | Usage: CALL EV2RAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC EV2RAN (or ®RT DATAPAC®DOC.EV2RAN) | Access: LIB NBS®DATAPAC

EXINT Computes sequences of exponential integrals E(N+K,X) K=0,...,M-1 or EXP(X) times same to specified tolerance. Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. Double precision version is DEXINT.

EXP Exponential function, \( = e^{+x} \).

EXPCDF Computes the cumulative distribution function value for the exponential distribution with mean = 1 and standard deviation = 1.

EXPINT Computes exponential integral E sub n (x). Change computer by changing one line. Portable double precision Fortran subprogram in STEGUN sublibrary of MATHWARE library.
| Class(es): C8 | Usage: CALL EXPINT(RN,X,ENX,EXPNX,IERR) | On-line doc: ®RT,MATHWARE®STEGUN.EXPINT/DOC | Access: See individual sublibrary documentation

EXPPDF Computes the probability density function value for the exponential distribution with mean = 1 and standard deviation = 1.

EXPPLT Generates an exponential probability plot with mean = 1 and standard deviation = 1.

EXPPPF Computes the percent point function value for the exponential distribution with mean = 1 and standard deviation = 1.

EXPRAN Generates a random sample of size N from the exponential distribution with mean = 1 and standard deviation = 1.

EXPREL Relative error exponential from first order, \( = ((e^{+x})^{1}/x \). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DEXPREL.
| Class(es): C8b | Usage: R = EXPREL (X) | On-line doc: CALL GAMSDOC EXPREL (or ®RT CMLIB®DOC.SUMMARY/FNLIB) | Access: LIB NBS®CMLIB

EXPFS Computes the sparsity function value for the exponential distribution with mean = 1 and standard deviation = 1.

EXTREM Performs an extreme value analysis on the data in the input vector X.

EXTRMD Finds extreme points of a double precision functional defined on a mesh.
| Proprietary double precision Fortran subprogram in PORT library. Single precision version is EXTRMR.
| Class(es): N5a D1a | Usage: CALL EXTRMD (NPTS, FN, NEX, IEKT, IMAX, IMIN, IMAG) | On-line doc: CALL GAMSDOC EXTRMD (or ®RT PORT®DOC.EXTRMD) | Access: LIB NBS®PORT
EXTRMI Finds extremal points of an integer function defined on a mesh. Property single precision Fortran subprogram in PORT library. Class(es): N5a D1a2 Usage: CALL EXTRMI (NPTS, FN, NEX, IEXT, IMAX, IMIN, IMAG) On-line doc: CALL GAMSDOC EXTRMI (or PRT PORT*DOC.EXTRMI) Access: LIB NBS*PORT

EXTRMR Finds extremal points of a real function defined on a mesh. Property single precision Fortran subprogram in PORT library. Double precision version is EXTRMD. Class(es): N5a D1a2 Usage: CALL EXTRMR (NPTS, FN, NEX, IEXT, IMAX, IMIN, IMAG) On-line doc: CALL GAMSDOC EXTRMR (or PRT PORT*DOC.EXTRMR) Access: LIB NBS*PORT


January 1984

GAMS: Module Dictionary

C 67

F

F01AAE Calculates the approximate inverse of a real matrix by Crout's method with partial pivoting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AAF. | Class(es): D2a1 | Usage: CALL F01AAE(A, IA, N, UNIT, IUNIT, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F01AAE (or @PRT NAG+DOC.F01AAE) | Access: LIB NBS+NAG

F01AAP Calculates the approximate inverse of a real matrix by Crout's method with partial pivoting. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AAE. | Class(es): D2a1 | Usage: CALL F01AAP(A, IA, N, UNIT, IUNIT, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F01AAP (or @PRT NAG+DOC.F01AAP) | Access: LIB NBS+NAG

F01ABE Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. (Simplified parameter list). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ABF. | Class(es): D2b1b | Usage: CALL F01ABE(A, IA, N, B, IB, Z, IFAIL) | On-line doc: CALL GAMSDOC F01ABE (or @PRT NAG+DOC.F01ABE) | Access: LIB NBS+NAG

F01ABF Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. (Simplified parameter list.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ABE. | Class(es): D2b1b | Usage: CALL F01ABF(A, IA, N, B, IB, Z, IFAIL) | On-line doc: CALL GAMSDOC F01ABF (or @PRT NAG+DOC.F01ABF) | Access: LIB NBS+NAG

F01ACE Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ACF. | Class(es): D2b1b | Usage: CALL F01ACF(N, EPS, A, IA, B, IB, Z, L, IFAIL) | On-line doc: CALL GAMSDOC F01ACF (or @PRT NAG+DOC.F01ACE) | Access: LIB NBS+NAG

F01ACF Calculates the accurate inverse of a real symmetric positive definite matrix by Cholesky's method and iterative refinement. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ACE. | Class(es): D2b1b | Usage: CALL F01ACF(N, EPS, A, IA, B, IB, Z, L, IFAIL) | On-line doc: CALL GAMSDOC F01ACF (or @PRT NAG+DOC.F01ACF) | Access: LIB NBS+NAG

F01ADE F01ADE calculates the approximate inverse of a real symmetric positive definite matrix by Cholesky's method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ADF. | Class(es): D2b1b | Usage: CALL F01ADE(N, A, IA, IFAIL) | On-line doc: CALL GAMSDOC F01ADE (or @PRT NAG+DOC.F01ADE) | Access: LIB NBS+NAG

F01ADF Calculates the approximate inverse of a real symmetric positive definite matrix by Cholesky's method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ADE. | Class(es): D2b1b | Usage: CALL F01ADF(N, A, IA, IFAIL) | On-line doc: CALL GAMSDOC F01ADF (or @PRT NAG+DOC.F01ADF) | Access: LIB NBS+NAG

F01AEE Reduces the generalized eigenproblem Ax = (lambda)Bx, where A is real symmetric and B is real symmetric positive definite, to the standard symmetric eigenproblem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AED. | Class(es): D4tc1 | Usage: CALL F01AEE(N, A, IA, B, IB, DL, IFAIL) | On-line doc: CALL GAMSDOC F01AEE (or @PRT NAG+DOC.F01AEE) | Access: LIB NBS+NAG

F01AED Reduces the generalized eigenproblem Ax = (lambda)Bx to the standard symmetric eigenproblem Fx = (lambda)x. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AEE. | Class(es): D4tc1 | Usage: CALL F01AED(N, A, IA, B, IB, DL, IFAIL) | On-line doc: CALL GAMSDOC F01AED (or @PRT NAG+DOC.F01AED) | Access: LIB NBS+NAG

F01AF Derives eigenvectors of several generalized eigenproblems from the corresponding eigenvectors of the related standard symmetric eigenproblems. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AFF. | Class(es): D4ct | Usage: CALL F01AF(N, M1, M2, B, IB, DL, Z, IZ) | On-line doc: CALL GAMSDOC F01AF (or @PRT NAG+DOC.F01AF) | Access: LIB NBS+NAG

F01AG Derives eigenvectors of the eigensystems of the tridiagonal form produced by F01AGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AGF. | Class(es): D4ctb1 | Usage: CALL F01AGE(N, TO, A, IA, D, E) | On-line doc: CALL GAMSDOC F01AGE (or @PRT NAG+DOC.F01AGE) | Access: LIB NBS+NAG | See also: F02BEF,F02AVF,F02BFF

F01AFG Gives the Householder reduction of a real symmetric matrix to tridiagonal form for use in F02BEF, F02AVF and F02BFF. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AGF. | Class(es): D4ctb1 | Usage: CALL F01AG(N, TO, A, IA, D, E) | On-line doc: CALL GAMSDOC F01AG (or @PRT NAG+DOC.F01AG) | Access: LIB NBS+NAG | See also: F02BEF,F02AVF,F02BFF

F01AH Derives eigenvectors of a real symmetric matrix from the eigenvectors of the tridiagonal form produced by F01AGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AHF. | Class(es): D4ct | Usage: CALL F01AH(N, M1, M2, A, IA, E, Z) | On-line doc: CALL GAMSDOC F01AH (or @PRT NAG+DOC.F01AH) | Access: LIB NBS+NAG | See also: F01AHE

F01AHF Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form where the tridiagonal matrix was produced by F01AG. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AHF. | Class(es): D4ct | Usage: CALL F01AHF(N, M1, M2, A, IA, E, Z) | On-line doc: CALL GAMSDOC F01AHF (or @PRT NAG+DOC.F01AHF) | Access:
F01AJE  Gives the Hessenberg reduction of a real symmetric matrix A to tridiagonal form for use in F02AME. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AJF. | Class(es): D4cb1 | Usage: CALL F01AJE (N, TOL, A, IA, D, E, Z, IZ) | On-line doc: CALL GAMSDOC F01AJE (or ®PRT NAG*DOC.F01AJE) | Access: LIB NBS*NAG | See also: F02AME

F01AJF  Gives the Hessenberg reduction of a real symmetric matrix A to tridiagonal form for use in F02AMF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AJE. | Class(es): D4cb1 | Usage: CALL F01AJF (N, TOL, A, IA, D, E, Z, IZ) | On-line doc: CALL GAMSDOC F01AJF (or ®PRT NAG*DOC.F01AJF) | Access: LIB NBS*NAG | See also: F02AMF

F01AKE  Reduces a real unsymmetric matrix to upper Hessenberg form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AKF. | Class(es): D4cb2 | Usage: CALL F01AKE (N, K, L, A, IA, INTGER) | On-line doc: CALL GAMSDOC F01AKE (or ®PRT NAG*DOC.F01AKE) | Access: LIB NBS*NAG | See also: F01APE

F01AKF  Reduces a real unsymmetric matrix to upper Hessenberg form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AKE. | Class(es): D4cb2 | Usage: CALL F01AKF (N, K, L, A, IA, INTGER) | On-line doc: CALL GAMSDOC F01AKF (or ®PRT NAG*DOC.F01AKF) | Access: LIB NBS*NAG | See also: F01AFP

F01ALE  Transforms eigenvectors of a Hessenberg matrix to those of a real unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ALF. | Class(es): D4c4 | Usage: CALL F01ALE (K, L, IR, A, IA, INTGER, Z, IZ, N) | On-line doc: CALL GAMSDOC F01ALE (or ®PRT NAG*DOC.F01ALE) | Access: LIB NBS*NAG | See also: F01AFP

F01ALF  Transforms eigenvectors of a Hessenberg matrix to those of a real unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ALE. | Class(es): D4c4 | Usage: CALL F01ALF (K, L, IR, A, IA, INTGER, Z, IZ, N) | On-line doc: CALL GAMSDOC F01ALF (or ®PRT NAG*DOC.F01ALF) | Access: LIB NBS*NAG

F01AME  Reduces a complex unsymmetric matrix to complex upper Hessenberg form. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AMF. | Class(es): D4cb2 | Usage: CALL F01AME (N, K, L, AR, IAR, AI, IA, INTGER) | On-line doc: CALL GAMSDOC F01AME (or ®PRT NAG*DOC.F01AME) | Access: LIB NBS*NAG

F01AMF  Reduces a complex unsymmetric matrix to complex upper Hessenberg form. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AME. | Class(es): D4cb2 | Usage: CALL F01AMF (N, K, L, AR, IAR, AI, IA, INTGER) | On-line doc: CALL GAMSDOC F01AMF (or ®PRT NAG*DOC.F01AMF) | Access: LIB NBS*NAG

F01ANE  Transforms eigenvectors of a complex upper Hessenberg matrix to those of a complex unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01ANF. | Class(es): D4c4 | Usage: CALL F01ANE (K, L, IR, AR, IAR, AI, IA, INTGER, ZR, IZR, ZI, IZI, N) | On-line doc: CALL GAMSDOC F01ANE (or ®PRT NAG*DOC.F01ANE) | Access: LIB NBS*NAG

F01ANF  Transforms eigenvectors of a complex upper Hessenberg matrix to those of a complex unsymmetric matrix from which the Hessenberg matrix has previously been derived. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01ANE. | Class(es): D4c4 | Usage: CALL F01ANF (K, L, IR, AR, IAR, AI, IA, INTGER, ZR, IZR, ZI, IZI, N) | On-line doc: CALL GAMSDOC F01ANF (or ®PRT NAG*DOC.F01ANF) | Access: LIB NBS*NAG

F01APE  Forms the matrix of accumulated transformations from information left by F01AKE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01APF. | Class(es): D4c4 | Usage: CALL F01APE (N, K, L, INTGER, H, IH, V, IV) | On-line doc: CALL GAMSDOC F01APE (or ®PRT NAG*DOC.F01APE) | Access: LIB NBS*NAG | See also: F01AKE

F01APF  Forms the matrix of accumulated transformations from information left by F01AKE. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01APE. | Class(es): D4c4 | Usage: CALL F01APF (N, K, L, INTGER, H, IH, V, IV) | On-line doc: CALL GAMSDOC F01APF (or ®PRT NAG*DOC.F01APF) | Access: LIB NBS*NAG


F01AUE  Transforms eigenvectors of a balanced matrix to those of the original real unsymmetric matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AUF. | Class(es): D4c4 | Usage: CALL F01AUE (N, K, L, M, D, Z, IZ) | On-line doc: CALL GAMSDOC F01AUE (or ®PRT NAG*DOC.F01AUE) | Access: LIB NBS*NAG

F01AUF  Transforms eigenvectors of a balanced matrix to those of the original real unsymmetric matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AUE. | Class(es): D4c4 | Usage: CALL F01AUF (N, K, L, M, D, Z, IZ) | On-line doc: CALL GAMSDOC F01AUF (or ®PRT NAG*DOC.F01AUF) | Access: LIB NBS*NAG

F01AVE  Balances a complex matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01AVF. | Class(es): D4ca | Usage: CALL F01AVE (N, IB, AR, IAR, AI, IA, K, L, D) | On-line doc: CALL GAMSDOC F01AVE (or ®PRT NAG*DOC.F01AVE) | Access: LIB NBS*NAG

F01AVF  Balances a complex matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01AVE.
January 1984  
GAMS: Module Dictionary  
C 69

<table>
<thead>
<tr>
<th>Class(es): D4c1a</th>
<th>Usage: CALL F01AVF (N, IB, AR, IAR, AI, IAI, K, L, D)</th>
<th>On-line doc: CALL GAMSDOC F01AVF (or @PRT NAG+DOC.F01AVF)</th>
<th>Access: LIB NBS+NAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>F01AWE Transforms eigenvectors of a balanced matrix to those of the original complex matrix.</td>
<td>Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01AWF.</td>
<td>Class(es): D4c4</td>
<td>Usage: CALL F01AWE (N, K, L, M, D, ZR, IZR, ZI, IZI)</td>
</tr>
<tr>
<td>F01AWF Transforms eigenvectors of a balanced matrix to those of the original complex matrix.</td>
<td>Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01AWE.</td>
<td>Class(es): D4c4</td>
<td>Usage: CALL F01AWF (N, K, L, M, D, ZR, IZR, ZI, IZI)</td>
</tr>
<tr>
<td>F01AXE Reduces an m x n real matrix, m.GE.n, to upper triangular form for use in F04AME and F04ANE. Uses Householder transformations with column pivoting.</td>
<td>Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01AXF.</td>
<td>Class(es): D5</td>
<td>Usage: CALL F01AXE (M, N, Q, QR, ALPHA, IPIV, Y, E, IFAIL)</td>
</tr>
<tr>
<td>F01AXF Reduces an m x n real matrix, m.GE.n, to upper triangular form for use in F04AMF and F04ANF. Uses Householder transformations with column pivoting.</td>
<td>Proprietary single precision Fortran subroutine in NAG library. Single precision version is F01AXE.</td>
<td>Class(es): D5</td>
<td>Usage: CALL F01AXF (M, N, Q, QR, ALPHA, IPIV, Y, E, IFAIL)</td>
</tr>
<tr>
<td>F01AYE Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02BEE, F02AVE, and F02BFE. More economical in storage than similar F01AGE. Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01AYF.</td>
<td>Class(es): D4c1b</td>
<td>Usage: CALL F01AYE (N, TOL, A, IA, D, E, E2)</td>
<td>On-line doc: CALL GAMSDOC F01AYE (or @PRT NAG+DOC.F01AYE)</td>
</tr>
<tr>
<td>F01AYF Gives the Householder reduction of a real symmetric matrix A to tridiagonal form for use in F02BFE, F02AVF, and F02BBF. More economical in storage than similar F01AGF. Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01AYE.</td>
<td>Class(es): D4c1b</td>
<td>Usage: CALL F01AYF (N, TOL, A, IA, D, E, E2)</td>
<td>On-line doc: CALL GAMSDOC F01AYF (or @PRT NAG+DOC.F01AYF)</td>
</tr>
<tr>
<td>F01AZE Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form produced by F01AYE. Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01AZF.</td>
<td>Class(es): D4c4</td>
<td>Usage: CALL F01AZE (N, M1, M2, A, IA, Z, IZ)</td>
<td>On-line doc: CALL GAMSDOC F01AZE (or @PRT NAG+DOC.F01AZE)</td>
</tr>
<tr>
<td>F01AZF Derives eigenvectors of a real symmetric matrix from eigenvectors of the tridiagonal form produced by F01AYF. Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01AZE.</td>
<td>Class(es): D4c4</td>
<td>Usage: CALL F01AZF (N, M1, M2, A, IA, Z, IZ)</td>
<td>On-line doc: CALL GAMSDOC F01AZF (or @PRT NAG+DOC.F01AZF)</td>
</tr>
<tr>
<td>F01BCE Gives the Householder reduction of a complex Hermitian matrix to tridiagonal form for use in F01AVE and F02AYE. Proprietary single precision Fortran subroutine in NAG library. Single precision version is F01BCF.</td>
<td>Class(es): D4c1b</td>
<td>Usage: CALL F01BCE (N, TOL, A, IA, B, IB, D, E, WK1, WK2)</td>
<td>On-line doc: CALL GAMSDOC F01BCE (or @PRT NAG+DOC.F01BCE)</td>
</tr>
<tr>
<td>F01BCF Reduces the Householder reduction of a complex Hermitian matrix to tridiagonal form for use in F01AVF or F02AYF. Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01BCE.</td>
<td>Class(es): D4c1b</td>
<td>Usage: CALL F01BCF (N, TOL, A, IB, B, IB, D, E, WK1, WK2)</td>
<td>On-line doc: CALL GAMSDOC F01BCF (or @PRT NAG+DOC.F01BCF)</td>
</tr>
<tr>
<td>F01BDE Reduces eigenproblems A<em>B=x=lambda</em>x, x(transpose)<em>B+A=lambda</em>x to standard symmetric eigenproblem Q^T<em>Z=Z</em>Q^T. Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01BDF.</td>
<td>Class(es): D4c1c</td>
<td>Usage: CALL F01BDE (N, A, IB, B, IB, DL, IFAIL)</td>
<td>On-line doc: CALL GAMSDOC F01BDE (or @PRT NAG+DOC.F01BDE)</td>
</tr>
<tr>
<td>F01BDF Reduces the eigenproblems A<em>B=x=lambda</em>x, x(transpose)<em>B+A=lambda</em>x to standard symmetric eigenproblem Q^T<em>Z=Z</em>Q^T. Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01BDE.</td>
<td>Class(es): D4c1c</td>
<td>Usage: CALL F01BDF (N, A, IB, B, IB, DL, IFAIL)</td>
<td>On-line doc: CALL GAMSDOC F01BDF (or @PRT NAG+DOC.F01BDF)</td>
</tr>
<tr>
<td>F01BEA Derives eigenvectors y of the problems y(transpose)<em>A+B=lambda</em>y(transpose) and B<em>y=lambda</em>y from corresponding eigenvectors of derived standard symmetric eigenproblem. Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01BEF.</td>
<td>Class(es): D4c4</td>
<td>Usage: CALL F01BEA (N, M1, M2, B, IB, DL, V, IV)</td>
<td>On-line doc: CALL GAMSDOC F01BEA (or @PRT NAG+DOC.F01BEA)</td>
</tr>
<tr>
<td>F01BEF Derives eigenvectors y of the problems y(transpose)<em>A+B=lambda</em>y(transpose) and B<em>y=lambda</em>y from corresponding eigenvectors of the derived standard symmetric eigenproblem. Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01BEE.</td>
<td>Class(es): D4c4</td>
<td>Usage: CALL F01BEF (N, M1, M2, B, IB, DL, V, IV)</td>
<td>On-line doc: CALL GAMSDOC F01BEF (or @PRT NAG+DOC.F01BEF)</td>
</tr>
<tr>
<td>F01BLE Calculates the rank and pseudo-inverse of an m x n real matrix A, m.GE.n, rank(A).LE.n, using Householder's factorization. Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01BLF.</td>
<td>Class(es): D9</td>
<td>Usage: CALL F01BLE (M, N, T, A, IA, IMAX, IRANK, INC, D, U, IU, DU, IFAIL)</td>
<td>On-line doc: CALL GAMSDOC F01BLE (or @PRT NAG+DOC.F01BLE)</td>
</tr>
<tr>
<td>F01BLF Calculates the rank and pseudo-inverse of an m x n real matrix A, m.GE.n, rank(A).LE.n, using Householder's factorization.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BBL. | Class(es): D9 | Usage: CALL F01BBL (M, N, T, A, IA, AJMAX, IANK, INC, D, U, DU, IFAIL) | On-line doc: CALL GAMSDOC F01BBL (or CPRT NAG*DOC.F01BBL) | Access: LIB NBS+NAG

F01BNE Performs the Cholesky decomposition of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BNF. | Class(es): D2d1b | Usage: CALL F01BNE (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BNE (or CPRT NAG*DOC.F01BNE) | Access: LIB NBS+NAG

F01BNEF Performs the Cholesky decomposition of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BNE. | Class(es): D2d1b | Usage: CALL F01BNEF (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BNEF (or CPRT NAG*DOC.F01BNEF) | Access: LIB NBS+NAG

F01BPE Determines the inverse of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BPF. | Class(es): D2d1b | Usage: CALL F01BPE (N, A, IA, V, IFAIL) | On-line doc: CALL GAMSDOC F01BPE (or CPRT NAG*DOC.F01BPE) | Access: LIB NBS+NAG

F01BPF Determines the inverse of a complex positive definite Hermitian matrix, given the lower triangle of the matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BPF. | Class(es): D2d1b | Usage: CALL F01BPF (N, A, IA, V, IFAIL) | On-line doc: CALL GAMSDOC F01BPF (or CPRT NAG*DOC.F01BPF) | Access: LIB NBS+NAG

F01BQF Forms the Cholesky decomposition of real symmetric matrix G whose lower triangle only is stored. If G not positive definite, forms Cholesky decomposition of G+E, E a diagonal matrix. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BQF. | Class(es): D2d1b | Usage: CALL F01BQF (N, EPS, RL, IRL, D, IFAIL) | On-line doc: CALL GAMSDOC F01BQF (or CPRT NAG*DOC.F01BQF) | Access: LIB NBS+NAG

F01BQF Performs the Cholesky decomposition of a real symmetric matrix G whose lower triangle only is stored. If G not positive definite, forms the Cholesky decomposition of G+E, E a diagonal matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BQF. | Class(es): D2d1b | Usage: CALL F01BQF (N, EPS, RL, IRL, D, IFAIL) | On-line doc: CALL GAMSDOC F01BQF (or CPRT NAG*DOC.F01BQF) | Access: LIB NBS+NAG

F01BRE Decomposes real sparse matrix. Either forms LU-decomposition of permutation of entire matrix, or permutes matrix to block lower triangular form and then decomposes diagonal blocks. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BRF. | Class(es): D2d1t | Usage: CALL F01BRE (N, NZ, A, LCN, IRN, LIRN, ICN, PIVOT, IKEEP, IW, W, LBLOCK, GROW, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BRE (or CPRT NAG*DOC.F01BRE) | Access: LIB NBS+NAG

F01BRF Decomposes a real sparse matrix. Either forms LU-decomposition of permutation of entire matrix, or permutes matrix to block lower triangular form, and then decomposes the diagonal blocks. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BRF. | Class(es): D2d1t | Usage: CALL F01BRF (N, NZ, A, LCN, IRN, LIRN, ICN, PIVOT, IKEEP, IW, W, LBLOCK, GROW, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BRF (or CPRT NAG*DOC.F01BRF) | Access: LIB NBS+NAG

F01BSE Decomposes a real sparse matrix using the pivotal sequence previously obtained by F01BRE when a matrix of the same sparsity pattern was decomposed. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BSE. | Class(es): D2d1t | Usage: CALL F01BSE (N, NZ, A, LCN, IVECT, JVECT, ICN, IKEEP, IW, W, GROW, ETA, RPMIN, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BSE (or CPRT NAG*DOC.F01BSE) | Access: LIB NBS+NAG

F01BSF Decomposes a real sparse matrix using the pivotal sequence previously obtained by F01BRF when a matrix of the same sparsity pattern was decomposed. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BSF. | Class(es): D2d1t | Usage: CALL F01BSF (N, NZ, A, LCN, IVECT, JVECT, ICN, IKEEP, IW, W, GROW, ETA, RPMIN, ABORT, IDISP, IFAIL) | On-line doc: CALL GAMSDOC F01BSF (or CPRT NAG*DOC.F01BSF) | Access: LIB NBS+NAG

F01BTE Decomposes a real matrix into product of triangular matrices LU by Gaussian elimination with partial pivoting. Block-column method used for efficiency on paged virtual machines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BTF. | Class(es): D2d1t | Usage: CALL F01BTE (N, A, IA, P, DP, IFAIL) | On-line doc: CALL GAMSDOC F01BTE (or CPRT NAG*DOC.F01BTE) | Access: LIB NBS+NAG


F01BUE Decomposes a symmetric positive definite matrix into form U*L*D*L`*Ut where U is unit upper triangular, L is unit lower triangular, D is diagonal. Precedes F01BVE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BUF. | Class(es): D2d1b | Usage: CALL F01BUE (N, , M1, K, A, I, W, IFAIL) | On-line doc: CALL GAMSDOC F01BUE (or CPRT NAG*DOC.F01BUE) | Access: LIB NBS+NAG

F01BUF Decomposes a symmetric positive definite matrix into form U*L*D*L`*Ut where U is unit upper triangular, L is unit lower triangular, D is diagonal. Precedes F01BVF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01BUF. | Class(es): D2d1b | Usage: CALL F01BUF (N, , M1, K, A, I, W, IFAIL) | On-line doc: CALL GAMSDOC F01BUF (or CPRT NAG*DOC.F01BUF) | Access: LIB NBS+NAG

F01BVF Transforms the generalized symmetric eigenproblem A*x=lambda*B*x to equivalent standard eigenproblem C*y=lambda*y. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01BVF. | Class(es): D4d1c | Usage: CALL F01BVE (N, MA1, MB1, M3, K, A, IA, IB, IC, W, IFAIL) | On-line doc: CALL GAMSDOC F01BVF (or CPRT NAG*DOC.F01BVF) | Access: LIB NBS+NAG | See also: F01BUE
GAMS: Module Dictionary

B, C symmetric band matrices, B positive definite + decomposed C\(=\)lambda B. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01BVE. | Class(es): D4cE | Usage: CALL F01BVF (N, M1, M2, M, A, IA, IB, B, IC, W, IFAIL) | On-line doc: CALL GAMSDOC F01BVF (or OPRT NAG*DOC.F01BVF) | Access: LIB NBS*NAG | See also: F01BVF

F01BWE Reduces a symmetric band matrix to tridiagonal form. This routine normally used with F02AVE to find all eigenvalues of A. For selected values, use F02BME. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01BWF. | Class(es): D4cbl1 | Usage: CALL F01BWF (N, M1, M, A, IA, D, E) | On-line doc: CALL GAMSDOC F01BWF (or OPRT NAG*DOC.F01BWE) | Access: LIB NBS*NAG | See also: F01BWF

F01BVF Reduces a symmetric band matrix to tridiagonal form. This routine normally used with F02AVF to find all eigenvalues of A. For selected eigenvalues, use F02BVF. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01BVE. | Class(es): D4cbl1 | Usage: CALL F01BVF (N, M1, M, A, IA, D, E) | On-line doc: CALL GAMSDOC F01BVF (or OPRT NAG*DOC.F01BVF) | Access: LIB NBS*NAG | See also: F01BVF

F01BKE Performs the Cholesky factorization \(U^T\) of a real symmetric positive definite matrix A. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01BKF. | Class(es): D2blb | Usage: CALL F01BKE (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BKE (or OPRT NAG*DOC.F01BKE) | Access: LIB NBS*NAG

F01BKF Performs the Cholesky factorization \(U^T\) of a real symmetric positive definite matrix A. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01BKE. | Class(es): D2blb | Usage: CALL F01BKE (N, A, IA, P, IFAIL) | On-line doc: CALL GAMSDOC F01BKF (or OPRT NAG*DOC.F01BKF) | Access: LIB NBS*NAG

F01CAB Sets elements of an \(m\times n\) matrix A to zero. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CAF. | Class(es): D1bl | Usage: CALL F01CAF (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CAF (or OPRT NAG*DOC.F01CAE) | Access: LIB NBS*NAG

F01CAF Sets elements of an \(m\times n\) matrix A to zero. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01CAB. | Class(es): D1bl | Usage: CALL F01CAF (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CAF (or OPRT NAG*DOC.F01CAF) | Access: LIB NBS*NAG

F01CBF Sets elements A(i,j) to one if \(i=j\) and zero otherwise, where 1.E.1.E.m and 1.E.1.E.n. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CBF. | Class(es): D1bl | Usage: CALL F01CBF (A, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CBF (or OPRT NAG*DOC.F01CBF) | Access: LIB NBS*NAG

F01CCE Subtracts elements of the matrix C from elements of the matrix B and stores the results in elements of the matrix A. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CEF. | Class(es): D1b5 | Usage: CALL F01CEF (A, B, C, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CDE (or OPRT NAG*DOC.F01CDE) | Access: LIB NBS*NAG

F01CDE Adds elements of the \(m\times n\) matrices B and C and stores the results in elements of the matrix A. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CDF. | Class(es): D1b5 | Usage: CALL F01CDF (A, B, C, M, N, IFAIL) | On-line doc: CALL GAMSDOC F01CDF (or OPRT NAG*DOC.F01CDF) | Access: LIB NBS*NAG

F01CF3 Copies elements of the matrix B into different positions in the matrix A. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CFF. | Class(es): D1b5 | Usage: CALL F01CFF (A, M, NA, NA, P, Q, B, MB, NB, MB, N1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CFF (or OPRT NAG*DOC.F01CFE) | Access: LIB NBS*NAG

F01CFF Copies elements of the matrix B into different positions in the matrix A. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01CFF. | Class(es): D1b5 | Usage: CALL F01CFF (A, M, NA, NA, P, Q, B, MB, NB, MB, M1, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CFF (or OPRT NAG*DOC.F01CFF) | Access: LIB NBS*NAG

F01CGE Adds elements of the matrix B to elements in different positions in the matrix A. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CGF. | Class(es): D1b5 | Usage: CALL F01CGF (A, M, NA, NA, P, Q, B, MB, NB, MB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CGF (or OPRT NAG*DOC.F01CGF) | Access: LIB NBS*NAG

F01CGF Adds elements of the matrix B to elements in different positions in the matrix A. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is F01CGF. | Class(es): D1b5 | Usage: CALL F01CGF (A, M, NA, NA, P, Q, B, MB, NB, MB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CGF (or OPRT NAG*DOC.F01CGF) | Access: LIB NBS*NAG

F01CHE Subtracts elements of the matrix B from elements in different position in the matrix A. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is F01CHF. | Class(es): D1b5 | Usage: CALL F01CHE (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL) | On-line doc: CALL GAMSDOC F01CHE (or OPRT NAG*DOC.F01CHE) | Access: LIB NBS*NAG
F01CHF  Subtracts elements of the matrix B from elements in a different position in the matrix A.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CHE.  |Class(es): D1b6  |Usage: CALL F01CHF (A, MA, NA, P, Q, B, MB, NB, M1, M2, N1, N2, IFAIL)  |On-line doc: CALL GAMSDOC F01CHF (or @PRT NAG+DOC.F01CHF)  |Access: LIB NBS+NAG

F01CKE  Returns with the result of multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CKF.  |Class(es): D1b6  |Usage: CALL F01CKE (A, B, C, N, P, M, Z, IZ, OPT, IFAIL)  |On-line doc: CALL GAMSDOC F01CKE (or @PRT NAG+DOC.F01CKE)  |Access: LIB NBS+NAG

F01CKF  Returns with the result of multiplication of two matrices B and C in the matrix A, with the option to overwrite B or C.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CKE.  |Class(es): D1b6  |Usage: CALL F01CKF (A, B, C, N, P, M, Z, IZ, OPT, IFAIL)  |On-line doc: CALL GAMSDOC F01CKF (or @PRT NAG+DOC.F01CKF)  |Access: LIB NBS+NAG

F01CLE  Post-multiplies the matrix B with the transpose of the matrix C and places the result in the matrix A.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CLF.  |Class(es): D1b6  |Usage: CALL F01CLE (A, B, C, N, P, M, IFAIL)  |On-line doc: CALL GAMSDOC F01CLE (or @PRT NAG+DOC.F01CLE)  |Access: LIB NBS+NAG

F01CLF  Post-multiplies the matrix B with the transpose of the matrix C and places the result in the matrix A.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CLE.  |Class(es): D1b6  |Usage: CALL F01CLF (A, B, C, N, P, M, IFAIL)  |On-line doc: CALL GAMSDOC F01CLF (or @PRT NAG+DOC.F01CLF)  |Access: LIB NBS+NAG

F01CME  Copies elements of one matrix into a row of a matrix.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CMF.  |Class(es): D1a5 D1b6  |Usage: CALL F01CME (V, M, A, LA, I)  |On-line doc: CALL GAMSDOC F01CME (or @PRT NAG+DOC.F01CME)  |Access: LIB NBS+NAG

F01CMF  Copies elements of one matrix into a row of a matrix.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CME.  |Class(es): D1a5 D1b6  |Usage: CALL F01CMF (V, M, A, LA, I)  |On-line doc: CALL GAMSDOC F01CMF (or @PRT NAG+DOC.F01CMF)  |Access: LIB NBS+NAG

F01CNE  Copies a vector of length M into a row of a matrix.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CNP.  |Class(es): D1a5 D1b6  |Usage: CALL F01CNE (V, M, A, LA, I)  |On-line doc: CALL GAMSDOC F01CNE (or @PRT NAG+DOC.F01CNE)  |Access: LIB NBS+NAG

F01CNP  Copies a vector of length M into a row of a matrix.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CNE.  |Class(es): D1a5 D1b6  |Usage: CALL F01CNP (V, M, A, LA, I)  |On-line doc: CALL GAMSDOC F01CNP (or @PRT NAG+DOC.F01CNP)  |Access: LIB NBS+NAG

F01CPE  Copies the contents of a vector into a second vector.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CPF.  |Class(es): D1a5  |Usage: CALL F01CPE (A, B, N)  |On-line doc: CALL GAMSDOC F01CPE (or @PRT NAG+DOC.F01CPE)  |Access: LIB NBS+NAG

F01CPF  Copies the contents of a vector into a second vector.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CPE.  |Class(es): D1a5  |Usage: CALL F01CPF (A, B, N)  |On-line doc: CALL GAMSDOC F01CPF (or @PRT NAG+DOC.F01CPF)  |Access: LIB NBS+NAG

F01CQE  Sets the elements of a vector to zero.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CQF.  |Class(es): D1a1  |Usage: CALL F01CQE (A, N)  |On-line doc: CALL GAMSDOC F01CQE (or @PRT NAG+DOC.F01CQE)  |Access: LIB NBS+NAG

F01CQF  Sets the elements of a vector to zero.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CQE.  |Class(es): D1a1  |Usage: CALL F01CQF (A, N)  |On-line doc: CALL GAMSDOC F01CQF (or @PRT NAG+DOC.F01CQF)  |Access: LIB NBS+NAG

F01CRE  Re-orders the elements of a vector of length m+n, containing an m+n matrix, A, so that the new vector contains the transpose matrix.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CRF.  |Class(es): D1b3  |Usage: CALL F01CRE(A, M, N, MN, MOVE, LMOVE, IFAIL)  |On-line doc: CALL GAMSDOC F01CRE (or @PRT NAG+DOC.F01CRE)  |Access: LIB NBS+NAG

F01CRF  Re-orders the elements of a vector of length m+n, containing an m+n matrix, A, so that the new vector contains the transpose matrix.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CRE.  |Class(es): D1b3  |Usage: CALL F01CRF(A, M, N, MN, MOVE, LMOVE, IFAIL)  |On-line doc: CALL GAMSDOC F01CRF (or @PRT NAG+DOC.F01CRF)  |Access: LIB NBS+NAG

F01CSE  Forms the product c=A*b where b is a vector and A is a symmetric matrix whose lower triangle is stored by rows in a one-dimensional array.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01CSF.  |Class(es): D1b4  |Usage: CALL F01CSE (A, LA, B, N, C)  |On-line doc: CALL GAMSDOC F01CSE (or @PRT NAG+DOC.F01CSE)  |Access: LIB NBS+NAG

F01CSF  Forms the product c=A*b where b is a vector and A is a symmetric matrix whose lower triangle is stored by rows in a one-dimensional array.  |Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01CSE.  |Class(es): D1b4  |Usage: CALL F01CSF (A, LA, B, N, C)  |On-line doc: CALL GAMSDOC F01CSF (or @PRT NAG+DOC.F01CSF)  |Access: LIB NBS+NAG

F01DAE  Returns the sum of an initial value and a scalar product, using basic precision arithmetic.  |Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DAF.  |Class(es): D1a4  |Usage: D = F01DAE (L, M, C1, IRA, ICB, A, LA, B, IB, N)  |On-line doc: CALL GAMSDOC F01DAE (or @PRT NAG+DOC.F01DAE)  |Access: LIB NBS+NAG

F01DAF  Returns the sum of an initial value and a scalar product, using basic precision arithmetic.  |Proprietary double precision Fortran subprogram in NAG library. Double precision version is F01DAF.  |Class(es): D1a4  |Usage: D = F01DAF (L, M, C1, IRA, ICB, A, LA, B, IB, N)  |On-line doc: CALL GAMSDOC F01DAF (or @PRT NAG+DOC.F01DAF)  |Access: LIB NBS+NAG
subprogram in NAG library. Single precision version is F01DAE.  | Class(es): D1a4  | Usage: D = F01DAF (L, M, CI, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DAF (or @PRT NAG+DOC.F01DAF) | Access: LIB NBS+NAG

F01DBE Returns the sum of an initial value and a scalar product, using additional precision arithmetic.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DBF.  | Class(es): D1a4  | Usage: D = F01DBE (L, M, CI, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DBE (or @PRT NAG+DOC.F01DBE) | Access: LIB NBS+NAG

F01DBF Returns the sum of an initial value and a scalar product, using additional precision arithmetic.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DBE.  | Class(es): D1a4  | Usage: D = F01DBF (L, M, CI, IRA, ICB, A, IA, B, IB, N) | On-line doc: CALL GAMSDOC F01DBF (or @PRT NAG+DOC.F01DBF) | Access: LIB NBS+NAG

F01DCE Computes the value of a complex scalar product and subtracts it from a complex initial value, using basic precision arithmetic.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DCF.  | Class(es): D1a4  | Usage: CALL F01DCE (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, CI) | On-line doc: CALL GAMSDOC F01DCE (or @PRT NAG+DOC.F01DCE) | Access: LIB NBS+NAG

F01DCF Computes the value of a complex scalar product and subtracts it from a complex initial value, using basic precision arithmetic.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DCE.  | Class(es): D1a4  | Usage: CALL F01DCF (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, CI) | On-line doc: CALL GAMSDOC F01DCF (or @PRT NAG+DOC.F01DCF) | Access: LIB NBS+NAG

F01DDE Computes the value of a complex scalar product and subtracts it from a complex initial value, using additional precision arithmetic.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DDF.  | Class(es): D1a4  | Usage: CALL F01DDE (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, CI) | On-line doc: CALL GAMSDOC F01DDE (or @PRT NAG+DOC.F01DDE) | Access: LIB NBS+NAG

F01DDF Computes the value of a complex scalar product and subtracts it from a complex initial value, using additional precision arithmetic.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DDE.  | Class(es): D1a4  | Usage: CALL F01DDF (L, M, CX, IRA, ICB, A, IA, B, IB, N, CR, CI) | On-line doc: CALL GAMSDOC F01DDF (or @PRT NAG+DOC.F01DDF) | Access: LIB NBS+NAG

F01DDE Returns the value of the scalar product of two arrays of length N, using basic precision arithmetic.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01DEF.  | Class(es): D1a4  | Usage: D = F01DEE (A, B, N) | On-line doc: CALL GAMSDOC F01DEE (or @PRT NAG+DOC.F01DEE) | Access: LIB NBS+NAG

F01DEF Returns the value of the scalar product of two arrays of length N, using basic precision arithmetic.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01DDE.  | Class(es): D1a4  | Usage: CALL F01DEF (A, B, N) | On-line doc: CALL GAMSDOC F01DEF (or @PRT NAG+DOC.F01DEF) | Access: LIB NBS+NAG

F01LBE LU-factorisation, real band matrix.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01LBF.  | Class(es): D2a2  | Usage: CALL F01LBE (N, M1, M2, A, IA, AL, IL, IN, IV, IFAIL) | On-line doc: CALL GAMSDOC F01LBE (or @PRT NAG+DOC.F01LBE) | Access: LIB NBS+NAG

F01LBF LU-factorisation, real band matrix.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01LBE.  | Class(es): D2a2  | Usage: CALL F01LBF (N, M1, M2, A, IA, AL, IL, IN, IV, IFAIL) | On-line doc: CALL GAMSDOC F01LBF (or @PRT NAG+DOC.F01LBF) | Access: LIB NBS+NAG

F01LZE Reduction by similarity transformations, real matrix to bidiagonal form.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01LZF.  | Class(es): D8  | Usage: CALL F01LZE (N, A, NRA, C, NRC, WANTB, B, WANTQ, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, D, E, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC F01LZE (or @PRT NAG+DOC.F01LZE) | Access: LIB NBS+NAG

F01LZF Reduction by similarity transformations, real matrix to bidiagonal form.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01LZE.  | Class(es): D8  | Usage: CALL F01LZF (N, A, NRA, C, NRC, WANTB, B, WANTQ, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, D, E, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC F01LZF (or @PRT NAG+DOC.F01LZF) | Access: LIB NBS+NAG

F01MCE LDLt-factorisation, real symmetric positive-definite variable-bandwidth matrix.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01MCF.  | Class(es): D2b2  | Usage: CALL F01MCE (N, A, LAL, NROW, AL, D, IFAIL) | On-line doc: CALL GAMSDOC F01MCE (or @PRT NAG+DOC.F01MCE) | Access: LIB NBS+NAG

F01MCF LDLt-factorisation, real symmetric positive-definite variable-bandwidth matrix.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01MCE.  | Class(es): D2b2  | Usage: CALL F01MCF (N, A, LAL, NROW, AL, D, IFAIL) | On-line doc: CALL GAMSDOC F01MCF (or @PRT NAG+DOC.F01MCF) | Access: LIB NBS+NAG

F01QAE QR-factorisation, real m x n matrix (m > n).  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01QAF.  | Class(es): D5  | Usage: CALL F01QAE (M, N, A, NRA, C, NRC, Z, IFAIL) | On-line doc: CALL GAMSDOC F01QAE (or @PRT NAG+DOC.F01QAE) | Access: LIB NBS+NAG

F01QAF QR-factorisation, real m x n matrix (m > n).  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01QAE.  | Class(es): D5  | Usage: CALL F01QAF (M, N, A, NRA, C, NRC, Z, IFAIL) | On-line doc: CALL GAMSDOC F01QAF (or @PRT NAG+DOC.F01QAF) | Access: LIB NBS+NAG

F01QBE RQ-factorisation, real m x n matrix (m < n).  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F01QBF.  | Class(es): D5  | Usage: CALL F01QBE (M, N, A, NRA, C, NRC, WORK, IFAIL) | On-line doc: CALL GAMSDOC
January 1984

GAMS: Module Dictionary

F01QBE (or @PRT NAG+DOC.F01QBE) | Access: LIB NBS*NAG

F01QBF
RQ-factorisation, real m x n matrix (m <= n). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F01QBF. | Class(es): D5 | Usage: CALL F01QBF (M, N, A, NRA, C, NRC, WORK, IFAIL) | On-line doc: CALL GAMSDOC F01QBF (or @PRT NAG+DOC.F01QBF) | Access: LIB NBS*NAG

F02AAE Real symmetric matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AAF. | Class(es): D4a4 | Usage: CALL F02AAE (A, IA, N, R, E, IFAIL) | On-line doc: CALL GAMSDOC F02AAE (or @PRT NAG+DOC.F02AAE) | Access: LIB NBS*NAG

F02AAF Real symmetric matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AEE. | Class(es): D4a4 | Usage: CALL F02AEE (A, IA, N, R, E, IFAIL) | On-line doc: CALL GAMSDOC F02AEE (or @PRT NAG+DOC.F02AEE) | Access: LIB NBS*NAG

F02ABE Real symmetric matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ABF. | Class(es): D4a4 | Usage: CALL F02ABF (A, IA, N, R, V, IV, E, IFAIL) | On-line doc: CALL GAMSDOC F02ABF (or @PRT NAG+DOC.F02ABF) | Access: LIB NBS*NAG

F02ABF Real symmetric matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ABE. | Class(es): D4a4 | Usage: CALL F02ABE (A, IA, N, R, V, IV, E, IFAIL) | On-line doc: CALL GAMSDOC F02ABE (or @PRT NAG+DOC.F02ABE) | Access: LIB NBS*NAG

F02ADE Generalised real symmetric eigenproblem $Ax - kBx$ with positive-definite $B$ (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ADF. | Class(es): D4b1 | Usage: CALL F02ADE (A, IA, B, IB, N, R, DE, IFAIL) | On-line doc: CALL GAMSDOC F02ADE (or @PRT NAG+DOC.F02ADE) | Access: LIB NBS*NAG

F02ADF Generalised real symmetric eigenproblem $Ax - kBx$ with positive-definite $B$ (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ADE. | Class(es): D4b1 | Usage: CALL F02ADE (A, IA, B, IB, N, R, DE, IFAIL) | On-line doc: CALL GAMSDOC F02ADE (or @PRT NAG+DOC.F02ADE) | Access: LIB NBS*NAG

F02AEE Generalised real symmetric eigenproblem $Ax - kBx$ with positive-definite $B$ (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is F02AEE. | Class(es): D4b1 | Usage: CALL F02AEE (A, IA, B, IB, N, R, V, IV, DE, E, IFAIL) | On-line doc: CALL GAMSDOC F02AEE (or @PRT NAG+DOC.F02AEE) | Access: LIB NBS*NAG

F02AEF Generalised real symmetric eigenproblem $Ax - kBx$ with positive-definite $B$ (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AEE. | Class(es): D4b1 | Usage: CALL F02AEE (A, IA, B, IB, N, R, V, IV, DE, E, IFAIL) | On-line doc: CALL GAMSDOC F02AEE (or @PRT NAG+DOC.F02AEE) | Access: LIB NBS*NAG

F02AFF Real matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AFF. | Class(es): D4a2 | Usage: CALL F02AFF (A, IA, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AFF (or @PRT NAG+DOC.F02AFF) | Access: LIB NBS*NAG

F02AGF Real matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AGF. | Class(es): D4a2 | Usage: CALL F02AGF (A, IA, N, RR, RI, VR, IVR, VI, IV, 1 INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AGF (or @PRT NAG+DOC.F02AGF) | Access: LIB NBS*NAG

F02AGF Real matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AGE. | Class(es): D4a2 | Usage: CALL F02AGE (A, IA, N, RR, RI, VR, IVR, VI, IV, 1 INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AGE (or @PRT NAG+DOC.F02AGE) | Access: LIB NBS*NAG

F02AHE Complex matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AHF. | Class(es): D4a4 | Usage: CALL F02AHE (AR, IAR, AI, IA, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AHE (or @PRT NAG+DOC.F02AHE) | Access: LIB NBS*NAG

F02AJF Complex matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AJE. | Class(es): D4a4 | Usage: CALL F02AJE (AR, IAR, AI, IA, N, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AJE (or @PRT NAG+DOC.F02AJE) | Access: LIB NBS*NAG

F02AKE Complex matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AKF. | Class(es): D4a4 | Usage: CALL F02AKE (AR, IAR, AI, IA, N, RR, RI, VR, IVR, VI, IVI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AKE (or @PRT NAG+DOC.F02AKE) | Access: LIB NBS*NAG

F02AKF Complex matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AKE. | Class(es): D4a4 | Usage: CALL F02AKE (AR, IAR, AI, IA, N, RR, RI, VR, IVR, VI, IVI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AKE (or @PRT NAG+DOC.F02AKE) | Access: LIB NBS*NAG

F02AME Real symmetric matrix, all eigenvalues and eigenvectors, after reduction to tridiagonal form by F01AJE, QL algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AMF. | Class(es): D4a2a | Usage: CALL F02AME (N, EPS, D, E, V, IV, IFAIL) | On-line doc: CALL GAMSDOC F02AME (or @PRT NAG+DOC.F02AME) | Access: LIB NBS*NAG
See also: F01AJE

F02AMF Real symmetric matrix, all eigenvalues and eigenvectors, after reduction to tridiagonal form by F01AJF, QL algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AME. | Class(es): D4c2a | Usage: CALL F02AMF (N, EPS, D, E, V, IV, IFAIL) | On-line doc: CALL GAMSDOC F02AMF (or @PRT NAG+DOC.F02AMF) | Access: LIB NBS+NAG | See also: F01AJF

F02ANE Complex upper Hessenberg matrix, all eigenvalues, LR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ANF. | Class(es): D4c2b | Usage: CALL F02ANE (N, EPS, H, HR, HI, IHI, RR, RI, IFAIL) | On-line doc: CALL GAMSDOC F02ANE (or @PRT NAG+DOC.F02ANF) | Access: LIB NBS+NAG

F02ANF Complex upper Hessenberg matrix, all eigenvalues, LR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ANF. | Class(es): D4c2b | Usage: CALL F02ANF (N, EPS, H, HR, HI, IHI, RR, RI, IFAIL) | On-line doc: CALL GAMSDOC F02ANF (or @PRT NAG+DOC.F02ANF) | Access: LIB NBS+NAG

F02APE Real upper Hessenberg matrix, all eigenvalues, QR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02APF. | Class(es): D4c2b | Usage: CALL F02APE (N, EPS, H, H, RR, RI, ICNT, IFAIL) | On-line doc: CALL GAMSDOC F02APF (or @PRT NAG+DOC.F02APF) | Access: LIB NBS+NAG

F02APF Real upper Hessenberg matrix, all eigenvalues, QR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02APF. | Class(es): D4c2b | Usage: CALL F02APF (N, EPS, H, H, RR, RI, ICNT, IFAIL) | On-line doc: CALL GAMSDOC F02APF (or @PRT NAG+DOC.F02APF) | Access: LIB NBS+NAG

F02AQE Real matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AKE and F01APE, QR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AQF. | Class(es): D4c2b | Usage: CALL F02AQE (N, K, L, EPS, H, H, V, V, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AQF (or @PRT NAG+DOC.F02AQE) | Access: LIB NBS+NAG

F02AQF Real matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AKF and F01APF, QR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AQF. | Class(es): D4c2b | Usage: CALL F02AQF (N, K, L, EPS, H, H, V, V, RR, RI, INTGER, IFAIL) | On-line doc: CALL GAMSDOC F02AQF (or @PRT NAG+DOC.F02AQF) | Access: LIB NBS+NAG

F02ARE Complex matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AME, LR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02ARF. | Class(es): D4c2b | Usage: CALL F02ARE (N, K, L, EPS, INTGER, HR, IHR, HI, IHI, RR, RI, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02ARF (or @PRT NAG+DOC.F02ARE) | Access: LIB NBS+NAG | See also: F01AMF

F02ARF Complex matrix, all eigenvalues and eigenvectors, after reduction to upper Hessenberg form by F01AMF, LR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02ARF. | Class(es): D4c2b | Usage: CALL F02ARF (N, K, L, EPS, INTGER, HR, IHR, HI, IHI, RR, RI, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02ARF (or @PRT NAG+DOC.F02ARF) | Access: LIB NBS+NAG

F02AVE Real symmetric tridiagonal matrix, all eigenvalues, QR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AVF. | Class(es): D4a5 | Usage: CALL F02AVE (N, EPS, D, E, IFAIL) | On-line doc: CALL GAMSDOC F02AVE (or @PRT NAG+DOC.F02AVE) | Access: LIB NBS+NAG

F02AVF Real symmetric tridiagonal matrix, all eigenvalues, QR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AVF. | Class(es): D4a5 | Usage: CALL F02AVF (N, EPS, D, E, IFAIL) | On-line doc: CALL GAMSDOC F02AVF (or @PRT NAG+DOC.F02AVF) | Access: LIB NBS+NAG

F02AWE Complex Hermitian matrix, (black box), all eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AWF. | Class(es): D4a5 | Usage: CALL F02AWE (AR, IAR, AI, IAI, N, R, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AWE (or @PRT NAG+DOC.F02AWE) | Access: LIB NBS+NAG

F02AWF Complex Hermitian matrix, (black box), all eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AWF. | Class(es): D4a5 | Usage: CALL F02AWF (AR, IAR, AI, IAI, N, R, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AWF (or @PRT NAG+DOC.F02AWF) | Access: LIB NBS+NAG

F02AXE Complex Hermitian matrix, (black box), all eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AXF. | Class(es): D4a5 | Usage: CALL F02AXE (AR, IAR, AI, IAI, N, R, VR, IVR, VI, IVI, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AXE (or @PRT NAG+DOC.F02AXE) | Access: LIB NBS+NAG

F02AXF Complex Hermitian matrix, (black box), all eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AXE. | Class(es): D4a5 | Usage: CALL F02AXF (AR, IAR, AI, IAI, N, R, VR, IVR, VI, IVI, WK1, WK2, WK3, IFAIL) | On-line doc: CALL GAMSDOC F02AXF (or @PRT NAG+DOC.F02AXF) | Access: LIB NBS+NAG

F02AYE Complex Hermitian matrix, all eigenvalues and eigenvectors, after reduction to real tridiagonal form by F01BCE, QR algorithm. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02AYF. | Class(es): D4c2a | Usage: CALL F02AYE (N, EPS, D, E, VR, IVR, VI, IVI, IFAIL) | On-line doc: CALL GAMSDOC F02AYE (or @PRT NAG+DOC.F02AYE) | Access: LIB NBS+NAG | See also: F01BCE

F02AYF Complex Hermitian matrix, all eigenvalues and eigenvectors, after reduction to real tridiagonal form by F01BCF, QR algorithm. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02AYE. | Class(es): D4c2a | Usage: CALL
F02AYF (N, EPS, D, E, VR, IVR, VI, IFAIL) | On-line doc: CALL GAMSDOC F02AYF (or QPRG NAG*DOC.F02AYF) | Access: LIB NBS+NAG | See also: F01BCF

F02BBE Real symmetric matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BBF. | Class(es): D4a1 | Usage: CALL F02BBE (A, IA, N, ALB, UB, M, MM, R, V, IV, D, E, E2, X, G, LOG, ICOUNT, IFAIL) | On-line doc: CALL GAMSDOC F02BBE (or QPRG NAG*DOC.F02BBE) | Access: LIB NBS+NAG


F02BDE Complex matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BDF. | Class(es): D4a4 | Usage: CALL F02BDE (AR, IAR, AI, IA, N, ALB, UB, M, MM, RR, RI, VR, IV, VI, INTGER, C, BR, IBR, BI, IBI, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BDF (or QPRG NAG*DOC.F02BDE) | Access: LIB NBS+NAG

F02BDF Complex matrix, (black box), selected eigenvalues and eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BDE. | Class(es): D4a4 | Usage: CALL F02BDF (AR, IAR, AI, IA, N, ALB, UB, M, MM, RR, RI, VR, IV, VI, INTGER, C, BR, IBR, BI, IBI, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BDF (or QPRG NAG*DOC.F02BDF) | Access: LIB NBS+NAG

F02BEE Real symmetric tridiagonal matrix, selected eigenvalues and eigenvectors, bisection and inverse iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BEF. | Class(es): D4a5 | Usage: CALL F02BEE (N, D, ALB, UB, EPS, EPS1, E, E2, M, MM, R, V, IV, ICOUNT, X, LOG, IFAIL) | On-line doc: CALL GAMSDOC F02BEE (or QPRG NAG*DOC.F02BEE) | Access: LIB NBS+NAG

F02BEF Real symmetric tridiagonal matrix, selected eigenvalues and eigenvectors, bisection and inverse iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BEE. | Class(es): D4a5 | Usage: CALL F02BEF (N, D, ALB, UB, EPS, EPS1, E, E2, M, MM, R, V, IV, ICOUNT, X, LOG, IFAIL) | On-line doc: CALL GAMSDOC F02BEF (or QPRG NAG*DOC.F02BEF) | Access: LIB NBS+NAG

F02BFE Real symmetric tridiagonal matrix, selected eigenvalues, bisection. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BFF. | Class(es): D4a5 | Usage: CALL F02BFE (D, E, E2, N, M1, M2, MM12, EPS1, EPS, EPS2, IZ, R, WU) | On-line doc: CALL GAMSDOC F02BFE (or QPRG NAG*DOC.F02BFE) | Access: LIB NBS+NAG

F02BFF Real symmetric tridiagonal matrix, selected eigenvalues, bisection. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BFE. | Class(es): D4a5 | Usage: CALL F02BFF (D, E, E2, N, M1, M2, MM12, EPS1, EPS, EPS2, IZ, R, WU) | On-line doc: CALL GAMSDOC F02BFF (or QPRG NAG*DOC.F02BFF) | Access: LIB NBS+NAG

F02BJE Generalised eigenproblem Ax-λBx, QZ algorithm (black box), real matrices, all eigenvalues and (optionally) eigenvectors. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BJF. | Class(es): D4b2 | Usage: CALL F02BJE (N, A, IA, B, IB, EPS1, ALFR, ALFI, BETA, MATV, V, IV, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02BJE (or QPRG NAG*DOC.F02BJE) | Access: LIB NBS+NAG

F02BJF Generalised eigenproblem Ax-λBx, QZ algorithm (black box), real matrices, all eigenvalues and (optionally) eigenvectors. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BJE. | Class(es): D4b2 | Usage: CALL F02BJF (N, A, IA, B, IB, EPS1, ALFR, ALFI, BETA, MATV, V, IV, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02BJF (or QPRG NAG*DOC.F02BJF) | Access: LIB NBS+NAG

F02BKE Compute selected eigenvectors of a real upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BKEF. | Class(es): D4c3 | Usage: CALL F02BKE (N, M, H, IH, RI, C, RR, V, IV, B, IB, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BKE (or QPRG NAG*DOC.F02BKE) | Access: LIB NBS+NAG | See also: F02AKE

F02BKF Compute selected eigenvectors of a real upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F02BKE. | Class(es): D4c3 | Usage: CALL F02BKF (N, M, H, IH, RI, C, RR, V, IV, B, IB, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BKF (or QPRG NAG*DOC.F02BKF) | Access: LIB NBS+NAG | See also: F02AKE

F02BLE Compute selected eigenvectors of a complex upper Hessenberg matrix by inverse iteration, given estimates of their associated eigenvalues. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F02BLF. | Class(es): D4c3 | Usage: CALL F02BLE (N, M, HR, IHR, HI, IHI, RI, C, RR, VR, IVR, VI, IIV, BR, IBR, BI, IBI, U, V, IFAIL) | On-line doc: CALL GAMSDOC F02BLE (or QPRG NAG*DOC.F02BLE) | Access: LIB NBS+NAG | See also: F02AKE

F02BLF Compute selected eigenvectors of a complex upper Hessenberg matrix by inverse iteration, given estimates of their associated
eigenvalues. | Proprietary single precision Fortran program in NAG library. Single precision version is F02BLE. | Class(es): D4c3 | Usage: CALL F02BFL (N, M, HR, IHR, IHI, IHI, RI, C, RR, VR, IVR, VI, IBI, BR, IBR, BI, IBI, U, W, IFAIL) | On-line doc: CALL GAMSDOC F02BFL (or @PRT NAG=DOC.F02BFL) | Access: LIB NBS+NAG | See also: F02AKP

F02GJE Generalised eigenproblem Ax=λBx, QZ algorithm (black box), complex matrices, eigenvalues and (optionally) eigenvectors. | Proprietary single precision Fortran program in NAG library. Double precision version is F02GJF. | Class(es): D4b4 | Usage: CALL F02GJE (N, A, IAR, AI, IAR, BR, IBR, BI, IBI, EPSI, ALFR, ALFI, BETA, MATV VR, IVR, VI, IBI, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02GJE (or @PRT NAG=DOC.F02GJE) | Access: LIB NBS+NAG

F02GJF Generalised eigenproblem Ax=λBx, QZ algorithm (black box), complex matrices, eigenvalues and (optionally) eigenvectors. | Proprietary double precision Fortran program in NAG library. Single precision version is F02GJE. | Class(es): D4b4 | Usage: CALL F02GJF (N, A, IAR, AI, IAR, BR, IBR, BI, IBI, EPSI, ALFR, ALFI, BETA, MATV VR, IVR, VI, IBI, ITER, IFAIL) | On-line doc: CALL GAMSDOC F02GJF (or @PRT NAG=DOC.F02GJF) | Access: LIB NBS+NAG

F02SDE Generalised real eigenproblem Ax=λBx, where A and B are band matrices, eigenvector by inverse iteration. | Proprietary single precision Fortran program in NAG library. Double precision version is F02SDF. | Class(es): D6b5 | Usage: CALL F02SDE (N, MAI, MBA, A, IA, B, IB, SYM, RELEP, RMU, VEC, D, INT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02SDE (or @PRT NAG=DOC.F02SDE) | Access: LIB NBS+NAG

F02SDF Generalised real eigenproblem Ax=λBx, where A and B are band matrices, eigenvector by inverse iteration. | Proprietary double precision Fortran program in NAG library. Single precision version is F02SDE. | Class(es): D6b5 | Usage: CALL F02SDF (N, MAI, MBA, A, IA, B, IB, SYM, RELEP, RMU, VEC, D, INT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02SDF (or @PRT NAG=DOC.F02SDF) | Access: LIB NBS+NAG

F02SZE Singular value decomposition of a real bidiagonal matrix. | Proprietary single precision Fortran program in NAG library. Double precision version is F02SZF. | Class(es): D6 | Usage: CALL F02SZE (N, D, E, SV, WANTB, B, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, WORK1 WORK2, WORK3, IFAIL) | On-line doc: CALL GAMSDOC F02SZE (or @PRT NAG=DOC.F02SZE) | Access: LIB NBS+NAG

F02SZF Singular value decomposition of a real bidiagonal matrix. | Proprietary double precision Fortran program in NAG library. Single precision version is F02SZE. | Class(es): D6 | Usage: CALL F02SZF (N, D, E, SV, WANTB, B, WANTY, Y, NRY, LY, WANTZ, Z, NRZ, NCZ, WORK1 WORK2, WORK3, IFAIL) | On-line doc: CALL GAMSDOC F02SZF (or @PRT NAG=DOC.F02SZF) | Access: LIB NBS+NAG

F02WAE Singular value decomposition of a real m x n matrix, singular values and right singular vectors, (m>—n). | Proprietary single precision Fortran program in NAG library. Double precision version is F02WAF. | Class(es): D6 | Usage: CALL F02WAE (N, M, N, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WAE (or @PRT NAG=DOC.F02WAE) | Access: LIB NBS+NAG

F02WAF Singular value decomposition of a real m x n matrix, singular values and right singular vectors, (m>—n). | Proprietary double precision Fortran program in NAG library. Single precision version is F02WAE. | Class(es): D6 | Usage: CALL F02WAF (N, M, N, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WAF (or @PRT NAG=DOC.F02WAF) | Access: LIB NBS+NAG

F02WBE Singular value decomposition of a real m x n matrix, singular values and right singular vectors, (m<—n). | Proprietary single precision Fortran program in NAG library. Double precision version is F02WBF. | Class(es): D6 | Usage: CALL F02WBE (M, N, A, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WBE (or @PRT NAG=DOC.F02WBE) | Access: LIB NBS+NAG

F02WBF Singular value decomposition of a real m x n matrix, singular values and right singular vectors, (m<—n). | Proprietary double precision Fortran program in NAG library. Single precision version is F02WBE. | Class(es): D6 | Usage: CALL F02WBF (M, N, A, NRA, WANTB, B, SV, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WBF (or @PRT NAG=DOC.F02WBF) | Access: LIB NBS+NAG

F02WCE Singular value decomposition of a real m x n matrix, singular values and left and right singular vectors. | Proprietary single precision Fortran program in NAG library. Double precision version is F02WCF. | Class(es): D6 | Usage: CALL F02WCE (M, N, MINMN, A, NRA, Q, NRQ, SV, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WCE (or @PRT NAG=DOC.F02WCE) | Access: LIB NBS+NAG

F02WCF Singular value decomposition of a real m x n matrix, singular values and left and right singular vectors. | Proprietary double precision Fortran program in NAG library. Single precision version is F02WCF. | Class(es): D6 | Usage: CALL F02WCF (M, N, MINMN, A, NRA, Q, NRQ, SV, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WCF (or @PRT NAG=DOC.F02WCF) | Access: LIB NBS+NAG

F02WDE S.V.D. of a real m x n matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation (m>—n). | Proprietary single precision Fortran program in NAG library. Double precision version is F02WDF. | Class(es): D5 D6 | Usage: CALL F02WDE (M, N, A, NRA, WANTB, B, TOI, TOL, SV, IRANK, Z, SV, WANTR, R, NRR, WANTPT, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WDE (or @PRT NAG=DOC.F02WDE) | Access: LIB NBS+NAG

F02WDP S.V.D. of a real m x n matrix, singular values and (optionally) right singular vectors, optionally or conditionally following QR-factorisation (m>—n). | Proprietary double precision Fortran program in NAG library. Single precision version is F02WDE. | Class(es): D5 D6 | Usage: CALL F02WDF (M, N, A, NRA, WANTB, B, TOI, TOL, SV, IRANK, Z, SV, WANTR, R, NRR, WANTPT, PT, NRPT, WORK, LWORK, IFAIL) | On-line doc: CALL GAMSDOC F02WDF (or @PRT NAG=DOC.F02WDF) | Access: LIB NBS+NAG

F03AAE Determinant (black box), real matrix. | Proprietary single precision Fortran program in NAG library. Double precision version is F03AAF. | Class(es): D3a1 | Usage: CALL F03AAE (A, IA, N, DET, WKSPEC, IFAIL) | On-line doc: CALL GAMSDOC F03AAE (or
GAMS: Module Dictionary

January 1984

@PRT NAG*DOC.F03AAE] | Access: LIB NBS*NAG

F03AAF Determinant (black box), real matrix. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F03AEF. | Class(es): D3a1 | Usage: CALL F03AAF (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03AAF (or @PRT NAG*DOC.F03AFAF) | Access: LIB NBS*NAG

F03ABE Determinant (black box), real symmetric positive-definite matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03ABF. | Class(es): D3b1 | Usage: CALL F03ABE (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ABE (or @PRT NAG*DOC.F03ABE) | Access: LIB NBS*NAG

F03ABF Determinant (black box), real symmetric positive-definite matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03ABE. | Class(es): D3b1 | Usage: CALL F03ABF (A, IA, N, DET, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ABF (or @PRT NAG*DOC.F03ABF) | Access: LIB NBS*NAG

F03ACE Determinant (black box), real symmetric positive-definite band matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03ACF. | Class(es): D3c2 | Usage: CALL F03ACE (A, IA, N, DETR, DETI, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ACE (or @PRT NAG*DOC.F03ACF) | Access: LIB NBS*NAG

F03ACF Determinant (black box), real symmetric positive-definite band matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03ACF. | Class(es): D3c2 | Usage: CALL F03ACF (A, IA, N, DETR, DETI, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ACF (or @PRT NAG*DOC.F03ACF) | Access: LIB NBS*NAG

F03ADE Determinant (black box), complex matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03AED. | Class(es): D3c1 | Usage: CALL F03ADE (A, IA, N, DETR, DETI, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ADE (or @PRT NAG*DOC.F03AED) | Access: LIB NBS*NAG

F03ADF Determinant (black box), complex matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03ADF. | Class(es): D3c1 | Usage: CALL F03ADF (A, IA, N, DETR, DETI, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F03ADF (or @PRT NAG*DOC.F03ADF) | Access: LIB NBS*NAG

F03AEE LDL-factorisation and determinant, real symmetric positive-definite matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03AEF. | Class(es): D2b1b D3b1b | Usage: CALL F03AEE (N, A, I, P, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AEE (or @PRT NAG*DOC.F03AEE) | Access: LIB NBS*NAG

F03AEF LDL-factorisation and determinant, real symmetric positive-definite matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03AEE. | Class(es): D2b1b D3b1b | Usage: CALL F03AEF (N, A, I, P, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AEF (or @PRT NAG*DOC.F03AEF) | Access: LIB NBS*NAG

F03AFE LU-factorisation and determinant, real matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03AFF. | Class(es): D2a1 D3a1 | Usage: CALL F03AFE (N, EPS, A, IA, D1, ID, P, IFAIL) | On-line doc: CALL GAMSDOC F03AFF (or @PRT NAG*DOC.F03AFF) | Access: LIB NBS*NAG

F03AFF LU-factorisation and determinant, real matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03AFF. | Class(es): D2a1 D3a1 | Usage: CALL F03AFF (N, EPS, A, IA, D1, ID, P, IFAIL) | On-line doc: CALL GAMSDOC F03AFF (or @PRT NAG*DOC.F03AFF) | Access: LIB NBS*NAG

F03AGE LDL-factorisation and determinant, real symmetric positive-definite band matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03AGF. | Class(es): D2b2 D3b2 | Usage: CALL F03AGE (N, M, A, IA, RL, IL, M1, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AGE (or @PRT NAG*DOC.F03AGE) | Access: LIB NBS*NAG

F03AGF LDL-factorisation and determinant, real symmetric positive-definite band matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03AGF. | Class(es): D2b2 D3b2 | Usage: CALL F03AGF (N, M, A, IA, RL, IL, M1, D1, ID, IFAIL) | On-line doc: CALL GAMSDOC F03AGF (or @PRT NAG*DOC.F03AGF) | Access: LIB NBS*NAG

F03AHE LU-factorisation and determinant, complex matrix. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03AHF. | Class(es): D2c1 D3c1 | Usage: CALL F03AHE (N, A, IA, DETR, DETI, ID, RINT, IFAIL) | On-line doc: CALL GAMSDOC F03AHE (or @PRT NAG*DOC.F03AHE) | Access: LIB NBS*NAG

F03AHF LU-factorisation and determinant, complex matrix. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03AHF. | Class(es): D2c1 D3c1 | Usage: CALL F03AHF (N, A, IA, DETR, DETI, ID, RINT, IFAIL) | On-line doc: CALL GAMSDOC F03AHF (or @PRT NAG*DOC.F03AHF) | Access: LIB NBS*NAG

F03AME Determinant of a complex Hermitian positive-definite matrix, after factorisation by F01BNE. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F03AMF. | Class(es): D3d1b | Usage: CALL F03AME (N, TEN, P, D1, D2) | On-line doc: CALL GAMSDOC F03AME (or @PRT NAG*DOC.F03AME) | Access: LIB NBS*NAG | See also: F01BNE

F03AMF Determinant of a complex Hermitian positive-definite matrix, after factorisation by F01BNF. | Proprietary double precision Fortran subprogram in NAG library.
Single precision version is F03AME. | Class(es): D3d1b | Usage: CALL F03AMF (N, TEN, P, D1, D2) | On-line doc: CALL GAMSDOC F03AMF (or @PRT NAG*DOC.F03AMF) | Access: LIB NBS*NAG | See also: F01BNF

F04AEF Simultaneous linear equations (black box), real matrix, approximate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is F04AFF. | Class(es): D2a1 | Usage: CALL F04AEF (A, IA, B, IB, N, M, C, IC, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04AEF (or @PRT NAG*DOC.F04AEF) | Access: LIB NBS*NAG
F04AAP Simultaneous linear equations (black box), real matrix, approximate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AAE. | Class(es): D2a1 Usage: CALL F04AAF | On-line loc: CALL GAMSDOC F04AAF (or @PRT NAG DOC F04AAF) Access: LIB NBS+NAG

F04ABE Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AFB. | Class(es): D2b1 Usage: CALL F04ABE | On-line loc: CALL GAMSDOC F04ABE (or @PRT NAG DOC F04ABE) Access: LIB NBS+NAG

F04ABF Simultaneous linear equations (black box), real symmetric positive-definite matrix, accurate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ABE. | Class(es): D2b1 Usage: CALL F04ABF | On-line loc: CALL GAMSDOC F04ABF (or @PRT NAG DOC F04ABF) Access: LIB NBS+NAG

F04ACE Simultaneous linear equations (black box), real symmetric positive-definite bandmatrix, approximate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ACF. | Class(es): D2b2 Usage: CALL F04ACE | On-line loc: CALL GAMSDOC F04ACE (or @PRT NAG DOC F04ACE) Access: LIB NBS+NAG

F04ACF Simultaneous linear equations (black box), real symmetric positive-definite bandmatrix, approximate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ACE. | Class(es): D2b2 Usage: CALL F04ACF | On-line loc: CALL GAMSDOC F04ACF (or @PRT NAG DOC F04ACF) Access: LIB NBS+NAG

F04ADE Simultaneous linear equations (black box), complex matrix, approximate solution, multiple right hand sides. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ADF. | Class(es): D2c1 Usage: CALL F04ADE | On-line loc: CALL GAMSDOC F04ADE (or @PRT NAG DOC F04ADE) Access: LIB NBS+NAG

F04ADF Simultaneous linear equations (black box), complex matrix, approximate solution, multiple right hand sides. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ADE. | Class(es): D2c1 Usage: CALL F04ADF | On-line loc: CALL GAMSDOC F04ADF (or @PRT NAG DOC F04ADF) Access: LIB NBS+NAG

F04AEF Calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AEF. | Class(es): D2b1 Usage: CALL F04AEF | On-line loc: CALL GAMSDOC F04AEF (or @PRT NAG DOC F04AEF) Access: LIB NBS+NAG

F04AFF Calculates the accurate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is F04AEF. | Class(es): D2b1 Usage: CALL F04AFF | On-line loc: CALL GAMSDOC F04AFF (or @PRT NAG DOC F04AFF) Access: LIB NBS+NAG

F04AGE Calculates the approximate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AGF. | Class(es): D2b1 Usage: CALL F04AGE | On-line loc: CALL GAMSDOC F04AGE (or @PRT NAG DOC F04AGE) Access: LIB NBS+NAG

F04AGF Calculates the approximate solution of a set of real symmetric positive definite linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AGF. | Class(es): D2b1 Usage: CALL F04AGF | On-line loc: CALL GAMSDOC F04AGF (or @PRT NAG DOC F04AGF) Access: LIB NBS+NAG

F04AHE Calculates the accurate solution of set of real linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AHE. | Class(es): D2a1 Usage: CALL F04AHE | On-line loc: CALL GAMSDOC F04AHE (or @PRT NAG DOC F04AHE) Access: LIB NBS+NAG

F04AHF Calculates the accurate solution of set of real linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AHF. | Class(es): D2a1 Usage: CALL F04AHF | On-line loc: CALL GAMSDOC F04AHF (or @PRT NAG DOC F04AHF) Access: LIB NBS+NAG

F04AJE Calculates the approximate solution of set of real linear equations with multiple right hand sides, AX = B, where A has been decomposed into triangular matrices using F03AEE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AJF. | Class(es): D2a1 Usage: CALL F04AJE | On-line loc: CALL GAMSDOC F04AJE (or @PRT NAG DOC F04AJE) Access: LIB NBS+NAG
F04AJF Calculates the approximate solution of a set of linear equations with multiple right-hand sides, AX = B, where A has been decomposed into triangular matrices using F03AFF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AJE. | Class(es): D2a1 | Usage: CALL F04AJF (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AJF (or @PRT NAG+DOC.F04AJF) | Access: LIB NBS+NAG | See also: F03AFF

F04AKE Calculates the approximate solution of a set of complex linear equations with multiple right-hand sides, AX = B, where A has been decomposed into triangular matrices using F03AHE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AKF. | Class(es): D2c1 | Usage: CALL F04AKE (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AKE (or @PRT NAG+DOC.F04AKE) | Access: LIB NBS+NAG | See also: F03AHE

F04AKF Calculates the approximate solution of a set of complex linear equations with multiple right-hand sides, AX = B, where A has been decomposed into triangular matrices using F03AHE. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AKE. | Class(es): D2c1 | Usage: CALL F04AKF (N, IR, A, IA, P, B, IB) | On-line doc: CALL GAMSDOC F04AKF (or @PRT NAG+DOC.F04AKF) | Access: LIB NBS+NAG | See also: F03AKF

F04ALE Calculates the approximate solution of a set of real symmetric positive definite band linear equations with multiple right-hand sides, AX = B, where A has been decomposed into triangular matrices using F03AGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ALF. | Class(es): D2b2 | Usage: CALL F04ALE (N, M, IR, RL, IRL, M1, B, IB, X, IX) | On-line doc: CALL GAMSDOC F04ALE (or @PRT NAG+DOC.F04ALE) | Access: LIB NBS+NAG | See also: F03ALF


F04ANE Least-squares, m real equations in n unknowns, rank = m > n, approximate solution (after factorisation by F01AXE). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ANF. | Class(es): D9 | Usage: CALL F04ANE (M, N, QR, IQR, ALPHA, E, Y, Z, R, IPIV, IFAIL) | On-line doc: CALL GAMSDOC F04ANE (or @PRT NAG+DOC.F04ANE) | Access: LIB NBS+NAG

F04ANF Least-squares, m real equations in n unknowns, rank = m > n, approximate solution (after factorisation by F01AXF). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ANF. | Class(es): D9 | Usage: CALL F04ANF (M, N, QR, IQR, ALPHA, E, Y, Z, R, IPIV, IFAIL) | On-line doc: CALL GAMSDOC F04ANF (or @PRT NAG+DOC.F04ANF) | Access: LIB NBS+NAG | See also: F01ANF

F04AQE Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right-hand side, AX = b, where A has been decomposed into LDLT using F01BQF. (Economical storage.). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04AQF. | Class(es): D2b1b | Usage: CALL F04AQE (N, M, RL, D, B, X) | On-line doc: CALL GAMSDOC F04AQE (or @PRT NAG+DOC.F04AQE) | Access: LIB NBS+NAG | See also: F01BQF

F04AQF Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right-hand side, AX = b, where A has been decomposed into LDLT using F01BQF. (Economical storage.). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04AQE. | Class(es): D2b1b | Usage: CALL F04AQF (N, M, RL, D, B, X) | On-line doc: CALL GAMSDOC F04AQF (or @PRT NAG+DOC.F04AQF) | Access: LIB NBS+NAG | See also: F01BQF

F04ARE Simultaneous linear equations (black box), real matrix, approximate solution, one right-hand side. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ARF. | Class(es): D2a1 | Usage: CALL F04ARE (A, IA, B, N, C, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04AR (or @PRT NAG+DOC.F04ARE) | Access: LIB NBS+NAG

F04ARF Simultaneous linear equations (black box), real matrix, approximate solution, one right-hand side. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ARE. | Class(es): D2a1 | Usage: CALL F04ARF (A, IA, B, N, C, WKSPCE, IFAIL) | On-line doc: CALL GAMSDOC F04ARF (or @PRT NAG+DOC.F04ARF) | Access: LIB NBS+NAG

F04ASE Simultaneous linear equations (black box), real matrix, accurate solution, one right-hand side. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ASF. | Class(es): D2b1b | Usage: CALL F04ASE (A, IA, B, N, C, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC F04ASE (or @PRT NAG+DOC.F04ASE) | Access: LIB NBS+NAG

F04ASF Simultaneous linear equations (black box), real matrix, accurate solution, one right-hand side. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04ASE. | Class(es): D2b1b | Usage: CALL F04ASF (A, IA, B, N, C, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC F04ASF (or @PRT NAG+DOC.F04ASF) | Access: LIB NBS+NAG

F04ATE Simultaneous linear equations (black box), real matrix, accurate solution, one right-hand side. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04ATF. | Class(es): D2a1 | Usage: CALL F04ATE (A, IA, B, N, C, AA, IAA,
GAMS: Module Dictionary

C 81

WKS1, WKS2, IFAIL) On-line doc: CALL GAMSDOC F04ATE (or @PRT NAG*DOC.F04ATE) Access: LIB NBS*NAG


F04AWE Calculates the approximate solutions of a set of complex linear equations with multiple right hand sides, AX = B, where A is positive definite Hermitian, following the Cholesky decomposition of A by F01BNE. Proprietary single precision Fortran program in NAG library. Double precision version is F04AWF. Class(es): D2d1b Usage: CALL F04AWE (N, IR, A, 1A, P, B, IB, X, IX) On-line doc: CALL GAMSDOC F04AWE (or @PRT NAG*DOC.F04AWE) Access: LIB NBS*NAG See also: F01BNE

F04AWF Calculates the approximate solutions of a set of complex linear equations with multiple right hand sides, AX = B, where A is positive definite Hermitian, following the Cholesky decomposition of A by F01BNE. Proprietary double precision Fortran program in NAG library. Single precision version is F04AWE. Class(es): D2d1b Usage: CALL F04AWF (N, IR, A, 1A, P, B, IB, X, IX) On-line doc: CALL GAMSDOC F04AWF (or @PRT NAG*DOC.F04AWF) Access: LIB NBS*NAG See also: F01BNE

F04AXE Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, AX = b or Ax = b, where A has been decomposed by F01BRE or F01BSE. Proprietary single precision Fortran program in NAG library. Double precision version is F04AXF. Class(es): D2a4 Usage: CALL F04AXE (N, A, LICN, ICN, IKEEP, RHS, W, MTYPE, IDISP, RESID) On-line doc: CALL GAMSDOC F04AXE (or @PRT NAG*DOC.F04AXE) Access: LIB NBS*NAG See also: F01BRE F01BSE

F04AFX Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, AX = b or Ax = b, where A has been decomposed by F01BFR or F01BSF. Proprietary double precision Fortran program in NAG library. Single precision version is F04AXE. Class(es): D2a4 Usage: CALL F04AFX (N, A, LICN, ICN, IKEEP, RHS, W, MTYPE, IDISP, RESID) On-line doc: CALL GAMSDOC F04AFX (or @PRT NAG*DOC.F04AFX) Access: LIB NBS*NAG See also: F01BFR F01BSF

F04AYE Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, AX = B, where A has been decomposed into triangular matrices using F01BTE. Proprietary single precision Fortran program in NAG library. Double precision version is F04AYF. Class(es): D2a1 Usage: CALL F04AYE (N, IR, A, 1A, P, B, IB, IFAIL) On-line doc: CALL GAMSDOC F04AYE (or @PRT NAG*DOC.F04AYE) Access: LIB NBS*NAG See also: F01BTE

F04AYF Calculates the approximate solution of a set of real sparse linear equations with a single right hand side, AX = B, where A has been decomposed into triangular matrices using F01BTF. Proprietary double precision Fortran program in NAG library. Single precision version is F04AYE. Class(es): D2a1 Usage: CALL F04AYF (N, IR, A, 1A, P, B, IB, IFAIL) On-line doc: CALL GAMSDOC F04AYF (or @PRT NAG*DOC.F04AYF) Access: LIB NBS*NAG See also: F01BTF

F04AZE Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, AX = b, where A has been decomposed into triangular matrices using F01BXE. Proprietary single precision Fortran program in NAG library. Double precision version is F04AZF. Class(es): D2b1b Usage: CALL F04AZE (N, IR, A, 1A, P, B, IB, IFAIL) On-line doc: CALL GAMSDOC F04AZE (or @PRT NAG*DOC.F04AZE) Access: LIB NBS*NAG See also: F01BXE

F04AZF Calculates the approximate solution of a set of real symmetric positive definite linear equations with a single right hand side, AX = b, where A has been decomposed into triangular matrices using F01BXP. Proprietary double precision Fortran program in NAG library. Single precision version is F04AZE. Class(es): D2b1b Usage: CALL F04AZF (N, IR, A, 1A, P, B, IB, IFAIL) On-line doc: CALL GAMSDOC F04AZF (or @PRT NAG*DOC.F04AZF) Access: LIB NBS*NAG See also: F01BXP

F04JAE Least-squares, m real equations in n unknowns, rank<=n, m>=n, minimal least-squares solution. Proprietary single precision Fortran program in NAG library. Double precision version is F04JAF. Class(es): D9 Usage: CALL F04JAE (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) On-line doc: CALL GAMSDOC F04JAE (or @PRT NAG*DOC.F04JAE) Access: LIB NBS*NAG

F04JAF Least-squares, m real equations in n unknowns, rank<=n, m>=n, minimal least-squares solution. Proprietary double precision Fortran program in NAG library. Single precision version is F04JAE. Class(es): D9 Usage: CALL F04JAF (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) On-line doc: CALL GAMSDOC F04JAF (or @PRT NAG*DOC.F04JAF) Access: LIB NBS*NAG

F04JDE Least-squares, m real equations in n unknowns, rank<=n, m>=n, minimal least-squares solution. Proprietary single precision Fortran program in NAG library. Double precision version is F04JDF. Class(es): D9 Usage: CALL F04JDE (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) On-line doc: CALL GAMSDOC F04JDE (or @PRT NAG*DOC.F04JDE) Access: LIB NBS*NAG

F04JDF Least-squares, m real equations in n unknowns, rank<=n, m>=n, minimal least-squares solution. Proprietary double precision Fortran program in NAG library. Single precision version is F04JDE. Class(es): D9 Usage: CALL F04JDF (M, N, A, NRA, B, TOL, SIGMA, IRANK, WORK, LWORK, IFAIL) On-line doc: CALL GAMSDOC F04JDF (or @PRT NAG*DOC.F04JDF) Access: LIB NBS*NAG

F04JGE Least-squares, m real equations in n unknowns, rank<=n, m>=n, least-squares solution if rank=n, otherwise minimal least-squares solution. Proprietary single precision Fortran program in NAG library. Double precision version is F04JGF. Class(es): D9 Usage: CALL F04JGE (M, N, A, NRA, B, TOL, SVD, SIGMA, IRANK, WORK, LWORK, IFAIL) On-line doc: CALL GAMSDOC F04JGE (or @PRT NAG*DOC.F04JGE) Access: LIB NBS*NAG

F04JGF Least-squares, m real equations in n unknowns, rank<=n, m>=n, least-squares solution if rank=n, otherwise minimal least-squares solution. Proprietary double precision Fortran program in NAG library. Single precision version is F04JGE. Class(es): D9 Usage: CALL F04JGF (M, N, A, NRA, B, TOL, SVD, SIGMA, IRANK, WORK, LWORK, IFAIL) On-line doc: CALL GAMSDOC F04JGF (or @PRT NAG*DOC.F04JGF) Access: LIB NBS*NAG

F04LDE Simultaneous linear equations (factorising the matrix of coefficients), real band matrix, approximate solution. Proprietary single precision Fortran program in NAG library. Double precision version is F04LDF. Class(es): D2a2 Usage: CALL F04LDE (N, M1,
F04LDF  Simultaneous linear equations (factorising the matrix of coefficients), real band matrix, approximate solution.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04LDF.  | Class(es): D2a2  | Usage: CALL F04LDF (N, M, M1, M2, IR, A, IA, IL, IN, B, IB, IFAIL)  | On-line doc: CALL GAMSDOC F04LDF (or @PRT NAG+DOC.F04LDF)  | Access: LIB NBS+NAG  | See also: F01LBE

F04MCE  Calculates the approximate solution of a system of real linear equations with multiple right hand sides, Ax = B, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorised by F01MCE. Related systems may also be solved.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F04MCF.  | Class(es): D2b2  | Usage: CALL F04MCE (N, AL, LAL, D, NROW, IR, B, NR, ISELCT, X, NRX, IFAIL)  | On-line doc: CALL GAMSDOC F04MCE (or @PRT NAG+DOC.F04MCE)  | Access: LIB NBS+NAG  | See also: F01MCE

F04MCF  Calculates the approximate solution of a system of real linear equations with multiple right hand sides, Ax = B, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorised by F01MCF. Related systems may also be solved.  | Proprietary double precision Fortran subprogram in NAG library. Single precision version is F04MCF.  | Class(es): D2b2  | Usage: CALL F04MCF (N, AL, LAL, D, NROW, IR, B, NR, ISELCT, X, NRX, IFAIL)  | On-line doc: CALL GAMSDOC F04MCF (or @PRT NAG+DOC.F04MCF)  | Access: LIB NBS+NAG  | See also: F01MCF

F05AFA  Schmidt orthogonalisation of n vectors of order m.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F05AFA.  | Class(es): D5  | Usage: CALL F05AFA (A, IA, M, N1, N2, S, CC, ICOL, IFAIL)  | On-line doc: CALL GAMSDOC F05AFA (or @PRT NAG+DOC.F05AFA)  | Access: LIB NBS+NAG  | See also: F01MCF

F05ABF  Approximate 2-norm of a vector.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is F05ABF.  | Class(es): D1a3b  | Usage: D = F05ABF (X, N)  | On-line doc: CALL GAMSDOC F05ABF (or @PRT NAG+DOC.F05ABF)  | Access: LIB NBS+NAG  | See also: CV

FAC  Factorial, — n!.  | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DFAC.  | Class(es): C1  | Usage: 1 = FAC (N)  | On-line doc: CALL GAMSDOC FAC (or @PRT CMLIB+DOC.FAC)  | Access: LIB NBS+CMLIB

FC  Fits piecewise polynomial to discrete data with equality and inequality constraints.  | Portable single precision Fortran subprogram in FC sublibrary of CMLIB library.  | Class(es): Klalal Kl2a2 L8a3  | Usage: CALL FC(NDATA, XDATA, YDATA, SDATA, NORD, NBKPT, BKP, NCONST, XCONST, YCONST, NDERIV, MODE, COEFF, W, JW)  | On-line doc: CALL GAMSDOC FC (or @PRT CMLIB+DOC.FC)  | Tests: CMLIB+TEST-SOURCE.$F/FC  | Access: LIB NBS+CMLIB  | See also: CV

FCDF  Computes the cumulative distribution function value for the F-distribution with degrees of freedom parameters NU1 and NU2.  | Portable single precision Fortran subprogram in DATAPAC library.  | Class(es): L6a1f  | Usage: CALL FCDF (X, NU1, NU2, CDF)  | On-line doc: CALL GAMSDOC FCDF (or @PRT DATAPAC+DOC.FCDF)  | Access: LIB NBS+DATAPAC

FFT  Compute FFT of complex data sequence (forward or inverse) any number of points. Useful for multivariate transforms. Uses only real arithmetic.  | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFT.  | Class(es): J1a2 J1b  | Usage: CALL FFT (A, B, NTOT, N, NSPAN, ISN)  | On-line doc: CALL GAMSDOC FFT (or @PRT PORT+DOC.FFT)  | Access: LIB NBS+PORT

FFT2C  Computes the fast Fourier transform of a complex valued sequence of length equal to a power two.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): J1a2  | Usage: CALL FFT2C (A, IMWK)  | On-line doc: CALL GAMSDOC FFT2C (or @PRT IMSL+DOC.FFT2C)  | Access: LIB NBS+IMSL

FFFT3D  Compute the fast Fourier transform of a complex valued 1, 2 or 3 dimensional array.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): J1b  | Usage: CALL FFT3D (A, IA1, IA2, N1, N2, N3, JOB, IWK, RWK, CWK)  | On-line doc: CALL GAMSDOC FFT3D (or @PRT IMSL+DOC.FFT3D)  | Access: LIB NBS+IMSL

FFTCC  Compute the fast Fourier transform of a complex valued sequence.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): J1a2  | Usage: CALL FFTCC (A, N, IWK, WK)  | On-line doc: CALL GAMSDOC FFTCC (or @PRT IMSL+DOC.FFTCC)  | Access: LIB NBS+IMSL

FFT1C  Finds the inverse fast Fourier transform, given the Fourier coefficients in the frequency domain.  | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFT1C.  | Class(es): J1a2  | Usage: CALL FFT1C (N, FR, FI)  | On-line doc: CALL GAMSDOC FFT1C (or @PRT PORT+DOC.FFT1C)  | Access: LIB NBS+PORT  | See also: FFTC

FFTR  Mixed radix fast Fourier transform to find the transform of 2N real data points.  | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFTR.  | Class(es): J1a1  | Usage: CALL FFTR (NNP2, A, B)  | On-line doc: CALL GAMSDOC FFTR (or @PRT PORT+DOC.FFTR)  | Access: LIB NBS+PORT  | See also: FFTR
FFTRC Computes the fast Fourier transform of a real valued sequence. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1a | Usage: CALL FFTRC (A,N,IXW,WK) | On-line doc: CALL GAMSDOC FFTRC (or @PRT IMSL*DOC.FFTRC) | Access: LIB NBS*IMSL

FFTRI Finds the inverse Fourier transform using Fourier coefficients assumed to arise from real data in the time domain. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFFTRI. | Class(es): J1a | Usage: CALL FFTRI (N,N,F,FI) | On-line doc: CALL GAMSDOC FFTRI (or @PRT PORT*DOC.FFTRI) | Access: LIB NBS*PORT | See also: FFTR

FFTSC Computes the sine and cosine transforms of a real valued sequence. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): J1a3 | Usage: CALL FFTSC (A,N,ST,CT,IXW,WK,CKW) | On-line doc: CALL GAMSDOC FFTSC (or @PRT IMSL*DOC.FFTSC) | Access: LIB NBS*IMSL

FIG1 Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMPLIB library. | Class(es): D4c1c | Usage: CALL FIG1 (NM,N,T,D,E,E2,IERR) | On-line doc: CALL GAMSDOC FIG1 (or @PRT CMPLIB*DOC.FIG1/EISPACK) | Access: LIB NBS*CMPLIB

FIG12 Transforms certain real non-symmetric tridiagonal matrix to symmetric tridiagonal matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMPLIB library. | Class(es): D4c1c | Usage: CALL FIG12 (NM,N,T,D,E,A1,IERR) | On-line doc: CALL GAMSDOC FIG12 (or @PRT CMPLIB*DOC.FIG12/EISPACK) | Access: LIB NBS*CMPLIB | See also: IMTQL2

FIT Performs linear least squares regression analysis of a general linear model. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a1 | Usage: CALL FIT (Y, XM, N, NP, SCRAT, NS, IXM, RES) | On-line doc: CALL GAMSDOC FIT (or @PRT STATLIB*DOC.FIT) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB

FITS Performs linear least squares regression analysis of a general linear model with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a1 | Usage: CALL FITS (Y, XM, N, NP, SCRAT, NS, IXM, RES, COEF, PV, SDPV, SDRES, VCV, IVCV, NPRT) | On-line doc: CALL GAMSDOC FITS (or @PRT STATLIB*DOC.FITS) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB


FITWS Performs weighted linear least squares regression analysis of a general linear model with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a2 | Usage: CALL FITWS (Y, XM, WT, N, NP, SCRAT, NS, IXM, RES, COEF, PV, SDPV, SDRES, VCV, IVCV, NPRT) | On-line doc: CALL GAMSDOC FITWS (or @PRT STATLIB*DOC.FITWS) | Tests: STATLIB*TEST.DEMO2 | Access: LIB NBS*STATLIB


FLR Finds the largest integer less than or equal to x. Input and output are real. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFLR. | Class(es): C1 | Usage: X = FLR (X) | On-line doc: CALL GAMSDOC FLR (or @PRT PORT*DOC.FLR) | Access: LIB NBS*PORT

FMIN Finds an approximate local minimum of a univariate user defined EXTERNAL function, f. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DFMIN. | Class(es): G1a2 | Usage: X = FMIN (F,X,A,B,T) | On-line doc: CALL GAMSDOC FMIN (or @PRT PORT*DOC.FMIN) | Access: LIB NBS*PORT

FOURI Performed a Fourier analysis of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L10a | Usage: CALL FOURIE(X,N) | On-line doc: CALL GAMSDOC FOURIE (or @PRT DATAPAC*DOC.FOURIE) | Access: LIB NBS*DATAPAC

FRAN Generates a random sample of size N from the F-distribution with degrees of freedom parameters = NU1 and NU2. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a | Usage: CALL FRAN (N,NU1,NU2,START,X) | On-line doc: CALL GAMSDOC FRAN (or @PRT DATAPAC*DOC.FRAN) | Access: LIB NBS*DATAPAC

FREQ Computes the sample frequency and sample cumulative frequency for the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1d | Usage: CALL FREQ(X,N) | On-line doc: CALL GAMSDOC FREQ (or @PRT DATAPAC*DOC.FREQ) | Access: LIB NBS*DATAPAC

FTARPS Preliminary estimation of the autoregressive parameters in an ARIMA stochastic model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L10e1 | Usage: CALL FTARPS (ACV,WBAR,IP,IQ,ARPS,PMAC,W1,IER) | On-line doc: CALL GAMSDOC FTARPS (or @PRT IMSL*DOC.FTARPS) | Access: LIB NBS*IMSL | See also: FTAUTO


FTCAST Time series forecasts and probability limits using an ARIMA (Box-Jenkins) model. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1e2 | Usage: CALL FTCAST (Z,ARPS,PMAS,PMAC,ALPHA,LV,DARPS,FCST,WNV,IER) | On-line doc: CALL GAMSDOC FTCAST (or @PRT IMSL*DOC.FTCAST) | Access: LIB NBS*IMSL

FTCP Non-seasonal ARIMA (Box-Jenkins) stochastic model analysis for a single time series with full parameter iteration and maximum...
likelihood estimation. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L10e | Usage: CALL FTCP (X, IND, DSEEED, ALPHFA, ARPS, PMAS, PMAC, WNY, FCST, SIM, WK, IER) | On-line doc: CALL GAMSDOC FTCP (or @PRT IMSL*DOC. FTCP) | Access: LIB NBS*IMSL

FTCROS Means, variances, cross-covariances, and cross-correlations for two mutually stationary n-channel time series. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L10g1 | Usage: CALL FTCROS (XY, NC, ISW, EMUSIG, ACV, IA, IB, AC, IC, ID, IER) | On-line doc: CALL GAMSDOC FTCROS (or @PRT IMSL*DOC. FTCROS) | Access: LIB NBS*IMSL

FTCRXY Cross-covariance of two mutually stationary time series. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L10g1 | Usage: CALL FTCRXY (X, Y, N, XBAR, YBAR, MLAG, IRS, C, IER) | On-line doc: CALL GAMSDOC FTCRXY (or @PRT IMSL*DOC. FTCRXY) | Access: LIB NBS*IMSL

FTFPS Fast Fourier transform estimates of power spectra and cross spectra of time series. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L10f L10g1 | Usage: CALL FTFPS (X, Y, N, L, IND, PSX, PSY, XPS, IW, WK, WK, CW, IER) | On-line doc: CALL GAMSDOC FTFPS (or @PRT IMSL*DOC. FTFPS) | Access: LIB NBS*IMSL

FTFREQ Single or multichannel time series analysis in the time and frequency domains. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L10f L10c1 | Usage: CALL FTFREQ (X, IND, XIND, XYM, ACV, FREQ, PSX, COV, XSPECT, AMPHAS, XER, COHER, IER) | On-line doc: CALL GAMSDOC FTFREQ (or @PRT IMSL*DOC. FTFREQ) | Access: LIB NBS*IMSL

FTGEN Generation of a time series from a given ARIMA (Box-Jenkins) stochastic model. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L6a20 | Usage: CALL FTGEN (ARPS, PMAS, PMAC, START, WNY, DSEEED, IP, IQ, LW, Wal) | On-line doc: CALL GAMSDOC FTGEN (or @PRT IMSL*DOC. FTGEN) | Access: LIB NBS*IMSL


FTML Maximum likelihood estimation of autoregressive and moving average parameters in an ARIMA (Box-Jenkins) stochastic model. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): L10e1 | Usage: CALL FTML (X, IND, ARPS, PMAS, PMAC, WNY, GR, A, IER) | On-line doc: CALL GAMSDOC FTML (or @PRT IMSL*DOC. FTML) | Access: LIB NBS*IMSL


GAMS: Module Dictionary

G

G01AAE Simple descriptive statistics, one variable, from raw data. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AAF. See also: L5a Usage: CALL G01AAE (N, X, IWT, WT, XMEAN, S2, S3, S4, XMIN, XMAX, WTSM, IFAIL) On-line doc: CALL GAMSDOC G01AAE (or @PRT NAG+DOC.G01AAE) Access: LIB NBS+NAG

G01AAF Simple descriptive statistics, one variable, from raw data. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AAE. See also: L5a Usage: CALL G01AAF (N, X, IWT, WT, XMEAN, S2, S3, S4, XMIN, XMAX, WTSM, IFAIL) On-line doc: CALL GAMSDOC G01AAF (or @PRT NAG+DOC.G01AAF) Access: LIB NBS+NAG

G01ABE Simple descriptive statistics, two variables, from raw data. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01ABF. See also: L5b Usage: CALL G01ABE (N, X1, X2, IWT, WT, RES, IFAIL) On-line doc: CALL GAMSDOC G01ABE (or @PRT NAG+DOC.G01ABE) Access: LIB NBS+NAG

G01ABF Simple descriptive statistics, two variables, from raw data. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01ABE. See also: L5b Usage: CALL G01ABF (N, X1, X2, IWT, WT, RES, IFAIL) On-line doc: CALL GAMSDOC G01ABF (or @PRT NAG+DOC.G01ABF) Access: LIB NBS+NAG

G01AED Simple descriptive statistics, one variable, from frequency table. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AED. See also: L5b Usage: CALL G01AED (K, X, IFREQ, XMEAN, S2, S3, S4, IFAIL) On-line doc: CALL GAMSDOC G01AED (or @PRT NAG+DOC.G01AED) Access: LIB NBS+NAG

G01ADF Simple descriptive statistics, one variable, from frequency table. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01ADE. See also: L5b Usage: CALL G01ADF (K, X, IFREQ, XMEAN, S2, S3, S4, IFAIL) On-line doc: CALL GAMSDOC G01ADF (or @PRT NAG+DOC.G01ADF) Access: LIB NBS+NAG

G01AEE Frequency table from raw data. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AEF. See also: L5b Usage: CALL G01AEE (N, K2, X, ICLASS, CINT, IFREQ, XMIN, XMAX, IFAIL) On-line doc: CALL GAMSDOC G01AEF (or @PRT NAG+DOC.G01AEF) Access: LIB NBS+NAG

G01AEF Frequency table from raw data. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AEE. See also: L5b Usage: CALL G01AEF (N, K2, X, ICLASS, CINT, IFREQ, XMIN, XMAX, IFAIL) On-line doc: CALL GAMSDOC G01AEF (or @PRT NAG+DOC.G01AEF) Access: LIB NBS+NAG

G01AFE Two-way contingency table analysis. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AFF. See also: L5b Usage: CALL G01AFE (INOB, IPRED, M, N, NOBS, NUM, PRED, CHIS, P, NPOS, NDF, M1, N1, IFAIL) On-line doc: CALL GAMSDOC G01AFF (or @PRT NAG+DOC.G01AFF) Access: LIB NBS+NAG

G01AFF Two-way contingency table analysis. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AFE. See also: L5b Usage: CALL G01AFF (INOB, IPRED, M, N, NOBS, NUM, PRED, CHIS, P, NPOS, NDF, M1, N1, IFAIL) On-line doc: CALL GAMSDOC G01AFF (or @PRT NAG+DOC.G01AFF) Access: LIB NBS+NAG

G01AGE Line printer scatter plot of two variables. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AGF. See also: L5a Usage: CALL G01AGE (X, Y, NOBS, ISORT, NSTEPX, NSTEPY, IFAIL) On-line doc: CALL GAMSDOC G01AGE (or @PRT NAG+DOC.G01AGE) Access: LIB NBS+NAG

G01AGF Line printer scatter plot of two variables. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AGE. See also: L5a Usage: CALL G01AGF (X, Y, NOBS, ISORT, NSTEPX, NSTEPY, IFAIL) On-line doc: CALL GAMSDOC G01AGF (or @PRT NAG+DOC.G01AGF) Access: LIB NBS+NAG

G01AHE Line printer scatter plot of one variable against normal scores. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AHF. See also: L5a Usage: CALL G01AHE (X, NOBS, NSTEPX, NSTEPY, IFAIL) On-line doc: CALL GAMSDOC G01AHE (or @PRT NAG+DOC.G01AHE) Access: LIB NBS+NAG See also: G01DAF

G01AHF Line printer scatter plot of one variable against normal scores. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AHE. See also: L5a Usage: CALL G01AHF (X, NOBS, NSTEPX, NSTEPY, IFAIL) On-line doc: CALL GAMSDOC G01AHF (or @PRT NAG+DOC.G01AHF) Access: LIB NBS+NAG See also: G01DAF

G01AJE Prints a histogram on a character printing device, with user control over size, positioning, and the range of data values included. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01AJF. See also: L5a Usage: CALL G01AJE(X,N,NSTEPX,NSTEPY,ITYPE,ISPACE,XMIN,XMAX,NSTEP,N1,MULTY,FAIL) On-line doc: CALL GAMSDOC G01AJE (or @PRT NAG+DOC.G01AJE) Access: LIB NBS+NAG

G01AJF Prints a histogram on a character printing device, with user control over size, positioning, and the range of data values included. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01AJE. See also: L5a Usage: CALL
G01A F (X, N, NSTEPX, NSTEPY, ITYPE, ISPACE, XMIN, XMAX, XSTEP, N1, MULTY, IFAIL) | On-line doc: CALL GAMSDOC G01A F (or @PRT NAG * DOC.G01A F) | Access: LIB NBS* NAG

G01BAE Student’s t distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BAF. | Class(es): L5a1 | Usage: D = G01BAE (IDF, T, IFAIL) | On-line doc: CALL GAMSDOC G01BAE (or @PRT NAG * DOC.G01BAE) | Access: LIB NBS* NAG

G01BAF Student’s t distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BAE. | Class(es): L5a1 | Usage: D = G01BAF (IDF, T, IFAIL) | On-line doc: CALL GAMSDOC G01BAF (or @PRT NAG * DOC.G01BAF) | Access: LIB NBS* NAG

G01BBE F (variance ratio) distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BBF. | Class(es): L5a1f | Usage: D = G01BBE (I1, I2, A, IFAIL) | On-line doc: CALL GAMSDOC G01BBE (or @PRT NAG * DOC.G01BBE) | Access: LIB NBS* NAG

G01BBF F (variance ratio) distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BBE. | Class(es): L5a1f | Usage: D = G01BBF (I1, I2, A, IFAIL) | On-line doc: CALL GAMSDOC G01BBF (or @PRT NAG * DOC.G01BBF) | Access: LIB NBS* NAG

G01BCE Chi-square distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BCF. | Class(es): L5a1c | Usage: D = G01BCE (X, N, IFAIL) | On-line doc: CALL GAMSDOC G01BCE (or @PRT NAG * DOC.G01BCE) | Access: LIB NBS* NAG

G01BCF Chi-square distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BCE. | Class(es): L5a1c | Usage: D = G01BCF (X, N, IFAIL) | On-line doc: CALL GAMSDOC G01BCF (or @PRT NAG * DOC.G01BCF) | Access: LIB NBS* NAG

G01BDE Beta distribution of first kind. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01BDF. | Class(es): L5a1b | Usage: D = G01BDE (X, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01BDE (or @PRT NAG * DOC.G01BDE) | Access: LIB NBS* NAG

G01BDF Beta distribution of first kind. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01BDE. | Class(es): L5a1b | Usage: D = G01BDF (X, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01BDF (or @PRT NAG * DOC.G01BDF) | Access: LIB NBS* NAG

G01CAE Inverse Student’s t distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01CAF. | Class(es): L5a2t | Usage: D = G01CAE (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CAE (or @PRT NAG * DOC.G01CAE) | Access: LIB NBS* NAG

G01CAF Inverse Student’s t distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01CAE. | Class(es): L5a2t | Usage: D = G01CAF (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CAF (or @PRT NAG * DOC.G01CAF) | Access: LIB NBS* NAG

G01CBE Inverse F (variance ratio) distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01CBF. | Class(es): L5a2f | Usage: D = G01CBE (P, M, N, IFAIL) | On-line doc: CALL GAMSDOC G01CBE (or @PRT NAG * DOC.G01CBE) | Access: LIB NBS* NAG

G01CBF Inverse F (variance ratio) distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01CBE. | Class(es): L5a2f | Usage: D = G01CBF (P, M, N, IFAIL) | On-line doc: CALL GAMSDOC G01CBF (or @PRT NAG * DOC.G01CBF) | Access: LIB NBS* NAG

G01CCE Inverse chi-square distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01CCF. | Class(es): L5a2c | Usage: D = G01CCE (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CCE (or @PRT NAG * DOC.G01CCE) | Access: LIB NBS* NAG

G01CCF Inverse chi-square distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01CCE. | Class(es): L5a2c | Usage: D = G01CCF (P, N, IFAIL) | On-line doc: CALL GAMSDOC G01CCF (or @PRT NAG * DOC.G01CCF) | Access: LIB NBS* NAG

G01CDE Inverse beta distribution of first kind. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01CDF. | Class(es): L5a2b | Usage: D = G01CDE (P, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01CDE (or @PRT NAG * DOC.G01CDE) | Access: LIB NBS* NAG

G01CDF Inverse beta distribution of first kind. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01CDE. | Class(es): L5a2b | Usage: D = G01CDF (P, A, B, IFAIL) | On-line doc: CALL GAMSDOC G01CDF (or @PRT NAG * DOC.G01CDF) | Access: LIB NBS* NAG

G01CEE Inverse normal distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G01CEF. | Class(es): L5a2n | Usage: D = G01CEE (P, IFAIL) | On-line doc: CALL GAMSDOC G01CEE (or @PRT NAG * DOC.G01CEE) | Access: LIB NBS* NAG

G01CEF Inverse normal distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G01CEE.
January 1984

GAMS: Module Dictionary

C 87

| Class(es): | L5a2n | Usage: D = G01CEF (P, IFAIL) | On-line doc: CALL GAMSDOC G01CEF (or @PRT NAG+DOC.G01CEF) | Access: LIB NBS+NAG |

G01DAE Calculation of normal scores. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G01DAF. | Class(es): L5a2n | Usage: CALL G01DAE (N, PP, ETOL, ERREST, WORK, IW, IFAIL) | On-line doc: CALL GAMSDOC G01DAE (or @PRT NAG+DOC.G01DAE) | Access: LIB NBS+NAG | See also: G01AHF |

G01DAF Calculation of normal scores. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is G01DAE. | Class(es): L5a2n | Usage: CALL G01DAF (N, PP, ETOL, ERREST, WORK, IW, IFAIL) | On-line doc: CALL GAMSDOC G01DAF (or @PRT NAG+DOC.G01DAF) | Access: LIB NBS+NAG |

G02BAE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BAF. | Class(es): L1e1 | Usage: CALL G02BAE (N, M, X, IX, XBAR, STD, SSP, SSP, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BAE (or @PRT NAG+DOC.G02BAE) | Access: LIB NBS+NAG |

G02BAP Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is G02BAE. | Class(es): L1e1 | Usage: CALL G02BAP (N, M, X, IX, XBAR, STD, SSP, SSP, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BAP (or @PRT NAG+DOC.G02BAP) | Access: LIB NBS+NAG |

G02BCE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BCF. | Class(es): L1e2 | Usage: CALL G02BCE (N, M, X, IX, MISS, XMISS, XBAR, STD, SSP, SSP, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BCE (or @PRT NAG+DOC.G02BCE) | Access: LIB NBS+NAG |

G02BCF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in the array X containing missing values. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is G02BCE. | Class(es): L1e2 | Usage: CALL G02BCF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSP, SSP, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BCF (or @PRT NAG+DOC.G02BCF) | Access: LIB NBS+NAG |

G02BDE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BDF. | Class(es): L1e1 | Usage: CALL G02BDE (N, M, X, IX, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BDE (or @PRT NAG+DOC.G02BDE) | Access: LIB NBS+NAG |

G02BDF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is G02BDE. | Class(es): L1e1 | Usage: CALL G02BDF (N, M, X, IX, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BDF (or @PRT NAG+DOC.G02BDF) | Access: LIB NBS+NAG |

G02BEE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BEF. | Class(es): L1e2 | Usage: CALL G02BEE (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BEE (or @PRT NAG+DOC.G02BEE) | Access: LIB NBS+NAG |

G02BEF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is G02BEF. | Class(es): L1e2 | Usage: CALL G02BEF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BEF (or @PRT NAG+DOC.G02BEF) | Access: LIB NBS+NAG |

G02BFF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in the array X. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is G02BFF. | Class(es): L1e2 | Usage: CALL G02BFF (N, M, X, IX, MISS, XMISS, XBAR, STD, SSPZ, ISSPZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BFF (or @PRT NAG+DOC.G02BFF) | Access: LIB NBS+NAG |
GAMS: Module Dictionary

January 1984

IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BFF (or @PRT NAG+DOC.G02BFF) | Access: LIB NBS+NAG

G02BGE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BGF. | Class(es): Lie1 | Usage: CALL G02BGE (N, M, X, IX, NVARS, KVAR, XBAR, STD, SSP, MISS, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BGE (or @PRT NAG+DOC.G02BGE) | Access: LIB NBS+NAG

G02BGF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BGE. | Class(es): Lie1 | Usage: CALL G02BGF (N, M, X, IX, NVARS, KVAR, XBAR, STD, SSP, MISS, R, IR, IFAIL) | On-line doc: CALL GAMSDOC G02BGF (or @PRT NAG+DOC.G02BGF) | Access: LIB NBS+NAG

G02BHE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BHE. | Class(es): Lie2 | Usage: CALL G02BHE (N, M, X, IX, MISS, XMISTYP, NVARS, XBAR, STD, SSP, MISS, R, IR, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BHE (or @PRT NAG+DOC.G02BHE) | Access: LIB NBS+NAG

G02BHF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BHE. | Class(es): Lie2 | Usage: CALL G02BHF (N, M, X, IX, MISS, XMISTYP, NVARS, XBAR, STD, SSP, MISS, R, IR, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BHF (or @PRT NAG+DOC.G02BHF) | Access: LIB NBS+NAG

G02BJE Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BJF. | Class(es): Lie2 | Usage: CALL G02BJE (N, M, X, IX, MISS, XMISTYP, NVARS, XBAR, STD, SSP, MISS, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BJE (or @PRT NAG+DOC.G02BJE) | Access: LIB NBS+NAG

G02BJF Computes means and standard deviations of variables, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BJE. | Class(es): Lie2 | Usage: CALL G02BJF (N, M, X, IX, MISS, XMISTYP, NVARS, XBAR, STD, SSP, MISS, R, IR, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BJF (or @PRT NAG+DOC.G02BJF) | Access: LIB NBS+NAG

G02BKE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BKF. | Class(es): Lie1 | Usage: CALL G02BKE (N, M, X, IX, NVARS, KVAR, XBAR, STD, SSPZ, MISSZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BKE (or @PRT NAG+DOC.G02BKE) | Access: LIB NBS+NAG

G02BKF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BKE. | Class(es): Lie1 | Usage: CALL G02BKF (N, M, X, IX, NVARS, KVAR, XBAR, STD, SSPZ, MISSZ, RZ, IRZ, IFAIL) | On-line doc: CALL GAMSDOC G02BKF (or @PRT NAG+DOC.G02BKF) | Access: LIB NBS+NAG

G02BLE Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BLF. | Class(es): Lie2 | Usage: CALL G02BLE (N, M, X, IX, MISS, XMISTYP, NVARS, KVAR, XBAR, STD, SSPZ, MISSZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BLE (or @PRT NAG+DOC.G02BLE) | Access: LIB NBS+NAG

G02BLF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BLF. | Class(es): Lie2 | Usage: CALL G02BLF (N, M, X, IX, MISS, XMISTYP, NVARS, KVAR, XBAR, STD, SSPZ, MISSZ, RZ, IRZ, NCASES, IFAIL) | On-line doc: CALL GAMSDOC G02BLF (or @PRT NAG+DOC.G02BLF) | Access: LIB NBS+NAG

G02BME Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BMF. | Class(es): Lie2 | Usage: CALL G02BME (N, M, X, IX, MISS, XMISTYP, NVARS, KVAR, XBAR, STD, SSPZ, MISSZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BME (or @PRT NAG+DOC.G02BME) | Access: LIB NBS+NAG

G02BMF Computes means and standard deviations of variables, sums of squares and cross-products about zero, and correlation-like coefficients for a set of data in specified columns of the array X containing missing values. | Proprietary single precision Fortran subroutine in NAG library. Double precision version is G02BMF. | Class(es): Lie2 | Usage: CALL G02BMF (N, M, X, IX, MISS, XMISTYP, NVARS, KVAR, XBAR, STD, SSPZ, MISSZ, RZ, IRZ, NCASES, COUNT, IC, IFAIL) | On-line doc: CALL GAMSDOC G02BMF (or @PRT NAG+DOC.G02BMF) | Access: LIB NBS+NAG
**G02BNE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, overwriting X with the ranks of the observations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BNF.

**Class(es):** L1e1b | Usage: CALL G02BNE (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BNE (or @PRT NAG+DOCG02BNE) | Access: LIB NBS+NAG

**G02BNF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, overwriting X with the ranks of the observations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BNE.

**Class(es):** L1e1b | Usage: CALL G02BNF (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BNF (or @PRT NAG+DOCG02BNF) | Access: LIB NBS+NAG

**G02BPE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, overwriting X with the ranks of the observations. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BPF.

**Class(es):** L1e1b | Usage: CALL G02BPE (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BPE (or @PRT NAG+DOCG02BPE) | Access: LIB NBS+NAG

**G02BPF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, overwriting X with the ranks of the observations. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BPE.

**Class(es):** L1e1b | Usage: CALL G02BPF (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BPF (or @PRT NAG+DOCG02BPF) | Access: LIB NBS+NAG

**G02BQE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, preserving X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BQF.

**Class(es):** L1e1b | Usage: CALL G02BQE (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BQE (or @PRT NAG+DOCG02BQE) | Access: LIB NBS+NAG

**G02BQF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X, preserving X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BQE.

**Class(es):** L1e1b | Usage: CALL G02BQF (N, M, X, IX, ITYPE, RR, IRR, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BQF (or @PRT NAG+DOCG02BQF) | Access: LIB NBS+NAG

**G02BRE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BRF.

**Class(es):** L1e1b | Usage: CALL G02BRE (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BRE (or @PRT NAG+DOCG02BRE) | Access: LIB NBS+NAG

**G02BRF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BRE.

**Class(es):** L1e1b | Usage: CALL G02BRF (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, INCASE, KWORKA, KWORKB, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BRF (or @PRT NAG+DOCG02BRF) | Access: LIB NBS+NAG

**G02BSE** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02BSF.

**Class(es):** L1e1b | Usage: CALL G02BSE (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, COUNT, IC, KWORKA KWORKB, KWORKD, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BSE (or @PRT NAG+DOCG02BSF) | Access: LIB NBS+NAG

**G02BSF** Computes Kendall and/or Spearman non-parametric rank correlation coefficients for a set of data in the array X containing missing values, preserving X. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02BSE.

**Class(es):** L1e1b | Usage: CALL G02BSF (N, M, X, IX, MISS, XMISS, ITYPE, RR, IRR, NCASES, COUNT, IC, KWORKA KWORKB, KWORKD, WORK1, WORK2, IFAIL) | On-line doc: CALL GAMSDOC G02BSF (or @PRT NAG+DOCG02BSF) | Access: LIB NBS+NAG

**G02CAE** Simple linear regression with constant term, no missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CAF.

**Class(es):** L8a1a | Usage: CALL G02CAE (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CAE (or @PRT NAG+DOCG02CAE) | Access: LIB NBS+NAG

**G02CAF** Simple linear regression with constant term, no missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CAE.

**Class(es):** L8a1a | Usage: CALL G02CAF (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CAF (or @PRT NAG+DOCG02CAF) | Access: LIB NBS+NAG

**G02CBE** Simple linear regression without constant term, no missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CBF.

**Class(es):** L8a1b | Usage: CALL G02CBE (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CBE (or @PRT NAG+DOCG02CBE) | Access: LIB NBS+NAG

**G02CBF** Simple linear regression without constant term, no missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CBE.

**Class(es):** L8a1b | Usage: CALL G02CBF (N, X, Y, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CBF (or @PRT NAG+DOCG02CBF) | Access: LIB NBS+NAG

**G02CCE** Simple linear regression with constant term, missing values. | Proprietary single precision Fortran subprogram in NAG library.
Double precision version is G02CCF. | Class(es): L.8a11b | Usage: CALL G02CCE (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CCE (or @PRT NAG+DOC.G02CCE) | Access: LIB NBS+NAG

G02CCF Simple linear regression with constant term, missing values. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CEE. | Class(es): L.8a11b | Usage: CALL G02CCF (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CCF (or @PRT NAG+DOC.G02CCF) | Access: LIB NBS+NAG

G02CDE Simple linear regression without constant term, missing values. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CFD. | Class(es): L.8a1b | Usage: CALL G02CDE (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CDE (or @PRT NAG+DOC.G02CDE) | Access: LIB NBS+NAG

G02CFF Service routines for multiple linear regression, select elements from vectors and matrices. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CFE. | Class(es): L.8b | Usage: CALL G02CFF (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CFF (or @PRT NAG+DOC.G02CFF) | Access: LIB NBS+NAG

G02CEF Service routines for multiple linear regression, select elements from vectors and matrices. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CFE. | Class(es): L.8b | Usage: CALL G02CEF (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CEF (or @PRT NAG+DOC.G02CEF) | Access: LIB NBS+NAG

G02CFE Service routines for multiple linear regression, re-order elements of vectors and matrices. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CFE. | Class(es): L.8b | Usage: CALL G02CFE (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CFE (or @PRT NAG+DOC.G02CFE) | Access: LIB NBS+NAG

G02CGF Performs a multiple linear regression on the set of variables whose means, sums of squares and cross-products of deviations from means, and Pearson product-moment correlation coefficients are given. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CGF. | Class(es): L.8a1c | Usage: CALL G02CGF (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CGF (or @PRT NAG+DOC.G02CGF) | Access: LIB NBS+NAG

G02CHE Performs a multiple linear regression with no constant on the set of variables whose sums of squares and cross-products about zero and correlation-like coefficients are given. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CHF. | Class(es): L.8a1c | Usage: CALL G02CHE (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CHE (or @PRT NAG+DOC.G02CHE) | Access: LIB NBS+NAG

G02CHF Performs a multiple linear regression with no constant on the set of variables whose sums of squares and cross-products about zero and correlation-like coefficients are given. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CHE. | Class(es): L.8a1c | Usage: CALL G02CHF (N, X, Y, XMIX, YMISS, RESULT, IFAIL) | On-line doc: CALL GAMSDOC G02CHF (or @PRT NAG+DOC.G02CHF) | Access: LIB NBS+NAG

G02CJE Performs one or more multiple linear regressions, regressing each of a set of dependent variables separately on the same set of independent variables. Input is raw data; output includes, for each dependent variable, estimates of regression coefficients and an estimate of the variance of residuals. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G02CJF. | Class(es): L.8a7 | Usage: CALL G02CJE (X, Y, X, Y, N, M, IR, THETA, IT, SIGSQ, C, IC, IPIV, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC G02CJE (or @PRT NAG+DOC.G02CJE) | Access: LIB NBS+NAG

G02CJF Performs one or more multiple linear regressions, regressing each of a set of dependent variables separately on the same set of independent variables. Input is raw data; output includes, for each dependent variable, estimates of regression coefficients and an estimate of the variance of residuals. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G02CJF. | Class(es): L.8a7 | Usage: CALL G02CJF (X, Y, X, Y, N, M, IR, THETA, IT, SIGSQ, C, IC, IPIV, WK1, WK2, IFAIL) | On-line doc: CALL GAMSDOC G02CJF (or @PRT NAG+DOC.G02CJF) | Access: LIB NBS+NAG
G04AD
Three-way analysis of variance, Latin square design. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G04ADP. | Class(es): L7a2b1a | Usage: CALL G04ADE (DATA, VAR, AMR, AMC, AMT, LCODE, IA, N, NN) | On-line doc: CALL GAMSDOC G04ADE (or @PRT NAG+DOC.G04ADE) | Access: LIB NBS+NAG

G04ADF
Three-way analysis of variance, Latin square design. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G04ADE. | Class(es): L7a2b1a | Usage: CALL G04ADF (DATA, VAR, AMR, AMC, AMT, LCODE, IA, N, NN) | On-line doc: CALL GAMSDOC G04ADF (or @PRT NAG+DOC.G04ADF) | Access: LIB NBS+NAG

G04AEE

G04AEF

G04AFE
Two-way analysis of variance, cross-classification, subgroups of equal size. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G04AFF. | Class(es): L7a2b1a | Usage: CALL G04AFE (Y, Y1, Y2, M, NR, NG, ROW, COL, CELL, ICELL, GM, SS, IDF, F, FF, IFAIL) | On-line doc: CALL GAMSDOC G04AFF (or @PRT NAG+DOC.G04AFF) | Access: LIB NBS+NAG

G04AFP
Two-way analysis of variance, cross-classification, subgroups of equal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G04AFP. | Class(es): L7a2b1a | Usage: CALL G04AFP (Y, Y1, Y2, M, NR, NG, ROW, COL, CELL, ICELL, GM, SS, IDF, F, FF, IFAIL) | On-line doc: CALL GAMSDOC G04AFP (or @PRT NAG+DOC.G04AFP) | Access: LIB NBS+NAG

G04AGE

G04AGF

G05CAE
Pseudo-random real numbers, uniform distribution over (0,0,1). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CAF. | Class(es): L6a21 | Usage: D = G05CAE (X) | On-line doc: CALL GAMSDOC G05CAE (or @PRT NAG+DOC.G05CAE) | Access: LIB NBS+NAG

G05CAF
Pseudo-random real numbers, uniform distribution over (0,0,1). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CAF. | Class(es): L6a21 | Usage: D = G05CAF (X) | On-line doc: CALL GAMSDOC G05CAF (or @PRT NAG+DOC.G05CAF) | Access: LIB NBS+NAG

G05CBE
Initialise random number generating routines, to give a repeatable sequence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CBF. | Class(es): L6c | Usage: CALL G05CBE (I) | On-line doc: CALL GAMSDOC G05CBE (or @PRT NAG+DOC.G05CBE) | Access: LIB NBS+NAG

G05CBF
Initialise random number generating routines, to give a repeatable sequence. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CBF. | Class(es): L6c | Usage: CALL G05CBF (I) | On-line doc: CALL GAMSDOC G05CBF (or @PRT NAG+DOC.G05CBF) | Access: LIB NBS+NAG

G05CCE
Initialise random number generating routines, to give nonrepeatable sequence. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CCF. | Class(es): L6c | Usage: CALL G05CCE | On-line doc: CALL GAMSDOC G05CCE (or @PRT NAG+DOC.G05CCE) | Access: LIB NBS+NAG

G05CCF
Initialise random number generating routines, to give nonrepeatable sequence. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CCF. | Class(es): L6c | Usage: CALL G05CCF | On-line doc: CALL GAMSDOC G05CCF (or @PRT NAG+DOC.G05CCF) | Access: LIB NBS+NAG

G05CFF
Save state of random number generating routines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CFF. | Class(es): L6c | Usage: CALL G05CFF (IA, NI, XI, XX, IFAIL) | On-line doc: CALL GAMSDOC G05CFF (or @PRT NAG+DOC.G05CFF) | Access: LIB NBS+NAG

G05CGE
Save state of random number generating routines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CGE. | Class(es): L6c | Usage: CALL G05CGE (IA, NI, XI, XX, IFAIL) | On-line doc: CALL GAMSDOC G05CGE (or @PRT NAG+DOC.G05CGE) | Access: LIB NBS+NAG

G05CGF
Restore state of random number generating routines. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05CGF. | Class(es): L6c | Usage: CALL G05CGF (IA, NI, XI, XX, IFAIL) | On-line doc: CALL GAMSDOC
G05CGE (or PRT NAG+DOC.G05CGE) | Access: LIB NBS+NAG

G05CGF Restore state of random number generating routines. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05CGE. | Class(es): L6c | Usage: CALL G05CGF (IA, NI, XA, NX, IFAIL) | On-line doc: CALL GAMSDOC G05CGF (or PRT NAG+DOC.G05CGF) | Access: LIB NBS+NAG

G05DAE Pseudo-random real numbers, uniform distribution over (a,b). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DAF. | Class(es): L6a21 | Usage: D = G05DAE (A, B) | On-line doc: CALL GAMSDOC G05DAE (or PRT NAG+DOC.G05DAE) | Access: LIB NBS+NAG

G05DAF Pseudo-random real numbers, uniform distribution over (a,b). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DAE. | Class(es): L6a21 | Usage: D = G05DAF (A, B) | On-line doc: CALL GAMSDOC G05DAF (or PRT NAG+DOC.G05DAF) | Access: LIB NBS+NAG

G05DBE Pseudo-random real numbers, exponential distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DBF. | Class(es): L6a5 | Usage: D = G05DBE (A) | On-line doc: CALL GAMSDOC G05DBE (or PRT NAG+DOC.G05DBE) | Access: LIB NBS+NAG

G05DBF Pseudo-random real numbers, exponential distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DBE. | Class(es): L6a5 | Usage: D = G05DBF (A) | On-line doc: CALL GAMSDOC G05DBF (or PRT NAG+DOC.G05DBF) | Access: LIB NBS+NAG

G05DCE Pseudo-random real numbers, logistic distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DCF. | Class(es): L6a12 | Usage: D = G05DCE (A, B) | On-line doc: CALL GAMSDOC G05DCE (or PRT NAG+DOC.G05DCE) | Access: LIB NBS+NAG

G05DCF Pseudo-random real numbers, logistic distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DCE. | Class(es): L6a12 | Usage: D = G05DCF (A, B) | On-line doc: CALL GAMSDOC G05DCF (or PRT NAG+DOC.G05DCF) | Access: LIB NBS+NAG

G05DDE Pseudo-random real numbers, normal distribution (a,b). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DDF. | Class(es): L6a14 | Usage: D = G05DDE (A, B) | On-line doc: CALL GAMSDOC G05DDE (or PRT NAG+DOC.G05DDE) | Access: LIB NBS+NAG

G05DDF Pseudo-random real numbers, normal distribution (a,b). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DDE. | Class(es): L6a14 | Usage: D = G05DDF (A, B) | On-line doc: CALL GAMSDOC G05DDF (or PRT NAG+DOC.G05DDF) | Access: LIB NBS+NAG

G05DEE Pseudo-random real numbers, lognormal distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DEF. | Class(es): L6a12 | Usage: D = G05DEE (A, B) | On-line doc: CALL GAMSDOC G05DEE (or PRT NAG+DOC.G05DEE) | Access: LIB NBS+NAG


G05DFE Pseudo-random real numbers, Cauchy distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DFF. | Class(es): L6a3 | Usage: D = G05DFE (A, B) | On-line doc: CALL GAMSDOC G05DFE (or PRT NAG+DOC.G05DFE) | Access: LIB NBS+NAG

G05DFF Pseudo-random real numbers, Cauchy distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DFE. | Class(es): L6a3 | Usage: D = G05DFF (A, B) | On-line doc: CALL GAMSDOC G05DFF (or PRT NAG+DOC.G05DFF) | Access: LIB NBS+NAG

G05DGE Pseudo-random real numbers, gamma distribution with parameters (g,h). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DGF. | Class(es): L6a7 | Usage: D = G05DGE (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DGE (or PRT NAG+DOC.G05DGE) | Access: LIB NBS+NAG

G05DGF Pseudo-random real numbers, gamma distribution with parameters (g,h). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DGE. | Class(es): L6a7 | Usage: D = G05DGF (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DGF (or PRT NAG+DOC.G05DGF) | Access: LIB NBS+NAG

G05DHE Pseudo-random real numbers, chi-square distribution. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DHF. | Class(es): L6a3 | Usage: D = G05DHE (N, IFAIL) | On-line doc: CALL GAMSDOC G05DHE (or PRT NAG+DOC.G05DHE) | Access: LIB NBS+NAG

G05DHF Pseudo-random real numbers, chi-square distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05DHE. | Class(es): L6a3 | Usage: D = G05DHF (N, IFAIL) | On-line doc: CALL GAMSDOC G05DHF (or PRT
G05DJF  Pseudo-random real numbers, Student's $t$ distribution.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05DFJ. | Class(es): L6a20 | Usage: D = G05DJF (N, IFAIL) | On-line doc: CALL GAMSDOC G05DJF (or @PRT NAG+DOC.G05DJF) | Access: LIB NBS+NAG

G05DKE  Pseudo-random real numbers, Snedecor's F-distribution.  | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05DKF. | Class(es): L6a6 | Usage: D = G05DKE (M, N, IFAIL) | On-line doc: CALL GAMSDOC G05DKE (or @PRT NAG+DOC.G05DKE) | Access: LIB NBS+NAG

G05DLF  Pseudo-random real numbers, Beta distribution of the first kind.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05DLF. | Class(es): L6a2 | Usage: D = G05DLF (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DLF (or @PRT NAG+DOC.G05DLF) | Access: LIB NBS+NAG

G05DMF  Pseudo-random real numbers, Beta distribution of the second kind.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05DMF. | Class(es): L6a2 | Usage: D = G05DMF (G, H, IFAIL) | On-line doc: CALL GAMSDOC G05DMF (or @PRT NAG+DOC.G05DMF) | Access: LIB NBS+NAG

G05DZF  Pseudo-random integer from uniform distribution.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05DFZ. | Class(es): L6a21 L6a9 | Usage: I = G05DZF (M, N) | On-line doc: CALL GAMSDOC G05DZF (or @PRT NAG+DOC.G05DZF) | Access: LIB NBS+NAG

G05DYF  Pseudo-random integer from uniform distribution.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05DFY. | Class(es): L6a21 L6a9 | Usage: I = G05DFY (M, N) | On-line doc: CALL GAMSDOC G05DFY (or @PRT NAG+DOC.G05DFY) | Access: LIB NBS+NAG

G05EYE  Call reference vector for generating pseudo-random integers, uniform distribution.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05EYE. | Class(es): L6a21 L6a9 | Usage: CALL G05EYE (M, N, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EYE (or @PRT NAG+DOC.G05EYE) | Access: LIB NBS+NAG

G05EYF  Call reference vector for generating pseudo-random integers, uniform distribution.  | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05EYF. | Class(es): L6a21 L6a9 | Usage: CALL G05EYF (M, N, R, NR, IFAIL) | On-line doc: CALL GAMSDOC G05EYF (or @PRT NAG+DOC.G05EYF) | Access: LIB NBS+NAG
G05ECF  Set up reference vector for generating pseudo-random integers, Poisson distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05ECE. | Class(es): L6a16  | Usage: CALL G05ECF (T, R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05ECF (or @PRT NAG+DOC.G05ECF)  | Access: LIB NBS+NAG  | See also: G05EYF

G05EDF  Set up reference vector for generating pseudo-random integers, binomial distribution. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EDE. | Class(es): L6a2  | Usage: CALL G05EDF (N, P, R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05EDE (or @PRT NAG+DOC.G05EDE)  | Access: LIB NBS+NAG  | See also: G05EYF

G05EEF  Set up reference vector for generating pseudo-random integers, hypergeometric distribution. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05EFF. | Class(es): L6a14  | Usage: CALL G05EEF (N, P, R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05EEF (or @PRT NAG+DOC.G05EEF)  | Access: LIB NBS+NAG  | See also: G05EYF

G05EHE  Performs a pseudo-random permutation of a vector of integers. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05EHE. | Class(es): L6a16  | Usage: CALL G05EHE(INDEX,N,IFAIL)  | On-line doc: CALL GAMSDOC G05EHE (or @PRT NAG+DOC.G05EHE)  | Access: LIB NBS+NAG

G05EHW  Performs a pseudo-random permutation of a vector of integers. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05EHW. | Class(es): L6a10  | Usage: CALL G05EHW(INDEX,N,IFAIL)  | On-line doc: CALL GAMSDOC G05EHW (or @PRT NAG+DOC.G05EHW)  | Access: LIB NBS+NAG

G05EJE  Selects a pseudo-random sample from an integer vector. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EJF. | Class(es): L6a10  | Usage: CALL G05EJE(IA,N,IB,M,IFAIL)  | On-line doc: CALL GAMSDOC G05EJE (or @PRT NAG+DOC.G05EJE)  | Access: LIB NBS+NAG

G05EJF  Selects a pseudo-random sample from an integer vector. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is G05EJF. | Class(es): L6a10  | Usage: CALL G05EJF(IA,N,IB,M,IFAIL)  | On-line doc: CALL GAMSDOC G05EJF (or @PRT NAG+DOC.G05EJF)  | Access: LIB NBS+NAG

G05EWE  Generate next term from ARMA time series using vector from G05EGE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EWF. | Class(es): L6a20  | Usage: D = G05EWE (R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05EWE (or @PRT NAG+DOC.G05EWE)  | Access: LIB NBS+NAG  | See also: G05EGE

G05EWF  Generate next term from ARMA time series using vector from G05EWF. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EWE. | Class(es): L6a20  | Usage: D = G05EWF (R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05EWF (or @PRT NAG+DOC.G05EWF)  | Access: LIB NBS+NAG  | See also: G05EWE

G05EXE  Set up reference vector from supplied cumulative distribution function or probability distribution function. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EXF. | Class(es): L6a7  | Usage: CALL G05EXE (P, NP, IP, LP, R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05EXE (or @PRT NAG+DOC.G05EXE)  | Access: LIB NBS+NAG  | See also: G05EYF

G05EXP  Set up reference vector from supplied cumulative distribution function or probability distribution function. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EXP. | Class(es): L6a7  | Usage: CALL G05EXP (P, NP, IP, LP, R, NR, IFAIL)  | On-line doc: CALL GAMSDOC G05EXP (or @PRT NAG+DOC.G05EXP)  | Access: LIB NBS+NAG  | See also: G05EYF
G05EYE  Pseudo-random integer from reference vector.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EYF.  |  Class(es): L6a9 |  Usage: I = G05EYE (R, NR) |  On-line doc: CALL GAMSDOC G05EYE (or ®PRT NAG+DOC.G05EYE) |  Access: LIB NBS+NAG |  See also: G05EAE G05ECE G05EDE G05EE G05EFE G05EXE

G05EYF  Pseudo-random integer from reference vector.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EYE.  |  Class(es): L6a9 |  Usage: I = G05EYF (R, NR) |  On-line doc: CALL GAMSDOC G05EYF (or ®PRT NAG+DOC.G05EYF) |  Access: LIB NBS+NAG |  See also: G05EAE G05ECE G05EDE G05EE G05EFE G05EXF

G05EZE  Returns a pseudo-random multivariate normal vector taken from a distribution described by a reference vector set up by G05EAE.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G05EZF.  |  Class(es): L6b14 |  Usage: CALL G05EZE(Z,N,R,NR,IFAIL) |  On-line doc: CALL GAMSDOC G05EZE (or ®PRT NAG+DOC.G05EZE) |  Access: LIB NBS+NAG |  See also: G05EAE

G05EZF  Returns a pseudo-random multivariate normal vector taken from a distribution described by a reference vector set up by G05EAF.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is G05EZE.  |  Class(es): L6b14 |  Usage: CALL G05EZF(Z,N,R,NR,IFAIL) |  On-line doc: CALL GAMSDOC G05EZF (or ®PRT NAG+DOC.G05EZF) |  Access: LIB NBS+NAG |  See also: G05EAE

G08AAE  Sign test on two paired samples.  |  Proprietary single precision Fortran subprogram in NAG library. Single precision version is G08AAF.  |  Class(es): L4b1b |  Usage: CALL G08AAE (X, Y, N, IS, N1, P, IFAIL) |  On-line doc: CALL GAMSDOC G08AAE (or ®PRT NAG+DOC.G08AAE) |  Access: LIB NBS+NAG

G08AAF  Sign test on two paired samples.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08AAE.  |  Class(es): L4b1b |  Usage: CALL G08AAF (X, Y, N, IS, N1, P, IFAIL) |  On-line doc: CALL GAMSDOC G08AAF (or ®PRT NAG+DOC.G08AAF) |  Access: LIB NBS+NAG

G08ABE  Wilcoxon matched pairs signed ranks test on two paired samples.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08ABF.  |  Class(es): L4b1b |  Usage: CALL G08ABE (X, Y, N, W1, W2, W, N1, P, IFAIL) |  On-line doc: CALL GAMSDOC G08ABE (or ®PRT NAG+DOC.G08ABE) |  Access: LIB NBS+NAG

G08ABF  Wilcoxon matched pairs signed ranks test on two paired samples.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08ABE.  |  Class(es): L4b1b |  Usage: CALL G08ABF (X, Y, N, W1, W2, W, N1, P, IFAIL) |  On-line doc: CALL GAMSDOC G08ABF (or ®PRT NAG+DOC.G08ABF) |  Access: LIB NBS+NAG

G08ACE  Median test on two samples of unequal size.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08ACF.  |  Class(es): L4b1b |  Usage: CALL G08ACE (X, N, N1, W, 11, 12, P, IFAIL) |  On-line doc: CALL GAMSDOC G08ACE (or ®PRT NAG+DOC.G08ACE) |  Access: LIB NBS+NAG

G08ACF  Median test on two samples of unequal size.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08ACE.  |  Class(es): L4b1b |  Usage: CALL G08ACF (X, N, N1, W, 11, 12, P, IFAIL) |  On-line doc: CALL GAMSDOC G08ACF (or ®PRT NAG+DOC.G08ACF) |  Access: LIB NBS+NAG

G08ADE  Mann-Whitney U-test on two samples of unequal size.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08ADF.  |  Class(es): L4b1b |  Usage: CALL G08ADE (X, N, N1, W, U, P, IFAIL) |  On-line doc: CALL GAMSDOC G08ADE (or ®PRT NAG+DOC.G08ADE) |  Access: LIB NBS+NAG


G08AEE  Friedman 2-way analysis of variance on k matched samples.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08AEF.  |  Class(es): L7a2a2 |  Usage: CALL G08AEE (X, IX, K, N, W1, W2, FR, P, IFAIL) |  On-line doc: CALL GAMSDOC G08AEE (or ®PRT NAG+DOC.G08AEE) |  Access: LIB NBS+NAG

G08AEF  Friedman 2-way analysis of variance on k matched samples.  |  Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08AEE.  |  Class(es): L7a2a2 |  Usage: CALL G08AEF (X, IX, K, N, W1, W2, FR, P, IFAIL) |  On-line doc: CALL GAMSDOC G08AEF (or ®PRT NAG+DOC.G08AEF) |  Access: LIB NBS+NAG


G08BAAE  Mood's and David's tests on two samples of unequal size.  |  Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08BAF.  |  Class(es): L4b1b |  Usage: CALL G08BAAE (X, N, N1, R, ITEST, W, V, FW, PV, IFAIL) |  On-line doc: CALL GAMSDOC G08BAAE (or ®PRT NAG+DOC.G08BAAE) |  Access: LIB NBS+NAG

January 1984
GAMS: Module Dictionary
C 95
G08BAF Mood's and David's tests on two samples of unequal size. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08BAE. | Class(es): L4lb1 | Usage: CALL G08BAF (X, N, N1, R, ITEST, W, V, PW, PV, IFAIL) | On-line doc: CALL GAMSDOC G08BAF (or @PRT NAG*DOC.G08BAF) | Access: LIB NBS*NAG

G08CAE Kolmogorov-Smirnov one-sample distribution test. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08CAF. | Class(es): L4ac | Usage: CALL G08CAE (N, X, NULL, NP, P, NEST, NTYPE, D, PROB, S, IND, IFAIL) | On-line doc: CALL GAMSDOC G08CAE (or @PRT NAG*DOC.G08CAE) | Access: LIB NBS*NAG

G08CAF Kolmogorov-Smirnov one-sample distribution test. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08CAE. | Class(es): L4ac | Usage: CALL G08CAF (N, X, NULL, NP, P, NEST, NTYPE, D, PROB, S, IND, IFAIL) | On-line doc: CALL GAMSDOC G08CAF (or @PRT NAG*DOC.G08CAF) | Access: LIB NBS*NAG

G08DAE Kendall's coefficient of concordance. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G08DAF. | Class(es): L4lb1 | Usage: CALL G08DAE (X, IX, K, N, RNK, W, P, IFAIL) | On-line doc: CALL GAMSDOC G08DAE (or @PRT NAG*DOC.G08DAE) | Access: LIB NBS*NAG

G08DAF Kendall's coefficient of concordance. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G08DAE. | Class(es): L4lb1 | Usage: CALL G08DAF (X, IX, K, N, RNK, W, P, IFAIL) | On-line doc: CALL GAMSDOC G08DAF (or @PRT NAG*DOC.G08DAF) | Access: LIB NBS*NAG


G13AFA Carries out non-seasonal and seasonal differencing on a time series. Information which allows the original series to be reconstituted from the differenced series is also produced. This information is required in time series forecasting. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AAF. | Class(es): L10b | Usage: CALL G13AFA (X, IX, K, N, RNK, W, P, IFAIL) | On-line doc: CALL GAMSDOC G13AFA (or @PRT NAG*DOC.G13AFA) | Access: LIB NBS*NAG

G13ABF Computes the sample autocorrelation function of a time series. It also computes the sample mean, the sample variance and a statistic which may be used to test the hypothesis that the true autocorrelation function is zero. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13ABE. | Class(es): L10c | Usage: CALL G13ABE (X, IX, K, N, RNK, W, P, IFAIL) | On-line doc: CALL GAMSDOC G13ABE (or @PRT NAG*DOC.G13ABE) | Access: LIB NBS*NAG

G13ACF Calculates partial autocorrelation coefficients given a set of autocorrelation coefficients. It also calculates the predictor error variance ratios for increasing order of finite lag autoregressive predictor, and the autoregressive parameters associated with the predictor of maximum order. | Proprietary single precision Fortran subroutine in NAG library. Single precision version is G13ACE. | Class(es): L10c | Usage: CALL G13ACF (R, NK, NL, P, V, AR, NVL, IFAIL) | On-line doc: CALL GAMSDOC G13ACF (or @PRT NAG*DOC.G13ACF) | Access: LIB NBS*NAG


G13AFA Iteratively fits seasonal autoregressive-integrated moving-average (ARIMA) model to observed time series using non-linear least squares procedure incorporating backforecasting. Returns parameter estimates, standard errors, residual series, and information for use by G13AGF or G13AHF in forecasting. | Proprietary double precision Fortran subroutine in NAG library. Single precision version is...


G13AGE Accepts new observation of fully specified (by G13AEE or G13AFE) time series and updates "state set" (from G13AEE or G13AFE) information for use in further forecasting. Returns residuals corresponding to the new observations, which may be used in checking that new observations conform to previously fitted model. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AGF. | Class(es): L10e2 | Usage: CALL G13AGE (ST, NST, MR, PAR, NPAR, C, ANX, NUV, ANEXR, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AGE (or @PRT NAG+DOC.G13AGE) | Access: LIB NBS+NAG | See also: G13AEE G13AIE

G13AGF Accepts new observation of fully specified (by G13AEE or G13AFE) time series and updates "state set" (from G13AEE or G13AFE) information for use in further forecasting. Returns residuals corresponding to the new observations, which may be used in checking that new observations conform to previously fitted model. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AIF. | Class(es): L10e2 | Usage: CALL G13AGF (ST, NST, MR, PAR, NPAR, C, ANX, NUV, ANEXR, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AGF (or @PRT NAG+DOC.G13AGF) | Access: LIB NBS+NAG | See also: G13AEE G13AIF

G13AHE Produces forecasts of a time series, given model already fitted (by G13AEE or G13AFE). Original observations are not required, since the subroutine uses state set produced originally by G13AEE or G13AFE or updated by G13AGE. Standard errors of the forecasts are also provided. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AHF. | Class(es): L10e2 | Usage: CALL G13AHE (ST, NST, MR, PAR, NPAR, C, RMS, NFV, FVA, FSD, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AHE (or @PRT NAG+DOC.G13AHE) | Access: LIB NBS+NAG | See also: G13AEE G13AIE

G13AHF Produces forecasts of a time series, given model already fitted (by G13AEE or G13AFE). Original observations are not required, since the subroutine uses state set produced originally by G13AEE or G13AFE or updated by G13AGF. Standard errors of the forecasts are also provided. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AHE. | Class(es): L10e2 | Usage: CALL G13AHF (ST, NST, MR, PAR, NPAR, C, RMS, NFV, FVA, FSD, WA, NWA, IFAIL) | On-line doc: CALL GAMSDOC G13AHF (or @PRT NAG+DOC.G13AHF) | Access: LIB NBS+NAG | See also: G13AEE G13AIF

G13AJE Applies a fully specified seasonal ARIMA model to an observed time series, generates the state set for forecasting and (optionally) derives a specified number of forecasts together with their standard deviations. Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13AFJ. | Class(es): L10e2 | Usage: CALL G13AJE(MR, PAR, NPAR, C, KFC, X, NX, RMS, ST, IST, NST, NFV, FVA, FSD, IFV, ISF, W, IW, IFAIL) | On-line doc: CALL GAMSDOC G13AJE (or @PRT NAG+DOC.G13AJE) | Access: LIB NBS+NAG

G13AFJ Applies a fully specified seasonal ARIMA model to an observed time series, generates the state set for forecasting and (optionally) derives a specified number of forecasts together with their standard deviations. Proprietary double precision Fortran subprogram in NAG library. Single precision version is G13AJE. | Class(es): L10e2 | Usage: CALL G13AFJ(MR, PAR, NPAR, C, KFC, X, NX, RMS, ST, IST, NST, NFV, FVA, FSD, IFV, ISF, W, IW, IFAIL) | On-line doc: CALL GAMSDOC G13AFJ (or @PRT NAG+DOC.G13AFJ) | Access: LIB NBS+NAG


G13CAE Calculates the smoothed sample spectrum of a univariate time series using one of four windows - rectangular, Bartlett, Tukey, or Parzen window. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is G13CAF. | Class(es):


GAMCDF Computes the cumulative distribution function value for the gamma distribution with tail length parameter = GAMMA.  | Portable single precision Fortran subprogram in DATAPAC library.  | Class(es): L5a1g  | Usage: CALL GAMCDF(X,GAMMA,CDF) | On-line doc:
CALL GAMSDOC GAMCDF (or @PRT DATAPAC*DOC.GAMCDF) | Access: LIB NBS*DATAPAC

GAMI Incomplete gamma function, \( \int_0^x \frac{e^{-t}}{t^{a-1}} dt \). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMI. | Class(es): C7a | Usage: \( Y = \text{GAMI}(A,X) \) | On-line doc: CALL GAMSDOC GAMI (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMIC Complementary incomplete gamma, \( \int_x^\infty \frac{e^{-t}}{t^{a-1}} dt \). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMIC. | Class(es): C7a | Usage: \( Y = \text{GAMIC}(A,X) \) | On-line doc: CALL GAMSDOC GAMIC (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMIT Tricomi's incomplete gamma, \( x^{-a} \exp(-t) \). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMIT. | Class(es): C7a | Usage: \( Y = \text{GAMIT}(A,X) \) | On-line doc: CALL GAMSDOC GAMIT (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMLN Computes natural log of Gamma function for non-negative argument. Portable single precision Fortran subprogram in AMOSLIB sublibrary of CMLIB library. | Class(es): C7a | Usage: \( Y = \text{GAMLN}(X) \) | On-line doc: CALL GAMSDOC GAMLN (or @PRT CMLIB*DOC.GAMLN/AMOSLIB) | Access: LIB NBS*CMLIB

GAMMA Gamma function. Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMMA. | Class(es): C7a | Usage: \( Y = \text{GAMMA}(X) \) | On-line doc: CALL GAMSDOC GAMMA (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB | See also: GAMILIM

GAMMA Evaluate the gamma function. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): C7a | Usage: \( X = \text{GAMMA}(Y) \) | On-line doc: CALL GAMSDOC GAMMA (or @PRT IMSL*DOC.GAMMA) | Access: LIB NBS*IMSL

GAMPLT Generates a gamma probability plot with tail length parameter = GAMMA, mean = GAMMA, and standard deviation = \( \text{sqrt}(\text{GAMMA}) \). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a7 | Usage: CALL GAMPLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC GAMPLT (or @PRT DATAPAC*DOC.GAMPLT) | Access: LIB NBS*DATAPAC

GAMPPF Computes the percent point function value for the gamma distribution with mean = GAMMA and standard deviation = \( \text{sqrt}(\text{GAMMA}) \). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2g | Usage: CALL GAMPPF(P,GAMMA,PPF) | On-line doc: CALL GAMSDOC GAMPPF (or @PRT DATAPAC*DOC.GAMPPF) | Access: LIB NBS*DATAPAC

GAMMR Reciprocal gamma function, \( 1 / \text{gamma}(x) \). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DGAMMR. | Class(es): C7a | Usage: \( Y = \text{GAMMR}(X) \) | On-line doc: CALL GAMSDOC GAMMR (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: LIB NBS*CMLIB

GAMRAN Generates a random sample of size N from the gamma distribution with tail length parameter = GAMMA, mean = GAMMA and standard deviation=\text{sqrt}(GAMMA). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a7 | Usage: CALL GAMRAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC GAMRAN (or @PRT DATAPAC*DOC.GAMRAN) | Access: LIB NBS*DATAPAC

GAUSQ Finds the abscissae and weights for Gauss quadrature on the interval (a,b) for a general weight function with known moments. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DGAUSSQ. | Class(es): H2c | Usage: CALL GAUSSQ(N,A,B,C,NU,X,W) | On-line doc: CALL GAMSDOC GAUSSQ (or @PRT PORT*DOC.GAUSSQ) | Access: LIB NBS*PORT


GECDF Computes the geometric cumulative distribution function value at the value X with parameter \( P \). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1g | Usage: CALL GECDFC(X,P,CDF) | On-line doc: CALL GAMSDOC GECDF (or @PRT DATAPAC*DOC.GECDF) | Access: LIB NBS*DATAPAC

GEOPLT Generates a geometric probability plot with parameter P. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3a4g | Usage: CALL GEOPLT(X,N,P) | On-line doc: CALL GAMSDOC GEOPLT (or @PRT DATAPAC*DOC.GEOPLT) | Access: LIB NBS*DATAPAC

GEOPPF Computes the percent point function value for the geometric distribution with parameter PPAR. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2g | Usage: CALL GEOPPF(P,PPAR,PPF) | On-line doc: CALL GAMSDOC GEOPPF (or @PRT DATAPAC*DOC.GEOPPF) | Access: LIB NBS*DATAPAC

GEORAN Generates a random sample of size N from the geometric distribution with parameter P. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a7 | Usage: CALL GEORAN(N,P,ISTART,X) | On-line doc: CALL GAMSDOC GEORAN (or @PRT DATAPAC*DOC.GEORAN) | Access: LIB NBS*DATAPAC

GFIT Chi-squared goodness of fit test. Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1c | Usage: CALL GFIT(CDF,K, OBS,N,CELLS,COMP,CS,IDL,F1ER) | On-line doc: CALL GAMSDOC GFIT (or @PRT IMSL*DOC.GFIT) | Access: LIB NBS*IMSL

GGAMR One parameter gamma random deviate generator, and usable as basis for 2 parameter gamma, exponential, chi-squared, chi, beta, t
January 1984

C 100

GAMS: Module Dictionary


GGCAV | Cauchy random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a3 | Usage: CALL GGCAV (DSEED,NE,NN,WK,R) | On-line doc: CALL GAMSDOC GGCAV (or @PRT IMSL+DOC.GGCAV) | Access: LIB NBS+IMSL

GGCHS | Chi-squared random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a3 | Usage: CALL GGCHS (DSEED,NE,NN,CHI2) | On-line doc: CALL GAMSDOC GGCHS (or @PRT IMSL+DOC.GGCHS) | Access: LIB NBS+IMSL

GGCG | Chi-squared random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a3 | Usage: CALL GGCG (DSEED,NE,NN,CHI2) | On-line doc: CALL GAMSDOC GGCG (or @PRT IMSL+DOC.GGCG) | Access: LIB NBS+IMSL


GGEX | Exponential random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a5 | Usage: CALL GGEX (DSEED,NE,XM,NN,R) | On-line doc: CALL GAMSDOC GGEX (or @PRT IMSL+DOC.GGEX) | Access: LIB NBS+IMSL

GGEXN | Exponential random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a5 | Usage: CALL GGEXN (DSEED,NE,XM,XM1,NN,R,R) | On-line doc: CALL GAMSDOC GGEXN (or @PRT IMSL+DOC.GGEXN) | Access: LIB NBS+IMSL

GGEXT | Random deviate generator for a mixture of two exponentials. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a5 | Usage: CALL GGEXT (DSEED,P,XM1,XM3,NN,R,R) | On-line doc: CALL GAMSDOC GGEXT (or @PRT IMSL+DOC.GGEXT) | Access: LIB NBS+IMSL

GGG | Hypergeometric random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a8 | Usage: CALL GGGHPR (DSEED,NL,M,NN,WK,R) | On-line doc: CALL GAMSDOC GGGHPR (or @PRT IMSL+DOC.GGGHPR) | Access: LIB NBS+IMSL


GGNML | Normal or Gaussian random deviate generator. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a14 | Usage: CALL GGNML (DSEED,NE,R) | On-line doc: CALL GAMSDOC GGNML (or @PRT IMSL+DOC.GGNML) | Access: LIB NBS+IMSL

GGNO | Create set of order statistics from normal distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a15 L6a15 | Usage: CALL GGNO (DSEED,I,FIRST,ILAST,NE,R) | On-line doc: CALL GAMSDOC GGNO (or @PRT IMSL+DOC.GGNO) | Access: LIB NBS+IMSL

GGNPM | Normal random deviate generator via the polar method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a14 | Usage: CALL GGNPM (DSEED,NE,R) | On-line doc: CALL GAMSDOC GGNPM (or @PRT IMSL+DOC.GGNPM) | Access: LIB NBS+IMSL

GGNPP | Nonhomogeneous Poisson process generator with rate function lambda(t) - fixed interval, fixed number, or one at a time. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L6a16 | Usage: CALL GGNPP (DSEED,T,L,TU,NB,FUNLAM,RLAMAX,RLAMIN,I,OPT,NE,R) | On-line doc: CALL GAMSDOC GGNPP (or @PRT IMSL+DOC.GGNPP) | Access: LIB NBS+IMSL


Note: The text contains a mix of programming-related terms and mathematical expressions, typical of a subroutine documentation manual.

GGPER  Generate a random permutation of the integers 1 to k. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a16 | Usage: CALL GGPER (DSEED, K, IPER) | On-line doc: CALL GAMSDOC GGPER (or @PRT IMSL*DOC.GGPER) | Access: LIB NBS*IMSL]

GGPON  Poisson random deviate generator where the Poisson parameter changes frequently. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a16 | Usage: CALL GGPN (RLAM, DSEED, NR, IR, IER) | On-line doc: CALL GAMSDOC GGPON (or @PRT IMSL*DOC.GGPON) | Access: LIB NBS*IMSL]

GGPOS  Poisson random deviate generator where the Poisson parameter does not change often. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a16 | Usage: CALL GGPOS (RLAM, DSEED, NR, IR, IER) | On-line doc: CALL GAMSDOC GGPOS (or @PRT IMSL*DOC.GGPOS) | Access: LIB NBS*IMSL]

GGSPH  Generation of uniform random deviates from the surface of the unit sphere in 3 or 4 space. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6b21 | Usage: CALL GGSPH (DSEED, NR, IOP, IZ, IER) | On-line doc: CALL GAMSDOC GGSPH (or @PRT IMSL*DOC.GGSPH) | Access: LIB NBS*IMSL]

GGSRS  Generate a simple random sample from a finite population. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a10 | Usage: CALL GGSRS (DSEED, IOP, NP, IP, MPOS, POP, NSAMP, MSAMP, SAM, IX, IER) | On-line doc: CALL GAMSDOC GGSRS (or @PRT IMSL*DOC.GGSRS) | Access: LIB NBS*IMSL]

GGSTA  Stable distribution random deviate generator. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a10 | Usage: CALL GGSTA (DSEED, ALPHA, BPRIM, NR, R) | On-line doc: CALL GAMSDOC GGSTA (or @PRT IMSL*DOC.GGSTA) | Access: LIB NBS*IMSL]

GGTAB  Generate a random contingency table with given row and column totals. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6b3 | Usage: CALL GGTAB (DSEED, NROW, NCOL, NCTOT, IND, IIT, IATB, IWK, IER) | On-line doc: CALL GAMSDOC GGTAB (or @PRT IMSL*DOC.GGTAB) | Access: LIB NBS*IMSL]

GGTRA  Triangular distribution random deviate generator. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a20 | Usage: CALL GGTRA (DSEED, NR, R) | On-line doc: CALL GAMSDOC GGTRA (or @PRT IMSL*DOC.GGTRA) | Access: LIB NBS*IMSL]

GGUBFS  Basic uniform (0, 1) random number generator. Function form of GGUBS. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a21 | Usage: X = GGUBFS (DSEED) | On-line doc: CALL GAMSDOC GGUBFS (or @PRT IMSL*DOC.GGUBFS) | Access: LIB NBS*IMSL]

GGUBS  Basic uniform (0, 1) pseudo-random number generator. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a21 | Usage: CALL GGUBS (DSEED, NR, R) | On-line doc: CALL GAMSDOC GGUBS (or @PRT IMSL*DOC.GGUBS) | Access: LIB NBS*IMSL]

GGUBT  Uniform (0, 1) pseudo-random number generator using alternate multiplier. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a21 | Usage: CALL GGUBT (DSEED, NR, R) | On-line doc: CALL GAMSDOC GGUBT (or @PRT IMSL*DOC.GGUBT) | Access: LIB NBS*IMSL]

GGUD  Discrete uniform random number generator. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a21 | Usage: CALL GGUD (DSEED, KN, NR, IR) | On-line doc: CALL GAMSDOC GGUD (or @PRT IMSL*DOC.GGUD) | Access: LIB NBS*IMSL]

GGOU  Generate set of order statistics from uniform (0, 1) distribution. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a15 L6a21 | Usage: CALL GGOU (DSEED, IFIRST, ILAST, N, IER) | On-line doc: CALL GAMSDOC GGOU (or @PRT IMSL*DOC.GGOU) | Access: LIB NBS*IMSL]

GGUW  Uniform (0, 1) random number generator with shuffling. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a21 | Usage: CALL GGUW (DSEED, NR, IOP, R) | On-line doc: CALL GAMSDOC GGUW (or @PRT IMSL*DOC.GGUW) | Access: LIB NBS*IMSL]


GGWIB  Bullwinkle random deviate generator. [Proprietary single precision Fortran subprogram in IMSL library.  Class(es): L6a23 | Usage: CALL GGWIB (DSEED, AN, R) | On-line doc: CALL GAMSDOC GGWIB (or @PRT IMSL*DOC.GGWIB) | Access: LIB NBS*IMSL]

GQ01N  Finds the abscissae and weights for Gauss Laguerre quadrature on the interval (0,∞). [Proprietary single precision Fortran subprogram in PORT library. Double precision version is DGQ01N.  Class(es): H2 | Usage: CALL GQ01N (N, X, W) | On-line doc: CALL GAMSDOC GQ01N (or @PRT PORT*DOC.GQ01N) | Access: LIB NBS*PORT]
GQM11 Finds the abscissae and weights for Gauss Legendre quadrature on the interval (-1,1). | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DGQM11. | Class(es): | Usage: | On-line doc: | CALL GAMSDOC GQM11 (or @PRT PORT*DOC.GQM11) | Access: | LIB NBS*PORT

GTCN Sample size or number of class intervals determination for chi-squared test applications. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4a1c | Usage: | On-line doc: | CALL GAMSDOC GTCN (Q,IOPT,B,K,N,IER) | Access: | LIB NBS*PORT


H01ABE Linear programming, simplex algorithm, one iteration. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01ABF. | Class(es): G2a1 | Usage: CALL H01ABE (NROW, NCOL, NMVR, NTVR, A, F, IND, IDIM, NOTAB, IRES, IBV, SC, Z) | On-line doc: CALL GAMSDOC H01ABE (or @PRT NAG+DOC.H01ABE) | Access: LIB NBS+NAG

H01ABF Linear programming, simplex algorithm, one iteration. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01ABE. | Class(es): G2a1 | Usage: CALL H01ABF (NROW, NCOL, NMVR, NTVR, A, F, IND, IDIM, NOTAB, IRES, IBV, SC, Z) | On-line doc: CALL GAMSDOC H01ABF (or @PRT NAG+DOC.H01ABF) | Access: LIB NBS+NAG

H01ADE Linear programming, revised simplex method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01ADF. | Class(es): G2a1 | Usage: CALL H01ADE (A, MMM, M, N, INEQ, RHS, MIN, MAXIT, MN, MM, M1, M2, CO, IN, II, NBV, D, B, ANS, OPT, NUMIT, IPAR, IFAIL) | On-line doc: CALL GAMSDOC H01ADE (or @PRT NAG+DOC.H01ADE) | Access: LIB NBS+NAG

H01ADF Linear programming, revised simplex method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01ADE. | Class(es): G2a1 | Usage: CALL H01ADF (A, MMM, M, N, INEQ, RHS, MIN, MAXIT, MN, MM, M1, M2, CO, IN, II, NBV, D, B, ANS, OPT, NUMIT, IPAR, IFAIL) | On-line doc: CALL GAMSDOC H01ADF (or @PRT NAG+DOC.H01ADF) | Access: LIB NBS+NAG

H01AFE Find feasible point or vertex which satisfies linear constraints. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01AFF. | Class(es): G4d | Usage: CALL H01AFE (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, STATE, LNGC, INITPT, MSGVLV MAXIT, VERTEX,X,NA,R,INDEX,LOCX,LEL,EL,ELQ,LTQ,LTQ,J, IW,LI,W,LW,LFFAIL) | On-line doc: CALL GAMSDOC H01AFE (or @PRT NAG+DOC.H01AFE) | Access: LIB NBS+NAG

H01AFF Find feasible point or vertex which satisfies linear constraints. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01AFE. | Class(es): G4d | Usage: CALL H01AFF (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, STATE, LNGC, INITPT, MSGVLV MAXIT, VERTEX,X,NA,R,INDEX,LOCX,LEL,EL,ELQ,LTQ,LTQ,J, IW,LI,W,LW,LFFAIL) | On-line doc: CALL GAMSDOC H01AFF (or @PRT NAG+DOC.H01AFF) | Access: LIB NBS+NAG

H01BAE Linear programming, numerically stable form of simplex method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H01BAF. | Class(es): G2a1 | Usage: CALL H01BAE (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, STATE, LNGC, INITPT, MSGVLV MAXIT, C,VERTEX,X,F,YMOD,NA,UMIN,R,U,INDEX,LOCX,EL,ELQ,LTQ,LTQ,J, IW,LI,W,LW) | On-line doc: CALL GAMSDOC H01BAE (or @PRT NAG+DOC.H01BAE) | Access: LIB NBS+NAG

H01BAF Linear programming, numerically stable form of simplex method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H01BAE. | Class(es): G2a1 | Usage: CALL H01BAF (N, NGC, AGC, LAGC, IBOUND, BL, BU, LB, STATE, LNGC, INITPT, MSGVLV MAXIT, C,VERTEX,X,F,YMOD,NA,UMIN,R,U,INDEX,LOCX,EL,ELQ,LTQ,LTQ,J, IW,LI,W,LW) | On-line doc: CALL GAMSDOC H01BAF (or @PRT NAG+DOC.H01BAF) | Access: LIB NBS+NAG

H02AAE Quadratic programming, Beale's method. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H02AAF. | Class(es): G2c6 | Usage: CALL H02AAE (MV, N1, MN, ISYM, P, LES1, LES2, MAR, COLK, E) | On-line doc: CALL GAMSDOC H02AAE (or @PRT NAG+DOC.H02AAE) | Access: LIB NBS+NAG

H02AAF Quadratic programming, Beale's method. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H02AAE. | Class(es): G2c6 | Usage: CALL H02AAF (MV, N1, MN, ISYM, P, LES1, LES2, MAR, COLK, E) | On-line doc: CALL GAMSDOC H02AAF (or @PRT NAG+DOC.H02AAF) | Access: LIB NBS+NAG

H02BAAE Integer linear programming, Gomory's method with Wilson's cuts. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H02BAFE. | Class(es): G2c6 | Usage: CALL H02BAAE (A, MM, N1, M, N, MAXIT, LL, X, NUMIT, OPT, IFAIL) | On-line doc: CALL GAMSDOC H02BAAE (or @PRT NAG+DOC.H02BAAE) | Access: LIB NBS+NAG

H02BAFE Integer linear programming, Gomory's method with Wilson's cuts. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H02BAAE. | Class(es): G2c6 | Usage: CALL H02BAFE (A, MM, N1, M, N, MAXIT, LL, X, NUMIT, OPT, IFAIL) | On-line doc: CALL GAMSDOC H02BAFE (or @PRT NAG+DOC.H02BAFE) | Access: LIB NBS+NAG

H03ABE Solves the classical Transportation ("Hitchcock") problem. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is H03ABF. | Class(es): G2b | Usage: CALL H03ABE(KOST,MMM,MA,MB,M,K18,MAXIT,K7,K9,NUMIT,K8,K8,K11, K12,2,IFAIL) | On-line doc: CALL GAMSDOC H03ABE (or @PRT NAG+DOC.H03ABE) | Access: LIB NBS+NAG

H03ABF Solves the classical Transportation ("Hitchcock") problem. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is H03ABE. | Class(es): G2b | Usage: CALL H03ABF (KOST, MMM, MA, MB, M, K15, MAXIT, K7, K9, NUMIT, K6, K8, K11, K12, Z, IFAIL) | On-line doc: CALL GAMSDOC H03ABF (or @PRT NAG+DOC.H03ABF) | Access: LIB NBS+NAG

HFNCDP Computes the cumulative distribution function value for the halfnormal distribution with mean = sqrt(2/pi) and standard deviation = 1. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5atb | Usage: CALL HFNCDP(X,DF) | On-line doc: CALL GAMSDOC HFNCDP (or @PRT DATAPAC+DOC.HFNCDP) | Access: LIB NBS+DATAPAC

HFNPLT Generates a halfnormal probability plot with mean = sqrt(2/pi) and standard deviation = 1. | Portable single precision Fortran
subprogram in DATAPAC library.  | Class(es): L3a4h  | Usage: CALL HFNPLT(X,N)  | On-line doc: CALL GAMSDOC HFNPLT (or @PRT DATAPAC+DOC.HFNPLT)  | Access: LIB NBS+DATAPAC

HFNPPF Computes the percent point function value for the halfnormal distribution with mean = sqrt(2/pi) and standard deviation = 1.  | Portable single precision Fortran subprogram in DATAPAC library.  | Class(es): L5a2h  | Usage: CALL HFNPPF(P,PPF)  | On-line doc: CALL GAMSDOC HFNPPF (or @PRT DATAPAC+DOC.HFNPPF)  | Access: LIB NBS+DATAPAC

HFNRAH Generates a random sample of size N from the halfnormal distribution with mean = sqrt(2/pi) and standard deviation = 1.  | Portable single precision Fortran subprogram in DATAPAC library.  | Class(es): L6a8  | Usage: CALL HFNRAH(N,JSTART,X)  | On-line doc: CALL GAMSDOC HFNRAH (or @PRT DATAPAC+DOC.HFNRAH)  | Access: LIB NBS+DATAPAC


HIST Produces 2 histograms (with differing class widths) of the data in the input vector X.  | Portable single precision Fortran subprogram in DATAPAC library.  | Class(es): L3a Q1  | Usage: CALL HIST(X,N)  | On-line doc: CALL GAMSDOC HIST (or @PRT DATAPAC+DOC.HIST)  | Access: LIB NBS+DATAPAC

HISTO Produces a histogram and summary statistics.  | Portable single precision Fortran subprogram in STATLIB library.  | Class(es): L3a  | Usage: CALL HISTO (Y, N, NCELL, YLB, YUB, SCRAT, NS)  | On-line doc: CALL GAMSDOC HISTO (or @PRT STATLIB+DOC.HISTO)  | Tests: STATLIB+TEST.DEMO1  | Access: LIB NBS+STATLIB

HISTOG Produces a histogram and summary statistics, with user control of the number of cells, and of the upper and lower histogram boundaries.  | Portable single precision Fortran subprogram in STATLIB library.  | Class(es): L3a  | Usage: CALL HISTOC (Y, N, NCELL, YLB, YUB, SCRAT, NS)  | On-line doc: CALL GAMSDOC HISTOC (or @PRT STATLIB+DOC.HISTOC)  | Tests: STATLIB+TEST.DEMO1  | Access: LIB NBS+STATLIB

HISTOGRAM Prints a histogram of the values in each of one or more vectors, with optional user specification of the first midpoint and the interval width.  | Command in MINITAB Proprietary interactive system.  | Class(es): L3a Q1  | Usage: HISTogram of C [...] and C] [first midpoint K, interval width K]  | On-line doc: HELP HISTOGRAM (in Minitab)  | Tests: MINITAB+TEST-SOURCE.  | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CRS)

HQR Computes eigenvalues of a real upper Hessenberg matrix using the QR method.  | Portable single precision Fortran subprogram in EISPACK sublibary of CMLIB library.  | Class(es): D4e2b  | Usage: CALL HQR(NM,N,LOW,IGH,H,WR,WI,IER,ERR)  | On-line doc: CALL GAMSDOC HQR (or @PRT CMLIB+DOC.HQR/EISPACK)  | Access: LIB NBS+CMLIB


HTRIB3 Computes eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from

HTRIBK | Forms eigenvectors of complex Hermitian matrix from eigenvectors of real symmetric tridiagonal matrix output from HTRID1. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL HTRIBK(NM,N,A,TAU,M,ZR,IZ) | On-line doc: CALL GAMSDOC HTRIBK (or @PRT CMLIB\*DOC.HTRIBK/EISPACK) | Access: LIB NBS\*CMLIB | See also: HTRID1

HTRID3 | Reduces complex Hermitian (packed) matrix to real symmetric tridiagonal matrix by unitary similarity transformations. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL HTRID3(NM,N,A,D,E2,TAU) | On-line doc: CALL GAMSDOC HTRID3 (or @PRT CMLIB\*DOC.HTRID3/EISPACK) | Access: LIB NBS\*CMLIB

HUMSL | Minimizes a general unconstrained objective function using (analytic) gradient and Hessian provided by the user. | Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DHUML. | Class(es): G1b1c | Usage: CALL HMLSL(N,D,CALCF,CALCGH,IV,LV,F,PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HUMSL (or @PRT CMLIB\*DOC.HUMSL/NL2SN) | Tests: CMLIB\*TEST-SOURCE.$F2/NL2SN, CMLIB\*TEST-SOURCE.$Q3/NL2SN | Access: LIB NBS\*CMLIB

HW3CRT | Solves the Helmholtz or Poisson equation in three dimensions using Cartesian coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a | Usage: CALL HW3CRT(XS,XY,XX,LD,DD,DDX,DXX,DSY,DFS,DXY,F,PS,PF,N,PB,PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HW3CRT (or @PRT CMLIB\*DOC.HW3CRT/FSHPK) | Tests: CMLIB\*TEST-SOURCE.HW3CRT/FSHPK | Access: LIB NBS\*CMLIB

HW3CRT | Solves the Helmholtz or Poisson equation in three dimensions using Cartesian coordinates. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a | Usage: CALL HW3CRT(A,B,M,MBBCN,D,B,B,D,D,N,MBBCN,D,C,D,ELMBD,F,IV,PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HW3CRT (or @PRT CMLIB\*DOC.HW3CRT/FSHPK) | Tests: CMLIB\*TEST-SOURCE.HW3CRT/FSHPK | Access: LIB NBS\*CMLIB

HW3SP | Solves a modified Helmholtz equation in spherical coordinates with axisymmetry. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a | Usage: CALL HW3SP(S,TS,TF,M,MBBCN,D,B,B,T,D,R,F,N,NBDN,D,B,C,D,ELMBD,F,IDMF,PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HW3SP (or @PRT CMLIB\*DOC.HW3SP/FSHPK) | Tests: CMLIB\*TEST-SOURCE.HW3SP/FSHPK | Access: LIB NBS\*CMLIB


HW3SP | Solves the Helmholtz or Poisson equation in spherical coordinates on the surface of a sphere. | Portable single precision Fortran subprogram in FSHPK sublibrary of CMLIB library. | Class(es): I2b1a | Usage: CALL HW3SP(S,T,M,MBBCN,D,B,B,T,D,R,F,N,NBDN,D,B,C,D,ELMBD,F,IDMF,PERTRB,IERROR,W) | On-line doc: CALL GAMSDOC HW3SP (or @PRT CMLIB\*DOC.HW3SP/FSHPK) | Tests: CMLIB\*TEST-SOURCE.HW3SP/FSHPK | Access: LIB NBS\*CMLIB
IIMACH  Provides integer machine dependent information, e.g. largest integer.  Portable single precision Fortran subprogram in MACHCONST sublibrary of CMLIB library.  Class(es): R1  Usage: J=IIMACH(I)  On-line doc: CALL GAMSDOC IIMACH (or @PRT CMLIB+DOC.IIMACH/MACHCONST)  Tests: CMLIB+TEST-SOURCE.$Q/MACHCONST  Access: LIB NBS+CMLIB

IIMACH  Provides the integer constants required to adapt PORT library programs to individual computers.  Proprietary single precision Fortran subprogram in PORT library.  Class(es): R1  Usage: I = IIMACH(I)  On-line doc: CALL GAMSDOC IIMACH (or @PRT PORT+DOC.IIMACH)  Access: LIB NBS+PORT

IASPEC  Computes the integrated sample periodogram of a series (not recommended for long series because of the algorithm used).  Portable single precision Fortran subprogram in STATLIB library.  Class(es): L10f Usage: CALL IASPEC(Y,N,SCRAT,NS)  On-line doc: CALL GAMSDOC IASPEC (or @PRT STATLIB+DOC.IASPEC)  Tests: STATLIB+TEST.DEMO4  Access: LIB NBS+STATLIB


ICEIL  Finds the smallest integer greater than or equal to x.  Portable single precision Fortran subprogram in IMSL library.  Class(es): C1 Usage: I = ICEIL(X)  On-line doc: CALL GAMSDOC ICEIL (or @PRT PORT+DOC.ICEIL)  Access: LIB NBS+PORT


ICSFKU  Least squares approximation by cubic splines fixed knots.  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): K1a1 Usage: CALL ICSFKU(X,F,NX,MODE,XK,NXY,C,IC,ERROR,WK,IER)  On-line doc: CALL GAMSDOC ICSFKU (or @PRT IMSL+DOC.ICSFKU)  Access: LIB NBS+IMSL  See also: ICSEVU DCSEVU

ICSICU  Interpolatory approximation by cubic splines with arbitrary second derivative end conditions.  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): E1a Usage: CALL ICSICU(Y,NX,BPAR,C,IC,IER)  On-line doc: CALL GAMSDOC ICSICU (or @PRT IMSL+DOC.ICSICU)  Access: LIB NBS+IMSL  See also: ICSEVU DCSEVU

ICSMOU  One-dimensional data smoothing by error detection.  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): K5 L10b Usage: CALL ICSMOU(X,Y,NX,DIS,SC,MAXIT,WK,IER)  On-line doc: CALL GAMSDOC ICSMOU (or @PRT IMSL+DOC.ICSMOU)  Access: LIB NBS+IMSL

ICSPEC  Displays plots of the integrated sample phase and co-spectra for a pair of series (not recommended for long series because of the algorithm used).  Portable single precision Fortran subprogram in STATLIB library.  Class(es): L10g1 Usage: CALL ICSPEC(Y1,Y2,NS,SCRAT)  On-line doc: CALL GAMSDOC ICSPEC (or @PRT STATLIB+DOC.ICSPEC)  Tests: STATLIB+TEST.DEMO4  Access: LIB NBS+STATLIB

ICSPLN  Cubic spline interpolation with periodic end conditions.  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): E1a Usage: CALL ICSPLN(X,Y,NX,C,IC,WK,IER)  On-line doc: CALL GAMSDOC ICSPLN (or @PRT IMSL+DOC.ICSPILL)  Access: LIB NBS+IMSL  See also: ICSEVU DCSEVU

ICSSCU  Cubic spline data smoother.  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): K5 K1a1 Usage: CALL ICSSCU(X,F,NX,Y,NX,SM,X,C,IC,WK,IER)  On-line doc: CALL GAMSDOC ICSSCU (or @PRT IMSL+DOC.ICSSCU)  Access: LIB NBS+IMSL  See also: ICSEVU DCSEVU

ICSSCV  Cubic spline data smoother (easy-to-use version).  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): K5 K1a1 Usage: CALL ICSSCV(X,F,NX,Y,NX,C,IC,WK,IER)  On-line doc: CALL GAMSDOC ICSSCV (or @PRT IMSL+DOC.ICSSCV)  Access: LIB NBS+IMSL  See also: ICSEVU DCSEVU

ICSVKU  Least squares approximation by cubic splines - variable knots.  Proprietary single precision Fortran subprogram in IMSL library.  Class(es): E1a Usage: CALL ICSVKU(X,Y,NX,BPAR,C,IC,IER)  On-line doc: CALL GAMSDOC ICSVKU (or @PRT IMSL+DOC.ICSVKU)  Access: LIB NBS+IMSL  See also: ICSEVU DCSEVU
GAMS: Module Dictionary

January 1984

Class(es): KI1a1 | Usage: CALL LCSVU (X,F,NX,KX,NK,X,Y,C,J,IPROR,WK,IUR) | On-line doc: CALL GAMSDOC LCSVU (or @PRT IMSL+DOCSVU) | Access: LIB NBS+IMSL | See also: ICSEVU DCSEVU

IDMAX Find smallest index of maximum magnitude component of a double precision vector. Portable double precision Fortran subprogram in BLAS sublibrary of CMLIB library. Single precision version is ISAMAX. | Class(es): D1A2 | Usage: IMA = IDMAX(N,DX,JNXC) | On-line doc: CALL GAMSDOC IDMAX (or @PRT CMLIB+DOCSIDMAX(BLAS)) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

IDCEIL Finds the smallest integer greater than or equal to x. Input is double precision output is integer. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): C1 | Usage: I = IDCEIL(X) | On-line doc: CALL GAMSDOC IDCEIL (or @PRT PORT+DOCSIDCEIL) | Access: LIB NBS+PORT

IDFLR Finds the largest integer less than or equal to x. Input is double precision, output is integer. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IFLR. | Class(es): C1 | Usage: I = IDFLR(X) | On-line doc: CALL GAMSDOC IDFLR (or @PRT PORT+DOCSIDFLR) | Access: LIB NBS+PORT

IDLUMB Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ILUMB. | Class(es): E3 K6 | Usage: I = IDLUMB(XB,NXB,NK,NX) | On-line doc: CALL GAMSDOC IDLUMB (or @PRT PORT+DOCSIDLUMB) | Access: LIB NBS+PORT

IDLUMD Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is ILUMD. | Class(es): E3 K6 | Usage: I = IDLUMD(XB,NXB,NK,NX) | On-line doc: CALL GAMSDOC IDLUMD (or @PRT PORT+DOCSIDLUMD) | Access: LIB NBS+PORT

IDMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): E3 K6 | Usage: I = IDMNPB(X,NX,N,K,NT) | On-line doc: CALL GAMSDOC IDMNPB (or @PRT PORT+DOCSIDMNPB) | Access: LIB NBS+PORT

IDPUMB Given a basic mesh, this subdivides each interval. Number of points per interval can vary, but uniform in each subdivision. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IPUMB. | Class(es): E3 K6 | Usage: I = IDPUMB(XB,NXB,NA,K,NN) | On-line doc: CALL GAMSDOC IDPUMB (or @PRT PORT+DOCSIDPUMB) | Access: LIB NBS+PORT

IDPUMD Given a basic mesh, this subdivides each interval with a variable number of points. Points are uniform in each interval. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IPUMD. | Class(es): E3 K6 | Usage: I = IDPUMD(XB,NXB,NA,NN) | On-line doc: CALL GAMSDOC IDPUMD (or @PRT PORT+DOCSIDPUMD) | Access: LIB NBS+PORT

IDUMB Given interval endpoints, this generates a uniform mesh for B-spline use. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IUMB. | Class(es): E3 K6 | Usage: I = IDUMB(A,B,NAB,K,NX) | On-line doc: CALL GAMSDOC IDUMB (or @PRT PORT+DOCSIDUMB) | Access: LIB NBS+PORT

IDUMD Given interval endpoints, this generates a uniform mesh. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is IUMD. | Class(es): E3 K6 | Usage: I = IDUMD(A,B,NAB) | On-line doc: CALL GAMSDOC IDUMD (or @PRT PORT+DOCSIDUMD) | Access: LIB NBS+PORT

IFL R Finds the largest integer less than or equal to x. Input is real output is integer. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDLR. | Class(es): C1 | Usage: I = IDLR(X) | On-line doc: CALL GAMSDOC IDLR (or @PRT PORT+DOCSIDLR) | Access: LIB NBS+PORT


ILUMB Given a basic mesh, this subdivides each interval into the same number of uniformly spaced points for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is ILUMB. | Class(es): E3 K6 | Usage: I = ILUMB(XB,NXB,NK,NX) | On-line doc: CALL GAMSDOC ILUMB (or @PRT PORT+DOCSILUMB) | Access: LIB NBS+PORT

ILUMD Given a basic mesh, this subdivides each interval into the same number or uniformly spaced points for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is ILUMD. | Class(es): E3 K6 | Usage: I = ILUMD(XB,NXB,NK,NX) | On-line doc: CALL GAMSDOC ILUMD (or @PRT PORT+DOCSILUMD) | Access: LIB NBS+PORT

IMNPB Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each mesh interval. | Proprietary single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4A5 D4V2A | Usage: CALL IMNPB(X,NX,N,K,NT) | On-line doc: CALL GAMSDOC IMNPB (or @PRT CMLIB+DOCSIMNPB) | Access: LIB NBS+PORT


IMTQL2 Computes eigenvalues and eigenvectors of symmetric tridiagonal matrix using implicit QL method. Portable single precision Fortran
subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL IMTQL2(NM,N,D,E,Z,IERR) | On-line doc: CALL GAMSDOC IMTQL2 (or @PRT CMLIB+DOC.IMTQL2/EISPACK) | Access: LIB NBS+CMLIB

IMTQLV Computes eigenvalues of symmetric tridiagonal matrix using implicit QL method. Eigenvectors may be computed later. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL IMTQLV(N,D,E2,W,IND,IERR,RV1) | On-line doc: CALL GAMSDOC IMTQLV (or @PRT CMLIB+DOC.IMTQLV/EISPACK) | Access: LIB NBS+CMLIB

INTRV Computes the index into a knot or breakpoint sequence corresponding to a given point X. | Portable single precision Fortran subprogram in BSPLINE sublibrary of CMLIB library. Double precision version is DINTRV. | Class(es): E3 K6 | Usage: CALL INTRV(KT,LXT,X,ID,L,T,MFLAG) | On-line doc: CALL GAMSDOC INTRV (or @PRT CMLIB+DOC.BSPLINE and CMLIB+DOC.SUMMARY/BSPLINE) | Tests: CMLIB+TEST-SOURCE.3F/BSPLINE | Access: LIB NBS+CMLIB

INTRVD Finds the interval in a double precision array to which a double precision element belongs. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is INTRVR. | Class(es): N5b | Usage: I = INTRVD(N, D, DD) | On-line doc: CALL GAMSDOC INTRVD (or @PRT PORT+DOC.INTRVD) | Access: LIB NBS+PORT

INTRVI Finds the interval in an integer array to which an integer element belongs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): N6d | Usage: I = INTRVI(N, l, II) | On-line doc: CALL GAMSDOC INTRVI (or @PRT PORT+DOC.INTRVI) | Access: LIB NBS+PORT

INTRVR Finds the interval in a real array to which a real element belongs. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is INTRVD. | Class(es): N5b | Usage: I = INTRVR(N, R, RR) | On-line doc: CALL GAMSDOC INTRVR (or @PRT PORT+DOC.INTRVR) | Access: LIB NBS+PORT

INVAR1 Interactive packages for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results. Line printer graphics only. | Portable stand-alone program using INVAR command language. | Class(es): L8g K1b1a1 K1b2a2 | Usage: -0- | On-line doc: CALL GAMSDOC INVAR1 (or @PRT INVAR1+DOC.INVAR) | Access: CALL SCD+CTSLIB.<program name> (in CTS)

INVAR2 Interactive packages for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISPLA graphics. | Portable stand-alone program using INVAR command language. | Class(es): L8g K1b1a1 K1b2a2 | Usage: -0- | On-line doc: CALL GAMSDOC INVAR2 (or @PRT INVAR2+DOC.INVAR) | Access: CALL SCD+CTSLIB.<program name> (in CTS)

INVIT Computes eigenvectors of upper Hessenberg (real) matrix associated with specified eigenvalues by inverse iteration. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2b | Usage: CALL INVIT(NM,N,A,WR,W1,SELECT,MM,M,Z,IERR,RM1,RV1,RV2) | On-line doc: CALL GAMSDOC INVIT (or @PRT CMLIB+DOC.INVIT/EISPACK) | Access: LIB NBS+CMLIB

IPUMB Given a basic mesh, this subdivide each interval. Number of points per interval can vary, but uniform in each subdivision. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDPUMB. | Class(es): E3 K8 | Usage: I = IPUMB(XB,NXB,NA,K,NX) | On-line doc: CALL GAMSDOC IPUMB (or @PRT PORT+DOC.IPUMB) | Access: LIB NBS+PORT

IPUMD Given a basic mesh, this subdivide each interval with a variable number of points. Points are uniform in each interval. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDPUMD. | Class(es): E3 K8 | Usage: I = IPUMD(XB,NXB,NA,NX) | On-line doc: CALL GAMSDOC IPUMD (or @PRT PORT+DOC.IPUMD) | Access: LIB NBS+PORT

IQHSCU Visually pleasing interpolant of one dimensional data via piecewise cubic Hermite function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E1a | Usage: CALL IQHSCU(X,Y,NX,C,IC,IER) | On-line doc: CALL GAMSDOC IQHSCU (or @PRT IMSL+DOC.IQHSCU) | Access: LIB NBS+IMSL

IQHSCV Smooth surface fitting with irregularly distributed data points. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): E2b | Usage: CALL IQHSCV(XD,YD,ZD,NX,NXI,NY,NI,ZI,IZI,IKW,IK,IER) | On-line doc: CALL GAMSDOC IQHSCV (or @PRT IMSL+DOC.IQHSCV) | Access: LIB NBS+IMSL


ISAMAX Find smallest index of maximum magnitude component of a single precision vector. | Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is IDAMAX. | Class(es): D1a2 | Usage: IMAK = ISAMAX(N, SX, ICX) |
GAMS: Module Dictionary

Op-line doc: CALL GAMSDOC ISAMAX (or @PRT CMLIB+DOC.ISAMAX/BLAS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

ISMIN Find the smallest index of the minimum magnitude component of a real vector. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a2 | Usage: i=ISMIN(N,SX,INCX) | On-line doc: CALL GAMSDOC ISMIN (or @PRT CMLIB+DOC.ISMIN/XBLAS) | Tests: CMLIB+TEST-SOURCE.$Q/XBLAS | Access: LIB NBS+CMLIB

ISMAX Find the smallest index of the maximum component of a real vector. | Portable single precision Fortran subprogram in XBLAS sublibrary of CMLIB library. | Class(es): D1a2 | Usage: i=ISMAX(N,SX,INCX) | On-line doc: CALL GAMSDOC ISMAX (or @PRT CMLIB+DOC.ISMAX/XBLAS) | Tests: CMLIB+TEST-SOURCE.$Q/XBLAS | Access: LIB NBS+CMLIB

ISTKGT Allocates (gets) an array from the storage stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I=ISTKGT (NITEMS,ITYPE) | On-line doc: CALL GAMSDOC ISTKGT (or @PRT PORT+DOC.ISTKGT) | Access: LIB NBS+PORT

ISTKIN Initialize the length of the dynamic storage stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: CALL ISTKIN (NITEMS,ITYPE) | On-line doc: CALL GAMSDOC ISTKIN (or @PRT PORT+DOC.ISTKIN) | Access: LIB NBS+PORT

ISTKMD Changes size of last stack allocation for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I=ISTKMD (NITEMS) | On-line doc: CALL GAMSDOC ISTKMD (or @PRT PORT+DOC.ISTKMD) | Access: LIB NBS+PORT

ISTKQU Returns the number of available items that remain in the stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I=ISTKQU (ITYPE) | On-line doc: CALL GAMSDOC ISTKQU (or @PRT PORT+DOC.ISTKQU) | Access: LIB NBS+PORT

ISTKRL Releases the last stack allocations requested for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: CALL ISTKRL (NUMBER) | On-line doc: CALL GAMSDOC ISTKRL (or @PRT PORT+DOC.ISTKRL) | Access: LIB NBS+PORT

ISTKST Returns information on the status of the stack for PORT library programs. | Proprietary mixed precision Fortran subprogram in PORT library. | Class(es): N4 | Usage: I=ISTKST (NFACT) | On-line doc: CALL GAMSDOC ISTKST (or @PRT PORT+DOC.ISTKST) | Access: LIB NBS+PORT

IUMB Given interval endpoints, this generates a uniform mesh for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDUMB. | Class(es): E3 K6 | Usage: I=IUMB (A,B,NAB,K,NX) | On-line doc: CALL GAMSDOC IUMB (or @PRT PORT+DOC.IUMB) | Access: LIB NBS+PORT

IUMD Given interval endpoints, this generates a uniform mesh. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is IDUMD. | Class(es): E3 K6 | Usage: I=IUMD (A,B,NAB) | On-line doc: CALL GAMSDOC IUMD (or @PRT PORT+DOC.IUMD) | Access: LIB NBS+PORT

JOIN Merges constants and/or vectors into vectors. Command in MINITAB Proprietary interactive system. Class(es): L2 Usage: JOIN E to the bottom of E [to the bottom of E,..., to E] put into C On-line doc: HELP JOIN (in Minitab) Tests: MINITAB+TEST-SOURCE Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

KRUSKAL-WALLIS Perform Kruskal-Wallis test, based on ranks, of the null hypothesis that there is no difference among K population locations against the alternative of at least one difference. (This is a K-sample generalization of the Mann-Whitney-Wilcoxon test and is a nonparametric alternative to one-way ANOVA.). | Command in MINITAB Proprietary interactive system. Class(es): L4b1b | Usage: KRUSkal-Wallis test for data in C, subscripts in C | On-line doc: HELP KRUSKAL-WALLIS (in Minitab) | Tests: MINITAB*TESTSOURCE. | Access: ®XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
**L2SFF** Obtains a weighted least square expansion of a known function in terms of B-splines of order \( K \), at given mesh points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DL2SFF. | Class(es): L2SFF | Usage: L2SFF (FW,K,T,NT,A) | On-line doc: CALL GAMSDOC L2SFF (or @PRT PORT+DOC.L2SFF) | Access: LIB NBS+PORT | See also: SPLNE SPLND SPLNI SPLN2 EEBSF EEBSI EESFF EESF1

**L2SFH** Obtains a weighted least square expansion of a known function in and its derivatives in terms of B-splines of order \( K \) at given mesh points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DL2SFH. | Class(es): L2SFH | Usage: L2SFH (FW,MD,K,T,NT,A) | On-line doc: CALL GAMSDOC L2SFH (or @PRT PORT+DOC.L2SFH) | Access: LIB NBS+PORT | See also: SPLNE SPLND SPLNI SPLN2


**LAMCDF** Computes the cumulative distribution function value for the (Tukey) lambda distribution with tail length parameter = ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1l | Usage: CALL LAMCDF(X,ALAMBA,CDF) | On-line doc: CALL GAMSDOC LAMCDF (or @PRT DATAPAC+DOC.LAMCDF) | Access: LIB NBS+DATAPAC

**LAMPDF** Computes the probability density function value for the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1l | Usage: CALL LAMPDF(X,ALAMBA,PDF) | On-line doc: CALL GAMSDOC LAMPDF (or @PRT DATAPAC+DOC.LAMPDF) | Access: LIB NBS+DATAPAC

**LAMPLT** Generates a (Tukey) lambda distribution probability plot with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3e4l | Usage: CALL LAMPLT(X,N,ALAMBA) | On-line doc: CALL GAMSDOC LAMPLT (or @PRT DATAPAC+DOC.LAMPLT) | Access: LIB NBS+DATAPAC

**LAMPPF** Computes the percent point function value for the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2l | Usage: CALL LAMPPF(P,ALAMBA,PFF) | On-line doc: CALL GAMSDOC LAMPPF (or @PRT DATAPAC+DOC.LAMPPF) | Access: LIB NBS+DATAPAC

**LAMRAN** Generates a random sample of size \( N \) from the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a12 | Usage: CALL LAMRAN(N,ALAMBA,ISTART,X) | On-line doc: CALL GAMSDOC LAMRAN (or @PRT DATAPAC+DOC.LAMRAN) | Access: LIB NBS+DATAPAC

**LAMSF** Computes the sarspy function value for the (Tukey) lambda distribution with tail length parameter ALAMBA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2l | Usage: CALL LAMSF(P,ALAMBA,SF) | On-line doc: CALL GAMSDOC LAMSF (or @PRT DATAPAC+DOC.LAMSF) | Access: LIB NBS+DATAPAC

**LEAVE** Restores prior error recovery mode and reset the stack for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3c | Usage: CALL LEAVE | On-line doc: CALL GAMSDOC LEAVE (or @PRT PORT+DOC.LEAVE) | Access: LIB NBS+PORT

**LEQ1PB** Linear equation solution - positive definite symmetric band matrix - band symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LEQ1PB (A,N,NC,IA,B,IB,M,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LEQ1PB (or @PRT IMSL+DOC.LEQ1PB) | Access: LIB NBS+IMSL

**LEQ1S** Linear equation solution - indefinite matrix - symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1a | Usage: CALL LEQ1S (A,N,B,M,IB,JOBI,ICHNG,DET,IER) | On-line doc: CALL GAMSDOC LEQ1S (or @PRT IMSL+DOC.LEQ1S) | Access: LIB NBS+IMSL

**LEQ2C** Linear equation solution - complex matrix high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2c1 | Usage: CALL LEQ2C (A,N,IA,B,M,IB,JOBI,WA,WK,IER) | On-line doc: CALL GAMSDOC LEQ2C (or @PRT IMSL+DOC.LEQ2C) | Access: LIB NBS+IMSL

**LEQ2PB** Linear equation solution - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LEQ2PB (A,N,NC,IA,B,IB,M,IDGT,D1,D2,WK,IER) | On-line doc: CALL GAMSDOC LEQ2PB (or @PRT IMSL+DOC.LEQ2PB) | Access: LIB NBS+IMSL


GAMS: Module Dictionary

January 1984

@PRT IMSL+DOC.LEQOF | Access: LIB NBS+IMSL

LEQT1B Linear equation solution - band storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a2 | Usage: CALL LEQT1B (A,N,NLC,NUC,IA,B,M,IB,JOB,XL,IER) | On-line doc: CALL GAMSDOC LEQT1B (or @PRT IMSL + DOC.LEQQT1B) | Access: LIB NBS+IMSL

LEQT1C Matrix decomposition, linear equation solution - space economizer solution complex matrices. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2c1 | Usage: CALL LEQT1C (A,N,IA,B,M,IB,JOB,WA,IER) | On-line doc: CALL GAMSDOC LEQT1C (or @PRT IMSL + DOC.LEQQT1C) | Access: LIB NBS+IMSL

LEQT1F Linear equation solution - full storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LEQT1F (A,M,N,IA,B,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LEQT1F (or @PRT IMSL + DOC.LEQQT1F) | Access: LIB NBS+IMSL

LEQT1P Linear equation solution - positive definite matrix - symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1 | Usage: CALL LEQT1P (A,N,IA,B,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LEQT1P (or @PRT IMSL + DOC.LEQQT1P) | Access: LIB NBS+IMSL


LGNCDF Computes the cumulative distribution function value for the lognormal distribution with mean sqrt(e). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a11 | Usage: CALL LGNCDF(X,CDF) | On-line doc: CALL GAMSDOC LGNCDF (or @PRT DATAPAC + DOC.LGNCDF) | Access: LIB NBS+DATAPAC

LGNPLT Generates a lognormal probability plot with mean = sqrt(e). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5c41 | Usage: CALL LGNPLT(X,N) | On-line doc: CALL GAMSDOC LGNPLT (or @PRT DATAPAC + DOC.LGNPLT) | Access: LIB NBS+DATAPAC

LGNNPF Computes the percent point function value for the lognormal distribution with mean = sqrt(e). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a21 | Usage: CALL LGNNPF(P,PPF) | On-line doc: CALL GAMSDOC LGNNPF (or @PRT DATAPAC + DOC.LGNPPF) | Access: LIB NBS+DATAPAC

LGNRAN Generates a random sample of size N from the lognormal distribution with mean = sqrt(e). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a12 | Usage: CALL LGNRAN(N,START,X) | On-line doc: CALL GAMSDOC LGNRAN (or @PRT DATAPAC + DOC.LGNRAN) | Access: LIB NBS+DATAPAC

LIN1PB Inversion of a matrix - positive definite band symmetric matrix - band symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LIN1PB (A,N,NC,IA,AINV,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LIN1PB (or @PRT IMSL + DOC.LIN1PB) | Access: LIB NBS+IMSL

LIN2PB Inversion of a matrix - positive definite band symmetric matrix - band symmetric storage mode - high accuracy solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b2 | Usage: CALL LIN2PB (A,N,NC,IA,AINV,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LIN2PB (or @PRT IMSL + DOC.LIN2PB) | Access: LIB NBS+IMSL

LINEQ Solves a real system of linear equations, AX = B, where B is allowed to be a matrix or a vector. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D2a1 | Usage: LINEQ (N, A, B, NB, X) | On-line doc: CALL GAMSDOC LINEQ (or @PRT PORT + DOC.LINEQ) | Access: LIB NBS+PORT

LINV1F Inversion of a matrix - full storage mode space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2a1 | Usage: CALL LINV1F (A,N,IA,AINV,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV1F (or @PRT IMSL + DOC.LINV1F) | Access: LIB NBS+IMSL

LINV1P Inversion of a matrix - positive definite symmetric storage mode - space economizer solution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D2b1 | Usage: CALL LINV1P (A,N,AINV,IDGT,D1,D2,IER) | On-line doc: CALL GAMSDOC LINV1P (or @PRT IMSL + DOC.LINV1P) | Access: LIB NBS+IMSL
LINV2F  Inversion of a matrix - full storage mode high accuracy solution.  |  Proprietary single precision Fortran subprogram in IMSL library.  
| Class(es): D2a1 | Usage: CALL LINV2F (A,N,IA,AINV,IDGT,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV2F (or @PRT IMSL+DOC.LINV2F) | Access: LIB NBS+IMSL

LINV2P  Inversion of a matrix - positive definite symmetric storage mode - high accuracy solution.  |  Proprietary single precision Fortran subprogram in IMSL library.  
| Class(es): D2b1b | Usage: CALL LINV2P (A,N,AINV,IDGT,D1,D2,WKAREA,IER) | On-line doc: CALL GAMSDOC LINV2P (or @PRT IMSL+DOC.LINV2P) | Access: LIB NBS+IMSL

LINVF  In place inverse, equation solution, and/or determinant evaluation - full storage mode.  |  Proprietary single precision Fortran subprogram in IMSL library.  
| Class(es): D2a1 D3a1 | Usage: CALL LINVF (A,B,JJOB,N,IA,D1,D2,WKAREA,IER) | On-line doc: CALL GAMSDOC LINVF (or @PRT IMSL+DOC.LINVF) | Access: LIB NBS+IMSL

LINV3F  In place inverse, equation solution, positive definite matrix - symmetric storage mode.  |  Proprietary single precision Fortran subprogram in IMSL library.  
| Class(es): D2b1 | Usage: CALL LINV3F (A,B,JJOB,N,IER) | On-line doc: CALL GAMSDOC LINV3F (or @PRT IMSL+DOC.LINV3F) | Access: LIB NBS+IMSL

LLBQF  Solution of linear least squares problem high accuracy solution.  |  Proprietary single precision Fortran subprogram in IMSL library.  

LLSIA  Computes least squares solution to AX=B with A an m by n matrix with m>g=n.  Flexible version of SGLS.  
| Portable single precision Fortran subprogram in SGLS subroutine of CMLIB library.  

LLSQF  Solution of a linear least squares problem.  |  Proprietary single precision Fortran subprogram in IMSL library.  
| Class(es): D0 | Usage: CALL LLSQF (A,IA,M,N,B,TOL,KBASIS,X,H,IP,IER) | On-line doc: CALL GAMSDOC LLSQF (or @PRT IMSL+DOC.LLSQF) | Access: LIB NBS+IMSL

LOC  Computes 4 estimates (midrange, mean, midmean, and median) of the data in the input vector X.  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L1a1 | Usage: CALL LOC(X,N) | On-line doc: CALL GAMSDOC LOC (or @PRT DATAPAC+DOC.LOC) | Access: LIB NBS+DATAPAC

LOGCDF  Computes the cumulative distribution function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L5a11 | Usage: CALL LOGCDF(X,CDF) | On-line doc: CALL GAMSDOC LOGCDF (or @PRT DATAPAC+DOC.LOGCDF) | Access: LIB NBS+DATAPAC

LOGPDF  Computes the probability density function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L5a11 | Usage: CALL LOGPDF(X,PDF) | On-line doc: CALL GAMSDOC LOGPDF (or @PRT DATAPAC+DOC.LOGPDF) | Access: LIB NBS+DATAPAC

LOGPLT  Generates a logistic probability plot with mean = 0 and standard deviation = pi/sqrt(3).  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L3c4 | Usage: CALL LOGPLT(X,N) | On-line doc: CALL GAMSDOC LOGPLT (or @PRT DATAPAC+DOC.LOGPLT) | Access: LIB NBS+DATAPAC

LOGPFF  Computes the percent point function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L5a2 | Usage: CALL LOGPFF(P,PFF) | On-line doc: CALL GAMSDOC LOGPFF (or @PRT DATAPAC+DOC.LOGPFF) | Access: LIB NBS+DATAPAC

LOGRAN  Generates a random sample of size N from the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L8a12 | Usage: CALL LOGRAN(N,START,X) | On-line doc: CALL GAMSDOC LOGRAN (or @PRT DATAPAC+DOC.LOGRAN) | Access: LIB NBS+DATAPAC

LOGSF  Computes the sparsity function value for the logistic distribution with mean = 0 and standard deviation = pi/sqrt(3).  
| Portable single precision Fortran subprogram in DATAPAC library.  
| Class(es): L5a2 | Usage: CALL LOGSF(P,PSF) | On-line doc: CALL GAMSDOC LOGSF (or @PRT DATAPAC+DOC.LOGSF) | Access: LIB NBS+DATAPAC

LOTPS  Passes smooth function thru points (X[I],Y[I]),F(I),I=1..NPI and returns an array of interpolated values on user specified grid.  
| Portable single precision Fortran subprogram in LOTPS subroutine of CMLIB library.  
| Class(es): E2b | Usage: CALL LOTPS(MODE,NPPR,NLXI,YI,FI,NX,XO,NY,YI,JW,NIW,NIKWU,WK,NFK,FOK) | On-line doc: CALL GAMSDOC LOTPS (or @PRT CMLIB+DOC.LOTPS) | Access: LIB NBS+CMLIB

LPDP  Solves least projected distance problem.  
| Portable single precision Fortran subprogram in FC subroutine of CMLIB library.  
| Class(es): K1a2a | Usage: CALL LPDP(A,MDA,M,N1,N2,PRGPOPT,X,WNORM,MODE,WS,IS) | On-line doc: CALL GAMSDOC LPDP (or @PRT CMLIB+DOC.LPDP/FC) | Access: LIB NBS+CMLIB

LPLOT  Prints a letter plot with symbols corresponding to numerical "tag" values. Scale specification is optional.  
| Command in MINITAB Proprietary interactive system.  
| Class(es): L3c2 Q1 | Usage: LPLOT C [from K to K] vs C [from K to K] using tags in C | On-line
January 1984
GAMS: Module Dictionary

C 115

doc: HELP LPOINT (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

LSEI Solves linearly constrained least squares problem with equality and inequality constraints. Covariance matrix opt. output.


LSTSQ Finds the least squares solution of a system of linear equations, AX=B. B may be a matrix. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DLSTSQ. | Class(es): D9 | Usage: CALL LSTSQ(MDM,NDIM,M,N,A,B,NB,X) | On-line doc: CALL GAMSDOC LSTSQ (or @PRT PORT+DOCLSTSQ) | Access: LIB NBS+PORT


LUDAPB Decomposition of a positive definite band symmetric matrix - band symmetric storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b2 | Usage: CALL LUDAPB(A,N,NC,IA,UL,LU,D1,D2,IER) | On-line doc: CALL GAMSDOC LUDAPB (or @PRT IML+DOC.LUDAPB) | Access: LIB NBS+IMSL

LUDATP L-U decomposition by the Crout algorithm with optional accuracy test. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2a1 | Usage: CALL LUDATP(A,LU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER) | On-line doc: CALL GAMSDOC LUDATP (or @PRT IML+DOC.LUDATP) | Access: LIB NBS+IMSL

LUDECP Decomposition of a positive definite matrix symmetric storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b1b | Usage: CALL LUDECP(A,UL,N,DI,D2,IER) | On-line doc: CALL GAMSDOC LUDECP (or @PRT IML+DOC.LUDECP) | Access: LIB NBS+IMSL

LUELMP Elimination part of solution of Ax=b (full storage mode). | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b1b | Usage: CALL LUELMP(A,B,IPVT,N,IA,X) | On-line doc: CALL GAMSDOC LUELMP (or @PRT IML+DOC.LUELMP) | Access: LIB NBS+IMSL | See also: LUDATF

LUELMP Elimination part of the solution of Ax=b - positive definite matrix - symmetric storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b1b | Usage: CALL LUELMP(A,B,N,X) | On-line doc: CALL GAMSDOC LUELMP (or @PRT IML+DOC.LUELMP) | Access: LIB NBS+IMSL | See also: LUDECP

LUELMP Elimination part of solution of Ax=b positive definite band symmetric matrix band symmetric storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b2 | Usage: CALL LUELMP(UL,B,N,NC,IA,X) | On-line doc: CALL GAMSDOC LUELMP (or @PRT IML+DOC.LUELMP) | Access: LIB NBS+IMSL | See also: LUDATF

LUMB Given a basic mesh, this subroutines each interval uniformly for B-spline. Multiplicities are allowed. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DLUMB. | Class(es): E3 K6 | Usage: CALL LUMB(XB,NXB,N,K,X,NX) | On-line doc: CALL GAMSDOC LUMB (or @PRT PORT+DOC.LUMB) | Access: LIB NBS+PORT

LUMD Given a basic mesh, this subroutines each interval into the same number of uniformly spaced points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DLUMD. | Class(es): E3 K6 | Usage: CALL LUMD(XB,NXB,N,K,N,NN) | On-line doc: CALL GAMSDOC LUMD (or @PRT PORT+DOC.LUMD) | Access: LIB NBS+PORT

LUREFF Refinement of solution to linear equations full storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2a1 | Usage: CALL LUREFF(A,B,UL,IPVT,N,IA,X,IDGT,RES,DX,IER) | On-line doc: CALL GAMSDOC LUREFF (or @PRT IML+DOC.LUREFF) | Access: LIB NBS+IMSL | See also: LUDATF LUELMP

LUREFF Refinement of solution to linear equations positive definite matrix - symmetric storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b1b | Usage: CALL LUREFF(A,B,UL,N,IDGT,RES,IER) | On-line doc: CALL GAMSDOC LUREFF (or @PRT IML+DOC.LUREFF) | Access: LIB NBS+IMSL | See also: LUDECP LUELMP

LUREPB Refinement of solution to linear equations positive definite band symmetric matrix band symmetric storage mode. | Proprietary single precision Fortran subprogram in IML library. | Class(es): D2b2 | Usage: CALL LUREPB(A,N,NC,IA,UL,LU,B,IDGT,RES,IER) | On-line doc: CALL GAMSDOC LUREPB (or @PRT IML+DOC.LUREPB) | Access: LIB NBS+IMSL | See also: LUDATF LUELMP

LVALS Prints letter-value display - median, hinges, etc., and optionally saves results. | Command in MINITAB Proprietary interactive system. | Class(es): L3d | Usage: LVALS display of C [put letter values into C [mids into C [spreads into C]] | On-line doc: HELP LVALS (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)
M01AAE  Passively sort a real vector into ascending order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AEP.  Class(es): N6a1b1  Usage: CALL M01AAE (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01AAE (or @PRT NAG*DOC.M01AAE) | Access: LIB NBS*NAG

M01AAF  Passively sort a real vector into ascending order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AED.  Class(es): N6a1b2  Usage: CALL M01AAF (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01AAF (or @PRT NAG*DOC.M01AED) | Access: LIB NBS*NAG

M01ABE  Passively sort a real vector into descending order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ABF.  Class(es): N6a1b1  Usage: CALL M01ABE (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ABE (or @PRT NAG*DOC.M01AEB) | Access: LIB NBS*NAG

M01ABF  Passively sort a real vector into descending order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ABE.  Class(es): N6a1b2  Usage: CALL M01ABF (A, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ABF (or @PRT NAG*DOC.M01AEB) | Access: LIB NBS*NAG

M01ACE  Passively sort an integer vector into ascending order. (Identical to M01ACF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ACF.  Class(es): N8a1a  Usage: CALL M01ACE (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ACE (or @PRT NAG*DOC.M01ACF) | Access: LIB NBS*NAG

M01ACF  Passively sort an integer vector into ascending order. (Identical to M01ACE.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ACF.  Class(es): N8a1a  Usage: CALL M01ACF (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ACF (or @PRT NAG*DOC.M01ACF) | Access: LIB NBS*NAG

M01ADE  Passively sort an integer vector into descending order. (Identical to M01ADF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ADF.  Class(es): N8a1a  Usage: CALL M01ADE (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ADE (or @PRT NAG*DOC.M01AED) | Access: LIB NBS*NAG

M01ADF  Passively sort an integer vector into descending order. (Identical to M01ADE.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ADF.  Class(es): N8a1a  Usage: CALL M01ADF (IA, M, N, IP, IST, IFAIL) | On-line doc: CALL GAMSDOC M01ADF (or @PRT NAG*DOC.M01AED) | Access: LIB NBS*NAG

M01AEF  Actively sort the rows of a real matrix into ascending order of an index column. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AEF.  Class(es): N6a2b1  Usage: CALL M01AEF (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AEF (or @PRT NAG*DOC.M01AEP) | Access: LIB NBS*NAG

M01AEP  Actively sort the rows of a real matrix into ascending order of an index column. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AEF.  Class(es): N6a2b2  Usage: CALL M01AEP (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AEP (or @PRT NAG*DOC.M01AEP) | Access: LIB NBS*NAG

M01AEE  Actively sort the rows of a real matrix into ascending order of an index column. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AEE.  Class(es): N6a2b1  Usage: CALL M01AEE (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AEE (or @PRT NAG*DOC.M01AED) | Access: LIB NBS*NAG

M01AED  Actively sort the rows of a real matrix into ascending order of an index column. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AEE.  Class(es): N6a2b2  Usage: CALL M01AED (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AED (or @PRT NAG*DOC.M01AED) | Access: LIB NBS*NAG

M01AFF  Actively sort the rows of a real matrix into descending order of an index column. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AFF.  Class(es): N6a2b1  Usage: CALL M01AFF (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AFF (or @PRT NAG*DOC.M01AFF) | Access: LIB NBS*NAG

M01AEG  Actively sort the rows of a real matrix into descending order of an index column. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AEG.  Class(es): N6a2b2  Usage: CALL M01AEG (A, NR, NC, IC, T, TT, IFAIL) | On-line doc: CALL GAMSDOC M01AEG (or @PRT NAG*DOC.M01AEG) | Access: LIB NBS*NAG

M01AGF  Actively sort the rows of a real matrix into ascending order of an index column. (Identical to M01AEG.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AEG.  Class(es): N6a2a  Usage: CALL M01AGF (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AGF (or @PRT NAG*DOC.M01AGF) | Access: LIB NBS*NAG

M01AGF  Actively sort the rows of a real matrix into descending order of an index column. (Identical to M01AEG.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AEG.  Class(es): N6a2a  Usage: CALL M01AGF (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AGF (or @PRT NAG*DOC.M01AGF) | Access: LIB NBS*NAG

M01AHE  Actively sort the rows of an integer matrix into ascending order of an index column. (Identical to M01AHE.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AHE.  Class(es): N6a2a  Usage: CALL M01AHE (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AHE (or @PRT NAG*DOC.M01AHE) | Access: LIB NBS*NAG

M01AHF  Actively sort the rows of an integer matrix into descending order of an index column. (Identical to M01AHE.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AHE.  Class(es): N6a2a  Usage: CALL M01AHF (IA, NR, NC, IC, K, L, IFAIL) | On-line doc: CALL GAMSDOC M01AHF (or @PRT NAG*DOC.M01AHF) | Access: LIB NBS*NAG

M01AJE  Actively sort a real vector into ascending order and provide an index to the original order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AJF.  Class(es): N6a2b1  Usage: CALL M01AJE (A, W, IND, INDW, N,
M01AJF  Actively sort a real vector into ascending order and provide an index to the original order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AJE. | Class(es): N6a2b2 | Usage: CALL M01AJF (A, W, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AJF (or ®PRT NAG+DOC.M01AJF) | Access: LIB NBS+NAG

M01AKE  Actively sort a real vector into descending order and provide an index to the original order. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AKF. | Class(es): N6a2b1 | Usage: CALL M01AKE (A, W, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AKE (or ®PRT NAG+DOC.M01AKE) | Access: LIB NBS+NAG

M01AKF  Actively sort a real vector into descending order and provide an index to the original order. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AKE. | Class(es): N6a2b2 | Usage: CALL M01AKF (A, W, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AKF (or ®PRT NAG+DOC.M01AKF) | Access: LIB NBS+NAG

M01ALE  Actively sort an integer vector into ascending order of an index column and provide an index to the original order. (Identical to M01ALF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ALF. | Class(es): N6a2a | Usage: CALL M01ALE (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01ALE (or ®PRT NAG+DOC.M01ALE) | Access: LIB NBS+NAG

M01ALF  Actively sort an integer vector into ascending order of an index column and provide an index to the original order. (Identical to M01ALE.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ALE. | Class(es): N6a2a | Usage: CALL M01ALF (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01ALF (or ®PRT NAG+DOC.M01ALF) | Access: LIB NBS+NAG

M01AME  Actively sort an integer vector into descending order of an index column and provide an index to the original order. (Identical to M01AMF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AMF. | Class(es): N6a2a | Usage: CALL M01AME (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AME (or ®PRT NAG+DOC.M01AME) | Access: LIB NBS+NAG

M01AMF  Actively sort an integer vector into descending order of an index column and provide an index to the original order. (Identical to M01AME.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AMF. | Class(es): N6a2a | Usage: CALL M01AMF (IA, IW, IND, INDW, N, NW, IFAIL) | On-line doc: CALL GAMSDOC M01AMF (or ®PRT NAG+DOC.M01AMF) | Access: LIB NBS+NAG

M01ANE  Actively sort a real vector into ascending order (Singleton’s implementation of Quicksort). | Proprietary single precision Fortran subprogram in NAG library. Single precision version is M01ANF. | Class(es): N6a2b1 | Usage: CALL M01ANE (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ANE (or ®PRT NAG+DOC.M01ANE) | Access: LIB NBS+NAG

M01ANF  Actively sort a real vector into ascending order (Singleton’s implementation of Quicksort). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ANF. | Class(es): N6a2b2 | Usage: CALL M01ANF (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ANF (or ®PRT NAG+DOC.M01ANF) | Access: LIB NBS+NAG

M01APE  Actively sort a real vector into descending order (Singleton’s implementation of Quicksort). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01APF. | Class(es): N6a2b2 | Usage: CALL M01APE (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01APE (or ®PRT NAG+DOC.M01APE) | Access: LIB NBS+NAG

M01APF  Actively sort a real vector into descending order (Singleton’s implementation of Quicksort). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01APE. | Class(es): N6a2b2 | Usage: CALL M01APF (A, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01APF (or ®PRT NAG+DOC.M01APF) | Access: LIB NBS+NAG

M01AQE  Actively sort an integer vector into ascending order (Singleton’s implementation of Quicksort). (Identical to M01AQF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01AQF. | Class(es): N6a2a | Usage: CALL M01AQE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01AQE (or ®PRT NAG+DOC.M01AQE) | Access: LIB NBS+NAG

M01AQF  Actively sort an integer vector into ascending order (Singleton’s implementation of Quicksort). (Identical to M01AQE.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01AQE. | Class(es): N6a2a | Usage: CALL M01AQF (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01AQF (or ®PRT NAG+DOC.M01AQF) | Access: LIB NBS+NAG

M01ARE  Actively sort an integer vector into descending order (Singleton’s implementation of Quicksort). (Identical to M01ARF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01ARF. | Class(es): N6a2a | Usage: CALL M01ARE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ARE (or ®PRT NAG+DOC.M01ARE) | Access: LIB NBS+NAG

M01ARF  Actively sort an integer vector into descending order (Singleton’s implementation of Quicksort). (Identical to M01ARE.) | Proprietary double precision Fortran subprogram in NAG library. Single precision version is M01ARE. | Class(es): N6a2a | Usage: CALL M01ARF (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01ARF (or ®PRT NAG+DOC.M01ARF) | Access: LIB NBS+NAG

M01BAE  Actively sort a character vector into reverse alphanumeric order (Singleton’s implementation of Quicksort). (Identical to M01BAF.) | Proprietary single precision Fortran subprogram in NAG library. Double precision version is M01BAF. | Class(es): N6a2c | Usage: CALL M01BAE (IA, I, J, IFAIL) | On-line doc: CALL GAMSDOC M01BAE (or ®PRT NAG+DOC.M01BAE) | Access: LIB NBS+NAG

M01BAF  Actively sort a character vector in reverse alphanumeric order (Singleton’s implementation of Quicksort). (Identical to M01BAE.).
M01BBE Actively sort a character vector into alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BBF).

M01BBF Actively sort a character vector in alphanumeric order (Singleton's implementation of Quicksort). (Identical to M01BBE).

M01BCE Actively sort the columns of a character matrix into reverse alphanumeric order of an index column. (Identical to M01BFC).

M01BCF Actively sort the columns of a character matrix into reverse alphanumeric order of an index column. (Identical to M01BCE).

M01BDE Actively sort the columns of a character matrix into alphanumeric order of an index column. (Identical to M01BDF).

M01BDF Actively sort the columns of a character matrix into alphanumeric order of an index column. (Identical to M01BDE).

MANN-WHITNEY Performs one- or two-sided two-sample rank test (a.k.a. Wilcoxon rank test) for the difference between two population medians, and calculates the corresponding point and confidence interval estimates. | Command in MINITAB Proprietary interactive system. Class(es): L6b1b | Usage: MANN-WHITNEY [alternative K], [percent confidence K] for data in C and C | On-line doc: HELP MANN-WHITNEY (in MiniTab) Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

MATLAB An interactive system for defining and manipulating matrices. It includes solving linear systems, linear least squares, eigenvalue and eigenvector calculation, QR decomposition, singular value decomposition and inverses, as well as other interesting and useful features. | Portable mixed precision Fortran subprogram in MATLAB library. Class(es): D | On-line doc: HELP MATLAB*DOC.SUMMARY | Access: @XQT NBS*MATLAB.MATLAB (or @XQT NBS*MATLAB.MATLAB)

MAX Computes the sample maximum of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): D1a2 | Usage: CALL MAX(X,N,IWRITE,XMAX) | On-line doc: CALL GAMSDOC MAX (or @PRT DATAPAC*DOC.MAX) | Access: LIB NBS*DATAPAC

MDBETA Beta probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1b | Usage: CALL MDBETA (X,A,B,P,IER) | On-line doc: CALL GAMSDOC MDBETA (or @PRT IMSL*DOC.MDBETA) | Access: LIB NBS*IMSL


MDBNOR Bivariate normal probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5b1b | Usage: CALL MDBNOR (X,Y,RHO,P,IER) | On-line doc: CALL GAMSDOC MDBNOR (or @PRT IMSL*DOC.MDBNOR) | Access: LIB NBS*IMSL

MDCH Chi-squared probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1c | Usage: CALL MDCH (CS,DF,P,IER) | On-line doc: CALL GAMSDOC MDCH (or @PRT IMSL*DOC.MDCH) | Access: LIB NBS*IMSL


MDFD F probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1f | Usage:
CALL MDFD (F,N1,N2,P,IER) | On-line doc: CALL GAMSDOC MDFD (or @PRT IMSL*DOC.MDFD) | Access: LIB NBS*IMSL

MDFDRE F probability distribution function (integer or fractional degrees of freedom). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1f | Usage: CALL MDFDRE (X,DFN,DFD,P,IER) | On-line doc: CALL GAMSDOC MDFDRE (or @PRT IMSL*DOC.MDFDRE) | Access: LIB NBS*IMSL

MDFI Inverse F probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2f | Usage: CALL MDFI (P,D1,D2,X,IER) | On-line doc: CALL GAMSDOC MDFI (or @PRT IMSL*DOC.MDFI) | Access: LIB NBS*IMSL

MDGAM Gamma probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1g | Usage: CALL MDGAM (X,P,PROB,IER) | On-line doc: CALL GAMSDOC MDGAM (or @PRT IMSL*DOC.MDGAM) | Access: LIB NBS*IMSL

MDGCI General cumulative probability distribution function, given ordinates of the density. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1h | Usage: CALL MDGCI (X,F,M,1OPT,B,C,JG,P,IER) | On-line doc: CALL GAMSDOC MDGCI (or @PRT IMSL*DOC.MDGCI) | Access: LIB NBS*IMSL

MDHYP Hypergeometric probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1j | Usage: CALL MDHYP (K,N,L,ND,PEQK,PLEK,IER) | On-line doc: CALL GAMSDOC MDHYP (or @PRT IMSL*DOC.MDHYP) | Access: LIB NBS*IMSL

MDNOR Normal or Gaussian probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1n C5a | Usage: CALL MDNOR (Y,P) | On-line doc: CALL GAMSDOC MDNOR (or @PRT IMSL*DOC.MDNOR) | Access: LIB NBS*IMSL

MDNRIS Inverse standard normal (Gaussian) probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2o | Usage: CALL MDNRIS (P,Y,IER) | On-line doc: CALL GAMSDOC MDNRIS (or @PRT IMSL*DOC.MDNRIS) | Access: LIB NBS*IMSL


MDSTI Inverse of a modification of Student's t probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2t | Usage: CALL MDSTI (Q,P,X,IER) | On-line doc: CALL GAMSDOC MDSTI (or @PRT IMSL*DOC.MDSTI) | Access: LIB NBS*IMSL

MPTD Students t probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1t | Usage: CALL MPTD (TVAL,DF, Q,IER) | On-line doc: CALL GAMSDOC MPTD (or @PRT IMSL*DOC.MPTD) | Access: LIB NBS*IMSL

MDTN Non-central t probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1t | Usage: CALL MDTN (TVAL,DF,D,P,IER) | On-line doc: CALL GAMSDOC MDTN (or @PRT IMSL*DOC.MDTN) | Access: LIB NBS*IMSL

MDTNF Integral related to calculation of noncentral t and bivariate normal probability distribution functions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1t L5b1n | Usage: CALL MDTNF (Y,ZEPS,T) | On-line doc: CALL GAMSDOC MDTNF (or @PRT IMSL*DOC.MDTNF) | Access: LIB NBS*IMSL

MDTPS Cumulative probability and, optionally, individual terms of the Poisson probability distribution function. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a1p | Usage: CALL MDTPS (K,RLAM,1OPT,T,P) | On-line doc: CALL GAMSDOC MDTPS (or @PRT IMSL*DOC.MDTPS) | Access: LIB NBS*IMSL

MEAN Computes the sample mean of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1a | Usage: CALL MEAN (X,N,1WRITE,XMEAN) | On-line doc: CALL GAMSDOC MEAN (or @PRT DATAPAC*DOC.MEAN) | Access: LIB NBS*DATAPAC

MEDIAN Computes the sample median of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1a | Usage: CALL MEDIAN (X,N,1WRITE,XMED) | On-line doc: CALL GAMSDOC MEDIAN (or @PRT DATAPAC*DOC.MEDIAN) | Access: LIB NBS*DATAPAC


MERRCZ Computes approximate values of $\exp(-z^2)\text{erfc}(iz)$ for complex $z$. | Proprietary double precision Fortran subprogram

January 1984  GAMS: Module Dictionary  C 119
in IML library. | Class(es): C1a | Usage: CALL MERRCZ(Z,W,IER) | On-line doc: CALL GAMSDOC MERRCZ (or @PRT IML*DOC.MERRCZ) | Access: LIB NBS*IMSL

**MIDM** Computes the sample midmean, i.e. the sample 25% (on each side) trimmed mean of the data in the input vector X. | Portable single precision Fortran program in DATAPAC library. | Class(es): L1a1 | Usage: CALL MIDM(X,N,IWRITE,XMIDM) | On-line doc: CALL GAMSDOC MIDM (or @PRT DATAPAC+DOC.MIDM) | Access: LIB NBS*DATAPAC

**MIN** Computes the sample minimum of the data in the input vector X. | Portable single precision Fortran program in DATAPAC library. | Class(es): Dia2 | Usage: CALL MIN(X,N,IWRITE,XMIN) | On-line doc: CALL GAMSDOC MIN (or @PRT DATAPAC+DOC.MIN) | Access: LIB NBS*DATAPAC


**MINITAB** Minibab's vector summarization commands include COUNT, SUM, MEAN, MAX, MIN, MEDIAN, N, NMISS (number of missing values), STDEV, SSQ (sum of squares), DESCRIBE (N, MEAN, MEDIAN, STDEV, MAX, MIN, 5% trimmed mean, quartiles) for columns or rows (use prefix R, e.g., RMEAN) of data in the Minibab worksheet. | Command in MINITAB Proprietary interactive system. Class(es): L1a1 L1a2 | On-line doc: HELP (in Minibab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

**MINITAB** Minibab's vector transformation commands include COPY (vectors, vectors to matrices, and conversely), DIAGONAL (create a diagonal matrix or extract the diagonal of a matrix), TRANSPOSE, INVERSE, and EIGEN (calculate eigenvalues and eigenvectors for a symmetric matrix). | Command in MINITAB Proprietary interactive system. Class(es): D | On-line doc: HELP (in Minibab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

**MINITAB** Minibab's vector transformation commands include SUBTRACT, MULTIPLY, DIVIDE, RAISE, SIN, COS, TAN, ASIN, ACOS, ATAN, LOGE, LOGTEN, EXPONENTIAL, ANTILOG, ABSOLUTE VALUE, ROUND, SIGNS, SQRT, INDICATOR, RECODE, SUBSTITUTE, CONVERT, PARSUM, PARPRODUCT, and LET (to combine commands). | Command in MINITAB Proprietary interactive system. Class(es): L2a | On-line doc: HELP (in Minibab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

**MMBS10** Modified Bessel function of the first kind of order zero. | Proprietary double precision Fortran program in IML library. | Class(es): C10b1 | Usage: D = MMBS10 (OPT, ARG, IER) | On-line doc: CALL GAMSDOC MMBS10 (or @PRT IML*DOC.MMBS10) | Access: LIB NBS*IMSL

**MMBS11** Modified Bessel function of the first kind of order one. | Proprietary double precision Fortran program in IML library. | Class(es): C10b1 | Usage: D = MMBS11 (OPT, ARG, IER) | On-line doc: CALL GAMSDOC MMBS11 (or @PRT IML*DOC.MMBS11) | Access: LIB NBS*IMSL

**MMBS1N** Computes approximate values of the modified Bessel function of the first kind of nonnegative integer order for real arguments. | Proprietary double precision Fortran program in IML library. | Class(es): C10b1 | Usage: CALL MMBS1N(ARG,N,B,IER) | On-line doc: CALL GAMSDOC MMBS1N (or @PRT IML*DOC.MMBS1N) | Access: LIB NBS*IMSL

**MMBSIR** Modified Bessel function of the first kind of nonnegative real order for real positive arguments with exponential scaling option. | Proprietary double precision Fortran program in IML library. | Class(es): C10b3 | Usage: CALL MMBSIR(ARG,ORDER,NB,OPT,B,IER) | On-line doc: CALL GAMSDOC MMBSIR (or @PRT IML*DOC.MMBSIR) | Access: LIB NBS*IMSL

**MMBSJ0** Bessel function of the first kind of order zero. | Proprietary double precision Fortran program in IML library. | Class(es): C10a1 | Usage: D = MMBSJ0 (ARG, IER) | On-line doc: CALL GAMSDOC MMBSJ0 (or @PRT IML*DOC.MMBSJ0) | Access: LIB NBS*IMSL

**MMBSJ1** Bessel function of the first kind of order one. | Proprietary double precision Fortran program in IML library. | Class(es): C10a1 | Usage: D = MMBSJ1 (ARG, IER) | On-line doc: CALL GAMSDOC MMBSJ1 (or @PRT IML*DOC.MMBSJ1) | Access: LIB NBS*IMSL

**MMBSJN** Bessel function of the first kind of nonnegative integer order for real arguments. | Proprietary double precision Fortran program in IML library. | Class(es): C10a1 | Usage: CALL MMBSJN(ARG,N,B,IER) | On-line doc: CALL GAMSDOC MMBSJN (or @PRT IML*DOC.MMBSJN) | Access: LIB NBS*IMSL
MMBSJR  Bessel function of the first kind of nonnegative real order for real positive arguments.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10a3  | Usage: CALL MMBSJR(ARG,ORDER,N,RJ,WK,IER)  | On-line doc: CALL GAMSDOC MMBSJR (or @PRT IMSL*DOC.MMBSJR)  | Access: LIB NBS*IMSL

MMBSKO  Modified Bessel function of the second kind of order zero.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10b1  | Usage: D = MMBSK0 (IOPT,ARG,IER)  | On-line doc: CALL GAMSDOC MMBSK0 (or @PRT IMSL*DOC.MMBSK0)  | Access: LIB NBS*IMSL

MMBSK1  Modified Bessel function of the second kind of order one.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10b1  | Usage: D = MMBSK1 (IOPT,ARG,IER)  | On-line doc: CALL GAMSDOC MMBSK1 (or @PRT IMSL*DOC.MMBSK1)  | Access: LIB NBS*IMSL

MMBSKR  Modified Bessel function of the second kind of nonnegative real fractional order for real positive arguments scaled by exp(arg).  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10b3  | Usage: CALL MMBSKR(ARG,ORDER,N,BK,IER)  | On-line doc: CALL GAMSDOC MMBSKR (or @PRT IMSL*DOC.MMBSKR)  | Access: LIB NBS*IMSL

MMBSYN  Bessel function of the second kind of nonnegative real fractional order for real positive arguments.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10a3  | Usage: CALL MMBSYN (ARG,ORDER,N,YN,IER)  | On-line doc: CALL GAMSDOC MMBSYN (or @PRT IMSL*DOC.MMBSYN)  | Access: LIB NBS*IMSL

MMBZIN  Modified Bessel function of the first kind of nonnegative integer order for complex arguments.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10b2  | Usage: CALL MMBZIN(X,Y,N,BR,BI,IER)  | On-line doc: CALL GAMSDOC MMBZIN (or @PRT IMSL*DOC.MMBZIN)  | Access: LIB NBS*IMSL

MMBZJN  Bessel function of the first kind of nonnegative integer order for complex arguments.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10a2  | Usage: CALL MMBZJN(X,Y,N,BR,BI,IER)  | On-line doc: CALL GAMSDOC MMBZJN (or @PRT IMSL*DOC.MMBZJN)  | Access: LIB NBS*IMSL

MMDAS  Dawson integral.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C8c  | Usage: D = MMDAS (ARG,IER)  | On-line doc: CALL GAMSDOC MMDAS (or @PRT IMSL*DOC.MMDAS)  | Access: LIB NBS*IMSL

MMDEI  Exponential integrals.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C5  | Usage: D = MMDEI (IOPT,ARG,IER)  | On-line doc: CALL GAMSDOC MMDEI (or @PRT IMSL*DOC.MMDEI)  | Access: LIB NBS*IMSL


MMDEN  Exponential integrals of integer order for real argument x scaled by exp(x).  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C5  | Usage: CALL MMDEN(X,N,F,IER)  | On-line doc: CALL GAMSDOC MMDEN (or @PRT IMSL*DOC.MMDEN)  | Access: LIB NBS*IMSL

MMKEL0  Kelvin functions of the first kind, (ber,bei), and of the second kind, (ker,kei) of order zero.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10c  | Usage: CALL MMKEL0 (X,BER,BEI,KER,KEI,IER)  | On-line doc: CALL GAMSDOC MMKEL0 (or @PRT IMSL*DOC.MMKEL0)  | Access: LIB NBS*IMSL

MMKEL1  Kelvin functions of the first kind, (ber,bei), and of the second kind, (ker,kei) of order one.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10c  | Usage: CALL MMKEL1 (X,BER1,BEI1,KER1,KEI1,IER)  | On-line doc: CALL GAMSDOC MMKEL1 (or @PRT IMSL*DOC.MMKEL1)  | Access: LIB NBS*IMSL

MMKELD  Derivatives of the Kelvin functions (ber,bei, ker, and kei) of order zero.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C10c  | Usage: CALL MMKELD (X,BERP,BEIP,KERP,KEIP,IER)  | On-line doc: CALL GAMSDOC MMKELD (or @PRT IMSL*DOC.MMKELD)  | Access: LIB NBS*IMSL

MMLINC  Computes an elementary integral from which inverse circular functions, logarithms or inverse hyperbolic functions may be computed.  | Proprietary double precision Fortran subprogram in IMSL library.  | Class(es): C4  | Usage: D = MMLINC(X,Y,IER)  | On-line doc: CALL GAMSDOC MMLINC (or @PRT IMSL*DOC.MMLINC)  | Access: LIB NBS*IMSL


| **NBS*IMSL** | **MMPSI** Logarithmic derivative of the gamma function. Proprietary double precision Fortran subprogram in IMSL library. **Class(es):** C7a **Usage:** D = MMPSI(A,IER) **On-line doc:** CALL GAMSDOC MMPSI (or @PRT IMSL*DOC.MMPSI) | **Access:** Lib NBS*IMSL |
| **MMWPL** Weierstrass $p$-function in the lemniscatic case for complex argument with unit period parallelogram. Proprietary double precision Fortran subprogram in IMSL library. **Class(es):** C15 **Usage:** CALL MMWPL(Z,PLEM,IER) **On-line doc:** CALL GAMSDOC MMWPL (or @PRT IMSL*DOC.MMWPL) | **Access:** Lib NBS*IMSL |
| **MMWPL1** First derivative of the Weierstrass $p$-function in the lemniscatic case for complex argument with unit period parallelogram. Proprietary double precision Fortran subprogram in IMSL library. **Class(es):** C15 **Usage:** CALL MMWPL1(Z,PLEM,IER) **On-line doc:** CALL GAMSDOC MMWPL1 (or @PRT IMSL*DOC.MMWPL1) | **Access:** Lib NBS*IMSL |
| **MMWPQ** Weierstrass $p$-function in the equianharmonic case for complex argument with unit period parallelogram. Proprietary double precision Fortran subprogram in IMSL library. **Class(es):** C15 **Usage:** CALL MMWPQ(Z,PEQ,IER) **On-line doc:** CALL GAMSDOC MMWPQ (or @PRT IMSL*DOC.MMWPQ) | **Access:** Lib NBS*IMSL |
| **MMWPQ1** First derivative of the Weierstrass $p$-function in the equianharmonic case for complex argument with unit period parallelogram. Proprietary double precision Fortran subprogram in IMSL library. **Class(es):** C15 **Usage:** CALL MMWPQ1(Z,PEQ,IER) **On-line doc:** CALL GAMSDOC MMWPQ1 (or @PRT IMSL*DOC.MMWPQ1) | **Access:** Lib NBS*IMSL |
| **MNPB** Creates a B-spline mesh from an array of fitting points, using at least n fitting points in each interval. Proprietary single precision Fortran subprogram in PORT library. Double precision version is MNPB2. **Class(es):** E3 K5 **Usage:** CALL MNPB(X,NX,N,K,T,NT) **On-line doc:** CALL GAMSDOC MNPB (or @PRT PORT*DOC.MNPB) | **Access:** Lib NBS*PORT |
| **MOLID** Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available. Portable single precision Fortran subprogram in PDELIB library. **Class(es):** 12a1 12a2 **Usage:** CALL MOLID(NPDE,NPTS,KEQN,KBC,METH,EPS,MORD,TINT,TLAST,MOUT,TOUT,UX,XM,KMOL) **On-line doc:** CALL GAMSDOC MOLID (or @PRT PDELIB*DOC.MOLID) | **Tests:** PDELIB*TEST.SOURCE.MOLID | **Access:** Lib NBS*PDELIB |
| **MONOD** Test if a double precision vector is monotone increasing or decreasing. Proprietary double precision Fortran subprogram in PORT library. Single precision version is MONOR. **Class(es):** R2 **Usage:** L = MONOR(X,N,INC) **On-line doc:** CALL GAMSDOC MONOD (or @PRT PORT*DOC.MONOD) | **Access:** Lib NBS*PORT |
| **MONO1** Test if an integer vector is monotone increasing or decreasing. Proprietary single precision Fortran subprogram in PORT library. **Class(es):** R2 **Usage:** L = MONO1(X,N,INC) **On-line doc:** CALL GAMSDOC MONO1 (or @PRT PORT*DOC.MONO1) | **Access:** Lib NBS*PORT |
| **MONOR** Test if a real vector is monotone increasing or decreasing. Proprietary single precision Fortran subprogram in PORT library. Double precision version is MONOD. **Class(es):** R2 **Usage:** L = MONOR(X,N,INC) **On-line doc:** CALL GAMSDOC MONOR (or @PRT PORT*DOC.MONOR) | **Access:** Lib NBS*PORT |
| **MOAVG** Computes a $k$-term symmetric moving average of a series. Portable single precision Fortran subprogram in STATLIB library. **Class(es):** L10b K5 **Usage:** CALL MOAVG(Y,N,K,H,YMA,YH,NYMA,YL,NYL) **On-line doc:** CALL GAMSDOC MOAVG (or @PRT STATLIB*DOC.MOAVG) | **Tests:** STATLIB*TEST.DEMO3 | **Access:** Lib NBS*STATLIB |
| **MOVE** Copies M elements of the vector $X$ (starting with position IX1) into the vector Y (starting with position IY1). Portable single precision Fortran subprogram in DATAPAC library. **Class(es):** D1a5 **Usage:** CALL MOVE(X,M,IX1,IY1,Y) **On-line doc:** CALL GAMSDOC MOVE (or @PRT DATAPAC*DOC.MOVE) | **Access:** Lib NBS*DATAPAC |
| **MOVEBC** Move a complex vector using backward DO loop. Proprietary single precision Fortran subprogram in PORT library. **Class(es):** D1a5 **Usage:** CALL MOVEBC(N,A,B) **On-line doc:** CALL GAMSDOC MOVEBC (or @PRT PORT*DOC.MOVEBC) | **Access:** Lib NBS*PORT |
| **MOVEBD** Move a double precision vector using backward DO loop. Proprietary double precision Fortran subprogram in PORT library. Single precision version is MOVEBR. **Class(es):** D1a5 **Usage:** CALL MOVEBD(N,A,B) **On-line doc:** CALL GAMSDOC MOVEBD (or @PRT PORT*DOC.MOVEBD) | **Access:** Lib NBS*PORT |
| **MOVEBI** Move an integer vector using backward DO loop. Proprietary single precision Fortran subprogram in PORT library. **Class(es):** D1a5 **Usage:** CALL MOVEBI(N,A,B) **On-line doc:** CALL GAMSDOC MOVEBI (or @PRT PORT*DOC.MOVEBI) | **Access:** Lib NBS*PORT |
| **MOVEBL** Move a logical vector using backward DO loop. Proprietary single precision Fortran subprogram in PORT library. **Class(es):** D1a5 **Usage:** CALL MOVEBL(N,A,B) **On-line doc:** CALL GAMSDOC MOVEBL (or @PRT PORT*DOC.MOVEBL) | **Access:** Lib NBS*PORT |
| **MOVEBR** Move a real vector using backward DO loop. Proprietary single precision Fortran subprogram in PORT library. Double precision version is MOVEBD. **Class(es):** D1a5 **Usage:** CALL MOVEBR(N,A,B) **On-line doc:** CALL GAMSDOC MOVEBR (or @PRT PORT*DOC.MOVEBR) | **Access:** Lib NBS*PORT |
| **MOVEFC** Move a complex vector using forward DO loops. Proprietary double precision Fortran subprogram in PORT library. **Class(es):** D1a5 **Usage:** CALL MOVEFC(N,A,B) **On-line doc:** CALL GAMSDOC MOVEFC (or @PRT PORT*DOC.MOVEFC) | **Access:** Lib NBS*PORT |
NBS*PORT

MOVEFD Move a double precision vector using forward DO loop. | Proprietary double precision Fortran subprogram in PORT library. | Class(es): Dida5 | Usage: CALL MOVEFD (N,A,B) | On-line doc: CALL GAMSDOC MOVEFD (or @PRT PORT+DOC.MOVEFD) | Access: LIB NBS*PORT


MOVEFL Move a logical vector using forward DO loop. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): Dida5 | Usage: CALL MOVEFL (N,A,B) | On-line doc: CALL GAMSDOC MOVEFL (or @PRT PORT+DOC.MOVEFL) | Access: LIB NBS*PORT

MOVEFR Move a real vector using forward DO loop. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is MOVRFD. | Class(es): Dida5 | Usage: CALL MOVEFR (N,A,B) | On-line doc: CALL GAMSDOC MOVEFR (or @PRT PORT+DOC.MOVEFR) | Access: LIB NBS*PORT

MPLT Displays a 50x100 character line printer plot of several dependent variables vs. a common independent variable. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLT (YM, X, N, M, IYM) | On-line doc: CALL GAMSDOC MPLT (or @PRT STATLIB+DOC.MPLT) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

MPLTH Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLTH (YM, X, N, M, IYM) | On-line doc: CALL GAMSDOC MPLTH (or @PRT STATLIB+DOC.MPLTH) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

MPLTL Displays a 50x50 character line printer plot of several dependent variables vs. a common independent variable, with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c3 Q1 | Usage: CALL MPLTL (YM, X, N, M, IYM, NOUT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC MPLTL (or @PRT STATLIB+DOC.MPLTL) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

MPOLISH Uses median polish to fit an additive model to a two-way layout which may be unbalanced and may have empty cells. Options: fit columns first, number of iterations, save results. | Command in MINITAB Proprietary interactive system. | Class(es): L0d | Usage: MPOLISH the data in C, row levels in C, column levels in C [put residuals into C [fits into C]] ; COLUMNS first; ITERATIONS = K; EFFECTS, put common into K, row effects into C, column effects into C; COMPARISON values, put into C. | On-line doc: HELP MPOLISH in MINITAB) | Tests: MINITAB+TEST.SOURCE | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

MSEN0 Expected values of normal order statistics. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a2n | Usage: CALL MSEN0 (IFIRST, ILAST, N, EX, IER) | On-line doc: CALL GAMSDOC MSEN0 (or @PRT IMSL+DOC.MSEN0) | Access: LIB NBS+IMSL

MSMRAT Ratio of the ordinate to the upper tail area of the standardized normal (Gaussian) distribution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L5a10 | Usage: CALL MSMRAT (X,RM,IER) | On-line doc: CALL GAMSDOC MSMRAT (or @PRT IMSL+DOC.MSMRAT) | Access: LIB NBS+IMSL
N

NADER Friedman’s test for randomised complete block designs. | Proprietary single precision Fortran program in IMSL library. | Class(es): L7a2a2 | Usage: CALL NAFRE (Y, NB, NT, ALPHA, STAT, RJ, D, IWK, WK, IER) | On-line doc: CALL GAMSDOC NAFRE (or @PRT IMSL+DOC.NAFRE) | Access: LIB NBS+IMSL


NAWNRNP Wilons ANOVA (2 or 3 way designs) without replicates. | Proprietary single precision Fortran program in IMSL library. | Class(es): L7a2a2 | Usage: CALL NAWNRNP (Y,N,IR,NF,NDV,STAT,IER) | On-line doc: CALL GAMSDOC NAWNRNP (or @PRT IMSL+DOC.NAWNRNP) | Access: LIB NBS+IMSL

NAWRPE Wilons ANOVA (1, 2, or 3 way designs) with equal replication. | Proprietary single precision Fortran program in IMSL library. | Class(es): L7a1b L7a2a2 | Usage: CALL NAWRPE (Y,N,REPS,IR,NF,NDV,STAT,IER) | On-line doc: CALL GAMSDOC NAWRPE (or @PRT IMSL+DOC.NAWRPE) | Access: LIB NBS+IMSL

NAWRPU Wilons ANOVA (1, 2, or 3 way designs) with unequal replication. | Proprietary single precision Fortran program in IMSL library. | Class(es): L7a1b L7a2a2 | Usage: CALL NAWRPU (Y,N,REPS,IR,NF,NDV,STAT,IER) | On-line doc: CALL GAMSDOC NAWRPU (or @PRT IMSL+DOC.NAWRPU) | Access: LIB NBS+IMSL

NBCDF Computes the cumulative distribution function value at X for the negative binomial distribution with parameters P and N. | Portable single precision Fortran program in DATAPAC library. | Class(es): L5a1n | Usage: CALL NBCDF(X,P,N,CDF) | On-line doc: CALL GAMSDOC NBCDF (or @PRT DATAPAC+DOC.NBCDF) | Access: LIB NBS+DATAPAC


NBPPF Computes the percent point function value at P for the negative binomial distribution with parameters PPAR and N. | Portable single precision Fortran program in DATAPAC library. | Class(es): L5a20 | Usage: CALL NBPPF(P,PPAR,N,PPF) | On-line doc: CALL GAMSDOC NBPPF (or @PRT DATAPAC+DOC.NBPPF) | Access: LIB NBS+DATAPAC

NBQT Cochran q test. | Proprietary single precision Fortran program in IMSL library. | Class(es): L4b1b | Usage: CALL NBQT (X,N,M,IA,Q,FQ,IER) | On-line doc: CALL GAMSDOC NBQT (or @PRT IMSL+DOC.NBQT) | Access: LIB NBS+IMSL

NBRAN Generates a random sample of size N from the negative binomial distribution with parameters P and NPAR. | Portable single precision Fortran program in DATAPAC library. | Class(es): L6a1d | Usage: CALL NBRAN(P,N,NPAR,START,X) | On-line doc: CALL GAMSDOC NBRAN (or @PRT DATAPAC+DOC.NBRAN) | Access: LIB NBS+DATAPAC


NDEST Evaluate probability density function at specified points. | Proprietary single precision Fortran program in IMSL library. | Class(es): L5a1g | Usage: CALL NDEST(Y,N,F,M,IOPT,B,WK,EST,IER) | On-line doc: CALL GAMSDOC NDEST (or @PRT IMSL+DOC.NDEST) | Access: LIB NBS+IMSL

NDFER Nonparametric probability density function (one dimensional) estimation by the kernel method. | Proprietary single precision Fortran program in IMSL library. | Class(es): L4a1e | Usage: CALL NDFER (X,N,H,DEL,XKER,B,NPT,F) | On-line doc: CALL GAMSDOC NDFER (or @PRT IMSL+DOC.NDFER) | Access: LIB NBS+IMSL

NDFMLE Nonparametric probability density function (one dimensional) estimation by the penalized likelihood method. | Proprietary single precision Fortran program in IMSL library. | Class(es): L4a1e | Usage: CALL NDFMLE (X,N,NMAX,IND,ALPHA,EPS,ITMAX,F,B,XJAC,IXJAC,LOHI,DELC,IER) | On-line doc: CALL GAMSDOC NDFMLE (or @PRT IMSL+DOC.NDFMLE) | Access: LIB NBS+IMSL


NERROH Provides the current error number for PORT library programs. | Proprietary single precision Fortran program in PORT library. | Class(es): R3 | Usage: I = NERROH (NERR) | On-line doc: CALL GAMSDOC NERROH (or @PRT PORT+DOC.ENERROH) | Access: LIB NBS+PORT

NFIT Performs nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm. | Portable single precision Fortran program in STATLIB library. | Class(es): L8g1a | Usage: CALL NFIT(Y,XM,MODEL,N,NP,NX,SCRAT,NS,IXM,RES,COEF) |
On-line doc: CALL GAMSDOC NFIT (or @PRT STATLIB+DOC.NFIT) | Tests: STATLIB+TEST.DEM02 | Access: Lib NBS+STATLIB

NFITS Performs nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user. Portable single precision Fortran subroutine in STATLIB library. | Class(es): Lg61a | Usage: CALL NFITS (Y, XM, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF, STAT, PV, SDPV, SDRES, VCV, IVCV, FLAMBD, FNU, MIT, STEP, STOPSS, STOPCR, NPRT, IEIG) | On-line doc: CALL GAMSDOC NFITS (or @PRT STATLIB+DOC.NFITS) | Tests: STATLIB+TEST.DEM02 | Access: Lib NBS+STATLIB

NFITW Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm. Portable single precision Fortran subroutine in STATLIB library. | Class(es): Lg62a | Usage: CALL NFTITW (Y, XM, WT, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF, STAT, PV, SDPV, SDRES, VCV, IVCV, FLAMBD, FNU, MIT, STEP, STOPSS, STOPCR, NPRT, IEIG) | On-line doc: CALL GAMSDOC NFITW (or @PRT STATLIB+DOC.NFITW) | Tests: STATLIB+TEST.DEM02 | Access: Lib NBS+STATLIB

NFITWS Performs weighted nonlinear least squares regression analysis using the Levenberg-Marquardt algorithm, with results returned to the user. Portable single precision Fortran subroutine in STATLIB library. | Class(es): Lg62a | Usage: CALL NFTITWS (Y, XM, WT, MODEL, N, NP, NX, SCRAT, NS, IXM, RES, COEF, STAT, PV, SDPV, SDRES, VCV, IVCV, FLAMBD, FNU, MIT, STEP, STOPSS, STOPCR, NPRT, IEIG) | On-line doc: CALL GAMSDOC NFITWS (or @PRT STATLIB+DOC.NFITWS) | Tests: STATLIB+TEST.DEM02 | Access: Lib NBS+STATLIB

NHEXT Fishers exact method for 2-by-2 tables. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L0a | Usage: CALL NHEXT (ITA, B, T, P, IER) | On-line doc: CALL GAMSDOC NHEXT (or @PRT IMSL+DOC.NHEXT) | Access: Lib NBS+IMSL | See also: BDC01U BDC02U

NHINC Includance test. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L4b1b | Usage: CALL NHINC (X, N, Y, M, I12, STAT, IER) | On-line doc: CALL GAMSDOC NHINC (or @PRT IMSL+DOC.NHINC) | Access: Lib NBS+IMSL

NKS1 Kolmogorov-Smirnov one-sample test. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L4alc | Usage: CALL NKS1 (PDF X, N, P, PDF, IER) | On-line doc: CALL GAMSDOC NKS1 (or @PRT IMSL+DOC.NKS1) | Access: Lib NBS+IMSL


NL2S1 Minimizes a nonlinear sum of squares using both residual and gradient values supplied by the user. Portable single precision Fortran subroutine in NL2SN sublibrary of CMLIB library. Double precision version is DNLS21. Class(es): Lg6lb Lg62b Kib1a2 | Usage: CALL NL2S1 (NP X, CALC X, CACL J, IV, LIV, LV, V, U, UIPARM, URFARM, UPFARM) | On-line doc: CALL GAMSDOC NLS21 (or @PRT CMLIB+DOC.NL2S1/NL2SN) | Tests: CMLIB+TEST-SOURCE.$FL/NL2SN, CMLIB+TEST-SOURCE.$Q1/NL2SN | Access: Lib NBS+CMLIB

NL2SN Minimizes a nonlinear sum of squares using residual values only. Portable single precision Fortran subroutine in NL2SN sublibrary of CMLIB library. Double precision version is DNLSN2. Class(es): Lg6lb Lg62a Kib1a1 | Usage: CALL NL2SN (NP X, CALC X, LIV, LV, V, U, UIPARM, URFARM, UPFARM) | On-line doc: CALL GAMSDOC NL2SN (or @PRT CMLIB+DOC.NL2S1/NL2SN) | Tests: CMLIB+TEST-SOURCE.$FL/NL2SN, CMLIB+TEST-SOURCE.$Q1/NL2SN | Access: Lib NBS+CMLIB

NMCC Calculate and test the significance of the Kendall coefficient of concordance. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L4b1b | Usage: CALL NMCC (X, M, N, IX, EPS, IR, R, STAT, IER) | On-line doc: CALL GAMSDOC NMCC (or @PRT IMSL+DOC.NMCC) | Access: Lib NBS+IMSL

NMKN Kendall's test for correlation (rank correlation coefficient). Proprietary single precision Fortran subroutine in IMSL library. Class(es): L4b1b | Usage: CALL NMKN (X, Y, N, STAT, XSTAT, IWK, WK, IER) | On-line doc: CALL GAMSDOC NMKN (or @PRT IMSL+DOC.NMKN) | Access: Lib NBS+IMSL

NMKSF Frequency distribution of K and the probability of equalling or exceeding K, where K, the total score from the Kendall rank correlation coefficient calculations, and N, the sample size, are given. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L4b1b | Usage: CALL NMKSF (K, N, Z, W, P) | On-line doc: CALL GAMSDOC NMKSF (or @PRT IMSL+DOC.NMKSF) | Access: Lib NBS+IMSL | See also: NMKN


NMTIE Tie statistics, given a sample of observations. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L1a1 | Usage: CALL NMTIE (X, XM, EPS, TIES) | On-line doc: CALL GAMSDOC NMTIE (or @PRT IMSL+DOC.NMTIE) | Access: Lib NBS+IMSL

NORCDF Computes the cumulative distribution function value for the normal (Gaussian) distribution with mean = 0 and standard deviation = 1. Portable single precision Fortran subroutine in DATAPAC library. Class(es): L5a1a | Usage: CALL NORCDF ($X, CDF) | On-line doc: CALL GAMSDOC NORCDF (or @PRT DATAPAC+DOC.NORCDF) | Access: Lib NBS+DATAPAC

NOROUT Performs a normal outlier analysis on the data in the input vector X. Portable single precision Fortran subroutine in DATAPAC library. Class(es): L4a1a14 | Usage: CALL NOROUT ($X, N) | On-line doc: CALL GAMSDOC NOROUT (or @PRT
DATAPAC**DOC.NOROUT** | Access: LIB NBS*DATAPAC

NORPDF Computes the probability density function value for the normal (Gaussian) distribution with mean $= 0$ and standard deviation $= 1$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a1n | Usage: CALL NORPDF(X,PDF) | On-line doc: CALL GAMSDOC NORPDF (or @PRT DATAPAC**DOC.NORPDF** | Access: LIB NBS*DATAPAC

NORPLT Generates a normal (Gaussian) probability plot with mean $= 0$ and standard deviation $= 1$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5c4n | Usage: CALL NORPLT(X,N) | On-line doc: CALL GAMSDOC NORPLT (or @PRT DATAPAC**DOC.NORPLT** | Access: LIB NBS*DATAPAC

NORPPF Computes the percent point function value for the normal (Gaussian) distribution with mean $= 0$ and standard deviation $= 1$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a2n | Usage: CALL NORPPF(P,PFF) | On-line doc: CALL GAMSDOC NORPPF (or @PRT DATAPAC**DOC.NORPPF** | Access: LIB NBS*DATAPAC

NORRAN Generates a random sample of size N from the normal (Gaussian) distribution with mean $= 0$ and standard deviation $= 1$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a14 | Usage: CALL NORRAN(N,START,X) | On-line doc: CALL GAMSDOC NORRAN (or @PRT DATAPAC**DOC.NORRAN** | Access: LIB NBS*DATAPAC

NORSF Computes the sparsity function value for the normal (Gaussian) distribution with mean $= 0$ and standard deviation $= 1$. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a2n | Usage: CALL NORSF(P,SR) | On-line doc: CALL GAMSDOC NORSF (or @PRT DATAPAC**DOC.NORSF** | Access: LIB NBS*DATAPAC

NRAND Generates a vector of normally distributed pseudo-random numbers. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L6a14 | Usage: CALL NRAND(Y, N, SEED) | On-line doc: CALL GAMSDOC NRAND (or @PRT STATLIB**DOC.NRAND** | Tests: STATLIB**TEST.DEM01** | Access: LIB NBS*STATLIB


NRBHA Bhapkar v test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4blb | Usage: CALL NRBHA(X,NPS,IC,IR,W,V,Q,IER) | On-line doc: CALL GAMSDOC NRBHA (or @PRT IMSL**DOC.NRBHA** | Access: LIB NBS*IMSL

NRWMD Wilcoxon signed rank test. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4blb | Usage: CALL NRWMD(IOPT,X,Y,NPS,DSEED,IR,STAT,IER) | On-line doc: CALL GAMSDOC NRWMD (or @PRT IMSL**DOC.NRWMD** | Access: LIB NBS*IMSL


NUMXER Get most current message number. | Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: I=NUMXER(NERR) | On-line doc: CALL GAMSDOC NUMXER (or @PRT CMLIB**DOC.SUMMARY/XERROR** | Access: LIB NBS*CMLIB

OCLINK Perform a single-linkage or complete-linkage hierarchical cluster analysis given a similarity matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L14a1 | Usage: CALL OCLINK (ND,IOPT,XSIM,LEVEL,ICLSON,JCRSON,IPTR,IER) | On-line doc: CALL GAMSDOC OCLINK (or @PRT IMSL+DOC.OCLINK) | Access: LIB NBS+IMSL

ODEQ Finds the integral of a set of functions over the same interval by using the differential equation solver ODES1. For smooth functions. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODEQ. | Class(es): H2a1a1 | Usage: CALL ODEQ (N, F, A, E, EPS, ANS) | On-line doc: CALL GAMSDOC ODEQ (or @PRT PORT+DOC.ODEEQ) | Access: LIB NBS+PORT

ODES Solves an initial value problem for a system of ordinary differential equations. Easy to use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODES. | Class(es): L1a1c | Usage: CALL ODES (F,X,NT,TSTART,TSTOP,DT,ERRPAR,HANDLE) | On-line doc: CALL GAMSDOC ODES (or @PRT PORT+DOC.ODES) | Access: LIB NBS+PORT

ODES1 Solves an initial value problem for a system of ordinary differential equations. Allows great flexibility and user control. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODES1. | Class(es): L1a1c | Usage: CALL ODES1 (F,X,NT,TSTART,TSTOP,DT,ERROR,ERRPAR,HANDLE,GLBMAX,ERPUTS) | On-line doc: CALL GAMSDOC ODES1 (or @PRT PORT+DOC.ODES1) | Access: LIB NBS+PORT | See also: ODES ODESE

ODSE Standard error subroutine for the routine ODES1. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODESE. | Class(es): I1c | Usage: L = ODESE (X,N,T,DT,ERRPAR,ERPUTS,E) | On-line doc: CALL GAMSDOC ODSE (or @PRT PORT+DOC.ODSE) | Access: LIB NBS+PORT | See also: ODESE

ODESH Default HANDLE routine for ODEs. Used to access the results at the end of each integration time step. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DODESH. | Class(es): I1c | Usage: CALL ODESH (T0,X0,T1,X1,NT,D,DT,TSTOP,E) | On-line doc: CALL GAMSDOC ODESH (or @PRT PORT+DOC.ODESH) | Access: LIB NBS+PORT | See also: ODES

ODFSIL Linear discriminant analysis method of Fisher for reducing the number of variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L12 | Usage: CALL ODFSIL (X,I,N,G,NV,ND,XM,XM,NNV,E,C,IC,SW,SB,ISB,EX,CX,1CX,IS,IER) | On-line doc: CALL GAMSDOC ODFSIL (or @PRT IMSL+DOC.ODFSIL) | Access: LIB NBS+IMSL


OFIMA3 Least squares solution to the matrix equation AT = B. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L1a8 a D9 | Usage: CALL OFIMA3 (A,IA,B,IB,NV,NF,NF,NF,T,T,WK,IER) | On-line doc: CALL GAMSDOC OFIMA3 (or @PRT IMSL+DOC.OFIMA3) | Access: LIB NBS+IMSL


OFPRI Compute an unrotated factor loading matrix according to a principal component model. | Proprietary single precision Fortran


OFSCOR Compute a set of factor scores given the factor score coefficient matrix. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13 | Usage: CALL OFSCOR (C,J,C,NV,NF,NT,Z,IZ,ZBAR,STD,FMEAN,SS,WK,IER) | On-line doc: CALL GAMSDOC OFSCOR (or @PRT IMSL*DOC.OFSCOR) | Access: LIB NBS*IMSL | See also: OFCOEF

OIND Wilks test for the independence of k sets of multi-normal variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L4e4a14 | Usage: CALL OIND (S,N,IP,K,STAT,WKAREA,IER) | On-line doc: CALL GAMSDOC OIND (or @PRT IMSL*DOC.OIND) | Access: LIB NBS*IMSL


ONEWY Performs one-way analysis of variance for two or more random samples. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L7a1a | Usage: CALL ONEWY (Y, TAG, N, SCRAT, NS) | On-line doc: CALL GAMSDOC ONEWY (or @PRT STATLIB*DOC.ONEWY) | Tests: STATLIB*TEST-DEMO1 | Access: LIB NBS*STATLIB

ONEWYS Performs one-way analysis of variance for two or more random samples, with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L7a1a | Usage: CALL ONEWYS (Y, TAG, N, SCRAT, NS, NPRT, TAGVAL, GRPNOB, GRPAVG, GRPSD) | On-line doc: CALL GAMSDOC ONEWYS (or @PRT STATLIB*DOC.ONEWYS) | Tests: STATLIB*TEST-DEMO1 | Access: LIB NBS*STATLIB

OPRINC Principal components of a multivariate sample of observations. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L13a | Usage: CALL OPRINC (S,M,IA,EVAL,EVEC,COMP,VAR,CL,CI,EUR) | On-line doc: CALL GAMSDOC OPRINC (or @PRT IMSL*DOC.ODPRINC) | Access: LIB NBS*IMSL

ORDER Sorts in ascending order the values in each of one or more vectors. | Command in MINITAB Proprietary interactive system. | Class(es): N02a2b | Usage: ORDER C [and C,...,C] put into C [and into C,...,C] | On-line doc: HELP ORDER (in Minimitab) | Tests: MINITAB*TEST-SOURCE. | Access: @XTQ NBS*MINTAB.MINITAB (or CALL MINITAB in CTS)

ORTBak Forms eigenvectors of general real matrix from eigenvectors of upper Hessenberg matrix output from ORTHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB Library. | Class(es): D4e4 | Usage: CALL ORTBAK(NM,LOW,IGH,H,ORT,M,Z) | On-line doc: CALL GAMSDOC ORTBak (or @PRT CMLIB*DOC.ORTBAK/EISPACK) | Access: LIB NBS*CMLIB | See also: ORTHES


ORThp Evaluates a polynomial expressed as a sum of general orthogonal polynomials. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DORTHP. | Class(es): C3 | Usage: X * ORTHP (N, ALPHA, X, A, B, C) | On-line doc: CALL GAMSDOC ORTHP (or @PRT PORT*DOC.ORTHP) | Access: LIB NBS*PORT

ORTRAN Accumulates orthogonal similarity transformations in reduction of real general matrix by ORTHES. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB Library. | Class(es): D4c4 | Usage: CALL ORTRAN(NM,N,LOW,HI,A,ORT,Z) | On-line doc: CALL GAMSDOC ORTRAN (or @PRT CMLIB*DOC.ORTRAN/EISPACK) | Access: LIB NBS*CMLIB | See also: ORTHES

OTMLNR Maximum likelihood estimation from grouped and/or censored normal data. | Proprietary single precision Fortran subprogram in
P

P01AAE Return value of error indicator, or terminate with an error message. Used exclusively by NAG library programs. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is P01AAF. | Class(es): R3c | Usage: 1 = P01AAE (IFAIL, IERROR, SRNAME) | On-line doc: CALL GAMSDOC P01AAE (or PRT NAG+DOC.P01AAE) | Access: LIB NBS*NAG

P01AAF Return value of error indicator, or terminate with an error message. Used exclusively by NAG library programs. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is P01AAE. | Class(es): R3c | Usage: 1 = P01AAF (IFAIL, IERROR, SRNAME) | On-line doc: CALL GAMSDOC P01AAF (or PRT NAG+DOC.P01AAF) | Access: LIB NBS*NAG

P1D Prints univariate statistics (mean, standard deviation, std. error of mean, coefficient of variation, extreme values, extreme z-scores, range) for each variable. Options: statistics for each level of each grouping variable, sorting, printing all cases OR only cases with values missing or values outside specified limits. | Proprietary stand-alone program using BMDP command language. | Class(es): L1a1 L1a2 | Usage: @NBS*PLIB8.BMDP BMDPID | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPID | Access: @NBS*PLIB8.BMDP BMD<program name>

P1L Estimates survival (time-to-response) distribution of patients who have been observed over varying periods of time by product-limit (Kaplan-Meier) or actuarial life table (Cutler-Ederer) method. Options: three forms of input, Mantel-Cox and Breslow test of equality of survival curves, five plots. | Proprietary stand-alone program using BMDP command language. | Class(es): L15 | Usage: @NBS*PLIB8.BMDP BMDP1L | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1L | Access: @NBS*PLIB8.BMDP BMD<program name>

P1M Stepwise cluster analysis of variables using one of four measures of similarity, three criteria for combining clusters, with printing of a summary table of clusters, shaded distance measure display, and a tree showing cluster formation. Options: form of input, additional printing and display. | Proprietary stand-alone program using BMDP command language. | Class(es): L14a1a | Usage: @NBS*PLIB8.BMDP BMDPM1 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPM1 | Access: @NBS*PLIB8.BMDP BMD<program name>

P1R Performs multiple linear regression and prints standard results. Options: weights, form of input, regression on subsets or groups and test of equality of regression lines, intercept term present or absent, more printing, five plots, save predicted values and residuals. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4 L8a7 | Usage: @NBS*PLIB8.BMDP BMDPR1 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPR1 | Access: @NBS*PLIB8.BMDP BMD<program name>

P1S At each pass through the data, computes univariate statistics (choose means, standard deviations, geometric means, harmonic means, extreme values), and transforms or edits the data using statistics computed in the previous pass. Options: printing, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L2a | Usage: @NBS*PLIB8.BMDP BMDPS1 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPS1 | Access: @NBS*PLIB8.BMDP BMD<program name>

P1T Interactive or batch spectral analysis of one or two time series, with estimates of spectral density and coherence between variables. Options: print, plot (variable vs. time, lagged plots, complex demodulation, periodogram), handle missing values, remove seasonal means and linear trend, filtering, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L10f | Usage: @NBS*PLIB8.BMDP BMDP1T | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDP1T | Access: @NBS*PLIB8.BMDP BMD<program name>

P1V Performs one-way ANOVA or ANCOVA with standard results. For ANCOVA, tests 1) equality of slopes, 2) zero slope, and 3) equality of adjusted cell means; plots the covariate for each group. Tests equality of pairs of means (or adjusted means). Options: linear contrasts, within-group correlations and statistics. | Proprietary stand-alone program using BMDP command language. | Class(es): L7a1a L7a3 | Usage: @NBS*PLIB8.BMDP BMDPV1 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPV1 | Access: @NBS*PLIB8.BMDP BMD<program name>

P2D For each variable, prints frequency and percent for each distinct value; mean, median, mode, standard deviation, std. errors of mean and median, skewness, kurtosis, half interquartile range; histogram, and stem-and-leaf plots. Options: initially round or truncate, three robust location estimates. | Proprietary stand-alone program using BMDP command language. | Class(es): L1a1 L1a2 | Usage: @NBS*PLIB8.BMDP BMDPD2 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPD2 | Access: @NBS*PLIB8.BMDP BMD<program name>

P2L Analyzes survival data with covariates using Cox proportional hazard regression model. Options: two forms of input, stepwise selection of covariates, time-dependent covariates, stratification, significance tests, three plots, print survival functions and residuals. | Proprietary stand-alone program using BMDP command language. | Class(es): L15 | Usage: @NBS*PLIB8.BMDP BMDPL2 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPL2 | Access: @NBS*PLIB8.BMDP BMD<program name>

P2M Stepwise cluster analysis of cases (observations) using one of four distance measures (including Euclidean and one for data that are frequency counts) and three linkage algorithms (single, centroid, k nearest neighbors), with a summary table of clusters and a cluster tree. Options: weights, standardized data. | Proprietary stand-alone program using BMDP command language. | Class(es): L14a1a | Usage: @NBS*PLIB8.BMDP BMDPM2 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPM2 | Access: @NBS*PLIB8.BMDP BMD<program name>

P2R Multiple linear regression, with standard results. Options: weights, forward or backward stepping, interactive stepping, stepping sets of variables (e.g. design variables), forcing variables into the model, eleven diagnostics (including Cook and AP statistics) available for printing, plotting, and saving. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4 L8a9 L8a5 | Usage: @NBS*PLIB8.BMDP BMDPR2 | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPR2 | Access: @NBS*PLIB8.BMDP BMD<program name>
P2T Interactive or batch Box-Jenkins time series analysis for univariate time domain models (including ARIMA, regression, intervention, and transfer function models) - model identification, parameter estimation, testing, forecasting. Options: print, plot, differencing and filtering, save results. Proprietary stand-alone program using BMDP command language. Class(es): L10 | Usage: @NBS*PLIB*.BMDP BMDP2T | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP2T | Access: @NBS*PLIB*.BMDP BMD<program name>

P2V ANOVA and ANCOVA for unbalanced fixed-effect models (including full and fractional factorial designs, Latin squares), and repeated measure models, or a combination of models, with Greenhouse-Geisser and Huynh-Feldt deg. of freedom adjustment. Options: orthogonal decomposition of within-effects, save results. Proprietary stand-alone program using BMDP command language. Class(es): L7a | Usage: @NBS*PLIB*.BMDP BMDP2V | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP2V | Access: @NBS*PLIB*.BMDP BMD<program name>

P3D One-sample t-test to test if one group mean is zero (e.g., matched pairs); two-sample t test with and without equal variances assumption, Levene's test for equal variances, histograms. Options: trimmed t test, Hotelling's T-squared and Mahalanobis' D-squared, within-group correlations, data listing. Proprietary stand-alone program using BMDP command language. Class(es): L1a4 | Usage: @NBS*PLIB*.BMDP BMDP3D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP3D | Access: @NBS*PLIB*.BMDP BMD<program name>

P3M Performs nonlinear least squares regression with standard results. Six functions are built in; others can be specified. Options: weights, evaluates functions of parameters (with standard errors), upper and lower limits on parameters, ridging, exact linear constraints, maximum likelihood estimates, and five plots. Proprietary stand-alone program using BMDP command language. Class(es): L8b1 L8b2 L8b | Usage: @NBS*PLIB*.BMDP BMDP3M | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP3M | Access: @NBS*PLIB*.BMDP BMD<program name>

P3S Computes and prints results from one or more of the following sign test, Wilcoxon signed-rank test, Mann-Whitney rank-sum test, Kruskal-Wallis one-way ANOVA, Friedman two-way ANOVA, Kendall's coefficient of concordance, Kendall and Spearman rank-correlation coefficients. Proprietary stand-alone program using BMDP command language. Class(es): L4a1b L6b1 L7a2a2 | Usage: @NBS*PLIB*.BMDP BMDP3S | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP3S | Access: @NBS*PLIB*.BMDP BMD<program name>

P3V Uses maximum likelihood (ML) and restricted ML approaches to balanced and unbalanced fixed and random coefficient models of quite arbitrary form (including having covariates), with parameter estimation, hypothesis testing, and printing. Weights optional. Proprietary stand-alone program using BMDP command language. Class(es): L7a4 | Usage: @NBS*PLIB*.BMDP BMDP3V | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP3V | Access: @NBS*PLIB*.BMDP BMD<program name>

P4D Counts frequency of each number, letter, or symbol in single-column fields (A1 format). Options: input case label variables in A4 format, diagnostic printing useful in preliminary data screening. Specified characters may be replaced by blanks or symbols. Proprietary stand-alone program using BMDP command language. Class(es): L1a4 N3 | Usage: @NBS*PLIB*.BMDP BMDP4D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP4D | Access: @NBS*PLIB*.BMDP BMD<program name>

P4P Forms two- or multi-way frequency tables. Options: percent; 25 tests and measures for two-way tables; fits and tests log-linear models, tests of marginal and partial association, stepwise models, three forms of input, structural zeros, cell and strata deletion, residuals. Proprietary stand-alone program using BMDP command language. Class(es): L9 | Usage: @NBS*PLIB*.BMDP BMDP4P | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP4P | Access: @NBS*PLIB*.BMDP BMD<program name>

P4M Provides four methods of initial factor extraction from a correlation or covariance matrix, and several methods of rotation, prints shaded correlations, factor loadings, factor score coefficients, factor scores, Mahalanobis distances, and plots. Options: weights, form of input, save results. Proprietary stand-alone program using BMDP command language. Class(es): L10 | Usage: @NBS*PLIB*.BMDP BMDP4M | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP4M | Access: @NBS*PLIB*.BMDP BMD<program name>

P4R Regression analysis for a dependent variable on a set of principal components computed from the independent variables in a stepwise manner determined either by magnitude of eigenvalue or correlations between dependent variable and components, with printing. Options: form of input, more printing, four plots. Proprietary stand-alone program using BMDP command language. Class(es): L8a4a1d | Usage: @NBS*PLIB*.BMDP BMDP4R | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP4R | Access: @NBS*PLIB*.BMDP BMD<program name>

P4V Interactive or batch univariate and multivariate ANOVA and ANCOVA, including nested, repeated measures, split-plot, and changeover designs, and model building features. Options: cell weights for hypothesis testing, contrasts, tests of simple effects, save cell means. Proprietary stand-alone program using BMDP command language. Class(es): L7 | Usage: @NBS*PLIB*.BMDP BMDP4V | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP4V | Access: @NBS*PLIB*.BMDP BMD<program name>

P5D Prints histograms with frequencies and percentages, normal and detrended normal probability plots, halfnormal plots, cumulative frequency distribution plots, and cumulative histograms for ungrouped data or for grouped data - either separately or combined in one plot. Plot options. Proprietary stand-alone program using BMDP command language. Class(es): L3b L3c4h L3c4n | Usage: @NBS*PLIB*.BMDP BMDP5D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP5D | Access: @NBS*PLIB*.BMDP BMD<program name>
P56R Least squares fit of a polynomial in one independent variable to the dependent variable. Prints standard results and goodness-of-fit statistics for each polynomial degree. Computations use orthogonal polynomials. Options: weights, additional printing, and three plots.

Proprietary stand-alone program using BMDP command language. Class(es): L850126 | Usage: @NBS*PLIB$.BMDP BMDP56R | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP56R | Access: @NBS*PLIB$.BMDP BMD P<program name>

P66D Bivariate (scatter) plots. Options: several variables, or subsets of one variable (symbols identify group membership), on the same plot; prints correlation and linear regression statistics (line is marked on plot frame); user control for plot size, scales, and symbols.

Proprietary stand-alone program using BMDP command language. Class(es): L1C | Usage: @NBS*PLIB$.BMDP BMDP66D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP66D | Access: @NBS*PLIB$.BMDP BMD P<program name>

P86M Computes canonical correlation analysis for two sets of variables and Bartlett's test for the significance of the remaining eigenvalues, with printing. Options: weights, form of input, additional printing and plotting, save results.

Proprietary stand-alone program using BMDP command language. Class(es): L11 | Usage: @NBS*PLIB$.BMDP BMDP86M | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP86M | Access: @NBS*PLIB$.BMDP BMD P<program name>

P86R Computes the partial correlations of a set of variables after removing the linear effects of a second set of variables. Can be used for regression, especially if multiple dependent variables are present. Prints standard results. Options: weights, form of input, additional printing and plots.

Proprietary stand-alone program using BMDP command language. Class(es): L11 L8517 | Usage: @NBS*PLIB$.BMDP BMDP86R | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP86R | Access: @NBS*PLIB$.BMDP BMD P<program name>

P76D Side-by-side histograms for each cell in one-way or two-way ANOVA, within-group summary statistics and ANOVA table (with equality of variance test and tests that do not assume equal variances). Options: trimmed mean analysis, ANOVA diagnostics, tests of pairwise mean comparisons, correlations, Winsorized means. Proprietary stand-alone program using BMDP command language.

Class(es): L711 L721a1 | Usage: @NBS*PLIB$.BMDP BMDP76D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP76D | Access: @NBS*PLIB$.BMDP BMD P<program name>

P76M Stepwise forward or backward discriminant analysis (including jackknifed classifications, percent correctly classified, Mahalanobis' distances, canonical variable coefficients, eigenvalues, scores, and plot of first two canonical variables). Options: interactive stepping, saving results. Proprietary stand-alone program using BMDP command language.

Class(es): L12L Usage: @NBS*PLIB$.BMDP BMDP76M | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP76M | Access: @NBS*PLIB$.BMDP BMD P<program name>

P66D Four methods to compute covariance and correlation matrices when data contain missing values or values out of range. Options: weights, summary statistics, save results, pairwise t-tests based on the pattern of incomplete data. Proprietary stand-alone program using BMDP command language.

Class(es): L12L Usage: @NBS*PLIB$.BMDP BMDP66D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP66D | Access: @NBS*PLIB$.BMDP BMD P<program name>


Class(es): L13 Usage: @NBS*PLIB$.BMDP BMDP66M | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP66M | Access: @NBS*PLIB$.BMDP BMD P<program name>

P86V ANOVA for complete designs with equal cell sizes - nested, crossed, partially nested, partially crossed designs for fixed-effect models, mixed models (including repeated measures), and random-effect models, with parameter estimation and printing. Proprietary stand-alone program using BMDP command language.

Class(es): L721a2a Usage: @NBS*PLIB$.BMDP BMDP86V | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP86V | Access: @NBS*PLIB$.BMDP BMD P<program name>

P96D Provides descriptive statistics (means, std. devs., frequencies, one-way ANOVA table) of groups (cells) for data classified into cells using one or more grouping variables. Options: miniplots of cell means (eight per page), plot frames are defined by combinations of levels of grouping variables. Proprietary stand-alone program using BMDP command language.

Class(es): L721 L721a2a Usage: @NBS*PLIB$.BMDP BMDP96D | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP96D | Access: @NBS*PLIB$.BMDP BMD P<program name>

P96M Scoring based on preference pairs - for each observation construct score as linear combination of variables with coefficients based on expert preference, in stepwise manner. Options: printing, plots, compare results when analysis is repeated for different judges.

Proprietary stand-alone program using BMDP command language. Class(es): L85a1e Usage: @NBS*PLIB$.BMDP BMDP96M | On-line doc: @PRT BMDP+DOC.SUMMARY | Access: @NBS*PLIB$.BMDP BMD P<program name>

P96R Estimates regression equations for "best" (by R-squared, adjusted R-squared, or Mallows' C(p) criterion) subset of predictor variables by Furnival-Wilson algorithm. Options: weights, form of input, Durbin-Watson statistic. Cook's distance and several types of residuals may be printed, plotted, or saved.

Proprietary stand-alone program using BMDP command language.

Class(es): L85a L85a9 Usage: @NBS*PLIB$.BMDP BMDP96R | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDP96R | Access: @NBS*PLIB$.BMDP BMD P<program name>

PAM Describes pattern of invalid values (missing or out of range) for multivariate data. Options: weights, grouping, estimates covariance and correlation matrices by one of three methods (including maximum likelihood), replace invalid values using mean or one of several regression procedures, plots, save results. | Proprietary stand-alone program using BMDP command language. | Class(es): Llxe2 | Usage: @NBS*PLIB.BMDP BMDFAM | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPAM | Access: @NBS*PLIB.BMDP BMD<program name>

PAR Performs nonlinear regression using pseudo-Gauss-Newton algorithm. Derivatives are NOT specified. Options: weights, linear inequality constraints, maximum likelihood, functions of parameters, ridding, four plots, fitting models defined by differential equations. | Proprietary stand-alone program using BMDP command language. | Class(es): L8g1a L8g2a L8b | Usage: @NBS*PLIB.BMDP BMDPAR | On-line doc: @PRT BMDP*DOC.SUMMARY | Tests: BMDP*TEST-SOURCE.BMDPAM | Access: @NBS*PLIB.BMDP BMD<program name>

PARCDF Computes the cumulative distribution function value for the Pareto distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8a1p | Usage: CALL PARCDF(X,GAMMA,CDF) | On-line doc: CALL GAMSDOC PARCDF (or @PRT DATAPAC*DOC.PARCDF) | Access: LIB NBS*DATAPAC

PARPLT Generates a Pareto probability plot with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8c1p | Usage: CALL PARPLT(X,N,GAMMA) | On-line doc: CALL GAMSDOC PARPLT (or @PRT DATAPAC*DOC.PARPLT) | Access: LIB NBS*DATAPAC

PARPFP Computes the percent point function value for the Pareto distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8a1p | Usage: CALL PARPFP(P,GAMMA,PPF) | On-line doc: CALL GAMSDOC PARPFP (or @PRT DATAPAC*DOC.PARPFP) | Access: LIB NBS*DATAPAC

PARRAN Generates a random sample of size N from the Pareto distribution with tail length parameter GAMMA. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L8a16 | Usage: CALL PARRAN(N,GAMMA,ISTART,X) | On-line doc: CALL GAMSDOC PARRAN (or @PRT DATAPAC*DOC.PARRAN) | Access: LIB NBS*DATAPAC

PCHFD Evaluates a piecewise cubic Hermite function and its first derivative at an array of points. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: CALL PCHFD(N,X,F,D,ICNF,D,SKIP,NE,XE,FE,DE,IERR) | On-line doc: CALL GAMSDOC PCHFD (or @PRT CMLIB*DOC.PCHIP/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB

PCHFE Evaluates a piecewise cubic Hermite function at an array of points. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: CALL PCHFE(N,X,F,D,ICNF,SKIP,NE,XE,FE,IERR) | On-line doc: CALL GAMSDOC PCHFE (or @PRT CMLIB*DOC.PCHIP/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB


PCHIC Determines a piecewise monotone, piecewise cubic Hermite interpolant to given data. User control is available over boundary conditions and/or treatment of points where monotonicity switches direction. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: CALL PCHIC(IC,VC,SWITCH,N,X,F,D,ICNF,WF,WNK,IERR) | On-line doc: CALL GAMSDOC PCHIC (or @PRT CMLIB*DOC.PCHIC/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB | See also: PCHFE for evaluation. See package documentation for other facilities.

PCHID Evaluates the definite integral of a piecewise cubic Hermite function over an interval whose endpoints are data points. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: R = PCHID(N,X,F,D,ICNF,D,SKIP,IA,IB,IERR) | On-line doc: CALL GAMSDOC PCHID (or @PRT CMLIB*DOC.PCHID/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB

PCHIM Determines a monotone piecewise cubic Hermite interpolant to given data. Default boundary values are provided which are compatible with monotonicity. The interpolant will have an extremum at each point where monotonicity switches direction. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: CALL PCHIM(N,X,F,D,ICNF,SKIP,ISMON,IERR) | On-line doc: CALL GAMSDOC PCHIM (or @PRT CMLIB*DOC.PCHIM/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB | See also: PCHFE for evaluation. See package documentation for other facilities.

PCHMC Checks a cubic Hermite function for monotonicity. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: CALL PCHMC(N,X,F,D,ICNF,SKIP,ISMON,IERR) | On-line doc: CALL GAMSDOC PCHMC (or @PRT CMLIB*DOC.PHMC/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB

PCHSP Determines the cubic spline interpolant to given data. User has control over boundary conditions. | Portable single precision Fortran subprogram in PCHIP sublibary of CMLIB library. | Class(es): E3 | Usage: CALL PCHSP(IC,VC,N,X,F,D,ICNF,WF,WNK,IERR) | On-line doc: CALL GAMSDOC PCHSP (or @PRT CMLIB*DOC.PCHSP/PCHIP) | Tests: CMLIB*TEST-SOURCE.$Q/PCHIP | Access: LIB NBS*CMLIB | See also: PCHFE for evaluation. See package documentation for other facilities.

PDECOL Solves general nonlinear systems of initial-boundary-value problems in one space dimension with general boundary conditions. Spatial derivatives may be of at most second order. Uses method of lines based on collocation of B-spline basis functions. | Portable single precision Fortran subprogram in PDELIB library. | Class(es): L8a1a | Usage: CALL
PDECOL(T6,TOUT,DT,XXBPT,EP,NT,FINT,KORD,NCC,FPDE,MF,NINDEX,WORK,IWORK) | On-line doc: CALL GAMSDOC PDECOL (or @PRT PDELIB+DOC.PDECOL) | Tests: PDELIB+TEST-SOURCE.PDECOL | Access: LIB NBS+PDELIB

PDETW0 Solves general nonlinear systems of initial-boundary-value problems in two spatial dimensions with quasi-linear boundary conditions. Uses the method of lines based on finite differences on a user-specified rectangular mesh. | Portable single precision Fortran subprogram in PDELIB library. | Class(es): L2a1b | Usage: CALL DRIVEP(NODE,T0,H,U1,TOUT,EP,MF,NINDEX,WORK,IWORK,X,Y) | On-line doc: CALL GAMSDOC PDETW0 (or @PRT PDELIB+DOC.PDETW0) | Tests: PDELIB+TEST-SOURCE.PDETW0 | Access: LIB NBS+PDELIB


PFITWS Performs weighted least squares regression analysis of a polynomial model with computed results returned to the user. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L8a2b2 | Usage: CALL PFITWS (Y, X, WT, N, ID, SCAT, NS, RES, COEF, PV, SDPv, SDRS, VCV, IVCV, NPT) | On-line doc: CALL GAMSDOC PFITWS (or @PRT STATLIB+DOC.PFITWS) | Tests: STATLIB+TEST-DEMO2 | Access: LIB NBS+STATLIB

PFQAD Computes integral on (X1,X2) of product of function and the ID-th derivative of B-spline which is in piecewise polynomial representation. | Portable single precision Fortran program in spline library of CMLIB library. Double precision version is DFPQAD. | Class(es): H2a2al E3 K6 | Usage: CALL PFQAD(F,LDC,C,XI,LI,K,ID,X1,X2,TOL,QUAD,IERR) | On-line doc: CALL GAMSDOC PFQAD (or @PRT CMLIB+DOC.PFQAD/BSPLINE and CMLIB+DOC.SUMMARY/BSPLINE) | Tests: CMLIB+TEST-SOURCE.3F/BSPLINE | Access: LIB NBS+CMLIB

PKM By k-means procedure, partitions a set of cases (observations) into k clusters - beginning with user-specified initial clusters or one cluster, proceeding in divisive stepwise manner, then doing iterative reallocation - prints cluster profile and plot, Options: weights, standardize data (four ways), save results. | Proprietary stand-alone program using BMDP command language. | Class(es): L1a2 | Usage: @NBS+PLIB$BMDP BMDPKM | On-line doc: @PRT BMDP+DOC.SUMMARY | Tests: BMDP+TEST-SOURCE.BMDPKM | Access: @NBS+PLIB$BMDP BMD<program name>

PLOD An easy to use interactive system for the solution of initial value problems for ordinary differential equations. Requires a Tektronix or Hewlett Packard graphics terminal. The user can change initial conditions, interval, parameters etc., and examine various plots on the terminal. Little programming needed. | Portable single precision Fortran subprogram in PLOD library. | Class(es): L1a | Usage: @PRT PLOD+DOC.SUMMARY | Access: @XQT NBS+PLOD.PLOD (or @XQT NBS+PLOD.PLOD in CTS)

PLOT Yields a one-page printer plot of Y(I) versus X(I). | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOT (Y,X,N) | On-line doc: CALL GAMSDOC PLOT (or @PRT DATAPAC+DOC.PLOT) | Access: LIB NBS+DATAPAC

PLOT Prints a scatter diagram, with optional scale specification. | Command in MINITAB Proprietary interactive system. | Class(es): L3c1 Q1 | Usage: PLOT Y in C [from K to K] vs X in C [from K to K] | On-line doc: HELP PLOT (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or @MINITAB in CTS)

PLOT0 Yields a one-page printer plot of Y(I) versus X(I) for a subset of the data, with special plot characters, and with specified axis limits and labels. | Portable single precision Fortran subprogram in the DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT0(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX,D,DMIN,DMAX,YAXID,XXAXID,PLCHID) | On-line doc: CALL GAMSDOC PLOT0 (or @PRT DATAPAC+DOC.PLOT0) | Access: LIB NBS+DATAPAC

PLOT6 Yields a one-page printer plot of Y(I) versus X(I) for specified axis limits. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOT6(Y,X,N,YMIN,YMAX,XMIN,XMAX) | On-line doc: CALL GAMSDOC PLOT6 (or @PRT DATAPAC+DOC.PLOT6) | Access: LIB NBS+DATAPAC

PLOT7 Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT7(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX) | On-line doc: CALL GAMSDOC PLOT7 (or @PRT DATAPAC+DOC.PLOT7) | Access: LIB NBS+DATAPAC

PLOT8 Yields a one-page printer plot of Y(I) versus X(I) with special plot characters for a subset of the data with specified axis limits. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL PLOT8(Y,X,CHAR,N,YMIN,YMAX,XMIN,XMAX,D,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLOT8 (or @PRT DATAPAC+DOC.PLOT8) | Access: LIB NBS+DATAPAC

PLOT9 Yields a one-page printer plot of Y(I) versus X(I) with special plot characters and for specified axis limits and axis labels. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c2 Q1 | Usage: CALL


PLOTy Yields a one-page printer plot of Y(I) versus X(I) with special characters for a subset of the data. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c1 Q1 | Usage: CALL PLOTTY(Y,X,CHAR,N,DMIN,DMAX) | On-line doc: CALL GAMSDOC PLOTTY(Y,X,CHAR,N,DMIN,DMAX) | Access: LIB NBS+DATAPAC


PLOTU Produces 4 plots: data plot (X(I) versus Y(I)), autoregression plot (X(I) versus X(I-l)), histogram, and normal probability plot. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 Q1 | Usage: CALL PLOTU(X,Y,N) | On-line doc: CALL GAMSDOC PLOTU(X,Y,N) | Access: LIB NBS+DATAPAC


PLOTXX Yields a one-page printer plot of X(I) versus X(I) for testing autocorrelation. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3c5 Q1 | Usage: CALL PLOTXX(X,Y,N) | On-line doc: CALL GAMSDOC PLOTXX(X,Y,N) | Access: LIB NBS+DATAPAC

PLR Stepwise logistic regression for binary dependent variable and categorical (design variables are formed) and continuous independent variables, using either maximum likelihood or approximate asymptotic estimates for stepping. Three options for generating design variables, plots, interactive stepping. | Proprietary stand-alone program using BMDP command language. | Class(es): L8a4d | Usage: @NBS*PLB$.*BMDP BMDPLR | On-line doc: @NBS*PLB$.*BMDP BMDPLR | Access: @NBS*PLB$.*BMDP BMD<program name>


POICH Pochhammer's symbol (a) sub x = Gamma(a+x) / Gamma(a). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DPOICH. | Class(es): C1 | Usage: Y = POICH(A,X) | On-line doc: CALL GAMSDOC POICH (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

POICH1 Pochhammer's symbol from first order, ((a) sub x -1) / x. Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DPOICH. | Class(es): C1 | Usage: Y = POICH1(A,X) | On-line doc: CALL GAMSDOC POICH1 (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

POICDF Computes the cumulative distribution function value at X for the Poisson distribution with tail length parameter ALAMBA. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a1p | Usage: CALL POICDF(X,ALAMBA,CDF) | On-line doc: CALL GAMSDOC POICDF (or @PRT DATAPAC+DOC.POIDF) | Access: LIB NBS+DATAPAC

POIPLT Generates a Poisson probability plot with tail length parameter ALAMBA, mean = ALAMBA and standard deviation = sqrt(ALAMBA). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5c4p | Usage: CALL POIPLT(X,N,ALAMBA) | On-line doc: CALL GAMSDOC POIPLT (or @PRT DATAPAC+DOC.POIPLT) | Access: LIB NBS+DATAPAC

POIPFF Computes the percent point function value at P for the Poisson distribution with mean and standard deviation = sqrt(ALAMBA). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L5a2p | Usage: CALL POIPFF(P,ALAMBA,PPF) | On-line doc: CALL GAMSDOC POIPFF (or @PRT DATAPAC+DOC.POIPFF) | Access: LIB NBS+DATAPAC

POIRAN Generates a random sample of size N from the Poisson distribution with mean = ALAMBA and standard deviation = sqrt(ALAMBA). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a16 | Usage: CALL POIRAN(N,ALAMBA,ISTART,X) | On-line doc: CALL GAMSDOC POIRAN (or @PRT DATAPAC+DOC.POIRAN) | Access: LIB NBS+DATAPAC


PRAND Generates K pseudo-random numbers from the Poisson distribution with specified population mean K. | Command in MINITAB Proprietary interactive system. Class(es): L6a16 | Usage: PRANDom K Poisson observations with population mean = K, put into C | On-line doc: HELP PRANDOM (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

PROPR Computes the sample proportion which is the proportion of data between XMIN and XMAX (inclusive) in the input vector X. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1d | Usage: CALL PROPR(X,N,XMIN,XMAX,IWRITE,XPROP) | On-line doc: CALL GAMSDOC PROPR (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+DATAPAC

PSI psi (digamma) = gamma'(x) / gamma(x). Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. Double precision version is DPSI. | Class(es): C7c | Usage: Y = PSI(X) | On-line doc: CALL GAMSDOC PSI (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

PUMB Given a basic mesh, this subdivides each interval into a uniform but variable number of points. Multiplicities can occur. Proprietary
single precision Fortran subprogram in PORT library. Double precision version is DPUMB. | Class(es): E3 K6 | Usage: CALL PUMB (XB,NXB,NA,K,X,NX) | On-line doc: CALL GAMSDOC PUMB (or @PRT PORT+DOC.PUMB) | Access: LIB NBS+PORT

**PUMD** Given a basic mesh, this subdivides each interval into a uniform but variable number of points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DPUMD. | Class(es): E3 K6 | Usage: CALL PUMD (XB,NXB,NA,X,NX) | On-line doc: CALL GAMSDOC PUMD (or @PRT PORT+DOC.PUMD) | Access: LIB NBS+PORT
Q1DA: Automatic evaluation of a user-defined function of one variable. Special features include randomization and singularity weakening. Portable single precision Fortran subroutine in Q1DA subdirectory of CMLIB library. 

Q1DAX: Flexible subroutine for the automatic evaluation of definite integrals of a user-defined function of one variable. Special features include randomization, singularity weakening, restarting, specification of an initial mesh (optional), and output of smallest and largest integrand values. Portable single precision Fortran subroutine in Q1DA subdirectory of CMLIB library. 

Q1DB: Automatic evaluation of a user-defined function of one variable. Integral must be a Fortran Function but user may select name. Special features include randomization and singularity weakening. Intermediate in usage difficulty between Q1DA and Q1DAX. Portable single precision Fortran subroutine in Q1DA subdirectory of CMLIB library. Users can call: CALL Q1DAX(F,A,B,EPS,R,E,NINT,RST,W,NMAX,FMIX,FMX,KP,IFLAG) On-line doc: CALL GAMSDOC Q1DA (or @PRT CMLIB+DOC.Q1DA/Q1DA) Tests: CMLIB+TEST-SOURCE.3Q/Q1DA | Access: Lib NBS+CMLIB

QAG: Automatic adaptive integrator, will handle many non-smooth integrands using Gauss Kronrod formulas. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAG. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAG. Users can call: CALL QAG(F,A,B,EPABS,EPREL,KEY,RESULT,ABERR,NEVAL,IER,LIMIT,LENW,LAST,W,WORK) On-line doc: CALL GAMSDOC QAG (or @PRT CMLIB+DOC.QAG/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.QAG/QUADSP, CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

QAGE: Automatic adaptive integrator for semi-infinite or infinite intervals. Uses nonlinear transformation and extrapolation. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAG. Users can call: CALL QAG(F,B,BOUND,INF,EPABS,EPREL,RESULT,ABERR,NEVAL,IER,LIMIT,LENW,LAST,W,WORK) On-line doc: CALL GAMSDOC QAGE (or @PRT CMLIB+DOC.QAGE/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.QAGE/QUADSP, CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

QAGI: Automatic adaptive integrator for semi-infinite or infinite intervals and general integrands, provides more information than QAG. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAGI. Users can call: CALL QAGI(F,BOUND,INF,EPABS,EPREL,RESULT,ABERR,NEVAL,IER,LIMIT,LENW,LAST,W,WORK) On-line doc: CALL GAMSDOC QAGI (or @PRT CMLIB+DOC.QAGI/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.QAGI/QUADSP, CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

QAGP: Automatic adaptive integrator, allows user to specify location of singularities or difficulties of integrand, uses extrapolation. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAGP. Users can call: CALL QAGP(F,A,B,NPTS2,POINTS,EPABS,EPREL,RESULT,ABERR,NEVAL,IER,LENW,LNW,LENW,W,WORK) On-line doc: CALL GAMSDOC QAGP (or @PRT CMLIB+DOC.QAGP/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.QAGP/QUADSP, CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

QAGPE: Automatic adaptive integrator for function with user specified endpoint singularities, provides more information than QAGP. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAGPE. Users can call: CALL QAGPE(F,A,B,NPTS2,POINTS,EPABS,EPREL,RESULT,ABERR,NEVAL,IER,AliST,BlIST,RLIST,PLIST,RTS,ORD,LEVEL,NDIN,LAST) On-line doc: CALL GAMSDOC QAGPE (or @PRT CMLIB+DOC.QAGPE/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.QAGPE/QUADSP, CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

QAGS: Automatic adaptive integrator, will handle most non-smooth integrands including those with endpoint singularities, uses extrapolation. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAGS. Users can call: CALL QAGS(F,A,B,EPABS,EPREL,RESULT,ABERR,NEVAL,IER,LIMIT,LENW,LAST,W,WORK) On-line doc: CALL GAMSDOC QAGS (or @PRT CMLIB+DOC.QAGS/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.QAGS/QUADSP, CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

QAGSE: Automatic adaptive integrator, can handle integrands with endpoint singularities provides more information than QAGS. Portable single precision Fortran subroutine in QUADSP subdirectory of CMLIB library. Double precision version is DQAGSE. Users can call: CALL QAGSE(F,A,B,EPABS,EPREL,RESULT,ABERR,NEVAL,IER,ALIstown,RLIST,ELIST,PLIST,RTS,ORD,LEVEL,NDIN,NDlast) On-line doc: CALL GAMSDOC QAGSE (or @PRT CMLIB+DOC.QAGSE/QUADSP and CMLIB+DOC.SUMMARY/QUADSP) Tests: CMLIB+TEST-SOURCE.3Q/QUADSP | Access: Lib NBS+CMLIB

C 138  GAMS: Module Dictionary  January 1984
QAWC  Cauchy principal value integrator, using adaptive Clenshaw Curtis method (real Hilbert transform). Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWC.  
Class(es): H2a2a1 J4  
Usage: CALL QAWCF(F,A,B,C,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWC (or @PRT CMLIB+DOC.QAWC/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWC/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWE  Cauchy Principal value integrator, provides more information than QAWC (real Hilbert transform). Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWE.  
Class(es): H2a2a1 J4  
Usage: CALL QAWE(F,A,B,C,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWE (or @PRT CMLIB+DOC.QAWE/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWE/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWF  Automatic integrator for Fourier integrals on (a, infinity) with factors sin(OMEGA*X), cos(OMEGA*X) by integrating between zeros. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWF.  
Class(es): H2a3a1  
Usage: CALL QAWF(F,A,OMEGA,INTEGR,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWF (or @PRT CMLIB+DOC.QAWF/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWF/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWF  Automatic integrator for Fourier integrals, with sin(OMEGA*X) factor on (A, INFINITY), provides more information than QAWF. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWF.  
Class(es): H2a3a1  
Usage: CALL QAWFE(F,A,OMEGA,INTEGR,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWFE (or @PRT CMLIB+DOC.QAWFE/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWFE/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWO  Automatic adaptive integrator for integrands with oscillatory sin or cosine factor. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWO.  
Class(es): H2a2a1  
Usage: CALL QAWOE(F,A,B,OMEGA,INTEGR,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWO (or @PRT CMLIB+DOC.QAWO/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWO/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWOE  Automatic integrator for integrands with explicit oscillatory sin or cosine factor, provides more information than QAWO. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWE.  
Class(es): H2a2a1  
Usage: CALL QAWOE(F,A,B,OMEGA,INTEGR,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAW& (or @PRT CMLIB+DOC.QAWE/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWE/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWS  Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWS.  
Class(es): H2a2a1  
Usage: CALL QAWS(F,A,B,OMEGA,INTEGR,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWS (or @PRT CMLIB+DOC.QAWS/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWS/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QAWS  Automatic integrator for functions with explicit algebraic and/or logarithmic endpoint singularities, more information than QAWS. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQAWS.  
Class(es): H2a2a1  
Usage: CALL QAWS(F,A,B,OMEGA,INTEGR,epsabs,epsrel,result,abserr,neval,ier,limit,ienum)  
On-line doc: CALL GAMSDOC QAWS (or @PRT CMLIB+DOC.QAWS/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QAWS/QUADSP, CMLIB+TEST-SOURCE.Q/ref/QUADSP  
Access: Lib NBS+CMLIB

QC25C  Uses 25 point Clenshaw-Curtis formula to estimate integral of F*W where W=1/(X-C). Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQC25C.  
Class(es): H2a2a2 J4  
Usage: CALL QC25CF (A,B,C,RESULT,ABSERR,KRUL,NEVAL)  
On-line doc: CALL GAMSDOC QC25C (or @PRT CMLIB+DOC.QC25C/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QC25C/QUADSP  
Access: Lib NBS+CMLIB

QC25F  Clenshaw-Curtis integration rule for function with cos or sin factor, also uses Gauss Kronrod formula. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQC25F.  
Class(es): H2a2a2  
Usage: CALL QC25FF (A,B,OMEGA,INTEGR,NROMOM,MAXP1,KSME,RESULT,ABSERR,NEVAL,RESABS,RESASC,MOMCOM,CHEOBO)  
On-line doc: CALL GAMSDOC QC25F (or @PRT CMLIB+DOC.QC25F/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QC25F/QUADSP  
Access: Lib NBS+CMLIB

QC25S  Estimates integral of function with algebraico-logarithmic singularities with 25 point Clenshaw-Curtis formula and gives error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQC25S.  
Class(es): H2a2a2  
Usage: CALL QC25SSF (A,B,OMEGA,INTEGR,NROMOM,MAXP1,KSME,RESULT,ABSERR,NEVAL,RESABS,RESASC,MOMCOM,CHEOBO)  
On-line doc: CALL GAMSDOC QC25S (or @PRT CMLIB+DOC.QC25S/QUADSP and CMLIB+DOC.SUMMARY/QUADSP)  
Tests: CMLIB+TEST-SOURCE.QC25S/QUADSP  
Access: Lib NBS+CMLIB
GAMS: Module Dictionary

January 1984

QK15 Evaluates integral of given function on an interval with a 15 point Gauss Kronrod formula and returns error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK15. 

QK15I Evaluates integral of given function on semi-infinite or infinite interval with a transformed 15 point Gauss Kronrod formula and gives error estimates. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK15I. 

QK15W Evaluates integral of given function times arbitrary weight function on interval with 15 point Gauss Kronrod formula and gives error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK15W. 

QK21 Evaluates integral of given function on an interval with a 21 point Gauss Kronrod formula and returns error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK21. 

QK31 Evaluates integral of given function on an interval with a 31 point Gauss Kronrod formula and returns error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK31. 

QK41 Evaluates integral of given function on an interval with a 41 point Gauss Kronrod formula and returns error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK41. 

QK51 Evaluates integral of given function on an interval with a 51 point Gauss Kronrod formula and returns error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK51. 

QK61 Evaluates integral of given function on an interval with a 61 point Gauss Kronrod formula and returns error estimate. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is QK61. 

QMOMO Computes integral of k-th degree Tchebycheff polynomial times selection of functions with various singularities. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQMOMO. 

QNG Automatic non-adaptive integrator for smooth functions, using Gauss Kronrod Patterson formulas. Portable single precision Fortran subprogram in QUADSP sublibrary of CMLIB library. Double precision version is DQNG. 

QUAD Finds the integral of a general user defined EXTERNAL function by an adaptive technique to given absolute accuracy. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DQUAD. 
Class(es): H2a1a1 | Usage: CALL QUAD(F,A,B,EPASBS,EPBREL,RESULT,ABSERR,NEVAL) On-line doc: CALL GAMSDOC QUAD (or PORT PORT+DOC.QUAD) | Access: LIB NBS+PORT

QZHES The first step of the QZ algorithm for solving generalized matrix eigenproblems. Accepts a pair of real general matrices and reduces one of them to upper Hessenberg form and the other to upper triangular form using orthogonal transformations. Usually followed by QZIT, QZVAL, QZVEC. Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. 
Class(es): D4eb3 | Usage: CALL QZHES(NM,N,A,B,MATZ,2) On-line doc: CALL GAMSDOC QZHES (or PORT CMLIB+DOC.QZHES/EISPACK) | Access: LIB NBS+CMLIB | See also: QZIT QZVAL QZVEC
QZIT  The second step of the QZ algorithm for generalized eigenproblems. Accepts an upper Hessenberg and an upper triangular matrix and reduces the former to quasi-triangular form while preserving the form of the latter. Usually preceded by QZHES and followed by QZVAL and QZVEC. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4clb3 | Usage: CALL QZIT(NM,N,A,B,EPS1,MATZ,Z,IERR) | On-line doc: CALL GAMSDOC QZIT (or @PRT CMLIB*DOC.QZIT/EISPACK) | Access: LIB NBS*CMLIB | See also: QZHES QZVAL QZVEC

QZVAL  The third step of the QZ algorithm for generalized eigenproblems. Accepts a pair of real matrices, one in quasi-triangular form and the other in upper triangular form and computes the eigenvalues of the associated eigenproblem. Usually preceded by QZHES, QZIT, and followed by QZVEC. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c2c | Usage: CALL QZVAL(NM,N,A,B,ALFR,ALF1,BETA,MATZ,Z) | On-line doc: CALL GAMSDOC QZVAL (or @PRT CMLIB*DOC.QZVAL/EISPACK) | Access: LIB NBS*CMLIB | See also: QZHES QZIT QZVEC

QZVEC  The optional fourth step of the QZ algorithm for generalized eigenproblems. Accepts a matrix in quasi-triangular form and another in upper triangular form and computes the eigenvectors of the triangular problem and transforms them back to the original coordinates. Usually preceded by QZHES, QZIT, QZVAL. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c3 | Usage: CALL QZVEC(NM,N,A,B,ALFR,ALFI,BETA,Z) | On-line doc: CALL GAMSDOC QZVEC (or @PRT CMLIB*DOC.QZVEC/EISPACK) | Access: LIB NBS*CMLIB | See also: QZHES QZIT QZVAL
R

RIMACH Provides single precision machine dependent information, e.g. RIMACH(4) returns machine epsilon. | Portable single precision Fortran subprogram in MACHCONST sublibrary of CMLIB library. Double precision version is DIAMACH. | Class(es): R1 | Usage: R=RIMACH(I) | On-line doc: CALL GAMSDOC RIMACH (or @PRT CMLIB+DOC.RIMACH/MACHCONST) | Tests: CMLIB+TEST-SOURCE.$Q/MACHCONST | Access: LIB NBS+CMLIB

RIMACH Provides the single precision machine-dependent constants required to adapt FORTRAN library programs to individual computers. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DIAMACH. | Class(es): R1 | Usage: X = RIMACH(I) | On-line doc: CALL GAMSDOC RIMACH (or @PRT PORT+DOC.RIMACH) | Access: LIB NBS+PORT

RANBYT Returns the real random variate generated by UNI, together with its bit pattern presented in four 8-bit bytes. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): L6a21 | Usage: CALL RANBYT (UNI,IBYTE) | On-line doc: CALL GAMSDOC RANBYT (or @PRT PORT+DOC.RANBYT) | Access: LIB NBS+PORT | See also: UNI,RANSET

RAND Uniform random number on [0,1]. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): L6a21 | Usage: Y = RAND (X) | On-line doc: CALL GAMSDOC RAND (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

RANGE Computes the sample range of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b | Usage: CALL RANGE (X,N,WRITE,XRANGE) | On-line doc: CALL GAMSDOC RANGE (or @PRT DATAPAC+DOC.RANGE) | Access: LIB NBS+DATAPAC

RANK Ranks (in ascending order) the N elements of the single precision vector X, and puts the resulting N ranks into the vector XR. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N6a1b1 | Usage: CALL RANK(X,N,XR) | On-line doc: CALL GAMSDOC RANK (or @PRT DATAPAC+DOC.RANK) | Access: LIB NBS+DATAPAC

RANK Ranks the values in a vector. Ties are assigned the average rank. | Command in MINITAB Proprietary interactive system. | Class(es): N6a1b1 | Usage: RANK the values in C, put ranks into C | On-line doc: HELP RANK (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @QXT NBS+MINITAB,MINITAB (or CALL MINITAB in GTS)

RANPER Generates a random permutation of size N of the values 1, 2, 3, ..., N=1, N. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a16 | Usage: CALL RANPER(N,ISTART,X) | On-line doc: CALL GAMSDOC RANPER (or @PRT DATAPAC+DOC.RANPER) | Access: LIB NBS+DATAPAC

RANSET Initializes the uniform random number generator UNI, to other than the default initial values. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): L6c | Usage: CALL RANSET (ICSEED,ITSEED) | On-line doc: CALL GAMSDOC RANSET (or @PRT PORT+DOC.RANSET) | Access: LIB NBS+PORT | See also: UNI,RANBYT


READ Performs a format-free read. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N1 | Usage: CALL READ(IOCOL1,ICOL2,X,N) | On-line doc: CALL GAMSDOC READ (or @PRT DATAPAC+DOC.READ) | Access: LIB NBS+DATAPAC

READG Performs a format-free read of data from input unit = IRD. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): N1 | Usage: CALL READG(IRD,IOCOL1,IOCOL2,X,N) | On-line doc: CALL GAMSDOC READG (or @PRT DATAPAC+DOC.READG) | Access: LIB NBS+DATAPAC

REBAK Forms eigenvectors of generalized symmetric eigenystem from eigenvectors of derived matrix output from REDUC or REDUC2. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL REBAK(NM,N,B,DL,M,Z) | On-line doc: CALL GAMSDOC REBAK (or @PRT CMLIB+DOC.REBAK/EISPACK) | Access: LIB NBS+CMLIB | See also: REDUC REDUC2

REBAKB Forms eigenvectors of generalized symmetric eigenystem from eigenvectors of derived matrix output from REDUC2. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4c4 | Usage: CALL REBAKB(NM,N,B,DL,M,Z) | On-line doc: CALL GAMSDOC REDUKB (or @PRT CMLIB+DOC.REBAK/EISPACK) | Access: LIB NBS+CMLIB | See also: REDUC REDUC2


GAMS: Module Dictionary

January 1984
REGRESS Performs simple or multiple linear regression, prints standard results. Options: amount of output, save results, weights, handle missing values, through the origin, compute and save regression diagnostics, lack of fit tests. | Command in MINITAB Proprietary interactive system. Class(es): L8a L8a9 Usage: REGRESS C on K predictors C,... C [put standardized residuals in C [fits in C] ]; subcommands NOCONSTANT; WEIGHTS in C; MSE into K; COEF into C; XPARINV into M; RMAHRT into M; H into C; RESIDS into C; TRESIDS into C; COOKD into C; DFITS into C; VIF; PURE error lack of fit test; XLOF experimental lack of fit. | On-line doc: HELP REGRESS (in Minotab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

RELSD Computes the sample relative standard deviation of the data in the vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L1a1b Usage: CALL RELSD(X,N,WRITE,XRELSD) | On-line doc: CALL GAMSDOC RELSD (or @PRT DATAPAC+DOC.RELSD) | Access: LIB NBS+DATAPAC

REPLAC Replaces (with the value XNEW) all observations in the vector X which are inside the interval [XMIN, XMAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2a Usage: CALL REPLAC(X,N,XMIN,XMAX,XNEW) | On-line doc: CALL GAMSDOC REPLAC (or @PRT DATAPAC+DOC.REPLAC) | Access: LIB NBS+DATAPAC

RETAIn Retains all observations in the vector X which are inside the interval [XMIN, XMAX]. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L2d Usage: CALL RETAIN(X,N,XMIN,XMAX,NEWN) | On-line doc: CALL GAMSDOC RETAIN (or @PRT DATAPAC+DOC.RETAIN) | Access: LIB NBS+DATAPAC

RETSRC Test and reset error recovery mode for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R3a Usage: CALL RETSRC (IROLD) | On-line doc: CALL GAMSDOC RETSRC (or @PRT PORT+DOC.RETSRC) | Access: LIB NBS+PORT


RFFTI Initialize WSAVE array for SUBROUTINE RFFTF and RFFTB. | Portable single precision Fortran subprogram in FFFTPKG sublibrary of CMLIB library. | Class(es): J1a1 Usage: CALL RFTI(N,WSAVE) | On-line doc: CALL GAMSDOC RFFTI (or @PRT CMLIB+DOC.RFTI/FFTPKG) | Tests: CMLIB+TEST-SOURCE.$Q/FFTPKG | Access: LIB NBS+CMLIB | See also: RFFTF, RFFTI

RG Computes eigenvalues and, optionally, eigenvectors of a real general matrix. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a2 Usage: CALL RG(NM,N,A,WR,RI,MA,IV1,FV1,IERR) | On-line doc: CALL GAMSDOC RG (or @PRT CMLIB+DOC.RG/EISPACK) | Access: LIB NBS+CMLIB

RGAUSS Normal random number. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): L6a14 Usage: Y = RGAUSS (XMEAN,SD) | On-line doc: CALL GAMSDOC RGAUSS (or @PRT FNLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

RGG Computes eigenvalues and eigenvectors of a general real generalized problem: $A \times (\lambda) = (B) \times X$. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4b2 Usage: CALL RG(NM,N,A,ALP,ALF,BETA,MA,IV1,FV1,IERR) | On-line doc: CALL GAMSDOC RG (or @PRT CMLIB+DOC.RG/EISPACK) | Access: LIB NBS+CMLIB


RLDCVA Variance estimates for decoded orthogonal polynomial regression coefficients. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a2b1 Usage: CALL RLDCVA (V,DA,B,SC,T,IT,IER) | On-line doc: CALL GAMSDOC RLDCVA (or @PRT IMSL+DOC.RLDCVA) | Access: LIB NBS+IMSL | See also: RLDCV

RLDCW Variances of coded orthogonal polynomial regression coefficients - for usage in conjunction with IMSL routines RLFOOT and RLFOTW, and provided to prepare input for IMSL routine RLDCVA. | Proprietary single precision Fortran subprogram in IMSL
GAMS: Module Dictionary
January 1984

<table>
<thead>
<tr>
<th>Class(es):</th>
<th>Usage:</th>
<th>Access:</th>
</tr>
</thead>
<tbody>
<tr>
<td>L8h</td>
<td>CALL RLDCW (SSE,X,W,N,ID,IOPT,A,B,V,P,IP,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a0a</td>
<td>CALL RLFITI (X,N,M,I,X,Y,L,Y,SS,NDF,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a10a</td>
<td>CALL RLFIT (X,N,M,I,Y,L,T,W,S,ND,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a2b1</td>
<td>CALL RLFOTW (X,Y,N,RSQ,MD,W,ID,P,C,S,A,B,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a2b2</td>
<td>CALL RLGQMI (X,N,M,I,XBAR)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a4alc</td>
<td>CALL RLINPF (CRIT,IOPT,STAT,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8h4</td>
<td>CALL RLLAV (X,Y,N,M,IOPT,BETA,SUM,ITER,IRANK,IKW,IK,D)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a4alc</td>
<td>CALL RLMUL (A,X,Y,BAR,N,M,ALFA,ANOVA,B,I,B,V,R,IP,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
<tr>
<td>L8a1a</td>
<td>CALL RLONE (X,Y,N,M,IOPT,PRED,ALBAP,DES,ANOVA,STAT,PRED,IP,NN,IER)</td>
<td>LIB NBS+IMSL</td>
</tr>
</tbody>
</table>

RLPOL  Generate orthogonal polynomials with the associated constants AA and BB.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): L8a2b L2a C3a2  | Usage: CALL RLPOL (X,N,1D,SM,SA,AA,BB,P,IP,IER)  | On-line doc: CALL GAMSDOC RLPOL (or @PRT IMSL+DOC.RLPOL)  | Access: LIB NBS+IMSL

RLPRDI  Confidence intervals for the true response and for the average of a set of future observations on the response - in-core version.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): L8a10c  | Usage: CALL RLPRDI (Y,V,N,TS,SR,C,IC)  | On-line doc: CALL GAMSDOC RLPRDI (or @PRT IMSL+DOC.RLPRDI)  | Access: LIB NBS+IMSL

RLPRDO  Confidence intervals for the true response and for the average of a set of future observations on the response - out-of-core version.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): L8a10c  | Usage: CALL RLPRDO (Y,V,TS,SR,C)  | On-line doc: CALL GAMSDOC RLPRDO (or @PRT IMSL+DOC.RLPRDO)  | Access: LIB NBS+IMSL

RLRES  Perform a residual analysis for a fitted regression model.  | Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): L8a10b  | Usage: CALL RLRES (XY,I,X,MM,N,NH,M,BETA,SDR,RES,IR,IER)  | On-line doc: CALL GAMSDOC RLRES (or @PRT IMSL+DOC.RLRES)  | Access: LIB NBS+IMSL


RLTR  An auxiliary routine for use together with FFT to transform 2N real data points. Uses less storage than FFTR.  | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DRLTR.  | Class(es): J1a1  | Usage: CALL RLTR (A,B,N,F,SN)  | On-line doc: CALL GAMSDOC RLTR (or @PRT PORT+DOC.RLTR)  | Access: LIB NBS+PORT


ROOTOOG Prints a sampled rootogram, i.e. a histogram which has been fitted with a Gaussian distribution based on square roots of the counts of data values in each bin and which uses medians and hinges. Options: specify Gaussian mean and standard deviation, save results.  | Command in MINITAB Proprietary interactive system.  | Class(es): L3d  | Usage: ROOTOOG data in C [use bin boundaries in C]; subcommands BOUNDARIES into C; DRRS into C; FITTED values into C; COUNTS into C; FREQUENCIES are in C [bin boundaries are in C]; MEAN = K; STDEV = K.  | On-line doc: HELP ROOTOOG (in Min tab)  | Tests: MINITAB+TEST-SOURCE.  | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)


RPOLY  Finds zeros of a polynomial with real coefficients. Output zeros are in a pair of arrays, for real and imaginary part. Portable single precision Fortran subprogram in PORT library.  | Class(es): F1a1a  | Usage: CALL RPOLY (DEGREE,COEFF,ZERO,ZERO)  | On-line doc: CALL GAMSDOC RPOLY (or @PRT PORT+DOC.RPOLY)  | Access: LIB NBS+PORT


RPZERO  Computes all the zeros of a polynomial with real coefficients. Error bounds are also computed, Uses Newton's Method for systems. Portable single precision Fortran subprogram in CPZERO sublibrary of CMLIB library.  | Class(es): F1a1a  | Usage: CALL RPZERO(N,A,T,IPFLAG,S)  | On-line doc: CALL GAMSDOC RPZERO (or @PRT CMLIB+DOC.SUMMARY/CPZERO and CMLIB+DOC.RPZERO/CPZERO)  | Tests: CMLIB+TEST-SOURCE.RPZERO/CPZERO  | Access: LIB NBS+CMLIB

RQUAD  Finds the integral of a general user defined EXTERNAL function by an adaptive technique. Combined absolute and rela-
GAMS: Module Dictionary

January 1984

tive error control. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DRQUAD. |
Class(es): H2a1a | Usage: CALL RQUAD (F,A,B,EPABS,EPREL,ANS,ERREST) | On-line doc: CALL GAMSDOC RQUAD (or @PRT PORT*DOC.RQUAD) | Access: LIB NBS+PORT


RSMITZ Least squares fit of the non-linear regression model y(i) = alpha+beta*gamma*+x(i)+e(i). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): L8a1a | Usage: CALL RSMITZ (X,I,X,Y,VAR,STAT,VCV,ITER,IER) | On-line doc: CALL GAMSDOC RSMITZ (or @PRT IMSL+DOC.RSMITZ) | Access: LIB NBS+IMSL

SMOOTH Computes resistant smoother by 2435H, twice (or 3RSSH, twice), i.e. successive application of running medians and Hanning (running weighted averages), and save results. | Command in MINITAB Proprietary interactive system. | Class(es): L8f | Usage: CALL SMOOTH C, put rough into C, smooth into C | subcommand SMOOTH 3RSSH, twice.] | On-line doc: HELP SMOOTH (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @QRT NBS+MINITAB.MINITAB (or CALL MINITAB in GTS)

RSP Compute eigenvalues and, optionally, eigenvectors of a real symmetric matrix packed into a one dimensional array. | Portable single precision Fortran subprogram in EISPACK sublibrary of CMLIB library. | Class(es): D4a1a | Usage: CALL RSP(N,M,N,NV,A,W,MATZ,Z,FV1,FV2,IER) | On-line doc: CALL GAMSDOC RSP (or @PRT CMLIB+DOC.RSP/EISPACK) | Access: LIB NBS+CMLIB


RUNIF Sequence of uniform random numbers on [0,1]. | Portable single precision Fortran subprogram in FNLIB sublibrary of CMLIB library. | Class(es): L6a21 | Usage: Y = RUNIF (T,N) | On-line doc: CALL GAMSDOC RUNIF (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

RUNS Performs a run analysis of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L4a1d | Usage: CALL RUNS(X,N) | On-line doc: CALL GAMSDOC RUNS (or @PRT DATAPAC+DOC.RUNS) | Access: LIB NBS+DATAPAC

NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

RYORK  Estimates simple linear regression coefficients when both variables are subject to errors which are not necessarily homogeneous in variance.  Portable single precision Fortran subprogram in SLRPACK sublibrary of CMLIB library.

S

S07AAE  

S07AAF  

S09AAE  

S09AAF  

S09ABE  

S10AAE  

S10AAF  

S10ABE  

S10ABF  

S10ACE  

S10ACF  

S11AAE  

S11AAF  

S11ABE  

S11ABF  

S11ACE  

S11ACF  

S13AAE  

S13AAF  

S13ACE  

S13ACF  

S13ADE
January 1984  GAMS: Module Dictionary  C 149

LIB NBS+NAG


S15AABE Cumulative normal distribution function, p(x). | Proprietary single precision Fortran subroutine in NAG library. Double precision version is S15ABF. | Class(es): C8a L5a1n | Usage: D = S15ABE (X, IFAIL) | On-line doc: CALL GAMSDOC S15ABE (or ®PRT NAG*DOC.S15ABE) | Access: LIB NBS+NAG


S15ACE Complement of cumulative normal distribution function, q(x). | Proprietary single precision Fortran subroutine in NAG library. Double precision version is S15ACE. | Class(es): C8a L5a1n | Usage: D = S15ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S15ACE (or ®PRT NAG*DOC.S15ACE) | Access: LIB NBS+NAG

S15ACF Complement of cumulative normal distribution function, q(x). | Proprietary double precision Fortran subroutine in NAG library. Single precision version is S15ACE. | Class(es): C8a L5a1n | Usage: D = S15ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S15ACF (or ®PRT NAG*DOC.S15ACF) | Access: LIB NBS+NAG

S15ADE Complement of error function, erfc(x). | Proprietary single precision Fortran subroutine in NAG library. Double precision version is S15ADF. | Class(es): C8a L5a1t | Usage: D = S15ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S15ADE (or ®PRT NAG*DOC.S15ADE) | Access: LIB NBS+NAG


S15AEE Error function, erf(x). | Proprietary single precision Fortran subroutine in NAG library. Double precision version is S15AEF. | Class(es): C8a L5a1t | Usage: D = S15AEE (X, IFAIL) | On-line doc: CALL GAMSDOC S15AEE (or ®PRT NAG*DOC.S15AEE) | Access: LIB NBS+NAG | See also: S15ADE

S15AEF Error function, erf(x). | Proprietary double precision Fortran subroutine in NAG library. Single precision version is S15AEE. | Class(es): C8a L5a1t | Usage: D = S15AEF (X, IFAIL) | On-line doc: CALL GAMSDOC S15AEF (or ®PRT NAG*DOC.S15AEF) | Access: LIB NBS+NAG | See also: S15ADF


S17ACE Bessel functions, y0(x). | Proprietary single precision Fortran subroutine in NAG library. Double precision version is S17ACF. | Class(es): C10a1 | Usage: D = S17ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S17ACE (or ®PRT NAG*DOC.S17ACE) | Access: LIB NBS+NAG

S17ACF Bessel functions, y0(x). | Proprietary double precision Fortran subroutine in NAG library. Single precision version is S17ACE. | Class(es): C10a1 | Usage: D = S17ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S17ACF (or ®PRT NAG*DOC.S17ACF) | Access: LIB NBS+NAG

S17ADE Bessel functions, y1(x). | Proprietary single precision Fortran subroutine in NAG library. Double precision version is S17ADF. | Class(es): C10a1 | Usage: D = S17ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S17ADE (or ®PRT NAG*DOC.S17ADE) | Access: LIB NBS+NAG

S17ADF Bessel functions, y1(x). | Proprietary double precision Fortran subroutine in NAG library. Single precision version is S17ADE. | Class(es): C10a1 | Usage: D = S17ADF (X, IFAIL) | On-line doc: CALL GAMSDOC S17ADF (or ®PRT NAG*DOC.S17ADF) | Access: LIB NBS+NAG
S17AEE  Bessel functions, \( j_0(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AEF.

Class(es): C10a1  Usage: D = S17AEE (X, IFAIL)  On-line doc: CALL GAMSDOC S17AEE (or @PRT NAG*DOC.S17AEE)  Access: LIB NBS*NAG

S17AEP  Bessel functions, \( j_0(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AEE.

Class(es): C10a1  Usage: D = S17AEP (X, IFAIL)  On-line doc: CALL GAMSDOC S17AEP (or @PRT NAG*DOC.S17AEP)  Access: LIB NBS*NAG

S17AFE  Bessel functions, \( j_1(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AFF.

Class(es): C10a1  Usage: D = S17AFE (X, IFAIL)  On-line doc: CALL GAMSDOC S17AFE (or @PRT NAG*DOC.S17AFE)  Access: LIB NBS*NAG

S17AFF  Bessel functions, \( j_1(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AFE.

Class(es): C10a1  Usage: D = S17AFF (X, IFAIL)  On-line doc: CALL GAMSDOC S17AFF (or @PRT NAG*DOC.S17AFF)  Access: LIB NBS*NAG

S17AGE  Airy functions, \( a(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AGF.

Class(es): C10d  Usage: D = S17AGE (X, IFAIL)  On-line doc: CALL GAMSDOC S17AGE (or @PRT NAG*DOC.S17AGE)  Access: LIB NBS*NAG

S17AGF  Airy functions, \( a(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AGE.

Class(es): C10d  Usage: D = S17AGF (X, IFAIL)  On-line doc: CALL GAMSDOC S17AGF (or @PRT NAG*DOC.S17AGF)  Access: LIB NBS*NAG

S17AHE  Airy functions, \( b(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AHF.

Class(es): C10d  Usage: D = S17AHE (X, IFAIL)  On-line doc: CALL GAMSDOC S17AHE (or @PRT NAG*DOC.S17AHE)  Access: LIB NBS*NAG

S17AHF  Airy functions, \( b(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AHE.

Class(es): C10d  Usage: D = S17AHF (X, IFAIL)  On-line doc: CALL GAMSDOC S17AHF (or @PRT NAG*DOC.S17AHF)  Access: LIB NBS*NAG

S17AJE  Airy functions, \( a'(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AJF.

Class(es): C10d  Usage: D = S17AJE (X, IFAIL)  On-line doc: CALL GAMSDOC S17AJE (or @PRT NAG*DOC.S17AJE)  Access: LIB NBS*NAG

S17AJF  Airy functions, \( a'(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AJE.

Class(es): C10d  Usage: D = S17AJF (X, IFAIL)  On-line doc: CALL GAMSDOC S17AJF (or @PRT NAG*DOC.S17AJF)  Access: LIB NBS*NAG

S17AKE  Airy functions, \( b'(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S17AKF.

Class(es): C10d  Usage: D = S17AKE (X, IFAIL)  On-line doc: CALL GAMSDOC S17AKE (or @PRT NAG*DOC.S17AKE)  Access: LIB NBS*NAG

S17AKF  Airy functions, \( b'(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S17AKE.

Class(es): C10d  Usage: D = S17AKF (X, IFAIL)  On-line doc: CALL GAMSDOC S17AKF (or @PRT NAG*DOC.S17AKF)  Access: LIB NBS*NAG

S18ACE  Modified Bessel functions, \( K_0(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18ACF.

Class(es): C10b1  Usage: D = S18ACE (X, IFAIL)  On-line doc: CALL GAMSDOC S18ACE (or @PRT NAG*DOC.S18ACE)  Access: LIB NBS*NAG

S18ACF  Modified Bessel functions, \( K_0(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18ACE.

Class(es): C10b1  Usage: D = S18ACF (X, IFAIL)  On-line doc: CALL GAMSDOC S18ACF (or @PRT NAG*DOC.S18ACF)  Access: LIB NBS*NAG

S18ADE  Modified Bessel functions, \( K_1(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18ADF.

Class(es): C10b1  Usage: D = S18ADE (X, IFAIL)  On-line doc: CALL GAMSDOC S18ADE (or @PRT NAG*DOC.S18ADE)  Access: LIB NBS*NAG

S18ADF  Modified Bessel functions, \( K_1(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18ADE.

Class(es): C10b1  Usage: D = S18ADF (X, IFAIL)  On-line doc: CALL GAMSDOC S18ADF (or @PRT NAG*DOC.S18ADF)  Access: LIB NBS*NAG

S18AEE  Modified Bessel functions, \( I_0(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18AEP.

Class(es): C10b1  Usage: D = S18AEE (X, IFAIL)  On-line doc: CALL GAMSDOC S18AEE (or @PRT NAG*DOC.S18AEE)  Access: LIB NBS*NAG

S18AEP  Modified Bessel functions, \( I_0(x) \).  Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18AEE.

Class(es): C10b1  Usage: D = S18AEP (X, IFAIL)  On-line doc: CALL GAMSDOC S18AEP (or @PRT NAG*DOC.S18AEP)  Access: LIB NBS*NAG

S18AFE  Modified Bessel functions, \( I_1(x) \).  Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18AFF.

Class(es): C10b1  Usage: D = S18AFE (X, IFAIL)  On-line doc: CALL GAMSDOC S18AFE (or @PRT NAG*DOC.S18AFE)  Access: LIB NBS*NAG
GAMS: Module Dictionary

S18AFF Modified Bessel functions, I\(n(x)\). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S18AFE. | Class(es): C16b1 | Usage: D = S18AFF (X, IFAIL) | On-line doc: CALL GAMSDOC S18AFF (or @PRT NAG+DOC.S18AFF) | Access: LIB NBS+NAG

S18CCE Returns a value of the scaled modified Bessel function, exp(x)K\(sub0(x)\). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CCF. | Class(es): C16b1 | Usage: R = S18CCE(X, IFAIL) | On-line doc: CALL GAMSDOC S18CCE (or @PRT NAG+DOC.S18CCE) | Access: LIB NBS+NAG

S18CCF Returns a value of the scaled modified Bessel function, exp(x)K\(sub0(x)\). | Proprietary single precision Fortran subprogram in NAG library. Single precision version is S18CCE. | Class(es): C16b1 | Usage: R = S18CCF(X, IFAIL) | On-line doc: CALL GAMSDOC S18CCF (or @PRT NAG+DOC.S18CCF) | Access: LIB NBS+NAG

S18CDE Returns a value of the scaled modified Bessel function, exp(x)K\(sub1(x)\). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CDF. | Class(es): C16b1 | Usage: R = S18CDE(X, IFAIL) | On-line doc: CALL GAMSDOC S18CDE (or @PRT NAG+DOC.S18CDE) | Access: LIB NBS+NAG

S18CDF Returns a value of the scaled modified Bessel function, exp(x)K\(sub1(x)\). | Proprietary single precision Fortran subprogram in NAG library. Single precision version is S18CDE. | Class(es): C16b1 | Usage: R = S18CDF(X, IFAIL) | On-line doc: CALL GAMSDOC S18CDF (or @PRT NAG+DOC.S18CDF) | Access: LIB NBS+NAG

S18CEE Returns a value of the scaled modified Bessel function, exp(-|x|)J\(sub0(x)\). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CEF. | Class(es): C16b1 | Usage: R = S18CEE(X, IFAIL) | On-line doc: CALL GAMSDOC S18CEF (or @PRT NAG+DOC.S18CEF) | Access: LIB NBS+NAG

S18CEF Returns a value of the scaled modified Bessel function, exp(-|x|)J\(sub0(x)\). | Proprietary single precision Fortran subprogram in NAG library. Single precision version is S18CEE. | Class(es): C16b1 | Usage: R = S18CEF(X, IFAIL) | On-line doc: CALL GAMSDOC S18CEF (or @PRT NAG+DOC.S18CEF) | Access: LIB NBS+NAG

S18CFE Returns a value of the scaled modified Bessel function, exp(-|x|)J\(sub1(x)\). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S18CFD. | Class(es): C16b1 | Usage: R = S18CFE(X, IFAIL) | On-line doc: CALL GAMSDOC S18CFE (or @PRT NAG+DOC.S18CFE) | Access: LIB NBS+NAG

S18CFD Returns a value of the scaled modified Bessel function, exp(-|x|)J\(sub1(x)\). | Proprietary single precision Fortran subprogram in NAG library. Single precision version is S18CFE. | Class(es): C16b1 | Usage: R = S18CFD(X, IFAIL) | On-line doc: CALL GAMSDOC S18CFD (or @PRT NAG+DOC.S18CFD) | Access: LIB NBS+NAG

S20ACE Fresnel integrals, s(x). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S20ACF. | Class(es): C8b | Usage: D = S20ACE (X, IFAIL) | On-line doc: CALL GAMSDOC S20ACF (or @PRT NAG+DOC.S20ACE) | Access: LIB NBS+NAG

S20ACF Fresnel integrals, s(x). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S20ACE. | Class(es): C8b | Usage: D = S20ACF (X, IFAIL) | On-line doc: CALL GAMSDOC S20ACF (or @PRT NAG+DOC.S20ACF) | Access: LIB NBS+NAG

S20ADE Fresnel integrals, c(x). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S20ADF. | Class(es): C8b | Usage: D = S20ADE (X, IFAIL) | On-line doc: CALL GAMSDOC S20ADF (or @PRT NAG+DOC.S20ADE) | Access: LIB NBS+NAG

S20ADF Fresnel integrals, c(x). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S20ADE. | Class(es): C8b | Usage: D = S20ADF (X, IFAIL) | On-line doc: CALL GAMSDOC S20ADF (or @PRT NAG+DOC.S20ADF) | Access: LIB NBS+NAG

S21BAX Elliptic integrals, degenerate symmetrised integral of 1st kind, rc(x, y). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BAX. | Class(es): C14 | Usage: D = S21BAX (X, Y, IFAIL) | On-line doc: CALL GAMSDOC S21BAX (or @PRT NAG+DOC.S21BAX) | Access: LIB NBS+NAG

S21BAF Elliptic integrals, degenerate symmetrised integral of 1st kind, rc(x, y). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BAE. | Class(es): C14 | Usage: D = S21BAF (X, Y, IFAIL) | On-line doc: CALL GAMSDOC S21BAF (or @PRT NAG+DOC.S21BAF) | Access: LIB NBS+NAG

S21BBE Elliptic integrals, symmetrised integral of 1st kind, rf(x,y,z). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BBF. | Class(es): C14 | Usage: D = S21BBE (X, Y, Z, IFAIL) | On-line doc: CALL GAMSDOC S21BBE (or @PRT NAG+DOC.S21BBE) | Access: LIB NBS+NAG

S21BBF Elliptic integrals, symmetrised integral of 1st kind, rf(x,y,z). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BBE. | Class(es): C14 | Usage: D = S21BBF (X, Y, Z, IFAIL) | On-line doc: CALL GAMSDOC S21BBF (or @PRT NAG+DOC.S21BBF) | Access: LIB NBS+NAG

S21BCE Elliptic integrals, symmetrised integral of 2nd kind, rd(x,y). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BCF. | Class(es): C14 | Usage: D = S21BCE (X, Y, Z, IFAIL) | On-line doc: CALL GAMSDOC S21BCE (or @PRT NAG+DOC.S21BCE) | Access: LIB NBS+NAG
GAMS: Module Dictionary

January 1984

S21BCF
Elliptic integrals, symmetrised integral of 2nd kind, \(rd(x,y,z)\). Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BCE. Class(es): C14 Usage: D = S21BCF (X, Y, Z, IFAIL) On-line doc: CALL GAMSDOC S21BCF (or @PRT NAG+DOC.S21BCF) | Access: LIB NBS+NAG

S21BDE
Elliptic integrals, symmetrised integral of 3rd kind, \(rj(x,y,z)\). Proprietary single precision Fortran subprogram in NAG library. Double precision version is S21BDF. Class(es): C14 Usage: D = S21BDE (X, Y, Z, R, IFAIL) On-line doc: CALL GAMSDOC S21BDE (or @PRT NAG+DOC.S21BDE) | Access: LIB NBS+NAG

S21BDF
Elliptic integrals, symmetrised integral of 3rd kind, \(rt(x,y,z)\). Proprietary double precision Fortran subprogram in NAG library. Single precision version is S21BDE. Class(es): C14 Usage: D = S21BDF (X, Y, Z, R, IFAIL) On-line doc: CALL GAMSDOC S21BDF (or @PRT NAG+DOC.S21BDF) | Access: LIB NBS+NAG

SAMPLE
Randomly selects without replacement values from one or more vectors, optionally carrying along other vectors. Command in MINITAB Proprietary interactive system. Class(es): Li1a10 Usage: SAMPLE K rows from C[...], put into C[...], [On-line doc: HELP SAMPLE (in Minitab)] Tests: MINITAB+TEST-SOURCE. Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

SAMPX
Computes the sample 100P percent point (where P is between 0.0 and 1.0, exclusively) of the data in the input vector X. Portable single precision Fortran subprogram in DATAPAC library. Class(es): Li1a1d Usage: CALL SAMPX(X,N,P,IWRITE,PP) [On-line doc: CALL GAMSDOC SAMPX (or @PRT DATAPAC+DOC.SAMPX)] Access: LIB NBS+DATAPAC

SASUM
Computes single precision sum of absolute values of components of vector. Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DASUM. Class(es): D1a3a Usage: S = SASUM(N,SX,INCX) [On-line doc: CALL GAMSDOC SASUM (or @PRT CMLIB+DOC.SASUM/BLAS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SAXPY
Computes a constant times a vector plus a vector, all single precision. Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DAXPY. Class(es): D1a7 Usage: CALL SAXPY(Y,N,SX,INCX,Y,SY,INCY) [On-line doc: CALL GAMSDOC SAXPY (or @PRT CMLIB+DOC.SAXPY/BLAS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SCALE
Computes 4 estimates of the scale (variation, scatter, dispersion) of the data in the input vector X. Portable single precision Fortran subprogram in DATAPAC library. Class(es): Li1a1b Usage: CALL SCALE(X,N) [On-line doc: CALL GAMSDOC SCALE (or @PRT DATAPAC+DOC.SCALE)] Access: LIB NBS+DATAPAC

SCASUM
Computes complex sum of absolute values of components of vector. Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Class(es): D1a3a Usage: S = SCASUM(N,CX,INCX) [On-line doc: CALL GAMSDOC SCASUM (or @PRT CMLIB+DOC.SCASUM/BLAS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SCHDC
Computes Cholesky decomposition of real positive definite matrix with optional pivoting. Portable single precision Fortran subprogram in LINPACK*s sublibrary of CMLIB library. Double precision version is DCHDC. Class(es): D2b1b Usage: CALL SCHDC(LDA,P,LDRP,WKJ,JPVT,JOV,INFO) [On-line doc: CALL GAMSDOC SCHDC (or @PRT CMLIB+DOC.SCHDC/LINPACKS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SCDIF
Computes Cholesky factorization of real positive definite matrix. Portable single precision Fortran subprogram in LINPACK*s sublibrary of CMLIB library. Double precision version is DCHDC. Class(es): D7b Usage: CALL SCHDIF(R,LDRP,LZ,LZD,NZ,Y,RHO,C,S,INFO) [On-line doc: CALL GAMSDOC SCHDIF (or @PRT CMLIB+DOC.SCHDIF/LINPACKS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SCHEX
Updates Cholesky factorization of real positive definite matrix. Portable single precision Fortran subprogram in LINPACK*s sublibrary of CMLIB library. Double precision version is DCHEX. Class(es): D7b Usage: CALL SCHEX(R,LDRP,K,LZ,LZD,NZ,C,S,JOB) [On-line doc: CALL GAMSDOC SCHEX (or @PRT CMLIB+DOC.SCHEX/LINPACKS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SCHUD

SCNRMM2
Computes the Euclidean length or L2 norm of a complex vector. Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Class(es): D1a3 Usage: S = SCNRMM2(N,CX,INCX) [On-line doc: CALL GAMSDOC SCNRMM2 (or @PRT CMLIB+DOC.SCNRMM2/BLAS)] Tests: CMLIB+TEST-SOURCE. Access: LIB NBS+CMLIB

SCOPY
Copies a vector X to a vector Y, both single precision. Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DCOPY. Class(es): D1a5 Usage: CALL SCOPY(N,SX,INCX,SY,INCY) [On-line doc: CALL
GAMS: Module Dictionary C 153

GAMSDOC SCOPY (or @PRT CMLIB+DOC.SCOPY/BLAS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

SCOPYM Copies negative of array SX into array SY, with corresponding increments INCX and INCY. Portable single precision Fortran subroutine in XBLAS sublibrary of CMLIB library. | Class(es): Dia5 | Usage: CALL SCOPYM(N,SX,INCX,SY,INCY) | On-line doc: CALL GAMSDOC SCOPYM (or @PRT CMLIB+DOC.SCOPYM/XBLAS) | Tests: CMLIB+TEST-SOURCE.$Q/XBLAS | Access: LIB NBS+CMLIB

SCOV Calculates covariance matrix for a nonlinear dating fitting problem. This subroutine is intended to be used after a successful return from either of the subprograms SNLS1 or SNLS1E. Portable single precision Fortran subroutine in SNLS1E sublibrary of CMLIB library. | Class(es): 6-0 | Usage: CALL SCOV(FCN,IOPT,M,N,X,YVEC,R,LDR,INFO,W1,W2,W3,W4) | On-line doc: CALL GAMSDOC SCOV (or @PRT CMLIB+DOC.SCOV/SNLS1E) | Tests: CMLIB+TEST-SOURCE | Access: LIB NBS+CMLIB

SD Computes the sample standard deviation (with denominator N-1) of the data in the input vector X. Portable single precision Fortran subroutine in DATAPAC library. | Class(es): Liatb | Usage: CALL SD(X,N,IWRITE,XSD) | On-line doc: CALL GAMSDOC SD (or @PRT DATAPAC+DOC.SD) | Access: LIB NBS+DATAPAC

SDASL Solves the system of differential/algebraic equations of the form g(t, y, y')=0, with given initial values. Portable single precision Fortran subroutine in SDASL sublibrary of CMLIB library. Double precision version is SDASSL. | Class(es): 11atb | Usage: CALL SDASL(RES,NEQ,T,Y,YPRIME,TOUT,INFO,RTOL,ATOL,ITD,RWORK,LRW,IERWORK,LIW,IRWORK,LPAR,IPAR,IAJ) | On-line doc: CALL GAMSDOC SDASL (or @PRT CMLIB+DOC.SDASL/SDASL) | Tests: CMLIB+TEST-SOURCE | Access: LIB NBS+CMLIB


SET Create a constant vector or a vector of integers in increments of 1 or more or with other patterns. Command in MINITAB Proprietary interactive system. | Class(es): L2 | Usage: SET [K repetitions of] (K,...,K) [repeated K times] into column C | On-line doc: HELP SET (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)
SETC  Set a specified number of values in a complex array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a1 | Usage: CALL SETC (N,V,B) | On-line doc: CALL GAMSDOC SETC (or @PRT PORT*DOC.SETC) | Access: LIB NBS*PORT

SETD  Set a specified number of values in a double precision array equal to a constant. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SETR. | Class(es): D1a1 | Usage: CALL SETD (N,V,B) | On-line doc: CALL GAMSDOC SETD (or @PRT PORT*DOC.SETD) | Access: LIB NBS*PORT

SETERR  Sets the error indicator and depending on options prints a message and provides a dump for PORT library programs. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1c3 | Usage: CALL SETERR (MESSG,NMESSG,NERR,JOPT) | On-line doc: CALL GAMSDOC SETERR (or @PRT PORT*DOC.SETERR) | Access: LIB NBS*PORT

SETI  Set a specified number of values in an integer array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): D1a1 | Usage: CALL SETI (N,V,B) | On-line doc: CALL GAMSDOC SETI (or @PRT PORT*DOC.SETI) | Access: LIB NBS*PORT

SETL  Set a specified number of values in a logical array equal to a constant. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SETD. | Class(es): D1a1 | Usage: CALL SETL (N,V,B) | On-line doc: CALL GAMSDOC SETL (or @PRT PORT*DOC.SETL) | Access: LIB NBS*PORT

SGBCO  Computes LU factorization of real band matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBCO. | Class(es): D2a2 | Usage: CALL SGBCO(ABD,lda,n,ml,mu,ipvt,rcnd,D2a2) | On-line doc: CALL GAMSDOC SGBCO (or @PRT CMLIB*DOC.SGBCO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS*CMLIB

SGBDI  Uses LU factorization of real band matrix to compute its determinant. (No provision for computing matrix inverse.) | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBDI. | Class(es): D2a2 | Usage: CALL SGBDI(ABD,lda,n,ml,mu,ipvt,DET) | On-line doc: CALL GAMSDOC SGBDI (or @PRT CMLIB*DOC.SGBDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS*CMLIB | See also: DGBDI, CMLIB*DOC.SGBDI

SGBFA  Computes LU factorization of real band matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBFA. | Class(es): D2a2 | Usage: CALL SGBFA(ABD,lda,n,ml,mu,ipvt,INFO) | On-line doc: CALL GAMSDOC SGBFA (or @PRT CMLIB*DOC.SGBFA/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS*CMLIB

SGBSL  Uses LU factorization of real band matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGBSL. | Class(es): D2a2 | Usage: CALL SGBSL(ABD,lda,n,ml,mu,ipvt,B,JOB) | On-line doc: CALL GAMSDOC SGBSL (or @PRT CMLIB*DOC.SGBSL/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS*CMLIB | See also: DGBSL, CMLIB*DOC.SGBSL

SGECO  Computes LU factorization of real general matrix and estimates its condition. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGECO. | Class(es): D2a1 | Usage: CALL SGECO(A,lda,n,ipvt,rcnd,D2a1) | On-line doc: CALL GAMSDOC SGECO (or @PRT CMLIB*DOC.SGECO/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SGECO, CMLIB*DOC.SGECO

SGEDI  Uses LU factorization of real general matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DGEDI. | Class(es): D2a1 D3a1 | Usage: CALL SGEDI(A,lda,n,ipvt,DET,WORK,JOBT) | On-line doc: CALL GAMSDOC SGEDI (or @PRT CMLIB*DOC.SGEDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS*CMLIB | See also: SGECO, SGEEV, CMLIB*DOC.SGEDI


SGEF5  Factors and solves a general NXN single precision system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMLIB library. Double precision version is DGEFS. | Class(es): D2a1 | Usage: CALL SGEFS(A,lda,n, vitask,ind,work,iwork) | On-line doc: CALL GAMSDOC SGEFS (or @PRT
CMLIB*DOC.SGEFS/LINDRIVES) | Tests: CMLIB+TEST-SOURCE.${F}/LINDRIVES | Access: Lib NBS+CMLIB


SGESL Uses LU factorization of real general matrix to solve systems. | Portable single precision Fortran program in LINPACKS sublibrary of CMLIB library. Double precision version is DGESL. | Class(es): D2a1 | Usage: CALL SGESL(A,LDA,N,IPVT,B,JOBS) | On-line doc: CALL GAMSDOC SGESL (or @PRT CMLIB*DOC.SGESL/LINPACKS) | Tests: CMLIB+TEST-SOURCE.${F}/LINPACKS | Access: Lib NBS+CMLIB | See also: SGECLA,SGEFA


SGTSL Factors a real tridiagonal matrix and simultaneously solves a system. | Portable single precision Fortran program in LINPACKS sublibrary of CMLIB library. Double precision version is DGTSCL. | Class(es): D2a2a | Usage: CALL SGTSL(N,C,D,E,B,INFO) | On-line doc: CALL GAMSDOC SGTSL (or @PRT CMLIB*DOC.SGTSCL/LINPACKS) | Tests: CMLIB+TEST-SOURCE.${F}/LINPACKS, CMLIB+TEST-SOURCE.${F}/LINPACKS | Access: Lib NBS+CMLIB

SICIEI Computes sine, cosine, exponential integral as well as hyperbolic sine, cosine, exponential integral. | Portable double precision Fortran program in STEGUN sublibrary of MATHWARE library. | Class(es): C5 C8 | Usage: CALL SICIEI(X,SIN,CI,CI,EI,EXEI,SHI,CHI,CHII,EII) | On-line doc: @PRTS MATHWARE*STEGUN.SICIEI/DOC | Access: See individual sublibrary documentation

SIN Sine of x. | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. | Class(es): C4a | Usage: Y = SIN(X) | On-line doc: CALL GAMSDOC SIN (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: Lib NBS+CMLIB

SINDG Computes the sine of an argument in degrees. | Portable single precision Fortran program in FNLIB sublibrary of CMLIB library. Double precision version is DSINDG. | Class(es): C4a | Usage: R = SINDG(X) | On-line doc: CALL GAMSDOC SINDG (or @PRT CMLIB*DOC.SUMMARY/FNLIB) | Access: Lib NBS+CMLIB

SINH Computes hyperbolic sine sinh(x). | Proprietary single precision Fortran program in PORT library. Double precision version is DSINH. | Class(es): C4e | Usage: X = SINH(X) | On-line doc: CALL GAMSDOC SINH (or @PRT PORT*DOC.SINH) | Access: Lib NBS+PORT


SINQI Initialize array WSAVE for SUBROUTINE SINQF and SINQP. | Portable single precision Fortran program in FFTPKG sublibrary of CMLIB library. | Class(es): -9- | Usage: CALL SINQI(N,WSAVE) | On-line doc: CALL GAMSDOC SINQI (or @PRT CMLIB*DOC.SINQI/FFTPKG) | Tests: CMLIB+TEST-SOURCE.${F}/FFTPKG | Access: Lib NBS+CMLIB | See also: SINQF,SINQP


SINTI Initialize array WSAVE for SUBROUTINE SINT. | Portable single precision Fortran program in FFTPKG sublibrary of CMLIB library. | Class(es): -9- | Usage: CALL SINTI(N,WSAVE) | On-line doc: CALL GAMSDOC SINTI (or @PRT CMLIB*DOC.SINTI/FFTPKG) | Tests: CMLIB+TEST-SOURCE.${F}/FFTPKG | Access: Lib NBS+CMLIB | See also: SINT

SKIPR Reads through (skips over) NI,HEAD lines from input unit = 5. | Portable single precision Fortran program in DATAPAC library. | Class(es): N1 | Usage: CALL SKIPR(NLHEAD) | On-line doc: CALL GAMSDOC SKIPR (or @PRT DATAPAC*DOC.SKIPR) | Access: Lib NBS+DATAPAC

SLIP1 Solves a fully implicit scheme for a one-dimensional system of parabolic differential equations with general boundary conditions on a specified equivalent spatial grid using a specified order spatial method. Order and stepsize in time and error estimate are computed. | Proprietary double precision Fortran program in SLDG/L library. | Class(es): I2a1 | Usage: CALL SLIP1(TOL,IERRO,IGLOB,IDOKE,IOUT,DTDR,ISTR,TAN,TSN,TEND,HAN,HMAX,HN,NOX,NOA,NF,N,F,UX,UT,UTFEST,UX,UX,U,UT,UX,DUT,UDU,UDUX,F,DGLT,DFGTU,DFGLT,DFGLX,DFXMAX,TAE,F,FUT,FX,FUX,Q,CMN,MOX,INTOK,OUT) | On-line doc: CALL GAMSDOC SLIP1 (or @PRT SLDG/L*DOC.SLIP1) | Tests:
SL1P2 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. User specifies equidistant grid and order in space, stepsize and order in time and error estimate computed. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P2 (TOL, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, NF, X, UX, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DXFMAX, DXFMIN, DXFUX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P2 (or @PRD SLDBG*DOC.SL1P2) | Tests: SLDBG*TEST-SOURCE.SL1P2,SLDBG*TEST-DATA.SL1P2 | Access: LIB NBS*SLDBG.

SL1P3 Solves a fully implicit difference system of parabolic equations with specified initial and boundary conditions. For a given relative accuracy an equidistant spatial grid and optimal order are computed as well as stepsize and order in time. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P3 (TOL, IGGLO, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, NF, X, XN, XEN, UH, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DXFMAX, TAM, E, FU, FUT, FUX, FUXX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P3 (or @PRD SLDBG*DOC.SL1P3) | Tests: SLDBG*TEST-SOURCE.SL1P3,SLDBG*TEST-DATA.SL1P3 | Access: LIB NBS*SLDBG.

SL1P4 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from the solution of an ode boundary value problem. Given a relative accuracy an equidistant spatial grid and optimal order are determined. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P4 (TOL, IGGLO, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, NF, X, XN, XEN, UH, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DXFMAX, TAM, E, FU, FUT, FUX, FUXX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P4 (or @PRD SLDBG*DOC.SL1P4) | Tests: SLDBG*TEST-SOURCE.SL1P4,SLDBG*TEST-DATA.SL1P4 | Access: LIB NBS*SLDBG.

SL1P5 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with specified initial and boundary conditions and non-equidistant spatial grid and spatial order. Stepsize and order in time are computed as well as an estimate of global error. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P5 (TOL, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, ABX, ABXX, FABXX, FabXX, X, XN, XZ, Z2, STX, UH, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFMAX, TAM, E, FU, FUT, FUX, FUXX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P5 (or @PRD SLDBG*DOC.SL1P5) | Tests: SLDBG*TEST-SOURCE.SL1P5,SLDBG*TEST-DATA.SL1P5 | Access: LIB NBS*SLDBG.

SL1P6 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. A non-equidistant spatial grid and spatial order are specified and the stepsize and order in time and error estimate are computed. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P6 (TOL, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, ABX, ABXX, FABXX, FabXX, X, XN, XZ, Z2, STX, UH, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFMAX, TAM, E, FU, FUT, FUX, FUXX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P6 (or @PRD SLDBG*DOC.SL1P6) | Tests: SLDBG*TEST-SOURCE.SL1P6,SLDBG*TEST-DATA.SL1P6 | Access: LIB NBS*SLDBG.

SL1P7 Solves a fully implicit difference scheme for a system of parabolic equations with specified initial and boundary conditions. For a specified relative accuracy a non-equidistant spatial grid and optimal order are computed as well as an estimate of global error. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P7 (TOL, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, ABX, ABXX, FabXX, FabXX, X, XN, XZ, Z2, STX, UH, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFMAX, TAM, E, FU, FUT, FUX, FUXX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P7 (or @PRD SLDBG*DOC.SL1P7) | Tests: SLDBG*TEST-SOURCE.SL1P7,SLDBG*TEST-DATA.SL1P7 | Access: LIB NBS*SLDBG.

SL1P8 Solves a fully implicit difference scheme for a one-dimensional system of parabolic equations with initial conditions determined from an ode boundary value problem. For a given relative accuracy a non-equidistant spatial grid and order and error estimate computed. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12a1 | Usage: CALL SL1P8 (TOL, IERROR, IGGLO, IDOKU, IOUT, DTDTR, ISTR, TANF, TEND, HANF, HMAC, HMIN, NV, NX, NOX, NA, ABX, ABXX, FabXX, FabXX, X, XN, XZ, Z2, STX, UH, U, UT, UTDF, UX, UXU, DU, DUX, DUXF, F, DFGLT, DFGLTU, DFGLTO, DFGLX, DFMAX, TAM, E, FU, FUT, FUX, FUXX, Q, MF, MINKX, MOX, INTOK, LOUT) | On-line doc: CALL GAMSIDOC SL1P8 (or @PRD SLDBG*DOC.SL1P8) | Tests: SLDBG*TEST-SOURCE.SL1P8,SLDBG*TEST-DATA.SL1P8 | Access: LIB NBS*SLDBG.

SL2E2 Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are automatically determined. | Proprietary double precision Fortran program in SLDBG library. | Class(es): 12b2 | Usage: CALL SL2E2 (MV, MV2, MVNX, MVNX, MOX, ABX, ABXX, ABYY, DF1, DF2, DU, DUX, FABX, FABY, FABXX, FABXY, FU, FUX, FUY, FUX, FUYY, ST, UMAM, UMIN, UH, MAXIT) | On-line doc: CALL GAMSIDOC SL2E2 (or @PRD SLDBG*DOC.SL2E2) | Tests: SLDBG*TEST-SOURCE.SL2E2,SLDBG*TEST-DATA.SL2E2 | Access: LIB NBS*SLDBG.


SL2E6 Solves by a fully implicit difference scheme a two-dimensional elliptic boundary value problem with general boundary conditions on a rectangle. For a prescribed equidistant grid and relative accuracy the routine automatically determines the order of the method. Error estimate provided. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL2E6 (MAXIT,MOX, MV, MV2, MNX, MNY, ABS, ABX, ABXX, ABYY, DF1, DF2, DU, DUH, FABX, FABY, FABXX, FABYY, FUX, FUY, FUXX, FUYY, ST, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL2E6 (or @PRT SLDGL*DOC.SL2E6) | Tests: SLDGL*TEST-SOURCE.SL2E6,SLDGL*TEST-DATA.SL2E6 | Access: LIB NBS+SLDGL

SL2E7 Solves by a fully implicit difference scheme for two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided. Line iteration used. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL2E7 (MAXIT, MOX, MV, MNX, MNY, AX, ABX, ABXX, ABYY, DU, DU2, DX, Q1, Q2, FABX, FABY, FABXX, FABYY, FU, FUX, FUY, FUXX, FUYY, ST, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL2E7 (or @PRT SLDGL*DOC.SL2E7) | Tests: SLDGL*TEST-SOURCE.SL2E7,SLDGL*TEST-DATA.SL2E7 | Access: LIB NBS+SLDGL

SL2E8 Solves by a fully implicit difference scheme for a two-dimensional elliptic boundary value problem on a rectangle with general boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are determined. Approximation solved by line iteration. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL2E8 (MAXIT, MOX, MV, MNX, MNY, ABX, ABXX, ABYY, DU, DU2, DX, Q1, Q2, FABX, FABY, FABXX, FABYY, FU, FUX, FUY, FUXX, FUYY, ST, UMAX, UMIN, DU, DEF, M1, M2) | On-line doc: CALL GAMSDOC SL2E8 (or @PRT SLDGL*DOC.SL2E8) | Tests: SLDGL*TEST-SOURCE.SL2E8,SLDGL*TEST-DATA.SL2E8 | Access: LIB NBS+SLDGL

SL2EB1 Solves by a fully implicit difference scheme for a system of two-dimensional elliptic equations on a general region with Dirichlet or Neumann boundary conditions specified. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is provided. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL2EB1 (MAXIT, MV, MN, MNX, MNY, MG, G, IT, UH, UMAX, UMIN, DFGLX, DFGLY, DU, DX, FU, FUX, FUY, FUXX, FUYY, RX, RP, A, PHAX, ALPHAY, NALX, NALY, IF, INABS, INREL, IFEL) | On-line doc: CALL GAMSDOC SL2EB1 (or @PRT SLDGL*DOC.SL2EB1) | Tests: SLDGL*TEST-SOURCE.SL2EB1,SLDGL*TEST-DATA.SL2EB1 | Access: LIB NBS+SLDGL

SL2P1 Solves by a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations on a rectangle with specified initial and boundary conditions. User provides non-equidistant spatial grid and spatial orders and stepsize and order in time and error estimate are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b1 | Usage: CALL SL2P1 (ISTART, MAXINT, MV, M2, MX, MN, MOX, ABX, ABY, ABXX, ABYY, UH, DU, DX, DFGLX, DFGLY, FABX, FABY, FABXX, FABYY, FU, FUT, FUX, FUY, FUXX, FUYY, ST, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL2P1 (or @PRT SLDGL*DOC.SL2P1) | Tests: SLDGL*TEST-SOURCE.SL2P1,SLDGL*TEST-DATA.SL2P1 | Access: LIB NBS+SLDGL


SL2P3 Solves by a fully implicit difference scheme for an implicit system of parabolic equations with given initial and boundary conditions. For a given error tolerance an optimal combination of spatial gridpoint distribution and order are computed as well as an estimate of the global error. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1b | Usage: CALL SL2P3 (ISTART, MAXINT, MV, M2, MX, MN, MOX, ABX, ABY, ABXX, ABYY, UH, DU, DX, DFGLX, DFGLY, FABX, FABY, FABXX, FABYY, FU, FUT, FUX, FUY, FUXX, FUYY, ST, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL2P3 (or @PRT SLDGL*DOC.SL2P3) | Tests: SLDGL*TEST-SOURCE.SL2P3,SLDGL*TEST-DATA.SL2P3 | Access: LIB NBS+SLDGL

SL2P4 Solves by a fully implicit difference scheme for an implicit system of two-dimensional parabolic equations with the initial conditions determined as the solution of an elliptic boundary value problem. For a given tolerance optimal spatial gridpoint distribution and order are computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2a1b | Usage: CALL SL2P4 (ISTART, MAXINT, MV, M2, MX, MN, MOX, ABX, ABY, ABXX, ABYY, HZ, DU, DX, DUH, DFGLX, DFGLY, FABX, FABY, FABXX, FABYY, FU, FUT, FUX, FUY, FUXX, FUYY, ST, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL2P4 (or @PRT SLDGL*DOC.SL2P4) | Tests: SLDGL*TEST-SOURCE.SL2P4,SLDGL*TEST-DATA.SL2P4 | Access: LIB NBS+SLDGL

SL3E1 Solves by a fully implicit difference scheme for a system of three-dimensional elliptic equations on a parallelepiped with general boundary conditions. A non-equidistant grid and the order of the method are prescribed. An estimate of the discretization error is computed. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): I2b2 | Usage: CALL SL3E1 (MAXIT, MOX, MV, MNX, MNY, MNZ, AX, ABX, ABXX, ABYY, ABYY, ABZZ, DU, DULAT, DX, DUX, FABX, FABY, FABZ, FABXX, FABYY, FABZZ, FU, FUX, FUY, FUXX, FUYY, FUZZ, ST, UH, UMAX, UMIN) | On-line doc: CALL GAMSDOC SL3E1 (or @PRT SLDGL*DOC.SL3E1) | Tests: SLDGL*TEST-SOURCE.SL3E1,SLDGL*TEST-DATA.SL3E1 | Access: LIB NBS+SLDGL
SL3P1 Solves a fully implicit difference scheme for a three-dimensional system of parabolic equations with specified initial and boundary conditions on a parallelepiped. The user provides a non-equidistant spatial grid and spatial orders and stepsize and order in time and an error estimate are computed. | Proprietary double precision Fortran program in SLDGL library. | Class(es): IAB | Usage: CALL SL3P1 (ISTART,MAXINT,MNX, MNY,MNZ,MX, ABX,ABY,ABZ,AIBXX, AYYY,ABZZ,DU,HU, DUXX,DFGL,DFGLALT,FABX,FABY,FABZ,FABXX, FABFY,FABZU,FUT,FUX, FUY,FUXX,FUYY,FUZZ,STX,STY,STZ,UMIN) | On-line doc: CALL GAMSDOC SL3P1 (or @PRT SLDGL+DOC.SL3P1) | Tests: SLDGL+TEST-SOURCE.SL3P1,SLDGL+TEST-DATA.SL3P1 | Access: LIB NBS+SLDGL

SLGA1 Solves by difference methods an initial value problem for a system of ordinary differential equations. The stepsize and order of the method are chosen automatically to maintain local discretization error bounds. | Proprietary double precision Fortran program in SLDGL library. | Class(es): IIAB | Usage: CALL SLGA1 (N,NMAX,TOL,REL, IABS,IGLOB,JOUT, IDOKU,DXDR,HANF, HMAX,XANF,XEND,YO,Y, YSTR,YSFEST,YJACOB,DY,DFEKT,GLFEHL,ALOFEH, UNFEHL,OBFEHL,EN,TAU,GW,LOUT) | On-line doc: CALL GAMSDOC SLGA1 (or @PRT SLDGL+DOC.SLGA1) | Tests: SLDGL+TEST-SOURCE.SLGA1,SLDGL+TEST-DATA.SLGA1 | Access: LIB NBS+SLDGL

SLGA2 Solves by difference methods a mixed implicit/algebraic initial value problem for a system of ordinary differential equations. Stepsize and order are chosen automatically to maintain estimated local discretization error within prescribed bounds. | Proprietary double precision Fortran program in SLDGL library. | Class(es): IIAB | Usage: CALL SLGA2 (N,NMAX,TOL,REL, IABS,IGLOB,JOUT, IDOKU,DXDR,HANF, HMAX,XANF,XEND,YO,Y, YSTR,YSFEST,YJACOB,DY,DFEKT,GLFEHL,ALOFEH, UNFEHL,OBFEHL,EN,TAU,GW,LOUT) | On-line doc: CALL GAMSDOC SLGA2 (or @PRT SLDGL+DOC.SLGA2) | Tests: SLDGL+TEST-SOURCE.SLGA2,SLDGL+TEST-DATA.SLGA2 | Access: LIB NBS+SLDGL

SLGA3 Solves by difference methods an initial value problem for an explicit system of ordinary differential equations. Chooses stepsize and order of method to maintain estimate of local discretization error within prescribed bounds. Also provides values of first derivative at output points. | Proprietary double precision Fortran program in SLDGL library. | Class(es): IIAB | Usage: CALL SLGA3 (N,NMAX,TOL,REL, IABS,IGLOB, JOUT, IDOKU,DXDR,ISTR, HANF, HMAX, XANF, XEND, YO,Y, YSTR,YSFEST,YJACOB,DY, DFEKT,GLFEHL,ALOFEH, UNFEHL,OBFEHL,EN,TAU,GW,LOUT) | On-line doc: CALL GAMSDOC SLGA3 (or @PRT SLDGL+DOC.SLGA3) | Tests: SLDGL+TEST-SOURCE.SLGA3,SLDGL+TEST-DATA.SLGA3 | Access: LIB NBS+SLDGL

SLG4 Solves by difference methods a mixed algebraic/implicit initial value problem for a first order system of ordinary differential equations. Chooses stepsize and order to maintain estimate of local discretization error within prescribed bounds. Also provides derivative at output points. | Proprietary double precision Fortran program in SLDGL library. | Class(es): IIAB | Usage: CALL SLG4 (N,NMAX,TOL,REL, IABS,IGLOB, JOUT, IDOKU,DXDR,ISTR, HANF, HMAX, XANF, XEND, YO,Y, YSTR,YSFEST,FY, FYX,YJACOB,DY, DFEKT,GLFEHL, ALOFEH, UNFEHL,OBFEHL,EN,TAU,GW,LOUT) | On-line doc: CALL GAMSDOC SLG4 (or @PRT SLDGL+DOC.SLG4) | Tests: SLDGL+TEST-SOURCE.SLGA4,SLDGL+TEST-DATA.SLGA4 | Access: LIB NBS+SLDGL

SLGR0 Solves by difference methods a non-equidistant grid a second-order implicit system of ordinary differential equations with implicit boundary conditions prescribed. An estimate of the discretization error is provided. | Proprietary double precision Fortran program in SLDGL library. | Class(es): HB2 | Usage: CALL SLGR0 (IDOKU,NX,NOX,NV, MNX,MX,MV,MX,NA,KONV, ABX,ABXX,DFGL, DY,DYX,DYXX,F,FABX,FABXX,FY,FYY,FZQ, RUNDF,ST,STX,STY,STZ,YY,YYX,YYZ,YYL, LOUT) | On-line doc: CALL GAMSDOC SLGR0 (or @PRT SLDGL+DOC.SLGR0) | Tests: SLDGL+TEST-SOURCE.SLGR0,SLDGL+TEST-DATA.SLGR0 | Access: LIB NBS+SLDGL

SLGR1 Solves by difference methods an equidistant grid an implicit system of second-order ordinary differential equations with prescribed implicit boundary conditions. An estimate of the discretization error is provided. | Proprietary double precision Fortran program in SLDGL library. | Class(es): HB2 | Usage: CALL SLGR1 (IDOKU,NX,NOX,NV, MNX,MX,MV,MX,NA,KONV,INX,INF,AFFX, AFFXX,DFGL,DFGLALT, DY,DYX,DYXX,F,FABX,FABXX,FY,FYY,FZQ, RUNDF,ST,STX,STY,STZ,YY,YYX,YYZ,YYL, LOUT) | On-line doc: CALL GAMSDOC SLGR1 (or @PRT SLDGL+DOC.SLGR1) | Tests: SLDGL+TEST-SOURCE.SLGR1,SLDGL+TEST-DATA.SLGR1 | Access: LIB NBS+SLDGL

SLGR2 Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a given relative accuracy gridpoints and order of method are chosen automatically to minimize number of gridpoints. | Proprietary double precision Fortran program in SLDGL library. | Class(es): HB2 | Usage: CALL SLGR2 (IAN,IDOKU,NA,NOX, NNE,NX,NN, MX,MN,MX,MY,ABXX,ABYX, DFGLALT,DFGLALTALT, DY,DYX,DYXX,F,FABX,FABXX,FY,FYY,FZQ, Q,DUXX,DUXXALT,DFULALT,DFULALTALT, RUNDF,ST,STL,XAN, XNEU,XS,Y, YNEU,YX,YYY,YXX,ZW,LOUT) | On-line doc: CALL GAMSDOC SLGR2 (or @PRT SLDGL+DOC.SLGR2) | Tests: SLDGL+TEST-SOURCE.SLGR2,SLDGL+TEST-DATA.SLGR2 | Access: LIB NBS+SLDGL

SLGR3 Solves by difference methods on an equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary values. For a prescribed relative accuracy the order of the method is chosen to minimize the number of gridpoints. | Proprietary double precision Fortran program in SLDGL library. | Class(es): HB2 | Usage: CALL SLGR3 (IAN,IDOKU,NA,NOX, NNE,NX,NN, MX,MN,MX,MY,MX,NA, KONV,INX,INF,AFFX, AFFXX,DFGLALT, DFGLALTALT, DY,DYX,DYXX,F,FABX, FABXX,FY,FYY,FZQ,Q,DUXX,DUXXALT, RUNDF,ST,STL,XAN, XNEU,XS,Y, YNEU,YX,YYY,ZF1,ZP2,C) | On-line doc: CALL GAMSDOC SLGR3 (or @PRT SLDGL+DOC.SLGR3) | Tests: SLDGL+TEST-SOURCE.SLGR3,SLDGL+TEST-DATA.SLGR3 | Access: LIB NBS+SLDGL

SLGR4 Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a prescribed relative accuracy the gridpoints and order of the method are chosen to minimize the number of gridpoints. | Proprietary double precision Fortran program in SLDGL library. | Class(es): HB2 | Usage: CALL SLGR4 (IAN,IDOKU, IFILE,IFL,LOPT, LOUT,MOX,MNX,MY,MX,NA, NOX,NNX,ABX,ABXX,DFGLALT, DFGLALTALT, DY,DYX,DYXX,F,FABX, FABXX,FY,FYY,FZQ,Q,DUXX,DUXXALT, RUNDF,ST,STL,XAN, XNEU,XS,Y, YNEU,YX,YYY,ZF1,ZP2,C) | On-line doc: CALL GAMSDOC SLGR4 (or @PRT SLDGL+DOC.SLGR4) | Tests: SLDGL+TEST-SOURCE.SLGR4,SLDGL+TEST-DATA.SLGR4 | Access: LIB NBS+SLDGL
SLGR5 Solves by difference methods on a non-equidistant grid an implicit system of second-order ordinary differential equations with implicit boundary conditions. For a given relative accuracy the gridpoints and order of method are chosen to minimize number of gridpts. Suited for boundary layer problems. | Proprietary double precision Fortran subprogram in SLDGL library. | Class(es): IIb2 | Usage: CALL SLGR5(IAN,IDO,K,U,F,Y,FY,FYY,FYXX,H,Q,RUNDF,ST,TOL,X,XAN,XEN,XNEU,XS,Y,YNEU,YX,YXX,ZF1,ZF2,C) | On-line doc: CALL GAMSDOC SLGR5 (or @PRT SLDGL+DOC.SLGR5) | Tests: SLDGL+TEST-SOURCE.SLGR5,SLDGL+TEST-DATA.SLGR5 | Access: LIB NBS+SLDGL

SLVBLK Solves \(Ax=b\) where \(A\) is an almost block diagonal matrix. These arise in finite element or piecewise polynomial approximation. | Portable single precision Fortran subprogram in SLVBLK sublibrary of CMILIB library. | Class(es): D2a2 | Usage: CALL SLVBLK(BLOKS,INTEGS,NBLK,(B),IPIV,T,X,IFLAG) | On-line doc: CALL GAMSDOC SLVBLK (or @PRT CMILIB+DOC.SLVBLK) | Tests: CMILIB+TEST-SOURCE.$Q/SLVBLK | Access: LIB NBS+CMILIB

SMONOD Test if a double precision vector is strictly monotone increasing or decreasing. | Proprietary double precision Fortran subprogram in PORT library. Single precision version is SMONOR. | Class(es): R2 | Usage: L = SMONOD(X, N, INC) | On-line doc: CALL GAMSDOC SMONOD (or @PRT PORT+DOC.SMONOD) | Access: LIB NBS+PORT

SMONO1 Test if an integer vector is strictly monotone increasing or decreasing. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): R2 | Usage: L = Smono1(X, N, INC) | On-line doc: CALL GAMSDOC Smono1 (or @PRT PORT+DOC.Smono1) | Access: LIB NBS+PORT

SMONOR Test if a real vector is strictly monotone increasing or decreasing. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is SMONOD. | Class(es): R2 | Usage: L = SMONOR(X, N, INC) | On-line doc: CALL GAMSDOC SMONOR (or @PRT PORT+DOC.SMONOR) | Access: LIB NBS+PORT

SMSNO Minimize a general unconstrained objective function using finite difference gradients and secant Hessian approximations. | Portable single precision Fortran program in NL2SN sublibrary of CMILIB library. Double precision version is DMSNOS. | Class(es): G1b1a | Usage: CALL SMSNO(N,D,X,CALCF,CLVF,LIV,LV,V,UPARM,UPARN,UPFARM) | On-line doc: CALL GAMSDOC SMSN0 (or @PRT CMILIB+DOC.SMSNO/NL2SN) | Tests: CMILIB+TEST-SOURCE.$F2/NL2SN, CMILIB+TEST-SOURCE.$Q2/NL2SN | Access: LIB NBS+CMILIB

SNBCO Factors a real band matrix by Gaussian elimination and estimates condition of the matrix. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMILIB library. Double precision version is DNBCO. | Class(es): D2a2 | Usage: CALL SNBco(abe,lda,n,ml,mu,ipvt,rcond,z) | On-line doc: CALL GAMSDOC DNBCO (or @PRT CMILIB+DOC.SNBCO/LINDRIVES) | Tests: CMILIB+TEST-SOURCE.$F1/LINDRIVES | Access: LIB NBS+CMILIB

SNBDI Computes the determinant of a single precision band matrix using factors previously computed. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMILIB library. Double precision version is DNBDI. | Class(es): D3a2 | Usage: CALL SNBDI(abe,lda,n,ml,mu,ipvt,det) | On-line doc: CALL GAMSDOC SNBDI (or @PRT CMILIB+DOC.SNBDI/LINDRIVES) | Tests: CMILIB+TEST-SOURCE.$F1/LINDRIVES | Access: LIB NBS+CMILIB | See also: SNBCO, SNBDI

SNBFA Factors a single precision band matrix by elimination. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMILIB library. Double precision version is DNBF. | Class(es): D2a2 | Usage: CALL SNBFA(abe,lda,n,ml,mu,ipvt,info) | On-line doc: CALL GAMSDOC SNBFA (or @PRT CMILIB+DOC.SNBDI/LINDRIVES) | Tests: CMILIB+TEST-SOURCE.$F1/LINDRIVES | Access: LIB NBS+CMILIB

SNBFS Factors and solves a general nonsymmetric single precision banded system of linear equations. | Portable single precision Fortran subprogram in LINDRIVES sublibrary of CMILIB library. Double precision version is DNBSF. | Class(es): D2a2 | Usage: CALL SNBFS(abe,lda,n,ml,mu,v,ita,ind,work,iwork) | On-line doc: CALL GAMSDOC SNBFS (or @PRT CMILIB+DOC.SNBSF/LINDRIVES) | Tests: CMILIB+TEST-SOURCE.$F1/LINDRIVES | Access: LIB NBS+CMILIB

SNBIR Factors and solves a general nonsymmetric single precision banded system of equations and estimates solution accuracy (needs Nx(2ML+MU) extra storage). | Portable single precision Fortran program in LINDRIVES sublibrary of CMILIB library. | Class(es): D2a2 | Usage: CALL SNBIR(abe,lda,n,ml,mu,v,ita,ind,work,iwork) | On-line doc: CALL GAMSDOC SNBIR (or @PRT CMILIB+DOC.SNBDI/LINDRIVES) | Tests: CMILIB+TEST-SOURCE.$F1/LINDRIVES | Access: LIB NBS+CMILIB

SNBSL Solves a general nonsymmetric single precision banded system of linear equations using factors computed previously. | Portable single precision Fortran program in LINDRIVES sublibrary of CMILIB library. Double precision version is DNBSL. | Class(es): D2a2 | Usage: CALL SNBSL(abe,lda,n,ml,mu,ipvt,b,job) | On-line doc: CALL GAMSDOC SNBSL (or @PRT CMILIB+DOC.SNBSl/LINDRIVES) | Tests: CMILIB+TEST-SOURCE.$F1/LINDRIVES | Access: LIB NBS+CMILIB | See also: SNBCO, SNBFA

SNLSL Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. Flexible usage, including various options for providing Jacobian. Covariance matrix is available via the subroutine SLS. | Portable single precision Fortran program in SNLSL1 sublibrary of CMILIB library. | Class(es): K1b1a1 K1b1a2 | Usage: CALL SNLSL(FCN,IOPT,M,N,X,FVEC,FJAC,DFSOL,XTOL,XTOL,MAXFVE,EPSCFN,DIAG,MODE,FACTOR,NPRINT,INFO,FY,FYX,FYY,FYXX,H,Q,RUNDF,ST,TOL,X,XAN,XEN,XNEU,XS,Y,YNEU,YX,YXX,ZF1,ZF2,C) | On-line doc: CALL GAMSDOC SNLSL (or @PRT CMILIB+DOC.SNLsL1/SNLSL1) | Tests: CMILIB+TEST-SOURCE.$Q/SNLSL1 | Access: LIB NBS+CMILIB | See also: SCOV, CHKDER checks user's Jacobian routine if desired
SNLS1E  Minimizes the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. An easy to use driver for SNLS1. The covariance matrix is available by calling the subroutine COV.

Portable single precision Fortran subroutine in SNLS1E sublibrary of CMLIB library. | Class(es): K1bl1a1 K1bl1a2 | Usage: CALL SNLS1E(FCN,IOPT,M,N,X,FVEC,TOL,NPRINT,INFO,JW,WA,LIW) | On-line doc: CALL GAMSDOC SNLS1E (or @PRT CMLIB*DOC.SNLS1E/SNLS1E) | Tests: CMLIB*TEST-SOURCE,.$Q/SNLS1E | Access: LIB NBS+CMLIB | See also: COV, CHKDER checks user’s Jacobian routine if desired


SNRM2  Finds the length (Euclidean norm) of a vector, without underflow or overflow. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DNRM2. | Class(es): D1a3b | Usage: X = SNRM2(N, X, INX) | On-line doc: CALL GAMSDOC SNRM2 (or @PRT PORT*DOC.SNRM2) | Access: LIB NBS+PORT


SODS  Solves an overdetermined system of linear equations. For full rank matrices the unique least squares solution is provided. The least squares solution of minimal length can be obtained in the rank deficient case. | Portable single precision Fortran subroutine in SUBSSODS sublibrary of CMLIB library. | Class(es): D9 | Usage: CALL SODS(A,X,B,NEQ,NUK,NRDA,IFLAG,WORK,JWORK) | On-line doc: CALL GAMSDOC SODS (or @PRT CMLIB*DOC.SODS/SUBSSODS) | Tests: CMLIB*TEST-SOURCE.SODS/SUBSSODS | Access: LIB NBS+CMLIB


SORT  Sorts (in ascending order) the N elements of the vector X and puts the resulting N sorted values into the vector Y. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): N6a2b1 | Usage: CALL SORT(X,Y,N) | On-line doc: CALL GAMSDOC SORT (or @PRT DATAPAC*DATAPAC.DOC.SORT) | Access: LIB NBS+DATAPAC

SORT  Sorts a vector in ascending order and optionally carries along other vectors. | Command in MINITAB Proprietary interactive system. | Class(es): N6a2b1 | Usage: SORT: the values in C [carry along corresponding rows of C1,..., Cn] put into C [corresponding rows into C1,...,Cn] | On-line doc: HELP SORT (in MINITAB) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

SORTC  Sorts (in ascending order) the N elements of the vector X and rearranges the elements of the vector Y. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): N6a2b1 | Usage: CALL SORTC(X,Y,N,XS,YC) | On-line doc: CALL GAMSDOC SORTC (or @PRT DATAPAC*DATAPAC.DOC.SORTC) | Access: LIB NBS+DATAPAC

SORTP  Sorts (in ascending order) the N elements of the vector X, puts the resulting N sorted values into the vector Y, and puts the position (in the original vector X) of each of the sorted values into the single precision vector XPOS. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): N6a2b1 | Usage: CALL SORTP(X,Y,N,XPOS) | On-line doc: CALL GAMSDOC SORTP (or @PRT DATAPAC*DATAPAC.DOC.SORTP) | Access: LIB NBS+DATAPAC


SPBDI  Uses factorization of real positive definite band matrix to compute its determinant. (No provision for matrix inverse.). | Portable single precision Fortran subroutine in LINPACKS sublibrary of CMLIB library. Double precision version is DPBDI. | Class(es): D2b2 | Usage: CALL SPBDI(ABD,LDA,N,M,DET) | On-line doc: CALL GAMSDOC SPBDI (or @PRT CMLIB*DOC.SPBDI/LINPACKS) |
GAMS: Module Dictionary

SPBFA

SPBSL
Uses factorization of real positive definite band matrix to solve systems. Portable single precision Fortran subprogram in LINPACS sublibrary of CMLIB library. Double precision version is DPBSL. Class(es): Dsb2 | Usage: CALL SPBSL(ABD,LDA,N,M,B) | On-line doc: CALL GAMSDOC SPBSL (or @PRT CMLIB+DOC.SPBFA/LINPACS) | Tests: CMLIB+TEST-SOURCE.$Q/LINPACS, CMLIB+TEST-SOURCE.$F/LINPACS | Access: LIB NBS+CMLIB

SPCCOR
Computes the Spearman rank correlation coefficient between the two sets of data in the input vectors X and Y. Portable single precision Fortran subprogram in LINPACS sublibrary of CMLIB library. Double precision version is DSCOR. Class(es): C5 | Usage: Y = SPCCOR (X) | On-line doc: CALL GAMSDOC SPCCOR (or @PRT CMLIB+DOC.SUMMARY/FNLIB) | Access: LIB NBS+CMLIB

SPLN1
Evaluates a function and derivatives described previously by an expansion in terms of B-splines. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSSPLN1. Class(es): E3 K6 | Usage: CALL SPLN1 (K,T,N,A,X,NX,ID,NID,FX) | On-line doc: CALL GAMSDOC SPLN1 (or @PRT PORT+DOC.SPLN1) | Access: LIB NBS+PORT

SPLN2
Evaluates a function described by a previously determined expansion in B-splines. More flexible than SPLN1. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSSPLN2. Class(es): E3 K6 | Usage: CALL SPLN2 (K,T,N,A,X,NX,ID,NID,FX,IDIM,ADIFF,IL0,ILEFT) | On-line doc: CALL GAMSDOC SPLN2 (or @PRT PORT+DOC.SPLN2) | Access: LIB NBS+PORT

SPLND
Evaluates at a given set of points a function described by a previously determined expansion in terms of B-splines. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSSPLND. Class(es): E3 K6 | Usage: CALL SPLND (K,T,N,A,X,NX,MD,FX) | On-line doc: CALL GAMSDOC SPLND (or @PRT PORT+DOC.SPLND) | Access: LIB NBS+PORT

SPLNE
Evaluates at a set of points, a function described by a previously determined expansion in terms of B-splines. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSSPLNE. Class(es): E3 K6 | Usage: CALL SPLNE (K,T,N,A,X,NX,FX) | On-line doc: CALL GAMSDOC SPLNE (or @PRT PORT+DOC.SPLNE) | Access: LIB NBS+PORT

SPLNI
Integrates a function described previously by an expansion in terms of B-splines. Several integrations can be performed in one call. Proprietary single precision Fortran subprogram in PORT library. Double precision version is DSSPLNI. Class(es): H2a2a1 E3 K6 | Usage: CALL SPLNI (K,T,N,A,X,NX,FIX) | On-line doc: CALL GAMSDOC SPLNI (or @PRT PORT+DOC.SPLNI) | Access: LIB NBS+PORT

SPLP
Solves linear optimization problems, that is, it minimizes the linear function (TRANSPOSE OF COSTS)X subject to A+X=W, where the entries of the vectors X and W may have simple upper or lower bounds. Uses a sparse storage mode for the matrix A and out-of-core scratch storage. Portable single precision Fortran subprogram in SPLP sublibrary of CMLIB library. Class(es): G2a2 | Usage: CALL SPLP (USRMAT,MRELAS,NVARS,COSTS,PROLOPT,DATT,DTT,BL,BL,IND,INFO,PRIMAL,DUALS,IBASIS,WORK,LI,WORK,LW) | On-line doc: CALL GAMSDOC SPLP (or @PRT CMLIB+DOC.SPLP/SPLP) | Tests: CMLIB+TEST-SOURCE.$Q/SPLP, CMLIB+TEST-SOURCE.$F/SPLP | Access: LIB NBS+CMLIB

SLTT
Displays a 50x100 character line printer scatter plot with user control of the plotting symbol used for each point. Portable single precision Fortran subprogram in STATLIB library. Class(es): L3c2 Q1 | Usage: CALL SPLIT (Y, X, N, ISYM) | On-line doc: CALL GAMSDOC SPLIT (or @PRT STATLIB+DOC.SPLIT) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

SPLTH
Displays a 50x50 character line printer scatter plot with user control of the plotting symbol used for each point. Portable single precision Fortran subprogram in STATLIB library. Class(es): L3c2 Q1 | Usage: CALL SPLTH (Y, X, N, ISYM, NOT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC SPLTH (or @PRT STATLIB+DOC.SPLTH) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

SPLTHL
Displays a 50x50 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point. Portable single precision Fortran subprogram in STATLIB library. Class(es): L3c2 Q1 | Usage: CALL SPLTHL (Y, X, N, ISYM, NOT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC SPLTHL (or @PRT STATLIB+DOC.SPLTHL) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

SPLTL
Displays a 50x100 character line printer scatter plot with user control of the plot limits and of the plotting symbol used for each point. Portable single precision Fortran subprogram in STATLIB library. Class(es): L3c2 Q1 | Usage: CALL SPLTL (Y, X, N, ISYM, NOT, YLB, YUB, XLB, XUB) | On-line doc: CALL GAMSDOC SPLTL (or @PRT STATLIB+DOC.SPLTL) | Tests: STATLIB+TEST.DEM01 | Access: LIB NBS+STATLIB

SPODI Uses factorization of real positive definite matrix to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPODI. | Class(es): D2blb D3blb | Usage: CALL SPODI(A,LDA,N,DET,JOB) | On-line doc: CALL GAMSDOC SPODI (or @PRT CMLIB+DOC.SPODI/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB | See also: SPOCO SPOFA


SPOS L Uses factorization of real positive definite matrix to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPOS L. | Class(es): D2blb | Usage: CALL SPOS L(A,LDA,N,B) | On-line doc: CALL GAMSDOC SPOS L (or @PRT CMLIB+DOC.SPOS L/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB | See also: SPOFA


SPPDI Uses factorization of real positive definite matrix stored in packed form to compute its determinant and/or inverse. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPPDI. | Class(es): D2blb D3blb | Usage: CALL SPPDI(A,P,N,DET,JOB) | On-line doc: CALL GAMSDOC SPPDI (or @PRT CMLIB+DOC.SPPDI/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB | See also: SPPCO SPPFA


SPPSL Uses factorization of real positive definite matrix stored in packed form to solve systems. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPFS L. | Class(es): D2blb | Usage: CALL SPPSL(A,P,N,B) | On-line doc: CALL GAMSDOC SPPSL (or @PRT CMLIB+DOC.SPPSL/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB | See also: SPPCO SPPFA

SPTSL Decomposes real symmetric positive definite tridiagonal matrix and simultaneously solves a system. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DPTS L. | Class(es): D2blb | Usage: CALL SPTSL(N,D,E,B) | On-line doc: CALL GAMSDOC SPTSL (or @PRT CMLIB+DOC.SPTSL/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB | See also: SPPCO SPPFA


SQRDC Computes QR decomposition of real general matrix. | Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DQRDC. | Class(es): D8 | Usage: CALL SQRDC(X,LDX,N,P,QRAX,JPV T,WORK,JOB) |
 January 1984

GAMS: Module Dictionary

 On-line doc: CALL GAMSDOC SQRDC (or @PRT CMLIB*DOC.SQRDC/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB


SQRLS Applies the output of SQRDC to compute coordinate transformations, projections, and least squares solutions (general real matrix). | Portable single precision Fortran subroutine in LINPACKS sublibrary of CMLIB library. Double precision version is DQRLS. | Class(es): D9 D2a1 | Usage: CALL SQRLS(X,LDX,N,K,QRANG,Y,QY,QT,B,RSD,XB,INFO) | On-line doc: CALL GAMSDOC SQRLS (or @PRT CMLIB*DOC.SQRLS/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB | See also: SQRDC

SRTQ Square root. | Portable single precision Fortran subroutine in FNLIB sublibrary of CMLIB library. | Class(es): C2 | Usage: Y = SQRT(X) | On-line doc: CALL GAMSDOC SQRT (or @PRT CMLIB*DOC.FORT/LINPACKS) | Access: LIB NBS+CMLIB

SROT Apply Givens plane rotation to a single precision vector. | Portable single precision Fortran subroutine in BLAS sublibrary of CMLIB library. Double precision version is DROT. | Class(es): D1a8 | Usage: CALL SROTN(N,SX,INCX,SY,INCY,SC,SS) | On-line doc: CALL GAMSDOC SROT (or @PRT CMLIB*DOC.SROTN/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB


SROTM Apply modified Givens plane rotation to single precision vector. | Portable single precision Fortran subroutine in BLAS sublibrary of CMLIB library. Double precision version is DROTM. | Class(es): D1a8 | Usage: CALL SROTM(N,SX,INCX,SY,INCY,SPARAM) | On-line doc: CALL GAMSDOC SROTM (or @PRT CMLIB*DOC.SROTM/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

SROTMG Construct modified Givens plane rotation of single precision matrix. | Portable single precision Fortran subroutine in BLAS sublibrary of CMLIB library. Double precision version is DROTMG. | Class(es): D1b10 | Usage: CALL SROTMG(S1,S2,SB2,PARAM) | On-line doc: CALL GAMSDOC SROTMG (or @PRT CMLIB*DOC.SROTMG/LINPACKS) | Tests: CMLIB+TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB

SRTAD Actively sorts double precision data into ascending order. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is SRTAR. | Class(es): N9a2b2 | Usage: CALL SRTAD(A,INT,N) | On-line doc: CALL GAMSDOC SRTAD (or @PRT PORT*DOC.SRTAD) | Access: LIB NBS+PORT

SRTAH Actively sorts Hollerith data into ascending order. | Proprietary single precision Fortran subroutine in PORT library. | Class(es): N8a2c | Usage: CALL SRTAH(A,L,INC,N) | On-line doc: CALL GAMSDOC SRTAH (or @PRT PORT*DOC.SRTAH) | Access: LIB NBS+PORT


SRTAR Actively sorts real data into ascending order. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is SRTAR. | Class(es): N8a2b1 | Usage: CALL SRTAR(A,INT,N) | On-line doc: CALL GAMSDOC SRTAR (or @PRT PORT*DOC.SRTAR) | Access: LIB NBS+PORT


SRTDH Actively sorts Hollerith data into descending order. | Proprietary single precision Fortran subroutine in PORT library. | Class(es): N8a2c | Usage: CALL SRTDH(A,L,INC,N) | On-line doc: CALL GAMSDOC SRTDH (or @PRT PORT*DOC.SRTDH) | Access: LIB NBS+PORT


SRTDR Actively sorts real data into descending order. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is SRTDR. | Class(es): N8a2b1 | Usage: CALL SRTDR(A,INT,N) | On-line doc: CALL GAMSDOC SRTDR (or @PRT PORT*DOC.SRTDR) | Access: LIB NBS+PORT

SRTPAD Passively sorts double precision data into ascending order. | Proprietary double precision Fortran subroutine in PORT library. Single precision version is SRTPAR. | Class(es): N8a1b2 | Usage: CALL SRTPAD(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPAD (or @PRT PORT*DOC.SRTPAD) | Access: LIB NBS+PORT | See also: SRTDR
SRTPAH  Passively sorts Hollerith data into ascending order.  Proprietary single precision Fortran subprogram in PORT library.  Class(es): N6alc | Usage: CALL SRTPAH(A,L,INC,N) | On-line doc: CALL GAMSDOC SRTPAH (or @PRT PORT*DOC.SRTPAH) | Access: LIB NBS*PORT | See also: SRTRH


SRTPAR  Passively sorts real data into ascending order.  Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTPAD.  Class(es): N6ai | Usage: CALL SRTPAR(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPAR (or @PRT PORT*DOC.SRTPAR) | Access: LIB NBS*PORT | See also: SRTRR

SRTPAD  Passively sorts double precision data into ascending order.  Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTPAD.  Class(es): N6ai | Usage: CALL SRTPBD(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPAD (or @PRT PORT*DOC.SRTPAD) | Access: LIB NBS*PORT | See also: SRTRD

SRTPDH  Passively sorts Hollerith data into descending order.  Proprietary single precision Fortran subprogram in PORT library.  Class(es): N6alc | Usage: CALL SRTPDH(A,L,INC,N) | On-line doc: CALL GAMSDOC SRTPDH (or @PRT PORT*DOC.SRTPDH) | Access: LIB NBS*PORT | See also: SRTRI

SRTPDI  Passively sorts integer data into descending order.  Proprietary single precision Fortran subprogram in PORT library.  Class(es): N6ai | Usage: CALL SRTPDI(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPDI (or @PRT PORT*DOC.SRTPDI) | Access: LIB NBS*PORT | See also: SRTRI

SRTPDR  Passively sorts real data into descending order.  Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTPDR.  Class(es): N6ai | Usage: CALL SRTPDR(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTPDR (or @PRT PORT*DOC.SRTPDR) | Access: LIB NBS*PORT | See also: SRTRR

SRTRD  Rearranges double precision data according to permutation stored in IP.  Proprietary double precision Fortran subprogram in PORT library. Single precision version is SRTRR.  Class(es): N8 | Usage: CALL SRTRD(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRD (or @PRT PORT*DOC.SRTRD) | Access: LIB NBS*PORT

SRTRH  Rearranges Hollerith data according to permutation stored in IP.  Proprietary single precision Fortran subprogram in PORT library.  Class(es): N8 | Usage: CALL SRTRH(A,L,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRH (or @PRT PORT*DOC.SRTRH) | Access: LIB NBS*PORT

SRTRI  Rearranges integer data according to permutation stored in IP.  Proprietary single precision Fortran subprogram in PORT library.  Class(es): N8 | Usage: CALL SRTRI(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRI (or @PRT PORT*DOC.SRTRI) | Access: LIB NBS*PORT

SRTRR  Rearranges real data according to permutation stored in IP.  Proprietary single precision Fortran subprogram in PORT library. Double precision version is SRTRR.  Class(es): N8 | Usage: CALL SRTRR(A,INTA,IP,INTP,N) | On-line doc: CALL GAMSDOC SRTRR (or @PRT PORT*DOC.SRTRR) | Access: LIB NBS*PORT

SSCAL  Computes a constant times a vector, both single precision.  Portable single precision Fortran subprogram in BLAS sublibrary of CMLIB library. Double precision version is DSCAL.  Class(es): D1a6 | Usage: CALL SSCLAL(N,S,SX,INCX) | On-line doc: CALL GAMSDOC SSCAL (or @PRT CMLIB*DOC.SSCAL/BLAS) | Tests: CMLIB*TEST-SOURCE.$Q/BLAS | Access: LIB NBS+CMLIB


SSIDI  Uses factorization of real symmetric indefinite matrix to compute its determinant and/or inverse.  Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library. Double precision version is DSIDI.  Class(es): D2bl1a D3bl1a | Usage: CALL SSIDI(A,LDA,N,KPVT,DET,INERT,WORK,JOB) | On-line doc: CALL GAMSDOC SSIDI (or @PRT CMLIB*DOC.SSIDI/LINPACKS) | Tests: CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS | Access: LIB NBS+CMLIB

SSIEV  Computes the eigenvalues and, optionally, the eigenvectors of a real symmetric matrix.  Portable single precision Fortran subprogram in LICEPACK sublibrary of CMLIB library.  Class(es): D4a1 | Usage: CALL SSIEV(A,LDA,N,E,WORK,JOB,INFO) | On-line doc: CALL GAMSDOC SSIEV (or @PRT CMLIB*DOC.SSIEV/LICEPACK) | Tests: CMLIB*TEST-SOURCE.SSIEV/LICEPACK | Access: LIB NBS+CMLIB


SSPAND Simple random sampling with proportion data - inferences regarding the population proportion and total. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L2c L4c Usage: CALL SSPAND (IP,NBR,ALPHA,STAT,IDIST,IER). On-line doc: CALL GAMSDOC SSPAND (or @PRT IMSL+DOC.SSPAND). Access: Lib NBS+IMSL

SSPBBLK Stratified random sampling with proportion data - inferences regarding the population proportion and total. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L2c L4c Usage: CALL SSPBLK (NBR,IN,ALPHA,PH,STAT,IER). On-line doc: CALL GAMSDOC SSPBLK (or @PRT IMSL+DOC.SSPBLK). Access: Lib NBS+IMSL


SSPD1 Uses factorization of real symmetric indefinite matrix stored in packed form to compute its determinant and/or inverse. Portable single precision Fortran subroutine in LINPACKS sublibrary of CMLIB library. Double precision version is DSPDI. Class(es): D2b4 D2b4 Usage: CALL SSPDI(AP,N,KPVT,DDET,INERT,WORK,IB). On-line doc: CALL GAMSDOC SSPDI (or @PRT CMLIB+DOC.SSPDI/LINPACKS) Tests: CMLIB+TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: Lib NBS+CMLIB | See also: SSPCO SSPFA


SSPSL Uses factorization of real symmetric indefinite matrix stored in packed form to solve systems. Portable single precision Fortran subroutine in LINPACKS sublibrary of CMLIB library. Double precision version is DSPSL. Class(es): D2b4 Usage: CALL SSPSL(AP,N,KPVT,B). On-line doc: CALL GAMSDOC SSPSL (or @PRT CMLIB+DOC.SSPSL/LINPACKS) Tests: CMLIB+TEST-SOURCE.$Q/LINPACKS, CMLIB+TEST-SOURCE.$F/LINPACKS | Access: Lib NBS+CMLIB | See also: SSPCO SSPFA

SSRAN1 Simple random sampling with continuous data - inferences regarding the population mean and total using ratio or regression estimation. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L2c L4a1 Usage: CALL SSRAND(Y,IY,IP,NBR,ALPHA,TEMP,XBAR,B,STAT,IER). On-line doc: CALL GAMSDOC SSRAND (or @PRT IMSL+DOC.SSRAND) Access: Lib NBS+IMSL

SSSAN1 Simple random sampling with continuous data - inferences regarding the population mean and total. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L2c L4a1 Usage: CALL SSAN1(Y,NBR,ALPHA,TEMP,HUMSIG,H,STAT,IER). On-line doc: CALL GAMSDOC SSAN1 (or @PRT IMSL+DOC.SSAN1) Access: Lib NBS+IMSL

SSBBLK Stratified random sampling with continuous data - inferences regarding the population mean and total. Proprietary single precision Fortran subroutine in IMSL library. Class(es): L2c L4a1 Usage: CALL SSBBLK(Y,NBR,NL,IN,ALPHA,TEMP,GMUSIG,H,STAT,IER). On-line doc: CALL GAMSDOC SSBBLK (or @PRT IMSL+DOC.SSBBLK) Access: Lib NBS+IMSL

SSSCAN Single stage cluster sampling with continuous data - inferences regarding the population mean and total.
Proproprietary single precision Fortran subprogram in IMSL library.  |  Class(es):  L2c  |  Usage:  CALL  SSSCAN(Y,IOPT,NBR,MC,IM,SIZE,TSIZE,ALPHA,TEMP,CMUSIG,IC,STAT,IER)  |  On-line doc:  CALL  GAMSDOC  SSSCAN  (or  @PRT  IMSL+DOC,SSSCAN)  |  Access:  LIB  NBS*IMSL

**SSSEST**  Two-stage sampling with continuous data and equisized primary units - inferences regarding the population mean and total.  |  Proprietary single precision Fortran subprogram in IMSL library.  |  Class(es):  L2c  L4a1a  |  Usage:  CALL  SSSEST(Y,NBR,ALPHA,TEMP,CMUSIG,JS,STAT,IER)  |  On-line doc:  CALL  GAMSDOC  SSSEST  (or  @PRT  IMSL+DOC,SSSEST)  |  Access:  LIB  NBS*IMSL


**STATS**  Computes 53 descriptive statistics for a single random sample.  |  Portable single precision Fortran subprogram in STATLIB library.  |  Class(es):  L1a1  |  Usage:  CALL  STATS(Y, N, SCRAT, NS)  |  On-line doc:  CALL  GAMSDOC  STATS  (or  @PRT  STATLIB+DOC,STATS)  |  Tests:  STATLIB*TEST-DEMO1  |  Access:  LIB  NBS*STATLIB

**STATSS**  Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation and computed results returned to the user.  |  Portable single precision Fortran subprogram in STATLIB library.  |  Class(es):  L1a1  |  Usage:  CALL  STATSS(Y, WT, N, SCRAT, NS, STAT, NPRINT)  |  On-line doc:  CALL  GAMSDOC  STATSS  (or  @PRT  STATLIB+DOC,STATSS)  |  Tests:  STATLIB*TEST-DEMO1  |  Access:  LIB  NBS*STATLIB

**STATSW**  Computes 53 descriptive statistics for a single random sample with individual weights assigned to each observation.  |  Portable single precision Fortran subprogram in STATLIB library.  |  Class(es):  L1a1  |  Usage:  CALL  STATSW(Y, WT, N, SCRAT, NS)  |  On-line doc:  CALL  GAMSDOC  STATSW  (or  @PRT  STATLIB+DOC,STATSW)  |  Tests:  STATLIB*TEST-DEMO1  |  Access:  LIB  NBS*STATLIB


**STEPWISE**  Performs stepwise linear regression using forward selection, backward elimination, conventional stepwise, or user intervention.  Options available through subcommands:  F-to-enter and F-to-remove, force and remove sets of variables, print next "best" (by the F-statistic) K alternatives.  |  Command in MINITAB Proprietary interactive system.  |  Class(es):  L8a5  |  Usage:  STEPwise regression of $y$ in $C_{i}$, predictors in $C_{i}$,...,$C_{j}$ [ subcommands FENTER = K; FREMOVE = K; FORCE $C_{i}$,...,$C_{j}$; ENTER $C_{i}$,...,$C_{j}$; REMOVE $C_{i}$,...,$C_{j}$; BEST K; STEPS = K ]  |  On-line doc:  HELP  STEPWISE  (in  Minitab)  |  Tests:  MINITAB*TEST-SOURCE.  |  Access:  @XQT  NBS*MINITAB.MINITAB (or  CALL  MINITAB in  CTS)

**STMOM3**  Computes the sample third central moment of the data in the input vector X.  |  Portable single precision Fortran subprogram in DATAPAC library.  |  Class(es):  L1a1c  |  Usage:  CALL  STMOM3(X,N,IWRITE,XSMOM3)  |  On-line doc:  CALL  GAMSDOC  STMOM3  (or  @PRT  DATAPAC+DOC,STMOM3)  |  Access:  LIB  NBS+DATAPAC

**STMOM4**  Computes the sample standardized fourth central moment of the data in the input vector X.  |  Portable single precision Fortran subprogram in DATAPAC library.  |  Class(es):  L1a1c  |  Usage:  CALL  STMOM4(X,N,IWRITE,XSMOM4)  |  On-line doc:  CALL  GAMSDOC  STMOM4  (or  @PRT  DATAPAC+DOC,STMOM4)  |  Access:  LIB  NBS+DATAPAC


**STRDI**  Computes determinant and/or inverse of real triangular matrix.  |  Portable single precision Fortran subprogram in LINPACKS sublibrary of CMLIB library.  |  Double precision version is DTRDI.  |  Class(es):  D2a3  D3a3  |  Usage:  CALL  STRDI(T,LDT,N,D,DET,JOB,INFO)  |  On-line doc:  CALL  GAMSDOC  STRDI  (or  @PRT  CMLIB+DOC,STRDI/LINPACKS)  |  Tests:  CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS  |  Access:  LIB  NBS+CMLIB

**STRLS**  Solves systems with real triangular matrix.  |  Portable single precision Fortran subprogram in LINPACKs sublibrary of CMLIB library.  |  Double precision version is DTRLS.  |  Class(es):  D2a3  |  Usage:  CALL  STRLS(T,LDT,N,B,JOB,INFO)  |  On-line doc:  CALL  GAMSDOC  STRLS  (or  @PRT  CMLIB+DOC,STRLS/LINPACKS)  |  Tests:  CMLIB*TEST-SOURCE.$Q/LINPACKS, CMLIB*TEST-SOURCE.$F/LINPACKS  |  Access:  LIB  NBS+CMLIB

**SUBSE1**  Carry over into Y all observations of vector X for which the corresponding elements in vector D are in the interval [DMIN,DMAX].  |  Portable single precision Fortran subprogram in DATAPAC library.  |  Class(es):  L2c  |  Usage:  CALL  SUBSE1(X,N,D,DMIN,DMAX,Y,NY)  |  On-line doc:  CALL  GAMSDOC  SUBSE1  (or  @PRT  DATAPAC+DOC,SUBSE1)  |  Access:  LIB  NBS+DATAPAC
SUBSE2  Carry over into Y all observations of vector X for which the corresponding elements in vector D1 are in the inclusive interval [D1MIN,D1MAX] and also for which the corresponding elements in D2 are in the interval [D2MIN,D2MAX]. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L2d | Usage: CALL SUBSE2(X,N,D1,D1MIN,D1MAX,D2,D2MIN,D2MAX,Y, NY) | On-line doc: CALL GAMSDOC SUBSE2 (or @PRT DATAPAC+DOC.SUBSE2) | Access: LIB NBS+DATAPAC

SUBSET  Retain all observations in vector X for which the corresponding elements in vector D are in the interval [DMIN,DMAX]. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L2d | Usage: CALL SUBSET (X,N,D,DMIN,DMAX,NEWN) | On-line doc: CALL GAMSDOC SUBSET (or @PRT DATAPAC+DOC.SUBSET) | Access: LIB NBS+DATAPAC

SUDS  Solves underdetermined systems of linear equations. For full rank matrices the minimum norm solution is returned, as well as an orthonormal basis for the null space of the matrix. If the system of equations is inconsistent only the least squares solution of minimal length is computed. Portable single precision Fortran subprogram in SUDSSODS sublibrary of CMLIB library. Class(es): D9 | Usage: CALL SUDS(A,X,B,NEQ,NUK,NRDA,IFLAG,MLSO,WORK,IWORK) | On-line doc: CALL GAMSDOC SUDS (or @PRT CMLIB+DOC.SUDS/SUDSSODS) | Tests: CMLIB+TEST-SOURCE.SUDS/SUDSSODS | Access: LIB NBS+CMLIB

SUMSL  Minimizes a general unconstrained objective function using analytic gradient and a Hessian approximation from a secant update. Portable single precision Fortran subprogram in NL2SN sublibrary of CMLIB library. Double precision version is DSUMSL. Class(es): Gl1b1b | Usage: CALL SUMSL(N,D,X,CALCF,GALCG,IV,LIV,LV,V,UIPARM,URPARM,UFPARM) | On-line doc: CALL GAMSDOC SUMSL (or @PRT CMLIB+DOC.SUMSL/NL2SN) | Tests: CMLIB+TEST-SOURCE.$F2/NL2SN, CMLIB+TEST-SOURCE.$Q2/NL2SN | Access: LIB NBS+CMLIB

TABLE Probes and prints one-way, two-way, and multi-way tables of counts with 20 optional subcommands for summarizing (e.g., cell mean, standard deviation), marginals, performing chi-square tests for each 2-way table, handling missing values, and selecting forms of input and output. | Command in MINITAB Proprietary interactive system. | Class(es): L4a1 L4a2 L9 | Usage: TABLE the data classified by C,...; [subcommands MEANS for C, ..., C; MEDIAN,...; SUMS,...; MINIMUMS,...; MAXIMUMS,...; STDEV,...; STATS,...; DATA,...; NONMISSING,...; PROPORTION,...; COUNTS; ROWPERCENTS; COLPERCENTS; TOTPERCENTS; CHISQUARE,...; NOALL; ALL,...; MISSING,...; FREQUENCIES,...; LAYOUT,...] | On-line doc: HELP TABLE (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

TAIL Performs a symmetric distribution tail length analysis on the input vector X. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L4a1a | Usage: CALL TAIL(X,N) | On-line doc: GAMSDOC TAIL (or @PRT DATAPAC+DOC.TAIL) | Access: LIB NBS+DATAPAC

TAN Computes the elementary tangent function. If your Fortran library includes this function, use that instead. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DTANH. | Class(es): C4a | Usage: X = TAN (X) | On-line doc: CALL GAMSDOC TAN (or @PRT PORT+DOC.TAN) | Access: LIB NBS+PORT

TANH Computes hyperbolic tangent, tanh(x). | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DTANH. | Class(es): C4c | Usage: X = TANH (X) | On-line doc: CALL GAMSDOC TANH (or @PRT PORT+DOC.TANH) | Access: LIB NBS+PORT

TCDF Computes the cumulative distribution function value for Student's t distribution with degrees of freedom parameter NU. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L6a1t | Usage: CALL TCDF(X,NU,CDF) | On-line doc: CALL GAMSDOC TCDF (or @PRT DATAPAC+DOC.TCDF) | Access: LIB NBS+DATAPAC

TCBP Evaluates a polynomial expressed as a sum of Chebyshev polynomials. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DTCHBP. | Class(es): C3a2 | Usage: X = TCBP (N, ALPHA, X, X0, X1) | On-line doc: CALL GAMSDOC TCBP (or @PRT PORT+DOC.TCBP) | Access: LIB NBS+PORT


TIME Performs a time series analysis on the data in the input vector X. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L10c | Usage: CALL TIME(X,N) | On-line doc: CALL GAMSDOC TIME (or @PRT DATAPAC+DOC.TIME) | Access: LIB NBS+DATAPAC

TINTERVAL Calculates a t-confidence interval with specified percent confidence. | Command in MINITAB Proprietary interactive system. | Class(es): L4a1a L4a2 | Usage: TINTERval with K percent confidence for data in column C | On-line doc: HELP TINTERVAL (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

TVI Calculates eigenvectors of symmetric tridiagonal matrix corresponding to some specified eigenvalues, using inverse iteration. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4e3 | Usage: CALL TINVIT(NM,N,D,E2,M,W,IND,Z,IERI,RV,TV,TV2,TV3,TV4,TV6) | On-line doc: CALL GAMSDOC TINVIT (or @PRT CMLIB+DOC.TINVIT/EISPACK) | Access: LIB NBS+CMLIB

TOR Computes normal and distribution-free tolerance limits for the data in the input vector X. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L4a1r | Usage: CALL TOL(X,N) | On-line doc: CALL GAMSDOC TOL (or @PRT DATAPAC+DOC.TOL) | Access: LIB NBS+DATAPAC

TPLOT Prints pseudo three-dimensional plot of y versus x versus z, with symbols indicating the values of z, and with optional scale specification. | Command in MINITAB Proprietary interactive system. | Class(es): L3q2t Q1 | Usage: TPLOT y in C [from K to K] vs x in C [from K to K] vs z in C | On-line doc: HELP TPLOT (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @QXT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

TPPLT Generates a Student's t probability plot with degrees of freedom parameter NU. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L3c4t | Usage: CALL TPPLT(X,N,NU) | On-line doc: CALL GAMSDOC TPPLT (or @PRT DATAPAC+DOC.TPLT) | Access: LIB NBS+DATAPAC

TPPF Computes the percent point function value for the Student's t distribution with degrees of freedom parameter NU. | Portable single precision Fortran subroutine in DATAPAC library. | Class(es): L5a2t | Usage: CALL TPPF(P,NU,PPF) | On-line doc: CALL GAMSDOC TPPF (or @PRT CMLIB+DOC.TPPF) | Access: LIB NBS+DATAPAC

TQL1 Computes eigenvalues of symmetric tridiagonal matrix by QR method. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4e2a | Usage: CALL TQL1(N,D,E,IERR) | On-line doc: CALL GAMSDOC TQL1 (or @PRT CMLIB+DOC.TQL1/EISPACK) | Access: LIB NBS+CMLIB

TQL2 Compute eigenvalues and eigenvectors of symmetric tridiagonal matrix. | Portable single precision Fortran subroutine in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4e2a | Usage: CALL TQL2(NM,N,D,E,Z,IERR) | On-line doc: CALL GAMSDOC TQL2...
TRBAK1 Forms the eigenvectors of real symmetric matrix from eigenvectors of symmetric tridiagonal matrix formed by TRED1.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRBAK1(NM,N,A,E,M,Z) | On-line doc: CALL GAMSdoc TRBAK1 (or @PRT CMLIB+DOC.TRBAK1/EISPACK) | Access: LIB NBS*CMLIB | See also: TRED1
TRBAK1 Forms the eigenvectors of real symmetric matrix from eigenvectors of symmetric tridiagonal matrix formed by TRED3.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRBAK3(NM,N,NV,A,M,Z) | On-line doc: CALL GAMSdoc TRBAK3 (or @PRT CMLIB+DOC.TRBAK3/EISPACK) | Access: LIB NBS+CMLIB | See also: TRED3
TRED1 Reduce real symmetric matrix to symmetric tridiagonal matrix using orthogonal similarity transformations.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRED1(NM,N,A,D,E,E2) | On-line doc: CALL GAMSdoc TRED1 (or @PRT CMLIB+DOC.TRED1/EISPACK) | Access: LIB NBS*CMLIB
TRED2 Reduce real symmetric matrix to symmetric tridiagonal matrix using and accumulating orthogonal transformations.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRED2(NM,N,A,D,E,Z) | On-line doc: CALL GAMSdoc TRED2 (or @PRT CMLIB+DOC.TRED2/EISPACK) | Access: LIB NBS+CMLIB | See also: TQL2,INTQ2L
TRED3 Reduce real symmetric matrix stored in packed form to symmetric tridiagonal matrix using orthogonal transformations.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4c1b1 | Usage: CALL TRED3(N,NV,A,D,E,E2) | On-line doc: CALL GAMSdoc TRED3 (or @PRT CMLIB+DOC.TRED3/EISPACK) | Access: LIB NBS*CMLIB
TRIDIB Computes eigenvalues of symmetric tridiagonal matrix in given interval using Sturm sequencing.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TRIDIB(N,EPS1,E,D,E2,LB,UB,M11,M,W,IND,IERR,RV4,RV5) | On-line doc: CALL GAMSdoc TRIDIB (or @PRT CMLIB+DOC.TRIDIB/EISPACK) | Access: LIB NBS*CMLIB
TRIGP Evaluates a trigonometric polynomial with given coefficients.
    Proprietary single precision Fortran program in PORT library.
    Double precision version is DTRIGP. | Class(es): C3at | Usage: X = TRIGP (N, ALPHA, BETA, THETA) | On-line doc: CALL GAMSdoc TRIGP (or @PRT PORT+DOC.TRIGP) | Access: LIB NBS*PORT
TRIM Computes the sample trimmed mean of the data in the input vector X.
    Portable single precision Fortran program in DATAPAC library. | Class(es): L1a1a | Usage: CALL TRIM(X,N,P1,P2,IWRITE,XTRIM) | On-line doc: CALL GAMSdoc TRIM (or @PRT DATAPAC+DOC.TRIM) | Access: LIB NBS*DATAPAC
TSDIFF Performs a user-controlled differing operation on a series.
    Portable single precision Fortran program in STATLIB library. | Class(es): L1b6 | Usage: CALL TSDIFF (Y, N, NDIFAC, ND, IOD, NDIF, NDIFF) | On-line doc: CALL GAMSdoc TSDIFF (or @PRT STATLIB+DOC.TSDIFF) | Tests: STATLIB+TEST.DEM03 | Access: LIB NBS*STATLIB
TSPLOT Prints a scatter diagram of a time series, optionally using symbols modulo the period. Handles missing values.
    Command in MINITAB Proprietary interactive system. | Class(es): L3c5 Q1 | Usage: TSPLOT [with period K [starting at K]] for data in C on-line | help: TSPLOT (in minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
TSTURM Computes eigenvalues of symmetric tridiagonal matrix in given interval and eigenvectors by Sturm sequencing.
    Portable single precision Fortran program in EISPACK sublibrary of CMLIB library. | Class(es): D4a5 D4c2a | Usage: CALL TSTURM(N,M,N,EPS1,D,E2,EB,UB,MB,M,W,Z,IERR,RV1,RV2,RV3,RV4,RV5) | On-line doc: CALL GAMSdoc TSTURM (or @PRT CMLIB+DOC.TSTURM/EISPACK) | Access: LIB NBS*DATAPAC
TTTEST Performs one- or two-sided t-tests.
TWSAMPLE Performs one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample or the pooled sample.
    Command in MINITAB Proprietary interactive system. | Class(es): L4b1a14 L4b2 | Usage: TWSAMPLE [K percent confidence] for data in C and C; subcommands ALTERNATIVE = K; POOLED. | On-line doc: HELP TWSAMPLE (in minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)
TWOT Performs one- or two-sided two-sample t-test of equality of two population means with variance estimated either from each sample
or the pooled sample. | Command in MINITAB Proprietary interactive system. Class(es): L4h1a14 L4h2 | Usage: TWOT [K percent confidence] for data in C, groups in C ]; subcommands ALTERNATIVE = K; POOLED.] | On-line doc: HELP TWOT (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

**TWOWAYAOV** Performs two-way analysis of variance for balanced data (equal number of observations, one or more, in each cell) and prints standard results. Options: fit additive model, save results. | Command in MINITAB Proprietary interactive system. Class(es): L7a2a1a | Usage: TWOWAYAOV for data in C, subscripts in C, C [store residuals in C [fits in C]]; subcommand ADDITIVE.] | On-line doc: HELP TWOWAYAOV (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)
U

UERSET Set message level for IMSL routine UERTST. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R3a | Usage: CALL UERSET (LEVEL,LEVOLD) | On-line doc: CALL GAMSDOC UERSET (or @PRT IMSL+DOC.UERSET) | Access: LIB NBS+IMSL

UERTST Print a message reflecting an error condition. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R3c | Usage: CALL UERTST (IER,NAME) | On-line doc: CALL GAMSDOC UERTST (or @PRT IMSL+DOC.UERTST) | Access: LIB NBS+IMSL

UGETTO To retrieve current values and to set new values for input and output unit identifiers. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R1 R3b | Usage: CALL UGETTO (IOTP,NIN,NOUT) | On-line doc: CALL GAMSDOC UGETTO (or @PRT IMSL+DOC.UGETTO) | Access: LIB NBS+IMSL

UHELP Display methods of obtaining info on IMSL conventions regarding various subjects provide means for individual sites to supply users with site specific info. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP | On-line doc: CALL GAMSDOC UHELP (or @PRT IMSL+DOC.UHELP) | Access: LIB NBS+IMSL

UHELP1 Write information regarding IMSL conventions and notation to an output file. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP1 | On-line doc: CALL GAMSDOC UHELP1 (or @PRT IMSL+DOC.UHELP1) | Access: LIB NBS+IMSL

UHELP2 Write information regarding IMSL input and output conventions. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP2 | On-line doc: CALL GAMSDOC UHELP2 (or @PRT IMSL+DOC.UHELP2) | Access: LIB NBS+IMSL

UHELP3 Write information regarding IMSL error detecting facilities. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP3 | On-line doc: CALL GAMSDOC UHELP3 (or @PRT IMSL+DOC.UHELP3) | Access: LIB NBS+IMSL

UHELP4 Write information regarding matrix/vector storage modes used in IMSL subroutines. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): R4 | Usage: CALL UHELP4 | On-line doc: CALL GAMSDOC UHELP4 (or @PRT IMSL+DOC.UHELP4) | Access: LIB NBS+IMSL

ULSIA Finds the minimal length solution of the undetermined system of equations AX=B where A is an m by n matrix with m <= n. Flexible version of SGLSS. | Portable single precision Fortran subprogram in SGLSS sublibrary of CMLIB library. | Class(es): D9 | Usage: CALL ULSIA(A,MDA,M,N,B,MB,NB,RE,AE,KEY,MOND,NC,RC,TKRAN,NSURE,RLM,NL,WR,WJL,WOK,LI,WINFO) | On-line doc: CALL GAMSDOC ULSIA (or @PRT CMLIB+DOC.ULSIA/SGLSS) | Access: LIB NBS+CMLIB

UMB Given interval endpoints, this generates a uniform mesh, with needed multiplicities for B-spline use. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DUMB. | Class(es): E3 K6 | Usage: CALL UMB (A,B,NAB,K,X,NX) | On-line doc: CALL GAMSDOC UMB (or @PRT PORT+DOC.UMB) | Access: LIB NBS+PORT

UMD Given interval endpoints, this generates a uniform mesh of distinct points. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DUMD. | Class(es): E3 K6 | Usage: CALL UMD (A,B,NAB,X) | On-line doc: CALL GAMSDOC UMD (or @PRT PORT+DOC.UMD) | Access: LIB NBS+PORT

UMKFL Decomposes a non-zero floating point number into a mantissa and an exponent. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DUMKFL. | Class(es): AEC | Usage: CALL UMKFL (F,E,M) | On-line doc: CALL GAMSDOC UMKFL (or @PRT PORT+DOC.UMKFL) | Access: LIB NBS+PORT

UNI Generates uniformly distributed random numbers on the interval [0,1]. UNI's main advantages are a long cycle and a high degree of reproducibility on other machines (it runs on any machine with at least 16 bit integer arithmetic). Portable single precision Fortran subprogram in RV sublibrary of CMLIB library. | Class(es): L6a21 | Usage: S=UNI(JD) | On-line doc: CALL GAMSDOC UNI (or @PRT CMLIB+DOC.UNI/RV) | Tests: CMLIB*TEST-SOURCE.UNI/UNI | Access: LIB NBS+CMLIB

UNI Returns a single random real variate from the uniform [0,1] distribution. | Proprietary single precision Fortran subprogram in PORT library. | Class(es): L6a21 | Usage: X = UNI (K) | On-line doc: CALL GAMSDOC UNI (or @PRT PORT+DOC.UNI) | Access: LIB NBS+PORT | See also: RANBY7,RANSET

UNICDF Computes the cumulative distribution function value for the uniform (rectangular) distribution on the unit interval [0,1]. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a1u | Usage: CALL UNICDF(X,CDF) | On-line doc: CALL GAMSDOC UNICDF (or @PRT DATAPAC+DOC.UNICDF) | Access: LIB NBS+DATAPAC

UNIPDF Computes the probability density function value for the uniform (rectangular) distribution on the unit interval [0,1]. Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L6a1u | Usage: CALL UNIPDF(X,PDF) | On-line doc: CALL GAMSDOC UNIPDF (or @PRT DATAPAC+DOC.UNIPDF) | Access: LIB NBS+DATAPAC

UNIPLT Generates a uniform probability plot on the unit interval (0,1) with mean = 0.5 and standard deviation = sqrt(1/12). Portable single precision Fortran subprogram in DATAPAC library. | Class(es): L3ctu | Usage: CALL UNIPLT(X,N) | On-line doc: CALL GAMSDOC UNIPLT (or @PRT DATAPAC+DOC.UNIPLT) | Access: LIB NBS+DATAPAC
UNIPPF Computes the percent point function value for the uniform (rectangular) distribution on the unit interval (0, 1). Portable single precision Fortran subroutine in DATAPAC library. [Class(es): L6a2u] Usage: CALL UNIPPF(P,PPF) On-line doc: CALL GAMSDOC UNIPPF (or @PRT DATAPAC+DOC.UNIPPF) Access: LIB NBS+DATAPAC

UNIRAN Generates a random sample of size N from the uniform (rectangular) distribution on the unit interval (0, 1). Portable single precision Fortran subroutine in DATAPAC library. [Class(es): L6a21] Usage: CALL UNIRAN(N,ISTART,X) On-line doc: CALL GAMSDOC UNIRAN (or @PRT DATAPAC+DOC.UNIRAN) Access: LIB NBS+DATAPAC

UNISF Computes the sparsity function value for the uniform (rectangular) distribution on the unit interval (0, 1). Portable single precision Fortran subroutine in DATAPAC library. [Class(es): L5a2u] Usage: CALL UNISF(P, SF) On-line doc: CALL GAMSDOC UNISF (or @PRT DATAPAC+DOC.UNISF) Access: LIB NBS+DATAPAC

URANDOM Generates K pseudo-random numbers from the uniform (0,1) distribution. Command in MINITAB Proprietary interactive system. Class(es): L6a21 Usage: URANDom K observations, put into C On-line doc: HELP URANDOM (in Minibat) Tests: MINITAB+TEST-SOURCE. Access: @XT T NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

USBOX Print a boxplot (k samples). Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3d Usage: CALL USBOX(X,K,N,MAXL,IERR) On-line doc: CALL GAMSDOC USBOX (or @PRT IMSL+DOC.USBOX) Access: LIB NBS+IMSL

USCWV Print a complex vector. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): N1 Usage: CALL USCWV(ITITLE, NC, A, INC, IOPT) On-line doc: CALL GAMSDOC USCWV (or @PRT IMSL+DOC.USCWV) Access: LIB NBS+IMSL

USHST Print a horizontal histogram. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3a Q1 Usage: CALL USHST(T, N, IOPT, IER) On-line doc: CALL GAMSDOC USHST (or @PRT IMSL+DOC.USHST) Access: LIB NBS+IMSL

USHST Print a vertical histogram. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3a Q1 Usage: CALL USHST(T, N, ISP, IER) On-line doc: CALL GAMSDOC USHST (or @PRT IMSL+DOC.USHST) Access: LIB NBS+IMSL

USHST2 Print a vertical histogram, plotting two frequencies with one bar of the histogram. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3a Usage: CALL USHST2(T, U, N, ISP, IER) On-line doc: CALL GAMSDOC USHST2 (or @PRT IMSL+DOC.USHST2) Access: LIB NBS+IMSL

USLEAP Print results of the best-regressions analysis performed by IMSL routine RLEAP. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L6b Usage: CALL USLEAP(IJOB, KZ, IXS, STAT, IXV, NVAR, IXB, BEST, IB) On-line doc: CALL GAMSDOC USLEAP (or @PRT IMSL+DOC.USLEAP) Access: LIB NBS+IMSL See also: RLEAP

USMNMX Determination of the minimum and maximum values of a vector. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): D1a2 Usage: CALL USMNMX(X,N,INC,XMIN,XMAX) On-line doc: CALL GAMSDOC USMNMX (or @PRT IMSL+DOC.USMNMX) Access: LIB NBS+IMSL

USPC Print a sample pdf, a theoretical pdf and confidence band information; plot these on option. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L1a3 Usage: CALL USPC (PDF,X,N,N12,N05,IP,IC,W) On-line doc: CALL GAMSDOC USPC (or @PRT IMSL+DOC.USPC) Access: LIB NBS+IMSL

USPDF Plot of two sample probability distribution functions against their spectra. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3b Usage: CALL USPDF (X,N,M,W,IW,IR) On-line doc: CALL GAMSDOC USPDF (or @PRT IMSL+DOC.USPDF) Access: LIB NBS+IMSL

USPLOR Printer plot of up to ten functions. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3b Q1 Usage: CALL USPLO (X, Y, N, M, NC, ITITLE, NTITLE, ILXLABL, NLXLABL, YLABL, NLYLABL, RANGE, ICHAR, IOPT, IER) On-line doc: CALL GAMSDOC USPLO (or @PRT IMSL+DOC.USPLO) Access: LIB NBS+IMSL

USPLOD Printer plot of up to ten functions. Proprietary double precision Fortran subroutine in IMSL library. [Class(es): L3b Q1 Usage: CALL USPLOD (X, Y, N, M, INC, ITITLE, NTITLE, ILXLABL, NLXLABL, YLABL, NLYLABL, RANGE, ICHAR, IOPT, IER) On-line doc: CALL GAMSDOC USPLOD (or @PRT IMSL+DOC.USPLOD) Access: LIB NBS+IMSL

USPRP Probability plot. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3d Usage: CALL USPRP (X, N, N1, N2, IDIST, IOPT, WK, IER) On-line doc: CALL GAMSDOC USPRP (or @PRT IMSL+DOC.USPRP) Access: LIB NBS+IMSL

USSLF Print a stem-and-leaf display. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L3d Usage: CALL USSLF (X,N,INT,MAXL) On-line doc: CALL GAMSDOC USSLF (or @PRT IMSL+DOC.USSLF) Access: LIB NBS+IMSL

USTREE Print a binary tree (which may represent the output of a clustering algorithm in chapter 9). Proprietary single precision Fortran subroutine in IMSL library. [Class(es): L14c N4 Q1 Usage: CALL USTREE (ND, JCLSON, ICORSON, CLEVEL, IND, XSM, IOUT, CLVLSK, NCLRST, LEFRTR, STARST, I) On-line doc: CALL GAMSDOC USTREE (or @PRT IMSL+DOC.USTREE) Access: LIB NBS+IMSL

USWBM Print a matrix stored in band storage mode. Proprietary single precision Fortran subroutine in IMSL library. [Class(es): N1 Usage: CALL USWBM (ITITLE, NC, A, IA, M, NUC, NLC, WK, IOPT) On-line doc: CALL GAMSDOC USWBM (or @PRT IMSL+DOC.USWBM)
USWBS  Print a matrix stored in band symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWBS (ITITLE,NC,A,IA,M,NLC,WK,IOPT) | On-line doc: CALL GAMSDOC USWBS (or @PRT IMSL+DOC.USWBS) | Access: LIB NBS*IMSL


USWCM  Print a complex matrix stored in full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWCM (ITITLE,NC,A,IA,N,M,IOPT) | On-line doc: CALL GAMSDOC USWCM (or @PRT IMSL+DOC.USWCM) | Access: LIB NBS*IMSL

USWF M  Print a matrix stored in full storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWF (ITITLE,NC,A,IA,N,M,IOPT) | On-line doc: CALL GAMSDOC USWF (or @PRT IMSL+DOC.USWF) | Access: LIB NBS*IMSL


USWSM  Print a matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): N1 | Usage: CALL USWSM (ITITLE,NC,A,M,IOPT) | On-line doc: CALL GAMSDOC USWSM (or @PRT IMSL+DOC.USWSM) | Access: LIB NBS*IMSL
VABM XF Maximum absolute value of the elements of a vector or a subset of the elements of a vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dia 3e | Usage: CALL VABM XF (V,L,INC,J,VMAX) | On-line doc: CALL GAMS DOC VABM XF (or @PRT IMS L+DOC.VABM XF) | Access: LIB NBS+1MSL

VABMX S Maximum absolute value of the elements of a row or column of a matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dia 3e | Usage: CALL VABM XS (V,L,RO J, J, VMAX) | On-line doc: CALL GAMS DOC VABM XS (or @PRT IMS L+DOC.VABM XS) | Access: LIB NBS+1MSL

VABSM F Sum of the absolute values of the elements of a vector or a subset of a vector. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dia 3a | Usage: CALL VABSM F (V,L,INC,VSUM) | On-line doc: CALL GAMS DOC VABSM F (or @PRT IMS L+DOC.VABSM F) | Access: LIB NBS+1MSL

VABSM S Sum of the absolute values of the elements of a row (or column) of a matrix stored in symmetric storage mode. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dia 3a | Usage: CALL VABSM S (V,L,RO J, VSUM) | On-line doc: CALL GAMS DOC VABSM S (or @PRT IMS L+DOC.VABSM S) | Access: LIB NBS+1MSL

VAR Computes the sample variance (with denominator N-1) of the data in the input vector X. | Portable single precision Fortran subprogram in DATAPAC library. | Class(es): Li a1b | Usage: VAR(X,N,IWRITE,XVAR) | On-line doc: CALL GAMS DOC VAR (or @PRT DATAPAC+DOC.VAR) | Access: LIB NBS+DATAPAC

VBTOD Converts a mantissa and exponent into a base 10 floating point number. | Proprietary single precision Fortran subprogram in PORT library. Double precision version is DVB TOD. | Class(es): A6b | Usage: CALL VBTOD (E, M, E10, M10) | On-line doc: CALL GAMS DOC VBTOD (or @PRT PORT+DOC.VBTOD) | Access: LIB NBS+PORT

VCONVO Vector convolution. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dia 10 J2 | Usage: CALL VCON VO (A,B,LA,LIB,WK) | On-line doc: CALL GAMS DOC VCON VO (or @PRT IMS L+DOC.VCON VO) | Access: LIB NBS+1MSL


VCVTFS Storage mode conversion of matrices (full to symmetric). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dib 9 | Usage: CALL VCVTFS (A,N,IA,B) | On-line doc: CALL GAMS DOC VCVTFS (or @PRT IMS L+DOC.VCVTFS) | Access: LIB NBS+1MSL

VCVTHC Storage mode conversion of matrices (Hermitian to full complex). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dib 9 | Usage: CALL VCVTHC (H,N,B,IB) | On-line doc: CALL GAMS DOC VCVTHC (or @PRT IMS L+DOC.VCVTHC) | Access: LIB NBS+1MSL


VCVTQS Storage mode conversion (band symmetric to symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dib 9 | Usage: CALL VCVTQS (A,N,NC,J,A,B) | On-line doc: CALL GAMS DOC VCVTQS (or @PRT IMS L+DOC.VCVTQS) | Access: LIB NBS+1MSL


VCVTSQ Storage mode conversion (symmetric to band symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): Dib 9 | Usage: CALL VCVTSQ (A,N,NC,B,IB) | On-line doc: CALL GAMS DOC VCVTSQ (or @PRT IMS L+DOC.VCVTSQ) | Access: LIB NBS+1MSL

VDCPS Decompose an integer into its prime factors. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): B A1 | Usage: CALL VDCPS (N,NF P, IPF, IE XP, IPWR) | On-line doc: CALL GAMS DOC VDCPS (or @PRT IMS L+DOC.VDCPS) | Access:
LIB NBS*IMSL

VDTOB  Convert a base-10 mantissa and exponent of a floating point number into a machine-base representation.  Proprietary single precision Fortran subprogram in PORT library. Double precision version is DVTOB.  | Class(es): Adc  | Usage: CALL VDTOB ( E10, M10, E, M )  | On-line doc: CALL GAMSDOC VDTOB ( or @PRT PORT*DOC.VDTOB )  | Access: Lib NBS*PORT


VHS12C Complex Householder transformation to zero a single element of a matrix.  Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): D1b11 | Usage: CALL VHS12C ( AJR, AJP1, AJP1, UP, VJ, VJP1 )  | On-line doc: CALL GAMSDOC VHS12C ( or @PRT IMSL*DOC.VHS12C )  | Access: Lib NBS*IMSL

VHS12R Real Householder transformation to zero a single element of a matrix.  Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): D1b11 | Usage: CALL VHS12R ( AJ, AJP1, U, UJP1, V, VJP1 )  | On-line doc: CALL GAMSDOC VHS12R ( or @PRT IMSL*DOC.VHS12R )  | Access: Lib NBS*IMSL


VIPRSS Vector inner product of two vectors or subsets of two vectors.  Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): D1a4 | Usage: CALL VIPRSS ( X,Y,L,X,Y,XYIP )  | On-line doc: CALL GAMSDOC VIPRSS ( or @PRT IMSL*DOC.VIPRSS )  | Access: Lib NBS*IMSL

VIPRSS Vector inner product of two vectors each of which is part of some matrix stored in symmetric mode.  Proprietary single precision Fortran subprogram in IMSL library.  | Class(es): D1a4 | Usage: CALL VIPRSS ( X,Y,L,IY,XYIP )  | On-line doc: CALL GAMSDOC VIPRSS ( or @PRT IMSL*DOC.VIPRSS )  | Access: Lib NBS*IMSL


IMSL*DOC.VMULQF) | Access: LIB NBS*IMSL

VMULQQ Matrix multiplication (band symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULQQ (A,N,NCA,IA,B,NCB,IB,C,IC) | On-line doc: CALL GAMSDOC VMULQQ (or @PRT IMSL*DOC.VMULQ)) | Access: LIB NBS*IMSL


VMULSS Matrix multiplication (symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VMULSS (A,B,N,NC,IC) | On-line doc: CALL GAMSDOC VMULSS (or @PRT IMSL*DOC.VMULSS) | Access: LIB NBS*IMSL

VNRMF1 1-norm of matrices (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VNRMF1 (A,N,IA,XNRMMA) | On-line doc: CALL GAMSDOC VNRMF1 (or @PRT IMSL*DOC.VNRMF1) | Access: LIB NBS*IMSL


VNRMF1 Infinity-norm of matrices (full storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VNRMF1 (A,N,IA,XNRMMA) | On-line doc: CALL GAMSDOC VNRMF1 (or @PRT IMSL*DOC.VNRMF1) | Access: LIB NBS*IMSL

VNRMFS 1-norm of matrices (symmetric storage mode). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): D1b6 | Usage: CALL VNRMFS (A,N,IA,XNRMMA) | On-line doc: CALL GAMSDOC VNRMFS (or @PRT IMSL*DOC.VNRMFS) | Access: LIB NBS*IMSL


VPLT Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLT (Y, N, NPL) | On-line doc: CALL GAMSDOC VPLT (or @PRT STATLIB*DOC.VPLT) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLT2 Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLT2 (Y1, Y2, N, NPL, NSCALE) | On-line doc: CALL GAMSDOC VPLT2 (or @PRT STATLIB*DOC.VPLT2) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLT2L Displays an Nx100 character line printer plot of the N values of each of two series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLT2L (Y1, Y2, N, NPL, NSCALE, LMSPEC, Y1LB, Y1UB, Y2LB, Y2UB) | On-line doc: CALL GAMSDOC VPLT2L (or @PRT STATLIB*DOC.VPLT2L) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLTB Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis). | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLTB (Y, N, NPL) | On-line doc: CALL GAMSDOC VPLTB (or @PRT STATLIB*DOC.VPLTB) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLTBL Displays an Nx100 character line printer bar plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLTBL (Y, N, NPL, YLB, YUB) | On-line doc: CALL GAMSDOC VPLTBL (or @PRT STATLIB*DOC.VPLTBL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB

VPLTL Displays an Nx100 character line printer plot of the N values of a series (horizontal axis) vs. their indices (vertical axis), with user control of the plot limits. | Portable single precision Fortran subprogram in STATLIB library. | Class(es): L3c5 Q1 | Usage: CALL VPLTL (Y, N, NPL, YLB, YUB) | On-line doc: CALL GAMSDOC VPLTL (or @PRT STATLIB*DOC.VPLTL) | Tests: STATLIB*TEST.DEMO1 | Access: LIB NBS*STATLIB
GAMS: Module Dictionary


VSAR  Sorting of matrices (with options).  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): N6a2b1 N6a1b1  | Usage: CALL VSAR(A,IA,NR,NC,IOp,KPOS,NK,IR,WK,IER)  | On-line doc: CALL GAMSDOC VSAR (or @PRT IMSL+DOC.VSAR)  | Access: LIB NBS+IMSL


VSRTU  Interchange the rows or columns of a matrix using a permutation vector such as the one obtained from IMSL routines VSRTP or VSRTR.  | Proprietary single precision Fortran subroutine in IMSL library.  | Class(es): N8  | Usage: CALL VSRTU (Z,IZ,N,M,IND,IR,WK)  | On-line doc: CALL GAMSDOC VSRTU (or @PRT IMSL+DOC.VSRTU)  | Access: LIB NBS+IMSL


January 1984
W

WALSH Calculates (X[i] + X[j])/2 and stores these average and their indices (useful for nonparametric tests and confidence intervals). Command in MINITAB Proprietary interactive system. Class(es): L2a | Usage: WALSh averages of values in C, put into C [indices into C and C] | On-line doc: HELP WALSH (in MinTab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)

WEIB Performs a Weibull distribution analysis on the data in the input vector X. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L4a1a23 | Usage: CALL WEIB(X,N) | On-line doc: CALL GAMSDOC WEIB (or @PRD DATAPAC*DOC.WEIB) | Access: LIB NBS*DATAPAC

WEICDF Computes the cumulative distribution function value for the Weibull distribution with shape parameter C. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L5a1w | Usage: CALL WEICDF(X,Gamma,CDF) | On-line doc: CALL GAMSDOC WEICDF (or @PRD DATAPAC*DOC.WEICDF) | Access: LIB NBS*DATAPAC

WEIPLT Generates a Weibull probability plot with tail length parameter GAMMA. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L5c4w | Usage: CALL WEIPLT(X,N,Gamma) | On-line doc: CALL GAMSDOC WEIPLT (or @PRD DATAPAC*DOC.WEIPLT) | Access: LIB NBS*DATAPAC

WEIPPF Computes the percent point function value for the Weibull distribution with tail length parameter GAMMA. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L5a2w | Usage: CALL WEIPPF(P,Gamma,PPF) | On-line doc: CALL GAMSDOC WEIPPF (or @PRD DATAPAC*DOC.WEIPPF) | Access: LIB NBS*DATAPAC

WEIRAN Generates a random sample of size N from the Weibull distribution with tail length parameter GAMMA. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L6a23 | Usage: CALL WEIRAN(N,Gamma,ISTART,X) | On-line doc: CALL GAMSDOC WEIRAN (or @PRD DATAPAC*DOC.WEIRAN) | Access: LIB NBS*DATAPAC

WIND Computes the sample Windsorized mean of the data in the input vector X. Portable single precision Fortran subprogram in DATAPAC library. Class(es): L1a1 | Usage: CALL WIND(X,N,P,P1,P2,IWRITE,XWIND) | On-line doc: CALL GAMSDOC WIND (or @PRD DATAPAC*DOC.WIND) | Access: LIB NBS*DATAPAC

WINTERVAL Calculates a one-sample Wilcoxon rank estimate and confidence interval for the center of a symmetric distribution. Command in MINITAB Proprietary interactive system. Class(es): L4a1b | Usage: WINTerval [percent confidence K] for data in C,...,C | On-line doc: HELP WINTERVAL (in MinTab) | Tests: MINITAB*TEST-SOURCE. | Access: @XQT NBS*MINITAB.MINITAB (or CALL MINITAB in CTS)


WRITE Writes out the contents of the vector X in an orderly and neat fashion. Portable single precision Fortran subprogram in DATAPAC library. Class(es): N1 | Usage: CALL WRITE(X,N,NNLINE,IWIDTH,IDEC) | On-line doc: CALL GAMSDOC WRITE (or @PRD DATAPAC*DOC.WRITE) | Access: LIB NBS*DATAPAC

X01AAE  Proprietary single precision Fortran subprogram in NAG library. Double precision version is X01AAB. | Class(es): R1 | Usage: D = X01AAE (X) | On-line doc: CALL GAMSDOC X01AAE (or @PTRT NAG*DOC.X01AAE) | Access: LIB NBS*NAG

X01AAB  Proprietary double precision Fortran subprogram in NAG library. Single precision version is X01AAE. | Class(es): R1 | Usage: D = X01AAB (X) | On-line doc: CALL GAMSDOC X01AAB (or @PTRT NAG*DOC.X01AAB) | Access: LIB NBS*NAG

X01ABE  Euler's constant, gamma. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X01ABF. | Class(es): R1 | Usage: D = X01ABE (X) | On-line doc: CALL GAMSDOC X01ABE (or @PTRT NAG*DOC.X01ABE) | Access: LIB NBS*NAG

X01ABF  Euler's constant, gamma. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X01ABF. | Class(es): R1 | Usage: D = X01ABF (X) | On-line doc: CALL GAMSDOC X01ABF (or @PTRT NAG*DOC.X01ABF) | Access: LIB NBS*NAG

X02AAE  Smallest possible e such that 1.0+e > 1.0. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AAF. | Class(es): R1 | Usage: D = X02AAE (X) | On-line doc: CALL GAMSDOC X02AAE (or @PTRT NAG*DOC.X02AAE) | Access: LIB NBS*NAG

X02AAF  Smallest possible e such that 1.0+e > 1.0. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AAE. | Class(es): R1 | Usage: D = X02AAF (X) | On-line doc: CALL GAMSDOC X02AAF (or @PTRT NAG*DOC.X02AAF) | Access: LIB NBS*NAG

X02ABF  Smallest representable positive real number. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02ABF. | Class(es): R1 | Usage: D = X02ABF (X) | On-line doc: CALL GAMSDOC X02ABF (or @PTRT NAG*DOC.X02ABF) | Access: LIB NBS*NAG

X02ACF  Largest representable positive real number. | Proprietary single precision Fortran subprogram in NAG library. Single precision version is X02ACF. | Class(es): R1 | Usage: D = X02ACF (X) | On-line doc: CALL GAMSDOC X02ACF (or @PTRT NAG*DOC.X02ACF) | Access: LIB NBS*NAG

X02ADE  Ratio of X02ABF to X02AAE. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02ADF. | Class(es): R1 | Usage: D = X02ADE (X) | On-line doc: CALL GAMSDOC X02ADE (or @PTRT NAG*DOC.X02ADE) | Access: LIB NBS*NAG

X02ADF  Ratio of X02ABF to X02AAE. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02ADE. | Class(es): R1 | Usage: D = X02ADF (X) | On-line doc: CALL GAMSDOC X02ADF (or @PTRT NAG*DOC.X02ADF) | Access: LIB NBS*NAG

X02AEE  Largest negative permissible argument for exp. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AEE. | Class(es): R1 | Usage: D = X02AEE (X) | On-line doc: CALL GAMSDOC X02AEE (or @PTRT NAG*DOC.X02AEE) | Access: LIB NBS*NAG

X02AEF  Largest negative permissible argument for exp. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AEE. | Class(es): R1 | Usage: D = X02AEF (X) | On-line doc: CALL GAMSDOC X02AEF (or @PTRT NAG*DOC.X02AEF) | Access: LIB NBS*NAG

X02AFE  Returns the value of the largest positive argument permitted for EXP. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AFF. | Class(es): R1 | Usage: R = X02AFE(X) | On-line doc: CALL GAMSDOC X02AFE (or @PTRT NAG*DOC.X02AFE) | Access: LIB NBS*NAG

X02AFF  Returns the value of the largest positive argument permitted for DEXP. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AFE. | Class(es): R1 | Usage: D = X02AFF (X) | On-line doc: CALL GAMSDOC X02AFF (or @PTRT NAG*DOC.X02AFF) | Access: LIB NBS*NAG

X02AGE  Smallest representable positive real number with representable inverse. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AGF. | Class(es): R1 | Usage: l = X02AGE (X) | On-line doc: CALL GAMSDOC X02AGE (or @PTRT NAG*DOC.X02AGE) | Access: LIB NBS*NAG

X02AGF  Smallest representable positive real number with representable inverse. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AGF. | Class(es): R1 | Usage: l = X02AGF (X) | On-line doc: CALL GAMSDOC X02AGF (or @PTRT NAG*DOC.X02AGF) | Access: LIB NBS*NAG
X02AHE Returns the value of the largest positive argument permitted for SIN and COS. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02AHF. | Class(es): R1 | Usage: R = X02AHE(X) | On-line doc: CALL GAMSDOC X02AHE (or @PRT NAG+DOC.X02AHE) | Access: LIB NBS*NAG

X02AHF Returns the value of the largest positive argument permitted for DSIN and DCOS. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02AHE. | Class(es): R1 | Usage: D = X02AHF(X) | On-line doc: CALL GAMSDOC X02AHF (or @PRT NAG+DOC.X02AHF) | Access: LIB NBS*NAG

X02BAE Base of floating-point arithmetic. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BDF. | Class(es): R1 | Usage: 1 = X02BAE(X) | On-line doc: CALL GAMSDOC X02BAE (or @PRT NAG+DOC.X02BAE) | Access: LIB NBS*NAG

X02BDF Base of floating-point arithmetic. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BAE. | Class(es): R1 | Usage: 1 = X02BDF(X) | On-line doc: CALL GAMSDOC X02BDF (or @PRT NAG+DOC.X02BDF) | Access: LIB NBS*NAG

X02BBF Largest representable integer. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BBF. | Class(es): R1 | Usage: 1 = X02BBF(X) | On-line doc: CALL GAMSDOC X02BBF (or @PRT NAG+DOC.X02BBF) | Access: LIB NBS*NAG

X02BCF Largest representable integer. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BCF. | Class(es): R1 | Usage: 1 = X02BCF(X) | On-line doc: CALL GAMSDOC X02BCF (or @PRT NAG+DOC.X02BCF) | Access: LIB NBS*NAG

X02BCE Largest positive integer power to which 2.0 can be raised without overflow. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BCF. | Class(es): R1 | Usage: 1 = X02BCE(X) | On-line doc: CALL GAMSDOC X02BCE (or @PRT NAG+DOC.X02BCE) | Access: LIB NBS*NAG

X02BDF Largest negative integer power to which 2.0 can be raised without overflow. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BDF. | Class(es): R1 | Usage: 1 = X02BDF(X) | On-line doc: CALL GAMSDOC X02BDF (or @PRT NAG+DOC.X02BDF) | Access: LIB NBS*NAG

X02BEE Maximum number of decimal digits that can be represented. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02BEE. | Class(es): R1 | Usage: 1 = X02BEE(X) | On-line doc: CALL GAMSDOC X02BEE (or @PRT NAG+DOC.X02BEE) | Access: LIB NBS*NAG

X02BEF Maximum number of decimal digits that can be represented. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02BEF. | Class(es): R1 | Usage: 1 = X02BEF(X) | On-line doc: CALL GAMSDOC X02BEF (or @PRT NAG+DOC.X02BEF) | Access: LIB NBS*NAG

X02CAF Estimate of active-set size (on machines with paged virtual store). | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02CAF. | Class(es): R1 | Usage: 1 = X02CAF(X) | On-line doc: CALL GAMSDOC X02CAF (or @PRT NAG+DOC.X02CAF) | Access: LIB NBS*NAG

X02CAE Estimate of active-set size (on machines with paged virtual store). | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02CAE. | Class(es): R1 | Usage: 1 = X02CAE(X) | On-line doc: CALL GAMSDOC X02CAE (or @PRT NAG+DOC.X02CAE) | Access: LIB NBS*NAG

X02DAE Switch for taking precautions to avoid underflow. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X02DAF. | Class(es): R1 | Usage: L = X02DAE(X) | On-line doc: CALL GAMSDOC X02DAE (or @PRT NAG+DOC.X02DAE) | Access: LIB NBS*NAG

X02DAF Switch for taking precautions to avoid underflow. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X02DAF. | Class(es): R1 | Usage: L = X02DAF(X) | On-line doc: CALL GAMSDOC X02DAF (or @PRT NAG+DOC.X02DAF) | Access: LIB NBS*NAG

X03AAE Real innerproduct added to initial value, basic/additional precision. | Proprietary single precision Fortran subprogram in NAG library. Double precision version is X03AAE. | Class(es): D1a4 | Usage: CALL X03AAE(A, ISIZEA, B, ISIZEB, N, ISTEPA, ISTEPB, C1, C2, D1, D2, SW, IPAIL) | On-line doc: CALL GAMSDOC X03AAE (or @PRT NAG+DOC.X03AAE) | Access: LIB NBS*NAG

X03AAF Real innerproduct added to initial value, basic/additional precision. | Proprietary double precision Fortran subprogram in NAG library. Single precision version is X03AEE. | Class(es): D1a4 | Usage: CALL X03AAF(A, ISIZEA, B, ISIZEB, N, ISTEPA, ISTEPB,
C1, C2, D1, D2, SW, IFAIL | On-line doc: CALL GAMSDOC X03AAF (or @PRT NAG+DOC.X03AAF) | Access: LIB NBS+NAG

X03ABE Complex innerproduct added to initial value, basic/additional precision. | Proprietary single precision Fortran submodule in NAG library. Double precision version is X03ABF. | Class(es): D1a4 | Usage: CALL X03ABE (A, ISIZEA, B, ISIZEB, I, ISTEPA, ISTEPB, CX, DX, SW, IFAIL) | On-line doc: CALL GAMSDOC X03ABE (or @PRT NAG+DOC.X03ABE) | Access: LIB NBS+NAG

X03ABF Complex innerproduct added to initial value, basic/additional precision. | Proprietary double precision Fortran submodule in NAG library. Single precision version is X03ABE. | Class(es): D1a4 | Usage: CALL X03ABF (A, ISIZEA, B, ISIZEB, I, ISTEPA, ISTEPB, CX, DX, SW, IFAIL) | On-line doc: CALL GAMSDOC X03ABF (or @PRT NAG+DOC.X03ABF) | Access: LIB NBS+NAG

X04AAE Return or set unit number for error messages for NAG library programs. | Proprietary single precision Fortran submodule in NAG library. Double precision version is X04AAF. | Class(es): R3b | Usage: CALL X04AAE (IFLAG, NERR) | On-line doc: CALL GAMSDOC X04AAE (or @PRT NAG+DOC.X04AAE) | Access: LIB NBS+NAG

X04AAF Return or set unit number for error messages for NAG library programs. | Proprietary double precision Fortran submodule in NAG library. Single precision version is X04AAE. | Class(es): R3b | Usage: CALL X04AAF (IFLAG, NERR) | On-line doc: CALL GAMSDOC X04AAF (or @PRT NAG+DOC.X04AAF) | Access: LIB NBS+NAG

X04ABE Return or set unit number for advisory messages for NAG library programs. | Proprietary single precision Fortran submodule in NAG library. Double precision version is X04ABF. | Class(es): R3b | Usage: CALL X04ABE (IFLAG, NADV) | On-line doc: CALL GAMSDOC X04ABE (or @PRT NAG+DOC.X04ABE) | Access: LIB NBS+NAG

X04ABF Return or set unit number for advisory messages for NAG library programs. | Proprietary double precision Fortran submodule in NAG library. Single precision version is X04ABE. | Class(es): R3b | Usage: CALL X04ABF (IFLAG, NADV) | On-line doc: CALL GAMSDOC X04ABF (or @PRT NAG+DOC.X04ABF) | Access: LIB NBS+NAG

XDLEGF Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. | Portable double precision Fortran submodule in FCNPAK sublibrary of CMLIB library. Single precision version is XSLEGF. | Class(es): C9 C3a2 | Usage: CALL XDLEGF(DNU1,NUDIFF,MU1,MU2,THETA,DI,DQ,A,QPA) | On-line doc: CALL GAMSDOC XDLEGF (or @PRT CMLIB+DOC.XDLEGF/FCNPAK) | Tests: CMLIB*TEST-SOURCE.XDLEGF/FCNPAK | Access: LIB NBS+CMLIB | See also: XDSET

XDNRMP Calculates a sequence of normalized Legendre polynomials for fixed degree and argument and variable order. | Portable double precision Fortran submodule in FCNPAK sublibrary of CMLIB library. Single precision version is XSNRMP. | Class(es): C9 C3a2 | Usage: CALL XDNRMP(NU,MU,ARG,MODE,FN,IPN,ISIG) | On-line doc: CALL GAMSDOC XDNRMP (or @PRT CMLIB+DOC.XDNRMP/FCNPAK) | Tests: CMLIB*TEST-SOURCE.XDNRMP/FCNPAK | Access: LIB NBS+CMLIB | See also: XDSET

XERABT Terminate run and print traceback. (Requires system dependent programming to execute properly, else just STOPs.). | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERABT(MESSG,NMESSG) | On-line doc: CALL GAMSDOC XERABT (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XERCLR Clear current message number. | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERCLR | On-line doc: CALL GAMSDOC XERCLR (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XERCTL Perform special error processing of one message. | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERCTL(MESSG,NMESSG,NERR,LEVEL,KONTRL) | On-line doc: CALL GAMSDOC XERCTL (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XERDMP Print error summary and clear tables. | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERDMP | On-line doc: CALL GAMSDOC XERDMP (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XERMAX Set limit of MAX times each message can be printed. | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERMAX(MAX) | On-line doc: CALL GAMSDOC XERMAX (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB


XERRWV Process a message with numeric values. | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XERRWV(MESSG,NMESSG,NERR,LEVEL,N1,N2,NR,R1,R2) | On-line doc: CALL GAMSDOC XERRWV (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XGETF Get current value of KONTRL. | Portable single precision Fortran submodule in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XGETF(KONTRL) | On-line doc: CALL GAMSDOC XGETF (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB
XGETUA Get current output unit number. Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XGETUA(IUNITA,N) | On-line doc: CALL GAMSDOC XGETUA (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XGETUN Get current output unit number. Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3c | Usage: CALL XGETUN(IUNIT) | On-line doc: CALL GAMSDOC XGETUN (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XSETF Set KONTRL for XERROR, default is =2. Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3a | Usage: CALL XSETF(KONTRL) | On-line doc: CALL GAMSDOC XSETF (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XSETUA Set up to 5 output unit numbers. Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3b | Usage: CALL XSETUA(IUNITA,N) | On-line doc: CALL GAMSDOC XSETUA (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XSETUN Set one output unit number. Portable single precision Fortran subprogram in XERROR sublibrary of CMLIB library. | Class(es): R3b | Usage: CALL XSETUN(IUNIT) | On-line doc: CALL GAMSDOC XSETUN (or @PRT CMLIB+DOC.SUMMARY/XERROR) | Access: LIB NBS+CMLIB

XSLEGF Calculates sequences of values of Legendre functions of the first kind of positive or negative order, as well as Legendre functions of the second kind, and normalized Legendre functions. Portable single precision Fortran subprogram in FCNPAK sublibrary of CMLIB library. Double precision version is XDLEGF. | Class(es): C9 C3a2 | Usage: CALL XSLEGF(DNU1,NUDIFF,MUl,MU2,THETA,ID,PQA.IPQA) | On-line doc: CALL GAMSDOC XSLEGF (or @PRT CMLIB+DOC.XSLEGF/FCNPAK) | Tests: CMLIB+TEST-SOURCE.XSLEGF/FCNPAK | Access: LIB NBS+CMLIB | See also: XSSET

XSNRMP Calculates a sequence of values of the normalized Legendre polynomials for fixed degree and argument and variable order. Portable single precision Fortran subprogram in FCNPAK sublibrary of CMLIB library. Double precision version is XDNRMP. | Class(es): C9 C3a2 | Usage: CALL XSNRMP(NU,MUl,MU2,ARG,MODE,PN,P,FQ,IFN,IFSIG) | On-line doc: CALL GAMSDOC XSNRMP (or @PRT CMLIB+DOC.XSNRMP/FCNPAK) | Tests: CMLIB+TEST-SOURCE.XSNRMP/FCNPAK | Access: LIB NBS+CMLIB | See also: XSSET
Z

ZANLYT Zeros of an analytic complex function using the Muller method with deflation. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): Fl1a2 | Usage: CALL ZANLYT (F,EPS,NSIG,KN,NGUESS,N,X,ITMAX,INFER,IER) | On-line doc: CALL GAMSDOC ZANLYT (or @PRT IMSL*DOC.ZANLYT) | Access: LIB NBS*IMSL

ZBRENT Zero of a function which changes sign in a given interval (Brent algorithm). | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): Flb | Usage: CALL ZBRENT (F,EPS,NSIG,A,B,MAXFN,IER) | On-line doc: CALL GAMSDOC ZBRENT (or @PRT IMSL*DOC.ZBRENT) | Access: LIB NBS*IMSL


ZERO Finds a single real root of a function within an interval specified by the user. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DZERO. | Class(es): Flb | Usage: X = ZERO (F,A,B,T) | On-line doc: CALL GAMSDOC ZERO (or @PRT PORT*DOC.ZERO) | Access: LIB NBS*PORT

ZERONIN Finds a zero of a user defined function on an interval given the endpoints A and B such that F(A)*F(B) < 0. | Portable single precision Fortran subroutine in ZERONIN sublibrary of CMLIB library. | Class(es): Flb | Usage: CALL ZERONIN (F,B,A,C,NSIG,NSIGW,K,IJOB) | On-line doc: CALL GAMSDOC ZERONIN (or @PRT CMLIB*DOC.ZEROIN/2) | Tests: CMLIB*TEST-SOURCE.ZEROIN/ZEROIN | Access: LIB NBS*CMLIB

ZFALSE Zero of a function given an interval containing the zero. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): Fl1a2 | Usage: CALL ZFALSE (F,EPS,NSIG,XL,XR,XAPP,ITMAX,IER) | On-line doc: CALL GAMSDOC ZFALSE (or @PRT IMSL*DOC.ZFALSE) | Access: LIB NBS*IMSL

ZINTERVAL Calculates a z-confidence interval with specified percent confidence and standard deviation. | Command in MINITAB Proprietary interactive system. | Class(es): Fl1a14 Fl1a2 | Usage: ZINTERVAL (F,B,C,NSIG,NSIGW,K,IJOB) | On-line doc: CALL GAMSDOC ZINTERVAL (or @PRT PORT*DOC.ZINTERVAL) | Access: LIB NBS*PORT

ZONE Finds a solution of a system of non-linear equations. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DZONE. | Class(es): F2a | Usage: CALL ZONE (FNC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC ZONE (or @PRT PORT*DOC.ZONE) | Access: LIB NBS*PORT

ZONEJ Finds a solution of a system of non-linear equations. User must provide a SUBROUTINE to compute the Jacobian matrix. | Proprietary single precision Fortran subroutine in PORT library. Double precision version is DZONEJ. | Class(es): F2a | Usage: CALL ZONEJ (FNC, Z1JAC, N, X, EPS, JMAX, F2NORM) | On-line doc: CALL GAMSDOC ZONEJ (or @PRT PORT*DOC.ZONEJ) | Access: LIB NBS*PORT


ZREAL1 The real zeros of a real function - to be used when initial guesses are poor. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): Fl1a2 | Usage: CALL ZREAL1 (F,EPS,PS2,ETA,NSIG,N,X,ITMAX,IER) | On-line doc: CALL GAMSDOC ZREAL1 (or @PRT IMSL*DOC.ZREAL1) | Access: LIB NBS*IMSL

ZREAL2 The real zeros of a real function - to be used when initial guesses are good. | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): Fl1a2 | Usage: CALL ZREAL2 (F,EPS,PS2,ETA,NSIG,N,X,ITMAX,IER) | On-line doc: CALL GAMSDOC ZREAL2 (or @PRT IMSL*DOC.ZREAL2) | Access: LIB NBS*IMSL


ZSPW Solve a system of nonlinear equations (uses function values only). | Proprietary single precision Fortran subroutine in IMSL library. | Class(es): Fl2a | Usage: CALL ZSPW (A,NDEG,Z,IER) | On-line doc: CALL GAMSDOC ZSPW (or...
GAMS: Module Dictionary

January 1984

@PRT IMSL+DOC.ZSPW | Access: LIB NBS+IMSL


**ZTEST** Performs a one- or two-sided z-test for a specified standard deviation. | Command in MINITAB Proprietary interactive system. | Class(es): L4a1a1 L4a2 | Usage: ZTEST of \( \mu \) = 0 assuming sigma \( \) = 0, on C | [subcommand ALTERNATIVE = 1.] | On-line doc: HELP ZTEST (in Minitab) | Tests: MINITAB+TEST-SOURCE. | Access: @XQT NBS+MINITAB.MINITAB (or CALL MINITAB in CTS)

**ZX0LP** Solve the linear programming problem (phase one or phase two) via the revised simplex algorithm. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2a1 | Usage: CALL ZX0LP (IFPHASE,C,D,icolms,ROW,K,M,N,ITMAX,IC,IR,COPI,IDES, XWA,IER) | On-line doc: CALL GAMSDOC ZX0LP (or @PRT IMSL+DOC.ZX0LP) | Access: LIB NBS+IMSL

**ZX3LP** Solve the linear programming problem via the revised simplex algorithm easy to use version. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2a1 | Usage: CALL ZX3LP (A,IA,B,C,N,M2,S,PSOL,DSOL,RW,IW,IER) | On-line doc: CALL GAMSDOC ZX3LP (or @PRT IMSL+DOC.ZX3LP) | Access: LIB NBS+IMSL

**ZX4LP** Solve the linear programming problem via the revised simplex algorithm (alternate easy to use version). | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2a1 | Usage: CALL ZX4LP (A,IA,B,C,N,M1,M2,S,PSOL,DSOL,RW,IW,IER) | On-line doc: CALL GAMSDOC ZX4LP (or @PRT IMSL+DOC.ZX4LP) | Access: LIB NBS+IMSL


**ZXGSN** One-dimensional unimodal function minimization using the golden section search method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1a2 | Usage: CALL ZXGSN (F,A,B,TOL,XMIN,IER) | On-line doc: CALL GAMSDOC ZXGSN (or @PRT IMSL+DOC.ZXGSN) | Access: LIB NBS+IMSL

**ZXGSP** One-dimensional unimodal function minimization using the golden section search method - data parameters specified. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1a2 | Usage: CALL ZXGSP (F,P1,P2,P3,P4,P5,A,B,TOL,XMIN,IER) | On-line doc: CALL GAMSDOC ZXGSP (or @PRT IMSL+DOC.ZXGSP) | Access: LIB NBS+IMSL

**ZXLSP** One-dimensional minimization of a smooth function using safeguarded quadratic interpolation. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1a1a | Usage: CALL ZXLSP (FUNCT,X,STEP,BOUND,XACC,MAXFN,IER) | On-line doc: CALL GAMSDOC ZXLSP (or @PRT IMSL+DOC.ZXLSP) | Access: LIB NBS+IMSL

**ZXMIN** Minimum of a function of n variables using a quasi-Newton method. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G1b1a | Usage: CALL ZXMIN (FUNCT,N,NSIG,M1,M2,S,PSOL,DSOL,X,F,IWORK,IER) | On-line doc: CALL GAMSDOC ZXMIN (or @PRT IMSL+DOC.ZXMIN) | Access: LIB NBS+IMSL

**ZXMD** Global minimum (with constraints) of a function of n variables. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): G2b1a1 | Usage: CALL ZXMDW (FUNC,N,NSIG,A,B,NSRCH,X,F,WORK,W,IER) | On-line doc: CALL GAMSDOC ZXMDW (or @PRT IMSL+DOC.ZXMDW) | Access: LIB NBS+IMSL

**ZXSSQ** Minimum of the sum of squares of n functions in n variables using a finite difference Levenberg-Marquardt algorithm. | Proprietary single precision Fortran subprogram in IMSL library. | Class(es): K1b1a1 L8g1a L8g2a | Usage: CALL ZXSSQ (FUNC,M,N,NSIG,EP,D,MAXFN,IOPT,PARM,X,SSQ,F,X,AJC,IXAC,XTJT,W,IER) | On-line doc: CALL GAMSDOC ZXSSQ (or @PRT IMSL+DOC.ZXSSQ) | Access: LIB NBS+IMSL
LIBRARY REFERENCE

The “library” is the largest unit cataloged in GAMS. A library may be a collection of Fortran subprograms, a collection of stand-alone programs, or even a complex interactive system. In this section we present a summary description of each of the libraries currently listed in GAMS, along with information about how to use them at NBS. Libraries are listed in alphabetical order. The following explains the terminology used in the summaries.

LEGEND

Type  indicates how the software is organized. Subprogram library: a collection of subprograms. In order to use the library, one writes a main program, usually in Fortran, which calls the desired subprogram, and then links the necessary components of the pre-compiled library with the user's main program. Stand-alone program library: a collection of stand-alone programs. In order to use the library, one prepares an input file of commands and data, and then executes the desired program. Program library: a collection containing both subprograms and stand-alone programs. Partitioned library: a library which is made up of smaller collections called packages. A package is a collection of modules, e.g., programs or subprograms, for a narrow class of problems and is obtained from a single developer. Interactive program: a stand-alone program which is executed interactively. The word system may be used when the library corresponds to a single interactive program which performs a large variety of tasks.

Version  indicates the version name, number, or date of the software documented in GAMS. Note that different versions may be implemented on different computers at NBS.

Portability  indicates restrictions on library usage and the ease of transporting the software to other machines. Portable: in the public domain, written in a commonly available subset of the programming language (usually ANSI Standard Fortran), and free of dependencies on the arithmetic properties of a specific machine. The source code for this software is available and may easily be transported to a large number of computers. Portable, some conversion required: while most of the code is portable, certain localized parts of the computation use proprietary software or machine-specific constructs. These portions in general will have to be rewritten to implement this software on different systems. Proprietary: use of the software is governed by a licensing agreement. This license may restrict use to a specific computer or it may allow site-wide use at NBS. Contact a GAMS consultant for further information.

Reference  cites a hard-copy description of the library.

Developer  lists at least one of the institutions responsible for the development of the software, and the name of a contact there.

Support level  indicates the level of support provided for users of the software at NBS. Fully supported: at least one of the developers is on the NBS staff and is available to provide assistance. Supported: help in locating errors in the software is available and liaison with the developers is maintained. Mildly supported: some help in locating errors in the software is available.

Summary doc  indicates how to obtain an on-line summary of the capabilities of the library. For the Sperry 1100, this takes the form “CTS command (EXEC command)”.

Detailed doc  indicates how to obtain detailed on-line documentation for individual modules in the library. For the Sperry 1100, this takes the form “CTS command (EXEC command)”.

Access  indicates how to gain access to the library. For the Sperry 1100, this takes the form “CTS command (EXEC command)”.

January 1984  GAMS: Library Reference  D 0
BMDP

Type: Stand-alone program library

Approximately 40 programs for statistical data analysis – data description, line-printer plotting, regression, analysis of variance, frequency tables, time series analysis, multivariate analysis (cluster, correlation, discriminant, factor), and life tables.

Portability: Proprietary

NBS USAGE

On Sperry 1100/FTN ...

Support Level: Supported
Summary Doc: OLD BMDP+DOC.SUMMARY (or &PRT BMDP+DOC.SUMMARY)
Detailed Doc: Refer to BMDP reference manual
Access: PXQT NBS+PLIBs.BMDP BMDprogram (or aNBS+PLIBs.BMDP BMDprogram)

On Cyber 750 ...

Support Level: Supported
Summary Doc: GET(BMDPDOC/UN=CAMLIB)
Detailed Doc: Refer to BMDP reference manual
Access: GET(BMDPGET/UN=CAMLIB)

BMDPDOC.
BMDPGET(BMDprogram)
BMDprogram.

Note: The current version of BMDP implemented on the Cyber 750 is the 1982 CDC version.

CONTENTS

Data description P1D, P2D, P4D, P9D, PAM
Graphics P5D, P6D
Data manipulation P1S
Elementary inference P3D, P3S
Analysis of variance P7D, P1V-P4V, P8V
Regression P1R-P6R, P9R, PAR, PLR, P9M
Contingency tables P4F
Time series analysis P1T, P2T
Correlation analysis P8D, P6M
Discriminant analysis P7M
Factor analysis P4M, P8M
Cluster analysis P1M-P3M, PKM, Q3M*
Life testing and survival analysis P1L, P2L

*Available only in CDC version.
CMLIB

Type: Partitioned Fortran subprogram library

Version: 1983

A collection of high-quality, easily transportable Fortran subroutine packages solving standard problems in many areas of mathematics and statistics.

Portability: Portable
Reference: NBS Core Math Library, vols. 1-4 (Gaithersburg: QA297.c69 in NBS Library, Admin E-120; Boulder: available in Radio Bldg., Rm. 4058)
Developer: Refer to package listings below

NBS USAGE

On Sperry 1100/FTN ...

Support Level: Supported
Summary Doc: CALL GAMSDOC/CMLIB (or @PRT CMLIB*DOC.SUMMARY)
Detailed Doc: CALL GAMSDOC/package (or @PRT CMLIB*DOC.SUMMARY/package)
Access: ASSUME LIBRARY NBS*CMLIB (or LIB NBS*CMLIB)

Source Availability: Source for each subprogram in a given package is available in the file element CMLIB*SOURCE.subprogram/package

Several CTS command procedures (subroutines) are available for the automatic retrieval of CMLIB source. Procedures with names containing GETMODS retrieve individual subprograms while those with names containing GETPKG retrieve whole packages. Procedures whose names end in X retrieve the requested source plus all external references in CMLIB.

CALL CMLIB+TOOLS.GETMODS list
CALL CMLIB+TOOLS.GETMODSX list
CALL CMLIB+TOOLS.GETPKG package
CALL CMLIB+TOOLS.GETPKGX package

Here list is a list of subprogram names separated by commas and package is the name of a CMLIB package. Output is left in the CTS work area.

On Cyber 750/FTN4 ...

Support Level: Supported
Summary Doc: GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc: See reference above
Access: ATTACH(CMLIB4/UN=CAMLIB,NA)
LIBRARY(CMLIB4)

Source Availability: Source for a given subprogram may be obtained using the commands
GET(CMSOURC/UN=CAMLIB)
CMSOURC(subprogram)

On Cyber 750/FTN5 ...

Support Level: Supported
Summary Doc: GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc: See reference above
Access : ATTACH(CMLIB5/UN=CMCMLIB,NA) 
LIBRARY(CMLIBS)

Source Availability : See description of Cyber 750/FTN4 above.

On Perkin-Elmer 32 bit systems ...

Support Level : Supported 
Summary Doc : See reference above 
Detailed Doc : See reference above 
Access : CMLIB.OBJ

Note: CMLIB is not automatically installed on all PE systems. However, it is available from the Scientific Computing Division upon request. (Contact Richard Freemire at FTS 921-2562.)

CONTENTS

AMOSLIB A collection of special function routines with particular emphasis on the special functions of statistics. 

BLAS Basic linear algebra subroutines. Perform various elementary matrix and vector operations. 

BSPLINE Subroutines for computing with piecewise polynomials (B-splines). Includes interpolation, differentiation and integration with B-splines. 

BVSUP Solves systems of linear two-point boundary value problems. 

CDRIV Solves initial value problems for systems of ordinary differential equations including stiff equations. 
Complex version of SDRIV—solves complex systems with real independent variable. 

CPQR79 Finds all zeros of real and complex polynomials via eigenvalue methods. 

CPZERO Computes all the zeros of real or complex coefficient polynomials. Error bounds are also computed. 

DBSPLINE Subroutines for computing with piecewise polynomials (B-splines). Double precision version of BSPLINE package. 

DDASSL Solves the system of differential/algebraic equations of the form \( g(t, y, y') = 0 \), with given initial values. Double precision version of SDASSL. 

DDRIV Solves initial value problems for systems of ordinary differential equations, including stiff systems. 
(Double precision version of SDRIV). 

DEPAC Solves systems of first order ordinary differential equations with arbitrary initial data. 

DNL2SN Solves nonlinear least squares problems and general optimization problems. (Double precision version of NL2SN). 


LINDRIVES Programs to solve various types of linear systems of algebraic equations. Provides an easy to use interface to the LINPACK package. | Version: 1980 | Reference: None | Developer: Los Alamos Scientific Laboratory, Los Alamos NM 87545


PCHIP Produces aesthetic looking interpolants to univariate data by using piecewise cubic Hermite functions. | Version: 1982 | Reference: F. Fritsch, Lawrence Livermore Laboratory Report UCRL-85104 | Developer: Lawrence Livermore Laboratory, Livermore, CA 94550 (F. Fritsch)


SNLS1E A suite of codes for nonlinear least squares problems (and computation of associated covariance matrices) and systems of nonlinear equations. | Version: 1983 | Reference: None | Developer: Sandia National Laboratory, Albuquerque, NM 87185 (K. Hiebert)


SSORT Fast in-core sorting of arrays. | Version: 1979 | Reference: None | Developer: Sandia Laboratories,
Albuquerque, NM 87185 (R. E. Jones), National Bureau of Standards, Washington, DC 20234 (D. Kahaner)


**XBLAS** Extended basic linear algebra subroutines. Perform various matrix and vector operations not found in the BLAS. | Version: 1980 | Reference: None | Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)


DATAPAC

Type: Fortran subprogram library
Version: 77.5

Approximately 170 subroutines for probability distribution, density, percent point, and sparsity function evaluation, random number generation, line-printer plotting, histograms, scatter diagrams, probability plots, data manipulation, general statistical analysis, time series analysis, polynomial regression, and ANOVA.

Portability: Portable
Reference: J. J. Filliben, User's Guide to Datapac (version 77.5)
Developer: National Bureau of Standards, Washington, DC 20234 (J. J. Filliben)

NBS USAGE

On Sperry 1100/FTN ...

Support Level: Fully supported
Summary Doc: CALL GAMSDOC /DATAPAC (or @PRT DATAPAC+DOC.SUMMARY)
Detailed Doc: CALL GAMSDOC subprogram (or @PRT DATAPAC+DOC.subprogram)
Access: ASSUME LIBRARY NBS+DATAPAC (or LIB NBS+DATAPAC)

Source Availability: Source for a given subprogram is available in the file element
DATAPAC+SOURCE.subprogram

On Cyber 750/FTN4 ...

Support Level: Fully supported
Summary Doc: GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc: See reference above (or refer to source code)
Access: ATTACH(DATAPAC/UN=CAMLIB,NA)
LIBRARY(DATAPAC)

Source Availability: Source for a given subprogram may be obtained using the commands
ATTACH(DPACMOD/UN=CAMLIB,NA)
MODIFY(P=DPACMOD,Z,LO=E)/DECK subprogram

CONTENTS

Individual statistics (20 subprograms)
Data manipulation (17 subprograms)
Probability plots (19 subprograms)
Printer (wide-carriage) plots (12 subprograms)
Terminal (narrow-width) plots (6 subprograms)
General analyses (10 subprograms)
Cumulative distribution function values (21 subprograms)
Probability density function values (7 subprograms)
Percent point function values (20 subprograms)
Sparsity function values (7 subprograms)
Random number generators (23 subprograms)
Polynomial regression (4 subprograms)
Time series analyses (5 subprograms)
Input/output (4 subprograms)
IMSL

Type: Fortran subprogram library

Version: Edition 9

Approximately 500 Fortran subprograms solving standard problems in many areas of mathematics and statistics.

Portability: Proprietary


Developer: IMSL Inc., 6th Floor NBC Bldg., 7500 Bellaire Blvd., Houston, TX 77036

NBS USAGE

On Sperry 1100/FTN...

Support Level: Supported
Summary Doc: See reference above
Detailed Doc: CALL GAMSDOC subprogram (or &PRT IMSL+DOC.subprogram)
Access: ASSUME LIBRARY NBS+CMLIB (or LIB NBS+IMSL)

On Cyber 750/FTN4...

Support Level: Supported (by NOAA/SEL)
Summary Doc: GET(LIBDOC/UN=LIBDOC)
LIBDOC(IMSL)
Detailed Doc: See reference above
Access: ATTACH(IMSL/UN=LIB,NA)
LIBRARY(IMSL)

On Cyber 750/FTN5...

Support Level: Supported (by NOAA/SEL)
Summary Doc: GET(LIBDOC5/UN=LIBDOC5)
LIBDOC5(IMSL)
Detailed Doc: See reference above
Access: ATTACH(IMSL/UN=LIB5,NA)
LIBRARY(IMSL)

CONTENTS

Chapter A: Analysis of Variance (17 subprograms)
Chapter B: Basic Statistics (22 subprograms)
Chapter C: Categorized Data Analysis (5 subprograms)
Chapter D: Differential Equations, Quadrature, Differentiation (14 subprograms)
Chapter E: Eigensystem Analysis (11 subprograms)
Chapter F: Forecasting, Econometrics, Time series, Transforms (23 subprograms)
Chapter G: Generation and Testing of Random Numbers (38 subprograms)
Chapter I: Interpolation, Approximation, Smoothing (16 subprograms)
Chapter L: Linear Algebraic Equations (42 subprograms)
Chapter M: Mathematical and Statistical Special Functions (67 subprograms)
Chapter N: Non-parametric Statistics (25 subprograms)
Chapter O: Observation Structure, Multivariate Statistics (18 subprograms)
Chapter R: Regression Analysis (29 subprograms)
Chapter S: Sampling (8 subprograms)
Chapter U: Utility Functions (28 subprograms)
Chapter V: Vector-Matrix Arithmetic (72 subprograms)
Chapter Z: Zeros and Extrema, Linear Programming (23 subprograms)

NOTES
(a) The IMSL library is available on a large number of computer systems, and the precision of each routine may vary from implementation to implementation. For example, some routines may be only available in single precision on 60-bit machines while only in double precision on 32-bit machines. Other routines may not be available on all machines. Users should consult the IMSL Library Reference Manual to determine the actual precision of each routine in each implementation.
INVAR

Type: Stand-alone interactive program library

Interactive programs for solving linear and/or nonlinear least squares problems using a variable separable algorithm. Adapted from the program VARPRO. Features statistical analysis of results and DISSPLA* graphics.

Portability: Portable, some conversion required*
Developer: National Bureau of Standards, Washington, DC 20234 (B. Rust)

NBS USAGE

On Sperry 1100 ...

Support Level: Fully supported
Summary Doc: CALL GAMSDOC /INVAR (or oPRT INVAR+DOC.SUMMARY)
Detailed Doc: See summary documentation.
Access: CALL SCD+CTSLIB.program (in CTS only)

CONTENTS

INVAR1 version with line-printer graphics only.
INVAR2 version with DISSPLA* graphics.

NOTES

*Graphics are implemented via calls to the DISSPLA proprietary graphics subprogram library (ISSCO, 4186 Sorrento Valley Blvd., San Diego, CA 92121). The remainder of the program is in the public domain.
MATHWARE

Type: Partitioned program library

A collection of packages, subroutines and other processors useful in solving a variety of scientific computing problems.

Portability: Portable, some conversion required
Reference: See package listings below
Developer: See package listings below

NBS USAGE

On Sperry 1100 ...

Support Level: Mildly supported
Summary Doc: OLD MATHWARE+DOC.SUMMARY (or aPRT MATHWARE+DOC.SUMMARY)
Detailed Doc: Refer to file MATHWARE+package.
Access: See individual sublibrary documentation

CONTENTS


MATLAB

Type: Interactive system

A system for matrix calculations, including solving linear systems, linear least squares problems, eigenvalue and eigenvector calculations, QR decomposition, singular value decomposition, and inverses. Based on LINPACK and EISPACK software.

Portability: Portable, some conversion required
Developer: University of New Mexico, Albuquerque, NM (C. Moler)

NBS USAGE

On Sperry 1100 ...

Support Level: Supported
Summary Doc: OLD MATLAB+DOC.SUMMARY (or @PRT MATLAB+DOC.SUMMARY)
Detailed Doc: HELP command (in MATLAB)
Access: XQT NBS+MATLAB.MATLAB (or @XQT NBS+MATLAB.MATLAB)

On Perkin-Elmer 32 bit systems ...

Support Level: Supported
Summary Doc: HELP (in MATLAB)
Detailed Doc: HELP command (in MATLAB)
Access: MATLAB

Note: MATLAB is not automatically installed on all PE systems. However, it is available from the Scientific Computing Division upon request. (Contact Richard Freemire at FTS 921-2562.)
MINITAB

Type: Interactive system

Version: 1982

Approximately 150 commands for data manipulation, plotting, random number generation, general purpose statistical analysis including regression, time series, EDA (exploratory data analysis), ANOVA, and analysis of tables.

Portability: Proprietary
Developer: Minitab Project, 215 Pond Laboratory, University Park, PA 16802 (T. Ryan)

NBS USAGE

On Sperry 1100 ...

Support Level: Supported
Summary Doc: OLD MINITAB*DOC.SUMMARY (or aPRT MINITAB*DOC.SUMMARY)
Detailed Doc: HELP command (in Minitab)
Access: CALL MINITAB (or aXQT NBS*MINITAB.MINITAB)

On Perkin-Elmer 32 bit systems ...

Support Level: Supported
Summary Doc: HELP (in Minitab)
Detailed Doc: HELP command (in Minitab)
Access: MINITAB

Note: Minitab is not automatically installed on all PE systems. However, it is available from the Scientific Computing Division upon request. (Contact Richard Freemire at FTS 921-2562.)

CONTENTS

Column operations and row operations (21 commands)
Editing (4 commands)
Manipulation (5 commands)
Sorting (3 commands)
Arithmetic (8 commands)
Functions (16 commands)
Histograms and scatter diagrams (6 commands)
Probability functions (3 commands)
Statistics on one column of data (7 commands)
Random number generation (8 commands)
Correlation analysis (1 command)
Analysis of variance (7 commands)
Regression (2 commands)
Tables (3 commands)
Time series analyses (6 commands)
EDA (exploratory data analysis) (9 commands)
Input/output (14 commands)
Macros and loops
Help
NAG

Type: Fortran subprogram library
Version: Mark 10

Approximately 500 Fortran subroutines for solving standard problems in many areas of mathematics, statistics and optimization.

Portability: Proprietary

NBS USAGE

On Sperry 1100/FTN ...
Support Level: Supported
Summary Doc: CALL GAMSDOC /NAG (or @PRT NAG+DOC.SUMMARY)
Detailed Doc: CALL GAMSDOC subprogram (or @PRT NAG+DOC.subprogram)
Access: ASSUME LIBRARY NBS*NAG (or LIB NBS*NAG)

Note: in the Sperry implementation all NAG subprograms are available in both single and double precision versions. Single precision routines have names which end in the letter E, while double precision routines have names which end in the letter F.

On Cyber 750/FTN4 ...
Support Level: Supported
Summary Doc: GET(NAGSUM/UN=CAMLIB)
 COPY(NAGSUM)
Detailed Doc: See reference above
Access: ATTACH(NAGLIB4/UN=CAMLIB,NA)
 LIBRARY(NAGLIB4)

Note: Only routines ending whose names end in F are available in the CDC implementation of the NAG library. These programs are single precision.

On Cyber 750/FTN5 ...
Support Level: Supported
Summary Doc: GET(NAGSUM/UN=CAMLIB)
 COPY(NAGSUM)
Detailed Doc: See reference above
Access: ATTACH(NAGLIB5/UN=CAMLIB,NA)
 LIBRARY(NAGLIB5)

Note: Only routines ending whose names end in F are available in the CDC implementation of the NAG library. These programs are single precision.

CONTENTS

Chapter A02 : Complex Arithmetic (3 subprograms)
Chapter C02 : Zeros of Polynomials (2 subprograms)
Chapter C05 : Roots of One or More Transcendental Equations (11 subprograms)
Chapter C06 : Summation of Series  (12 subprograms)
Chapter D01 : Quadrature  (25 subprograms)
Chapter D02 : Ordinary Differential Equations  (34 subprograms)
Chapter D03 : Partial Differential Equations  (9 subprograms)
Chapter D04 : Numerical Differentiation  (1 subprograms)
Chapter D05 : Integral Equations  (2 subprograms)
Chapter E01 : Interpolation  (7 subprograms)
Chapter E02 : Curve and Surface Fitting  (22 subprograms)
Chapter E04 : Maximizing or Minimizing a Function  (35 subprograms)
Chapter F01 : Matrix Operations, including Inversion  (60 subprograms)
Chapter F02 : Eigenvalue and Eigenvectors  (32 subprograms)
Chapter F03 : Determinants  (9 subprograms)
Chapter F04 : Simultaneous Linear Equations  (26 subprograms)
Chapter F05 : Orthogonalization  (2 subprograms)
Chapter G01 : Simple Calculations on Statistical Data  (17 subprograms)
Chapter G02 : Correlation and Regression Analysis  (26 subprograms)
Chapter G04 : Analysis of Variance  (4 subprograms)
Chapter G05 : Random Number Generators  (32 subprograms)
Chapter G08 : Nonparametric Statistics  (9 subprograms)
Chapter G13 : Time Series Analysis  (18 subprograms)
Chapter H : Operations Research  (7 subprograms)
Chapter M01 : Sorting  (20 subprograms)
Chapter P01 : Error Trapping  (1 subprograms)
Chapter S : Approximations of Special Functions  (41 subprograms)
Chapter X01 : Mathematical Constants  (2 subprograms)
Chapter X02 : Machine Constants  (15 subprograms)
Chapter X03 : Innerproducts  (2 subprograms)
Chapter X04 : Input/Output Utilities  (2 subprograms)

NOTES
(a) THE NAG library is available for a wide variety of machines/compilers. "Standard precision" routines in each implementation end in the letter F (for Fortran). On some machines this may represent a single precision routine while on others it may represent a double precision routine. In some implementations both precisions are available. In this case the alternate precision routines end in E if they are single precision or D if they are double precision. Users are urged to consult the NAG Fortran Library Manual to determine which routines are available in the version implemented for their machine.
PDELIB

Type: Fortran subprogram library

An informal collection of portable, public-domain Fortran subprograms which solve general systems of nonlinear initial-boundary-value partial differential equations in one or two space dimensions. Each program is based upon the method of lines.

Portability: Portable
Reference: See subprogram listing below.
Developer: See subprogram listing below.

NBS USAGE

On Sperry 1100/FTN...

Support Level: Supported
Summary Doc: CALL GAMSDOC/PDELIB (or @PRT PDELIB*DOC.SUMMARY)
Detailed Doc: CALL GAMSDOC subprogram (or @PRT PDELIB*DOC.subprogram)
Access: ASSUME LIBRARY NBS*PDELIB (or LIB NBS*PDELIB)

CONTENTS

MOL1D Solves systems of linear or nonlinear initial-boundary-value problems in one space dimension. Can solve hyperbolic equations with or without discontinuities, parabolic equations (including reaction-diffusion equations). Uses the method of lines based on equi-spaced finite differences. Graphical output available. | Author: J.M. Hyman, Los Alamos National Lab, Los Alamos, NM 87545

PDECOL Solves general nonlinear systems of initial-boundary-value problems in one space dimension with general boundary conditions. Spatial derivatives may be of at most second order. Uses method of lines based on collocation of B-spline basis functions. | Reference: ACM Transactions on Mathematical Software, vol. 5 (1979), pp. 326-351 | Authors: N. Madsen, Lawrence Livermore National Lab, Livermore, CA 94550 and R. Sincovec, Univ. of Colorado at Colorado Springs, Colorado Springs, CO 80907

PLOD

Type: Stand-alone interactive program

Version: 1983

Solves systems of ordinary differential equations. User may change various conditions, parameters, intervals, etc., interactively and plot results on an Hewlett Packard or Tektronix terminal. Very easy to use. Almost no programming experience required.

Portability: Portable, some conversion required*
Reference: None
Developer: National Bureau of Standards, Washington, DC 20234 (D. Kahaner)

NBS USAGE

On Sperry 1100/FTN ...

Support Level: Fully supported
Summary Doc: OLD PLOD*DOC.SUMMARY (or aPRT PLOD*DOC.SUMMARY)
Detailed Doc: See summary documentation
Access: XQT NBS*PLOD.PLOD (or aXQT NBS*PLOD.PLOD)

NOTES

*Graphics are implemented via calls to the DISSPLA proprietary graphics subprogram library (ISSCO, 4186 Sorrento Valley Blvd., San Diego, CA 92121). The remainder of the program is in the public domain.
PORT

Type: Fortran subprogram library

Version: 2

Approximately 200 Fortran subroutines for problems in areas such as curve fitting, ordinary differential equations, integration, etc.

Portability: Proprietary
Reference: The PORT Mathematical Subroutine Library, Manual (QA297.P2 in NBS Library, Admin E-120, Gaithersburg)
Developer: Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974 (P. Fox)

NBS USAGE

On Sperry 1100/FTN ...

Support Level: Supported
Summary Doc: CALL GAMSDOC /PORT
Detailed Doc: CALL GAMSDOC subprogram (or PRT PORT*DOC.subprogram)
Access: ASSUME LIBRARY NBS*PORT (or LIB NBS*PORT)

CONTENTS

Approximation (35 subprograms)
Arithmetic (8 subprograms)
Differential Equations (4 subprograms)
Linear Algebra (6 subprograms)
Optimization (1 subprogram)
Probability-Statistics (3 subprograms)
Quadrature and Differentiation (9 subprograms)
Roots (5 subprograms)
Special Functions (18 subprograms)
Transforms (6 subprograms)
Utility (88 subprograms)
SLDGL

Type : Fortran subprogram library

Portran subprograms for the numerical solution of initial value problems and boundary value problems for ordinary differential equations, and for elliptic and parabolic partial differential equations.

Portability : Proprietary
Developer : Universitat Karlsruhe, D-7500 Karlsruhe, W. Germany (W. Schonauer)

NBS USAGE

On Sperry 1100/FTN ...

Support Level : Supported
Summary Doc : CALL GAMSDOC /SLDGL (or $PRT SLDGL*DOC.SUMMARY)
Detailed Doc : CALL GAMSDOC subprogram (or $PRT SLDGL*DOC.subprogram)
Access : ASSUME LIBRARY NBS*SLDGL (or LIB NBS*SLDGL)

CONTENTS

Ordinary initial value problems
Ordinary boundary value problems
Two-dimensional elliptic boundary value problems on rectangular domains
Two-dimensional elliptic boundary value problems on non-rectangular domains
Three-dimensional elliptic boundary value problems on rectangular boxes
Initial boundary value problems for parabolic equations in one space dimension
Initial boundary value problems for parabolic equations in two space dimensions
Initial boundary value problems for parabolic equations in three space dimensions

NOTES

(a) In each case n nonlinear equations in n unknown functions are admitted.

(b) When multiple subprograms are available users must choose between types of initial conditions, grids, or numerical methods. Modules which adaptively choose the spatial mesh and the order of accuracy of the spatial discretization scheme are available.

(c) All subprograms are based upon finite difference methods. Time integration is done using a variable-order variable-step scheme. Solution of discrete elliptic equations is done by either direct (band elimination) or iterative (relaxation) methods.
SPECTRLAN

Type: Stand-alone interactive program

An interactive spectral analysis program for time series data. Produces periodograms, cumulative periodograms, continuous Fourier power spectra, cumulative power spectra, Fourier amplitude spectra, maximum entropy spectra, and integrated maximum entropy spectra. With DISSPLA* graphics.

<table>
<thead>
<tr>
<th>Portability</th>
<th>Portable, some conversion required*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer</td>
<td>National Bureau of Standards, Washington, DC 20234 (B. Rust)</td>
</tr>
</tbody>
</table>

NBS USAGE

On Sperry 1100/FTN ...

<table>
<thead>
<tr>
<th>Support Level</th>
<th>Fully supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary Doc</td>
<td>CALL GAMSDOC /SPECTRLAN (or PRT SPECTRLAN+DOC.SUMMARY)</td>
</tr>
<tr>
<td>Detailed Doc</td>
<td>See summary documentation</td>
</tr>
<tr>
<td>Access</td>
<td>CALL SCD+CTSLIB.SPECAN (in CTS only)</td>
</tr>
</tbody>
</table>

NOTES

*Graphics are implemented via calls to the DISSPLA proprietary graphics subprogram library (ISSCO, 4186 Sorrento Valley Blvd., San Diego, CA 92121). The remainder of the program is in the public domain.
STATLIB

Type: Fortran subprogram library

Version: 1978

Fifty-four Fortran subroutines for statistical data analysis. The easy-to-use subroutines, which automatically provide comprehensive printing and plotting, do elementary plotting, analyze univariate samples, do linear and nonlinear regression analysis, and time series analysis.

Portability: Portable

NBS USAGE

On Sperry 1100/FTN ...

Support Level: Fully supported
Summary Doc: CALL GAMSDOC /STATLIB (or ▲PRT STATLIB^DOC.SUMMARY)
Detailed Doc: CALL GAMSDOC subprogram (or ▲PRT STATLIB^DOC.subprogram)
Access: ASSUME LIBRARY NBS*STATLIB (or LIB NBS*STATLIB)

On Cyber 750/FTN4 ...

Support Level: Fully Supported
Summary Doc: GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc: See reference above
Access: ATTACH(STATLIB/UN=CAMLIB,NA)
LIBRARY(STATLIB)

On Cyber 750/FTN5 ...

Support Level: Fully Supported
Summary Doc: GET(CAMGUID/UN=CAMLIB)
CAMGUID.
Detailed Doc: See reference above
Access: ATTACH(SL2FTN5/UN=CAMLIB,NA)
LIBRARY(SL2FTN5)

CONTENTS

Plotting (histograms, Y vs. X, symbol, multiple, time series) (21 subprograms)
Analysis of a single random sample (3 subprograms)
Correlation analysis (2 subprograms)
Random number generation (1 subprogram)
One-way analysis of variance (2 subprograms)
Linear least squares analysis (8 subprograms)
Nonlinear least squares analysis (4 subprograms)
Time series analyses (16 subprograms)
A

Adams' predictor-corrector method (see ORDINARY DIFFERENTIAL EQUATIONS)
Adaptive quadrature (see QUADRATURE)
Airy function (see SPECIAL FUNCTIONS)
Aitken's method (see INTERPOLATION)
Algebraic-differential systems (see ORDINARY DIFFERENTIAL EQUATIONS)
Analysis of variance and analysis of covariance (see EXPERIMENTAL DESIGN)
Anger functions (see SPECIAL FUNCTIONS)

APPROXIMATION K

Chebyshev K2
Constrained linear least squares K1a2
Constrained nonlinear least squares K1b2
Curve fitting K1a1
Evaluation of fitted functions K0
L1 K3
L2 K1a
L∞ K2
Least absolute value (L1) K3
Levenberg-Marquardt method K1b1a
Linear least square (L2) K1a
Mesh generation K6
Minimax (Chebyshev or L∞) K2
Multivariate K1a1b
Nonlinear least squares K1b
Pade K4
Piecewise polynomials K1a1a1
Polynomial splines K1a1a1
Polynomials K1a1a2
Rational functions K1a1a3
Service routines (e.g., mesh generation, evaluation of fitted functions) K0
Smoothing K5
Splines K1a1a1, K1a1b, K5, K6
Surface fitting K1a1b
Taylor polynomial K4
Trigonometric functions K1a1a3
Unconstrained linear least squares K1a1
Unconstrained nonlinear least squares K1b1

also see REGRESSION

ARITHMETIC A

Base conversion A8b
Change of representation A6
Complex (single, double, and extended precision, extended range) A4
Construction of machine-base numbers A8c
Convergence acceleration A7
Decomposition of machine-base numbers A8c
Integer A1
Interval A5
Rational A2
Real (single, double, and extended precision, extended range) A3
Sequences A7
Type conversion A6a

also see MATRICES, VECTORS

ARMA and ARIMA modeling and forecasting (see TIME SERIES)
Assignments problem (see OPTIMIZATION)
Autocorrelation analysis (see TIME SERIES)

B

Balanced designs (see EXPERIMENTAL DESIGN)
Barrier method (see OPTIMIZATION)
Base conversion (see ARITHMETIC)
Bessel functions and their integrals (see SPECIAL FUNCTIONS)
Beta and related functions (see SPECIAL FUNCTIONS)
Beta distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Biased regression (see REGRESSION)
Biharmonic equation (see PARTIAL DIFFERENTIAL EQUATIONS)
Binary search (see OPTIMIZATION)
Binomial coefficient (see SPECIAL FUNCTIONS)
Binomial distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Bisection method (see NONLINEAR EQUATIONS, OPTIMIZATION)
Bit manipulation (see DATA HANDLING)
Boolean distribution (see RANDOM NUMBERS)
Boundary value problems (see ORDINARY DIFFERENTIAL EQUATIONS)
Box plots (see GRAPHICS)
Box-Jenkins (see TIME SERIES)
Bulirsch-Stoer method (see ORDINARY DIFFERENTIAL EQUATIONS)

C

Calibration (inverse regression) (see REGRESSION)

CATEGORICAL DATA L9

2-by-2 tables L9a
EDA (e.g., median polish) L9d
Log-linear model L9c
Two-way tables L9b

Cauchy distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Ceiling (see SPECIAL FUNCTIONS)
Character manipulation (see DATA HANDLING)
Chebyshev approximation (see APPROXIMATION, REGRESSION)
Chebyshev polynomials (see SPECIAL FUNCTIONS)
Chebyshev series (see SPECIAL FUNCTIONS)
Chi-squared distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Cholesky decomposition (see MATRICES)
Clenshaw-Curtis method (see QUADRATURE)
Cluster analysis (see STATISTICS)
Collocation method (see ORDINARY DIFFERENTIAL EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Complete designs (see EXPERIMENTAL DESIGN)
Complex arithmetic (see ARITHMETIC)
Complex demodulation (see TIME SERIES)
COMPUTATIONAL GEOMETRY (e.g., closest- and farthest-point, line intersection, convex hull, minimum spanning tree, triangulation, Voronoi diagram, polygon intersection, hidden line) P

Confidence intervals (see STATISTICS)
Confluent hypergeometric functions (see SPECIAL FUNCTIONS)
Conjugate gradient method (see OPTIMIZATION)
Constrained problems (see APPROXIMATION, OPTIMIZATION)
Contingency tables (see CATEGORICAL DATA, RANDOM NUMBERS)
Contrasts (see EXPERIMENTAL DESIGN)
Convergence acceleration (see ARITHMETIC)
Conversion (base and type) (see ARITHMETIC)
Convolutions (see INTEGRAL TRANSFORMS, VECTORS)
Copy (see MATRICES, VECTORS)
Correlation analysis (see STATISTICS)
Correlation coefficient (see STATISTICS)
Correlation matrix (see RANDOM NUMBERS)
Cosine integrals (see SPECIAL FUNCTIONS)
Coulomb wave functions (see SPECIAL FUNCTIONS)
Covariance, analysis of (see EXPERIMENTAL DESIGN)
Covering problems (see OPTIMIZATION)
Cross-correlation analysis (see TIME SERIES)
Curve fitting (see APPROXIMATION, INTERPOLATION, REGRESSION)
Cyclic reduction method (see PARTIAL DIFFERENTIAL EQUATIONS)

DATA HANDLING N

Bit manipulation N2
Character manipulation N3
Heaps N4
Input and output N1
Merging N7
Permuting N8
Ranking N9a1
Searching (for extreme value, insertion position, on a key) N5
Sorting (internal (passive and active; integer, real, and character) and external) N6
Stacks N4
Storage management (e.g., stacks, heaps, trees) N4
Trees N4

Data manipulation (see STATISTICS)
Data summarization (see STATISTICS)
Dawson's integral (see SPECIAL FUNCTIONS)
Decomposition (see MATRICES)
Density functions (see PROBABILITY FUNCTIONS, STATISTICS)
Determinants (see MATRICES)
Diagonalization (see MATRICES)
Differential equations (see ORDINARY DIFFERENTIAL EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)

DIFFERENTIATION H1

Splines H1
Digital filtering (see TIME SERIES)
Discriminant analysis (see STATISTICS)
Dispersion (see STATISTICS)
Distribution functions (see GRAPHICS, PROBABILITY FUNCTIONS, STATISTICS)
Distribution-free analysis (see EXPERIMENTAL DESIGN, STATISTICS)
Documentation retrieval (see SERVICE ROUTINES)
Domain triangulation (see PARTIAL DIFFERENTIAL EQUATIONS)
Dot product (see VECTORS)
Double exponential distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Double precision arithmetic (see ARITHMETIC)
Dynamic programming (see OPTIMIZATION)

EDA (exploratory data analysis) (see CATEGORICAL DATA, GRAPHICS, REGRESSION)
Eigenvalue problems (see MATRICES, PARTIAL DIFFERENTIAL EQUATIONS)
Eigenvalues and eigenvectors (see MATRICES)
Elliptic boundary value problems (see PARTIAL DIFFERENTIAL EQUATIONS)
Elliptic equations (see PARTIAL DIFFERENTIAL EQUATIONS)
Elliptic integrals (see SPECIAL FUNCTIONS)
Error analysis (see ARITHMETIC)
Error checking (e.g., check monotonicity) (see SERVICE ROUTINES)
Error functions, their inverses, and their integrals (see PROBABILITY FUNCTIONS, SPECIAL FUNCTIONS)
Error handling (see SERVICE ROUTINES)
Errors in variables (see REGRESSION)
Euclidean ($L_2$) norm (see MATRICES, VECTORS)
Evaluation of fitted functions (see APPROXIMATION, INTERPOLATION)
Exchange (see VECTORS)

EXPERIMENTAL DESIGN L7

Analysis of covariance (ANCOVA) L7a3
Analysis of variance (ANOVA) L7a1, L7a2
Balanced complete designs L7a2a
Balanced incomplete designs L7a2b
Contrasts L7a1a1
Distribution-free analysis L7a1b, L7a2a2, L7a2b2, L7a4b
Factorial designs L7a2a1b
General linear model (unbalanced design) L7a4
Latin square L7a2b1a
Lattice designs L7a2b1b
Multiple comparisons L7a1a1
Multivariate analysis of variance (MANOVA) L7b
Nested designs L7a2a1c
Nonparametric analysis L7a1b, L7a2a2, L7a2b2, L7a4b
One-way analysis of variance (ANOVA) L7a1
Repeated measures L7
Student-Newman-Keuls test L7a1a1
Two-way analysis of variance L7a2a1a
Unbalanced designs L7a4
Variance components L7a1a2

Exploratory data analysis (see CATEGORICAL DATA, GRAPHICS, REGRESSION)
Exponential distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Exponential functions and integrals (see SPECIAL FUNCTIONS)
Extended precision and extended range arithmetic (see ARITHMETIC)
Extrapolation methods (see ORDINARY DIFFERENTIAL EQUATIONS)
Extreme value distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

F

F-distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Factor analysis (see STATISTICS)
Factorial (see SPECIAL FUNCTIONS)
Factorial designs (see EXPERIMENTAL DESIGN)
Fast Fourier transforms (see INTEGRAL TRANSFORMS, TIME SERIES)
Filtering (see TIME SERIES)
Fletcher and Reeves' method (see OPTIMIZATION)
Floor (see SPECIAL FUNCTIONS)
Forecasting (see TIME SERIES)
Fourier transform (see INTEGRAL TRANSFORMS)
Fredholm integral equations (see INTEGRAL EQUATIONS)
Fresnel integrals (see SPECIAL FUNCTIONS)
Function evaluation (see PROBABILITY FUNCTIONS, SPECIAL FUNCTIONS)

G

Galerkin's method (see ORDINARY DIFFERENTIAL EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Gamma and related functions (see SPECIAL FUNCTIONS)
Gamma distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Gaussian (normal) distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Gear's method (see ORDINARY DIFFERENTIAL EQUATIONS)
General (continuous and discrete) distribution (see PROBABILITY FUNCTIONS, RANDOM NUMBERS)
General linear model (see EXPERIMENTAL DESIGN)
Generalized inverse (see MATRICES)
Geometric distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Geometric programming (see OPTIMIZATION)
Givens transformation (see MATRICES, VECTORS)
Golden section search (see OPTIMIZATION)
Goodness-of-fit tests (see STATISTICS)
Gram-Schmidt orthogonalization (see MATRICES)

GRAPHICS L3, Q

Box plots L3d
Distribution functions L3b
EDA graphics (e.g., stem-and-leaf, box plots) L3d
Histograms L3a
Lag plots L3c5
Line printer plotting Q1, L3
Multiple plots L3c3
Probability plots L3c4 (specific distributions itemized in subclasses)
Scatter diagrams L3c
Stem-and-leaf plots L3d
Symbol plots L3c2
Time series plots L3c5
Vertical plots L3c5

Grid generation (see INTERPOLATION)
Grouped data (see STATISTICS)
H

Halfnormal distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Heaps (see DATA HANDLING)
Helmholtz equation (see PARTIAL DIFFERENTIAL EQUATIONS)
Hermite polynomials (see SPECIAL FUNCTIONS)
Hessenberg matrix (see MATRICES)
Hilbert transforms (see INTEGRAL TRANSFORMS)
Histograms (see GRAPHICS)
Horner's rule (see SPECIAL FUNCTIONS)
Householder transformation (see MATRICES, VECTORS)
Hyperbolic and inverse hyperbolic functions (see SPECIAL FUNCTIONS)
Hyperbolic initial boundary value problems (see PARTIAL DIFFERENTIAL EQUATIONS)
Hypergeometric distribution (see PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Hypergeometric function (see SPECIAL FUNCTIONS)
Hypothesis testing (see STATISTICS)

I

Incomplete designs (see EXPERIMENTAL DESIGN)
Inference (see STATISTICS)
Initial boundary value problems (see PARTIAL DIFFERENTIAL EQUATIONS)
Initial value problems (see ORDINARY DIFFERENTIAL EQUATIONS)
Inner product (see VECTORS)
Input (see DATA HANDLING)
Integer arithmetic (see ARITHMETIC)
Integer programming (see OPTIMIZATION)
Integers (see RANDOM NUMBERS)

INTEGRAL EQUATIONS I3

Fredholm integral equations I3
Volterra integral equations I3

INTEGRAL TRANSFORMS J

Convolutions J2
Fast Fourier transforms J1
Hilbert transforms J4
Laplace transforms J3
Multidimensional fast Fourier transforms J1b
One-dimensional fast Fourier transforms (real, complex, and trigonometric) J1

also see QUADRATURE, TIME SERIES

Integrals (see SPECIAL FUNCTIONS)
Integration (see QUADRATURE, ORDINARY DIFFERENTIAL EQUATIONS)

INTERPOLATION E

Aitken's method E1b
Curve fitting E1
Evaluation of fitted functions E3
Grid generation E3
Multivariate (for gridded and for scattered data) E2
Piecewise polynomials E1a
Polynomial splines E1a
Polynomials E1b
Rational functions E1c
Service routines (e.g., grid generation, evaluation of fitted functions) E3
Splines E1a, E2a, E3
Surface fitting E2
Trigonometric functions E1c
Univariate E1

Interval arithmetic (see ARITHMETIC)
Inverse distribution functions (see PROBABILITY FUNCTIONS)
Inverse regression (see REGRESSION)
Inversion (see LINEAR EQUATIONS, MATRICES)

J
Jacobi method (see LINEAR EQUATIONS, MATRICES, PARTIAL DIFFERENTIAL EQUATIONS)
Jacobian elliptic functions (see SPECIAL FUNCTIONS)
Jordan normal form (see MATRICES)

K
Kalman filtering (see TIME SERIES)
Kelvin functions (see SPECIAL FUNCTIONS)
Kendall's coefficient of concordance (see STATISTICS)
Kendall's rank correlation coefficient (see STATISTICS)
Knapsack problems (see OPTIMIZATION)
Kolmogorov-Smirnov distribution (see PROBABILITY FUNCTIONS)
Kolmogorov-Smirnov test (see STATISTICS)
Kurtosis (see STATISTICS)

L
$L_1$ (see APPROXIMATION, REGRESSION, VECTORS)
$L_2$ (see APPROXIMATION, REGRESSION, VECTORS)
$L_{\infty}$ (see APPROXIMATION, REGRESSION, VECTORS)
Lack-of-fit tests (see REGRESSION)
Lag plots (see GRAPHICS)
Laguerre method (see NONLINEAR EQUATIONS)
Laguerre polynomials (see SPECIAL FUNCTIONS)
Laguerre quadrature (see QUADRATURE)
Lambda distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Laplace equation (see PARTIAL DIFFERENTIAL EQUATIONS)
Laplace transforms (see INTEGRAL TRANSFORMS)
Latin square (see EXPERIMENTAL DESIGN)
Lattice designs (see EXPERIMENTAL DESIGN)
Leaps-and-bounds algorithm (see REGRESSION)
Least absolute value ($L_1$) (see APPROXIMATION, REGRESSION)
Least squares ($L_2$) (see APPROXIMATION, REGRESSION)
Legendre functions (see SPECIAL FUNCTIONS)
Legendre polynomials (see SPECIAL FUNCTIONS)
Levenberg-Marquardt method (see APPROXIMATION, REGRESSION)
Life testing (see STATISTICS)
Line-printer plotting (see GRAPHICS)
Linear algebra (see MATRICES, VECTORS)

LINEAR EQUATIONS D2, D8, D9

- Associated operations (e.g., matrix reorderings) D2e
- Banded systems D2a2, D2b2, D2c2, D2d2
- Complex Hermitian systems D2d
- Complex non-Hermitian systems D2c
- Hermitian indefinite systems D2d1a
- Jacobi method D2a4, D2b4
- Other matrix equations (e.g., $AX + BX = C$) D8
- Overdetermined systems of equations D9
- Positive definite systems D2b1b, D2b2, D2d1b, D2d2
- Real nonsymmetric systems D2a
- Real symmetric systems D2b
- Relaxation methods D2b4
- Singular systems D9
- SOR method D2b4
- Sparse systems D2a4, D2b4, D2c4, D2d4
- Successive over-relaxation method D2b4
- Symmetric indefinite systems D2b1a
- Triangular systems D2a3, D2c3
- Tridiagonal systems D2a2a, D2b2a, D2c2a, D2d2a
- Underdetermined systems of equations D9

also see MATRICES

Linear programming (see OPTIMIZATION)
Location (see STATISTICS)
Location problems (see OPTIMIZATION)
Log gamma function (see SPECIAL FUNCTIONS)
Log-linear model (see CATEGORICAL DATA)
Logarithmic functions and integrals (see SPECIAL FUNCTIONS)
Logical distribution (see RANDOM NUMBERS)
Logistic distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Logistic regression (see REGRESSION)
Lognormal distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
LR algorithm (see MATRICES)
LU decomposition (see MATRICES)

M

Machine-dependent constants (see SERVICE ROUTINES)
Mann-Whitney rank-sum test (see STATISTICS)
Markov models (see SIMULATION AND STOCHASTIC MODELING)
Matching problems (see OPTIMIZATION)
Mathieu functions (see SPECIAL FUNCTIONS)

MATRICES D

- Addition and subtraction D1b5
- Bandwidth and profile reduction D2e
- Cholesky decomposition D2b1b
- Copy D1b8
- Determinants D3
Diagonalization D4  
Eigenvalues and eigenvectors (ordinary eigenvalue problems, generalized eigenvalue problems and associated operations (e.g., transform problem, balance matrix, reduce to compact form, standardize problem, compute eigenvalues of matrix in compact form, form eigenvectors from eigenvalues, back transform eigenvectors) D4  
Euclidean ($L_2$) norm D1b2  
Generalized inverse D9  
Givens transformation D1b10  
Gram-Schmidt orthogonalization D5  
Hessenberg matrix D4  
Householder transformation D1b11  
Inversion D2  
Jacobi method D4a1, D4b1  
Jordan normal form D4c5  
LR algorithm D4e  
LU decomposition D2, D7a  
Matrix polynomial D1b7  
Multiplication D1b8  
Multiplication by vector D1b4  
Norm D1b2  
Pseudo-inverse D9  
QR decomposition D5  
QZ algorithm D4  
Reflection D1b11  
Reorderings (of rows and columns) D2e  
Rotation D1b10  
Set to zero and identity D1b1  
Singular value decomposition D6  
Storage mode conversion D1b9  
Transpose D1b3  
Update matrix decompositions (LU, Cholesky, QR, and singular value) D7  

also see LINEAR EQUATIONS

Maximum flow problems (see OPTIMIZATION)  
Maximum likelihood (see STATISTICS)  
Median polish (see CATEGORICAL DATA)  
Merging (see DATA HANDLING)  
Mesh generation (see APPROXIMATION)  
Minimax ($L_{\infty}$) (see APPROXIMATION, REGRESSION)  
Minimum spanning tree (see OPTIMIZATION)  
Missing values (see REGRESSION, STATISTICS)  
Moments (see STATISTICS)  
Monte Carlo integration (see QUADRATURE)  
Moving averages (see TIME SERIES)  
Muller’s method (see NONLINEAR EQUATIONS)  
Multi-dimensional integrals (see QUADRATURE)  
Multinomial distribution (see RANDOM NUMBERS)  
Multiphase regression (see REGRESSION)  
Multiple comparisons (see EXPERIMENTAL DESIGN)  
Multiple regression (see REGRESSION)  
Multivariate analysis of variance (see EXPERIMENTAL DESIGN)  
Multivariate distribution and density functions (see PROBABILITY FUNCTIONS)  
Multivariate regression (see REGRESSION)
N

Negative binomial distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Nested designs (see EXPERIMENTAL DESIGN)
Network reliability (see SIMULATION AND STOCHASTIC MODELING)
Network problems (see OPTIMIZATION)

NONLINEAR EQUATIONS  F

Bisection method  F1b
Laguerre method  F1a1a
Muller’s method  F1a2
Polynomial (real and complex coefficients)  F1a1
Service routines (e.g., check user-supplied derivatives)  F3
Single nonlinear equation (smooth or general (no smoothness assumed))  F1
System of equations (smooth or general (no smoothness assumed))  F2

Nonlinear problems (see APPROXIMATION, NONLINEAR EQUATIONS, OPTIMIZATION, REGRESSION)
Nonparametric statistics (see EXPERIMENTAL DESIGN, STATISTICS)
Nonseparable elliptic boundary value problems (see PARTIAL DIFFERENTIAL EQUATIONS)
Norm (see MATRICES, VECTORS)
Normal distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Normal scores (see PROBABILITY FUNCTIONS)

NUMBER THEORY (e.g., prime numbers)  B

O

One-way analysis of variance (see EXPERIMENTAL DESIGN)
Optimal control (see OPTIMIZATION)

OPTIMIZATION  G

Assignments problem  G2b
Barrier method  G2h3
Binary search  G1a2
Bisection method  G1a2
Conjugate gradient method  G1b1
Constrained  G2
Covering problems  G2c2
Dynamic programming  G2g
Fletcher and Reeves’ method  G1b1
Geometric programming  G2f
Global solution to nonconvex problems  G2i
Golden section search  G1a2
Integer programming  G2c
Knapsack problems  G2c3
Linear programming (including simplex method)  G2a
Location problems  G2c5
Matching problems  G2c4
Maximum flow problems  G2d3
Minimum spanning tree  G2d2
Network problems (including test problem generation)  G2d
Nonlinear programming (simple bounds, linear equality or inequality constraints, nonlinear constraints)  G2h
Optimal control  G3
Packing problems  G2c2
Penalty method  G2h
Quadratic programming G2e
Quasi-Newton methods G1b1
Routing problems G2c5
Scheduling problems G2c5
Service routines (including problem input (e.g., matrix generation), problem scaling, checking user-supplied derivatives, finding feasible point, and checking for redundancy) G4
Shortest path problems G2d1
Simplex method G1b2, G2a
Transportation problems G2b
Unconstrained G1

Order statistics (see RANDOM NUMBERS)

ORDINARY DIFFERENTIAL EQUATIONS II

Adams’ predictor-corrector method IIa1b
Algebraic-differential systems IIa2
Boundary value problems IIb
Bulirsch-Stoer method IIa1c
Collocation method IIb
Extrapolation methods IIa1c
Galerkin’s method IIb
Gear’s method IIa2
Initial value problems IIa
Initial value problems - general, non-stiff or mildly stiff IIa1
Initial value problems - stiff and mixed algebraic-differential equations IIa2
Multi-point boundary value problems (linear, nonlinear, and eigenvalue problems) IIb
Multistep methods IIa1b
One-step methods IIa1a
Predictor-corrector methods IIa1b
Runge-Kutta methods IIa1a
Service routines (interpolation of solutions, error handling) IIc
Shooting method IIb
Stiff initial value problems IIa2
Sturm-Liouville IIb3

Ordinary least squares (see REGRESSION)
Oscillatory integrals (see QUADRATURE)
Orthogonal matrix (see RANDOM NUMBERS)
Orthogonal polynomials (see REGRESSION, SPECIAL FUNCTIONS)
Output (see DATA HANDLING)
Overdetermined systems of equations (see LINEAR EQUATIONS)

P

Packing problems (see OPTIMIZATION)
Pade approximation (see APPROXIMATION)
Parabolic cylinder functions (see SPECIAL FUNCTIONS)
Parabolic initial boundary value problems (see PARTIAL DIFFERENTIAL EQUATIONS)
Parameter estimation (see STATISTICS)
Pareto distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)

PARTIAL DIFFERENTIAL EQUATIONS I2

Biharmonic equation I2b1c
Collocation method II
Cyclic reduction method I2b4b
Domain triangulation I2b4a
Elliptic boundary value problems (linear, nonlinear, and eigenvalue problems) I2b
Galerkin's method I2
Helmholtz equation I2b1a1
Hyperbolic initial boundary value problems I2a2
Initial boundary value problems I2a
Jacobi method I2b4b
Laplace equation I2b1a1
Nonseparable elliptic boundary value problems I2b1a3
Parabolic initial boundary value problems I2a1
Poisson equation I2b1a1
Relaxation methods I2b4b
Separable elliptic boundary value problems I2b1a1, I2b1a2
Service routines (including domain triangulation and solution of discretized elliptic equations) I2b4
SOR method I2b4b
Stone's procedure I2b4b
Strongly implicit procedure I2b4b
Successive over-relaxation method I2b4b
Triangulation I2b4a

Penalty method (see OPTIMIZATION)
Permutations (see DATA HANDLING, RANDOM NUMBERS)
PERT (see SIMULATION AND STOCHASTIC MODELING)
Piecewise polynomials (see APPROXIMATION, INTERPOLATION, REGRESSION)
Plotting (see GRAPHICS)
Poisson distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Poisson equation (see PARTIAL DIFFERENTIAL EQUATIONS)
Polygamma function (see SPECIAL FUNCTIONS)
Polynomial splines (see APPROXIMATION, INTERPOLATION, REGRESSION)
Polynomials (see APPROXIMATION, INTERPOLATION, NONLINEAR EQUATIONS, REGRESSION, SPECIAL FUNCTIONS)
Powers (see SPECIAL FUNCTIONS)
Predictor-corrector methods (see ORDINARY DIFFERENTIAL EQUATIONS)
Prime numbers (see NUMBER THEORY)
Principal components (see REGRESSION, STATISTICS)
Principal value integrals (see QUADRATURE)
Probability (see STATISTICS)

PROBABILITY FUNCTIONS L5

Density and distribution functions L5a1 (specific distributions itemized in subclasses)
Inverse distribution functions (percent point functions) L5a2 (specific distributions itemized in subclasses)
Multivariate distribution and density functions L5b1 (specific distributions itemized in subclasses)
Sparsity functions L5a2 (specific distributions itemized in subclasses)

also see SPECIAL FUNCTIONS

Probability plots (see GRAPHICS)
Project optimization (e.g., PERT) (see SIMULATION AND STOCHASTIC MODELING)
Proportional data (see STATISTICS)
Pseudo-inverse (see MATRICES)
Psi function (see SPECIAL FUNCTIONS)
QR decomposition (see MATRICES)
Quadratic programming (see OPTIMIZATION)

QUADRATURE (numerical evaluation of definite integrals) H2

Adaptive quadrature H2
Clenshaw-Curtis method H2a2a
Finite interval integrals (general integrand) H2a1
Finite interval integrals (special integrands including weight functions, oscillatory and singular integrands, principal value integrals, splines) H2a2
Infinite interval integrals (including $\exp(-x^2)$ weight function) H2a4
Laguerre quadrature H2c
Monte Carlo integration H2b1a2
Multi-dimensional integrals over hyper-rectangular regions H2b1
Multi-dimensional integrals over non-rectangular regions H2b2
Oscillatory integrals H2a2
Principal value integrals H2a2
Romberg integration H2a1a1
Semi-infinite interval integrals (including $\exp(-x)$ weight function) H2a3
Service routines (compute weights and nodes for quadrature formulas) H2c
Splines H2a1b2, H2a2a1

Qualitative data (see STATISTICS)
Quality control (see SIMULATION AND STOCHASTIC MODELING)
Quasi-Newton methods (see OPTIMIZATION)
Queueing (see SIMULATION AND STOCHASTIC MODELING)
QZ algorithm (see MATRICES)

R

RANDOM NUMBERS L6

Multivariate L6b (specific distributions itemized in subclasses)
Univariate L6a (specific distributions itemized in subclasses)

Rank tests (see STATISTICS)
Ranking (see DATA HANDLING)
Rational arithmetic (see ARITHMETIC)
Rational functions (see APPROXIMATION, INTERPOLATION)
Real arithmetic (see ARITHMETIC)
Reciprocal gamma function (see SPECIAL FUNCTIONS)
Reciprocals (see SPECIAL FUNCTIONS)
Rectangular (uniform) distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Reflection (see MATRICES, VECTORS)

REGRESSION L8

Biased (ridge) L8b
Calibration (inverse regression) L8a1d
Chebyshev L8d
Correlation data, from L8a4a1e
Design L8a6
Diagnostics L8a9
EDA (exploratory data analysis) L8f
Errors in variables L8a1c, L8a4b
Exploratory data analysis L8f
Hypothesis testing L8a10
Inference L8a10
Inverse regression L8a1d
$L_1$ L8c
$L_2$ L8a
$L_\infty$ L8d
Lack-of-fit tests L8a10a
Leaps-and-bounds algorithm L8a5
Levenberg-Marquardt method L8g1a, L8g2a
Linear least absolute value ($L_1$) L8c
Linear least squares ($L_2$) L8a
Linear minimax ($L_\infty$ or Chebyshev) L8d
Logistic L8a4d
Missing values L8a1a1b, L8a4a1b
Multiphase L8a3
Multiple L8a4
Multivariate L8a8
Nonlinear L8g
Ordinary L8a1a, L8a4a
Orthogonal polynomials L8a2b
Piecewise polynomial L8a3
Polynomial L8a2
Polynomial spline L8a3
Preference pairs, using L8a4a1e
Principal components, using L8a4a1d
Residual analysis L8a10b
Ridge L8b
Robust L8e
Several multiple regressions L8a7
Simple L8a1
Spline L8a3
Stepwise L8a5
Through the origin L8a1b
Unweighted L8a1a, L8a2a, L8a2b1, L8a4a1, L8g1
Variable selection L8a5
Weighted L8a1a2, L8a2a2, L8a2b2, L8a4a2, L8g2

also see APPROXIMATION

Regression design (see REGRESSION)
Relaxation methods (see LINEAR EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Reliability (see SIMULATION AND STOCHASTIC MODELING)
Repeated measures (see EXPERIMENTAL DESIGN)
Residual analysis (see REGRESSION)
Ridge regression (see REGRESSION)
Riemann zeta function (see SPECIAL FUNCTIONS)
Robust regression (see REGRESSION)
Romberg integration (see QUADRATURE)
Roots (see SPECIAL FUNCTIONS)
Rotation (see MATRICES, VECTORS)
Routing problems (see OPTIMIZATION)
Runge-Kutta methods (see ORDINARY DIFFERENTIAL EQUATIONS)
S

Samples (see RANDOM NUMBERS)
Scale (see STATISTICS)
Scatter diagrams (see GRAPHICS)
Scheduling problems (see OPTIMIZATION)
Scorer functions (see SPECIAL FUNCTIONS)
Searching (see DATA HANDLING)
Separable elliptic boundary value problems (see PARTIAL DIFFERENTIAL EQUATIONS)
Sequences (see ARITHMETIC, STATISTICS)

SERVICE ROUTINES R

Documentation retrieval R4
Error checking (e.g., check monotonicity) R2
Error handling (including setting criteria for fatal error, setting unit number for error messages) R3
Machine-dependent constants R1

Shape (see STATISTICS)
Shooting method (see ORDINARY DIFFERENTIAL EQUATIONS)
Shortest path problems (see OPTIMIZATION)
Simple regression (see REGRESSION)
Simplex method (see OPTIMIZATION)

SIMULATION AND STOCHASTIC MODELING M

Project optimization (e.g., PERT) M4
Queueing M2
Reliability (quality control and electrical network) M3
Simulation (discrete and continuous (Markov models)) M1

Sine integrals (see SPECIAL FUNCTIONS)
Single precision arithmetic (see ARITHMETIC)
Singular value decomposition (see MATRICES)
Skewness (see STATISTICS)
Smoothing (see APPROXIMATION, TIME SERIES)

SOFTWARE DEVELOPMENT TOOLS S

Dynamic analysis S3
Program transformation S1
Static analysis S2

SOR method (see LINEAR EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Sorting (see DATA HANDLING)
Sparsity functions (see PROBABILITY FUNCTIONS)
Spearman rank-correlation coefficient (see STATISTICS)

SPECIAL FUNCTIONS C

Airy function C10d
Anger functions C10e
Bessel functions C10
Bessel functions, integrals of C10f
Beta and related functions C7b, C7f
Binomial coefficient C1
Ceiling C1
Chebyshev polynomials C3a2
Chebyshev series C3a2
Confluent hypergeometric functions C11
Cosine integrals C6
Coulomb wave functions C12
Dawson's integral C8c
Elliptic integrals C14
Error functions, their inverses, and their integrals C8
Exponential functions C4b
Exponential integrals C5
Factorial C1
Floor C1
Fresnel integrals C8b
Gamma and related functions C7
Hermite polynomials C3a4
Horner's rule C3
Hyperbolic and inverse hyperbolic functions C4c
Hypergeometric function C11
Integer-valued functions (including floor, ceiling, factorial, and binomial coefficient) C1
Integrals of elementary transcendental functions C4d
Jacobian elliptic functions C13
Kelvin functions C10c
Laguerre polynomials C3a3
Legendre functions C9
Legendre polynomials C3a2
Logarithmic functions C4b
Logarithmic integrals C5
Mathieu functions C17
Non-orthogonal polynomials C3b
Orthogonal polynomials C3a
Parabolic cylinder functions C10
Polygamma function C7d
Polynomials C3
Powers C2
Psi function C7c
Reciprocal gamma function C7a
Reciprocals C2
Riemann zeta function C7g
Roots C2
Scorer functions C10d
Sine integrals C6
Spheroidal wave functions C18
Struve functions C10e
Theta functions C13
Transcendental functions C4
Trigonometric and inverse trigonometric functions C4a
Trigonometric integrals C6
Trigonometric polynomials C3a1
Weber functions C10e
Weierstrass elliptic functions C15

also see PROBABILITY FUNCTIONS

Spectral analysis (see TIME SERIES)
Spheroidal wave functions (see SPECIAL FUNCTIONS)
Splines (see APPROXIMATION, DIFFERENTIATION, INTERPOLATION, QUADRATURE, REGRESSION)
Stable distribution (see RANDOM NUMBERS)
Stacks (see DATA HANDLING)
Standard deviation (see STATISTICS)
STATISTICS L

Analysis of variance L7
Categorical data analysis L9
Cluster analysis L14
Confidence intervals L4
Contingency tables L2b, L6b, L9
Correlation analysis L10c, L10g, L11
Correlation coefficient L1e1b, L4b, L4d
Density functions L1a1d, L4a1e, L5a1, L5b1
Discriminant analysis L12
Dispersion L1a1b, L1a3b, L4
Distribution functions L1a1d, L4a1e, L5a1, L5b1
Distribution-free analysis L4a1b, L4b1b, L4e1b, L4e2b, L4e3b, L7a1b, L7a2a2, L7a2b2, L7a4b
EDA (experimental data analysis) L3d, L8f, L9d
Experimental design L7
Factor analysis L13
Goodness-of-fit tests L4a1c
Graphics L3
Grouped data L1a3, L1e3, L2b, L4a3, L4b3, L4e3
Hypothesis testing L4
Inference L4
Kendall's coefficient of concordance L4a1b
Kendall's rank correlation coefficient L4b1b
Kolmogorov-Smirnov test L4a1c, L4b1b
Kurtosis L1a1c, L1a3e
Life testing L15
Location L1a1a, L1a3a, L1e1a, L4
Manipulation, data L2
Mann-Whitney rank-sum test L4a1b
Maximum likelihood L7a4, L8a4d, L8g, L10e, L10g1, L13
Missing values L1a2, L1e2, L4a2, L4b2, L4e2, L8a1a1b, L8a4a1b
Moments L1, L4
Nonparametric statistics L4a1b, L4b1b, L4e1b, L4e2b, L4e3b, L7a1b, L7a2a2, L7a2b2, L7a4b
Parameter estimation (e.g., binomial, extreme value, normal or Gaussian, Poisson, uniform or rectangular, Weibull) L4a
Principal components L8a4a1d, L13a
Probability functions L5
Proportional data L1c, L1e, L4c, L4d, L4e
Qualitative data L1c, L1e, L4c, L4d, L4e
Random numbers L6
Rank tests L4a1b, L4b1b
Regression L8
Sample L2c
Scale L1a1b, L1a3b, L4
Sequences of numbers L4a1d
Shape L1a1c, L1a3c
Skewness L1a1c, L1a3c
Spearman rank-correlation coefficient L4a1b
Standard deviation L1a1b, L1a3b, L4
Subset L2d
Summarization, data L1
Survival analysis L15
Time series L10
Tolerance limits L4a1f
Transformations L2a, L10a
Variance L1a1b, L1a3b, L4
Wilcoxon rank-sum test L4b1b
Wilcoxon signed-rank test L4a1b

Stem-and-leaf (see GRAPHICS)
Stiff initial value problems (see ORDINARY DIFFERENTIAL EQUATIONS)
Stone's procedure (see PARTIAL DIFFERENTIAL EQUATIONS)
Storage management (see DATA HANDLING)
Storage mode conversion (see MATRICES)
Strongly implicit procedure (see PARTIAL DIFFERENTIAL EQUATIONS)
Struve functions (see SPECIAL FUNCTIONS)
Sturm-Liouville (see ORDINARY DIFFERENTIAL EQUATIONS)
Successive over-relaxation method (see LINEAR EQUATIONS, PARTIAL DIFFERENTIAL EQUATIONS)
Surface fitting (see APPROXIMATION, INTERPOLATION)
Survival analysis (see STATISTICS)
Swap (see VECTORS)
Symbol plots (see GRAPHICS)

SYMBOLIC COMPUTATION O

System of equations (see NONLINEAR EQUATIONS)

t-distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS)
Taylor polynomial approximation (see APPROXIMATION)
Theta functions (see SPECIAL FUNCTIONS)

TIME SERIES L10

ARMA and ARIMA modeling and forecasting L10e
Autocorrelation analysis L10c
Box-Jenkins analysis L10e
Complex demodulation L10d
Cross-correlation analysis L10g
Filtering L10b
Kalman filtering L10b
Moving averages L10b, L10e1
Plots (see GRAPHICS)
Random number generation L6a20
Smoothing L10b
Spectral analysis L10f
Transformations L10a
Transforms L10a (also see INTEGRAL EQUATIONS)

Tolerance limits (see STATISTICS)
Transcendental functions and their integrals (see SPECIAL FUNCTIONS)
Transformations (see MATRICES, STATISTICS, TIME SERIES, VECTORS)
Transforms (see INTEGRAL TRANSFORMS, TIME SERIES)
Transportation problems (see OPTIMIZATION)
Transpose (see MATRICES)
Trees (see DATA HANDLING)
Triad (see VECTORS)
Triangular distribution (see RANDOM NUMBERS)
Triangulation (see PARTIAL DIFFERENTIAL EQUATIONS)
Trigonometric and inverse trigonometric functions (see SPECIAL FUNCTIONS)
Trigonometric functions (see APPROXIMATION, INTERPOLATION)
Trigonometric integrals (see SPECIAL FUNCTIONS)
Trigonometric polynomials (see SPECIAL FUNCTIONS)
Two-way analysis of variance (see EXPERIMENTAL DESIGN)
Two-way tables (see CATEGORICAL DATA)
Type conversion (see ARITHMETIC)

U

Unbalanced design (see EXPERIMENTAL DESIGN)
Unconstrained approximation (see APPROXIMATION)
Unconstrained optimization (see OPTIMIZATION)
Underdetermined systems of equations (see LINEAR EQUATIONS)
Uniform distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)

V

Variable selection in regression (see REGRESSION)
Variance (see STATISTICS)
Variance components (see EXPERIMENTAL DESIGN)

VECTORS Dla

Convolutions Dla10
Copy Dla5
Dot product Dla4
Exchange Dla5
Euclidean ($L_2$) norm Dla3b
Givens transformation Dla8
Householder transformation Dla9
Inner product Dla4
$L_1$ norm (sum of magnitudes) Dla3a
$L_2$ norm Dla3b
$L_\infty$ norm Dla3c
Minimum and maximum components Dla2
Multiplication by scalar Dla6
Norm Dla3
Reflection Dla9
Rotation Dla8
Set to constant Dla1
Swap Dla5
Triad (ax+ay for vectors x, y and scalar a) Dla7

Vertical plots (see GRAPHICS)
Volterra integral equation (see INTEGRAL EQUATIONS)
Von Mises distribution (see RANDOM NUMBERS)

W

Weber functions (see SPECIAL FUNCTIONS)
Weibull distribution (see GRAPHICS, PROBABILITY FUNCTIONS, RANDOM NUMBERS, STATISTICS)
Weierstrass elliptic functions (see SPECIAL FUNCTIONS)
Weighted regression (see REGRESSION)
Wilcoxon rank-sum test (see STATISTICS)
Wilcoxon signed-rank test (see STATISTICS)
The second edition of the Guide to Available Mathematical Software (GAMS) provides summary documentation of software available to NBS staff on a variety of computers. The fifteen libraries documented in GAMS are: BMDP, CMLIB, (containing three dozen public-domain packages), DATAPAC, IMSL, INVAR, MATHWARE, MATLAB, MINITAB, NAG, PDELIB, PLOD, PORT, SLDGL, SPECTRLAN, STATLIB. GAMS is based on an extensive problem-oriented scheme for classifying software for mathematical computations including special functions, linear algebra, optimization, differentiation and integration, differential and integral equations, and statistics and probability. The document contains the classification scheme, a catalog of software organized by class, a dictionary of the software, library references, and an index.
The following people can provide assistance using GAMS. This includes identifying the most appropriate software, using software, or finding further information. Please contact one of them if you find errors in this document.

Help using software on the Sperry 1100/82 and Perkin-Elmer minicomputers in Gaithersburg is available from:

Paul Boggs ...................... numerical analysis (301 or FTS 921-3395)
Ronald Boisvert ................ numerical analysis (301 or FTS 921-3395)
Elsie Clark ...................... statistics (301 or FTS 921-3395)
Karla Hoffman .................. linear and nonlinear programming (301 or FTS 921-3855)
Sally Howe ...................... statistics (301 or FTS 921-3395)
Ric Jackson ...................... optimization (301 or FTS 921-3855)
David Kahaner .................. numerical analysis (301 or FTS 921-3395)
Janice Knapp-Cordes .......... IMSL, NAG, and PORT (301 or FTS 921-3395)
Daniel Lozier ................... special functions (301 or FTS 921-2631)
Patsy Saunders ................... simulation (301 or FTS 921-3855)

To establish an account on the Sperry 1100, or for other assistance in using the central computing facility in Gaithersburg, please contact the NBS Computer Services Division at 301 or FTS 921-3580 (for accounts, call 301 or FTS 921-3364).

Help using software on the Cyber 750 and Perkin-Elmer minicomputers in Boulder is available from:

Janet Donaldson ................ statistics (303-497-5114 or FTS 320-5114)
Lloyd Fosdick .................... numerical analysis (303 497-3836 or FTS 320-3836)
John Gary ......................... numerical analysis (303 497-3369 or FTS 320-3369)
John Koontz ...................... statistics (303 497-5180 or FTS 320-5180)
Linda Lindgren .................. numerical analysis (303 497-5149 or FTS 320-5149)
Roland Sweet ..................... numerical analysis (303 497-5871 or FTS 320-5871)

To establish an account on the Cyber 750, or for other assistance in using the central computing facility in Boulder, please contact the NOAA Computer Services Division at (303) 497-5850 or FTS 320-5850.