NISTIR 89-4159

FACTUNC: A USER-FRIENDLY SYSTEM FOR UNCONSTRAINED OPTIMIZATION

Richard H.F. Jackson
Garth P. McCormick
Arieia Sofer

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Engineering Laboratory
Center for Manufacturing Engineering
Gaithersburg, MD 20899

A report for:
U. S. Department of Commerce
National Institute of Standards
and Technology
Center for Manufacturing Engineering
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

NISTIR 89-4159

FACTUNC: A USER-FRIENDLY SYSTEM FOR UNCONSTRAINED OPTIMIZATION

Richard H.F. Jackson
Garth P. McCormick
Ariela Sofer

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Engineering Laboratory
Center for Manufacturing Engineering
Gaithersburg, MD 20899

August 1989

A report for:
U. S. Department of Commerce
National Institute of Standards
and Technology
Center for Manufacturing Engineering
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

FACTUNC: A USER-FRIENDLY SYSTEM FOR UNCONSTRAINED OPTIMIZATION

Richard H. F. Jackson
Center for Manufacturing Engineering
National Bureau of Standards
Gaithersburg, MD 20899

Garth P. McCormick
Department of Operations Research
George Washington University
Washington, DC 20052

Ariela Sofer
Department of Operations Research and Applied Statistics
George Mason University
Fairfax, VA 22030

ABSTRACT

FACTUNC is a system for solving unconstrained minimization problems based on the concept of factorable programming. This concept enables the user to provide the problem function and data in a user friendly way and does not require user-supplied derivatives. The system utilizes the factorable function concept to obtain the first and second derivatives required for unconstrained optimization. In all cases derivatives are obtained rapidly and accurately (up to roundoff errors due to machine precision), as opposed to finite differencing.

As a system for nonlinear minimization, FACTUNC allows several options. First the user can solve regression (nonlinear least squares) problems by providing the regression equation and the data for the dependent and independent variables. The second option allows for the minimization of the sum of an indexed function. The user provides the function, and the indexed data. This can be used for example to solve maximum likelihood estimation problems when the user provides the negative of the (weighted) log of the frequency function and the data. The third option is simply to minimize a function supplied by the user. Utilizing barrier function methodology, this third option can sometimes be used to solve constrained problems.

TABLE OF CONTENTS

1. INTRODUCTION

We report here on FACTUNC, a system for solving unconstrained minimization problems developed at the National Bureau of Standards. FACTUNC differs from other optimization systems in one fundamental aspect: the internal representation of functions as "factorable". In section 2 we discuss the concept and merits of factorable functions in more detail. One advantage of this approach is that once a function is represented in factorable form, its gradient, Hessian and even higher order derivatives may be calculated automatically and exactly. In order to use FACTUNC, the user need only supply the system with a statement of the problem function in a FORTRAN-like expression. The system then proceeds to translate this function statement into factorable form, and uses this representation to compute the gradient and Hessians as required.

The Hessian matrices computed by FACTUNC, are initially obtained in dyadic form, i.e., as a sum of rank one matrices. Moreover, the factors of these rank one matrices are obtained from the gradient evaluation. The advantages of this representation, and how it can be utilized are explained in more detail in section 2.

There are a number of different ways to use FACTUNC. For example, FACTUNC may be used as a general unconstrained minimization system. The system also provides a user-friendly front-end for objective functions which involve an indexed sum of terms of the same form (such as the negative of the logarithm of the likelihood function which is used to solve maximum likelihood). When working with this option, the user need only supply the general form of a typical term in the indexed sum (e.g., a typical term in the logarithm of the likelihood function). The nonlinear least squares problem is another example of an indexed sum. Since nonlinear regression problems arise frequently in applications, FACTUNC has a special option for solving this type of problem. In order to solve such a least squares minimization problem, the user need only supply the general form of the nonlinear regression equation, and, of course, the data. Finally, by using a sequential unconstrained minimization approach for solving constrained minimization, FACTUNC may be incorporated as part of a constrained optimization system. Thus, FACTUNC may be used to solve the sequence of unconstrained problems which arise in the barrier method, or in the method of centers. In each of these cases, the option of using an indexed sum can be invoked.

The use of the various options available in FACTUNC is explained in detail in sections 3 - 5. For ease of presentation, we have chosen to start with the nonlinear least squares option. The underlying approach in implementing FACTUNC is common to all the various options, and is described in section 1.1 below.

1.1 Representation of Functions in FACTUNC

Computer programs for solving nonlinear problems are widely available. A major difficulty in using the software is addressed in the FACTUNC system. This difficulty is how to construct a user-friendly way to represent the information which is required for the nonlinear minimization, concerning the nonlinear function, which can also be accessed in a computationally efficient manner by algorithms for minimization. It is solved in FACTUNC by allowing the user to provide a FORTRAN-like expression to represent the nonlinear objective function. This is then processed by the system and represented internally in the form of a factorable function. In this form the quantities required by algorithms for solving the problem: function values, first derivatives, and second derivatives are automatically computed by the system.

The statement for the function allows the regular operations such as plus (+), minus (-), product (*), division (/), power (**), and parentheses. If parentheses do not conform (e.g., an extra left parenthesis, or some other mismatch, an error statement is produced). In addition, the statement allows for single variable functions. The single-variable functions available for use by the system are given in Table 1. Note that any single-variable function can theoretically be used. These are just the ones programmed to date.

FACTUNC also accepts complex terms which involve the nesting of parentheses and single variable transformations within other, provided that the resulting expression represents a well defined mathematical expression. Thus FACTUNC will be able to accept and process a function of the form

```
A*SQRT(X**3+ (Y/(X+Y))*ENTROP(Z))-1.7*DECAY(SIN(Y)).
```

However, the expression

```
(GAMMA(X+Y))/(2*LOG(Y)))
```

is incorrect. The system will notify the user of the excess in the number of right parentheses. Removal of the last parenthesis, will result in an expression acceptable to FACTUNC.

TABLE 1. SINGLE ARGUMENT FUNCTIONS AVAILABLE IN THE FACTUNC SYSTEM

FACTUNC NAME	ALGEBRAIC NOTATION	DESCRIPTION
EXP(X)	e^{-x}	Exponential
LOG(X)	$\log_{10} x$	Common Logarithm
LN(X)	$\log_e x$	Natural Logarithm
SIN(X)	$\sin x$	Sine
COS(X)	$\cos x$	Cosine
TAN(X)	$\tan x$	Tangent
GAMMA(X)	$\Gamma(x)$	Gamma
ARCSIN(X)	$\arcsin x$	Arcsine
ARCTAN(X)	$\arctan x$	Arctangent
CUMNOR(X)	$\frac{1}{\sqrt{2\pi}} \int_{\infty}^{x} \mathrm{EXP}(-t^2/2)dt$	Cumulative Normal
SQRT(X)	$\sqrt{x}$	Square Root
DECAY(X)	$1-e^{-x}$	Decay Function
ENTROP(X)	$x \cdot \ln x$	Entropy
NORDEN(X)	$\frac{1}{\sqrt{2\pi}} \mathrm{EXP}(-x^2/2)$	Normal Density
BARLN(X)	$\begin{cases} \infty, & x \le 0 \\ -\ln x, & x > 0 \end{cases}$	Logarithmic Barrier Function

2. MATHEMATICAL BACKGROUND ON FACTORABLE FUNCTIONS

This section is provided as background for the interested reader. It is not necessary to understand the theory of factorable functions in order to use the FACTUNC system. Those interested only in using the system may therefore skip this section.

A factorable function is a multivariable function that can be written as the last of a finite sequence of functions, in which the first n functions in the sequence are just the coordinate variables, and each function beyond the nth is a sum, a product, or a single-variable transformation of previous functions in the sequence. More rigorously, let $[f_1(x), f_2(x), \ldots, f_L(x)]$ be a finite sequence of functions such that $f_i : R^n \rightarrow R$, where each $f_i(x)$ is defined according to one of the following rules.

Rule 1. For i=1, ..., n, $f_i(x)$ is the value of the i^{th} Euclidean coordinate:

$$f_i(x) = x_i . \tag{2.1}$$

Rule 2. For i=n+1, ..., L, $f_i(x)$ is formed using one of the following compositions:

$$\begin{aligned} &a.) \quad f_i(x) = f_{j(i)}(x) + f_{k(i)}(x);\ \text{or} \\ &b.) \quad f_i(x) = f_{j(i)}(x) * f_{k(i)}(x);\ \text{or} \\ &c.) \quad f_i(x) = T_i[f_{j(i)}(x)]; \end{aligned} \tag{2.2}$$

where $j(i) < i$, $k(i) < i$, and T_i is a function of a single variable. Then $f(x) = f_L(x)$ is a factorable function and $[f_1(x), f_2(x), \ldots, f_L(x)]$ is a factored sequence. Thus a function, f(x), will be called factorable if it can be formed according to Rules 1 and 2, and the resulting sequence of functions will be called a factored sequence or, at times, the function written in factored form.

Although it is not always immediately grasped, the concept of a factorable function is actually a very natural one. In fact, it is just a formalization of the natural procedure one follows in evaluating a complicated function. Consider for example the fuhction

$$f(x) = (a^T x)(\sin[b^T x])(\exp[c^T x]), \tag{2.3}$$

where a,b,c and x are (2 x 1) vectors. The natural approach to evaluating this function for specific values of x_1 and x_2 is first to compute the quantities within the parentheses, then to apply the sine and exponential functions, and finally to multiply the three resulting quantities. This might be done in stages as follows.

$$
\begin{array}{lll}
f_1 = x_1 & f_6 = b_1 f_1 & f_{11} = f_9 + f_{10} \\
f_2 = x_2 & f_7 = b_2 f_2 & f_{12} = \sin(f_8) \\
f_3 = a_1 f_1 & f_8 = f_6 + f_7 & f_{13} = \exp(f_{11}) \\
f_4 = a_2 f_2 & f_9 = c_1 f_1 & f_{14} = f_5 \cdot f_{12} \\
f_5 = f_3 + f_4 & f_{10} = c_2 f_2 & f_{15} = f_{13} \cdot f_{14}
\end{array}
\qquad (2.4)
$$

This is one possible factored sequence for the function in (2.3).

In order to appreciate fully the value of factorable functions the concept of an outer product matrix must be introduced. An (m x n) matrix A is called an outer product matrix if there exists a scalar α, an (m x 1) vector a, and an (n x 1) vector b such that

$$A = a\alpha b^T .$$

The expression $a\alpha b^T$ is called an outer product or a dyad. Note that a dyad is conformable since the dimensions of the product are (m x 1)(1 x 1)(1 x n), which yields the (m x n) outer product matrix A as desired. A useful property of outer product matrices is that, if they are kept as dyads, matrix multiplication with them is simplified to inner products alone, saving the computations required to form the matrices involved. For example,

$$
\begin{aligned}
Ac &= a\alpha[b^T c], \\
d^T A &= [d^T a]\alpha b^T, \text{ and} \\
AF &= a\alpha[b^T F],
\end{aligned}
$$

where c is (n x 1), d is (m x 1) and F is (n x m).

Factorable functions possess two very special properties that can be exploited to produce efficient (fast and accurate) algorithms: (i) once written in factorable form, their gradients and Hessians may be computed exactly, automatically, and efficiently; and (ii) their Hessians occur naturally as sums of dyads whose vector factors are gradients of terms in the factored sequence. The first of these properties is utilized in FACTUNC for providing the first and second derivatives of nonlinear functions. The second has obviated the task of multiplying a matrix by a vector, reducing it to a series of inner products, as noted above, which in many cases results in less effort.

In order to clarify these concepts, we consider again the illustrative function in (2.3). Table 1 is a display of the gradient and Hessian of this function. The entries in each column are the summands in the expressions for the gradient and Hessian. For example,

$$\nabla f = [\sin(b^T x)][\exp(c^T x)]a + [a^T x][\cos(b^T x)][\exp(c^T x)]b$$

$$+ [a^T x][\sin(b^T x)][\exp(c^T x)]c.$$

The table also illustrates the left-to-right, tree-like structure of the derivatives involved. From the table it can be seen that both the gradient and Hessian naturally have the dyadic structure discussed above. Notice too that the vectors in the monads and dyads are drawn from the set {a,b,c}, each of which is the gradient of a factored sequence function in (2.4).

TABLE 1.
MONADIC AND DYADIC STRUCTURE OF GRADIENT AND HESSIAN OF ILLUSTRATIVE FUNCTION

Function	Gradient Summands	Hessian Summands
$a^T x \cdot \sin[b^T x]\exp[c^T x]$	$(\sin[b^T x]\exp[c^T x]:a)$	$(\cos[b^T x]\exp[c^T x]:ab)$ $(\sin[b^T x]\exp[c^T x]:ac)$
	$(a^T x \cdot \cos[b^T x]\exp[c^T x]:b)$	$(\cos[b^T x]\exp[c^T x]:ba)$ $(-a^T x \cdot \sin[b^T x]\exp[c^T x]:bb)$ $(a^T x \cdot \cos[b^T x]\exp[c^T x]:bc)$
	$(a^T x \cdot \sin[b^T x]\exp[c^T x]:c)$	$(\sin[b^T x]\exp[c^T x]:ca)$ $(a^T x \cdot \cos[b^T x]\exp[c^T x]:cb)$ $(a^T x \cdot \sin[b^T x]\exp[c^T x]:cc)$

It is important to understand that the derivative calculations performed by the FACTUNC system are not estimations, but mathematically exact calculations. Furthermore, they are also compact, since factored sequences mimic hand calculations, and thus this technique is different from symbolic manipulation techniques for differentiation, which tend to produce large amounts of code. The techniques used in Factorable Programming are efficient exploitations of the special structure inherent in factorable functions and their partial derivative arrays. These results were extended to higher order derivatives by Jackson and McCormick [1986], who showed that factorable functions have arrays of n^{th} order derivatives (tensors) which are naturally computed as sums of generalized outer product matrices (polyads).

3. SOLVING REGRESSION (LEAST SQUARES) PROBLEMS (OPTION 1)

We begin the discussion of the use of the code with a description of the simplest way to use it: for solving nonlinear regression problems.

A regression equation relates a dependent variable to a function involving one or more independent variables and parameters. Given a set of values of the dependent variable and simultaneously observed values of the independent variables, the "best" values of the parameters in a least squares sense are those which minimize the sum of the squared differences between the observed dependent values and the estimated ones. Models giving rise to least squares problems are legion in the physical and social sciences. The FACTUNC system provides a user friendly way to represent the data and the regression function. This is particularly important when the parameters enter nonlinearly in the regression function.

In the FACTUNC system, one variable, y, is the dependent variable and there may be one or more independent variables $x_1,\ldots,x_n$. The problem modeler must specify the form of the function f which is believed to represent the relationship between the variables. The function has parameters $a_1,\ldots,a_m$ which must be calculated in a way to give a "best" fit to the observed data. Ideally, the relationship $y = f(a_1,\ldots,a_m;x_1,\ldots,x_n)$ would hold precisely, i.e.,

$$y_i = f(a_1, \ldots, a_m;x_1^i, \ldots, x_n^i)$$

for each i=1,2,...,k, where k is the total number of observations and the super/subscript i indicates the values of the variables at the i^{th} observation. However, it will not usually be possible to get an exact fit. In the least squares method, the parameters are chosen to minimize the quantity

$$\sum_{i=1}^{k} [y_i-f(a_1, \ldots, a_m;x_1^i, \ldots, x_n^i)]^2,$$

the sum of the squares of the residuals. A least squares problem is said to be nonlinear if the parameters appear nonlinearly in the regression equation. For more discussion see McCormick [1983], pp. 93-99.

To solve a nonlinear least squares problem using FACTUNC, the user must provide an input file giving the regression equation, the names of the parameters along with their starting values and upper and lower bounds, and a list of the observed data. The general structure of the input file is explained in the following.

Input File Format

(1) NLSQ

All characters on this line are ignored and can be used as a comment to entitle the problem.

(2) This can also be used to identify the regression equation.

(3) Regression equation - given in the following format:

(dependent variable) = (expression involving functions, parameters, constants, and independent variables)$

The notation follows the usual FORTRAN conventions (e.g. "*" indicates multiplication). See the table for a list of the allowed functions. The equation may be written on several lines, but the "=" must appear on the first line. The end of the equation is signalled by the "$". Spaces are ignored.

(4) Comment - demarcates problem constants section (must be present).

(5) Problem constants - number of parameters, constants, independent variables, and observations (in free format).

(6) Comment - used to demarcate parameter section (must be present).

(7) Parameter section - A line is given for each parameter in the regression equation showing the parameter name in quotes, a starting value, a lower bound, and an upper bound (in free format).

(8) Comment - used to demarcate constant section (must be present).

(9) Constant section - A line is given for each named constant in the regression equation showing the constant name in quotes and its value. This section may be blank, but the preceding comment (8) must always be present.

(10) Comment - used to demarcate data section (must be present).

(11) Data section - A line is given showing the names of the variables (in quotes) in the regression equation. Under each variable name a column of data values is listed. Each row of this section represents one observation.

As an example of the use of the regression (least squares) option consider the data in Table 2. The elements in the table are the ratios of those people who died in the category indicated in the appropriate row and column, divided by those who would be expected to die if the sample were taken from a general population of nonsmokers. If the matrix elements are denoted by DEATH, the row indicator by DEPTH, and the column indicator by CIGS, a reasonable regression equation is

$$\text{DEATH} = 1 + (a_1 + a_2\text{DEPTH})(1 - \text{EXP}(-a_3\text{CIGS}))$$

where (a_1, a_2, a_3) are parameters. The values of DEPTH and CIGS need to be quantified. The values used for this example are given in Figure 1, the input file for FACTUNC. The optimization problem is:

$$\min_{a_i} \sum_{i=1}^{16} [\text{DEATH}_i - \{1 + (a_1 + a_2\text{DEPTH}_i)(1 - \text{EXP}(-a_3\text{CIGS}_i))\}]^2$$

TABLE 2. DATA ON SMOKING AND HEALTH

Degree of Inhalation	Number of Cigarettes per Day			
	1-9	10-19	20-40	40+
None	1.29	1.46	1.56	2.05
Slight	1.29	1.68	1.84	1.97
Moderate	1.61	1.82	1.84	2.01
Deep	1.88	1.76	2.18	2.50

The general format of the input file for this example is given in Figure 1, with each line of the file numbered and keyed to the input file format given above. Figure 2 is the exact input file for this example, while Figure 3 is part of the FACTUNC output. Table 1 is a list of the functions of a single argument which can be used in describing input functions.

FIGURE 1. SAMPLE INPUT FILE FOR LEAST SQUARES OPTIONS.*

```
 (1) NLSQ - EXCESS DEATHS AS FCN OF DAILY SMOKES AND DEPTH OF INHALATION

 (2) REGRESSION EQUATION

 (3) DEATH=1.+(A1*DEPTH+A2)*(C1-EXP(-A3*CIGS))$

 (4) NUMBER OF PARAMETERS, CONSTANTS, INDEPENDENT VARIABLES, OBSERVATIONS

 (5)      3    1    2    16

 (6) PARAMETERS, STARTING VALUE, LOWER BOUND, UPPER BOUND

 (7)       'A1'    0.1     0.    10.
           'A2'    0.6     0.    10.
           'A3'   0.01     0.    10.

 (8) CONSTANTS: NAME, VALUE

 (9)       'C1'    1.

(10) VARIABLES: NAME ABOVE COLUMN

(11)      'DEATH'    'CIGS'       'DEPTH'

           1.29        5.          0.
           1.46       15.          0.
           1.56       30.          0.
           2.05       65.          0.
           1.29        5.           .3333
           1.68       15.           .3333
           1.84       30.           .3333
           1.97       65.           .3333
           1.61        5.           .6667
           1.82       15.           .6667
           1.84       30.           .6667
           2.01       65.           .6667
           1.88        5.          1.0000
           1.76       15.          1.0000
           2.18       30.          1.0000
           2.50       65.          1.0000
```

*All inputs are free form. The numbers in parentheses correspond to the numbers in the explanation above, but are not part of the input file.

FIGURE 2. Exact Input File for Smoking and Health Regression Problem

```
NLSQ - SMOKING AND HEALTH REGRESSION
    REGRESSION EQUATIONS
DEATH = 1. +(A1*DEPTH+A2)*(C1-EXP(-A3*CIGS))$
NUMBER OF PARAMETERS,CONSTANTS,INDEPENDENT VARIABLES,OBSERVATIONS
     3    2    2    16
PARAMETERS , STARTING VALUE, LOWER BOUND, UPPER BOUND
     'A1'   4.   0.   10.
     'A2'   3.   0.   10.
     'A3'    .1  0.   10.
CONSTANTS: NAME, VALUE
     'DUM'    2.2
     'C1'     1.
VARIABLES: NAME ABOVE COLUMN: THOSE NOT HERE ARE ASSUMED ON FILES
     'DEATH'          'CIGS'            'DEPTH'
       1.29             5.                0.
       1.46            15.                0.
       1.56            30.                0.
       2.05            65.                0.
       1.29             5.              .3333
       1.68            15.              .3333
       1.84            30.              .3333
       1.97            65.              .3333
       1.61             5.              .6667
       1.82            15.              .6667
       1.84            30.              .6667
       2.01            65.              .6667
       1.88             5.             1.000
       1.76            15.             1.000
       2.18            30.             1.000
       2.50            65.             1.000
```

FIGURE 3. Partial Output for Least Squares Example.

```
NLSQ - SMOKING AND HEALTH REGRESSION
LEAST SQUARES PROBLEM
HESSIAN FORMED EXPLICITLY
     REGRESSION EQUATION
     DEATH = 1. +(A1*DEPTH+A2)*(C1-EXP(-A3*CIGS))$
SINGLE CHARACTER OUTPUT STRING OF (FUNCTION) EQUATION
     1.+(A1*DEPTH+A2)*(C1-EXP(-A3*CIGS))$
NUMBER OF PARAMETERS,CONSTANTS,INDEPENDENT VARIABLES,OBSERVATIONS
THE NUMBER OF UNKNOWNS IS =  3
THE NUMBER OF SYMBOLIC CONSTANTS IN THE EQUATION (FUNCTION) =  2
THE NUMBER OF INDEPENDENT VARIABLES =   2
THE NUMBER OF DATA OBSERVATIONS =  16
PARAMETERS , STARTING VALUE, LOWER BOUND, UPPER BOUND
     A1          .400000E+01    .000000E+00    .100000E+02
     A2          .300000E+01    .000000E+00    .100000E+02
     A3          .100000E+00    .000000E+00    .100000E+02
CONSTANTS: NAME, VALUE
     DUM    .220000E+01
     C1     .100000E+01
VARIABLES: NAME ABOVE COLUMN
       DEATH      CIGS         DEPTH
     1.29000     5.00000      .00000
     1.46000    15.00000      .00000
     1.56000    30.00000      .00000
     2.05000    65.00000      .00000
     1.29000     5.00000      .33330
     1.68000    15.00000      .33330
     1.84000    30.00000      .33330
     1.97000    65.00000      .33330
     1.61000     5.00000      .66670
     1.82000    15.00000      .66670
     1.84000    30.00000      .66670
     2.01000    65.00000      .66670
     1.88000     5.00000     1.00000
     1.76000    15.00000     1.00000
     2.18000    30.00000     1.00000
     2.50000    65.00000     1.00000

THE OPTIMAL VALUES OF THE UNKNOWNS ARE
     1 A1      =       .54488E+00
     2 A2      =       .74750E+00
     3 A3      =       .93753E-01
THE LEAST SQUARES VALUE IS =      .46916E+00

THE GRADIENT IS     -.14019E-09    -.25256E-09    -.23803E-08
```

4. MINIMIZING THE SUM OF AN INDEXED FUNCTION (OPTION 2)

The general structure of the input file for minimizing the sum of an indexed function is explained in the following. This option is triggered by placing the word MAXL in columns 1-4 of the first line in the input file. The main difference between the input format of this option and the least squares option is that on line three the "=" sign and name of the dependent variable do not appear. The user merely writes on line 3 (and for as many following lines as needed) the indexed function whose sum is to be minimized. The remainder of the input file is exactly that described in Section 3.

4.1 MAXIMUM LIKELIHOOD ESTIMATION

Another common method of finding estimates of parameter values that explain observed data is the method of maximum likelihood estimation. Let $b = (b_1, \ldots, b_r)$ be r unknown parameters of a frequency function $g_o(y,b)$ of the random variable y. Let $y_1, \ldots, y_m$ be m observations of y. The likelihood function associated with these observations and frequency function is

$$L(y_1,\ldots,y_m,b_1,\ldots,b_r) = g_o(y_1,b)g_o(y_2,b)\ldots g_o(y_m,b). \tag{4.1}$$

An important method of estimating the values of $b_1,\ldots,b_r$ based on the observed $y_1,\ldots,y_m$ is to maximize the likelihood function (4.1). The $b_1,\ldots,b_r$ that maximize (4.1) are called maximum likelihood estimators and have many desirable statistical properties. The reader is referred to Cramer [??] for a fuller discussion.

The problem of maximizing $L(y_1,\ldots, y_m, b)$ is an unconstrained mathematical programming problem. Since the logarithm of the likelihood function achieves its maximum at the same b as the likelihood function itself, the general problem of likelihood estimation is stated as follows.

Find values $(\bar{b}_1, \ldots, \bar{b}_r)$ that

$$\text{maximize} \sum_{i=1}^{r} \ln g_o(y_i,b), \tag{4.2}$$

or

$$\text{minimize} -\sum_{i=1}^{r} \ln g_o(y_i,b), \tag{4.3}$$

The example for this section comes from the world of biomedicine and is taken from [Bracken and McCormick, 1968]. The mathematical technique illustrated is the minimization of the negative of the log-likelihood function.

It is hypothesized that the population of systolic blood pressures can be separated into three separate groups. The distribution of blood pressures within each of these groups can be represented by a normal frequency function. Let p_1, p_2, and p_3 represent the proportions of the population in each of the three groups. Let (μ_1, σ_1), (μ_2, σ_2), and (μ_3, σ_3) be the means and standard deviations of the normal frequency functions corresponding to each group. These nine values correspond to the unknown $\{b_j\}$ parameters.

Then under these assumptions the frequency function for the random variable y, which denotes systolic blood pressure, is obtained by summing the frequency functions of the individual groups times their probability of occurrence to yield

$$\frac{1}{\sqrt{2\pi}} \sum_{k=1}^{3} \frac{p_3}{\sigma_k} \exp\left[- \frac{(y - \mu_k)^2}{2\sigma_k^2} \right]$$

where

$$p_1 + p_2 + p_3 = 1.$$

There are eight parameters in this frequency function since one proportion, or probability, can be eliminated. Let $p_3 = 1 - p_1 - p_2$. Let n_i equal the frequency of occurence of the i^{th} observation. The problem then is to find values of $(p_1, p_2, \mu_1, \mu_2, \mu_3, \sigma_1, \sigma_2, \sigma_3)$ that

$$\text{minimize} \sum_{i=1}^{m} -n_i \ln \left(\frac{1}{\sqrt{2\pi}} \left\{ \frac{p_1}{\sigma_1} \exp\left[- \frac{(y_i - \mu_1)^2}{2\sigma_1^2} \right] + \frac{p_2}{\sigma_2} \exp\left[- \frac{(y_i - \mu_2)^2}{2\sigma_2^2} \right] \right.\right.$$

$$\left.\left. + \frac{1 - p_1 - p_2}{\sigma_3} \exp\left[- \frac{(y_i - \mu_3)^2}{2\sigma_3^2} \right] \right\} \right)$$

The data for this are given in Table 4. Using the FACTUNC Systems yields estimates. (See Appendix A)

$\bar{p}_1 = .255$	$\bar{\mu}_1 = 128.8$	$\bar{\sigma}_1 = 10.3$
$\bar{p}_2 = .594$	$\bar{\mu}_2 = 158.3$	$\bar{\sigma}_2 = 21.7$
$\bar{p}_3 = .151$	$\bar{\mu}3 = 222.2$	$\bar{\sigma}_3 = 18.5$

The input file for this problem is given in Figure 4. Abbreviated output is given in the Appendix A.

TABLE 4. SYSTOLIC BLOOD PRESSURE VALUES WITH FREQUENCY OF OCCURRENCE

Systolic Blood Pressure	Frequency of Occurrence	Systolic Blood Pressure	Frequency of Occurrence	Systolic Blood Pressure	Frequency of Occurrence
95	1	150	17	200	3
105	1	155	4	205	3
110	4	160	20	210	8
115	4	165	8	215	1
120	15	170	17	220	6
125	15	175	8	225	0
130	15	180	6	230	5
135	13	185	6	235	1
140	21	190	7	240	7
145	12	195	4	245	1
				260	2

FIGURE 4. INPUT FILE FOR SYSTOLIC BLOOD PRESSURE PARAMETERS ESTIMATION.

```
MAXL-FIND PARAMETERS DEFINING FREQUENCY FUNCTION MIN -(LOG LIKE FN)
   FREQUENCY FUNCTION IS WEIGHTED SUM OF THREE NORMALS
   (-FREQ)*LN((P/SIG)*NORDEN((Y-MU)/SIG)+(((1.-P-Q)/SIG2)*
NORDEN((Y-MU2)/SIG2))+(Q/SIG3)*NORDEN((Y-MU3)/SIG3))$
NUMBER OF UNKNOWN PARAMETERS,CONSTANTS,IND. VARS. AND OBSERVATIONS
   8 0 2 31
PARAMETERS: STARTING VALUE, LOWER AND UPPER BOUND
   'MU'   133.2425  120.   200.
   'SIG' 9.47758  8.   70.
   'P'  .28802    .2 .9
   'MU2' 171.10458   141.    230.
   'SIG2' 33.76752  10.   100.
   'Q'  .2  .1  .9
   'MU3'  200.  100. 300.
   'SIG3' 30.   10.   50.
CONSTANTS ( NONE FOR THIS PROBLEM)
INDEXED DATA
   'Y'   'FREQ'
    95.    1.
   105.    1.
   110.    4.
   115.    4.
   120.   15.
   125.   15.
   130.   15.
   135.   13.
   140.   21.
   145.   12.
   150.   17.
   155.    4.
   160.   20.
   165.    8.
   170.   17.
   175.    8.
   180.    6.
   185.    6.
   190.    7.
   195.    4.
   200.    3.
   205.    3.
   210.    8.
   215.    1.
   220.    6.
   225.    0.
   230.    5.
   235.    1.
   240.    7.
   245.    1.
   260.    2.
```

5. MINIMIZING A GENERAL UNCONSTRAINED FUNCTION (OPTION 3)

This option is triggered by placing UNCO in the first four positions of the first line of input. The main difference between the input format of this and that for an indexed function (See Section 4) is that there is no data defining the index at the end of the input.

This option will be illustrated after a discussion on how to solve constrained problems by solving a sequence of unconstrained problems.

5.1 SOLVING CONSTRAINED PROBLEMS VIA UNCONSTRAINED OPTIMIZATION

Although FACTNLS is primarily for unconstrained problems, it is possible to use it as an aid for solving constrained optimization problems. There is a well-known algorithm, SUMT, which solves constrained problems by solving a sequence of unconstrained problems. For more information the reader is referred to [Fiacco and McCormick 1968] or [McCormick 1983, Chapter 16]. A brief description of the SUMT algorithm follows.

The optimization problem is assumed to be in the form:

$$\begin{aligned} &\text{minimize } f(x) \\ \text{s.t.}\quad &g_i(x) \geq 0, \text{ for } i=1, \ldots, m, \\ &h_j(x) = 0, \text{ for } j=1, \ldots, p. \end{aligned}$$

The vector $x = (x_1, \ldots, x_n)^T$. Any optimization problem can be put into the above form. Denote $R^\circ = \{x: \; g_i(x) > 0, \text{ for } i=1, \ldots, m\}$. The SUMT algorithm solves the problem

$$\underset{x \epsilon R^\circ}{\text{minimize}}\; P(x,r_k) = f(x) - r_k \sum_{i=1}^{m} \ln[g_i(x)] + \sum_{j=1}^{p} h_j^2(x)/r_k$$

for a sequence of values $\{r_k\}$ which decrease strictly to zero. If $x(r_k)$ denotes a solution to the problem for r_k, the starting point for the new problem with r_{k+1} is $x(r_k)$. For r_k small, $x(r_k)$ is an approximation to the solution of the original constrained nonlinear programming problem. There is an extrapolation method which can be used to estimate the solution accurately using two vectors $x(r_k)$ and $x(r_{k+1})$ on the trajectory of unconstrained minimizers. The chemical equilibrium problem will be used to explain this.

The example illustrating this option is to solve a constrained problem by solving a sequence of unconstrained problems. The theory behind this approach is given after a description of the optimization problem: CHEMICAL EQUILIBRIUM. The example and data are taken from Bracken and McCormick [1968].

The problem of determining the chemical composition of a complex mixture under chemical equilibrium conditions has long been of interest. Such problems arise in the analysis of the performance of fuels and propellants and in the synthesis of complex organic compounds.

A mixture of chemical species held at a constant temperature and pressure reaches its chemical equilibrium state concurrently with reduction of the free energy of the mixture to a minimum. This is a consequence of the second law of thermodynamics. The objective function to be minimized in the chemical equilibrium model is the expression of the free energy of the chemical mixture under study. The value of the free energy of the mixture is minimized subject to the chemical reactions possible between species of the mixture.

White, Johnson, and Dantzig formulated the chemical equilibrium problem as a mathematical programming problem with linear mass balance constraints representing the possible chemical combinations of the chemical species of the mixture, and a nonlinear objective function representing the free energy of the mixture (to be minimized). They investigated steepest descent and piecewise linear programming approaches to formulating the problem. In a second paper, they explored the piecewise linear programming problem further. The problem is discussed briefly by Dantzig, who used it to illustrate the method of generalized linear programming.

Consider a mixture of m chemical elements. It has been predetermined that the m different types of atoms can combine chemically to produce n compounds, where the monotonic atom is regarded for our purpose as a possible compound. Define

x_j = the number of moles of compound j in the mixture at equilibrium,

$\bar{x}$ = the total number of moles in the mixture, where $\bar{x} = \sum_{j=1}^{n} x_j$,

a_{ij} = the number of atoms of element i in a molecule of compound j,

b_i = the number of atomic weights of element i in the mixture.

The mass balance relationships that must hold among the m elements are

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i=1, \ldots, m. \tag{5.1}$$

and

$$x_j \geq 0, \quad j=1, \ldots, n. \tag{5.2}$$

Determination of the composition of the mixture at equilibrium is equivalent to determination of the values of x_j (j=1,...,n) that satisfy (5.1) and (5.2) and also minimize the total free energy of the mixture. The total free energy of the mixture is given by

$$\sum_{j=1}^{n} x_j \; [c_j + \ln(x_j/\bar{x})] \tag{5.3}$$

where

$$c_j = (F_j^\circ/RT) + \ln P$$

where (F_j°/RT_j) is the modal standard (Gibbs) free energy function for the jth compound, which may be found in tables, and P is the total pressure in atmospheres.

Thus the nonlinear programming problem is as follows. Choose x_j $(j=1,\ldots,n)$ to minimize the nonlinear objective function (5.3) subject to linear constraints (5.1) and non-negativity restrictions (5.2).

We consider the example problem formulated and solved by White, Johnson, and Dantzig [??]. We solve the nonlinear programming problem by the sequential unconstrained minimization technique and obtain similar answers.

The problem considered is the determination of the equilibrium composition resulting from subjecting the compound $1/2N_2H_4 + 1/2O_2$ to a temperature of 3500°K and a pressure of 750 psi. In Table 5 we show for each compound j of 10 possible compounds (where the monotonic atoms are termed compounds) the Gibbs free energy function $(F^\circ/RT)_j$, the computed value of c_j for P = 750 psi, and the number atoms of H, N, and O per molecule. The number of atomic weights of H, N, and O in the mixture are $b_1=2$, $b_2=1$, and $b_3=1$.

Formulating the nonlinear programming model, the nonlinear objective function to be minimized is

$$\begin{aligned} & x_1\ [\ -6.089 + \ln(x_1/\bar{x})] \\ + & \ldots \\ + & x_{10}\ [\ -22.179 + \ln(x_{10}/\bar{x})] \end{aligned}$$

and the linear constraints of the nonlinear programming problem are as follows:

$$\begin{aligned} x_1 + 2x_2 + 2x_3 + x_6 + x_{10} &= 2, \\ x_4 + 2x_5 + x_6 + x_7 &= 1, \\ x_3 + x_7 + x_8 + 2x_9 + x_{10} &= 1, \\ x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_{10} &\geq 0. \end{aligned}$$

Solving the above nonlinear programming problem, we obtain the values of x_j $(j=1,\ldots, 10)$, the number of moles of the 10 compounds present in the equilibrium mixture, which are given in Table 5. These values agree with those obtained in [??]. The corresponding value of the objective function is -47.76.

TABLE 5. DATA ON $1/2N_2H_4 + 1/2O_2$ AT 3500°K, 750 psi.

					a_{ij}	
				i=1	i=2	i=3
j	Compound	$(F°/RT)_j$	c_j	H	N	O
1	H	-10.021	- 6.089	1	0	0
2	H_2	-21.096	-17.164	2	0	0
3	H_2O	-37.986	-34.054	2	0	1
4	N	- 9.846	- 5.914	0	1	0
5	N_2	-28.653	-24.721	0	2	0
6	NH	-18.918	-14.986	1	1	0
7	NO	-28.032	-24.100	0	1	1
8	O	-14.640	-10.708	0	0	1
9	O_2	-30.594	-26.662	0	0	2
10	OH	-26.111	-22.179	1	0	1

TABLE 6.
COMPOSITION OF $1/2N_2H_4 + 1/2O_2$
AT 3500°K, 750 PSI

j	Compound	x_j
1	H	.0407
2	H_2	.1477
3	H_2O	.7831
4	N	.0014
5	N_2	.4853
6	NH	.0007
7	NO	.0274
8	O	.0180
9	O_2	.0373
10	OH	.0969

In Figure 7 is the input file for the chemical equilibrium problem with an initial value of r_k = .01. In Appendix A is a portion of the output. The final values at the unconstrained minimizer are then used in the input file shown in Figure 8, which has a value of "RK" = .001. The solution of this second unconstrained problem is given in Figure 10. With these two vectors,

FIGURE 7. INPUT FILE FOR CHEMICAL EQUILIBRIUM PROBLEM WITH r_k=.01.

```
UNCO--CHEMICAL EQUILIBRIUM PROBLEM USING BARRIER FUNCTION
SOURCE- PP 253-254 OF NONLINEAR PROGRAMMING BY GPMCC
      C1*H+C2*H2+C3*H2O+C4*N+C5*N2+C6*NH+C7*NO+C8*O+
      C9*O2+C10*OH +RK*(BARLN(H)+BARLN(H2)+BARLN(H2O)
      +BARLN(N) + BARLN(N2) + BARLN(NH) + BARLN(NO) + BARLN(O)
      +BARLN(O2) +BARLN(OH) ) + ((H+2.*H2+2.*H2O+NH+OH-B1)**2
      +(N +2.*N2 +NH +NO-B2)**2 +(H2O+NO+O+2.*O2+OH-B3)**2)/RK
      -ENTROP(H+H2+H2O+N+N2+NH+NO+O+O2+OH) + ENTROP(H) + ENTROP(H2)
      + ENTROP(H2O) + ENTROP(N) + ENTROP(N2) + ENTROP(NH) + ENTROP(NO)
      + ENTROP(O) + ENTROP(O2)+ ENTROP(OH) $
NUMBER OF UNKNOWNS, CONSTANTS, QUANTITIES, AND OBSERVATIONS
      10 14 0 0
STARTING VALUES FOR UNKNOWNS
      'H'  .1 .00001     10.
      'H2' .1   0.00001   10.
      'H2O' 0.1  0.00001   10.
      'N'   .1  0.00001   10.
      'N2'  0.1  0.00001   10.
      'NH'  .1  0.00001   10.
      'NO'  .1  0.00001   10.
      'O'   .1  0.00001   10.
      'O2'  0.1  0.00001   10.
      'OH'  .1  0.00001   10.
CONSTANTS
      'C1' -6.089
      'C2' -17.164
      'C3' -34.054
      'C4' -5.914
      'C5' -24.721
      'C6' -14.986
      'C7' -24.100
      'C8' -10.708
      'C9' -26.662
      'C10' -22.179
      'B1'  2.
      'B2'  1.
      'B3'  1.
      'RK'  .01
```

FIGURE 8: INPUT FILE FOR CHEMICAL EQUILIBRIUM PROBLEM WITH r_k=.001

```
UNCO--CHEMICAL EQUILIBRIUM PROBLEM USING BARRIER FUNCTION
SOURCE- PP 253-254 OF NONLINEAR PROGRAMMING BY GPMCC(RK=.001)
 C1*H+C2*H2+C3*H20+C4*N+C5*N2+C6*NH+C7*N0=C8*0+
        C9*02+C10*0H +RK*(BARLN(H)+BARLN(H2)+BARLN(H20)
       +BARLN(N) + BARLN(N2)+BARLN(NH)+BARLN(N0)+BARLN(0)
       +BARLN(02) +BARLN(0H) )+((H+2. *H2+2. *H20+NH+0H-B1)**2
       +(N +2. *N2 +NH +N0-B2)**2 +(H20+N0+0+2. *02+0h-B3)**2)/RK
  -ENTROP(H+H2+H20+N+N+N2+NH+N0+0+02+0H)
 + ENTROP(H)+ENTROP(N2)+ENTROP(h20)+ENTROP(N)+ENTROP(N2)+ENTROP(NH)+ENTROP
 ENTROP(0) +ENTROP(02)+ ENTROP(0H) $
NUMBER OF UNKNOWNS, CONSTANTS, QUANTITIES, AND OBSERVATIONS
  10 14 0 0
STARTING VALUES FOR UNKNOWNS
 'H'   .049942 .00001    10.
 'H2'  .15007   0.00001  10.
 'H20'  .79116  0.00001  10.
 'N'     .0066461   0.00001   10.
 'N2'    .50722   0.00001    10.
 'NH'    .0050182   0.00001   10.
 'N0'    .038836   0.00001    10.
 '0'     .02846   0.00001    10.
 '02'    .052818   0.00001    10.
 '0H'    .11177   0.00001    10.
CONSTANTS
 'C1'  -6.089
 'C2'  -17.164
 'C3'  -34.054
 'C4'  -5.914
 'C5'  -24.721
 'C6'  -14.986
 'C7'  -24.100
 'C8'  -10.708
 'C9'  -26.662
 'C10'  -22.179
 'B1'  2.
 'B2'  1.
 'B3'  1.
 'RK'  .001
VARIABLES
```

the solution can be approximated using extrapolation theory. The two point extrapolation is

$$x^*(est) = [cx(r_{k+1}) - x(r_k)]/(c - 1)$$

where

$$c = r_k/r_{k+1}.$$

In Table 7, the extrapolation is performed using the output from the two unconstrained minimizations.

Eventually, solving constrained problems will be an automatic procedure, For now, with care, the FACTUNC program can be used to find solutions to constrained problems.

When implementing the SUMT approach, it is important to use the single argument function BARLN(X). This prevents the sequence of iterates from straying from R^o, the interior of the feasible region. The use of LN(X) will result in disaster. The user must also be sure that the initial starting point is in the strict interior of the feasible region.

Choosing the r_k and r_{k+1} is still more of and art than a science. Experience is often a help in doing this. A large value to start with is usually better then a small one, which creates a number of difficult unconstrained minimization problems.

6. FUTURE WORK

The system described in this manual is just the first step in a proposed series of optimization programs. One basic addition that is envisioned is the implementation of a general indexing capability based on modern data base concepts. Allowing "subscripted" data will extend the applicability of the system to large problems without a concomitant increase in program size.

The second generalization is to the solution of constrained optimization problems. Although this can be accomplished (as indicated in subsection 5.1) using a barrier function approach, it is important for a user to specify in a user friendly way inequality and equality constraints on the problem unknowns.

A third extension will be to allow the use of this system by algorithmists to test their methodology. The system can be used to represent the problems and to compute automatically the derivatives required by the algorithms.

TABLE 7.
EXTRAPOLATION FOR CHEMICAL EQUILIBRIUM PROBLEM
USING VALUES FROM TWO SUMT MINIMIZERS

Compound	Solution for r_k=.01	Solution for r_k=.001	Extrapo-lation	True Solution
H	.049942	.041726	.0408	.0407
H_2	.15007	.14829	.1481	.1477
H_2O	.79116	.78409	.7833	.7831
N	.0066461	.0022263	.0017	.0014
N_2	.50722	.48779	.4856	.4853
NH	.0050182	.00008635	-.0005	.0007
NO	.038836	.028593	.0275	.0274
O	.028460	.019089	.0180	.0180
O_2	.052818	.038759	.0372	.0373
OH	.11177	.098314	.0968	.0969
Free Energy	-48.702	-47.850	-47.755	-47.76

7. REFERENCES

Bracken, J. and McCormick, G.P. (1968), Selected Applications of Nonlinear Programming, Wiley, New York.

Cramer, H. (1955), The Elements of Probability Theory, Wiley, New York.

Dantzig, G., Johnson, S., and White, W. (1958), A linear programming approach to the chemical equilibrium problem, Management Science, 5, 38-43.

Fiacco, A.V. and McCormick, G.P. (1968), Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York.

Jackson, R.H.F. and McCormick, G.P. (1986), The polyadic structure of factorable function tensors with applications to high-order minimization techniques, J.O.T.A., 51, 63-94.

McCormick, G.P. (1983), Nonlinear Programming: Theory, Algorithms and Applications, Wiley, New York.

Sofer, A. (1983), Computationally Efficient Techniques for Generalized Inversion, Dissertation, Department of Operations Research, The George Washington University, Washingtón, D.C.

White, W.B., Johnson, S.H. and Dantzig, G.B. (1958), Chemical equilibrium in complex mixtures, J. Chem. Phys., 28, 751-755.

APPENDIX A:

ABBREVIATED OUTPUT FOR MAXL OPTION EXAMPLE.

```
MAXL-FIND PARAMETERS DEFINING FREQUENCY FUNCTION MIN -(LOG LIKE FN)
MINIMIZING INDEXED SUM
HESSIAN FORMED EXPLICITLY

FREQUENCE FUNCTION IS WEIGHTED SUM OF THREE NORMALS
 (-FREQ)*LN((P/SIG)*NORDEN((Y-MU)/SIG)+(((1.-P-Q)/SIG2)*
   NORDEN((Y-MU2)/SIG2))+(Q/SIG3)*NORDEN((Y-MU3)/SIG3))$
SINGLE CHARACTER OUTPUT STRING OF (FUNCTION) EQUATION
(-FREQ)*LN((P/SIG)*NORDEN((Y-MU)/SIG)+(((1.-P-Q)/S
IG2)*NORDEN((Y-MU2)/SIG2))+(Q/SIG3)*NORDEN((Y-MU3)
/SIG3))$

NUMBER OF UNKNOWN PARAMETERS,CONSTANTS,DATA COLS., OBSERVATIONS
THE NUMBER OF UNKNOWNS IS =  8
THE NUMBER OF SYMBOLIC CONSTANTS IN THE EQUATION (FUNCTION) =  0
THE NUMBER OF DATA COLUMNS =   2
THE NUMBER OF DATA OBSERVATIONS =  31

PARAMETERS: STARTING VALUE, LOWER AND UPPER BOUND
         MU          .133243E+03    .120000E+03    .200000E+03
         SIG         .947758E+01    .800000E+01    .700000E+02
         P           .288020E+00    .200000E+00    .900000E+00
         MU2         .171105E+03    .141000E+03    .230000E+03
         SIG2        .337675E+02    .100000E+02    .100000E+03
         Q           .200000E+00    .100000E+00    .900000E+00
         MU3         .200000E+03    .100000E+03    .300000E+03
         SIG3        .300000E+02    .100000E+02    .500000E+02

CONSTANTS ( NONE FOR THIS PROBLEM)

INDEXED DATA
      Y        FREQ
   95.00000     1.00000
  105.00000     1.00000
  110.00000     4.00000
  115.00000     4.00000
  120.00000    15.00000
  125.00000    15.00000
  130.00000    15.00000
  135.00000    13.00000
  140.00000    21.00000
  145.00000    12.00000
  150.00000    17.00000
  155.00000     4.00000
  160.00000    20.00000
  165.00000     8.00000
  170.00000    17.00000
```

```
    175.00000      8.00000
    180.00000      6.00000
    185.00000      6.00000
    190.00000      7.00000
    195.00000      4.00000
    200.00000      3.00000
    205.00000      3.00000
    210.00000      8.00000
    215.00000      1.00000
    220.00000      6.00000
    225.00000       .00000
    230.00000      5.00000
    235.00000      1.00000
    240.00000      7.00000
    245.00000      1.00000
    260.00000      2.00000

THE OPTIMAL VALUES OF THE UNKNOWNS ARE
  1 MU      =      .12878E+03
  2 SIG     =      .10297E+02
  3 P       =      .25475E+00
  4 MU2     =      .15826E+03
  5 SIG2    =      .21741E+02
  6 Q       =      .15123E+00
  7 MU3     =      .22224E+03
  8 SIG3    =      .18505E+02

THE LEAST SQUARES VALUE IS =      .11384E+04

 THE GRADIENT IS     -.37851E-13    -.60924E-13    -.60966E-11    -.43141E-13
 -.10113E-13

 THE GRADIENT IS     -.76739E-11     .67502E-13    -.20073E-12
```

APPENDIX B:

ABBREVIATED OUTPUT FOR CHEMICAL EQUILIBRIUM PROBLEM (r_k=.01)

```
 UNCO--CHEMICAL EQUILIBRIUM PROBLEM USING BARRIER FUNCTION
 UNCONSTRAINED PROBLEM
 HESSIAN FORMED EXPLICITLY

 SOURCE- PP 253-254 OF NONLINEAR PROGRAMMING BY GPMCC
  C1*H+C2*H2+C3*H2O+C4*N+C5*N2+C6*NH+C7*NO+C8*O+
       C9*O2+C10*OH +RK*(BARLN(H)+BARLN(H2)+BARLN(H2O)
      +BARLN(N) + BARLN(N2)+BARLN(NH)+BARLN(NO)+BARLN(O)
      +BARLN(O2) +BARLN(OH) )+((H+2.*H2+2.*H2O+NH+OH-B1)**2
      +(N +2.*N2 +NH +NO-B2)**2 +(H2O+NO+O+2.*O2+OH-B3)**2)/RK
   -ENTROP(H+H2+H2O+N+N2+NH+NO+O+O2+OH)
  +
ENTROP(H)+ENTROP(H2)+ENTROP(H2O)+ENTROP(N)+ENTROP(N2)+ENTROP(NH)+ENTROP(NO)+
  ENTROP(O) + ENTROP(O2)+ ENTROP(OH) $
 SINGLE CHARACTER OUTPUT STRING OF (FUNCTION) EQUATION
 C1*H+C2*H2+C3*H2O+C4*N+C5*N2+C6*NH+C7*NO+C8*O+C9*O
 2+C10*OH+RK*(BARLN(H)+BARLN(H2)+BARLN(H2O)+BARLN(N
 )+BARLN(N2)+BARLN(NH)+BARLN(NO)+BARLN(O)+BARLN(O2)
 +BARLN(OH))+((H+2.*H2+2.*H2O+NH+OH-B1)**2+(N+2.*N2
 +NH+NO-B2)**2+(H2O+NO+O+2.*O2+OH-B3)**2)/RK-ENTROP
 (H+H2+H2O+N+N2+NH+NO+O+O2+OH)+ENTROP(H)+ENTROP(H2)
 +ENTROP(H2O)+ENTROP(N)+ENTROP(N2)+ENTROP(NH)+ENTRO
 P(NO)+ENTROP(O)+ENTROP(O2)+ENTROP(OH)$

 NUMBER OF UNKNOWNS, CONSTANTS, QUANTITIES, AND OBSERVATIONS
 THE NUMBER OF UNKNOWNS IS = 10
 THE NUMBER OF SYMBOLIC CONSTANTS IN THE EQUATION (FUNCTION) = 14
 THE NUMBER OF INDEPENDENT VARIABLES =   0
 THE NUMBER OF DATA OBSERVATIONS =   0

 STARTING VALUES FOR UNKNOWNS
          H          .100000E+00    .100000E-04    .100000E+02
          H2         .100000E+00    .100000E-04    .100000E+02
          H2O        .100000E+00    .100000E-04    .100000E+02
          N          .100000E+00    .100000E-04    .100000E+02
          N2         .100000E+00    .100000E-04    .100000E+02
          NH         .100000E+00    .100000E-04    .100000E+02
          NO         .100000E+00    .100000E-04    .100000E+02
          O          .100000E+00    .100000E-04    .100000E+02
          O2         .100000E+00    .100000E-04    .100000E+02
          OH         .100000E+00    .100000E-04    .100000E+02

 CONSTANTS
 C1    -.608900E+01
 C2    -.171640E+02
 C3    -.340540E+02
 C4    -.591400E+01
```

```
C5     -.247210E+02
C6     -.149860E+02
C7     -.241000E+02
C8     -.107080E+02
C9     -.266620E+02
C10    -.221790E+02
B1      .200000E+01
B2      .100000E+01
B3      .100000E+01
RK      .100000E-01

VARIABLES

THE OPTIMAL VALUES OF THE UNKNOWNS ARE
  1 H       =       .49942E-01
  2 H2      =       .15007E+00
  3 H2O     =       .79116E+00
  4 N       =       .66461E-02
  5 N2      =       .50722E+00
  6 NH      =       .50182E-02
  7 NO      =       .38836E-01
  8 O       =       .28460E-01
  9 O2      =       .52818E-01
 10 OH      =       .11177E+00

THE LEAST SQUARES VALUE IS =       -.48702E+02

 THE GRADIENT IS      -.22860E-07     -.28291E-07     -.22719E-08     -.10537E-03
 -.94860E-09

 THE GRADIENT IS      -.67860E-05     -.49977E-09     -.20685E-09     -.51097E-07
 -.49182E-07
```

APPENDIX C:

PARTIAL OUTPUT FOR CHEMICAL EQUILIBRIUM (r_k= .001)

```
UNCO--CHEMICAL EQUILIBRIUM PROBLEM USING BARRIER FUNCTION
UNCONSTRAINED PROBLEM
HESSIAN FORMED EXPLICITYLY

SOURCE- PP 253-254 OF NONLINEAR PROGRAMMING BY GPMCC (RK=.001)
 C1*H+C2*H2+C3*H2O+C4*N+C5*N2+C6*NH+C7*NO+C8*O+
      C9*O2+C10*OH +RK*(BARLN(H)+BARLN(H2)+BARLN(H2O)
     +BARLN(N) + BARLN(N2)+BARLN(NH)+BARLN(NO)+BARLN(O)
     +BARLN(O2) +BARLN(OH) )+((H+2. *H2+2. *H2O+NH+OH-B1)**2
     +(N +2. *N2 +NH +NO-B2)**2 +(H2O+NO+O+2. *O2+OH-B3)**2)/RK
  -ENTROP(H+H2+H2O+N+N2+NH+NO+O+O2+OH)
+ENTROP(H)+ENTROP(H2)+ENTROP(H2O)+ENTROP(N)+ENTROP(N2)+ENTROP(N2)+ENTROP(NH)
+ENTROP(NO)+ENTROP (O) + ENTROP (O2)+ ENTROP(OH) $
SINGLE CHARACTER OUTPUT STRING OF (FUNCTION) EQUATION
C1*H+C2*H2+C3*H2O+C4*N+C5*N2+C6*NH+C7*NO+C8*O+C9*O
2+C10*OH+RK*(BARLN(H)+BARLN(H2)+BARLN(H2O)+BARLN(N
)+BARLN(N2)+BARLN(NH)+BARLN(NO)+BARLN(O)+BARLN(O2)
+BARLN(OH))+((H+2.*H2+2.*H2O+NH+OH-B1)**2+(N+2.*N2
+NH+NO-B2)**2+(H2O+NO+O+2. *O2+OH-B3)**2)/RK-ENTROP
(H+H2+H2O+N+N2+NH+NO+O+O2+OH)+ENTROP(H)+ENTROP(H2)
+ENTROP(H2O)+ENTROP(N)+ENTROP(N2)+ENTROP(NH)+ENTROP
(NO)+ENTROP(O)+ENTROP(O2)+ENTROP(OH)$

NUMBER OF UNKNOWNS, CONSTANTS, QUANTITIES, AND OBSERVATIONS
THE NUMBER OF UNKNOWNS IS = 10
THE NUMBER OF SYMBOLIC CONSTANTS IN THE EQUATION (FUNCTION) = 14
 THE NUMBER OF INDEPENDENT VARIABLES =  0
 THE NUMBER OF DATA OBSERVATIONS = 0

 STARTING VALUES FOR UNKNOWNS
    H       .499420E-01           .100000E-04     .100000E+02
    H2      .150070E+00           .100000E-04     .100000E+02
    H2O     .791160E+00           .100000E-04     .100000E+02
    N       .664610E-02           .100000E-04     .100000E+02
    N2      .507220E+00           .100000E-04     .100000E+02
    NH      .501820E-02           .100000E-04     .100000E+02
    NO      .388360E-01           .100000E-04     .100000E+02
    O       .284600E-01           .100000E-04     .100000E+02
    O2      .528180E-01           .100000E-04     .100000E+02
    OH      .111770E+oo           .100000E-04     .100000E+02

CONSTANTS
C1  -.608900E+01
C2  -.171640E+02
C3  -.340540E+02
C4  -.591400E+01
C5  -.247210E+02
C6  -.149860E+02
```

```
C7  -.241000E+02
C8  -.107080E+02
C9  -.266620E+02
C10 -.221790E+02
B1   .200000E+01
B2   .100000E+01
B3   .100000E+01
RK   .100000E-02

THE OPTIMAL VALUES OF THE UNKNOWNS ARE

THE CURRENT VALUES OF THE UNKNOWS ARE
  1 H      =         .41726E-01
  2 H2     =         .14829E+00
  3 H2O    =         .78409E+00
  4 N      =         .22263E-02
  5 N2     =         .48779E+00
  6 NH     =         .86349E-04
  7 NO     =         .28593E-01
  8 O      =         .19089E-01
  9 O2     =         .38759E-01
 10 OH     =         .98314E-01

THE LEAST SQUARES VALUE IS  =       -.47850E+02

 THE GRADIENT IS     -.20583E-09    -.21088E-08     -.17735E-10  -.20692

 THE GRADIENT IS     -.13664E+02    -.10834E-09     -.24742E-08  -.68331
```

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET *(See instructions)*	1. PUBLICATION OR REPORT NO. NISTIR 89-4159	2. Performing Organ. Report No.	3. Publication Date SEPTEMBER 1989

4. TITLE AND SUBTITLE

FACTUNC: A USER-FRIENDLY SYSTEM FOR UNCONSTRAINED OPTIMIZATION

5. AUTHOR(S)

Richard H.F. Jackson; Garth P. McCormick; Ariela Sofer

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

10. SUPPLEMENTARY NOTES

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

FACTUNC is a system for solving unconstrained minimization problems based on the concept of factorable programming. This concept enables the user to provide the problem function and data in a user friendly way and does not require user-supplied derivatives. The system utilizes the factorable function concept to obtain the first and second derivatives required for unconstrained optimization. In all cases derivatives are obtained rapidly and accurately (up to roundoff errors due to machine precision), as opposed to finite differencing.

As a system for nonlinear minimization, FACTUNC allows several options. First the user can solve regression (nonlinear least squares) problems by providing the regression equation and the data for the dependent and independent variables. The second option allows for the minimization of the sum of an indexed function. The user provides the function, and the indexed data. This can be used for example to solve maximum likelihood estimation problems when the user provides the negative of the (weighted) log of the frequency function and the data. The third option is simply to minimize a function supplied by the user. Utilizing barrier function methodology, this third option can sometimes be used to solve constrained problems.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

FACTUNC; machine; optimization; precision; programming; regression

13. AVAILABILITY

☒ Unlimited
□ For Official Distribution. Do Not Release to NTIS
□ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

36

15. Price

$12.95