JOURNAL OF RESEARCH of the National Bureau of Standards — B. Mathematical Sciences
Vol. 78B. No. 4, October–December 1974

Automatic Computing Methods for Special Functions. Part II. The Exponential Integral $E_n(x)$

Irene A. Stegun and Ruth Zucker

Institute for Basic Standards, National Bureau of Standards, Washington, D.C. 20234

(September 4, 1974)

Accurate, automatic, efficient methods for computing the exponential integral $E_n(x)$ are detailed and implemented in an American National Standard FORTRAN program. The driver program and test results are also included.

Key words: Computer programs; continued fraction; exponential integral; key values; recurrence relation.

1. Introduction

The exponential integral itself occurs in many physical problems and often many other integrals are expressible in terms of the exponential integral. In view of its importance and the difficulties accompanying the repeated application of its recurrence relation, we have chosen this integral as Part II. (For Part I, see J. of Research NBS, Vol. **74B**, July-September 1970, pp. 211–224.)

Accurate, efficient, automatic computing methods implemented in American National Standard FORTRAN, covering the entire range of machine legitimate arguments and/or functional values will be supplied. The number of terms in series, the number of convergents in an iterative process, etc., are all determined by the computer as a function of word length, arguments, the accuracy desired, etc. In cases of error returns, more realistic results will be returned. To further ensure correct limiting values of related functions, the proper analytic behavior of the function will always be retained. The driver program and test results are also included.

2. Mathematical Formulas

Relevant formulas are collected here for completeness and ease of reference. In keeping with the convention of the Handbook [1], [1] x here is a real variable.

A. Definition

$$
\begin{aligned}
E_n(x) &= \int_1^\infty t^{-n} e^{-xt}\,dt \qquad (n=0,1,2,\ldots;\ x>0)\\
&= \int_0^1 u^{n-2} e^{-x/u}\,du\\
&= x^{n-1}\int_x^\infty t^{-n} e^{-t}\,dt.
\end{aligned}
$$

AMS Subject classification: 33–04, 65D20.

[1] Figures in brackets indicate references on page 205.

B. Series Expansion

$$E_n(x) = \frac{(-x)^{n-1}}{(n-1)!}[-\ln x + \Psi(n)] - \sum_{\substack{m=0 \\ m\neq n-1}}^{\infty}{}' \frac{(-x)^m}{m!\,(m-n+1)} \qquad (x > 0)$$

$$\Psi(1) = -\gamma,\ \Psi(n) = -\gamma + \sum_{m=1}^{n-1} \frac{1}{m} \qquad (n > 1)$$

γ(Euler's constant) = .57721 56649. . ..

C. Continued Fraction

$$E_n(x) = e^{-x}\left(\frac{1}{x+}\,\frac{n}{1+}\,\frac{1}{x+}\,\frac{n+1}{1+}\,\frac{2}{x+}\cdots\right) \qquad (x > 0).$$

D. Asymptotic Expansion

$$E_n(x) \sim \frac{e^{-x}}{x}\left[1 - \frac{n}{x} + \frac{n(n+1)}{x^2} - \frac{n(n+1)(n+2)}{x^3} + \cdots\right].$$

E. Special Values

$$E_n(0) = \frac{1}{n-1} \qquad (n > 1)$$

$$E_0(x) = \frac{e^{-x}}{x}.$$

F. Recurrence Relation

$$E_{n+1}(x) = \frac{e^{-x}}{n} - \frac{x}{n}E_n(x) \qquad n = 1, 2, 3, \ldots.$$

G. Differentiation Formula

$$\frac{d^k}{dx^k}E_n(x) = (-1)^k E_{n-k}(x) \qquad n = 1, 2, 3, \ldots.$$

H. Inequalities

$$(x > 0,\ n = 1, 2, 3, \ldots)$$

$$\frac{n-1}{n}E_n(x) < E_{n+1}(x) < E_n(x)$$

$$\frac{1}{x+n} < e^x E_n(x) \leqslant \frac{1}{x+n-1}.$$

I. Related Functions

Incomplete Gamma Function

$\Gamma(1-n, x) = x^{1-n} E_n(x)$

Confluent Hypergeometric Function

$U(n, n, x) = e^x x^{1-n} E_n(x)$.

3. Method

Examination of the recurrence relation indicates that an independent computation must be carried out for at least one value of $n \geq 1$. The function $E_n(x)$ is always positive, for real x has no zeros, and is a slowly decreasing function with increasing n and/or x. Repeated application of the recurrence relation therefore will yield an increasing round off error even if, with more complicated scaling, both forward and backward recurrence relations are used starting at $n = [x]$. Generally, in physical applications the first few orders only are needed. Consequently, we have chosen an independent computing method valid for all orders.

The implementing American National Standard FORTRAN program has been set up in such a way as to require a minimal number of changes for varying precision, either single or double precision on the same or different computers. The program checks for positive arguments n (=RN) and x and integer n. If either is negative, an error indicator is set (IERR = 1) and an impossible value, the negative of the maximum machine value (=RINF), is returned for both functional values, $E_n(x)$ (=ENX) and $e^x E_n(x)$ (=EXPENX). If $n >$ RMAXI, the maximum integer convertible to a floating point number, n is assumed to be an integer. If $n \leq$ RMAXI, an integer test is applied to RN. If it fails the test, IERR is set equal to 2, and RINF is returned for both functional values. To assure ready portability, additional tests are performed for negative zero and allowance for round off errors due either to machine arithmetic or the system being used.

The special cases are treated independently. When $n = 0$, to avoid machine difficulty, x is tested against the reciprocal of the maximum machine value; if x equals, or is less than that number, the maximum value is returned for both functional values. If x is greater, then ENX $= e^{-x}/x$ and EXPENX $= 1/x$. When $x = 0$, and $n = 0$ or 1, ENX = EXPENX = RINF; for $n > 1$, ENX = EXPENX $= 1/(n-1)$.

Computation with the series results in a greater round-off error in the neighborhood of $x = 1$, in particular for small values of n. Even if the form of the series were changed from the present alternating form, there is still a loss due to the logarithmic term. While the continued fraction is convergent for $x > 0$, the number of terms increases rapidly as x approaches 0. The integrand is not particularly amenable to automatic numerical integration by any of the low order formulas. In general ($n \leq$ RMAXI) the most accurate and efficient methods of automatic computation are the alternating power series for $x \leq 1$(ULPS) and the "even" form of the continued fraction for $x > 1$.

For $1 \leq n \leq$ RMAXI and $0 < x \leq 1$, the power series is used in the following form

$$E_n(x) = \mathrm{ENX} = \sum_{M=0}^{K} T_M = \mathrm{SUM} + TM\,(M = 0, 1, \ldots, K)$$

where $T_M = \dfrac{-(-x)^M}{M!(M-n+1=D)} = \dfrac{\mathrm{PTERM}}{D}$ for $M \neq n-1$

and $T_M = \mathrm{PTERM}[\mathrm{LOG}(x) - \mathrm{PSI}(n)]$ for $M = n-1$.

PTERM is obtained by recurrence with PTERM(0) $= -1$ and PTERM$(M+1) =$ PTERM $(M) \cdot (-x)/M+1$.

Rather than compute a previously determined fixed number of terms, it is valid, provided $M \neq n-1$, to terminate the series when the relative error, computed as $|TM/\text{SUM}|$ is less than or equal to a prescribed tolerance (=TOLER). In the present program, $\text{TOLER}=2^{-NBM}$, where NBM is the maximum number of binary digits in the mantissa of a floating point number. It may, however, be set to the accuracy desired.

Since the series is an alternating one, SUM may equal zero at some M. However, since the function has no zeros for real x, the relative error test may be safely bypassed if SUM$=0$ and additional terms computed. In our manner of evaluating the series, TM may likewise be zero if $\log(x)=\Psi(n)$. (Values of $e^{\Psi(n)}$ for $n=1(1)10$ are given in sec. 10.) The logarithmic term enters the computation for $n \leqslant 2$ since at least two terms of the series must be used with the above relative error test. Consequently, using the power series for $x \leqslant 1$, this situation will not then be critical when $n=1$ and $x \approx 0.56$. However, since the code may be used for experimental purposes, the relative error test is not applied to the logarithmic term.

For $n \geqslant 1$ and $x > 1$, and $n > \text{RMAXI}$, $x > 0$, the continued fraction in its "even" form

$$E_n(x)=e^{-x}\left[\frac{1}{x+n-}\,\frac{1\cdot n}{x+n+2-}\,\frac{2(n+1)}{x+n+4-}\cdots\right]$$

$$=e^{-x}F=e^{-x}\text{EXPENX}$$

is evaluated in the forward direction. The first convergent $F_1/G_1=A_1/B_1$ where $A_1=1$, $A_M=-(M-1)(n+M-2)$, $B_M=x+n+2(M-1)$. If we define

$$F_{-1}=1,\ F_0=0,\ G_{-1}=0 \text{ and } G_0=1$$

then successive convergents F_M/G_M for $M=1, 2, \ldots$ may be obtained by the following recurrence relation

$$F_M=B_MF_{M-1}+A_MF_{M-2}$$

$$G_M=B_MG_{M-1}+A_MG_{M-2}.$$

Since in the basic continued fraction A_M and B_M are always positive with $A_M \geqslant 1$, F_M and G_M will always be positive and increasing with increasing M. Care must then be taken to ensure that overflow does not occur in generating the successive convergents. For the "even" form of the continued fraction, since $A_M < 0$ for $M > 1$, it is sufficient to check that G_M is always less than RINF/B_M. If G_M is greater, then F_M, F_{M-1}, G_M and G_{M-1} are all scaled by dividing by B_M.

Since, for the above continued fraction, the successive convergents form a monotonically increasing sequence, if through round-off errors,

$$1-\frac{(F_{M-1}/G_{M-1})(=\text{PREV})}{F_M/G_M} \leqslant 0 \qquad (F_M/G_M \leqslant F_{M-1}/G_{M-1})$$

then $F_{M-1}/G_{M-1}=F$.

If $0 < 1-\dfrac{F_{M-1}/G_{M-1}}{F_M/G_M} \leqslant \text{TOLER}\ (=2^{-NBM})$, then $F_M/G_M=F$.

The following table gives an indication of the number of terms needed to obtain maximum machine accuracy for particular values of NBM, x and n with the various methods of computation.

Method	Number of Terms			
	NBM = 27		NBM = 60	
$x = 1$	$n = 1$	$n = 10$	$n = 1$	$n = 10$
Power Series	13	14	21	22
Continued Fraction	26	17	113	76
Numerical Integration (Trapezoidal or Simpson's Rule)	129	513	16385	65537
$x = 22$				
Asymptotic Expansion	22	...	...	...
Continued Fraction	5	6	12	16
Numerical Integration	2049			
$x = 44$				
Asymptotic Expansion	10	35	45	...
Continued Fraction	4	5	9	11
$x = 70$				
Asymptotic Expansion	8	15	23	62
Continued Fraction	4	5	8	10

4. Range

For the function $e^x E_n(x)$, the sum of x and n must be less than or equal to the maximum machine value. If the sum is greater, both $e^x E_n(x)$ and $E_n(x)$ are set equal to zero.

For the function $E_n(x)$ the range of x is dominant and essentially equivalent to the range for the exponential function. In single precision on the Univac 1108, $E_n(x) = 0$ beyond $x \approx 85$; in double precision beyond $x \approx 704$.

5. Accuracy

Using the Univac 1108 in computing $e^x E_n(x)$, we find the maximum relative error is 1.3(− 7) for the single precision computation and 2.4(− 17) for the double precision computation.

In computing $E_n(x)$, largely due to the error of the exponential routine, the maximum relative error is 4(− 7) for the single precision computation and 4.5(− 16) for the double precision computation.

The number of accurate binary digits is essentially the lesser of $\text{NBM} - \text{NBM}^{1/2}$ or NBM−I where I is the number of binary digits representing the integer part of x.

6. Precision

The precision may be set lower than the maximum by varying NBM or if desirable, deleting NBM and setting the proper value of TOLER.

7. Timing (Seconds—Univac 1108)

For $n=1, 2, 20$ over the range	NBM=27	NBM=60
$x=0(0.02)1$ (153 values)	0.18	0.46
$x=2(1)85$ (252 values)	.19	.61
$x=85(10)715$ (192 values)	—	.28
Maximum Time/Evaluation	.003	.015

8. Testing

The double precision results obtained were compared against available published values. Further check values were obtained by utilizing multi-precision packages. In addition, double precision check values were obtained, where appropriate, with the asymptotic expansion, numerical integration, and the "odd" form of the continued fraction. Single precision results were then checked against the double precision results. In all cases, the results obtained agreed within the reported accuracy.

9. Driver Program and Its Results

In the appendix we have included a driver program and its results. The usage of the subroutine is thus illustrated and a reasonable set of check values given to simplify the checkout of modifications to the subroutine. Two tables are given; one for the function $E_n(x)$ and the other for $e^x E_n(x)$. The functions are tabulated to 18 significant figures for $n=1$, 2, and 20, for $x=0$, $10^J(10^J)10^{J+1}$ with $J=$JBEGIN(1) JEND. The value of JBEGIN has been set at -2; the value of JEND is the minimum value of J for which $E_1(x)=0$.

10. Special Constants

$E_1(1)=0.21938\ 39343\ 95520\ 27367\ 71637\ 75460\ 12164$

$$\sum_1^\infty (-1)^{n+1}/n\cdot n! = E_1(1)+\gamma = 0.79659\ 95992\ 97053\ 13428\ 36758\ 65542\ 52408$$

$\Psi(1)=-\gamma$(Euler's constant)

$=-0.57721\ 56649\ 01532\ 86060\ 65120\ 90082\ 40243$

$\Psi(2)=1-\gamma$

$=0.42278\ 43350\ 98467\ 13939\ 34879\ 09917\ 59756$

$x=e^{\Psi(1)}=0.56145\ 94835\ 66885\ 16982\ 41432\ 14790\ 88078$

$e^{\Psi(2)}=1.52620\ 51115\ 95863\ 88047\ 48887\ 15036\ 77561$

$e^{\Psi(3)}=2.51628\ 68309\ 39363\ 58025\ 62371\ 75591\ 18353$

$e^{\Psi(4)}=3.51176\ 11663\ 39476\ 18136\ 68183\ 63142\ 18018$

$e^{\Psi(5)}=4.50919\ 05949\ 16874\ 93725\ 13380\ 93818\ 19453$

$e^{\Psi(6)}=5.50753\ 78297\ 01368\ 13780\ 61569\ 70031\ 98287$

$e^{\Psi(7)}=6.50638\ 71643\ 69172\ 08390\ 16850\ 25480\ 57156$

$e^{\Psi(8)}=7.50554\ 04760\ 51118\ 60659\ 92002\ 80578\ 56992$

$e^{\Psi(9)}=8.50489\ 15798\ 67776\ 22381\ 29291\ 76413\ 19981$

$e^{\Psi(10)}=9.50437\ 85180\ 84354\ 74744\ 37262\ 91500\ 55559$

$\log_e 2=0.69314\ 71805\ 59945\ 30941\ 72321\ 21458\ 17656$

$\log_e 10=2.30258\ 50929\ 94045\ 68401\ 79914\ 54684\ 36420$

$2^{-24} = 0.59604\ 64477\ 53906\ 25\ (-7)$
$2^{-27} = \ .74505\ 80596\ 92382\ 8125\ (-8)$
$2^{-36} = \ .14551\ 91522\ 83668\ 51806\ 64062\ 5\ (-10)$
$2^{-48} = \ .35527\ 13678\ 80050\ 09293\ 55621\ 33789\ 0625\ (-14)$
$2^{-60} = \ .86736\ 17379\ 88403\ 54720\ 59622\ 40695\ 95336\ (-18)$
$2^{-108} = \ .30814\ 87911\ 01957\ 73648\ 89564\ 70813\ 58837\ (-32)$

Maximum and Minimum Machine Values and Their Natural Logarithms

NBC = Number of binary digits in the (biased) characteristic of a floating point number

$$2^{-(2^{NBC-1}+1)} \leqslant x < 2^{2^{NBC-1}-1}$$

NBC = 8

$2^{127} = 0.17014\ 11834\ 60469\ 23173\ 16873\ 03715\ 88410\ (39)$
$2^{-129} = 0.14693\ 67938\ 52785\ 93849\ 60920\ 67152\ 78070\ (-38)$
$\log_e(2^{127}) = 88.02969\ 19311\ 13054\ 29598\ 84794\ 25188\ 42414$
$\log_e(2^{-129}) = -89.41598\ 62922\ 32944\ 91482\ 29436\ 68104\ 77728$

NBC = 11

$2^{1023} = 0.89884\ 65674\ 31157\ 95386\ 46525\ 95394\ 51236\ (308)$
$2^{-1025} = 0.27813\ 42323\ 13400\ 17288\ 62790\ 89666\ 55050\ (-308)$
$\log_e(2^{1023}) = 709.08956\ 57128\ 24051\ 53382\ 84602\ 51714\ 62914$
$\log_e(2^{-1025}) = -710.47586\ 00739\ 43942\ 15266\ 29244\ 94630\ 98227$

11. References

[1] Abramowitz, M. and Stegun, I. A., Handbook of Mathematical Functions, Nat. Bur. Stand. (U.S.), Appl. Math. Ser. 55, (1964).

[2] Blanch, G., Numerical evaluation of continued fractions, SIAM Review 6, 4, 383–421 (1964).

[3] Clenshaw, C. W., Chebyshev Series for Mathematical Functions, Mathematical Tables, Vol. 5, National Physical Laboratory, 1962.

[4] Cody, W. J. and Thacher, H. C. Jr., Rational Chebyshev approximations for the exponential integral $E_1(x)$, Math. Comp. **22**, 641–649 (1968).

[5] Harris, F. E., Tables of the exponential integral $Ei(x)$, MTAC **11**, 9–16 (1957).

[6] Lee, K. L., High-Precision Chebyshev Series Approximation to the Exponential Integral, NASA TN D–5953, 1970.

[7] Miller, G. F., Tables of Generalized Exponential Integrals, National Physical Laboratory Mathematical Tables Volume 3, Her Majesty's Stationery Office, 1960.

[8] Murnaghan, F. D., and Wrench, J. W. Jr., The converging factor for the exponential integral. Report 1535, David Taylor Model Basin, 1963.

[9] Pagurova, V. I., Tables of the Exponential Integral $E_\nu(x) = \int_1^\infty e^{-xu} u^{-\nu} du$

(Translated from the Russian by D. G. Fry) (Pergamon Press, New York, 1961).

[10] Placzek, G., The functions $E_n(x) = \int_1^\infty e^{-xu} u^{-n} du$

Tables of functions and of zeros of functions, Nat. Bur. Stand. (U.S.), Appl. Math. Ser. 37, 57–111 (Nov. 1954).

[11] Wall, H. S., Analytic Theory of Continued Fractions (D. Van Nostrand Co., New York, 1948).

Appendix

Implementing Program

```
      SUBROUTINE EXPINT (RN,X,ENX,EXPENX,IERR)
C
C LANGUAGE.  AMERICAN NATIONAL STANDARD FORTRAN
C
C DEFINITIONS.
C   EN(X)= INTEGRAL (EXP(-X*T)DT/(T**N)), FROM 1 TO INFINITY
C             RN(=N), POSITIVE INTEGER
C                  X, REAL AND POSITIVE
C
C  SPECIAL CASES
C   EN(0)=INFINITY(=RINF, MAXIMUM MACHINE VALUE)    N .LE. 1
C   EN(0)= 1/N-1      N .GT. 1
C
C   E0(X)= EXP(-X)/X     X .GT. 1/RINF
C   E0(X)= INFINITY      X .LE. 1/RINF
C
C USAGE.     CALL EXPINT (RN,X,ENX,EXPENX,IERR)
C
C  FORMAL PARAMETERS
C   RN  REAL OR DOUBLE PRECISION TYPE                        INPUT
C       FOR A POSITIVE INTEGER N
C   X                           (SAME TYPE AS RN)            INPUT
C   ENX= EN(X)                   (SAME TYPE AS X)           OUTPUT
C   EXPENX= EXP(X)*EN(X)         (SAME TYPE AS X)           OUTPUT
C   IERR INTEGER VARIABLE                                   OUTPUT
C         NORMAL RETURN  IERR= 0
C         ERROR RETURN   IERR= 1, X AND/OR N NEGATIVE
C                    ENX=EXPENX=-INFINITY (IMPOSSIBLE VALUE)
C                        IERR= 2, N NON-INTEGER
C                    ENX=EXPENX=INFINITY
C
C  MODIFICATIONS
C   DOUBLE PRECISION UNIVAC 1108 RESULTS ARE OBTAINED IF AS
C   SET UP BELOW WHERE
C
C     NBM=ACCURACY DESIRED OR MAXIMUM NUMBER OF BINARY
C          DIGITS IN THE MANTISSA OF A FLOATING POINT NUMBER
C
C   WITH
C    (1) THE DOUBLE PRECISION TYPE STATEMENT
C    (2) THE MAXIMUM MACHINE VALUE AND THE MAXIMUM INTEGER
C        (=RMAXI) CONVERTIBLE TO A FLOATING POINT NUMBER
C        GIVEN AS DOUBLE PRECISION CONSTANTS
C    (3) DOUBLE PRECISION DECIMAL CONSTANTS
C    (4) DATA STATEMENT NBM=60 FOR THE CONTROL VARIABLE
C    (5) FUNCTION TYPE STATEMENTS - DEXP, DLOG.
C
C   SINGLE PRECISION UNIVAC 1108 RESULTS ARE OBTAINED BY
C      (1) DELETING THE DOUBLE PRECISION TYPE STATEMENT
C      (2) ADJUSTING MAXIMUM MACHINE VALUE
C      (3) CHANGING THE D'S TO E'S ON THE DATA CARDS FOR ALL
C          DECIMAL CONSTANTS
C      (4) SETTING NBM=27 IN THE CONTROL VARIABLE DATA
C      (5) CHANGING FUNCTION TYPE - EXP, LOG.
C
```

```
C      FOR OTHER COMPUTERS THE APPROPRIATE VALUE OF NBM MUST
C      BE INSERTED AND ALL VALUES ADJUSTED ACCORDINGLY.
C
C    IF A PRECOMPUTED VALUE OF TOLER(=2**(-NBM)) IS
C    INCLUDED IN A DATA STATEMENT, COMPUTATION OF THE CONTROL
C    VARIABLE MAY BE OMITTED AND THE DATA STATEMENT FOR NBM
C    DELETED.
C
C    CAUTION - THE SUBROUTINE CANNOT READILY BE ADAPTED TO
C              COMPUTE THE EXPONENTIAL INTEGRAL FOR A COMPLEX
C              ARGUMENT AS THE CONTINUED FRACTION IS INVALID
C              ALONG THE NEGATIVE REAL AXIS. IN ADDITION MANY
C              OF THE COMPARISONS BECOME MEANINGLESS.
C
C METHOD.
C  POWER SERIES,     X .LE. 1(=ULPS, UPPER LIMIT FOR POWER
C                                    SERIES)
C    ENX = SUM(TM),     M=0,1,2,...,K
C     TM =-((-X)**M/1*2*3...M)/(M-N+1=D)        M .NE. N-1
C        = PTERM/D
C     TM = PTERM*(LOG(X)-PSI(N))     M .EQ. N-1
C      PTERM(0)=-1
C      PTERM(M+1)= PTERM(M)*(-X)/(M+1)
C      PSI(N)= -EULER+1+1/2+ ... +1/(N-1)
C      IF R.E.(=ABS(TM/SUM)) .LE. TOLER (=2**(-NBM)),       M=K
C
C  CONTINUED FRACTION,     X .GT. 1
C    EN(X)=EXP(-X)*(1I/I(X+N)-    1*N I/I(X+N+2)-
C                                2*(N+1)I/I(X+N+4)-...
C    EN(X)=EXP(-X)*II(AM I/I BM)          M=1,2,...,K
C      AM(1)=1      AM(M)=-(M-1)*(N+M-2)
C      BM(M)=X+N+2*(M-1)
C
C    EN(X)=EXP(-X)*FM(K)/GM(K)=EXP(-X)*F(K)
C         IF R.E. (=ABS(1-PREV/F)) .LE. TOLER(=2**(-NBM)),M=K
C      F=FM/GM      PREV=FMM1/GMM1
C
C RANGE.
C  FOR EXP(X)*EN(X)     (N+X) .LE. MAXIMUM MACHINE VALUE
C  FOR EN(X)    X=APPROXIMATELY 85.0, SINGLE PRECISION
C                                704.0, DOUBLE PRECISION
C  BEYOND THIS RANGE EN(X)=0
C
C ACCURACY.  THE NUMBER OF ACCURATE BINARY DIGITS IS ESSEN-
C            TIALLY THE LESSER OF
C                NBM - SQUARE ROOT OF NBM
C                NBM - I(NUMBER OF BINARY DIGITS REPRESENT-
C                      ING THE INTEGER PART OF X)
C                      (ACCURACY OF THE EXPONENTIAL ROUTINE)
C
C PRECISION. VARIABLE - BY SETTING THE DESIRED NBM OR TOLER.
C
C MAXIMUM TIME.                  UNIVAC 1108-EXEC II
C    (SECONDS)                     NBM=27     NBM=60
C                                   .0025      .015
C
C STORAGE.   COMPILED BY THE UNIVAC 1108, EXEC 8/FORTRAN V,
C            THIS SUBROUTINE REQUIRES 478 WORDS OF STORAGE.
C
C
C
```

```
C
C
C
C
C
C               MACHINE DEPENDENT STATEMENTS
C
          DOUBLE PRECISION AM,ASUM,BM,D,ENX,EULER,
         1       EXPENX,EXPNX,F,FM,FMM1,FMM2,GM,GMM1,
         2       GMM2,HALF,ONE,ONPTFV,PREV,PSI,PTERM,
         3       RINF,RM,RMAXI,RN,RNM1,RRN,SUM,TEMP,
         4       TM,TOLER,TWO,ULPS,X,XLOG,ZERO
C
C                     CONSTANTS
C                       MAXIMUM MACHINE VALUE
C                       MAXIMUM CONVERTIBLE INTEGER
C
          DATA RINF / .898846567431157954D308 /
          DATA RMAXI / 134217727.D0 /
C
C                     ADDITIONAL CONSTANTS
C
          DATA EULER / .577215664901532861D0 /
          DATA HALF,ONE,ONPTFV,TWO,ULPS,ZERO /
         1   .5D0,1.D0,1.5D0,2.D0,1.D0,0.0D0 /
C
C                     CONTROL VARIABLE
C
C         NBM=ACCURACY DESIRED OR MAXIMUM NUMBER OF BINARY
C              DIGITS IN THE MANTISSA OF A FLOATING POINT NUMBER
C
          DATA NBM / 60 /
          TOLER=TWO**(-NBM)
C
C               VALIDITY TEST FOR INPUT PARAMETERS
C               ERROR CONDITION, IMPOSSIBLE VALUES RETURNED
C
C                   NEGATIVE ZERO CHECKS
C
          IERR=0
          IF (RN .LT. ZERO) GO TO 3
          IF (RN .GT. RMAXI) GO TO 10
C
C                   VALIDITY TEST FOR N INTEGER
C
          N=RN
          RRN=N
          IF (RRN) 2,1,2
        1 IF ((RN-RRN)-TOLER) 10,10,2
        2 IF ((ONE-(RRN/RN)) .LE. TOLER) GO TO 10
          RRN=RRN+ONE
          IF ((ONE-(RN/RRN)) .LE. TOLER) GO TO 10
C
C                    ERROR RETURN
C                    N NON-INTEGER
C
          IERR=2
          ENX=RINF
          EXPENX=ENX
          RETURN
C
```

```
    3  IF (-RN) 12,10,12
C
   10  IF (X) 11,30,20
   11  IF (-X) 12,30,12
C
C                    ERROR RETURN
C                    X AND/OR N NEGATIVE
C
   12  IERR=1
       ENX=-RINF
       EXPENX=ENX
       RETURN
C
C              FUNCTION TYPE STATEMENTS
C
   20  EXPNX=DEXP(-X)
       XLOG=DLOG(X)
C
       IF (RN .GE. HALF) GO TO 60
C
C              SPECIAL CASES
C
       IF (X .LE. ONE/RINF) GO TO 40
       EXPENX=ONE/X
       ENX=EXPNX*EXPENX
       RETURN
C
   30  IF (RN .LT. ONPTFV) GO TO 40
       ENX=ONE/(RN-ONE)
       GO TO 50
C
   40  ENX=RINF
   50  EXPENX=ENX
       RETURN
C
   60  IF (RN .GT. RMAXI) GO TO 70
       IF (X .LE. ULPS) GO TO 100
   70  IF (X .LE. (RINF-RN)) GO TO 200
       EXPENX=ZERO
       ENX=EXPENX
       RETURN
C
C          METHOD --- POWER SERIES
C
  100  RM=0
       PTERM=-ONE
       SUM=0
       PSI=-EULER
       D=-(RN-ONE)
C
  110  IF (D .GE. HALF) GO TO 130
         IF (-D .GE. HALF) GO TO 120
C
C              COMPUTE TM FOR M .EQ. N-1
C
         SUM=PTERM*(XLOG-PSI)+SUM
         GO TO 170
C
C              COMPUTE PSI(N)
C
  120    PSI=PSI+(ONE/(RM+ONE))
```

```
C
C              COMPUTE TM FOR M .NE. N-1
C
 130    TM=PTERM/D
        SUM=TM+SUM
C
C              TOLERANCE CHECK
C
C                  ZERO CHECKS
C
        IF (SUM .LT. ZERO) GO TO 140
        ASUM=SUM
        GO TO 150
C
 140    ASUM=-SUM
 150    IF (ASUM) 160,170,160
 160    IF (TM .LT. ZERO) TM=-TM
        IF ( TM/ASUM .GT. TOLER) GO TO 170
C
      ENX= SUM
      EXPENX=ENX/EXPNX
      RETURN
C
C              ADDITIONAL TERMS
C
 170    RM=RM+ONE
        D=D+ONE
        PTERM=-(X*PTERM)/RM
        GO TO 110
C
C         METHOD --- CONTINUED FRACTION
C
 200  RM=ONE
      FMM2=ONE
      GMM2=0
      FMM1=0
      GMM1=ONE
      PREV=FMM1/GMM1
      AM=ONE
      BM=X+RN
      RNM1=RN-ONE
C
 210  FM=BM*FMM1 + AM*FMM2
        GM=BM*GMM1 + AM*GMM2
        F=FM/GM
C
C              TOLERANCE CHECK
C
        TEMP=ONE-(PREV/F)
        IF (TEMP .GT. ZERO) GO TO 220
C
      EXPENX=PREV
      GO TO 230
C
 220    IF (TEMP .GT. TOLER) GO TO 240
C
      EXPENX=F
 230  ENX=EXPNX*EXPENX
      RETURN
C
 240    IF( GM .LT. RINF/BM) GO TO 250
```

```
      C
      C           SCALING
      C
      C BOTH FM AND GM MUST BE TESTED IF N=1 AND X .LT. .44
      C SCALING SHOULD NOT BE DELETED AS THE VALUES OF FM AND GM
      C MAY OVERFLOW FOR SOME CHOICES OF THE PARAMETERS.
      C
                 FMM1=FMM1/BM
                 GMM1=GMM1/BM
                 FM=FM/BM
                 GM=GM/BM
      C
      C           ADDITIONAL CONVERGENTS
      C
        250     AM=-RM*(RNM1+RM)
                RM=RM+ONE
                BM=BM+TWO
                FMM2=FMM1
                GMM2=GMM1
                FMM1=FM
                GMM1=GM
                PREV=F
                GO TO 210
      C
            END
```

Driver (Test) Program

```
C
C DOUBLE PRECISION TEST PROGRAM
C ALGORITHM FOR EXPONENTIAL INTEGRAL EN(X) AND EXP(X)*EN(X)
C RN(=N,A POSITIVE INTEGER), X, REAL AND POSITIVE
C (AMERICAN NATIONAL STANDARD FORTRAN - UNIVAC 1108)
C
C USAGE OF CALL EXPINT (RN,X,ENX,EXPENX,IERR)
C
C        TABLE OF ENX(X(L),N(K)) AND EXPENX(X(L),N(K))
C                     JT=1                  JT=2
C                L=1,2,...,NCX,   K=1,2,...,NN
C  N=1,2,20    X=0, 10**J (10**J) 10**(J+1)    J=JBEGIN(1)JEND
C                   (10**J)(ID=1,2,...,9)         JBEGIN=-2
C  JEND=MIN J FOR ENX(X=(10**J)*ID,N(1))=0
C
C   NLINE = NUMBER OF AVAILABLE PRINT LINES PER PAGE
C   NFCOL = NUMBER OF FUNCTIONAL COLUMNS PER PAGE
C   LINHD = NUMBER OF HEADING LINES PER PAGE
C
C   RMAXI = MAXIMUM CONVERTIBLE INTEGER
C
C  DIMENSION N(NN),X(NX),ENX(NX,NN),EXPENX(NX,NN)
C
C               MACHINE DEPENDENT STATEMENTS
C               INPUT DATA
C
      DOUBLE PRECISION D,DFCTR,DX,ENX,ETPTFV,EXPENX,
     1                 PTFIVE,RMAXI,RN,TEN,X,ZERO
C
      DIMENSION N(3),X(100),ENX(100,3),EXPENX(100,3)
C
      DATA RMAXI / 134217727.D0 /
      DATA ETPTFV,PTFIVE,ZERO / 8.5D0,.5D0,0.0D0 /
C
      DATA NN,NX,NLINE,NFCOL,LINHD/ 3,100,58,3,3 /
      DATA N(1),N(2),N(3) / 1,2,20 /
      DATA JBEGIN / -2 /
C
1000  FORMAT (1H1,38X,5HEN(X)//)
1001  FORMAT (4X,1HX,9X,2HN=,1I12,4(13X,I12))
1002  FORMAT (1D7.1,5D25.18)
1003  FORMAT (1H1,35X,12HEXP(X)*EN(X)//)
C
C               FUNCTION TYPE STATEMENT
C
      MAXJI=DLOG10(RMAXI)
C
C                 COMPUTATION AND STORAGE OF FUNCTIONS
C
C                   SET UP INITIAL X
C
      D=ZERO
C
C                   SET UP INITIAL DECADE
C
      TEN=10
      L=1
      J=JBEGIN
      ID=1
```

```
C
      IF (IABS(J) .GT. MAXJT) GO TO 1
      DFCTR=10**(IABS(J))
      GO TO 2
C
    1 DFCTR=TEN**(IABS(J))
C
    2 IF (J .LE. 0) GO TO 3
        DX=D*DFCTR
        GO TO 4
C
    3   DX=D/DFCTR
C
    4   X(L)=DX
C
        DO 5 K=1,NN
          RN=N(K)
    5     CALL EXPINT(RN,DX,ENX(L,K),EXPFNX(L,K),IERR)
C
        IF (ENX(L,1) .LE. ZERO) GO TO 9
        IF (L .GE. NX) GO TO 9
        IF (D .LT. PTFIVE) GO TO 8
        IF (D .GT. ETPTFV) GO TO 6
        ID=ID+1
        GO TO 8
C
    6   J=J+1
        ID=1
        IF (J .LE. 0) GO TO 7
        DFCTR=DFCTR*TEN
        GO TO 8
C
    7   DFCTR=DFCTR/TEN
C
    8   D=ID
        L=L+1
        GO TO 2
C
    9 NCX=L
C
C              PRINTING OF TABLES
C              L=1,2,...,NCX
C              K=1,2,...,NN
C
      JT=1
C
   21 NA=1
        NB=NFCOL
C
   22   IF (NB .GT. NN) NB=NN
          L=1
          IF (X(1)) 24,23,24
   23     NPLINE=1
          LCOUNT=NLINE-LINHD-1
          INDEXS=9
          GO TO 26
C
   24     NPLINE=0
C
   25     LCOUNT=NLINE-LINHD
            INDEXS=1
```

```
  26        NPLINE=NPLINE+9*(LCOUNT/10)
            IF (JT .EQ. 1) GO TO 27
            WRITE (6,1003)
            GO TO 28
C
  27        WRITE (6,1000)
  28        WRITE (6,1001) (N(K),K=NA,NB)
C
  29        IF (JT .EQ. 1) GO TO 30
              WRITE (6,1002) X(L),(EXPENX(L,K),K=NA,NB)
              GO TO 31
C
  30          WRITE (6,1002) X(L),(ENX(L,K),K=NA,NB)
  31          IF (L .GE. NCX) GO TO 35
              IF (L .GE. NPLINE) GO TO 34
              IF (INDEXS .EQ. 9) GO TO 32
              INDEXS=INDEXS+1
              GO TO 33
C
  32          WRITE (6,1004)
              INDEXS=1
C
  33          L=L+1
              GO TO 29
C
C                    EXCESS X'S
C
  34        L=L+1
            GO TO 25
C
C                  EXCESS N'S
C
  35      IF (NB .GE. NN) GO TO 36
          NA=NA+NFCOL
          NB=NB+NFCOL
          GO TO 22
C
C                ADDITIONAL FUNCTION
C
  36    IF (JT .GE. 2) STOP
        JT=JT+1
        GO TO 21
C
1004  FORMAT (1H )
C
      END
```

Test Results

EN(X)

X	N= 1	2	20
.0	.898846567431157951+308	.100000000000000000+001	.526315789473684210-001
.1-001	.403792957653811384+001	.949670537983786916+000	.520789541793351476-001
.2-001	.335470778330970952+001	.913104517640561112+000	.515321496513528285-001
.3-001	.295911872402128069+001	.881671971827869758+000	.509911038545502872-001
.4-001	.268126368902527992+001	.853538891591312014+000	.504557559326025865-001
.5-001	.246789848850997437+001	.827834500075215292+000	.499260456747777298-001
.6-001	.229530691814378233+001	.804046118495621770+000	.494019135090578269-001
.7-001	.215083818025679885+001	.781835147287972310+000	.488833004953339252-001
.8-001	.202694100258574148+001	.760961066179776466+000	.483701483186737100-001
.9-001	.191874477003266286+001	.741244155968288530+000	.478623992826612893-001
.1+000	.182292395841939067+001	.722545022194020509+000	.473599963028082897-001
.2+000	.122265054418389309+001	.574200644241203245+000	.426178656910138496-001
.3+000	.905676651675846714+000	.469115225178963854+000	.383518490868851917-001
.4+000	.702380118865662480+000	.389367998489374312+000	.345140002098048623-001
.5+000	.559773594776160810+000	.326643862324553017+000	.310612173936309823-001
.6+000	.454379503189402111+000	.276183934180385169+000	.279547522194935671-001
.7+000	.373768843233509144+000	.234947113527953114+000	.251597681997253041-001
.8+000	.310596578545543035+000	.200851701280787165+000	.226449443896666456-001
.9+000	.260183939325999640+000	.172404114347199437+000	.203821193314687553-001
.1+001	.219383934395520273+000	.148495506775922048+000	.183459712067558733-001
.2+001	.489005107080611205-001	.375342618204904530-001	.641430585532489937-002
.3+001	.130483810941970376-001	.106419250852728306-001	.224864806414000919-002
.4+001	.377935240984890646-002	.319822924933855437-002	.790189099803715192-003
.5+001	.114829559127532581-002	.996469042708838118-003	.278274592885730814-003
.6+001	.360082452162658656-003	.318257463690406465-003	.981884227302490187-004
.7+001	.115481731610338216-003	.103509844282148693-003	.347068486248843707-004
.8+001	.376656228439249018-004	.341376451511126247-004	.122877543566663061-004
.9+001	.124473541780062721-004	.113836164846231004-004	.435687424044296172-005
.1+002	.415696892968532427-005	.383024046563160875-005	.154693627987772485-005
.2+002	.983552529064988168-010	.940485643085814897-010	.521648146507963053-010
.3+002	.302155201068881253-014	.292966936773736969-014	.188625975171560854-014
.4+002	.103677326145165697-018	.101261209484961107-018	.711928232079808894-019
.5+002	.378326402955045901-023	.371178331886882736-023	.276642308097688871-023
.6+002	.143586756568125679-027	.141305368608979607-027	.109793192761257778-027
.7+002	.560030628581343844-032	.552353358392398957-032	.442791440616991968-032
.8+002	.222854325868847290-036	.220167808946368433-036	.180841147365357070-036
.9+002	.900547405886741103-041	.890859710098454963-041	.746125498994224778-041
.1+003	.368359776168203217-045	.364782143388037826-045	.310431603951609625-045
.2+003	.688522610630763559-089	.685130547521041110-089	.629301793252036997-089
.3+003	.171038427680451011-132	.170473919984834340-132	.160912502553792433-132
.4+003	.477601358642097222-176	.476416214561680312-176	.456044233853820537-176
.5+003	.142207678225363842-219	.141924954730934210-219	.137021182168391364-219
.6+003	.440998979450983796-263	.440267629840803335-263	.427505488621821494-263
.7+003	.140651876623403292-306	.140451801215403972-306	.136945221165125589-306
.8+003	.000000000000000000	.000000000000000000	.000000000000000000

EXP(X)*EN(X)

X	N= 1	2	20
.0	.898846567431157951+308	.100000000000000000+001	.526315789473684210-001
.1-001	.407851144345642585+001	.959214885565435741+000	.526023563704061981-001
.2-001	.342247737593075321+001	.931550452481384936+000	.525731681287694728-001
.3-001	.304923730567447425+001	.908522880829765773+000	.525440141582159931-001
.4-001	.279068813598834047+001	.888372474560466383+000	.525148943946742833-001
.5-001	.259443034976061332+001	.870278482511969335+000	.524858087742430369-001
.6-001	.243724077122346628+001	.853765553726592023+000	.524567572331905115-001
.7-001	.230679154487934795+001	.838524591858445643+000	.524277397079539275-001
.8-001	.219575897504124847+001	.824339281996700124+000	.523987561351388700-001
.9-001	.209944118436360746+001	.811050293407275329+000	.523698064515186901-001
.1+000	.201464254470845168+001	.798535745529154834+000	.523408905940339148-001
.2+000	.149334874693223961+001	.701330250613552081+000	.520535787019039881-001
.3+000	.122253560508058556+001	.633239318475824335+000	.517695812767576684-001
.4+000	.104782800845600643+001	.580868796617597430+000	.514888379273828789-001
.5+000	.922910632483730466+000	.538544683758134765+000	.512112898107201955-001
.6+000	.827933435273508820+000	.503239938835894713+000	.509368795793975367-001
.7+000	.752678020029587137+000	.473125385979289004+000	.506655513315264279-001
.8+000	.691245397802831489+000	.447003681757734811+000	.503972505626390015-001
.9+000	.639949226639299739+000	.424045696024630237+000	.501319241196527574-001
.1+001	.596347362323194074+000	.403652637676805926+000	.498695201567573519-001
.2+001	.361328616888222592+000	.277342766223554833+000	.473956658006950287-001
.3+001	.262083740255318501+000	.213748779234044509+000	.451653037195386256-001
.4+001	.206345649901055832+000	.174617400395776667+000	.431428630256384291-001
.5+001	.170422176284732204+000	.147889118576338992+000	.412996114281546988-001
.6+001	.145267629233886893+000	.128394224596678636+000	.396120369170190107-001
.7+001	.126641096076632765+000	.113512327463570647+000	.380606810266053526-001
.8+001	.112279639253499312+000	.101762885972005506+000	.366292794923112188-001
.9+001	.100861955580640929+000	.922423997742316384-001	.353041176322005723-001
.1+002	.915633339397880818-001	.843666606021191810-001	.340735390554722559-001
.2+002	.477185454959608417-001	.456290900807831660-001	.253085524935690803-001
.3+002	.322897387589801252-001	.313078372305962434-001	.201574668908614936-001
.4+002	.244041150796285762-001	.238353968148569492-001	.167577416876913709-001
.5+002	.196151099301148704-001	.192445034942564817-001	.143430890423517574-001
.6+002	.163977137080465268-001	.161371775172083932-001	.125384637484814277-001
.7+002	.140872270003268122-001	.138941099771231436-001	.111381471287998954-001
.8+002	.123475166636786097-001	.121986669057112264-001	.100197250911190282-001
.9+002	.109903102083356454-001	.108720812497919106-001	.910574016947094732-002
.1+003	.990194228673301838-002	.980577132669815934-002	.834476515943799403-002
.2+003	.497524632317935662-002	.495073536412867515-002	.454731825027371299-002
.3+003	.332229556527070706-002	.331133041878788067-002	.312560692284812426-002
.4+003	.249378101793988503-002	.248759282404598474-002	.238122114425943504-002
.5+003	.199601590476041089-002	.199204761979455498-002	.192321864972550092-002
.6+003	.166389810215794723-002	.166113870523165916-002	.161298688732910590-002
.7+003	.142653641830088669-002	.142450718937931575-002	.138894233048444955-002
.8+003	.124844139167435033-002	.124688666051973815-002	.121954838359794140-002

(Paper 78B4–415)