SAND76-8209
Unlimited Release

User's Guide to the Sandia Mathematical Program Library at Livermore

(Supersedes SAND74-8222 and SAND75-8281)

R. E. Huddleston, T. H. Jefferson

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87115 and Livermore, California 94550 for the United States Atomic Energy Commission under Contract AT (29-1)-789

Printed March 1976

Sandia Laboratories

SF 2900 Q(7-73)

When printing a copy of any digitized SAND Report, you are required to update the markings to current standards.

Issued by Sandia Laboratories, operated for the United States Energy Research and Development Administration by Sandia Corporation.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SAND76-8209
Unlimited Release
Printed March 1976

User's Guide to the Sandia Mathematical Program Library at Livermore

(Supersedes SAND74-8222 and SAND75-8281)

R. E. Huddleston
T. H. Jefferson
Numerical Applications Division 8322
Sandia Laboratories, Livermore

ABSTRACT

The Sandia Mathematical Program Library is a collection of general-purpose mathematical subroutines which are maintained within Sandia on a quick service basis. This document is intended to be a reference guide for using the library at Sandia Laboratories, Livermore.

Contents

Page

1. Introduction 7

2. Obtaining the Routine. 7

3. Comments on Other Libraries. 7

4. Subroutines Grouped According to Task. 8

Data Fitting. 8

Eigenvalues and Eigenvectors of Matrices. 9

Fourier Transforms. 9

Linear Algebraic Equations. 10

Systems of Nonlinear Algebraic Equations. 10

Linear Programming and Game Theory. 10

Numerical Quadrature. 10

Numerical Sorting 11

Ordinary Differential Equations 11

Special Functions 11

Zeros of Functions and Optimization 12

Zeros of Polynomials. 13

Library Error Check Routine and User Options. 13

5. Calling Sequences for Subroutines (Alphabetical Order) 15

AVINT 15

BESI. 16

BESI01. 17

BESJ. 17

BESJ01. 18

BESKN 19

	Page
BESKO1	20
BESYN	20
BESYO1	21
CAXBI	22
CBND2	24
CHAA	25
CHAN	26
CNAA	27
CNAN	28
CNPCOF	29
CNPFIT	30
CNPVAL	32
COSH	33
CPQR	34
ERF	35
ERFC	35
ERRCHK	36
ERRGET	36
ERXSET	37
FOURT	38
FOURTH	39
FOURTR	40
GAMLN	41
GAMMA	41
GAMMAZ	42
GAUS8	43
GERK	44
HRMITE	46
LPGAME	47
MINA	49
ODE	50
ODERT	52
PCOEF	54
POLCOF	55
POLFIT	56
POLINT	57
POLYVL	58
PSMTH1	59
PVALUE	60
QN	61
QNC3	63
QNC7	64
RBND2	65
RDET	66
RFFT	67
RFFTI	68
RKF	69
RNAA	72
RNAN	73
RPQR	74
RSAA	75
RSAN	76
SAXB	77
SAXBI	78
SICONT	80
SIMIN	81
SINH	82
SMOO	83
SPLIFT	84
SPLINT	85
SPLIQ	86
SSORT	87
STEP1	87
SUPORT	89
TJMAR1	93
ZEROIN	96

1. INTRODUCTION

THE SANDIA MATHEMATICAL PROGRAM LIBRARY (SMPL) IS A COLLECTION OF GENERAL PURPOSE ROUTINES WHICH ARE PRIMARILY MATHEMATICAL IN NATURE. THESE ROUTINES ARE OF GOOD QUALITY AND EACH ONE IS MAINTAINED ON AN IN-HOUSE BASIS. CONSULTANTS FOR THESE ROUTINES AT SANDIA LIVERMORE INCLUDE

R. E. HUDDLESTON

T. H. JEFFERSON

DIVISION 2642 HAS PRIME RESPONSIBILITY FOR THE SANDIA MATHEMATICAL PROGRAM LIBRARY PROJECT. QUESTIONS REGARDING THE LIBRARY PROJECT SHOULD BE DIRECTED TO ONE OF THE FOLLOWING MEMBERS OF THE SANDIA MATHEMATICAL PROGRAM LIBRARY PROJECT COMMITTEE :

M. R. SCOTT	2642
R. E. JONES	2642
L. F. SHAMPINE	5122
D. E. AMOS	5122
R. Y. LEE	8322
R. E. HUDDLESTON	8322

2. OBTAINING THE ROUTINE

IN SECTION 4, THE ROUTINES ARE GROUPED ACCORDING TO THEIR TASK. HAVING LOCATED A SUITABLE ROUTINE, ONE TURNS TO SECTION 5 FOR AN EXPLANATION OF THE CALLING SEQUENCE. USING THIS EXPLANATION, HE WRITES THE CALL INTO HIS FORTRAN PROGRAM. MOST USERS WILL PROCESS THEIR DECKS WITH A CONTROL CARD SEQUENCE SUCH AS

```
--FORTRAN EXTENDED COMPILER--
    JOB CARD
    ACCOUNT CARD
    FTN.
    ATTACH(MATHFTN,ID=MATHFTN)
    LDSET(LIB=MATHFTN)
    LGO.
```

```
--RUN COMPILER--
    JOB CARD
    ACCOUNT CARD
    RUN,S.
    ATTACH(MATHRUN,ID=MATHRUN)
    LDSET(LIB=MATHRUN)
    LGO.
```

THE LDSET(LIB=------) CARD WILL CAUSE A SEARCH THROUGH YOUR PROGRAM FOR ANY CALLS TO SUBROUTINES ON THE MATH LIBRARY AND WILL AUTOMATICALLY LOAD ONLY THOSE ROUTINES NEEDED INTO YOUR PROGRAM.

3. COMMENTS ON OTHER LIBRARIES

IN ADDITION TO THE ROUTINES ON THE SANDIA MATHEMATICAL PROGRAM LIBRARY WE HAVE AVAILABLE A LARGE COLLECTION OF WORTHWHILE SUBROUTINES WHICH ARE NOT SUPPORTED IN THE SAME MANNER AS THOSE OF THE SANDIA LIBRARY. INFORMATION CONCERNING THESE ROUTINES MAY BE OBTAINED FROM R. E. HUDDLESTON.

IMSL -

THE INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARY (IMSL) IS AN EXTENSIVE LIBRARY WHICH WE RENT. IN PARTICULAR IT HAS A NUMBER OF STATISTICALLY ORIENTED ROUTINES WHICH ARE NOT AVAILABLE ON THE SANDIA LIBRARY.

ORNL -

THE OAK RIDGE NATIONAL LABORATORY LIBRARY

LASL -

THE LOS ALAMOS SCIENTIFIC LABORATORY LIBRARY

AUXLIB -

THE SANDIA AUXILLARY LIBRARY

ARL -

THE APPLIED RESEARCH LABORATORY LIBRARY OF LINEAR ALGEBRA ROUTINES.

HARWELL -

THE LIBRARY OF THE ATOMIC ENERGY RESEARCH ESTABLISHMENT AT HARWELL, ENGLAND.

4. SUBROUTINES GROUPED ACCORDING TO TASK

****

DATA FITTING

CNPFIT
CNPVAL
CNPCOF

SUBROUTINE CNPFIT COMPUTES LEAST - SQUARE POLYNOMIAL FITS TO DATA SUBJECT TO CERTAIN CONSTRAINTS WHICH THE USER MAY WISH TO IMPOSE ON THE VALUE OF THE FIT (AND ITS DERIVATIVES) AT CERTAIN POINTS. CNPVAL COMPUTES VALUES OF THE FIT (AND ITS DERIVATIVES) PRODUCED BY CNPFIT. CNPCOF COMPUTES THE COEFFICIENTS OF THE FIT.

POLINT
HRMITE
POLYVL
POLCOF

SUBROUTINE POLINT CALCULATES THE UNIQUE INTERPOLATING POLYNOMIAL DEFINED BY A SET OF DATA. IF THE DATA INCLUDE FUNCTION VALUES AND DERIVATIVE VALUES, THEN SUBROUTINE HRMITE WILL CALCULATE THE INTERPOLATING POLYNOMIAL. SUBROUTINE POLYVL CALCULATES THE VALUE OF THE INTERPOLATING POLYNOMIAL (AND DERIVATIVES) AS PRODUCED BY EITHER POLINT OR HRMITE. SUBROUTINE POLCOF CALCULATES THE COEFFICIENTS OF THE INTERPOLATING POLYNOMIAL PRODUCED BY EITHER POLINT OR HRMITE.

POLFIT
PVALUE
PCOEF

SUBROUTINE POLFIT COMPUTES LEAST-SQUARE POLYNOMIAL FITS TO DATA USING ORTHOGONAL POLYNOMIALS FOR AN INTERNAL REPRESENTATION. PVALUE EVALUATES THE FIT (AND DERIVATIVES) PRODUCED BY POLFIT. PCOEF COMPUTES THE COEFFICIENTS OF THE FIT PRODUCED BY POLFIT.

PSMTH1

SUBROUTINE PSMTH1 IS A COMPUTATIONAL PROCEDURE FOR POLYNOMIAL SMOOTHING OF DATA AND FOR CALCULATING DERIVATIVES FROM A STRING OF DATA. SHORT STRINGS OF OVERLAPPING DATA ARE USED FOR THE POLYNOMIAL FITS TOGETHER WITH ROUTINES FOR SELECTING THE PROPER DEGREE OF FIT FOR EACH STRING.

SMOO

COMPUTES THE PARAMETERS OF A SMOOTHING SPLINE FIT TO DATA. ESPECIALLY RECOMMENDED FOR DIFFERENTIATING NOISY DATA.

SPLIFT
COMPUTES THE PARAMETERS OF AN EXACT SPLINE FIT TO DATA.

SPLIQ
INTEGRATES A CUBIC SPLINE (DEFINED BY SPLIFT, SMOOTH, ETC.)

SPLINT
INTERPOLATES VALUES ON A SPLINE USING PARAMETERS FROM EITHER SPLIFT OR SMOOTH.

TJMAR1
TJMAR1 IS A SUBROUTINE DESIGNED FOR NONLINEAR LEAST SQUARES PARAMETER ESTIMATION. THE PRINCIPAL APPLICATIONS OF THE ROUTINE ARE IN DATA FITTING AND IN SOLVING SYSTEMS OF SIMULTANEOUS NONLINEAR ALGEBRAIC EQUATIONS, ALTHOUGH ANY PROBLEM WHICH CAN BE CAST AS THE MINIMIZATION OF THE SUM OF SQUARES OF ARBITRARY RESIDUAL FUNCTIONS IS APPROPRIATE.

****

EIGENVALUES AND EIGENVECTORS OF MATRICES
(CONTACT R. E. HUDDLESTON, EXT. 2120, FOR A MUCH MORE COMPLETE SET OF ROUTINES IF NEEDED)

CHAA
COMPUTES ALL EIGENVECTORS AND EIGENVALUES OF A COMPLEX HERMITIAN MATRIX.

CHAN
COMPUTES ALL OF THE EIGENVALUES OF A COMPLEX HERMITIAN MATRIX.

CNAA
COMPUTES ALL EIGENVECTORS AND EIGENVALUES OF A COMPLEX NON - HERMITIAN MATRIX.

CNAN
COMPUTES ALL OF THE EIGENVALUES OF A COMPLEX NON - HERMITIAN MATRIX.

RSAA
COMPUTES ALL EIGENVALUES AND ALL EIGENVECTORS OF A REAL SYMMETRIC MATRIX.

RSAN
COMPUTES ALL OF THE EIGENVALUES OF A REAL SYMMETRIC MATRIX.

RNAA
COMPUTES ALL OF THE EIGENVALUES AND EIGENVECTORS OF A REAL MATRIX.

RNAN
COMPUTES ALL OF THE EIGENVALUES OF A REAL MATRIX.

****

FOURIER TRANSFORMS

FOURT
FAST FOURIER TRANSFORM ROUTINE FOR N-DIMENSIONAL COMPLEX DATA WITH AN ARBITRARY NUMBER OF VALUES IN EACH DIMENSION.

FOURTR
PERFORMS A FORWARD FAST FOURIER TRANSFORM ON A ONE-DIMENSIONAL SET OF REAL DATA. (SEE DESCRIPTION OF RFFT)

FOURTH
PERFORMS AN INVERSE FAST FOURIER TRANSFORM TO YIELD A ONE-DIMENSIONAL SET OF REAL VALUES. (SEE DESCRIPTION OF RFFTI)

RFFT
RFFTI

RFFT PERFORMS A FORWARD FAST FOURIER TRANSFORM ON A ONE-DIMENSIONAL SET OF REAL VALUES.
RFFTI PERFORMS AN INVERSE FAST FOURIER TRANSFORM TO YIELD A ONE-DIMENSIONAL SET OF REAL VALUES.
$ NOTE:
$ THE TASKS PERFORMED BY RFFT AND RFFTI CAN BE PERFORMED BY FOURTR
$ AND FOURTH. HOWEVER, RFFT AND RFFTI ARE THREE TO FIVE TIMES FASTER
$ (PARTLY DUE TO THE FACT THAT RFFT AND RFFTI ARE WRITTEN IN THE
$ CDC6600 ASSEMBLY LANGUAGE)
$$$$$

****
LINEAR ALGEBRAIC EQUATIONS

CAXBI
SOLVES A SYSTEM OF COMPLEX LINEAR ALGEBRAIC EQUATIONS, AX = B, AND OPTIONALLY IMPROVES THE SOLUTION AND COMPUTES AN ERROR BOUND FOR THE SOLUTION.

SAXB
SOLVES A SYSTEM OF REAL EQUATIONS, AX = B, USING GAUSSIAN ELIMINATION WITH IMPLICIT SCALING AND ROW PIVOTING. SAXB REQUIRES LESS TIME AND LESS SPACE THAN SAXBI BUT PROVIDES LESS ACCURACY.
(SAXB REPLACES SUBROUTINE AXB)

SAXBI
SOLVES A SYSTEM OF REAL EQUATIONS, AX = B, USING GAUSSIAN ELIMINATION WITH IMPLICIT SCALING, ROW PIVOTING, AND ITERATIVE IMPROVEMENT. SAXBI IS RECOMMENDED AS THE BEST CHOICE SINCE IT PROVIDES GREATER ACCURACY THAN SAXB AND ALSO PROVIDES AN ERROR ESTIMATE. SAXBI DOES REQUIRE MORE TIME AND SPACE THAN SAXB .
(SAXBI REPLACES SUBROUTINE AXBI)

RDET
EVALUATES THE DETERMINANT OF A REAL MATRIX. NOTE THAT SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS SHOULD ALWAYS BE SOLVED DIRECTLY USING SAXBI OR SAXB RATHER THAN USING THE SLOWER LESS ACCURATE CRAMERS RULE WITH DETERMINANT EVALUATION.

****
SYSTEMS OF NONLINEAR ALGEBRAIC EQUATIONS

QN
SOLVES A SYSTEM OF NONLINEAR ALGEBRAIC EQUATIONS

****
LINEAR PROGRAMMING AND GAME THEORY

LPGAME
LPGAME IS A MULTI-PURPOSE ROUTINE WHICH CAN BE USED TO SOLVE LINEAR PROGRAMMING PROBLEMS OR TWO-PERSON GAMES.

****
NUMERICAL QUADRATURE (NUMERICAL EVALUATION OF DEFINITE INTEGRALS)

AVINT

INTEGRATION OF TABULATED DATA. A METHOD OF OVERLAPPING PARABOLAS IS USED.

GAUS8
ADAPTIVE INTEGRATION USING 8 POINT GAUSS-LEGENDRE QUADRAOURE FOR HIGH ACCURACY OR FOR SMOOTH FUNCTIONS.

QNC3
ADAPTIVE INTEGRATION USING 3 POINT NEWTON COTES ALGORITHM (SIMPSON≠S RULE) FOR RELATIVELY LOW ACCURACY ON ROUGH FUNCTIONS.

QNC7
ADAPTIVE INTEGRATION USING 7 POINT NEWTON COTES ALGORITHM FOR MODERATE ACCURACY. QNC7 IS OFTEN THE PREFERABLE CHOICE FOR A WIDE CLASS OF FUNCTIONS AND ACCURACIES ON THE CDC 6600.

SICONT
INTEGRATION OF FUNCTIONS CONTAINING AN EXPLICIT SIN(WT) OR COS(WT) IN THE INTEGRAND.

SPLIQ
INTEGRATES A CUBIC SPLINE (DEFINED BY SPLIFT, SMOOTH, ETC.)

****

NUMERICAL SORTING

SSORT
SORTS AN ARRAY OF REAL VALUES IN EITHER ASCENDING OR DESCENDING NUMERICAL ORDER AND OPTIONALLY CARRIES ALONG A SECOND ARRAY OF REAL VALUES.

****

ORDINARY DIFFERENTIAL EQUATIONS

GERK
GERK IS DESIGNED TO SOLVE SYSTEMS OF DIFFERENTIAL EQUATIONS WHEN IT IS IMPORTANT TO HAVE A READILY AVAILABLE GLOBAL ERROR ESTIMATE.

ODE VARIABLE STEP-SIZE, VARIABLE ORDER PREDICTOR CORRECTOR METHOD. THIS THE RECOMMENDED ROUTINE.

ODERT
INTEGRATES A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS. INTEGRATION CONTINUES UNTIL A ZERO OF A USER DEFINED LINEAR OR NONLINEAR FUNCTION OF THE DEPENDENT VARIABLES IS LOCATED.

RKF
RKF IS A RUNGE-KUTTA-FEHLBERG SCHEME FOR SOLVING NON-STIFF DIFFERENTIAL EQUATIONS WHEN DERIVATIVE EVALUATIONS ARE CHEAP. RKF SHOULD GENERALLY NOT BE USED WHEN HIGH ACCURACY IS DEMANDED. SUBROUTINE ODE IS PREFERRED IN THESE CASES.

STEP1
SUBROUTINE STEP1 IS NORMALLY USED INDIRECTLY THROUGH SUBROUTINE ODE. BECAUSE ODE SUFFICES FOR MOST PROBLEMS AND IS MUCH EASIER TO USE, USING ODE SHOULD BE CONSIDERED BEFORE USING STEP1 ALONE.

SUPORT
SOLVES A LINEAR TWO-POINT BOUNDARY VALUE PROBLEM.

****

SPECIAL FUNCTIONS

BESI
BESI COMPUTES AN N MEMBER SEQUENCE OF I BESSEL FUNCTIONS
I/SUB(ALPHA+K-1)/(X), K=1,2,...,N OR SCALED BESSEL FUNCTIONS FOR
NON-NEGATIVE ALPHA AND X.

BESI01
I BESSEL FUNCTIONS OF ORDER ONE OR TWO FOR REAL ARGUMENTS

BESK01
K BESSEL FUNCTIONS OF ORDER ONE OR TWO FOR REAL ARGUMENTS

BESY01
Y BESSEL FUNCTIONS OF ORDER ONE OR TWO FOR REAL ARGUMENTS

BESJ01
J BESSEL FUNCTIONS OF ORDER ONE OR TWO FOR REAL ARGUMENTS

BESJ
BESJ COMPUTES AN N MEMBER SEQUENCE OF J BESSEL FUNCTIONS
J/SUB(ALPHA+K-1)/(X), K=1,2,...,N FOR NON-NEGATIVE ALPHA AND X.

BESKN
BESKN COMPUTES AN N MEMBER SEQUENCE OF INTEGER ORDER K BESSEL FUNCTIONS
K/SUB(NU+I-1)/(X), OR SCALED BESSEL FUNCTIONS, FOR REAL X .GT. 0 AND A
NON-NEGATIVE INTEGER NU.

BESYN
BESYN COMPUTES AN N MEMBER SEQUENCE OF INTEGER ORDER Y BESSEL FUNCTIONS
Y/SUB(NU+I-1)/(X), OR SCALED BESSEL FUNCTIONS, FOR REAL X .GT. 0 AND A
NON-NEGATIVE INTEGER NU.

COSH
HYPERBOLIC COSINE FUNCTION.

ERF
THE ERROR FUNCTION 2/SQRT(PI) *(INTEGRAL FROM 0 TO X OF
EXP(-T**2) DT)

ERFC
THE COMPLEMENTARY ERROR FUNCTION 2/SQRT(PI) *(INTEGRAL FROM X TO
INFINITY OF EXP(-T**2) DT)
ERFC CAN ALSO BE USED TO EVALUATE THE NORMAL PROBABILITY INTEGRAL.

GAMLN
GAMLN COMPUTES THE NATURAL LOG OF THE GAMMA FUNCTION FOR REAL POSITIVE
ARGUMENTS.

GAMMA
GAMMA FUNCTION FOR A REAL ARGUMENT (ACTUALLY A SUBROUTINE).

GAMMAZ
GAMMA FUNCTION FOR A COMPLEX ARGUMENT (ACTUALLY A SUBROUTINE).

SINH
HYPERBOLIC SINE FUNCTION.

****
ZEROS OF FUNCTIONS AND OPTIMIZATION

MINA
SEARCHES FOR A MINIMUM OF A REAL VALUED FUNCTION OF SEVERAL VARIABLES IN A
REGION.

SIMIN
MINIMIZES A REAL FUNCTION OF TWO OR MORE REAL VARIABLES

ZEROIN
SEARCHES FOR A ZERO OF A REAL VALUED FUNCTION OF ONE VARIABLE IN AN

INTERVAL USING AN EFFICIENT COMBINATION OF BISECTION AND SECANT METHOD.

ALSO SEE TJMAF1 LISTED UNDER DATA FITTING FOR SOLVING SYSTEMS OF NONLINEAR ALGEBRAIC EQUATIONS AND FOR MINIMIZING THE SUM OF SQUARES OF RESIDUALS.

****

ZEROS OF POLYNOMIALS

CBND2

COMPUTES A POSTERIORI ERROR BOUNDS AND CLUSTER COUNTS FOR APPROXIMATE ZEROS OF A POLYNOMIAL WITH COMPLEX COEFFICIENTS. IT IS RECOMMENDED THAT CBND2 BE USED TO DETERMINE THE ACCURACY AND PROBABLE MULTIPLICITY OF ZEROS COMPUTED BY CPQR.

CPQR

COMPUTES ALL OF THE ZEROS (BOTH REAL AND COMPLEX) OF A POLYNOMIAL WITH COMPLEX COEFFICIENTS AND OF DEGREE LESS THAN 20.

RBND2

COMPUTES A POSTERIORI ERROR BOUNDS AND CLUSTER COUNTS FOR APPROXIMATE ZEROS OF A POLYNOMIAL WITH REAL COEFFICIENTS. IT IS RECOMMENDED THAT RBND2 BE USED TO DETERMINE THE ACCURACY AND PROBABLE MULTIPLICITY OF ZEROS COMPUTED BY RPQR.

RPQR

COMPUTES ALL OF THE ZEROS (BOTH REAL AND COMPLEX) OF A POLYNOMIAL WITH REAL COEFFICIENTS AND DEGREE LESS THAN 20.

****

LIBRARY ERROR CHECK ROUTINE AND USER OPTIONS

ERXSET

ERXSET SETS THE STATE OF TWO PARAMETERS IN THE LIBRARY ERROR CHECK ROUTINE WHICH CONTROL THE PRINTING OF DIAGNOSTIC MESSAGES AND THE TERMINATION OF EXECUTION OF THE USER≠S PROGRAM. IN PARTICULAR, BY CALLING ERXSET THE USER MAY MAKE MATHLIB MESSAGES NONFATAL. (SEE ERRCHK FOR USAGE INFORMATION.)

ERRCHK

ERRCHK PROCESSES MESSAGES FROM OTHER ROUTINES IN THE MATHLIB FILE. SUCH MESSAGES ARE NORMALLY FATAL ERRORS UNLESS THE NONFATAL MODE WAS SELECTED PREVIOUSLY BY CALLING ERRSET. USUALLY ERRCHK IS NOT CALLED DIRECTLY BY THE USER.

ERRGET

ERRGET RETURNS THE VALUES OF TWO PARAMETERS CONTAINED WITHIN THE LIBRARY ERROR CHECK ROUTINE. THIS, TOGETHER WITH ERRSET, PERMITS THE USER TO DETERMINE THE STATE OF THE PARAMETERS, TO CHANGE THEM, AND THEN TO RESTORE THEM TO THEIR ORIGINAL STATE. (SEE ERRCHK FOR USAGE INFORMATION.)

5. CALLING SEQUENCES FOR SUBROUTINES (ALPHABETICAL ORDER)

AVINT AVINT AVINT AVINT AVINT AVINT AVINT AVINT

```
*****************************************
*****************************
*******************
**********
```

```
SUBROUTINE AVINT (X,Y,N,XLO,XUP,ANS,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
       R. E. HUDDLESTON  -   DIVISION 8322
       T. H. JEFFERSON   -   DIVISION 8322

ORIGINAL PROGRAM FROM *NUMERICAL INTEGRATION* BY DAVIS+RABINOWITZ.
ADAPTATION AND MODIFICATIONS FOR SANDIA MATHEMATICAL PROGRAM
LIBRARY BY RONDALL E JONES.

ABSTRACT
    AVINT INTEGRATES A FUNCTION TABULATED AT ARBITRARILY SPACED
    ABSCISSAS.  THE LIMITS OF INTEGRATION NEED NOT COINCIDE
    WITH THE TABULATED ABSCISSAS.

    A METHOD OF OVERLAPPING PARABOLAS FITTED TO THE DATA IS USED
    PROVIDED THAT THERE ARE AT LEAST 3 ABSCISSAS BETWEEN THE
    LIMITS OF INTEGRATION.  AVINT ALSO HANDLES TWO SPECIAL CASES.
    IF THE LIMITS OF INTEGRATION ARE EQUAL, AVINT RETURNS A RESULT
    OF ZERO REGARDLESS OF THE NUMBER OF TABULATED VALUES.
    IF THERE ARE ONLY TWO FUNCTION VALUES, AVINT USES THE
    TRAPEZOID RULE.

DESCRIPTION OF PARAMETERS
    THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
         X(N), Y(N)

    INPUT--
    X    - REAL ARRAY OF ABSCISSAS, WHICH MUST BE IN INCREASING
           ORDER.
    Y    - REAL ARRAY OF FUNCTIONAL VALUES. I.E., Y(I)=FUNC(X(I))
    N    - THE INTEGER NUMBER OF FUNCTION VALUES SUPPLIED.
           N .GE. 2 UNLESS XLO = XUP.
    XLO  - REAL LOWER LIMIT OF INTEGRATION
    XUP  - REAL UPPER LIMIT OF INTEGRATION.  MUST HAVE XLO.LE.XUP.

    OUTPUT--
    ANS  - COMPUTED APPROXIMATE VALUE OF INTEGRAL
    IERR - A STATUS CODE
         --NORMAL CODE
           =1 MEANS THE REQUESTED INTEGRATION WAS PERFORMED.
         --ABNORMAL CODES
           =2 MEANS XUP WAS LESS THAN XLO.
           =3 MEANS THE NUMBER OF X(I) BETWEEN XLO AND XUP
              (INCLUSIVE) WAS LESS THAN 3 AND NEITHER OF THE TWO
              SPECIAL CASES DESCRIBED IN THE ABSTRACT OCCURRED.
              NO INTEGRATION WAS PERFORMED.
           =4 MEANS THE RESTRICTION X(I+1).GT.X(I) WAS VIOLATED.
           =5 MEANS THE NUMBER N OF FUNCTION VALUES WAS .LT. 2.
           ANS IS SET TO ZERO IF IERR=2,3,4,OR 5.

AVINT IS DOCUMENTED COMPLETELY IN SC-M-69-335
```

```
BESI    BESI    BESI    BESI    BESI       BESI    BESI    BESI    BESI
              ************************************
                     ******************************
                          *********************
                               **********
      SUBROUTINE BESI(X,ALPHA,KODE,N,Y,NZ)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY D. E. AMOS AND S. L. DANIEL, JANUARY,1975.

      REFERENCE SAND-75-0152

      ABSTRACT
          BESI COMPUTES AN N MEMBER SEQUENCE OF I BESSEL FUNCTIONS
          I/SUB(ALPHA+K-1)/(X), K=1,...,N OR SCALED BESSEL FUNCTIONS
          EXP(-X)*I/SUB(ALPHA+K-1)/(X), K=1,...,N FOR NON-NEGATIVE ALPHA
          AND X. A COMBINATION OF THE POWER SERIES, THE ASYMPTOTIC
          EXPANSION FOR X TO INFINITY, AND THE UNIFORM ASYMPTOTIC
          EXPANSION FOR NU TO INFINITY ARE APPLIED OVER SUBDIVISIONS OF
          THE (NU,X) PLANE. FOR VALUES NOT COVERED BY ONE OF THESE
          FORMULAE, THE ORDER IS INCREMENTED BY AN INTEGER SO THAT ONE
          OF THESE FORMULAE APPLY. BACKWARD RECURSION IS USED TO REDUCE
          ORDERS BY INTEGER VALUES. THE ASYMPTOTIC EXPANSION FOR X TO
          INFINITY IS USED ONLY WHEN THE ENTIRE SEQUENCE (SPECIFICALLY
          THE LAST MEMBER) LIES WITHIN THE REGION COVERED BY THE
          EXPANSION. LEADING TERMS OF THESE EXPANSIONS ARE USED TO TEST
          FOR OVER OR UNDERFLOW WHERE APPROPRIATE. IF A SEQUENCE IS
          REQUESTED AND THE LAST MEMBER WOULD UNDERFLOW, THE RESULT IS
          SET TO ZERO AND THE NEXT LOWER ORDER TRIED, ETC., UNTIL A
          MEMBER COMES ON SCALE OR ALL ARE SET TO ZERO. AN OVERFLOW
          CANNOT OCCUR WITH SCALING. BESI CALLS FUNCTION GAMLN.

      DESCRIPTION OF ARGUMENTS

          INPUT
            X       - X.GE.0
            ALPHA   - ORDER OF FIRST MEMBER OF THE SEQUENCE, ALPHA.GE.0
            KODE    - A PARAMETER TO INDICATE THE SCALING OPTION
                      KODE=1 RETURNS
                             Y(K)=        I/SUB(ALPHA+K-1)/(X),
                                  K=1,...,N
                      KODE=2 RETURNS
                             Y(K)=EXP(-X)*I/SUB(ALPHA+K-1)/(X),
                                  K=1,...,N
            N       - NUMBER OF MEMBERS IN THE SEQUENCE, N.GE.1

          OUTPUT
            Y       - A VECTOR WHOSE FIRST N COMPONENTS CONTAIN
                      VALUES FOR I/SUB(ALPHA+K-1)/(X) OR SCALED
                      VALUES FOR EXP(-X)*I/SUB(ALPHA+K-1)/(X),
                      K=1,...,N DEPENDING ON KODE
            NZ      - NUMBER OF COMPONENTS OF Y SET TO ZERO DUE TO
                      UNDERFLOW,
                      NZ=0    , NORMAL RETURN, COMPUTATION COMPLETED
                      NZ.NE.0, LAST NZ COMPONENTS OF Y SET TO ZERO,
                               Y(K)=0., K=N-NZ+1,...,N.

      ERROR CONDITIONS
          IMPROPER INPUT ARGUMENTS - A FATAL ERROR
          OVERFLOW WITH KODE=1 - A FATAL ERROR
          UNDERFLOW - A NON-FATAL ERROR(NZ.NE.0)
```

```
BESI01    BESI01    BESI01    BESI01         BESI01    BESI01    BESI01
              ****************************************
                   ******************************
                        ********************
                             **********
     FUNCTICN BESI01(X,NU,KODE)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974.

     REFERENCE SAND-75-0149

     ABSTRACT
         BESI01 COMPUTES BESSEL FUNCTIONS I/SUB(NU)/(X), NU=0 OR 1
         OR SCALED BESSEL FUNCTIONS EXP(-ABS(X))*I/SUB(NU)/(X),
         NU=0 OR 1 FOR REAL X. CHEBYSHEV SUMS, ASYMPTOTICALLY SCALED
         FOR SMALL AND LARGE ABS(X), ARE USED ON INTERVALS 0.LE.X.LE.4,
         4.LT.X.LE.8, AND X.GT.8. THE SIGN IS FIXED ACCORDING TO THE
         EVENNESS OR ODDNESS OF THE FUNCTION. THE OVERFLOW TEST IS MADE
         ON ABS(X).LE.ELIM WITH ELIM=667.

     DESCRIPTICN OF ARGUMENTS

         INPUT
           X      - ABS(X).LE.667. FOR KODE=1, UNRESTRICTED FOR KODE=2
           NU     - ORDER DESIRED, NU=0 OR 1
           KODE   - A PARAMETER TC INDICATE THE SCALING OPTION
                    KODE=1 RETURNS ANS=        I/SUB(NU)/(X), NU=0 OR 1
                    KODE=2 RETURNS ANS=EXP(-X)*I/SUB(NU)/(X), NU=0 OR 1

         OUTPUT
           BESI01 - I BESSEL FUNCTION OF ORDER NU AT X SCALED ACCORDING
                    TO KODE

     ERROR CONDITIONS
         IMPROPER INPUT ARGUMENTS - A FATAL ERROR
         OVERFLOW FCR KODE=1 - A FATAL ERROR
```

```
BESJ    BESJ    BESJ    BESJ    BESJ       BESJ    BESJ    BESJ    BESJ
              ****************************************
                   ******************************
                        ********************
                             **********
     SUBROUTINE BESJ(X,ALPHA,N,Y,NZ)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY D.E. AMOS, S.L. DANIEL AND M.K. WESTON, JANUARY, 1975.

     REFERENCE SAND-75-0147

     ABSTRACT
         BESJ COMPUTES AN N MEMBER SEQUENCE OF J BESSEL FUNCTIONS
         J/SUB(ALPHA+K-1)/(X), K=1,...,N FOR NON-NEGATIVE ALPHA AND X.
         A COMBINATION OF THE POWER SERIES, THE ASYMPTOTIC EXPANSION
         FOR X TO INFINITY AND THE UNIFORM ASYMPTOTIC EXPANSICN FOR
         NU TO INFINITY ARE APPLIED OVER SUBDIVISIONS OF THE (NU,X)
         PLANE. FOR VALUES OF (NU,X) NOT COVERED BY ONE OF THESE
         FORMULAE, THE ORDER IS INCREMENTED OR DECREMENTED BY INTEGER
```

```
        VALUES INTO A REGION WHERE ONE OF THE FORMULAE APPLY. BACKWARD
        RECURSION IS APPLIED TO REDUCE ORDERS BY INTEGER VALUES EXCEPT
        WHERE THE ENTIRE SEQUENCE LIES IN THE OSCILLATORY REGION. IN
        THIS CASE FORWARD RECURSION IS STABLE AND VALUES FROM THE
        ASYMPTOTIC EXPANSION FOR X TO INFINITY START THE RECURSION
        WHEN IT IS EFFICIENT TO DO SO. LEADING TERMS OF THE SERIES AND
        UNIFORM EXPANSION ARE TESTED FOR UNDERFLOW. IF A SEQUENCE IS
        REQUESTED AND THE LAST MEMBER WOULD UNDERFLOW, THE RESULT IS
        SET TO ZERO AND THE NEXT LOWER ORDER TRIED, ETC., UNTIL A
        MEMBER COMES ON SCALE OR ALL MEMBERS ARE SET TO ZERO. OVERFLOW
        CANNOT OCCUR. BESJ CALLS SUBROUTINE JAIRY AND FUNCTION GAMLN.

    DESCRIPTION OF ARGUMENTS

        INPUT
          X       - X.GE.0
          ALPHA   - ORDER OF FIRST MEMBER OF THE SEQUENCE, ALPHA.GE.0
          N       - NUMBER OF MEMBERS IN THE SEQUENCE, N.GE.1

        OUTPUT
          Y       - A VECTOR WHOSE FIRST N COMPONENTS CONTAIN
                    VALUES FOR J/SUB(ALPHA+K-1)/(X), K=1,...,N
          NZ      - NUMBER OF COMPONENTS OF Y SET TO ZERO DUE TO
                    UNDERFLOW,
                    NZ=0    , NORMAL RETURN, COMPUTATION COMPLETED
                    NZ.NE.0, LAST NZ COMPONENTS OF Y SET TO ZERO,
                             Y(K)=0., K=N-NZ+1,...,N.

    ERROR CONDITIONS
        IMPROPER INPUT ARGUMENTS - A FATAL ERROR
        UNDERFLOW  - A NON-FATAL ERROR(NZ.NE.0)
```

```
BESJ01    BESJ01    BESJ01    BESJ01          BESJ01    BESJ01    BESJ01
              ****************************************
                     ******************************
                          ********************
                               **********
    FUNCTION BESJ01(X,NU)

    SANDIA MATHEMATICAL PROGRAM LIBRARY

        CONSULTANTS AT SLL INCLUDE -
            R. E. HUDDLESTON  -   DIVISION 8322
            T. H. JEFFERSON   -   DIVISION 8322

    WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974

    REFERENCE SAND-75-0148

    ABSTRACT
        BESJ01 COMPUTES BESSEL FUNCTIONS J/SUB(NU)/(X), NU=0 OR 1
        FOR REAL, UNRESTRICTED X. RATIONAL CHEBYSHEV APPROXIMATIONS,
        ASYMPTOTICALLY SCALED FOR SMALL AND LARGE ABS(X), ARE USED ON
        0.LE.X.LE.8, AND X.GT.8. THE SIGN IS FIXED ACCORDING TO THE
        EVENNESS OR ODDNESS OF THE FUNCTION.

    DESCRIPTION OF ARGUMENTS

        INPUT
          X       - UNRESTRICTED
          NU      - ORDER DESIRED, NU=0 OR 1

        OUTPUT
          BESJ01 - J BESSEL FUNCTION OF ORDER NU AT X

    ERROR CONDITIONS
        IMPROPER INPUT ARGUMENTS - A FATAL ERROR
```

```
BESKN    BESKN    BESKN    BESKN       BESKN    BESKN    BESKN    BESKN
              ***************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE BESKN (X,NU,KODE,N,Y,NZ)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974.

      REFERENCE SAND-75-0151

      ABSTRACT
          BESKN IMPLEMENTS FORWARD RECURSION ON THE THREE TERM
          RECURSION RELATION FOR A SEQUENCE OF INTEGER ORDER BESSEL
          FUNCTIONS K/SUB(NU+I-1)/(X), OR SCALED BESSEL FUNCTIONS EXP(X)
          *K/SUB(NU+I-1)/(X), I=1,...,N FOR REAL X.GT.0 AND A
          NON-NEGATIVE INTEGER NU.  IF NU.LT.NULIM, ORDERS 0 AND 1 ARE
          OBTAINED FROM FUNCTION BESK01 TO START THE RECURSION. IF
          NU.GE.NULIM, THE UNIFORM ASYMPTOTIC EXPANSION IS USED FOR
          ORDERS NU AND NU+1 TO START THE RECURSION.  NULIM IS 35 OR
          70 DEPENDING ON WHETHER N=1 OR N.GE.2.  UNDER AND OVERFLOW
          TESTS ARE MADE ON THE LEADING TERM OF THE ASYMPTOTIC EXPANSION
          BEFORE ANY EXTENSIVE COMPUTATION IS DONE. BESKN CALLS FUNCTION
          BESK01 AND SUBROUTINE ASKBES. BESK01 CALLS BESI01.

      DESCRIPTION OF ARGUMENTS

          INPUT
            X       - X.GT.0
            NU      - ORDER OF THE INITIAL K FUNCTION, NU=0,1,2,...
            KODE    - A PARAMETER TO INDICATE THE SCALING OPTION
                      KODE=1 RETURNS Y(K)=       K/SUB(NU+I-1)/(X),
                                          I=1,...,N
                      KODE=2 RETURNS Y(K)=EXP(X)*K/SUB(NU+I-1)/(X),
                                          I=1,...,N
            N       - NUMBER OF MEMBERS IN THE SEQUENCE, N.GE.1

          OUTPUT
            Y       - A VECTOR WHOSE FIRST N COMPONENTS CONTAIN VALUES
                      FOR THE SEQUENCE
                      Y(I)=       K/SUB(NU+I-1)/(X), I=1,...,N  OR
                      Y(I)=EXP(X)*K/SUB(NU+I-1)/(X), I=1,...,N
                      DEPENDING ON KODE
            NZ      - NUMBER OF COMPONENTS OF Y SET TO ZERO DUE TO
                      UNDERFLOW WITH KODE=1,
                      NZ=0    , NORMAL RETURN, COMPUTATION COMPLETED
                      NZ.NE.0, FIRST NZ COMPONENTS OF Y SET TO ZERO
                               DUE TO UNDERFLOW, Y(K)=0., K=1,...,NZ

      ERROR CONDITIONS
          IMPROPER INPUT ARGUMENTS - A FATAL ERROR
          OVERFLOW - A FATAL ERROR
          UNDERFLOW WITH KODE=1 -  A NON-FATAL ERROR(NZ.NE.0)
```

```
BESK01    BESK01    BESK01    BESK01         BESK01    BESK01    BESK01
            ***************************************
                  ****************************
                       ********************
                            **********
      FUNCTION BESK01(X,NU,KODE,NZ)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974.

      REFERENCE SAND-75-0149

      ABSTRACT
          BESK01 COMPUTES BESSEL FUNCTIONS K/SUB(NU)/(X), NU=0 OR 1
          OR SCALED BESSEL FUNCTIONS EXP(X)*K/SUB(NU)/(X),
          NU=0 OR 1 FOR X.GT.0. CHEBYSHEV EXPANSIONS, PROPERLY SCALED
          FOR SMALL AND LARGE X, ARE USED ON INTERVALS 0.LT.X.LE.2,
          2.LT.X.LE.5, AND X.GT.5. THE UNDERFLOW TEST IS MADE ON
          X.LE.ELIM WITH ELIM=667. BESK01 CALLS FUNCTION BESI01.

      DESCRIPTION OF ARGUMENTS

          INPUT
            X       - 0.LT.X.LE.667. FOR KODE=1, X.GT.0 FOR KODE=2
            NU      - ORDER DESIRED, NU=0 OR 1
            KODE    - A PARAMETER TO INDICATE THE SCALING OPTION
                      KODE=1 RETURNS ANS=       K/SUB(NU)/(X), NU=0 OR 1
                      KODE=2 RETURNS ANS=EXP(X)*K/SUB(NU)/(X), NU=0 OR 1

          OUTPUT
            BESK01 - K BESSEL FUNCTION OF ORDER NU AT X SCALED ACCORDING
                      TO KODE
            NZ      - UNDERFLOW INDICATOR
                      NZ=0    , NORMAL RETURN, COMPUTATION COMPLETED
                      NZ.NE.0, ANS SET TO ZERO DUE TO UNDERFLOW WITH
                               KODE=1 AND X.GT.667
      ERROR CONDITIONS
          IMPROPER INPUT ARGUMENTS - A FATAL ERROR
          UNDERFLOW WITH KODE=1 -  A NON-FATAL ERROR(NZ.NE.0)
```

```
BESYN     BESYN     BESYN     BESYN          BESYN     BESYN     BESYN     BESYN
            ***************************************
                  ****************************
                       ********************
                            **********
      SUBROUTINE BESYN(X,NU,N,Y)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974.

      REFERENCE SAND-75-0150

      ABSTRACT
          BESYN IMPLEMENTS FORWARD RECURSION ON THE THREE TERM
          RECURSION RELATION FOR A SEQUENCE OF INTEGER ORDER BESSEL
          FUNCTIONS Y/SUB(NU+K-1)/(X), K=1,...,N FOR REAL X.GT.0 AND A
          NON-NEGATIVE INTEGER NU. IF NU.LT.NULIM, ORDERS 0 AND 1 ARE
          OBTAINED FROM FUNCTION BESY01 TO START THE RECURSION. IF
```

```
NU.GE.NULIM, THE UNIFORM ASYMPTOTIC EXPANSION IS USED FOR
ORDERS NU AND NU+1 TO START RECURSION.  NULIM=100 RESTRICTS
FORWARD RECURSION TO RETAIN ACCURACY.  AN OVERFLOW TEST IS
MADE ON THE LEADING TERM OF THE ASYMPTOTIC EXPANSION BEFORE
ANY EXTENSIVE COMPUTATION IS DONE. BESYN CALLS FUNCTION BESY01
AND SUBROUTINE ASYBES. BESY01 CALLS FUNCTION BESJ01. ASYBES
CALLS SUBROUTINE YBAIRY.

DESCRIPTION OF ARGUMENTS

    INPUT
      X       - X.GT.0
      NU      - ORDER OF THE INITIAL Y FUNCTION, NU=0,1,2,...
      N       - NUMBER OF MEMBERS IN THE SEQUENCE, N.GE.1

    OUTPUT
      Y       - A VECTOR WHOSE FIRST N COMPONENTS CONTAIN VALUES
                FOR Y(K)=Y/SUB(NU+K-1)/(X), K=1,...,N

ERROR CONDITIONS
    IMPROPER INPUT ARGUMENTS - A FATAL ERROR
    OVERFLOW - A FATAL ERROR
```

```
BESY01    BESY01    BESY01    BESY01         BESY01    BESY01    BESY01
          ****************************************
                    ******************************
                         ********************
                              **********

FUNCTION BESY01(X,NU,ANSJ)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974

REFERENCE SAND-75-0148

ABSTRACT
    BESY01 COMPUTES BESSEL FUNCTIONS J/SUB(NU)/(X) AND
    Y/SUB(NU)/(X), NU=0 OR 1 FOR X.GT.0. RATIONAL CHEBYSHEV
    APPROXIMATIONS, ASYMPTOTICALLY SCALED FOR SMALL AND LARGE X,
    ARE USED ON INTERVALS 0.LE.X.LE.8 AND X.GT.8. THE COST IN
    RETURNING J/SUB(NU)/(X) IS MINIMAL SINCE THIS FUNCTION IS
    NEEDED IN THE ASYMPTOTIC FORM FOR 0.LT.X.LE.8 AND ONLY
    REQUIRES A REARRANGEMENT OF 4 FACTORS NEEDED FOR X.GT.8.
    BESY01 CALLS FUNCTION BESJ01.

DESCRIPTION OF ARGUMENTS

    INPUT
      X       - X.GT.0.
      NU      - ORDER DESIRED, NU=0 OR 1

    OUTPUT
      BESY01  - Y BESSEL FUNCTION OF ORDER NU AT X
      ANSJ    - J BESSEL FUNCTION OF ORDER NU AT X

ERROR CONDITIONS
    IMPROPER INPUT ARGUMENTS - A FATAL ERROR
```

```
CAXBI    CAXBI    CAXBI    CAXBI       CAXBI    CAXBI    CAXBI    CAXBI
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE CAXBI(ND,N,M,A,B,X,INIT,IMP,RC,W,IN,KER)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY CARL B. BAILEY, AUGUST 1974.

      ABSTRACT
          CAXBI SOLVES A NONSINGULAR SYSTEM OF COMPLEX LINEAR ALGEBRAIC
          EQUATIONS, AX=B, AND OPTIONALLY IMPROVES THE SOLUTION AND
          COMPUTES AN ERROR BOUND FOR THE SOLUTION.  THE COEFFICIENT
          MATRIX FOR AN EQUIVALENT SYSTEM OF REAL EQUATIONS IS FORMED
          AND STORED IN -W- AND THEN THAT REAL SYSTEM IS SOLVED.
          THE METHOD USED IS LU DECOMPOSITION (GAUSSIAN ELIMINATION)
          WITH IMPLICIT ROW SCALING AND PARTIAL (ROW) PIVOTING FOLLOWED
          BY FORWARD-BACKWARD SUBSTITUTION AND OPTIONALLY BY ITERATIVE
          IMPROVEMENT.  A SEQUENCE OF SYSTEMS OF EQUATIONS ALL HAVING
          THE SAME COEFFICIENT MATRIX CAN BE SOLVED VERY EFFICIENTLY
          USING CAXBI.  THE LU FACTORS OF -A- ARE COMPUTED AND STORED
          IN -W- DURING THE INITIAL CALL.  ON SUBSEQUENT CALLS, THESE
          PREVIOUSLY COMPUTED FACTORS CAN BE USED TO SOLVE A NEW SYSTEM
          BY PERFORMING ONLY THE FORWARD-BACKWARD SUBSTITUTION AND
          OPTIONALLY ITERATIVE IMPROVEMENT.

          CAXBI CALLS THE ROUTINE RLUD TO PERFORM LU DECOMPOSITION, RFBS
          TO PERFORM FORWARD-BACKWARD SUBSTITUTION, AND OPTIONALLY CALLS
          CRIMP TO PERFORM ITERATIVE IMPROVEMENT OF THE SOLUTION.

      REFERENCE
        1.  G.E.FORSYTHE AND C.B.MOLER, COMPUTER SOLUTION OF LINEAR
            ALGEBRAIC EQUATIONS, PRENTICE-HALL, 1967

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               A(ND,N), B(ND,M), X(ND,M), W(2*ND,2*ND+1), IN(2*N)
          IF M=1 THEN THE DIMENSION OF B AND X MAY BE B(N), X(N).
          THE ARRAYS -A-, -B-, AND -X- MUST BE OF COMPLEX DATA TYPE.

        --INPUT FOR AN INITIAL CALL--
                    AN INITIAL CALL IS THE CALL FOR THE FIRST SYSTEM OF
                    EQUATIONS IN A SEQUENCE OF SYSTEMS ALL OF WHICH HAVE
                    THE SAME COEFFICIENT MATRIX.
          ND      - THE ACTUAL FIRST DIMENSION OF -A-.
                    (I.E. THE MAXIMUM NUMBER OF EQUATIONS THAT CAN BE
                    SOLVED USING -A- TO STORE THE COEFFICIENTS.)
          N       - THE NUMBER OF COMPLEX EQUATIONS TO BE SOLVED.
                    (1 .LE. N .LE. ND)
          M       - NUMBER OF COLUMNS OF -B- AND -X-.  (NORMALLY M=1)
          A       - A COMPLEX ARRAY DIMENSIONED WITH EXACTLY -ND- ROWS
                    AND AT LEAST -N- COLUMNS.  THE I,J-TH ELEMENT OF THE
                    COEFFICIENT MATRIX MUST BE STORED IN A(I,J).
          B       - A COMPLEX ARRAY WITH EITHER ONE OR TWO DIMENSIONS.
                    IF M=1, -B- MAY BE A ONE-DIMENSIONAL ARRAY
                    DIMENSIONED AT LEAST -N-.  THE I-TH ELEMENT OF THE
                    CONSTANT VECTOR MUST BE STORED IN B(I).
                    IF M.GT.1, -B- MUST BE A TWO-DIMENSIONAL ARRAY WITH
                    EXACTLY -ND- ROWS AND AT LEAST -M- COLUMNS.  THE
                    I,J-TH ELEMENT OF THE CONSTANT MATRIX MUST BE
                    STORED IN B(I,J).
          INIT    - IS A FLAG WHICH PROVIDES FOR THE ESPECIALLY EFFICIENT
                    SOLUTION OF A SEQUENCE OF SYSTEMS OF EQUATIONS HAVING
                    THE SAME -A- BUT DIFFERENT -B- VECTORS.
                    ON THE INITIAL CALL FOR A SEQUENCE OF RELATED SYSTEMS
                    OF EQUATIONS, -INIT- MUST BE ZERO.
          IMP     - SPECIFIES THAT ITERATIVE IMPROVEMENT IS TO BE
                    PERFORMED IF -IMP- IS NONZERO.
```

```
--OUTPUT--
  X       - A COMPLEX ARRAY WITH EITHER ONE OR TWO DIMENSIONS.
            IF M=1, -X- MAY BE A ONE-DIMENSIONAL ARRAY
            DIMENSIONED AT LEAST -N-.  THE I-TH ELEMENT OF THE
            SOLUTION VECTOR WILL BE STORED IN X(I).
            IF M.GT.1, -X- MUST BE A TWO-DIMENSIONAL ARRAY WITH
            EXACTLY -ND- ROWS AND AT LEAST -M- COLUMNS.  THE
            I,J-TH ELEMENT OF THE SOLUTION MATRIX WILL BE
            STORED IN X(I,J).
  RC      - IF ITERATIVE IMPROVEMENT WAS REQUESTED (IMP.NE.0), RC
            WILL BE THE RATIO OF THE MAXIMUM NORM OF THE FIRST
            CORRECTION TO THE MAXIMUM NORM OF THE INITIAL
            APPROXIMATE SOLUTION.  THE CONDITION NUMBER OF -A-
            AND ERROR BOUNDS FOR THE COMPUTED SOLUTION ARE
            RELATED TO -RC-.  A SMALL VALUE FOR -RC- INDICATES
            A WELL-CONDITIONED SYSTEM AND SMALL UNCERTAINTIES
            IN THE SOLUTION.  A LARGE VALUE FOR -RC- INDICATES
            AN ILL-CONDITIONED SYSTEM AND LARGE UNCERTAINTIES
            IN THE SOLUTION.
  W       - REAL ARRAY WITH -2ND- ROWS AND AT LEAST 2N+1 COLUMNS.
            THE LEADING -2N- BY -2N- SUBARRAY WILL CONTAIN L-I+U
            WHERE -L- AND -U- ARE TRIANGULAR FACTORS OF -A-,
            -L- IS UNIT LOWER TRIANGULAR, AND -I- IS IDENTITY.
            (ACTUALLY, IT IS NOT L-I+U WHICH IS STORED IN -A- BUT
            LL-I+U WHERE LL IS A REARRANGEMENT OF ELEMENTS OF L.)
            THE 2N+1ST COLUMN CONTAINS THE LAST CORRECTION TO THE
            REAL AND IMAGINARY COMPONENTS OF -X-.  IF INIT = 0,
            THE LU FACTORS OF THE REAL EQUIVALENT OF -A- WILL BE
            COMPUTED AND STORED IN -W-.
  IN      - WILL CONTAIN THE ROW INTERCHANGE INDICES THAT WERE
            COMPUTED DURING LU DECOMPOSITION.
  KER     - AN ERROR CODE
          --NORMAL CODES
            0 MEANS NO ERRORS WERE DETECTED
          --ABNORMAL CODES
            1 MEANS -ND- WAS NOT IN THE RANGE 1 .LT. ND .LE. 130
            2 MEANS -N- WAS NOT IN THE RANGE 1 .LE. N .LE. ND.
            3 MEANS THE TRIANGULAR FACTOR -U- OF -A- IS SINGULAR.
            4 MEANS -A- IS TOO ILL-CONDITIONED FOR ITERATIVE
              IMPROVEMENT TO BE EFFECTIVE.

--INPUT FOR A SUBSEQUENT CALL--
            A SUBSEQUENT CALL MAY BE MADE ONLY IF AN INITIAL CALL
            HAS BEEN MADE PREVIOUSLY FOR THE SAME COEFFICIENT
            MATRIX.  THE VALUES OF -ND-, -N-, -A-, -W-, AND -IN-
            MUST BE THE SAME AS THEY WERE WHEN THAT INITIAL CALL
            WAS COMPLETED.
  M       - MUST BE THE NUMBER OF COLUMNS IN THE NEW CONSTANT
            VECTOR OR CONSTANT MATRIX.
  B       - THE NEW CONSTANT VECTOR OR CONSTANT MATRIX MUST BE
            STORED IN -B- AS DESCRIBED FOR AN INITIAL CALL.
  INIT    - MUST BE NONZERO (ONLY FOR SUCH A SUBSEQUENT CALL).
            THIS CAUSES THE PREVIOUSLY COMPUTED LU FACTORS OF -A-
            TO BE USED TO SOLVE THE NEW SYSTEM VERY EFFICIENTLY.
  IMP     - MAY BE NONZERO OR ZERO AS ITERATIVE IMPROVEMENT
            IS OR IS NOT DESIRED RESPECTIVELY.
NOTE --- ND, N, M, A, B, INIT, AND IMP ARE NOT ALTERED BY CAXBI.
         THE USER MUST PROVIDE SEPARATE STORAGE FOR THE ARRAYS
         A, B, X, W, AND IN WHENEVER ITERATIVE IMPROVEMENT IS
         REQUESTED (IMP .NE. 0).  THE MAXIMUM NUMBER OF EQUATIONS
         THAT CAN BE SOLVED WITH ITERATIVE IMPROVEMENTS IS 130.
         IF ITERATIVE IMPROVEMENT IS NOT REQUESTED (IMP .EQ. 0),
         THEN THE USER MAY ECONOMIZE ON STORAGE BY EQUIVALENCING
         (A,W) AND (B,X) IN WHICH CASE A AND B WILL BE ALTERED.
         THE MAXIMUM NUMBER OF EQUATIONS THAT CAN BE SOLVED IN
         THIS LATTER CASE IS 160.
```

```
CBND2     CBND2     CBND2     CBND2        CBND2     CBND2     CBND2     CBND2
                 ************************************
                      ****************************
                           ********************
                                **********
      SUBROUTINE CBND2(N,CR,CI,WR,WI,ABSERR,RELERR,KLUST,KER)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY CARL B. BAILEY AND WILLIAM R. GAVIN

      ABSTRACT

          THIS ROUTINE COMPUTES ERROR BOUNDS AND CLUSTER COUNTS FOR
          APPROXIMATE ZEROS OF A POLYNOMIAL WITH COMPLEX COEFFICIENTS.
          THE ZEROS MAY HAVE BEEN COMPUTED BY ANY APPROPRIATE ROUTINE.
          (FOR EXAMPLE, BY CPQR)
          THE METHOD USED IS BASED ON THE FACT THAT THE VALUE OF A
          POLYNOMIAL AT ANY POINT IS EQUAL TO THE LEADING COEFFICIENT
          TIMES THE PRODUCT OF THE DISTANCES FROM THAT POINT TO EACH
          OF THE ZEROES.  GIVEN THE VALUE OF THE POLYNOMIAL AT AN
          APPROXIMATE ZERO, CBND2 COMPUTES FOR EACH APPROXIMATE ZERO
          THE RADIUS OF A CIRCLE ABOUT THAT APPROXIMATE ZERO WHICH
          CONTAINS A TRUE ZERO OF THE POLYNOMIAL.  USING THE KNOWN
          DISTRIBUTION OF APPROXIMATE ZEROES, AN ITERATIVE PROCEDURE
          IS USED TO SHRINK THE RADII OF THE CIRCLES.

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               CR(N+1), CI(N+1), WR(N), WI(N), ABSERR(N), RELERR(N)
               KLUST(N)

          INPUT---
          N       - DEGREE OF THE POLYNOMIAL (NUMBER OF ZEROS).
          CR      - REAL ARRAY OF N+1 REAL PARTS OF COEFFICIENTS.
          CI      - REAL ARRAY OF N+1 IMAGINARY PARTS OF COEFFICIENTS.
                    THE COEFFICIENTS COEF(I) = CR(I)+CI(I)*I MUST BE
                    IN THE ORDER OF DESCENDING POWERS OF Z, I.E.
                    P(Z) = (CR(1)+I*CI(1))*Z**N  + ... +
                           (CR(N)+I*CI(N))*Z + (CR(N+1)+I*CI(N+1))
          WR      - REAL ARRAY OF N REAL PARTS OF APPROXIMATE ZEROS.
          WI      - REAL ARRAY OF N IMAGINARY PARTS OF APPROXIMATE ZEROS.

          OUTPUT--
          ABSERR - REAL ARRAY OF ABSOLUTE ERROR BOUNDS.  ABSERR(I) IS
                   THE ABSOLUTE ERROR BOUND IN THE ZERO (WR(I),WI(I)).
          RELERR - REAL ARRAY OF RELATIVE ERROR BOUNDS.  RELERR(I) IS
                   THE RELATIVE ERROR BOUND IN THE ZERO (WR(I),WI(I)).
          KLUST  - INTEGER ARRAY OF CLUSTER COUNTS FOR ZEROS.  THE TRUE
                   ZERO CORRESPONDING TO I-TH APPROXIMATE ZERO LIES IN
                   A CIRCLE OF RADIUS ABSERR(I).  KLUST(I) IS THE NUMBER
                   OF CIRCLES INCLUDING THE I-TH CIRCLE WHICH OVERLAP
                   THE I-TH CIRCLE.  THE CLUSTER COUNT OFTEN INDICATES
                   THE MULTIPLICITY OF A ZERO.
          KER     - AN ERROR CODE
                  --NORMAL CODES
                    0  MEANS THE BOUNDS AND COUNTS WERE COMPUTED.
                  --ABNORMAL CODES
                    1  N (DEGREE) MUST BE .GE. 1
                    2  LEADING COEFFICIENT IS ZERO
```

```
CHAA    CHAA    CHAA    CHAA    CHAA       CHAA    CHAA    CHAA    CHAA
              ****************************************
                   ******************************
                        *******************
                             **********
      SUBROUTINE CHAA(NDIM,N,AR,AI,EV,VECR,VECI,IERR)
```

SANDIA MATHEMATICAL PROGRAM LIBRARY

CONSULTANTS AT SLL INCLUDE -
R. E. HUDDLESTON - DIVISION 8322
T. H. JEFFERSON - DIVISION 8322

EISPACK IS A COLLECTION OF CODES FOR SOLVING THE ALGEBRAIC EIGENVALUE PROBLEM. THE ORIGINAL ALGOL CODES WERE WRITTEN BY J. H. WILKINSON, ET.AL.,AND SUBSEQUENTLY TRANSLATED TO FORTRAN AND TESTED AT ARGONNE NATIONAL LABORATORY.

THIS INTERFACE TO EISPACK WAS WRITTEN BY M. K. GORDON.

ABSTRACT
THIS SUBROUTINE COMPUTES ALL EIGENVALUES AND CORRESPONDING EIGENVECTORS OF A COMPLEX HERMITIAN MATRIX. THE MATRIX IS REDUCED TO SYMMETRIC TRIDIAGONAL FORM BY UNITARY SIMILARITY TRANSFORMATIONS. QL TRANSFORMATIONS ARE USED TO FIND THE EIGENSYSTEM OF THE TRIDIAGONAL MATRIX.

TO COMPUTE ONLY THE EIGENVALUES OF A COMPLEX HERMITIAN MATRIX, SEE SUBROUTINE CHAN. FOR EIGENSYSTEMS OF ARBITRARY COMPLEX MATRICES, SEE SUBROUTINES CNAA AND CNAN. FOR EIGENSYSTEMS OF REAL MATRICES, SEE SUBROUTINES RSAA,RSAN,RNAA,RNAN.

DESCRIPTION OF ARGUMENTS

ON INPUT

NDIM — MUST BE THE ROW DIMENSION OF THE ARRAYS AR,AI,VECR, AND VECI IN THE CALLING PROGRAM DIMENSION STATEMENT.

N — IS THE ORDER OF THE MATRIX. N MUST NOT EXCEED NDIM. N*NDIM MUST NOT EXCEED 22500=150*150=53744(OCTAL). N MUST NOT EXCEED 150. N MAY BE 1.

AR,AI — ARRAYS WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS. THE LEADING N BY N SUBARRAYS MUST CONTAIN THE REAL AND IMAGINARY PARTS RESPECTIVELY OF THE COMPLEX HERMITIAN MATRIX WHOSE EIGENSYSTEM IS TO BE COMPUTED. ONLY THE DIAGONAL AND LOWER TRIANGLE OF AR,AI NEED BE DEFINED.

ON OUTPUT

EV — CONTAINS THE REAL COMPUTED EIGENVALUES IN ASCENDING ORDER.

VECR,VECI — CONTAIN AN ORTHONORMAL SET OF EIGENVECTORS IN THE COLUMNS OF THE LEADING N BY N SUBARRAYS. THE J-TH COLUMNS OF VECR,VECI CONTAIN AN EIGENVECTOR OF LENGTH ONE CORRESPONDING TO THE EIGENVALUE IN THE J-TH ELEMENT OF EV.

IERR — IS A STATUS CODE.
--NORMAL CODE.
0 MEANS THE QL ITERATIONS CONVERGED.

--ABNORMAL CODES.
J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN 30 ITERATIONS. THE FIRST J-1 ELEMENTS CONTAIN THE UNORDERED EIGENVALUES ALREADY FOUND. THE FIRST J-1 COLUMNS OF VECR, VECI CONTAIN THE CORRESPONDING COMPUTED EIGENVECTORS.
-1 MEANS THE INPUT VALUES OF N, NDIM ARE TOO LARGE OR INCONSISTENT.

AR,AI — THE LOWER TRIANGLES OF BOTH MATRICES AND THE

DIAGONAL OF AI ARE DESTROYED. THE UPPER
TRIANGLES AND THE DIAGONAL OF AR ARE UNALTERED.

```
CHAN    CHAN    CHAN    CHAN    CHAN       CHAN    CHAN    CHAN    CHAN
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE CHAN(NDIM,N,AR,AI,EV,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

       EISPACK IS A COLLECTION OF CODES FOR SOLVING THE ALGEBRAIC
       EIGENVALUE PROBLEM.  THE ORIGINAL ALGOL CODES WERE WRITTEN BY
       J. H. WILKINSCN, ET.AL., AND SUBSEQUENTLY TRANSLATED TO FORTRAN
       AND TESTED AT ARGONNE NATIONAL LABORATORY.

       THIS INTERFACE TO EISPACK WAS WRITTEN BY M. K. GORDON.

       ABSTRACT
         THIS SUBROUTINE COMPUTES ALL EIGENVALUES OF A COMPLEX HERMITIAN
         MATRIX.  THE MATRIX IS REDUCED TO SYMMETRIC TRIDIAGONAL FORM
         BY UNITARY SIMILARITY TRANSFORMATIONS.  QL TRANSFORMATIONS
         ARE USED TO FIND THE EIGENVALUES OF THE TRIDIAGONAL MATRIX.

         TO COMPUTE THE EIGENVALUES AND EIGENVECTORS OF A COMPLEX
         HERMITIAN MATRIX, SEE SUBROUTINE CHAA.  FOR EIGENSYSTEMS
         OF ARBITRARY COMPLEX MATRICES, SEE CNAA AND CNAN.  FOR EIGEN-
         SYSTEMS OF REAL MATRICES, SEE RSAA,RSAN,RNAA,RNAN.

       DESCRIPTION OF ARGUMENTS
         ON INPUT
            NDIM     MUST BE THE ROW DIMENSION OF AR AND AI IN THE
                     CALLING PROGRAM DIMENSION STATEMENT.

            N        IS THE ORDER OF THE MATRIX.  N MUST NOT EXCEED NDIM.
                     N*NDIM MUST NOT EXCEED 50625=225*225=142701(OCTAL).
                     N MUST NOT EXCEED 225.  N MAY BE 1.

            AR,AI    ARRAYS WITH EXACTLY NDIM ROWS AND AT LEAST N
                     COLUMNS.  THE LEADING N BY N SUBARRAYS MUST
                     CONTAIN THE REAL AND IMAGINARY PARTS RESPECTIVELY
                     OF THE ARBITRARY COMPLEX MATRIX WHOSE EIGENVALUES
                     ARE TO BE COMPUTED.  ONLY THE DIAGONALS AND
                     LOWER TRIANGLES OF AR,AI NEED BE DEFINED.

         ON OUTPUT
            EV       CONTAINS THE REAL COMPUTED EIGENVALUES IN
                     ASCENDING ORDER.

            IERR     IS A STATUS CODE.
                   --NORMAL CODE.
                     0 MEANS THE QL ITERATIONS CONVERGED.
                   --ABNORMAL CODES.
                     J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN
                       30 ITERATIONS.  THE FIRST J-1 ELEMENTS OF EV
                       CONTAIN THE UNORDERED EIGENVALUES ALREADY FOUND.
                    -1 MEANS THE INPUT VALUES OF N, NDIM ARE TOO LARGE
                       OR INCONSISTENT.

            AR,AI    THE LOWER TRIANGLES OF BOTH MATRICES AND THE
                     DIAGONAL OF AI ARE DESTROYED.  THE UPPER TRIANGLES
                     AND THE DIAGONAL OF AR ARE UNALTERED.
```

```
CNAA    CNAA    CNAA    CNAA    CNAA       CNAA    CNAA    CNAA    CNAA
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE CNAA(NDIM,N,AR,AI,EVR,EVI,VECR,VECI,IERR)
```

SANDIA MATHEMATICAL PROGRAM LIBRARY

CONSULTANTS AT SLL INCLUDE -
R. E. HUDDLESTON - DIVISION 8322
T. H. JEFFERSON - DIVISION 8322

EISPACK IS A COLLECTION OF CODES FOR SOLVING THE ALGEBRAIC EIGENVALUE PROBLEM. THE ORIGINAL ALGOL CODES WERE WRITTEN BY J. H. WILKINSON, ET.AL.,AND SUESEQUENTLY TRANSLATED TO FORTRAN AND TESTED AT ARGONNE NATIONAL LABORATORY.

THIS INTERFACE TO EISPACK WAS WRITTEN BY M. K. GORDON.

ABSTRACT

THIS SUBROUTINE COMPUTES ALL EIGENVALUES AND CORRESPONDING EIGENVECTORS OF AN ARBITRARY COMPLEX MATRIX. THE MATRIX IS BALANCED BY EXACT NORM REDUCING SIMILARITY TRANSFORMATIONS AND THEN IS REDUCED TO COMPLEX HESSENBERG FORM BY STABILIZED ELEMENTARY SIMILARITY TRANSFORMATIONS. A MODIFIED LR ALGORITHM IS USED TO COMPUTE THE EIGENVALUES OF THE HESSENBERG MATRIX.

TO COMPUTE ONLY THE EIGENVALUES OF AN ARBITRARY COMPLEX MATRIX, SEE SUBROUTINE CNAN. FOR EIGENSYSTEMS OF COMPLEX HERMITIAN MATRICES, SEE SUBROUTINES CHAA AND CHAN. FOR EIGENSYSTEMS OF REAL MATRICES, SEE SUBROUTINES RSAA,RSAN, RNAA,RNAN.

DESCRIPTION OF ARGUMENTS

ON INPUT

NDIM — MUST BE THE ROW DIMENSION OF THE ARRAYS AR,AI, VECR, AND VECI IN THE CALLING PROGRAM DIMENSION STATEMENT.

N — IS THE ORDER OF THE MATRIX. N MUST NOT EXCEED NDIM. N*NDIM MUST NOT EXCEED 22500=150*150=53744(OCTAL). N MUST NOT EXCEED 150. N MAY BE 1.

AR,AI — ARRAYS WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS. THE LEADING N BY N SUBARRAYS MUST CONTAIN THE REAL AND IMAGINARY PARTS RESPECTIVELY OF THE ARBITRARY COMPLEX MATRIX WHOSE EIGENSYSTEM IS TO BE COMPUTED.

ON OUTPUT

EVR,EVI — CONTAIN THE REAL AND IMAGINARY PARTS RESPECTIVELY OF THE COMPUTED EIGENVALUES. THE EIGENVALUES ARE NOT ORDERED IN ANY WAY.

VECR,VECI — CONTAIN IN THE LEADING N BY N SUBARRAYS THE REAL AND IMAGINARY PARTS RESPECTIVELY OF THE COMPUTED EIGENVECTORS. THE J-TH COLUMNS OF VECR AND VECI CONTAIN THE EIGENVECTOR ASSOCIATED WITH EVR(J) AND EVI(J). THE EIGENVECTORS ARE NOT NORMALIZED IN ANY WAY.

IERR — IS A STATUS CODE.
--NORMAL CODE.
0 MEANS THE LR ITERATIONS CONVERGED.
--ABNORMAL CODES.
J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN 30 ITERATIONS. THE FIRST J-1 ELEMENTS OF EVR AND EVI CONTAIN THOSE EIGENVALUES ALREADY FOUND. NO EIGENVECTORS ARE COMPUTED.
-1 MEANS THE INPUT VALUES OF N, NDIM ARE TOO LARGE OR INCONSISTENT.

```
AR,AI       ARE DESTRCYED.
```

```
CNAN    CNAN    CNAN    CNAN    CNAN       CNAN    CNAN    CNAN    CNAN
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE CNAN(NDIM,N,AR,AI,EVR,EVI,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      EISPACK IS A COLLECTION OF CODES FOR SOLVING THE ALGEBRAIC
      EIGENVALUE PROBLEM.  THE ORIGINAL ALGOL CODES WERE WRITTEN BY
      J. H. WILKINSON, ET.AL.,AND SUBSEQUENTLY TRANSLATED TO FORTRAN
      AND TESTED AT ARGONNE NATIONAL LABORATORY.

      THIS INTERFACE TO EISPACK WAS WRITTEN BY M. K. GORDON.

      ABSTRACT
        THIS SUBROUTINE COMPUTES ALL EIGENVALUES OF AN ARBITRARY
        COMPLEX MATRIX.  THE MATRIX IS BALANCED BY EXACT NORM REDUCING
        SIMILARITY TRANSFORMATIONS AND IS THEN REDUCED TO COMPLEX
        HESSENBERG FORM BY STABILIZED ELEMENTARY SIMILARITY
        TRANSFORMATIONS.  A MODIFIED LR ALGORITHM IS USED TO COMPUTE
        THE EIGENVALUES OF THE HESSENBERG MATRIX.

        TO COMPUTE THE EIGENVALUES AND EIGENVECTORS OF AN ARBITRARY
        COMPLEX MATRIX, SEE SUBROUTINE CNAA.  FOR EIGENSYSTEMS OF
        COMPLEX HERMITIAN MATRICES, SEE SUBROUTINES CHAN AND CHAA.
        FOR EIGENSYSTEMS OF REAL MATRICES, SEE RSAA,RSAN,RNAA,RNAN.

      DESCRIPTION OF ARGUMENTS

        ON INPUT
           NDIM     MUST BE THE ROW DIMENSION OF THE ARRAYS AR AND AI
                    IN THE CALLING PROGRAM DIMENSION STATEMENT.

           N        IS THE ORDER OF THE MATRIX.  N MUST NOT EXCEED NDIM.
                    N*NDIM MUST NOT EXCEED 50625=225*225=142701(OCTAL).
                    N MUST NOT EXCEED 225.  N MAY BE 1.

           AR,AI    ARRAYS WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS.
                    THE LEADING N BY N SUBARRAYS MUST CONTAIN THE REAL
                    AND IMAGINARY PARTS RESPECTIVELY OF THE ARBITRARY
                    COMPLEX MATRIX WHOSE EIGENVALUES ARE TO BE COMPUTED.

        ON OUTPUT
           EVR,EVI  CONTAIN THE REAL AND IMAGINARY PARTS RESPECTIVELY
                    OF THE COMPUTED EIGENVALUES.  THE EIGENVALUES ARE
                    NOT ORDERED IN ANY WAY.

           IERR     IS A STATUS CODE.
                  --NORMAL CODE.
                    0 MEANS THE LR ITERATIONS CONVERGED.
                  --ABNORMAL CODES.
                    J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN
                      30 ITERATIONS.  THE FIRST J-1 ELEMENTS OF EVR,
                      EVI CONTAIN THOSE EIGENVALUES ALREADY FOUND.
                   -1 MEANS THE INPUT VALUES OF N, NDIM ARE TOO LARGE
                      OR INCONSISTENT.

           AR,AI    ARE DESTROYED.
```

```
CNPCOF    CNPCOF    CNPCOF    CNPCOF         CNPCOF    CNPCOF    CNPCOF
                ************************************
                     ******************************
                          ********************
                               **********
      SUBROUTINE CNPCOF(L,W,IORDER,NORD,N,XX,A,C,WORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

      ***** ABSTRACT *****
      *
      *   SUBROUTINE CNPCOF COMPUTES THE COEFFICIENTS OF THE CONSTRAINED
      * LEAST SQUARES POLYNOMIAL ,P, PRODUCED BY SUBROUTINE CNPFIT. THE
      * COEFFICIENTS (STORED IN C) OF THE FIT ,P, ARE FOR THE TAYLOR
      * EXPANSION OF P ABOUT W. THAT IS, THE EXPANSION  FOR P HAS THE
      * FORM:
      *   P(X) = C(1) + C(2)*(X-W) + C(3)*((X-W)**2) + ... +
      *                                        C(L+1)*((X-W)**L)
      * OPTIONALLY, THE COEFFICIENTS MAY BE OBTAINED IN REVERSE ORDER.
      ******************************************************************

      *****  INPUT PARAMETERS  *****

      L       - THE ORDER OF THE POLYNOMIAL FOR WHICH THE COEFFICIENTS
                ARE DESIRED. L MUST BE GREATER THAN OR EQUAL TO N-1
                AND LESS THAN OR EQUAL TO NORD. THE CONSTRAINTS FORCE
                THE FITTING POLYNOMIAL TO BE OF DEGREE N-1 AT A MINIMUM.
                CHOOSING L = N-1  WILL PRODUCE THE COEFFICIENTS OF THE
                POLYNOMIAL DEFINED BY THE CONSTRAINTS --- EXCLUSIVE OF
                THE X,Y DATA.

      W       - THE POINT ABOUT WHICH THE TAYLOR EXPANSION IS TO BE MADE

      IORDER  - IF IORDER = 1, THE COEFFICIENTS OF THE TAYLOR EXPANSION
                ARE STORED IN C IN THE FORM:
                P(X) = C(1) + C(2)*(X-W) + C(3)*((X-W)**2) + ... +
                                                   C(L+1)*((X-W)**L)
                IF IORDER =-1, THE COEFFICIENTS OF THE TAYLOR EXPANSION
                ARE STORED IN C IN THE FORM:
                P(X) = C(1)*((X-W)**L) + C(2)*((X-W)**(L-1)) + ... +
                                                 C(L)*(X-W) + C(L+1)

      NORD    - *****
                *
      N       - *   NORD , N , XX , AND A  MUST REMAIN UNCHANGED
                *
      XX      - *   BETWEEN  THE CALL TO CNPFIT AND THE CALL TO CNPCOF.
                *
      A       - *****


      *****  OUTPUT PARAMETERS  *****

      C       - COEFFICIENTS OF THE POLYNOMIAL FIT, P(X), OF ORDER L.
                IF IORDER=1, THE COEFFICIENTS ARE THOSE IN THE FORM:
                P(X) = C(1) + C(2)*(X-W) + C(3)*((X-W)**2) + ... +
                                                   C(L+1)*((X-W)**L)
                IF IORDER=-1, THE COEFFICIENTS ARE IN THE REVERSE ORDER.


      *****  STORAGE PARAMETER  *****

      WORK    - THIS IS AN ARRAY TO PROVIDE INTERNAL WORKING STORAGE.
                IT MUST BE DIMENSIONED IN THE CALLING PROGRAM BY AT
                LEAST   L + 2*N + 4  .  (THE LARGEST POSSIBLE VALUE
                FOR L IS MAXDEG. HENCE  MAXDEG + 2*N + 4  WILL ALWAYS
                SUFFICE FOR THE DIMENSION OF WORK).
```

```
CNPFIT    CNPFIT    CNPFIT    CNPFIT         CNPFIT    CNPFIT    CNPFIT
            ****************************************
                 ******************************
                      ********************
                           **********
      SUBROUTINE CNPFIT(M,X,Y,WEIGHT,MAXDEG,N,XX,YY,IS,EPS,R,NORD,IER,A)
```

SANDIA MATHEMATICAL PROGRAM LIBRARY

CONSULTANTS AT SLL INCLUDE -
R. E. HUDDLESTON - DIVISION 8322
T. H. JEFFERSON - DIVISION 8322

WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

***** ABSTRACT *****

SUBROUTINE CNPFIT IS DESIGNED TO COMPUTE A CONSTRAINED LEAST-SQUARES POLYNOMIAL FIT. THAT IS, CNPFIT COMPUTES A POLYNOMIAL, P(Z), OF DEGREE NORD THAT SATISFIES BOTH OF THE FOLLOWING CONDITIONS.

(1) GIVEN THE DATA (X(I),Y(I)), I=1,...,M ,
P MINIMIZES THE MEAN-WEIGHTED-SQUARE ERROR

$$(1/M) * \sum_{I=1}^{M} WEIGHT(I)*(P(X(I))-Y(I))**2$$

WHERE WEIGHT IS A USER SUPPLIED ARRAY OF WEIGHTS.

(2) P MAY BE MADE TO PASS THROUGH GIVEN POINTS AND ITS DERIVATIVES MAY BE CONSTRAINED TO TAKE ON USER SUPPLIED VALUES. LET (K)P(Z) DENOTE THE DERIVATIVE OF ORDER K OF P EVALUATED AT Z. THE CONSTRAINTS ARE OF THE FORM

(IS(J))P(XX(J)) = YY(J) .

THE USER MUST SUPPLY THE CONSTRAINT ARRAYS XX(J), YY(J), AND IS(J), J = 1, ... ,N DEFINED BY

J	XX(J)	YY(J)	IS(J)
1	Z1	P(Z1)	0
2	Z1	(1)P(Z1)	1
3	Z1	(2)P(Z1)	2
.	.	.	.
.	.	.	.
.	.	.	.
N1	Z1	(N1-1)P(Z1)	N1-1
N1+1	Z2	P(Z2)	0
N1+2	Z2	(1)P(Z2)	1
N1+3	Z2	(2)P(Z2)	2
.	.	.	.
.	.	.	.
.	.	.	.
N1+N2	Z2	(N2-1)P(Z2)	N2-1
.	.	.	.
.	.	.	.
.	.	.	.
N			

THE CONSTRAINTS FORCE THE POLYNOMIAL FIT ,P(Z), TO BE OF DEGREE N-1 AT LEAST. THE USER MUST SUPPLY THE MAXIMUM DEGREE, MAXDEG, OF THE POLYNOMIAL FIT TO BE CONSIDERED. MAXDEG MUST BE GREATER THAN OR EQUAL TO N-1. *** NOTE *** IF THE DERIVATIVE OF ORDER K AT Z IS TO BE SPECIFIED, THEN THE VALUE OF P AND ALL OF THE DERIVATIVES OF P THROUGH ORDER K MUST BE SPECIFIED. ***

OPTIONALLY, CNPFIT MAY BE USED TO COMPUTE A FIT EVEN WHEN DATA

IS GIVEN FOR ONLY ONE OF THE TWO CONDITIONS LISTED ABOVE.
IF N=0 (NO CONSTRAINTS) POLFIT COULD BE USED INSTEAD OF CNPFIT.
IF M=0 (CONSTRAINTS ONLY) HRMITE COULD BE USED INSTEAD OF CNPFIT.

AFTER COMPUTING A FIT WITH CNPFIT, THE FITTING POLYNOMIAL AND ITS DERIVATIVES MAY BE EVALUATED AT ANY ABSCISSA USING CNPVAL AND THE COEFFICIENTS OF THE POLYNOMIAL MAY BE COMPUTED USING CNPCOF.

***** INPUT PARAMETERS *****

M - THE NUMBER OF DATA POINTS GIVEN IN X AND Y
X - THE ARRAY OF ABSCISSAS OF THE DATA POINTS
Y - THE ARRAY OF ORDINATES OF THE DATA POINTS
WEIGHT - THE ARRAY OF WEIGHTS CORRESPONDING TO THE DATA POINTS. IF W(1) IS NEGATIVE, CNPFIT WILL SET ALL OF THE WEIGHTS EQUAL TO 1.0 . IF THE USER WISHES RELATIVE ERROR, THEN THE WEIGHTS SHOULD BE WEIGHT(I)=1./Y(I)**2 ,I=1,...,M .
MAXDEG - THE MAXIMUM DEGREE OF POLYNOMIAL TO BE CONSIDERED FOR THE CONSTRAINED POLYNOMIAL FIT. SINCE THE N CONSTRAINTS FORCE THE POLYNOMIAL TO BE OF DEGREE N-1 AT LEAST, THEN MAXDEG MUST BE GREATER THAN OR EQUAL TO THE MAXIMUM OF N-1 AND ZERO.
- IF EPS IS GREATER THAN OR EQUAL TO ZERO, THEN MAXDEG MUST BE LESS THAN OR EQUAL TO M+N-1. IF MAXDEG = M+N-1 THEN THE FITTING POLYNOMIAL HAS JUST ENOUGH DEGREES OF FREEDOM TO SATISFY THE CONSTRAINTS AND TO INTERPOLATE ALL OF THE DATA (THIS CAN BE ACCOMPLISHED WITH EPS = 0 AND MAXDEG = M+N-1).
- IF EPS IS LESS THAN ZERO (THE STATISTICAL SELECTION CASE) THEN MAXDEG MUST BE LESS THAN M+N-1.
N - THE NUMBER OF CONSTRAINTS
XX - THE ARRAY OF ABSCISSAS OF THE CONSTRAINTS
YY - THE ARRAY OF VALUES OF THE CONSTRAINTS
IS - THE ARRAY WHICH SPECIFIES THE ORDER OF DERIVATIVES FOR EACH CONSTRAINT
*** NOTE *** IF THE DERIVATIVE OF ORDER K OF THE POLYNOMIAL P IS TO BE SPECIFIED AT AN ABSCISSA Z, THEN THE VALUE OF P AT Z AND ALL OF THE DERIVATIVES OF P AT Z THROUGH ORDER K MUST BE SPECIFIED.
EPS - ON INPUT, EPS SPECIFIES THE CRITERION TO BE USED IN DETERMINING THE ORDER, NORD, OF FIT TO BE COMPUTED:
(1) IF EPS IS INPUT NEGATIVE, CNPFIT CHOOSES THE ORDER, NORD, BASED ON A STATISTICAL F-TEST. IF EPS = -1 , THE ROUTINE WILL AUTOMATICALLY CHOOSE A SIGNIFICANCE LEVEL BASED ON THE NUMBER OF DATA POINTS AND THE MAXIMUM DEGREE OF POLYNOMIAL TO BE CONSIDERED. IF EPS IS INPUT AS -.01, -.05, OR -.10, SIGNIFICANCE LEVELS OF 1 PERCENT, 5 PERCENT, OR 10 PERCENT, RESPECTIVELY, WILL BE USED.

*** RECOMMENDATION *** UNLESS YOU KNOW ENOUGH ABOUT YOUR DATA TO MAKE A GOOD ESTIMATE OF THE RMS (ROOT MEAN SQUARE) ERROR
RMS = SQRT(SUMDIF/M), WHERE
SUMDIF = SUM(I=1,M)(WEIGHT(I)*(P(X(I))-Y(I))**2)
THEN THE BEST, AND EASIEST, WAY TO USE CNPFIT IS TO SET EPS = -1.0 .

(2) IF EPS IS SET TO 0.0 , CNPFIT SIMPLY COMPUTES THE POLYNOMIAL OF DEGREE MAXDEG.
(3) IF EPS IS INPUT POSITIVE, EPS IS THE RMS ERROR TOLERANCE WHICH MUST BE SATISFIED BY THE FITTED POLYNOMIAL. CNPFIT WILL INCREASE THE ORDER OF THE FIT UNTIL THIS OCCURS OR UNTIL THE MAXIMUM ORDER, MAXDEG, IS REACHED.

***** OUTPUT PARAMETERS *****

Y - IF M .GT. 0 AND MAXDEG .GT. N-1 THEN EACH Y(I) IS MODIFIED BY SUBTRACTING FROM IT THE VALUE AT X(I) OF THE POLYNOMIAL THAT SATISFIES JUST THE CONSTRAINTS. LATER THAT VALUE IS ADDED TO THE MODIFIED Y(I) TO RESTORE ITS VALUE . NOTE THAT THE RESTORED VALUE OF EACH Y(I) IS NOT NECESSARILY IDENTICAL TO ITS INPUT VALUE,

```
          BUT USUALLY WILL BE EQUAL OR NEARLY EQUAL TO THAT VALUE.
EPS     - IF  M .GT. 0  AND  MAXDEG .GT. N-1  THEN THE RMS ERROR
          OF THE POLYNOMIAL FIT OF DEGREE NORD IS RETURNED IN EPS.
R       - AN ARRAY CONTAINING THE VALUES OF THE POLYNOMIAL FIT OF
          ORDER NORD.  R(I), I=1,...,M  CONTAINS THE VALUE OF THE
          FIT AT X(I).
NORD    - THE HIGHEST ORDER OF POLYNOMIAL WHICH WAS CALCULATED.
IER     - OUTPUT ERROR FLAG WITH THE FOLLOWING POSSIBLE VALUES:
        = 1 INDICATES NORMAL EXECUTION, I.E. EITHER
            (1) THE INPUT VALUE OF EPS WAS NEGATIVE, AND THE
                COMPUTED POLYNOMIAL FIT OF ORDER NORD SATISFIES
                THE SPECIFIED F-TEST, OR
            (2) THE INPUT VALUE OF EPS WAS 0, AND THE FITS OF ALL
                ORDERS UP TO MAXDEG ARE COMPLETE, OR
            (3) THE INPUT VALUE OF EPS WAS POSITIVE, AND THE
                POLYNOMIAL OF ORDER NORD SATISFIES THE RMS ERROR
                REQUIREMENT.
        = 2  INDICATES THAT MAXDEG WAS LESS THAN N-1 (FATAL ERROR)
        = 3  INDICATES THAT EPS IS GREATER THAN OR EQUAL TO ZERO
             AND MAXDEG IS GREATER THAN M+N-1.  (FATAL ERROR)
        = 4  INDICATES THAT EPS IS LESS THAN ZERO AND MAXDEG IS
             GREATER THAN OR EQUAL TO M+N-1.  (FATAL ERROR)
        = 5  INDICATES THAT EPS IS LESS THAN -1 AND IS, THEREFORE,
             MEANINGLESS. (FATAL ERROR)
        = 6  INDICATES THAT THE RMS ERROR REQUIREMENT (SET BY
             CHOOSING EPS GREATER THAN ZERO) CANNOT BE SATISFIED
             WITH A POLYNOMIAL OF DEGREE NO GREATER THAN MAXDEG.
             THE FIT OF DEGREE MAXDEG IS RETURNED (NORD = MAXDEG).
        = 7  INDICATES THAT THE STATISTICAL TEST FOR SIGNIFICANCE
             (CHOSEN BY SETTING EPS LESS THAN ZERO) CANNOT BE
             SATISFIED USING THE CURRENT VALUE OF MAXDEG. IN THIS
             CASE NORD WILL HAVE ONE OF THE FOLLOWING VALUES :
             MAXDEG, MAXDEG-1, OR MAXDEG-2.  RERUNNING THE PROBLEM
             WITH A LARGER VALUE FOR MAXDEG MAY RESULT IN A
             BETTER FIT.
        ***  NOTE.  ERRCHK PROCESSES DIAGNOSTICS FOR CODES 2,3,4,5.
A       - WORK AND OUTPUT ARRAY WHICH MUST BE DIMENSIONED BY AT
          LEAST 2N + 2 + MAXIMUM (3M + 3MAXDEG - 3N + 4, 2N + 2).
          VALUES IN A ARE NEEDED IF CNPVAL OR CNPCOF ARE TO BE
          CALLED SUBSEQUENTLY.

***** DIMENSIONING INFORMATION *****

THE ARRAYS X, Y, WEIGHT, AND R MUST BE DIMENSIONED BY AT LEAST M
IN THE CALLING PROGRAM.
THE ARRAYS XX, YY, AND IS MUST BE DIMENSIONED BY AT LEAST N IN THE
CALLING PROGRAM.
THE ARRAY A MUST BE DIMENSIONED BY AT LEAST
      2N + 2 + MAXIMUM(3M + 3MAXDEG - 3N + 3 , 2N + 2)
IN THE CALLING PROGRAM.
```

CNPVAL CNPVAL CNPVAL CNPVAL CNPVAL CNPVAL CNPVAL

```
                ****************************************
                     ******************************
                          ********************
                               **********
SUBROUTINE CNPVAL(L,NDER,XFIT,YFIT,YP,NORD,N,XX,A,WORK,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

*****  ABSTRACT *****
```

SUBROUTINE CNPVAL EVALUATES THE CONSTRAINED LEAST - SQUARES POLYNOMIAL FIT PRODUCED BY CNPFIT.

***** INPUT PARAMETERS *****

L - THE ORDER OF POLYNOMIAL TO BE EVALUATED. L MUST BE GREATER THAN OR EQUAL TO N-1 AND LESS THAN OR EQUAL TO NORD. THE CONSTRAINTS FORCE THE FITTING POLYNOMIAL TO BE OF DEGREE N-1 AT A MINIMUM. THE LEAST SQUARES INFLUENCE ON THE FIT ACTUALLY TAKES PLACE BETWEEN ORDER N AND NORD. CHOOSING L = N-1 WILL RETURN THE VALUE OF THE POLYNOMIAL DEFINED BY THE CONSTRAINTS EXCLUSIVE OF THE X,Y DATA. NOTE THAT L MUST BE GREATER THAN OR EQUAL TO ZERO (EVEN IF N=0).

NDER - THE NUMBER OF DERIVATIVES TO BE EVALUATED.

XFIT - THE ABSCISSA AT WHICH THE FITTING POLYNOMIAL AND ITS DERIVATIVES ARE TO BE EVALUATED.

```
NORD -  *****
        *
N    -  *    NORD , N , XX , AND A MUST REMAIN UNCHANGED BETWEEN
        *    THE CALL TO CNPFIT AND THE CALL TO CNPVAL.
XX   -  *
        *
A    -  *****
```

***** OUTPUT PARAMETERS *****

YFIT - THE VALUE OF THE FITTING POLYNOMIAL AT XFIT

YP - THE ARRAY OF DERIVATIVES OF THE FITTING POLYNOMIAL EVALUATED AT XFIT. THE DERIVATIVE OF ORDER J IS STORED IN YP(J) , J=1,NDER.

IERR - OUTPUT ERROR FLAG WITH THE FOLLOWING POSSIBLE VALUES :
--NORMAL CODE
= 1 INDICATES NORMAL EXECUTION
--ABNORMAL CODES
= 2 INDICATES THAT L IS LESS THAN N-1
= 3 INDICATES THAT L IS GREATER THAN NORD

***** STORAGE PARAMETER *****

WORK - THIS IS AN ARRAY TO PROVIDE INTERNAL WORKING STORAGE. IT MUST BE DIMENSIONED BY AT LEAST 3*NDER + 2*N

COSH COSH COSH COSH COSH COSH COSH COSH COSH

```
*****************************************
*****************************
*******************
**********
```

FUNCTION COSH(X)

SANDIA MATHEMATICAL PROGRAM LIBRARY

CONSULTANTS AT SLL INCLUDE -
R. E. HUDDLESTON - DIVISION 8322
T. H. JEFFERSON - DIVISION 8322

WRITTEN BY CARL B. BAILEY, NOVEMBER 1971

ABSTRACT

COSH EVALUATES THE HYPERBOLIC COSINE FUNCTION. THAT IS,

```
        COSH(X) = (EXP(X) + EXP(-X)) / 2

    ACTUALLY, COSH(ABS(X)) IS COMPUTED TO REDUCE THE ROUND-OFF
    ERROR INCURRED WHEN X IS NEGATIVE.

DESCRIPTION OF ARGUMENT

    X - ANY REAL VALUE FOR WHICH EXP(ABS(X)) IS REPRESENTABLE.
```

```
CPQR    CPQR    CPQR    CPQR    CPQR       CPQR    CPQR    CPQR    CPQR
              ****************************************
                   ******************************
                        ********************
                             **********
SUBROUTINE CPQR(NDEG,CR,CI,WR,WI,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

THIS ROUTINE IS AN INTERFACE TO AN EIGENVALUE ROUTINE IN EISPACK.
THIS INTERFACE WAS WRITTEN BY WILLIAM R. GAVIN.

ABSTRACT

    THIS ROUTINE COMPUTES ALL ZEROS OF A POLYNOMIAL
    OF DEGREE TWENTY OR LESS WITH COMPLEX COEFFICIENTS
    BY COMPUTING THE EIGENVALUES OF THE COMPANION MATRIX.

DESCRIPTION OF PARAMETERS
    THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
         CR(NDEG+1), CI(NDEG+1), WR(NDEG), WI(NDEG)

 --INPUT--
   NDEG     DEGREE OF POLYNOMIAL

   CR,CI    REAL AND IMAGINARY PARTS OF COEFFICIENTS IN
            DESCENDING ORDER.  I.E.,
            P(Z) = (CR(1)+I*CI(1))*(Z**NDEG) + ...
              + (CR(NDEG)+I*CI(NDEG))*Z + (CR(NDEG+1)+I*CI(NDEG+1))

--OUTPUT--
   WR,WI    REAL AND IMAGINARY PARTS OF COMPUTED ROOTS

   IERR     OUTPUT ERROR CODE
         - NORMAL CODE
         0  MEANS THE ROOTS WERE COMPUTED.
         - ABNORMAL CODES
         1  MORE THAN 30 QR INTERATIONS ON SOME
            EIGENVALUE OF THE COMPANION MATRIX
         2  CR(1)=0.0 AND  CI(1)=0.0
         3  NDEG GREATER THAN 20 OR LESS THAN 1
```

```
ERF     ERF     ERF     ERF     ERF          ERF     ERF     ERF     ERF     ERF
                 ****************************************
                      ******************************
                           ********************
                                **********

     FUNCTION ERF(XX)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY J.E. VOGEL FROM APPROXIMATIONS DERIVED BY W.J. CODY .

     ABSTRACT

          ERF(X) COMPUTES 2.0/SQRT(PI) TIMES THE INTEGRAL FROM 0 TO X
          OF EXP(-X**2). THIS IS DONE USING RATIONAL APPROXIMATIONS.
          ELEVEN CORRECT SIGNIFICANT FIGURES ARE PROVIDED.

     DESCRIPTION OF PARAMETERS

          X MAY BE ANY REAL VALUE

     ERF IS DOCUMENTED COMPLETELY IN SC-M-70-275
```

```
ERFC    ERFC    ERFC    ERFC    ERFC         ERFC    ERFC    ERFC    ERFC
                 ****************************************
                      ******************************
                           ********************
                                **********

     FUNCTION ERFC(XX)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY J.E. VOGEL FROM APPROXIMATIONS DERIVED BY W.J. CODY .

     ABSTRACT

         ERFC(X) COMPUTES 2.0/SQRT(PI) TIMES THE INTEGRAL FROM X TO
         INFINITY OF EXP(-X**2). THIS IS DONE USING RATIONAL APPROX-
         IMATIONS.  ELEVEN CORRECT SIGNIFICANT FIGURES ARE PROVIDED.

     DESCRIPTION OF PARAMETERS

         X MAY BE ANY REAL VALUE

     ERFC IS DOCUMENTED COMPLETELY IN SC-M-70-275.
```

```
ERRCHK    ERRCHK    ERRCHK    ERRCHK         ERRCHK    ERRCHK    ERRCHK
                 ****************************************
                      ******************************
                           ********************
                                **********
     SUBROUTINE ERRCHK(NCHARS,NARRAY)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     ORIGINAL 6600 VERSION WRITTEN BY CARL B. BAILEY IN DECEMBER, 1968.
     LATEST REVISION OF COMMENTS BY R. E. JONES, NOVEMBER 1975

     ABSTRACT
         THE ROUTINES ERRCHK, ERXSET, AND ERRGET TOGETHER PROVIDE
         A UNIFORM METHOD WITH SEVERAL OPTIONS FOR THE PROCESSING
         OF DIAGNOSTICS AND WARNING MESSAGES WHICH ORIGINATE
         IN THE MATHEMATICAL PROGRAM LIBRARY ROUTINES.

     DESCRIPTION OF ARGUMENTS
         (BOTH ARGUMENTS ARE INPUT ONLY.)
         NCHARS - NUMBER OF CHARACTERS IN HOLLERITH MESSAGE.
                  IF NCHARS IS NEGATED, ERRCHK WILL UNCONDITIONALLY
                  PRINT THE MESSAGE AND STOP EXECUTION.  OTHERWISE,
                  THE BEHAVIOR OF ERRCHK MAY BE CONTROLLED BY
                  AN APPROPRIATE CALL TO ERXSET.
         NARRAY - NAME OF ARRAY OR VARIABLE CONTAINING THE MESSAGE,
                  OR ELSE A LITERAL HOLLERITH CONSTANT CONTAINING
                  THE MESSAGE.  BY CONVENTION, ALL MESSAGES SHOULD
                  BEGIN WITH *IN SUBNAM, ...*, WHERE SUBNAM IS THE
                  NAME OF THE ROUTINE CALLING ERRCHK.

     EXAMPLES
         1. TO ALLOW CONTROL BY CALLING ERXSET, USE
            CALL ERRCHK(30,30HIN QUAD, INVALID VALUE OF ERR.)
         2. TO UNCONDITIONALLY PRINT A MESSAGE AND STOP EXECUTION, USE
            CALL ERRCHK(-30,30HIN QUAD, INVALID VALUE OF ERR.)
```

```
ERRGET    ERRGET    ERRGET    ERRGET         ERRGET    ERRGET    ERRGET
                 ****************************************
                      ******************************
                           ********************
                                **********
     SUBROUTINE ERRGET(NFATAL,NTRACE)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     ORIGINAL 6600 VERSION WRITTEN BY CARL B. BAILEY IN DECEMBER, 1968.
     LATEST REVISION OF COMMENTS BY R. E. JONES, NOVEMBER 1975

     ABSTRACT
         THE ROUTINES ERRCHK, ERXSET, AND ERRGET TOGETHER PROVIDE
         A UNIFORM METHOD WITH SEVERAL OPTIONS FOR THE PROCESSING
         OF DIAGNOSTICS AND WARNING MESSAGES WHICH ORIGINATE
         IN THE MATHEMATICAL PROGRAM LIBRARY ROUTINES.

     DESCRIPTION OF ARGUMENTS
         BOTH ARGUMENTS ARE OUTPUT ARGUMENTS OF DATA TYPE INTEGER.
         NFATAL - CURRENT VALUE OF FATAL-ERROR / MESSAGE-LIMIT FLAG
                  (SEE DESCRIPTION OF ERXSET)
         NTRACE - CURRENT VALUE OF WALKBACK TRACE FLAG.
```

(SEE DESCRIPTION OF ERXSET)

```
ERXSET    ERXSET    ERXSET    ERXSET        ERXSET    ERXSET    ERXSET
                 ****************************************
                      ******************************
                           ********************
                                **********
     SUBROUTINE ERXSET(NFATAL,NTRACE)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     ORIGINAL 6600 VERSION WRITTEN BY CARL B. BAILEY IN DECEMBER, 1968.
     LATEST REVISION OF COMMENTS BY R. E. JONES, NOVEMBER 1975

     ABSTRACT
         THE ROUTINES ERRCHK, ERXSET, AND ERRGET TOGETHER PROVIDE
         A UNIFORM METHOD WITH SEVERAL OPTIONS FOR THE PROCESSING
         OF DIAGNOSTICS AND WARNING MESSAGES WHICH ORIGINATE
         IN THE MATHEMATICAL PROGRAM LIBRARY ROUTINES.

     DESCRIPTION OF ARGUMENTS
         BOTH ARGUMENTS ARE INPUT ARGUMENTS OF DATA TYPE INTEGER.
         NFATAL - IS A FATAL-ERROR / MESSAGE-LIMIT FLAG. A NEGATIVE
                  VALUE DENOTES THAT DETECTED DIFFICULTIES ARE TO BE
                  TREATED AS FATAL ERRORS.  NONNEGATIVE MEANS NONFATAL.
                  A NONNEGATIVE VALUE IS THE MAXIMUM NUMBER OF NONFATAL
                  WARNING MESSAGES WHICH WILL BE PRINTED BY ERRCHK,
                  AFTER WHICH NONFATAL MESSAGES WILL NOT BE PRINTED.
                  (DEFAULT VALUE IS -1.)
         NTRACE - .GE.1 WILL CAUSE A TRACE-BACK TO BE GIVEN,.
                  .LE.0 WILL SUPPRESS ANY TRACE-BACK, EXCEPT FOR
                        CASES WHEN EXECUTION IS TERMINATED.
                  (DEFAULT VALUE IS 0.)

         *NOTE* -- SOME CALLS TO ERRCHK WILL CAUSE UNCONDITIONAL
         TERMINATION OF EXECUTION.  ERXSET HAS NO EFFECT ON SUCH CALLS.

     EXAMPLES
         1. TO PRINT UP TO 100 MESSAGES AS NONFATAL WARNINGS USE
            CALL ERXSET(100,0)
         2. TO SUPPRESS ALL MATHLIB WARNING MESSAGES USE
            CALL ERXSET(0,0)
```

```
FOURT    FOURT    FOURT    FOURT      FOURT    FOURT    FOURT    FOURT
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY N M BRENNER, MIT LINCOLN LAB
      PREPARED FOR THE MATH LIBRARY BY R E JONES

      ABSTRACT

          FOURT PERFORMS AN N-DIMENSIONAL FAST FOURIER TRANSFORM ON AN
          N-DIMENSIONAL ARRAY OF COMPLEX DATA.  THE TRANSFORM PERFORMED
          MAY BE EXPRESSED AS FOLLOWS --

          TRANSFORM(J1,J2,...)=SUM(DATA(I1,I2,...)*W1**((I1-1)*(J1-1))
                                                  *W2**((I2-1)*(J2-1))
                                                  *... )
          WHERE I1 AND J1 RUN FROM 1 TO NN(1), AND
                I2 AND J2 RUN FROM 1 TO NN(2), ETC.
          AND
                W1 = EXP(ISIGN*2*PI*SQRT(-1)/NN(1)) , ETC.

          FOR ONE DIMENSION, THE TRANSFORM IS PRECISELY

          TRANSFORM(J1) = SUM(DATA(I1)*W1**((I1-1)*(J1-1)))

          FOURT IS FASTEST WHEN THE NUMBER OF DATA VALUES IN EACH
          DIMENSION IS A HIGHLY COMPOSITE (FACTORABLE) NUMBER.

          FOR FAST FOURIER TRANSFORMS OF DATA WHICH IS REAL (IN THE
          TIME DOMAIN) SEE SUBROUTINES FOURTR AND FOURTH.

      DESCRIPTION OF PARAMETERS
          DATA -- COMPLEX ARRAY IN WHICH THE DATA TO BE TRANSFORMED
                  IS PLACED.  UPON RETURN TO CALLING PROGRAM DATA
                  CONTAINS THE TRANSFORM VALUES.
          NN   - INTEGER ARRAY GIVING THE (POSITIVE) NUMBER OF POINTS,
                  OR VALUES, IN EACH DIMENSION, RESPECTIVELY.
          NDIM - NUMBER OF DIMENSIONS (INTEGER)  NDIM.GE.1
          ISIGN - INTEGER GIVING DIRECTION OF TRANSFORM TO BE DONE.
                  = -1 IMPLIES FORWARD
                  = +1 IMPLIES BACKWARD
          IFORM - INTEGER PARAMETER DESCRIBING THE FORM OF THE DATA.
                  = 1 IMPLIES THE DATA IS COMPLEX (NON-TRIVIALLY).
                  = 0 IMPLIES THE DATA IS ACTUALLY REAL.  I.E., THE
                      IMAGINARY PART OF EACH COMPLEX ELEMENT OF DATA
                      IS ZERO.  FOURT IS SIGNIFICANTLY FASTER WHEN
                      IFORM=0.
          WORK - COMPLEX WORK ARRAY.  WORK MUST BE DIMENSIONED AS
                  LARGE AS THE LARGEST DIMENSION OF DATA WHICH IS NOT
                  A POWER OF TWO.  IF ALL DIMENSIONS OF DATA ARE
                  POWERS OF TWO THEN WORK NEED NOT BE DIMENSIONED.
```

```
FOURTH    FOURTH    FOURTH    FOURTH        FOURTH    FOURTH    FOURTH
            ****************************************
                 ******************************
                      ********************
                           **********
      SUBROUTINE FOURTH(DATA,NN,ISIGN,WORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY RONDALL E JONES
      REFERENCE-- COOLEY, LEWIS, AND WELSH, *THE FAST FOURIER TRANSFORM
      AND ITS APPLICATIONS* (IBM RESEARCH PAPER RC-1743) SECTION 2.6

      ABSTRACT

          FOURTH PERFORMS ONE DIMENSIONAL INVERSE FAST FOURIER
          TRANSFORMS, GIVEN FOURIER COEFFICIENTS IN THE FORM RETURNED
          BY FOURTR (OR BY FCURT, IF THE FORWARD TRANSFORM WAS DONE
          USING NDIM=1, NN A MULTIPLE OF 4, AND IFORM=0).
          SPECIFICALLY, THE INVERSE TRANSFORM DONE AMOUNTS TO THE
          FOLLOWING, FOR K=1 TO NN.
          (DATA ON THE RIGHT SIDE OF THE EQUALITY REFERS TO INPUT
          VALUES.  DATA ON LEFT REFERS TO COMPUTED VALUES.)

            DATA(K) =  DATA(1)
                     + 2*SUM( DATA(2*I+1)*COS(2*PI*I*(K-1)/NN) )
                - ISIGN*2*SUM( DATA(2*I+2)*SIN(2*PI*I*(K-1)/NN) )
                     + DATA(NN+1)*COS(PI*(K-1))

          WHERE SUMS ARE FROM I=1 TO NN/2-1.
          THUS, FOR I=0 TO NN/2, THE INPUT DATA(2*I+1) AND DATA(2*I+2)
          MUST BE THE COSINE AND SINE COEFFICIENTS FOR THE FREQUENCY OF
          I*DF, WHERE DF IS THE FREQUENCY SPACING.
          NOTE THAT THE INPUT DATA(2) AND DATA(NN+2) ARE ASSUMED TO = 0.

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               DATA(NN+2), (WORK(NN) ONLY IF NN IS NOT A POWER OF TWO)

          INPUT---
          DATA  - ARRAY CONTAINING THE NN+2 FOURIER COEFFICIENTS,
                  IN THE FORM RETURNED BY FOURTR.
          NN    - THE NUMBER OF FOURIER COEFFICIENTS IS NN+2.
                  NN MUST BE A MULTIPLE OF 4, AND MUST BE AT LEAST 8.
          ISIGN - NORMALLY SHOULD BE + 1.  IN SOME SPECIAL CASES IT
                  MAY NEED TO BE -1 (SEE DEFINING EQUATIONS ABOVE).

          OUTPUT--
          DATA  - WILL CONTAIN THE NN REAL VALUES OF THE INVERSE
                  TRANSFORM.  (DATA(NN+1) AND DATA(NN+2) WILL BE ZERO.)
                  THE TIME SPACING OF THESE VALUES IS DT = 1/(NN*DF).

          WORK----
          WORK  - IF NN IS NOT A POWER OF TWO, WORK MUST BE AN ARRAY
                  OF AT LEAST NN WORDS.  IF NN IS A POWER OF TWO,
                  THEN WORK NEED NOT BE DIMENSIONED.
```

```
FOURTR    FOURTR    FOURTR    FOURTR        FOURTR    FOURTR    FOURTR
               ****************************************
                    ******************************
                         ********************
                              **********
      SUBROUTINE FOURTR(DATA,NN,ISIGN,WORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY RONDALL E JONES
      REFERENCE-- COOLEY, LEWIS, AND WELSH, *THE FAST FOURIER TRANSFORM
      AND ITS APPLICATIONS* (IBM RESEARCH PAPER RC-1743) SECTION 2.6

      ABSTRACT

          FOURTR PERFORMS A ONE-DIMENSIONAL FAST FOURIER TRANSFORM
          OF AN ARRAY OF NN REAL DATA VALUES, WHERE NN IS A MULTIPLE
          OF 4.  FOURTR RETURNS ONLY THE NON-REDUNDANT COEFFICIENTS
          (I.E., THE COEFFIENTS FOR FREQUENCIES 0 TO NN/2 CYCLES).
          THESE COEFFICIENTS ARE DEFINED AS FOLLOWS (FOR K=0 TO NN/2)
          (DATA ON THE RIGHT SIDE OF THE EQUALITY REFERS TO INPUT
          VALUES.  DATA ON LEFT REFERS TO COMPUTED VALUES.)

            DATA(2*K+1) =       SUM ( DATA(I)*COS(2*PI*(I-1)*K/NN) )
            DATA(2*K+2) =ISIGN*SUM ( DATA(I)*SIN(2*PI*(I-1)*K/NN) )

          WHERE SUMS ARE FROM I=1 TO NN.
          THUS, FOR I=1 TO NN, THE INPUT DATA(I) MUST BE THE TIME DOMAIN
          VALUE FOR THE TIME (I-1)*DT, WHERE DT IS THE TIME SPACING.
          NOTE THAT THE COMPUTED DATA(2) AND DATA(NN+2) WILL ALWAYS = 0.

          (SEE FOURTH FOR CORRESPONDING INVERSE TRANSFORMS.)

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               DATA(NN+2), (WORK(NN) ONLY IF NN IS NOT A POWER OF TWO)

          INPUT---
          DATA  - REAL ARRAY WHICH CONTAINS THE DATA TO BE TRANSFORMED.
                  DATA MUST BE DIMENSIONED AT LEAST NN+2, THE FIRST NN
                  WORDS CONTAINING THE VALUES TO BE TRANSFORMED.
          NN    - NUMBER OF VALUES IN DATA TO BE TRANSFORMED.
                  NN MUST BE A MULTIPLE OF 4, AND MUST BE AT LEAST 8.
          ISIGN - NORMALLY SHOULD BE -1.  IN SOME SPECIAL CASES IT
                  MAY NEED TO BE +1 (SEE DEFINING EQUATIONS ABOVE).

          OUTPUT--
          DATA  - WILL CONTAIN THE NN+2 REAL COSINE AND SINE
                  COFFICIENTS OF THE DISCRETE FOURIER TRANSFORM.
                  (DATA(2) AND DATA(NN+2) WILL BE ZERO.)
                  THE FREQUENCY SPACING OF THESE VALUES IS DF=1/(NN*DT).

          WORK----
          WORK  - IF NN IS NOT A POWER OF TWO, WORK MUST BE AN ARRAY
                  OF AT LEAST NN WORDS.  IF NN IS A POWER OF TWO,
                  THEN WORK NEED NOT BE DIMENSIONED.
```

```
GAMLN    GAMLN    GAMLN    GAMLN       GAMLN    GAMLN    GAMLN    GAMLN
              ****************************************
                   ******************************
                        ********************
                             **********

     FUNCTION GAMLN(X)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY D.E. AMOS AND S.L. DANIEL, FEBRUARY,1974.

     REFERENCE SAND-75-0152

     ABSTRACT
         GAMLN COMPUTES THE NATURAL LOG OF THE GAMMA FUNCTION FOR
         X.GT.0. A RATIONAL CHEBYSHEV APPROXIMATION IS USED ON
         8.LT.X.LT.1000., THE ASYMPTOTIC EXPANSION FOR X.GE.1000. AND
         BACKWARD RECURSION FOR 0.LT.X.LT.8 FOR NON-INTEGRAL X. FOR
         X=1.,...,8., GAMLN IS SET TO NATURAL LOGS OF FACTORIALS.

     DESCRIPTION OF ARGUMENTS

         INPUT
           X       - X.GT.0

         OUTPUT
           GAMLN   - NATURAL LOG OF THE GAMMA FUNCTION AT X

     ERROR CONDITIONS
         IMPROPER INPUT ARGUMENT - A FATAL ERROR
```

```
GAMMA    GAMMA    GAMMA    GAMMA       GAMMA    GAMMA    GAMMA    GAMMA
              ****************************************
                   ******************************
                        ********************
                             **********

     SUBROUTINE GAMMA(XX,GX,IERR)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     ROUTINE ORIGINALLY CONTAINED IN SCIENTIFIC SUBROUTINE PACKAGE.
     PREPARED AND MODIFIED FOR THE SANDIA MATHEMATICAL LIBRARY
     BY RONALD D. HALBGEWACHS, AUGUST 23,1968.

     ABSTRACT

        THIS ROUTINE COMPUTES THE GAMMA FUNCTION FOR REAL ARGUMENTS.
        RECURRENCE FORMULAS ARE USED TO REDUCE THE ARGUMENT TO THE
        RANGE 0 TO 1, WHERE A POLYNOMIAL APPROXIMATION IS USED.
        ACCURACY IS APPROXIMATELY 7 SIGNIFICANT FIGURES.
        FOR COMPLEX ARGUMENTS SEE SUBROUTINE GAMMAZ.

     DESCRIPTION OF PARAMETERS

             XX    = INPUT,REAL ARGUMENT OF THE GAMMA FUNCTION.
                     THE ARGUMENT MUST NOT BE WITHIN 1.0E-07 OF A
                     NEGATIVE INTEGER OR ZERO AND THE BOUNDS ON X ARE
                     X .GT. -163.0, X .LE. 177.0.

             GX    = OUTPUT,RESULTANT GAMMA FUNCTION VALUE.
```

```
IERR = OUTPUT,ERROR FLAG FOR THE CONDITIONS

       -- NORMAL CODE
          =1, NORMAL - NO ERRORS
       -- ABNORMAL CODES
          =2, XX IS WITHIN 1.0E-07 OF BEING A NEGATIVE
              INTEGER OR ZERO
          =3, ARGUMENT .LE. -163.0, OUT OF RANGE
          =4, ARGUMENT .GT. 177.0, OUT OF RANGE
```

```
GAMMAZ    GAMMAZ    GAMMAZ    GAMMAZ         GAMMAZ    GAMMAZ    GAMMAZ
               ****************************************
                    ******************************
                         ********************
                              **********
SUBROUTINE GAMMAZ(ZR,ZI,GR,GI,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

FROM A CO-OP LIBRARY ROUTINE ORIGINALLY WRITTEN BY W.H.K. LEE.
PREPARED AND MODIFIED (CHECKS FOR POLES, REGION OF APPLICABILITY,
AND ADDITION OF THE ASYMPTOTIC SECTION) FOR THE SANDIA
MATHEMATICAL LIBRARY BY RONALD D. HALBGEWACHS, OCTOBER 14,1968.

ABSTRACT

        THIS ROUTINE COMPUTES THE GAMMA FUNCTION FOR COMPLEX
        ARGUMENTS.  RECURRENCE AND REFLECTION FORMULAS ARE USED
        TO REDUCE THE ARGUMENT TO THE UNIT SQUARE, WHERE A
        PADE APPROXIMATION IS APPLIED.  FOR LARGE ARGUMENTS
        STIRLING-S ASYMPTOTIC EXPANSION IS USED.  APPROXIMATELY
        EIGHT CORRECT SIGNIFICANT FIGURES ARE PROVIDED.
        FOR REAL ARGUMENTS SEE SUBROUTINE GAMMA.

DESCRIPTION OF PARAMETERS

        ZR   = INPUT, REAL PART OF THE COMPLEX ARGUMENT.
               THE BOUNDS ON ZR DEPEND ON THE VALUE OF ZI.
               IF ZI IS ZERO, THE LOWER BOUND ON ZR IS -160.0
               AND THE UPPER BOUND IS 175.  AS ZI INCREASES IN
               MAGNITUDE, THE BOUNDS ON ZR INCREASE ALGEBRAICALLY.
               IF THE ARGUMENT IS STRICTLY REAL THEN ZR MUST
               NOT BE WITHIN 1.0E-07 OF A NEGATIVE INTEGER
               OR ZERO.

        ZI   = INPUT, IMAGINARY PART OF THE COMPLEX ARGUMENT.
               THE COMPLEX ARGUMENT (ZR,ZI) MUST NOT BE WITHIN
               1.0E-7 OF A NEGATIVE INTEGER OR ZERO.

        GR   = OUTPUT, REAL PART OF THE RESULTANT GAMMA FUNCTION
               VALUE.

        GI   = OUTPUT, IMAGINARY PART OF THE RESULTANT GAMMA
               FUNCTION VALUE.

        IERR = OUTPUT, ERROR FLAG FOR THE CONDITIONS

               -- NORMAL CODE
                  =1, NORMAL - NO ERRORS
               -- ABNORMAL CODES
                  =2, (ZR,ZI) IS WITHIN 1.0E-7 OF A NEGATIVE
                      INTEGER OR ZERO.
                  =3, ARGUMENT IS TO THE LEFT OF THE LEFT BOUNDS
```

```
                    =4, ARGUMENT IS TO THE RIGHT OF THE RIGHT BOUNDS
                    =5, ABS(ZI) .GT. 1800 AND RESULTS MAY BE
                        INACCURATE.
```

```
GAUS8    GAUS8    GAUS8    GAUS8       GAUS8    GAUS8    GAUS8    GAUS8
              ****************************************
                   ******************************
                        *******************
                            **********
      SUBROUTINE GAUS8  (FUN,A,B,ERR,ANS,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      BY RONDALL E JONES, SANDIA LABORATORIES
      SALIENT FEATURES -- INTERVAL BISECTION, COMBINED RELATIVE/ABSOLUTE
      ERROR CONTROL, COMPUTED MAXIMUM REFINEMENT LEVEL WHEN A IS
      CLOSE TO B.

      ABSTRACT
         GAUS8 INTEGRATES REAL FUNCTIONS OF ONE VARIABLE OVER FINITE
         INTERVALS, USING AN ADAPTIVE 8-POINT LEGENDRE-GAUSS ALGORITHM.
         GAUS8 IS INTENDED PRIMARILY FOR HIGH ACCURACY INTEGRATION
         OR INTEGRATION OF SMOOTH FUNCTIONS.  FOR LOWER ACCURACY
         INTEGRATION OF FUNCTIONS WHICH ARE NOT VERY SMOOTH,
         EITHER QNC3 OR QNC7 MAY BE MORE EFFICIENT.

      DESCRIPTION OF ARGUMENTS

         INPUT--
         FUN - NAME OF EXTERNAL FUNCTION TO BE INTEGRATED.  THIS NAME
               MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM.
               FUN MUST BE A FUNCTION OF ONE REAL ARGUMENT.  THE VALUE
               OF THE ARGUMENT TO FUN IS THE VARIABLE OF INTEGRATION
               WHICH RANGES FROM A TO B.
         A   - LOWER LIMIT OF INTEGRAL
         B   - UPPER LIMIT OF INTEGRAL (MAY BE LESS THAN A)
         ERR - IS A REQUESTED ERROR TOLERANCE.  NORMALLY PICK A VALUE OF
               ABS(ERR).LT.1.E-3.  ANS WILL NORMALLY HAVE NO MORE ERROR
               THAN ABS(ERR) TIMES THE INTEGRAL OF THE ABSOLUTE VALUE
               OF FUN(X).  USUALLY, SMALLER VALUES FOR ERR YIELD
               MORE ACCURACY AND REQUIRE MORE FUNCTION EVALUATIONS.
               A NEGATIVE VALUE FOR ERR CAUSES AN ESTIMATE OF THE
               ABSOLUTE ERROR IN ANS TO BE RETURNED IN ERR.

         OUTPUT--
         ERR - WILL BE AN ESTIMATE OF THE ERROR IN ANS IF THE INPUT
               VALUE OF ERR WAS NEGATIVE.  THE ESTIMATED ERROR IS SOLELY
               FOR INFORMATION TO THE USER AND SHOULD NOT BE USED AS
               A CORRECTION TO THE COMPUTED INTEGRAL.
         ANS - COMPUTED VALUE OF INTEGRAL
         IERR- A STATUS CODE
             --NORMAL CODES
                 1 ANS MOST LIKELY MEETS REQUESTED ERROR TOLERANCE,
                   OR A=B.
                -1 A AND B ARE TOO NEARLY EQUAL TO ALLOW NORMAL
                   INTEGRATION.  ANS IS SET TO ZERO.
             --ABNORMAL CODE
                 2 ANS PROBABLY DOES NOT MEET REQUESTED ERROR TOLERANCE.
```

```
GERK    GERK    GERK    GERK    GERK       GERK    GERK    GERK    GERK
              ****************************************
                   ******************************
                        *******************
                             **********
      SUBROUTINE GERK(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,GERROR,
     1                WORK,IWORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      FEHLBERG FOURTH(FIFTH) ORDER RUNGE-KUTTA METHOD WITH
      GLOBAL ERROR ASSESSMENT

      WRITTEN BY H.A.WATTS AND L.F.SHAMPINE
                    SANDIA LABORATORIES

     GERK IS DESIGNED TO SOLVE SYSTEMS OF DIFFERENTIAL EQUATIONS WHEN
     IT IS IMPORTANT TO HAVE A READILY AVAILABLE GLOBAL ERROR ESTIMATE.
     PARALLEL INTEGRATION IS PERFORMED TO YIELD TWO SOLUTIONS ON
     DIFFERENT MESH SPACINGS AND GLOBAL EXTRAPOLATION IS APPLIED TO
     PROVIDE AN ESTIMATE OF THE GLOBAL ERROR IN THE MORE ACCURATE
     SOLUTION.

**********************************************************************
 ABSTRACT
**********************************************************************

     SUBROUTINE  GERK  INTEGRATES A SYSTEM OF NEQN FIRST ORDER
     ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
              DY(I)/DT = F(T,Y(1),Y(2),...,Y(NEQN))
               WHERE THE Y(I) ARE GIVEN AT T .
     TYPICALLY THE SUBROUTINE IS USED TO INTEGRATE FROM T TO TOUT BUT IT
     CAN BE USED AS A ONE-STEP INTEGRATOR TO ADVANCE THE SOLUTION A
     SINGLE STEP IN THE DIRECTION OF TOUT. ON RETURN,AN ESTIMATE OF THE
     GLOBAL ERROR IN THE SOLUTION AT T IS PROVIDED AND THE PARAMETERS IN
     THE CALL LIST ARE SET FOR CONTINUING THE INTEGRATION. THE USER HAS
     ONLY TO CALL GERK AGAIN (AND PERHAPS DEFINE A NEW VALUE FOR TOUT).
     ACTUALLY, GERK  IS MERELY AN INTERFACING ROUTINE WHICH ALLOCATES
     VIRTUAL STORAGE IN THE ARRAYS WORK,IWORK AND CALLS SUBROUTINE GERKS
     FOR THE SOLUTION.  GERKS  IN TURN CALLS SUBROUTINE  FEHL  WHICH
     COMPUTES AN APPROXIMATE SOLUTION OVER ONE STEP.

     GERK  USES THE RUNGE-KUTTA-FEHLBERG (4,5) ' METHOD DESCRIBED
     IN THE REFERENCE
     E.FEHLBERG , LOW-ORDER CLASSICAL RUNGE-KUTTA FORMULAS WITH STEPSIZE
                  CONTROL , NASA TR R-315


     THE PARAMETERS REPRESENT-
       F -- SUBROUTINE F(T,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DT
       NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
       Y(*) -- SOLUTION VECTOR AT T
       T -- INDEPENDENT VARIABLE
       TOUT -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED
       RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
             ERROR TEST. AT EACH STEP THE CODE REQUIRES THAT
                  ABS(LOCAL ERROR) .LE. RELERR*ABS(Y) + ABSERR
             FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
       IFLAG -- INDICATOR FOR STATUS OF INTEGRATION
       GERROR(*) -- VECTOR WHICH ESTIMATES THE GLOBAL ERROR AT T. THAT
                    IS, GERROR(I) APPROXIMATES  Y(I)-TRUE SOLUTION(I).
       WORK(*) -- ARRAY TO HOLD INFORMATION INTERNAL TO GERK WHICH IS
             NECESSARY FOR SUBSEQUENT CALLS. MUST BE DIMENSIONED
             AT LEAST  3+8*NEQ
       IWORK(*) -- INTEGER ARRAY USED TO HOLD INFORMATION INTERNAL TO
             GERK WHICH IS NECESSARY FOR SUBSEQUENT CALLS. MUST BE
             DIMENSIONED AT LEAST  5

**********************************************************************
```

```
  FIRST CALL TO GERK
*********************************************************************

    THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
    IN THE CALL LIST  -       Y(NEQN) , WORK(3+8*NEQN) , IWORK(5)  ,
    DECLARE F IN AN EXTERNAL STATEMENT, SUPPLY SUBROUTINE F(T,Y,YP) AND
    INITIALIZE THE FOLLOWING PARAMETERS-

      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED.  (NEQN .GE. 1)
      Y(*) -- VECTOR OF INITIAL CONDITIONS
      T -- STARTING POINT OF INTEGRATION , MUST BE A VARIABLE
      TOUT -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED.
            T=TOUT IS ALLOWED ON THE FIRST CALL ONLY,IN WHICH CASE GERK
            RETURNS WITH IFLAG=2 IF CONTINUATION IS POSSIBLE.
      RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
            WHICH MUST BE NON-NEGATIVE BUT MAY BE CONSTANTS. WE CAN
            USUALLY EXPECT THE GLOBAL ERRORS TO BE SOMEWHAT SMALLER
            THAN THE REQUESTED LOCAL ERROR TOLERANCES. TO AVOID
            LIMITING PRECISION DIFFICULTIES THE CODE ALWAYS USES THE
            LARGER OF RELERR AND AN INTERNAL RELATIVE ERROR PARAMETER
            WHICH IS MACHINE DEPENDENT.
      IFLAG -- +1,-1  INDICATOR TO INITIALIZE THE CODE FOR EACH NEW
            PROBLEM. NORMAL INPUT IS +1. THE USER SHOULD SET IFLAG=-1
            ONLY WHEN ONE-STEP INTEGRATOR CONTROL IS ESSENTIAL. IN THIS
            CASE, GERK ATTEMPTS TO ADVANCE THE SOLUTION A SINGLE STEP
            IN THE DIRECTION OF TOUT EACH TIME IT IS CALLED. SINCE THIS
            MODE OF OPERATION RESULTS IN EXTRA COMPUTING OVERHEAD, IT
            SHOULD BE AVOIDED UNLESS NEEDED.


*********************************************************************
  OUTPUT FROM GERK
*********************************************************************

      Y(*) -- SOLUTION AT T
      T -- LAST POINT REACHED IN INTEGRATION.
      IFLAG = 2 -- INTEGRATION REACHED TOUT.INDICATES SUCCESSFUL RETURN
                   AND IS THE NORMAL MODE FOR CONTINUING INTEGRATION.
            =-2 -- A SINGLE SUCCESSFUL STEP IN THE DIRECTION OF TOUT
                   HAS BEEN TAKEN. NORMAL MODE FOR CONTINUING
                   INTEGRATION ONE STEP AT A TIME.
            = 3 -- INTEGRATION WAS NOT COMPLETED BECAUSE MORE THAN
                   9000 DERIVATIVE EVALUATIONS WERE NEEDED. THIS
                   IS APPROXIMATELY 500 STEPS.
            = 4 -- INTEGRATION WAS NOT COMPLETED BECAUSE SOLUTION
                   VANISHED MAKING A PURE RELATIVE ERROR TEST
                   IMPOSSIBLE. MUST USE NON-ZERO ABSERR TO CONTINUE.
                   USING THE ONE-STEP INTEGRATION MODE FOR ONE STEP
                   IS A GOOD WAY TO PROCEED.
            = 5 -- INTEGRATION WAS NOT COMPLETED BECAUSE REQUESTED
                   ACCURACY COULD NOT BE ACHIEVED USING SMALLEST
                   ALLOWABLE STEPSIZE. USER MUST INCREASE THE ERROR
                   TOLERANCE BEFORE CONTINUED INTEGRATION CAN BE
                   ATTEMPTED.
            = 6 -- GERK IS BEING USED INEFFICIENTLY IN SOLVING
                   THIS PROBLEM. TOO MUCH OUTPUT IS RESTRICTING THE
                   NATURAL STEPSIZE CHOICE. USE THE ONE-STEP
                   INTEGRATOR MODE.
            = 7 -- INVALID INPUT PARAMETERS (FATAL ERROR UNLESS
                   OVERRIDDEN BY CALL TO ERXSET)
                   THIS INDICATOR OCCURS IF ANY OF THE FOLLOWING IS
                   SATISFIED -   NEQN .LE. 0
                                 T=TOUT  AND  IFLAG .NE. +1 OR -1
                                 RELERR OR ABSERR .LT. 0.
                                 IFLAG .EQ. 0  OR  .LT. -2  OR  .GT. 7
      GERROR(*) -- ESTIMATE OF THE GLOBAL ERROR IN THE SOLUTION AT T
      WORK(*),IWORK(*) -- INFORMATION WHICH IS USUALLY OF NO INTEREST
                   TO THE USER BUT NECESSARY FOR SUBSEQUENT CALLS.
                   WORK(1),...,WORK(NEQN) CONTAIN THE FIRST DERIVATIVES
                   OF THE SOLUTION VECTOR Y AT T. WORK(NEQN+1) CONTAINS
                   THE STEPSIZE H TO BE ATTEMPTED ON THE NEXT STEP.
                   IWORK(1) CONTAINS THE DERIVATIVE EVALUATION COUNTER.
```

```
*********************************************************************
  SUBSEQUENT CALLS TO GERK
*********************************************************************

    SUBROUTINE GERK RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE THE
    INTEGRATION. IF THE INTEGRATION REACHED TOUT,THE USER NEED ONLY
    DEFINE A NEW TOUT AND CALL GERK AGAIN. IN THE ONE-STEP INTEGRATOR
    MODE (IFLAG=-2) THE USER MUST KEEP IN MIND THAT EACH STEP TAKEN IS
    IN THE DIRECTION OF THE CURRENT TOUT. UPON REACHING TOUT (INDICATED
    BY CHANGING IFLAG TO 2),THE USER MUST THEN DEFINE A NEW TOUT AND
    RESET IFLAG TO -2 TO CONTINUE IN THE ONE-STEP INTEGRATOR MODE.

    IF THE INTEGRATION WAS NOT COMPLETED BUT THE USER STILL WANTS TO
    CONTINUE (IFLAG=3 CASE), HE JUST CALLS GERK AGAIN. THE FUNCTION
    COUNTER IS THEN RESET TO 0 AND ANOTHER 9000 FUNCTION EVALUATIONS
    ARE ALLOWED.

    HOWEVER,IN THE CASE IFLAG=4, THE USER MUST FIRST ALTER THE ERROR
    CRITERION TO USE A POSITIVE VALUE OF ABSERR BEFORE INTEGRATION CAN
    PROCEED. IF HE DOES NOT,EXECUTION IS TERMINATED.

    ALSO,IN THE CASE IFLAG=5, IT IS NECESSARY FOR THE USER TO RESET
    IFLAG TO 2 (OR -2 WHEN THE ONE-STEP INTEGRATION MODE IS BEING USED)
    AS WELL AS INCREASING EITHER ABSERR,RELERR OR BOTH BEFORE THE
    INTEGRATION CAN BE CONTINUED. IF THIS IS NOT DONE, EXECUTION WILL
    BE TERMINATED. THE OCCURRENCE OF IFLAG=5 INDICATES A TROUBLE SPOT
    (SOLUTION IS CHANGING RAPIDLY,SINGULARITY MAY BE PRESENT) AND IT
    OFTEN IS INADVISABLE TO CONTINUE.

    IF IFLAG=6 IS ENCOUNTERED, THE USER SHOULD USE THE ONE-STEP
    INTEGRATION MODE WITH THE STEPSIZE DETERMINED BY THE CODE. IF THE
    USER INSISTS UPON CONTINUING THE INTEGRATION WITH GERK IN THE
    INTERVAL MODE, HE MUST RESET IFLAG TO 2 BEFORE CALLING GERK AGAIN.
    OTHERWISE,EXECUTION WILL BE TERMINATED.

    IF IFLAG=7 IS OBTAINED, INTEGRATION CAN NOT BE CONTINUED UNLESS
    THE INVALID INPUT PARAMETERS ARE CORRECTED.

    IT SHOULD BE NOTED THAT THE ARRAYS WORK,IWORK CONTAIN INFORMATION
    REQUIRED FOR SUBSEQUENT INTEGRATION. ACCORDINGLY, WORK AND IWORK
    SHOULD NOT BE ALTERED.

*********************************************************************
```

```
HRMITE    HRMITE    HRMITE    HRMITE         HRMITE    HRMITE    HRMITE
              ****************************************
                   ******************************
                        ********************
                             **********
     SUBROUTINE HRMITE (N,X,Y,IS,C,D)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

     ****  ABSTRACT   *****

        SUBROUTINE HRMITE IS DESIGNED TO PRODUCE A POLYNOMIAL FIT WHICH
     PASSES THROUGH GIVEN POINTS AND TAKES ON PRESCRIBED VALUES OF ITS
     DERIVATIVES. TO BE MORE SPECIFIC LET THE FOLLOWING N PIECES OF
     DATA BE GIVEN:

       K        X(K)              Y(K)            IS(K)
       ---------------------------------------------
```

```
1          Z1                  Y(1)              0
2          Z1                  Y(2)              1
3          Z1                  Y(3)              2
.          .                    .                .
.          .                    .                .
.          .                    .                .
N1         Z1                  Y(N1)             N1-1
N1+1       Z2                  Y(N1+1)           0
N1+2       Z2                  Y(N1+2)           1
N1+3       Z2                  Y(N1+3)           2
.          .                    .                .
.          .                    .                .
.          .                    .                .
N1+N2      Z2                  Y(N1+N2)          N2-1
N1+N2+1    Z3                  Y(N1+N2+1)        0
.          .                    .                .
.          .                    .                .
.          .                    .                .
N
```

LET (K)P(Z) DENOTE THE DERIVATIVE OF ORDER K OF THE FUNCTION P EVALUATED AT Z. THEN HRMITE DETERMINES A POLYNOMIAL P SUCH THAT

```
(IS(K))P(X(K)) = Y(K)   , K=1,2,...,N.
```

P IS DETERMINED IN THE FORM

```
P(Z) = C(1) + C(2)*(Z-X(1)) + C(3)*(Z-X(1))*(Z-X(2)) + ...
       + C(N)*(Z-X(1))*(Z-X(2))*...*(Z-X(N-1))
```

SUBROUTINE POLYVL CAN BE USED TO EVALUATE P AND ITS DERIVATIVES. SUBROUTINE POLCOF CAN BE USED TO DETERMINE THE COEFFICIENTS OF P IN A MORE STANDARD FORM.

EACH OF THE ARRAYS X, Y, IS, C, AND D MUST BE DIMENSIONED AT LEAST N. D IS A WORK ARRAY.

```
LPGAME    LPGAME    LPGAME    LPGAME         LPGAME    LPGAME    LPGAME
                 ****************************************
                      ******************************
                           ********************
                                **********
```

```
      SUBROUTINE LPGAME (LTYPE,M,N,A,IDIMA,B,NFSC,NEQS,SOLN,OBJ,ROW,COL,
     1       IERROR,WORK)
```

MODIFIED AND EXTENDED BY R. C. BASINGER FROM A SIMPLEX CODE DOCUMENTED AS AFWL-TR-65-195

ABSTRACT

THIS ROUTINE SOLVES EITHER FOR THE OPTIMAL STRATEGIES AND VALUE OF A TWO-PERSON ZERO-SUM GAME OR A LINEAR PROGRAMMING PROBLEM. IF A GAME IS SOLVED, THE GAME MATRIX (PAYOFF TO ROW PLAYER) IS ENTERED AS BELOW OR, IF A LINEAR PROGRAMMING PROGRAM IS SOLVED, THE LP MATRIX IS ENTERED AS BELOW.

GAME MATRIX

```
A(1,1)   A(1,2)   . . .   A(1,M)
A(2,1)   A(2,2)   . . .   A(2,M)
  .
  .
  .
A(N,1)   A(N,2)   . . .   A(N,M)
```

LP MATRIX (PRIMAL FORMULATION)

FIND X(1),X(2),...,X(M) THAT MAXIMIZE THE SUM

A(1,1)*X(1) + A(1,2)*X(2) + . . . + A(1,M)*X(M)

SUBJECT TO THE FOLLOWING CONSTRAINTS

```
A(2,1)*X(1) + . . . + A(2,M)*X(M)    .LE. OR .EQ.   B(2)
  .
  .
  .
A(N,1)*X(1) + . . .   A(N,M)*X(M)    .LE. OR .EQ.   B(N)
```

AND THE ADDITIONAL CONDITION THAT ALL EXCEPT THE FIRST NFSC VARIABLES X(1),...,X(NFSC) ARE NON-NEGATIVE. THE QUANTITY NFSC MAY BE ZERO, IN WHICH CASE ALL X(I),I=1,M ARE REQUIRED TO BE NON-NEGATIVE.
THE NUMBER OF EQUALITY CONSTRAINTS (AS OPPOSED TO INEQUALITY CONSTRAINTS) IS SPECIFIED BY THE INPUT PARAMETER NEQS, DESCRIBED BELOW.

INPUT PARAMETERS

LTYPE USE 0 FOR GAME, USE 1 FOR PRIMAL LP PROBLEM (ABOVE), OR USE 2 FOR THE ASSOCIATED DUAL LP PROBLEM. IT GENERALLY IS MORE EFFICIENT TO USE THE FORMULATION (PRIMAL OR DUAL) IN WHICH THERE ARE FEWER CONSTRAINTS THAN VARIABLES.

M NUMBER OF COLUMNS (GAME) OR NUMBER OF VARIABLES (LP).

N NUMBER OF ROWS (GAME) OR 1+NUMBER OF CONSTRAINTS (LP).
NOTE. IT IS REQUIRED THAT M+N LE 300.

A THE GAME (OR LP) MATRIX DESCRIBED ABOVE.

B THE VECTOR CORRESPONDING TO CONSTANTS IN THE LP CONSTRAINT EQUATIONS.
NOTICE B(1) IS NOT USED. B IS A WORK ARRAY FOR GAME PROBLEM.

IDIMA THE ACTUAL FIRST DIMENSION OF MATRIX A IN THE CALLING PROGRAM. A SHOULD BE DOUBLY DIMENSIONED WITH IDIMA.GE.(M+N+2) AND THE PRODUCT OF THE DIMENSIONS AT LEAST (M+N+2)*(M+N+2).

NFSC NUMBER OF VARIABLES IN LP WHICH ARE FREE OF SIGN CONSTRAINT IF NFSC.GT.0 , THE FIRST NFSC VARIABLES ARE UNCONSTRAINED. (NFSC=0 FOR THE GAME PROBLEM)

NEQS NUMBER OF EQUALITY CONSTRAINTS IN LP PROBLEM. ALL EQUALITY CONSTRAINTS PRECEDE INEQUALITY CONSTRAINTS. (NEQS=0 FOR THE GAME PROBLEM)

OUTPUT PARAMETERS

SOLN SOLUTION VECTOR FOR LP CODE.
SOLN(1),...,SOLN(M) ARE OPTIMUM VALUES FOR X(1),...,X(M) RESPECTIVELY AND THE REMAINING COMPONENTS ARE ASSOCIATED SLACK VARIABLE VALUES. IF LTYPE=2, THE SOLN VECTOR CONTAINS OPTIMAL VALUES ASSOCIATED WITH THE PRIMAL PROBLEM VIA THE DUAL. SOLN IS A WORK ARRAY FOR THE GAME PROBLEM.

OBJ THE MAXIMUM VALUE(PRIMAL) OR MINIMUM VALUE(DUAL) OF THE OBJECTIVE FUNCTION FOR LP PROBLEM. FOR THE GAME PROBLEM OBJ IS THE VALUE OF THE GAME.

ROW THE VECTOR OF OPTIMAL ROW PLAYER STRATEGIES.
(ROW(I) IS THE FREQUENCY OF PLAY FOR THE ITH ROW)
ROW IS A WORK ARRAY FOR LP PROBLEM.

COL THE VECTOR OF OPTIMAL COLUMN PLAYER STRATEGIES.
(COL(I) IS THE FREQUENCY OF PLAY FOR THE ITH COLUMN)
COL IS A WORK ARRAY FOR LP PROBLEM.

IERROR = 1 NORMAL TERMINATION OF PROBLEM.

= 2 ILLEGAL INPUT PARAMETER DETECTED.EXAMINE LTYPE, M,N,NFSC,NEQS, AND VERIFY M+N LE 300.

= 3 ELEMENT OF A MATRIX EITHER INDEFINITE OR OUT OF RANGE.

= 4 ELEMENT OF B VECTOR EITHER INDEFINITE OR OUT OF RANGE.

= 5 CONTRADICTORY CONSTRAINT EQUATIONS.

```
        = 6     NO FEASIBLE SOLUTION EXISTS.

        = 7     NO BOUNDED SOLUTION EXISTS.


.....................................................................
   THE CALLING PROGRAM MUST DIMENSION
       A(IDIMA,  *  ),B(M+N+2),SOLN(M+N-1),ROW(N),COL(M),
       WORK(M+N+2,M+N+2).
.....................................................................
```

```
MINA    MINA    MINA    MINA    MINA       MINA    MINA    MINA    MINA
              ****************************************
                   ******************************
                        ********************
                             **********
     SUBROUTINE MINA(FN,NV,NDIV,DEL,A,GUESS,X,FOFX,IERR)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     ORIGINAL ROUTINE WAS H2 SAND MIN, BY Z. BEISINGER AND S. BELL
     PRESENT VERSION BY R E JONES

     ABSTRACT
        MINA FINDS AN APPROXIMATE MINIMUM OF A REAL FUNCTION OF
        NV VARIABLES, GIVEN AN INITIAL ESTIMATE OF THE POSITION OF
        THE MINIMUM AND RANGES FOR EACH OF THE VARIABLES.
        MINA USES A SELECTIVE DIRECTED SEARCH OF A SURROUNDING
        NV-DIMENSIONAL GRID OF POINTS TO FIND A DIRECTION IN WHICH
        THE FUNCTION DECREASES.  IT THEN PROCEEDS IN THIS DIRECTION
        AS FAR AS THE FUNCTION DECREASES, THEN DETERMINES A NEW
        DIRECTION TO TRAVEL.  WHEN NO SUCH DIRECTION IS FOUND THE
        SEARCH INCREMENT FACTOR IS DECREASED AND THE PROCESS
        IS REPEATED.

     DESCRIPTION OF ARGUMENTS
        THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
              A(NV,2), GUESS(NV), X(NV)

        INPUT--
        FN    - NAME OF FUNCTION OF NV VARIABLES TO BE MINIMIZED.
                (THIS NAME MUST APPEAR IN AN EXTERNAL STATEMENT.)
                FORM OF THE CALLING SEQUENCE MUST BE FUNCTION FN(X),
                WHERE X IS AN ARRAY OF NV VARIABLE VALUES. THE
                ORDERING OF THE VARIABLES IS ARBITRARY, EXCEPT
                THAT IT MUST AGREE WITH THE ORDERING USED IN
                ARRAYS A AND GUESS.
        NV    - NUMBER OF VARIABLES.  (NV .GE. 1)
        NDIV  - NUMBER OF REFINEMENTS OF THE SEARCH INCREMENTS TO USE.
                AT EACH REFINEMENT, THE INCREMENT IN EACH DIMENSION
                IS DIVIDED BY 10.  (USUALLY NDIV IS ABOUT 3 OR 4.)
        DEL   - FRACTION OF VARIABLE RANGE (IN EACH DIMENSION) TO USE
                AS THE INITIAL INCREMENT (IN THAT DIMENSION)
        A     - ARRAY OF SEARCH BOUNDS, DIMENSIONED NV BY 2.
                A(I,1) SHOULD BE THE LOWER BOUND OF THE I-TH VARIABLE.
                A(I,2) SHOULD BE THE UPPER BOUND OF THE I-TH VARIABLE.
        GUESS- ARRAY OF NV INITIAL VALUES.  GUESS(I) SHOULD BE THE
                INITIAL VALUE TO USE FOR THE I-TH VARIABLE.

        OUTPUT--
        X     - ARRAY (DIMENSIONED NV) GIVING THE VALUES OF THE
                VARIABLES AT THE MINIMUM.  X(I) WILL BE THE VALUE
                OF THE I-TH VARIABLE.
```

```
      FOFX - FUNCTION VALUE AT THE MINIMUM
      IERR - A STATUS CODE
             -NORMAL CODE
              =1 MEANS THE SEARCH FOR A MINIMUM PROCEEDED FOR THE
                 SPECIFIED NUMBER OF REFINEMENTS.
             -ABNORMAL CODES
              =2 MEANS NV IS GREATER THAN 50
              =3 MEANS A RANGE MINIMUM IS GREATER THAN THE
                 CORRESPONDING MAXIMUM
```

```
ODE    ODE    ODE    ODE    ODE        ODE    ODE    ODE    ODE    ODE
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE ODE(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY L. F. SHAMPINE AND M. K. GORDON

     ABSTRACT


     SUBROUTINE  ODE  INTEGRATES A SYSTEM OF  NEQN  FIRST ORDER
     ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
              DY(I)/DT = F(T,Y(1),Y(2),...,Y(NEQN))
              Y(I) GIVEN AT  T .
     THE SUBROUTINE INTEGRATES FROM  T  TO  TOUT .  ON RETURN THE
     PARAMETERS IN THE CALL LIST ARE SET FOR CONTINUING THE INTEGRATION.
     THE USER HAS ONLY TO DEFINE A NEW VALUE  TOUT  AND CALL  ODE  AGAIN.

     THE DIFFERENTIAL EQUATIONS ARE ACTUALLY SOLVED BY A SUITE OF CODES
     DE ,  STEP , AND  INTRP .  ODE  ALLOCATES VIRTUAL STORAGE IN THE
     ARRAYS  WORK  AND  IWORK  AND CALLS  DE .  DE  IS A SUPERVISOR WHICH
     DIRECTS THE SOLUTION.  IT CALLS ON THE ROUTINES  STEP  AND  INTRP
     TO ADVANCE THE INTEGRATION AND TO INTERPOLATE AT OUTPUT POINTS.
     STEP  USES A MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE
     FORMULAS AND LOCAL EXTRAPOLATION.  IT ADJUSTS THE ORDER AND STEP
     SIZE TO CONTROL THE LOCAL ERROR PER UNIT STEP IN A GENERALIZED
     SENSE.  NORMALLY EACH CALL TO  STEP  ADVANCES THE SOLUTION ONE STEP
     IN THE DIRECTION OF  TOUT .  FOR REASONS OF EFFICIENCY  DE
     INTEGRATES BEYOND  TOUT  INTERNALLY, THOUGH NEVER BEYOND
     T+10*(TOUT-T), AND CALLS  INTRP  TO INTERPOLATE THE SOLUTION AT
     TOUT .  AN OPTION IS PROVIDED TO STOP THE INTEGRATION AT  TOUT  BUT
     IT SHOULD BE USED ONLY IF IT IS IMPOSSIBLE TO CONTINUE THE
     INTEGRATION BEYOND  TOUT .

     THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
     COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
     VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.

     THE PARAMETERS REPRESENT:
        F -- SUBROUTINE F(T,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DT
        NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
        Y(*) -- SOLUTION VECTOR AT  T
        T -- INDEPENDENT VARIABLE
        TOUT -- POINT AT WHICH SOLUTION IS DESIRED
        RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
             ERROR TEST.  AT EACH STEP THE CODE REQUIRES
               ABS(LOCAL ERROR) .LE. ABS(Y)*RELERR + ABSERR
             FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
        IFLAG -- INDICATES STATUS OF INTEGRATION
        WORK(*),IWORK(*) -- ARRAYS TO HOLD INFORMATION INTERNAL TO CODE
```

```
        WHICH IS NECESSARY FOR SUBSEQUENT CALLS

FIRST CALL TO ODE --

THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
IN THE CALL LIST,
   Y(NEQN), WORK(100+21*NEQN), IWORK(5),
DECLARE  F  IN AN EXTERNAL STATEMENT, SUPPLY THE SUBROUTINE
F(T,Y,YP) TO EVALUATE
   DY(I)/DT = YP(I) = F(T,Y(1),Y(2),...,Y(NEQN))
AND INITIALIZE THE PARAMETERS:
   NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
   Y(*) -- VECTOR OF INITIAL CONDITIONS
   T -- STARTING POINT OF INTEGRATION
   TOUT -- POINT AT WHICH SOLUTION IS DESIRED
   RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
   IFLAG -- +1,-1.  INDICATOR TO INITIALIZE THE CODE.  NORMAL INPUT
        IS +1.  THE USER SHOULD SET IFLAG=-1 ONLY IF IT IS
        IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .
ALL PARAMETERS EXCEPT  F ,  NEQN  AND  TOUT  MAY BE ALTERED BY THE
CODE ON OUTPUT SO MUST BE VARIABLES IN THE CALLING PROGRAM.

OUTPUT FROM  ODE  --

   NEQN -- UNCHANGED
   Y(*) -- SOLUTION AT  T
   T -- LAST POINT REACHED IN INTEGRATION.  NORMAL RETURN HAS
        T = TOUT .
   TOUT -- UNCHANGED
   RELERR,ABSERR -- NORMAL RETURN HAS TOLERANCES UNCHANGED.  IFLAG=3
        SIGNALS TOLERANCES INCREASED
   IFLAG = 2 -- NORMAL RETURN.  INTEGRATION REACHED  TOUT
         = 3 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE ERROR
                TOLERANCES TOO SMALL.  RELERR ,  ABSERR  INCREASED
                APPROPRIATELY FOR CONTINUING
         = 4 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE MORE THAN
                500 STEPS NEEDED
         = 5 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE EQUATIONS
                APPEAR TO BE STIFF
         = 6 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE SOLUTION
                VANISHED MAKING PURE RELATIVE ERROR IMPOSSIBLE.
                MUST USE NON-ZERO  ABSERR  TO CONTINUE.
         = 7 -- INVALID INPUT PARAMETERS (FATAL ERROR)
        THE VALUE OF  IFLAG  IS RETURNED NEGATIVE WHEN THE INPUT
        VALUE IS NEGATIVE AND THE INTEGRATION DOES NOT REACH  TOUT ,
        I.E., -3, -4, -5, -6.
   WORK(*),IWORK(*) -- INFORMATION GENERALLY OF NO INTEREST TO THE
        USER BUT NECESSARY FOR SUBSEQUENT CALLS.

SUBSEQUENT CALLS TO  ODE --

SUBROUTINE  ODE  RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE
THE INTEGRATION.  IF THE INTEGRATION REACHED  TOUT , THE USER NEED
ONLY DEFINE A NEW  TOUT  AND CALL AGAIN.  IF THE INTEGRATION DID NOT
REACH  TOUT  AND THE USER WANTS TO CONTINUE, HE JUST CALLS AGAIN.
IN THE CASE  IFLAG=6 , THE USER MUST ALSO ALTER THE ERROR CRITERION.
THE OUTPUT VALUE OF  IFLAG  IS THE APPROPRIATE INPUT VALUE FOR
SUBSEQUENT CALLS.  THE ONLY SITUATION IN WHICH IT SHOULD BE ALTERED
IS TO STOP THE INTEGRATION INTERNALLY AT THE NEW  TOUT , I.E.,
CHANGE OUTPUT  IFLAG=2  TO INPUT  IFLAG=-2 .  ERROR TOLERANCES MAY
BE CHANGED BY THE USER BEFORE CONTINUING.  ALL OTHER PARAMETERS MUST
REMAIN UNCHANGED.
```

```
ODERT    ODERT    ODERT    ODERT       ODERT    ODERT    ODERT    ODERT
              ***************************************
                   ****************************
                        *********************
                             **********
      SUBROUTINE ODERT(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK,IWORK,
     1  G,REROOT,AEROOT)

   WRITTEN BY M. K. GORDON, 5122

*********************************************************************
   ABSTRACT
*********************************************************************
   SUBROUTINE  ODERT  INTEGRATES A SYSTEM OF  NEQN  FIRST ORDER
   ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
            DY(I)/DT = F(T,Y(1),...,Y(NEQN))
            Y(I) GIVEN AT T.
   THE SUBROUTINE INTEGRATES FROM  T  IN THE DIRECTION OF  TOUT  UNTIL
   IT LOCATES THE FIRST ROOT OF THE NONLINEAR EQUATION
         G(T,Y(1),...,Y(NEQN),YP(1),...,YP(NEQN)) = 0.
   UPON FINDING THE ROOT, THE CODE RETURNS WITH ALL PARAMETERS IN THE
   CALL LIST SET FOR CONTINUING THE INTEGRATION TO THE NEXT ROOT OR
   THE FIRST ROOT OF A NEW FUNCTION  G  .  IF NO ROOT IS FOUND, THE
   INTEGRATION PROCEEDS TO  TOUT  .  AGAIN ALL PARAMETERS ARE SET TO
   CONTINUE.

   THE DIFFERENTIAL EQUATIONS ARE ACTUALLY SOLVED BY A SUITE OF CODES,
   DERT ,  STEP , AND  INTRP .  ODERT  ALLOCATES VIRTUAL STORAGE IN
   THE WORK ARRAYS  WORK  AND  IWORK  AND CALLS  DERT .  DERT  IS A
   SUPERVISOR WHICH DIRECTS THE INTEGRATION.  IT CALLS ON  STEP  TO
   ADVANCE THE SOLUTION AND  INTRP  TO INTERPOLATE THE SOLUTION AND
   ITS DERIVATIVE.  STEP  USES A MODIFIED DIVIDED DIFFERENCE FORM OF
   THE ADAMS PECE FORMULAS AND LOCAL EXTRAPOLATION.  IT ADJUSTS THE
   ORDER AND STEP SIZE TO CONTROL THE LOCAL ERROR PER UNIT STEP IN A
   GENERALIZED SENSE.  NORMALLY EACH CALL TO  STEP  ADVANCES THE
   SOLUTION ONE STEP IN THE DIRECTION OF  TOUT .  FOR REASONS OF
   EFFICIENCY  ODERT  INTEGRATES BEYOND  TOUT  INTERNALLY, THOUGH
   NEVER BEYOND T+10*(TOUT-T), AND CALLS  INTRP  TO INTERPOLATE THE
   SOLUTION AND DERIVATIVE AT  TOUT .  AN OPTION IS PROVIDED TO STOP
   THE INTEGRATION AT  TOUT  BUT IT SHOULD BE USED ONLY IF IT IS
   IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .

   AFTER EACH INTERNAL STEP,  DERT  EVALUATES THE FUNCTION  G  AND
   CHECKS FOR A CHANGE IN SIGN IN THE FUNCTION VALUE FROM THE
   PRECEDING STEP.  SUCH A CHANGE INDICATES A ROOT LIES IN THE
   INTERVAL OF THE STEP JUST COMPLETED.  DERT  THEN CALLS SUBROUTINE
   ROOT  TO REDUCE THE BRACKETING INTERVAL UNTIL THE ROOT IS
   DETERMINED TO THE DESIRED ACCURACY.  SUBROUTINE  ROOT  USES A
   COMBINATION OF THE SECANT RULE AND BISECTION TO DO THIS.  THE
   SOLUTION AND DERIVATIVE VALUES REQUIRED ARE OBTAINED BY
   INTERPOLATION WITH  INTRP .  THE CODE LOCATES ONLY THOSE ROOTS
   FOR WHICH  G  CHANGES SIGN IN  (T,TOUT)  AND FOR WHICH A
   BRACKETING INTERVAL EXISTS.  IN PARTICULAR, IT WILL NOT DETECT A
   ROOT AT THE INITIAL POINT  T .

   THE CODES  STEP  , INTRP  , ROOT  , AND THAT PORTION OF  DERT
   WHICH DIRECTS THE INTEGRATION ARE EXPLAINED AND DOCUMENTED IN THE
   TEXT, COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS, THE
   INITIAL VALUE PROBLEM, BY L. F. SHAMPINE AND M. K. GORDON.

   DETAILS OF THE USE OF ODERT ARE GIVEN IN SAND-75-0211.

*********************************************************************
   THE PARAMETERS FOR  ODERT  ARE
*********************************************************************
      F -- SUBROUTINE F(T,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DT
      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
      Y(*) -- SOLUTION VECTOR AT  T
      T -- INDEPENDENT VARIABLE
      TOUT -- ARBITRARY POINT BEYOND THE ROOT DESIRED
      RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
           ERROR TEST.  AT EACH STEP THE CODE REQUIRES
             ABS(LOCAL ERROR) .LE. ABS(Y)*RELERR + ABSERR
           FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
```

```
      IFLAG -- INDICATES STATUS OF INTEGRATION
      WORK,IWORK -- ARRAYS TO HOLD INFORMATION INTERNAL TO THE CODE
           WHICH IS NECESSARY FOR SUBSEQUENT CALLS
      G - FUNCTION OF T, Y(*), YP(*) WHOSE ROOT IS DESIRED.
      REROOT, AEROOT -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR
           ACCEPTING THE ROOT.  THE INTERVAL CONTAINING THE ROOT IS
           REDUCED UNTIL IT SATISFIES
            0.5*ABS(LENGTH OF INTERVAL) .LE. REROOT*ABS(ROOT)+AEROOT
           WHERE ROOT IS THAT ENDPOINT YIELDING THE SMALLER VALUE OF
           G  IN MAGNITUDE.  PURE RELATIVE ERROR IS NOT RECOMMENDED
           IF THE ROOT MIGHT BE ZERO.
*********************************************************************
   FIRST CALL TO  ODERT --
*********************************************************************
   THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE
   ARRAYS IN THE CALL LIST,
             Y(NEQN), WORK(100+21*NEQN), IWORK(5)
   AND DECLARE  F , G  IN AN EXTERNAL STATEMENT.  HE MUST SUPPLY THE
   SUBROUTINE  F(T,Y,YP)  TO EVALUATE
           DY(I)/DT = YP(I) = F(T,Y(1),...,Y(NEQN))
   AND THE FUNCTION  G(T,Y,YP)  TO EVALUATE
           G = G(T,Y(1),...,Y(NEQN),YP(1),...,YP(NEQN)).
   NOTE THAT THE ARRAY YP IS AN INPUT ARGUMENT AND SHOULD NOT BE
   COMPUTED IN THE FUNCTION SUBPROGRAM.  FINALLY THE USER MUST
   INITIALIZE THE PARAMETERS
      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
      Y(*) -- VECTOR OF INITIAL CONDITIONS
      T -- STARTING POINT OF INTEGRATION
      TOUT -- ARBITRARY POINT BEYOND THE ROOT DESIRED
      RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
                       FOR INTEGRATING THE EQUATIONS
      IFLAG -- +1,-1.  INDICATOR TO INITIALIZE THE CODE.  NORMAL INPUT
           IS +1.  THE USER SHOULD SET IFLAG=-1 ONLY IF IT IS
           IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .
      REROOT,AEROOT -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR
                       COMPUTING THE ROOT OF  G

   ALL PARAMETERS EXCEPT F, G, NEQN, TOUT, REROOT AND AEROOT MAY BE
   ALTERED BY THE CODE ON OUTPUT SO MUST BE VARIABLES IN THE CALLING
   PROGRAM.
*********************************************************************
   OUTPUT FROM  ODERT --
*********************************************************************
      NEQN -- UNCHANGED
      Y(*) -- SOLUTION AT  T
      T -- LAST POINT REACHED IN INTEGRATION.  NORMAL RETURN HAS
           T = TOUT OR T = ROOT
      TOUT -- UNCHANGED
      RELERR,ABSERR -- NORMAL RETURN HAS TOLERANCES UNCHANGED.  IFLAG=3
           SIGNALS TOLERANCES INCREASED
      IFLAG = 2 -- NORMAL RETURN.  INTEGRATION REACHED  TOUT
            = 3 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE ERROR
                   TOLERANCES TOO SMALL.  RELERR ,  ABSERR  INCREASED
                   APPROPRIATELY FOR CONTINUING
            = 4 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE MORE THAN
                   500 STEPS NEEDED
            = 5 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE EQUATIONS
                   APPEAR TO BE STIFF
            = 6 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE SOLUTION
                   VANISHED MAKING PURE RELATIVE ERROR IMPOSSIBLE.
                   MUST USE NON-ZERO  ABSERR  TO CONTINUE
            = 7 -- INVALID INPUT PARAMETERS (FATAL ERROR)
            = 8 -- NORMAL RETURN.  A ROOT WAS FOUND WHICH SATISFIED
                   THE ERROR CRITERION OR HAD A ZERO RESIDUAL
            = 9 -- ABNORMAL RETURN.  AN ODD ORDER POLE OF  G  WAS
                   FOUND.
           =10 -- ABNORMAL RETURN.  TOO MANY EVALUATIONS OF  G  WERE
                   REQUIRED (AS PROGRAMMED 500 ARE ALLOWED.)
           THE VALUE OF  IFLAG  IS RETURNED NEGATIVE WHEN THE INPUT
           VALUE IS NEGATIVE AND THE INTEGRATION DOES NOT REACH
           TOUT , I.E., -3,...,-6,-8,-9,-10.
      WORK(*),IWORK(*) -- INFORMATION GENERALLY OF NO INTEREST TO THE
                          USER BUT NECESSARY FOR SUBSEQUENT CALLS
      REROOT,AEROOT -- UNCHANGED
```

```
*********************************************************************
   SUBSEQUENT CALLS TO  ODERT --
*********************************************************************
   SUBROUTINE  ODERT  RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE
   THE INTEGRATION.  IF THE INTEGRATION DID NOT REACH  TOUT  AND THE
   USER WANTS TO CONTINUE, HE JUST CALLS AGAIN.  IF THE INTEGRATION
   REACHED  TOUT , THE USER NEED ONLY DEFINE A NEW  TOUT  AND CALL
   AGAIN.  THE OUTPUT VALUE OF  IFLAG  IS THE APPROPRIATE INPUT VALUE
   FOR SUBSEQUENT CALLS.  THE ONLY SITUATION IN WHICH IT SHOULD BE
   ALTERED IS TO STOP THE INTEGRATION INTERNALLY AT THE NEW  TOUT ,
   I.E., CHANGE OUTPUT  IFLAG=2  TO INPUT  IFLAG=-2 .  ONLY THE ERROR
   TOLERANCES AND THE FUNCTION  G  MAY BE CHANGED BY THE USER BEFORE
   CONTINUING.  ALL OTHER PARAMETERS MUST REMAIN UNCHANGED.  A NEW
   FUNCTION  G  IS DETECTED AUTOMATICALLY BY COMPARING ITS VALUE AT A
   SPECIFIED INTERNAL POINT WITH A STORED VALUE FOR THE PRECEDING
   FUNCTION EVALUATED AT THE SAME POINT.
```

```
PCOEF    PCOEF    PCOEF    PCOEF       PCOEF    PCOEF    PCOEF    PCOEF
              ****************************************
                   ******************************
                        ********************
                             **********
     SUBROUTINE PCOEF (L,C,TC,A)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY L. F. SHAMPINE AND S. M. DAVENPORT.

     ABSTRACT

     POLFIT  COMPUTES THE LEAST SQUARES POLYNOMIAL FIT OF ORDER  L  AS
     A SUM OF ORTHOGONAL POLYNOMIALS.  PCOEF  CHANGES THIS FIT TO ITS
     TAYLOR EXPANSION ABOUT ANY POINT  C , I.E. WRITES THE POLYNOMIAL
     AS A SUM OF POWERS OF (X-C).  TAKING  C=0.  GIVES THE POLYNOMIAL
     IN POWERS OF X, BUT A SUITABLE NON-ZERO  C  OFTEN LEADS TO
     POLYNOMIALS WHICH ARE BETTER SCALED AND MORE ACCURATELY EVALUATED.

     THE PARAMETERS FOR  PCOEF  ARE

     INPUT:
         L -       INDICATES THE ORDER OF POLYNOMIAL TO BE CHANGED TO
                   ITS TAYLOR EXPANSION.  TO OBTAIN THE TAYLOR
                   COEFFICIENTS IN REVERSE ORDER, INPUT  L  AS THE
                   NEGATIVE OF THE ORDER DESIRED.  THE ABSOLUTE VALUE OF
                   L  MUST BE LESS THAN OR EQUAL TO  NORD , THE HIGHEST
                   ORDER POLYNOMIAL FITTED BY  POLFIT .
         C -       THE POINT ABOUT WHICH THE TAYLOR EXPANSION IS TO BE
                   MADE.
         A -       WORK AND OUTPUT ARRAY CONTAINING VALUES FROM LAST
                   CALL TO  POLFIT .

     OUTPUT:
         TC -      VECTOR CONTAINING THE FIRST LL+1 TAYLOR COEFFICIENTS
                   WHERE LL=IABS(L).  IF  L.GT.0 , THE COEFFICIENTS ARE
                   IN THE USUAL TAYLOR SERIES ORDER, I.E.
                     P(X) = TC(1) + TC(2)*(X-C) + ... + TC(N+1)*(X-C)**N
                   IF L .LT. 0, THE COEFFICIENTS ARE IN REVERSE ORDER,
                   I.E.
                     P(X) = TC(1)*(X-C)**N + ... + TC(N)*(X-C) + TC(N+1)
```

```
POLCOF    POLCOF    POLCOF    POLCOF         POLCOF    POLCOF    POLCOF
                 ****************************************
                      ******************************
                           ********************
                                **********
      SUBROUTINE POLCOF (XX,N,X,C,D,WORK)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

      ABSTRACT
         SUBROUTINE POLCOF COMPUTES THE COEFFICIENTS OF THE POLYNOMIAL
      FIT (INCLUDING HERMITE POLYNOMIAL FITS ) PRODUCED BY A PREVIOUS
      CALL TO HRMITE OR POLINT. THE COEFFICIENTS OF THE POLYNOMIAL,
      EXPANDED ABOUT XX, ARE STORED IN THE ARRAY D. THE EXPANSION IS OF
      THE FORM
      P(Z) = D(1) + D(2)*(Z-XX) +D(3)*((Z-XX)**2) + ... +
                                                   D(N)*((Z-XX)**(N-1)).
      BETWEEN THE CALL TO POLINT (OR TO HRMITE) AND THE CALL TO POLCOF
      THE VARIABLE N AND THE ARRAYS X AND C MUST NOT BE ALTERED.

      *****  INPUT PARAMETERS

      XX   - THE POINT ABOUT WHICH THE TAYLOR EXPANSION IS TO BE MADE.

      N    - ****
             *     N, X, AND C MUST REMAIN UNCHANGED BETWEEN THE
      X    - *     CALL TO POLINT OR THE CALL TO HRMITE AND THE
             *     CALL TO POLCOF.
      C    - ****

      *****  OUTPUT PARAMETER

      D    - THE ARRAY OF COEFFICIENTS FOR THE TAYLOR EXPANSION AS
             EXPLAINED IN THE ABSTRACT


      *****  STORAGE PARAMETER

      WORK - THIS IS AN ARRAY TO PROVIDE INTERNAL WORKING STORAGE. IT
             MUST BE DIMENSIONED BY AT LEAST 2*N IN THE CALLING PROGRAM.


      **** NOTE - THERE ARE TWO METHODS FOR EVALUATING THE FIT PRODUCED
      BY POLINT OR HRMITE. YOU MAY CALL POLYVL TO PERFORM THE TASK, OR
      YOU MAY CALL POLCOF TO OBTAIN THE COEFFICIENTS OF THE TAYLOR
      EXPANSION AND THEN WRITE YOUR OWN EVALUATION SCHEME. DUE TO THE
      INHERENT ERRORS IN THE COMPUTATIONS OF THE TAYLOR EXPANSION FROM
      THE NEWTON COEFFICIENTS PRODUCED BY POLINT OR HRMITE, MUCH MORE
      ACCURACY MAY BE EXPECTED BY CALLING POLYVL AS OPPOSED TO WRITING
      YOUR OWN SCHEME.
```

```
POLFIT    POLFIT    PCLFIT    POLFIT         POLFIT    POLFIT    POLFIT
                 ****************************************
                      ******************************
                           ********************
                                **********
      SUBROUTINE POLFIT (N,X,Y,W,MAXORD,NORD,EPS,R,IERR,A)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY L. F. SHAMPINE AND S. M. DAVENPORT.  THE STATISTICAL
      OPTIONS PROVIDED WERE WRITTEN BY R. E. HUDDLESTON.

      ABSTRACT

      GIVEN A COLLECTICN OF POINTS X(I) AND A SET OF VALUES Y(I) WHICH
      CORRESPOND TO SOME FUNCTION OR MEASUREMENT AT EACH OF THE X(I),
      SUBROUTINE  PCLFIT  COMPUTES THE WEIGHTED LEAST-SQUARES POLYNOMIAL
      FITS OF ALL ORDERS UP TO SOME ORDER EITHER SPECIFIED BY THE USER
      OR DETERMINED BY THE ROUTINE.  THE FITS THUS OBTAINED ARE IN
      ORTHOGONAL POLYNOMIAL FORM.  SUBROUTINE  PVALUE  MAY THEN BE
      CALLED TO EVALUATE THE FITTED POLYNOMIALS AND ANY OF THEIR
      DERIVATIVES AT ANY POINT.  THE SUBROUTINE  PCOEF  MAY BE USED TO
      EXPRESS THE POLYNOMIAL FITS AS POWERS OF (X-C) FOR ANY SPECIFIED
      POINT C.

      THE PARAMETERS FCR  POLFIT  ARE

      INPUT:
          N -       THE NUMBER OF DATA POINTS.  THE ARRAYS X, Y, W, R
                    MUST BE DIMENSIONED AT LEAST  N  (N .GE. 1).
          X -       ARRAY OF VALUES OF THE INDEPENDENT VARIABLE.  THESE
                    VALUES MAY APPEAR IN ANY ORDER AND NEED NOT ALL BE
                    DISTINCT.
          Y -       ARRAY OF CORRESPONDING FUNCTION VALUES.
          W -       ARRAY OF POSITIVE VALUES TO BE USED AS WEIGHTS.  IF
                    W(1) IS NEGATIVE,  POLFIT  WILL SET ALL THE WEIGHTS
                    TO 1.0, WHICH MEANS ABSOLUTE ERROR WILL BE MINIMIZED.
                    TO MINIMIZE RELATIVE ERROR, THE USER SHOULD SET
                    WEIGHTS TO:  W(I) = 1.0/Y(I)**2, I = 1,...,N .
          MAXORD -  MAXIMUM ORDER TO BE ALLOWED FOR POLYNOMIAL FIT.
                    MAXORD  MAY BE ANY NON-NEGATIVE INTEGER LESS THAN  N.
                    NOTE -- MAXORD  CANNOT BE EQUAL TO  N-1  WHEN A
                    STATISTICAL TEST IS TO BE USED FOR ORDER SELECTION,
                    I.E., WHEN INPUT VALUE OF  EPS  IS NEGATIVE.
          EPS -     SPECIFIES THE CRITERION TO BE USED IN DETERMINING
                    THE ORDER OF FIT TO BE COMPUTED:
                    (1)  IF  EPS  IS INPUT NEGATIVE,  PCLFIT  CHOOSES THE
                         ORDER BASED ON A STATISTICAL F TEST OF
                         SIGNIFICANCE.  ONE OF THREE POSSIBLE
                         SIGNIFICANCE LEVELS WILL BE USED:  .01, .05 OR
                         .10.  IF  EPS=-1.0 , THE ROUTINE WILL
                         AUTOMATICALLY SELECT ONE OF THESE LEVELS BASED
                         ON THE NUMBER OF DATA POINTS AND THE MAXIMUM
                         ORDER TC BE CONSIDERED.  IF  EPS  IS INPUT AS
                         -.01, -.05, OR -.10, A SIGNIFICANCE LEVEL OF
                         .01, .05, OR .10, RESPECTIVELY, WILL BE USED.
                    (2)  IF  EPS  IS SET TO 0.,  POLFIT  COMPUTES THE
                         POLYNOMIALS OF ORDERS 0 THROUGH  MAXORD .
                    (3)  IF  EPS  IS INPUT POSITIVE,  EPS  IS THE RMS
                         ERROR TOLERANCE WHICH MUST BE SATISFIED BY THE
                         FITTED POLYNOMIAL.  POLFIT  WILL INCREASE THE
                         ORDER OF FIT UNTIL THIS CRITERION IS MET OR
                         UNTIL THE MAXIMUM ORDER IS REACHED.

      OUTPUT:
          NORD -    ORDER OF THE HIGHEST ORDER FIT COMPUTED.
          EPS -     RMS ERROR OF THE POLYNOMIAL OF ORDER  NORD .
          R -       VECTOR CONTAINING VALUES OF THE FIT OF ORDER  NORD
                    AT EACH OF THE  X(I) .  EXCEPT WHEN THE STATISTICAL
                    TEST IS USED, THESE VALUES ARE MORE ACCURATE THAN
```

```
              RESULTS FROM SUBROUTINE  PVALUE  NORMALLY ARE.
      IERR -  ERROR FLAG WITH THE FOLLOWING POSSIBLE VALUES:
          1 -- INDICATES NORMAL EXECUTION, I.E. EITHER
               (1)  THE INPUT VALUE OF  EPS  WAS NEGATIVE, AND THE
                    COMPUTED POLYNOMIAL FIT OF ORDER  NORD
                    SATISFIES THE SPECIFIED F TEST, OR
               (2)  THE INPUT VALUE OF  EPS  WAS 0., AND THE FITS OF
                    ALL ORDERS UP TO  MAXORD  ARE COMPLETE, OR
               (3)  THE INPUT VALUE OF  EPS  WAS POSITIVE, AND THE
                    POLYNOMIAL OF ORDER  NORD  SATISFIES THE RMS
                    ERROR REQUIREMENT.
          2 -- INVALID INPUT PARAMETER.  AT LEAST ONE OF THE INPUT
               PARAMETERS HAS AN ILLEGAL VALUE AND MUST BE CORRECTED
               BEFORE  POLFIT  CAN PROCEED.  VALID INPUT RESULTS
               WHEN THE FOLLOWING RESTRICTIONS ARE OBSERVED:
                    N .GE. 1
                    0 .LE. MAXORD .LE. N-1  FOR  EPS .GE. 0.
                    0 .LE. MAXORD .LE. N-2  FOR  EPS .LT. 0.
                    W(1)=-1.0  OR  W(I) .GT. 0., I=1,...,N .
          3 -- CANNOT SATISFY THE RMS ERROR REQUIREMENT WITH A
               POLYNOMIAL OF ORDER NO GREATER THAN  MAXORD .  BEST
               FIT FOUND IS OF ORDER  MAXORD .
          4 -- CANNOT SATISFY THE TEST FOR SIGNIFICANCE USING
               CURRENT VALUE OF  MAXORD .  STATISTICALLY, THE
               BEST FIT FOUND IS OF ORDER  NORD .  (IN THIS CASE,
               NORD WILL HAVE ONE OF THE VALUES:  MAXORD-2,
               MAXORD-1, OR MAXORD).  USING A HIGHER VALUE OF
               MAXORD  MAY RESULT IN PASSING THE TEST.
      A -      WORK AND OUTPUT ARRAY HAVING AT LEAST 3N+3MAXORD+3
               LOCATIONS

NOTE:  POLFIT  CALCULATES ALL FITS OF ORDERS UP TO AND INCLUDING
       NORD .  ANY OR ALL OF THESE FITS CAN BE EVALUATED OR
       EXPRESSED AS POWERS OF (X-C) USING  PVALUE  AND  PCOEF
       AFTER JUST ONE CALL TO  POLFIT .
```

POLINT POLINT POLINT POLINT POLINT POLINT POLINT

```
*****************************************
*****************************
*******************
**********

SUBROUTINE POLINT (N,X,Y,C)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

ABSTRACT
   SUBROUTINE POLINT IS DESIGNED TO PRODUCE THE POLYNOMIAL WHICH
INTERPOLATES THE DATA  (X(I),Y(I)), I=1,...,N.  POLINT SETS UP
INFORMATION IN THE ARRAY C WHICH CAN BE USED BY SUBROUTINE POLYVL
TO EVALUATE THE POLYNOMIAL AND ITS DERIVATIVES AND BY SUBROUTINE
POLCOF TO PRODUCE THE COEFFICIENTS.

FORMAL PARAMETERS
N  - THE NUMBER OF DATA POINTS  (N .GE. 1)
X  - THE ARRAY OF ABSCISSAS (ALL OF WHICH MUST BE DISTINCT)
Y  - THE ARRAY OF ORDINATES
C  - AN ARRAY OF INFORMATION USED BY SUBROUTINES
*******  DIMENSIONING INFORMATION  *******
ARRAYS X,Y, AND C MUST BE DIMENSIONED AT LEAST N IN THE CALLING
PROGRAM.
```

```
POLYVL    POLYVL    POLYVL    POLYVL         POLYVL    POLYVL    POLYVL
               ****************************************
                    ******************************
                         ********************
                              **********
      SUBROUTINE POLYVL (NDER,XX,YFIT,YP,N,X,C,WORK,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

      ABSTRACT -
         SUBROUTINE POLYVL CALCULATES THE VALUE OF THE POLYNOMIAL AND
      ITS FIRST NDER DERIVATIVES WHERE THE POLYNOMIAL WAS PRODUCED BY
      A PREVIOUS CALL TO HRMITE OR POLINT.
         THE VARIABLE N AND THE ARRAYS X AND C MUST NOT BE ALTERED
      BETWEEN THE CALL TO HRMITE OR POLINT AND THE CALL TO POLYVL.

      ******  DIMENSIONING INFORMATION *******

      YP   MUST BE DIMENSIONED BY AT LEAST NDER
      X    MUST BE DIMENSIONED BY AT LEAST N (SEE THE ABSTRACT )
      C    MUST BE DIMENSIONED BY AT LEAST N (SEE THE ABSTRACT )
      WORK MUST BE DIMENSIONED BY AT LEAST 2*N IF NDER IS .GT. 0.

      *** NOTE ***
        IF NDER = 0 NEITHER YP NOR WORK NEED TO BE DIMENSIONED VARIABLES
        IF NDER = 1 YP DOES NOT NEED TO BE A DIMENSIONED VARIABLE


      *****  INPUT PARAMETERS

      NDER - THE NUMBER OF DERIVATIVES TO BE EVALUATED

      XX   - THE ARGUMENT AT WHICH THE POLYNOMIAL AND ITS DERIVATIVES
             ARE TO BE EVALUATED.

      N    - *****
             *       N, X, AND C MUST NOT BE ALTERED BETWEEN THE CALL
      X    - *       TO HERMITE (OR THE CALL TO POLINT) AND THE CALL
             *       TO POLYVL.
      C    - *****


      *****  OUTPUT PARAMETERS

      YFIT - THE VALUE OF THE POLYNOMIAL AT XX

      YP   - THE DERIVATIVES OF THE POLYNOMIAL AT XX. THE DERIVATIVE OF
             ORDER J AT XX IS STORED IN  YP(J) , J = 1,...,NDER.

      IERR - OUTPUT ERROR FLAG WITH THE FOLLOWING POSSIBLE VALUES:
           = 1  INDICATES NORMAL EXECUTION

      ***** STORAGE PARAMETERS

      WORK  = THIS IS AN ARRAY TO PROVIDE INTERNAL WORKING STORAGE FOR
              POLYVL. IT MUST BE DIMENSIONED BY AT LEAST 2*N IF NDER IS
              .GT. 0.  IF NDER=0, WORK DOES NOT NEED TO BE A DIMENSIONED
              VARIABLE.
```

```
PSMTH1    PSMTH1    PSMTH1    PSMTH1          PSMTH1    PSMTH1    PSMTH1
                ***************************************
                      ******************************
                            ********************
                                 **********
      SUBROUTINE PSMTH1(X,Y,NTOTAL,NDERIV,WEIGHT,LPARAM,WORK,YP)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY ROBERT E. HUDDLESTON, SANDIA LABORATORIES, LIVERMORE

      *****  ABSTRACT  *****

          GIVEN DATA  (X(I),Y(I)), I=1,...,NTOTAL  WHICH IS GENERALLY
      ASSUMED TO BE NOISY (BOT NOT SO NECESSARILY), SUBROUTINE PSMTH1 IS
      DESIGNED TO CALCULATE SMOOTHED Y(I) VALUES (AND DERIVATIVES AT
      THE X(I) IF NDERIV IS GREATER THAN ZERO ).  THE SIMPLEST WAY TO
      USE THIS ROUTINE IS TO CHOOSE THE DEFAULT VALUES FOR LPARAM (SEE
      THE EXPLANATION OF LPARAM BELOW).  IF YOU WISH TO CHANGE THE
      DEFAULT VALUES, THEN YOU SHOULD READ THE REST OF THIS ABSTRACT
      AND THE EXPLANATION OF LPARAM. IN ADDITION THERE IS A REFERENCE
      DOCUMENT ,  SAND74-8200 , AVAILABLE FROM ROBERT E. HUDDLESTON
      DIVISION 8322 , EXT. 2120.
              THE SUBROUTINE FITS ONLY IPTS POINTS (DEFAULT VALUE IS 9)
              AT A TIME. THESE POINTS ARE FIT WITH LEAST SQUARE
              POLYNOMIALS OF DEGREE 0,1,2,...,NDEG (DEFAULT VALUE = 3)
              AND A STATISTICAL DECISION SUBROUTINE IS THEN CALLED TO
              SELECT THE MOST REPRESENTATIVE DEGREE LESS THAN OR EQUAL
              TO NDEG BASED ON AN F DISTRIBUTION TEST. USING THIS DEGREE
              OF POLYNOMIAL A ROUTINE IS THEN CALLED WHICH EVALUATES
              NDERIV DERIVATIVES OF THE POLYNOMIAL. THESE DERIVATIVES
              ARE EVALUTED AT NEVAL POINTS (DEFAULT VALUE = 3) CENTERED
              ABOUT THE MIDDLE OF THE IPTS USED IN THE POLYNOMIAL FITS.
              (NOTE THAT NEVAL AND IPTS ARE BOTH ODD.) THE FITTING
              POINTS ARE THEN SHIFTED BY NEVAL POINTS AND THE PROCESS
              CARRIED OUT REPEATEDLY UNTIL NTOTAL POINTS HAVE BEEN
              PROCESSED. THE NON-SYMMETRIES AT THE LEFT END AND RIGHT
              END OF THE DATA STRING ARE TAKEN CARE OF AUTOMATICALLY
              UNLESS IFLAG IS CHANGED FROM ITS DEFAULT VALUE OF 4.


      ****  INPUT PARAMETERS  ******

      X - SINGLY DIMENSIONED INPUT ARRAY OF ABSCISSAS.  THE X(I) ARE
          ASSUMED TO BE MONOTONICALLY NON-DECREASING. IF THEY ARE NOT,
          YOU SHOULD MAKE A CALL TO A SORTING ROUTINE SUCH AS SSORT
          (AVAILABLE ON THE SANDIA MATHEMATICAL PROGRAM LIBRARY). X MUST
          BE DIMENSIONED BY AT LEAST NTOTAL.
      Y - SINGLY DIMENSIONED INPUT ARRAY OF ORDINATES CORRESPONDING TO
          THE X ARRAY. Y MUST BE DIMENSIONED BY AT LEAST NTOTAL.
      NTOTAL - THE NUMBER OF INPUT POINTS IN EACH OF X AND Y.
      NDERIV - THE NUMBER OF DERIVATIVES ONE WISHES TO COMPUTE.
      WEIGHT - WEIGHTING FOR ABSOLUTE OR RELATIVE ERROR IN THE
               POLYNOMIAL FITS:
               SET WEIGHT = -1.0 FOR ABSOLUTE ERROR
               SET WEIGHT =  1.0 FOR RELATIVE ERROR

      LPARAM - AN INTEGER ARRAY CONTAINING CERTAIN PARAMETERS WHICH ARE
               DESCRIBED IN THE ABSTRACT BELOW. LPARAM MUST BE
               DIMENSIONED BY AT LEAST 4 IN THE CALLING PROGRAM.
               *****  IF THE USER SETS LPARAM(I) = 0 , I =1,...,4  IN
               *****  THE CALLING PROGRAM, SUBROUTINE PSMTH1 WILL USE
               *****  DEFAULT VALUES FOR THE PARAMETERS.
               THE ELEMENTS OF THE ARRAY LPARAM CORRESPOND TO :
                  LPARAM(1) = IPTS  (IPTS MUST BE ODD. THE DEFAULT VALUE
                                     OF IPTS IS 9 )
                  LPARAM(2) = NEVAL (NEVAL MUST BE ODD. THE DEFAULT
                                     VALUE OF NEVAL IS 3 )
                  LPARAM(3) = NDEG  (NDEG MUST BE LESS THAN OR EQUAL
                                     TO  IPTS-1 . IT MUST BE LESS THAN
```

```
                                IPTS-1  IF ANY SMOOTHING IS TO BE
                                ACHIEVED. THE DEFAULT VALUE FOR
                                NDEG IS 3 )
            LPARAM(4) = IFLAG (THE DEFAULT VALUE OF IFLAG IS 4 .
                                THIS SHOULD NOT BE ALTERED UNLESS
                                YOU HAVE READ SAND74-8200  )

****  OUTPUT PARAMETERS  ******

Y  - CONTAINS THE SMOOTHED VALUES OF THE ORDINATES (I.E. Y(I)
     CONTAINS THE VALUE OF THE LEAST SQUARE POLYNOMIAL FIT
     EVALUATED AT X(I)).

YP - SINGLY DIMENSIONED OUTPUT ARRAY OF DERIVATIVES. THE J TH
     DERIVATIVE EVALUATED AT THE ABSCISSA  X(I) IS LOCATED IN
     YP(I + (J-1)*NTOTAL). YP MUST BE DIMENSIONED BY AT LEAST
     NTOTAL*DERIV.  AN ALTERNATIVE IS FOR YP TO BE A TWO
     DIMENSIONAL ARRAY DIMENSIONED YP(NTOTAL,NDERIV).  THEN
     YP(I,J) IS THE J TH DERIVATIVE AT X(I).

****  STORAGE PARAMETER  ****

WORK  - SINGLY DIMENSIONED WORK ARRAY TO PROVIDE THE NECESSARY
        INTERNAL STORAGE FOR THE SUBROUTINE.  IT MUST BE
        DIMENSIONED BY AT LEAST  5*IPTS + 3*NDEG + 3.
        IF ONE USES THE DEFAULT VALUES FOR IPTS AND NDEG, THEN
        WORK MUST BE DIMENSIONED BY AT LEAST 57.

****  NOTE.  INVALID INPUT IS DIAGNOSED AND THE DIAGNOSTICS ARE
             PROCESSED BY ERRCHK.
```

PVALUE PVALUE PVALUE PVALUE PVALUE PVALUE PVALUE

```
                 ****************************************
                      ******************************
                           ********************
                                *********

SUBROUTINE PVALUE (L,NDER,X,YFIT,YP,A)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

WRITTEN BY L. F. SHAMPINE AND S. M. DAVENPORT.

ABSTRACT

THE SUBROUTINE  PVALUE  USES THE COEFFICIENTS GENERATED BY  POLFIT
TO EVALUATE THE POLYNOMIAL FIT OF ORDER  L , ALONG WITH THE FIRST
NDER  OF ITS DERIVATIVES, AT A SPECIFIED POINT.  COMPUTATIONALLY
STABLE RECURRENCE RELATIONS ARE USED TO PERFORM THIS TASK.

THE PARAMETERS FOR  PVALUE  ARE

INPUT:
    L -       THE ORDER OF POLYNOMIAL TO BE EVALUATED.  L  MAY BE
              ANY NON-NEGATIVE INTEGER WHICH IS LESS THAN OR EQUAL
              TO  NORD , THE HIGHEST ORDER POLYNOMIAL PROVIDED
              BY  POLFIT .
    NDER -    THE NUMBER OF DERIVATIVES TO BE EVALUATED.  NDER
              MAY BE 0 OR ANY POSITIVE VALUE.  IF NDER IS LESS
              THAN 0, IT WILL BE TREATED AS 0.
    X -       THE ARGUMENT AT WHICH THE POLYNOMIAL AND ITS
              DERIVATIVES ARE TO BE EVALUATED.
    A -       WORK AND OUTPUT ARRAY CONTAINING VALUES FROM LAST
              CALL TO  POLFIT .
```

```
      OUTPUT:
          YFIT -    VALUE OF THE FITTING POLYNOMIAL OF ORDER  L  AT  X
          YP -      ARRAY CONTAINING THE FIRST THROUGH  NDER  DERIVATIVES
                    OF THE POLYNOMIAL OF ORDER  L .  YP  MUST BE
                    DIMENSIONED AT LEAST  NDER  IN THE CALLING PROGRAM.


QN    QN    QN    QN    QN    QN       QN    QN    QN    QN    QN    QN
                 ****************************************
                      ******************************
                           ********************
                                **********
      SUBROUTINE QN(FOFX,NEQN,X,MBAND,DISMAX,RELERR,ABSERR,IFLAG,RES,
     1  WORK,IWORK)


   WRITTEN BY L. F. SHAMPINE AND M. K. GORDON, 5122

*********************************************************************
   ABSTRACT
*********************************************************************
   SUBROUTINE QN SOLVES A SYSTEM OF NEQN SIMULTANEOUS, NONLINEAR
   EQUATIONS IN NEQN UNKNOWNS.  THAT IS, IT SOLVES THE PROBLEM F(X) = 0
   WHERE X IS A VECTOR WITH COMPONENTS X(1),...,X(NEQN) AND F IS A
   VECTOR OF NONLINEAR FUNCTIONS F(1),...,F(NEQN) EACH OF THE FORM

            F(I)(X(1),...,X(NEQN)) = 0.0

   THE SOLUTION VECTOR X IS LOCATED BY A QUASI-NEWTON ITERATION
   SCHEME IN WHICH THE EQUATIONS ARE REPEATEDLY LINEARIZED AND SOLVED
   UNTIL SUCCESSIVE ITERATES CONVERGE.  THE USER SUPPLIES AN INITIAL
   GUESS FOR THE SOLUTION AND A REGION ABOUT THAT GUESS KNOWN TO
   CONTAIN THE SOLUTION.

   DETAILS OF THE USE OF THE CODE AND OF THE ALGORITHM ARE FOUND IN
   SAND 75-0450.

*********************************************************************
   DECLARATIONS
*********************************************************************
            IN THE CALLING PROGRAM THE USER MUST DECLARE FOFX IN AN
            EXTERNAL STATEMENT AND DIMENSION THE ARRAYS X AND DISMAX
            AT LEAST NEQN, THE ARRAY WORK AT LEAST 2*NEQN**2+8*NEQN+4
            AND IWORK AT LEAST NEQN+7
*********************************************************************
   ON INPUT THE PARAMETERS IN THE CALL LIST ARE
*********************************************************************
            FOFX - EXTERNAL SUBROUTINE OF THE FORM FOFX(X,F) TO
                EVALUATE THE EQUATIONS
                  F(I)(X(1),X(2),...,X(NEQN))=0,I=1,...,NEQN
            NEQN - NUMBER OF EQUATIONS TO BE SOLVED.  (DIMENSION OF
                X AND F IN FOFX)
            X(*) - VECTOR CONTAINING AN INITIAL GUESS FOR THE SOLUTION
            MBAND - THE SYSTEM OF EQUATIONS IS SAID TO HAVE HALF BAND
                WIDTH MBAND IF FOR EACH I, EQUATION I INVOLVES ONLY
                THE VARIABLES X(J) WITH ABS(I-J) .LE. MBAND.  IF THERE
                IS NO BAND STRUCTURE OR THE STRUCTURE IS NOT KNOWN,
                SET MBAND .GE. NEQN/2
            DISMAX(*) - VECTOR SPECIFYING SIZE OF REGION IN WHICH
                SOLUTION IS SOUGHT.  THE I-TH COMPONENT OF THE SOLUTION
                MUST BE BETWEEN X(I)-DISMAX(I) AND X(I)+DISMAX(I).
                ALL COMPONENTS OF DISMAX MUST BE POSITIVE NUMBERS
            RELERR,ABSERR - RELATIVE AND ABSOLUTE ERROR TOLERANCES
                RESPECTIVELY IN THE CONVERGENCE TEST.  THE ITERATES ARE
                SAID TO HAVE CONVERGED WHEN THE CODE BELIEVES THAT
                ABS(ERROR(I)) .LE. RELERR*ABS(X(I)) + ABSERR
                                          FOR I=1,2,...,NEQN
                NOTE THE DANGER OF TAKING ABSERR=0.0 IF ANY SOLUTION
                COMPONENT IS ZERO
```

```
          IFLAG - INDICATOR FOR STATUS OF COMPUTATICN.  ON FIRST CALL
              SET IT TO 1
          WORK(*) - REAL ARRAY USED FOR INTERNAL STORAGE
          IWORK(*) - INTEGER ARRAY FOR INTERNAL STORAGE
*****************************************************************
  ON OUTPUT THE ALTERED PARAMETERS ARE
*****************************************************************
          X(*) - THE CURRENT APPROXIMATE SOLUTION (ITERATE)
          IFLAG - INDICATOR OF STATUS OF COMPUTATION
                = 2  THE ITERATES CONVERGED
                = 3  THE RESIDUAL OF THE CURRENT ITERATE IS TEN
                     ORDERS CF MAGNITUDE SMALLER THAN THAT OF
                     INITIAL GUESS
                = 4  TOO MUCH WORK.  THE ITERATES APPEAR TO BE
                     CONVERGING SLOWLY
                = 5  TOO MUCH WORK.  THE ITERATES DO NOT APPEAR TO BE
                     CONVERGING.  SEE IFLAG=6 FOR POSSIBLE REMEDIES
                = 6  THE ITERATES ARE NOT CONVERGING.  POSSIBLE
                     REASCNS ARE
                          1.   A POOR INITIAL GUESS.  CHOOSE
                               ANOTHER INITIAL GUESS
                          2.   A POORLY SCALED PROBLEM.  RESCALE
                               THE INDEPENDENT VARIABLES X AND/OR THE
                               DEPENDENT VARIABLES F
                          3.   THE ACCURACY REQUESTED IS NOT POSSIBLE
                               DUE TO ERROR IN THE FUNCTION
                               EVALUATIONS.  INCREASE ERRCR
                               TOLERANCES OR EVALUATE FUNCTIONS
                               MORE ACCURATELY.
                = 7  THE JACOBIAN MATRIX FORMED BY DIFFERENCING
                     APPEARS SINGULAR.  CHOSE ANOTHER INITIAL GUESS,
                     RESCALE, INCREASE DISMAX, OR REORDER THE
                     EQUATICNS AND TRY AGAIN
                = 8  SUCCESSIVE ITERATES LIE OUTSIDE THE REGION
                     SPECIFIED BY DISMAX.  CHOOSE ANOTHER INITIAL
                     GUESS OR INCREASE DISMAX AND TRY AGAIN
                = 9  INVALID INPUT. VALID INPUT IS
                        NEQN .GE. 1
                        MBAND .GE. 1
                        DISMAX(I) .GT. 0, I=1,...,NEQN
                        RELERR .GE. 0, ABSERR .GE. 0, AND
                        AMAX1(RELERR,ABSERR) .GT. 0
                        1 .LE. IFLAG .LE. 8
          RES - THE SIZE OF THE RESIDUAL F. (THE LENGTH OF
              THE VECTOR)
          WORK(*) - COMPONENTS 2,3,...,NEQN+1 CONTAIN THE ITERATE
              ASSOCIATED WITH THE SMALLEST RESIDUAL SEEN IN THE
              COMPUTATION AND WORK(1) CONTAINS THE SIZE OF
              THAT RESIDUAL.  USUALLY, BUT NOT ALWAYS, THESE ARE THE
              SAME VALUES AS STORED IN X AND RES, RESPECTIVELY.

  THE USER MAY CONTINUE THE ITERATION PROCESS BY CALLING QN AGAIN WITH
  THE OUTPUT VALUES OF X AND IFLAG.  THIS IS REASONABLE IN THE CASE
  IFLAG=3 WHEN THE SOLUTION RETURNED IS NOT SUFFICIENTLY ACCURATE; IT
  MAY BE REASONABLE WHEN IFLAG=4.  WITH ALL REMAINING FLAGS FOR
  NON-CONVERGENCE, THE USER SHOULD TAKE THE SUGGESTED ACTION AND
  CALL QN AGAIN WITH IFLAG = 1 (RESTART).


  THE USER HAS THE OPTION OF EXAMINING THE SOLUTION VECTOR AND
  RESIDUAL AFTER EACH ITERATION.  TO DO SO, HE MUST SUPPLY A
  SUBROUTINE NAMED QNCHK OF THE FORM SUBROUTINE QNCHK(X,RES) WHICH
  PRINTS OUT OR OTHERWISE USES THE INFORMATION.  FOR EXAMPLE,

     SUBROUTINE QNCHK(X,RES)
     DIMENSION X(1)
     PRINT 1,X,RES

   1 FORMAT(...)
     RETURN
     END
```

```
QNC3    QNC3    QNC3    QNC3    QNC3       QNC3    QNC3    QNC3    QNC3
              ****************************************
                   ******************************
                        ********************
                             **********
SUBROUTINE QNC3 (FUN,A,B,ERR,ANS,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

ORIGINAL TECHNIQUE WAS ALGORITHM 182 CACM 6 (1963) 315
PRESENT VERSION BY R E JONES, SANDIA LABORATORIES
SALIENT FEATURES -- INTERVAL BISECTION, COMBINED RELATIVE/ABSOLUTE
ERROR CONTROL, ESTIMATION OF TOTAL QUADRATURE ERROR, COMPUTED
MAXIMUM REFINEMENT LEVEL WHEN A IS CLOSE TO B.

ABSTRACT
   QNC3 INTEGRATES REAL FUNCTIONS OF ONE VARIABLE OVER FINITE
   INTERVALS, USING AN ADAPTIVE SIMPSONS-RULE (3-POINT NEWTON-
   COTES) ALGORITHM.  FOR VALUES OF ERR SMALLER THAN ABOUT 1.0E-6
   QNC3 BECOMES RELATIVELY INEFFICIENT AND QNC7 OR GAUS8 SHOULD
   BE USED INSTEAD.

DESCRIPTION OF ARGUMENTS

   INPUT--
   FUN - NAME OF EXTERNAL FUNCTION TO BE INTEGRATED.  THIS NAME
         MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM.
         FUN MUST BE A FUNCTION OF ONE REAL ARGUMENT.  THE VALUE
         OF THE ARGUMENT TO FUN IS THE VARIABLE OF INTEGRATION
         WHICH RANGES FROM A TO B.
   A   - LOWER LIMIT OF INTEGRAL
   B   - UPPER LIMIT OF INTEGRAL (MAY BE LESS THAN A)
   ERR - IS A REQUESTED ERROR TOLERANCE.  NORMALLY PICK A VALUE OF
         ABS(ERR).LT.1.E-3.  ANS WILL NORMALLY HAVE NO MORE ERROR
         THAN ABS(ERR) TIMES THE INTEGRAL OF THE ABSOLUTE VALUE
         OF FUN(X).  USUALLY, SMALLER VALUES FOR ERR YIELD
         MORE ACCURACY AND REQUIRE MORE FUNCTION EVALUATIONS.
         A NEGATIVE VALUE FOR ERR CAUSES AN ESTIMATE OF THE
         ABSOLUTE ERROR IN ANS TO BE RETURNED IN ERR.

   OUTPUT--
   ERR - WILL BE AN ESTIMATE OF THE ERROR IN ANS IF THE INPUT
         VALUE OF ERR WAS NEGATIVE.  THE ESTIMATED ERROR IS SOLELY
         FOR INFORMATION TO THE USER AND SHOULD NOT BE USED AS
         A CORRECTION TO THE COMPUTED INTEGRAL.
   ANS - COMPUTED VALUE OF INTEGRAL
   IERR- A STATUS CODE
       --NORMAL CODES
           1 ANS MOST LIKELY MEETS REQUESTED ERROR TOLERANCE,
             OR A=B.
          -1 A AND B ARE TOO NEARLY EQUAL TO ALLOW NORMAL
             INTEGRATION.  ANS IS SET TO ZERO.
       --ABNORMAL CODE
           2 ANS PROBABLY DOES NOT MEET REQUESTED ERROR TOLERANCE.
```

```
QNC7    QNC7    QNC7    QNC7    QNC7       QNC7    QNC7    QNC7    QNC7
              ****************************************
                     ******************************
                          ********************
                               **********
      SUBROUTINE QNC7 (FUN,A,B,ERR,ANS,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 6322
              T. H. JEFFERSON   -   DIVISION 6322

      ORIGINAL ROUTINE BY DAVID L. KAHANER, LASL
      PRESENT VERSION BY R E JONES, SANDIA LABORATORIES
      SALIENT FEATURES -- INTERVAL BISECTION, COMBINED RELATIVE/ABSOLUTE
      ERROR CONTROL, ESTIMATION OF TOTAL QUADRATURE ERROR, COMPUTED
      MAXIMUM REFINEMENT LEVEL WHEN A IS CLOSE TO B.

      ABSTRACT
         QNC7  INTEGRATES REAL FUNCTIONS OF ONE VARIABLE OVER FINITE
         INTERVALS, USING AN ADAPTIVE 7-POINT NEWTON-COTES ALGORITHM.
         QNC7 IS EFFICIENT OVER A WIDE RANGE OF ACCURACIES, BUT QNC3
         MAY BE MORE EFFICIENT ON DIFFICULT LOW ACCURACY PROBLEMS,
         AND GAUS8 MAY BE MORE EFFICIENT ON HIGH ACCURACY PROBLEMS
         (ERR LESS THAN 1.0E-8, SAY) OR ON PROBLEMS INVOLVING VERY
         SMOOTH FUNCTIONS.

      DESCRIPTION OF ARGUMENTS

         INPUT--
         FUN - NAME OF EXTERNAL FUNCTION TO BE INTEGRATED.  THIS NAME
               MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM.
               FUN MUST BE A FUNCTION OF ONE REAL ARGUMENT.  THE VALUE
               OF THE ARGUMENT TO FUN IS THE VARIABLE OF INTEGRATION
               WHICH RANGES FROM A TO B.
         A   - LOWER LIMIT OF INTEGRAL
         B   - UPPER LIMIT OF INTEGRAL (MAY BE LESS THAN A)
         ERR - IS A REQUESTED ERROR TOLERANCE.  NORMALLY PICK A VALUE OF
               ABS(ERR).LT.1.E-3.  ANS WILL NORMALLY HAVE NO MORE ERROR
               THAN ABS(ERR) TIMES THE INTEGRAL OF THE ABSOLUTE VALUE
               OF FUN(X).  USUALLY, SMALLER VALUES FOR ERR YIELD
               MORE ACCURACY AND REQUIRE MORE FUNCTION EVALUATIONS.
               A NEGATIVE VALUE FOR ERR CAUSES AN ESTIMATE OF THE
               ABSOLUTE ERROR IN ANS TO BE RETURNED IN ERR.

         OUTPUT--
         ERR - WILL BE AN ESTIMATE OF THE ERROR IN ANS IF THE INPUT
               VALUE OF ERR WAS NEGATIVE.
         ANS - COMPUTED VALUE OF INTEGRAL
         IERR- A STATUS CODE
             --NORMAL CODES
                1 ANS MOST LIKELY MEETS REQUESTED ERROR TOLERANCE,
                  OR A=B.
               -1 A AND B ARE TOO NEARLY EQUAL TO ALLOW NORMAL
                  INTEGRATION.  ANS IS SET TO ZERO.
             --ABNORMAL CODE
                2 ANS PROBABLY DOES NOT MEET REQUESTED ERROR TOLERANCE.
```

```
RBND2    RBND2    RBND2    RBND2    RBND2    RBND2    RBND2    RBND2
          ****************************************
               ******************************
                    ********************
                         **********
      SUBROUTINE RBND2(N,COEF,WR,WI,ABSERR,RELERR,KLUST,KER)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY CARL B. BAILEY AND MODIFIED BY WILLIAM R. GAVIN

      ABSTRACT

          THIS ROUTINE COMPUTES ERROR BOUNDS AND CLUSTER COUNTS
          FOR APPROXIMATE ZEROS OF A POLYNOMIAL WITH REAL COEFFICIENTS.
          THE ZEROS MAY HAVE BEEN COMPUTED BY ANY APPROPRIATE ROUTINE.
          (FOR EXAMPLE, BY RPQR)
          THE METHOD USED IS BASED ON THE FACT THAT THE VALUE OF A
          POLYNOMIAL AT ANY POINT IS EQUAL TO THE LEADING COEFFICIENT
          TIMES THE PRODUCT OF THE DISTANCES FROM THAT POINT TO EACH
          OF THE ZEROES.  GIVEN THE VALUE OF THE POLYNOMIAL AT AN
          APPROXIMATE ZERO, RBND2 COMPUTES FOR EACH APPROXIMATE ZERO
          THE RADIUS OF A CIRCLE ABOUT THAT APPROXIMATE ZERO WHICH
          CONTAINS A TRUE ZERO OF THE POLYNOMIAL.  USING THE KNOWN
          DISTRIBUTION OF APPROXIMATE ZEROES, AN ITERATIVE PROCEDURE
          IS USED TO SHRINK THE RADII OF THE CIRCLES.

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               COEF(N+1), WR(N), WI(N), ABSERR(N), RELERR(N), KLUST(N)

          INPUT---
          N      - DEGREE OF THE POLYNOMIAL (NUMBER OF ZEROS).
          COEF   - REAL ARRAY OF N+1 COEFFICIENTS IN ORDER OF DESCENDING
                   POWERS OF Z, I.E.
                   P(Z) = COEF(1)*(Z**N) + ... + COEF(N)*Z + COEF(N+1)
          WR     - REAL ARRAY OF N REAL PARTS OF APPROXIMATE ZEROS.
          WI     - REAL ARRAY OF N IMAGINARY PARTS OF APPROXIMATE ZEROS.

          OUTPUT--
          ABSERR - REAL ARRAY OF ABSOLUTE ERROR BOUNDS.  ABSERR(I) IS
                   THE ABSOLUTE ERROR BOUND IN THE ZERO (WR(I),WI(I)).
          RELERR - REAL ARRAY OF RELATIVE ERROR BOUNDS.  RELERR(I) IS
                   THE RELATIVE ERROR BOUND IN THE ZERO (WR(I),WI(I)).
          KLUST  - INTEGER ARRAY OF CLUSTER COUNTS FOR ZEROS.  THE TRUE
                   ZERO CORRESPONDING TO I-TH APPROXIMATE ZERO LIES IN
                   A CIRCLE OF RADIUS ABSERR(I).  KLUST(I) IS THE NUMBER
                   OF CIRCLES INCLUDING THE I-TH CIRCLE WHICH OVERLAP
                   THE I-TH CIRCLE.  THE CLUSTER COUNT OFTEN INDICATES
                   THE MULTIPLICITY OF A ZERO.
          KER    - AN ERROR FLAG
                  -NORMAL CODE
                   0  MEANS THE BOUNDS AND COUNTS WERE COMPUTED.
                  -ABNORMAL CODES
                   1  N (DEGREE) MUST BE .GE. 1
                   2  LEADING COEFFICIENT IS ZERO
```

```
RDET    RDET    RDET    RDET    RDET       RDET    RDET    RDET    RDET
              ************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE RDET(ND,N,A,DET,KER)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY CARL B. BAILEY, MAY 1972.

      ABSTRACT
          RDET EVALUATES THE DETERMINANT OF A REAL MATRIX -A-.
          THE MATRIX -A- IS DECOMPOSED BY GAUSSIAN ELIMINATION INTO THE
          PRODUCT OF TRIANGULAR FACTORS -L- AND -U-.  THE DETERMINANT OF
          -A- IS COMPUTED THEN AS THE DETERMINANT OF -L- TIMES THE
          DETERMINANT OF -U-.

          RDET SHOULD NOT BE USED TO SOLVE SYSTEMS OF LINEAR ALGEBRAIC
          EQUATIONS, SAY BY CRAMER-S RULE.  SYSTEMS OF LINEAR ALGEBRAIC
          EQUATIONS SHOULD BE SOLVED DIRECTLY USING SAXB OR SAXBI.
          RDET CALLS THE ROUTINE RLUD TO PERFORM LU DECOMPOSITION.

      REFERENCE
        1.  G.E.FORSYTHE AND C.B.MOLER, COMPUTER SOLUTION OF LINEAR
            ALGEBRAIC EQUATIONS, PRENTICE-HALL, 1967

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               A(ND,N)

        --INPUT---
          ND      - THE ACTUAL FIRST DIMENSION OF THE ARRAY -A-.
          N       - NUMBER OF ROWS IN MATRIX -A- (1 .LE. N .LE. ND)
          A       - AN ARRAY DIMENSIONED WITH EXACTLY -ND- ROWS AND
                    AT LEAST -N- COLUMNS.  THE -N- BY -N- LEADING
                    SUBARRAY MUST CONTAIN THE COEFFICIENT MATRIX -A-.

        --OUTPUT--
          A       - WILL BE DESTROYED DURING EVALUATION OF A DETERMINANT.
          DET     - WILL BE THE DETERMINANT OF -A- UNLESS KER .NE. 0.
          KER     - AN ERROR CODE
                  --NORMAL CODES
                    0 MEANS NO ERRORS WERE DETECTED
                  --ABNORMAL CODES
                    1 MEANS -ND- WAS NOT IN THE RANGE 1 .LE. ND .LE. 325
                    2 MEANS -N- WAS NOT IN THE RANGE 1 .LE. N .LE. ND.
```

```
RFFT    RFFT    RFFT    RFFT    RFFT        RFFT    RFFT    RFFT    RFFT
                ****************************************
                     ******************************
                          ********************
                               **********

      IDENT RFFT

    SUBROUTINE RFFT(DATA,N)

    ABSTRACT
       RFFT COMPUTES THE ONE-DIMENSIONAL FAST FOURIER TRANSFORM OF
       REAL DATA OF LENGTH N, WHERE N IS A POWER OF TWO.
       IF N IS A POWER OF TWO, THE COMPUTATION PERFORMED BY THE CALL,
          CALL RFFT(DATA,N)
       IS EQUIVALENT TO THAT PERFORMED BY THE CALL,
          CALL FOURTR(DATA,N,-1,0)
       EXCEPT THAT RFFT IS THREE TO FIVE TIMES FASTER THAN FOURTR.
       EXECUTION TIME FOR RFFT IS ABOUT 2.8E-6*N*LOG2(N) SECONDS
       ON THE CDC6600.  FOR N=1024, THIS IS ABOUT 28 MILLISECONDS.

       RFFT IS WRITTEN IN THE CDC6600 ASSEMBLY LANGUAGE, COMPASS.
       THUS, IT IS NOT CONVERTIBLE TO OTHER COMPUTER SYSTEMS.
       USERS NEEDING MACHINE INDEPENDENCE SHOULD USE THE
       FOURTR/FOURTH/FOURT PACKAGE (OR OTHER COMPARABLE ROUTINES
       AVAILABLE FROM THE MATH LIBRARY PROJECT) INSTEAD OF RFFT/RFFTI.

       RFFT COMPUTES ONLY THE NON-REDUNDANT COEFFICIENTS OF THE
       DISCRETE FOURIER TRANSFORM.  THAT IS, IT COMPUTES THE
       FOURIER COSINE AND SINE COEFFICIENTS FOR 0 TO N/2 CYCLES
       OVER THE GIVEN TIME (OR SPACE) INTERVAL.  THESE TWO
       COEFFICIENTS, FOR THE FREQUENCY OF K CYCLES OVER THE GIVEN
       INTERVAL, ARE DEFINED AS FOLLOWS--

                                N
          COS COEF(K) =   SUM ( DATA(I)*COS(2*PI*(I-1)*K/N) )
          SIN COEF(K) = -SUM ( DATA(I)*SIN(2*PI*(I-1)*K/N) )
                         I=1

       THESE TWO COEFFICIENTS ARE RETURNED IN DATA(2*K+1) AND
       DATA(2*K+2), RESPECTIVELY, FOR K=0 TO N/2.
       THUS, THE ARRAY, DATA, MUST BE DIMENSIONED AT LEAST
       2*(N/2+1) = N+2.  COEFFICIENTS FOR FREQUENCIES
       FROM N/2+1 TO N-1, IF DESIRED, MAY BE COMPUTED BY THE SIMPLE
       RELATIONS,
          COS COEF(N-K) =    COS COEF(K)
          SIN COEF(N-K) = - SIN COEF(K)
       NOTE THAT THE INPUT VALUES MUST CORRESPOND TO EQUALLY
       SPACED TIME (OR SPACE) VALUES.  NOTE ALSO THAT SIN COEF(0)
       AND SIN COEF(N/2) WILL ALWAYS BE ZERO.

       SEE SUBROUTINE RFFTI FOR THE CORRESPONDING INVERSE TRANSFORM.

    DESCRIPTION OF ARGUMENTS
       THE USER MUST DIMENSION THE ARRAY, DATA(N+2)

       INPUT---
       DATA - REAL ARRAY WHICH CONTAINS THE DATA TO BE TRANSFORMED.
              DATA MUST BE DIMENSIONED AT LEAST N+2, THE FIRST N
              WORDS CONTAINING THE VALUES TO BE TRANSFORMED.
       N    - NUMBER OF VALUES TO BE TRANSFORMED.  N MUST BE A
              POWER OF TWO, AND  IT MUST BE IN THE RANGE OF 4 TO
              65536=2**16.  IF N IS NOT A POWER OF TWO OR IS OUT
              OF THE STATED RANGE, A FATAL PROGRAM ERROR WILL RESULT.

       OUTPUT---
       DATA - WILL CONTAIN THE COSINE AND SINE COEFFICIENTS FOR
              FREQUENCIES 0 TO N/2, AS DESCRIBED IN THE ABSTRACT.

    REFERENCES
    (1) R C SINGLETON, *ON COMPUTING THE FAST FOURIER TRANSFORM*,
        COMM. ACM, VOL 10, 1967, PP 647-654.
    (2) LASL LIBRARY ROUTINE LA-F501A, BY B R HUNT.

    AUTHOR
```

```
THE ORIGINAL COMPASS VERSION OF THIS CODE WAS WRITTEN
BY B R HUNT OF LASL.  THIS VERSION WAS PREPARED FOR THE
SANDIA MATH LIBRARY BY R E JONES, DIV 2642, APRIL 1975
```

```
RFFTI    RFFTI    RFFTI    RFFTI       RFFTI    RFFTI    RFFTI    RFFTI
              ****************************************
                   ******************************
                        ********************
                             **********

      IDENT  RFFTI

   SUBROUTINE RFFTI(DATA,N)

   ABSTRACT
      RFFTI COMPUTES THE ONE-DIMENSIONAL INVERSE FAST FOURIER
      TRANSFORM, GIVEN N/2+1 COSINE AND N/2+1 SINE COEFFICIENTS
      IN THE FORM RETURNED BY RFFT (OR BY FOURTR, OR BY FOURT, IF
      IFORM=0 WAS USED), WHERE N IS A POWER OF TWO.
      IF N IS A POWER OF TWO, THE COMPUTATION PERFORMED BY THE CALL,
         CALL RFFTI(DATA,N)
      IS EQUIVALENT TO THAT PERFORMED BY THE CALL,
         CALL FOURTH(DATA,N,+1,0)
      EXCEPT THAT RFFTI IS THREE TO FIVE TIMES FASTER THAN FOURTH.
      EXECUTION TIME FOR RFFTI IS ABOUT 2.9E-6*N*LOG2(N) SECONDS
      ON THE CDC6600.  FOR N=1024, THIS IS ABOUT 29 MILLISECONDS.

      RFFTI IS WRITTEN IN THE CDC6600 ASSEMBLY LANGUAGE, COMPASS.
      THUS, IT IS NOT READILY CONVERTIBLE TO OTHER COMPUTER SYSTEMS.
      USERS NEEDING MACHINE INDEPENCE SHOULD USE THE
      FOURTR/FOURTH/FOURT PACKAGE (OR OTHER COMPARABLE ROUTINES
      AVAILABLE FROM THE LIBRARY PROJECT) INSTEAD OF RFFT/RFFTI.

      THE CALCULATION PERFORMED BY RFFTI IS EQUIVALENT
      TO THE FOLLOWING, WHERE THE SUMS ARE FROM I=1 TO I=N/2-1,
      AND K=1 TO N.

         RESULT(K) =    DATA(1)
                     + 2*SUM( DATA(2*I+1)*COS(2*PI*I*(K-1)/N) )
                     - 2*SUM( DATA(2*I+2)*SIN(2*PI*I*(K-1)/N) )
                     +   DATA(N+1)*COS(PI*(K-1))

      THUS, FOR I=0 TO N/2, THE INPUT DATA(2*I+1) AND DATA(2*I+2)
      MUST BE THE COSINE AND SINE COEFFICIENTS FOR THE FREQUENCY OF
      I*DF, WHERE DF IS THE FREQUENCY SPACING.
      (NOTE THAT THE INPUT DATA(2) AND DATA(N+2) ARE ASSUMED TO = 0.)
      RESULT(1) TO RESULT(N) WILL BE RETURNED IN DATA(1) TO DATA(N),
      AND DATA(N+1) AND DATA(N+2) WILL BE SET TO ZERO.
      RESULT(I) WILL CORRESPOND TO A TIME VALUE OF (I-1)/(N*DF),
      FOR I=1 TO N.

   NOTE-- A CALL TO RFFT FOLLOWED BY A CALL TO RFFTI  (WITH NO
      NO OTHER CALCULATIONS DONE IN BETWEEN) WILL RESULT IN
      MULTIPLICATION OF THE ORIGINAL DATA BY THE VALUE OF N.
      THIS FACTOR OF N MUST BE ACCOUNTED FOR AS APPROPRIATE
      IN THE GIVEN APPLICATION.

      SEE SUBROUTINE RFFT FOR THE CORRESPONDING FORWARD TRANSFORM.

   DESCRIPTION OF ARGUMENTS
      THE USER MUST DIMENSION THE ARRAY, DATA(N+2)

   INPUT---
      DATA - REAL OR COMPLEX ARRAY CONTAINING THE N/2+1 PAIRS OF
             COEFFICIENTS, IN THE FORM DISCUSSED ABOVE.  (I.E., IN
             THE FORM RETURNED BY RFFT.)
      N    - THE NUMBER OF REAL VALUES THAT ARE TO RESULT FROM
             THE INVERSE TRANSFORM.  N MUST BE A POWER OF TWO
```

```
            AND IT MUST BE IN THE RANGE OF 4 TO 65536=2**16.
            IF N IS NOT A POWER OF TWO OR IT IS OUT OF THE STATED
            RANGE, A FATAL PROGRAM ERROR WILL RESULT.

   OUTPUT---
      DATA - WILL CONTAIN THE REQUESTED INVERSE TRANSFORM IN
             DATA(1) TO DATA(N). DATA(N+1) AND DATA(N+2) WILL = 0.

   REFERENCES
   (1) R C SINGLETON, *ON COMPUTING THE FAST FOURIER TRANSFORM*,
       COMM. ACM, VOL 10, 1967, PP 647-654.
   (2) LASL LIBRARY ROUTINE LA-F502A, BY B R HUNT.

   AUTHOR
      THE ORIGINAL COMPASS VERSION OF THIS CODE WAS WRITTEN
      BY B R HUNT OF LASL.  THIS VERSION WAS PREPARED FOR THE
      SANDIA MATH LIBRARY BY R E JONES, DIV 2642, APRIL 1975
```

```
RKF    RKF    RKF    RKF    RKF         RKF    RKF    RKF    RKF    RKF
                 ****************************************
                      ******************************
                           ********************
                                **********
     SUBROUTINE RKF(F,NEQ,Y,X,XOP,RELERR,ABSERR,IFLAG,WORK,IWORK)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     FEHLBERG FOURTH-FIFTH ORDER RUNGE-KUTTA METHOD

     WRITTEN BY H.A.WATTS AND L.F.SHAMPINE

    RKF IS PRIMARILY DESIGNED TO SOLVE NON-STIFF AND MILDLY STIFF
    DIFFERENTIAL EQUATIONS WHEN DERIVATIVE EVALUATIONS ARE CHEAP.
    RKF SHOULD GENERALLY NOT BE USED WHEN THE USER IS DEMANDING
    HIGH ACCURACY. INSTEAD, USE SUBROUTINE  ODE , AND FOR STIFF
    PROBLEMS USE SUBROUTINE  STIFF (AVAILABLE IN MATH2 LIBRARY).

    THE CODE ATTEMPTS TO JUDGE WHETHER OR NOT THE GIVEN PROBLEM CAN BE
    EFFICIENTLY SOLVED BY RKF.  THIS DECISION IS BASED UPON THE
    REQUESTED ACCURACY, THE NUMBER OF DIFFERENTIAL EQUATIONS, AND THE
    REAL TIME COST INCURRED IN SOLVING THE PROBLEM.  THE COST
    EFFECTIVENESS OF RKF IS ROUGHLY COMPARED TO THE USE OF ODE.

*********************************************************************
 ABSTRACT
*********************************************************************

    SUBROUTINE  RKF  INTEGRATES A SYSTEM OF NEQ FIRST ORDER
    ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
             DY(I)/DX = F(X,Y(1),Y(2),...,Y(NEQ))
              WHERE THE Y(I) ARE GIVEN AT X .
    TYPICALLY THE SUBROUTINE IS USED TO INTEGRATE FROM X TO XOP BUT IT
    CAN BE USED AS A ONE-STEP INTEGRATOR TO ADVANCE THE SOLUTION A
    SINGLE STEP IN THE DIRECTION OF XOP.  ON RETURN THE PARAMETERS IN
    THE CALL LIST ARE SET FOR CONTINUING THE INTEGRATION. THE USER HAS
    ONLY TO CALL RKF AGAIN (AND PERHAPS DEFINE A NEW VALUE FOR XOP).
    ACTUALLY, RKF IS AN INTERFACING ROUTINE WHICH CALLS SUBROUTINE RKFS
    FOR THE SOLUTION.  RKFS  IN TURN CALLS SUBROUTINE  FEHL  WHICH
    COMPUTES AN APPROXIMATE SOLUTION OVER ONE STEP.

    RKF  USES THE RUNGE-KUTTA-FEHLBERG (4,5)  METHOD DESCRIBED
    IN THE REFERENCE
    E.FEHLBERG , LOW-ORDER CLASSICAL RUNGE-KUTTA FORMULAS WITH STEPSIZE
```

```
              CONTROL , NASA TR R-315

    THE PARAMETERS REPRESENT-
      F -- SUBROUTINE F(X,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DX
      NEQ -- NUMBER OF EQUATIONS TO BE INTEGRATED
      Y(*) -- SOLUTION VECTOR AT X
      X -- INDEPENDENT VARIABLE
      XOP -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED
      RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
            ERROR TEST. AT EACH STEP THE CODE REQUIRES THAT
                 ABS(LOCAL ERROR) .LE. RELERR*ABS(Y) + ABSERR
            FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
      IFLAG -- INDICATOR FOR STATUS OF INTEGRATION
      WORK(*) -- ARRAY TO HOLD INFORMATION INTERNAL TO RKF WHICH IS
            NECESSARY FOR SUBSEQUENT CALLS. MUST BE DIMENSIONED
            AT LEAST  3+6*NEQ
      IWORK(*) -- INTEGER ARRAY USED TO HOLD INFORMATION INTERNAL TO
            RKF WHICH IS NECESSARY FOR SUBSEQUENT CALLS. MUST BE
            DIMENSIONED AT LEAST  6

*********************************************************************
  FIRST CALL TO RKF
*********************************************************************

    THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
    IN THE CALL LIST  -       Y(NEQ) , WORK(3+6*NEQ) , IWORK(6)  ,
    DECLARE F IN AN EXTERNAL STATEMENT, SUPPLY SUBROUTINE F(X,Y,YP) AND
    INITIALIZE THE FOLLOWING PARAMETERS-

      NEQ -- NUMBER OF EQUATIONS TO BE INTEGRATED.  (NEQ .GE. 1)
      Y(*) -- VECTOR OF INITIAL CONDITIONS
      X -- STARTING POINT OF INTEGRATION , MUST BE A VARIABLE
      XOP -- OUTPUT POINT AT WHICH SOLUTION IS DESIRED.
            X=XOP IS ALLOWED ON THE FIRST CALL ONLY, IN WHICH CASE RKF
            RETURNS WITH IFLAG=2 IF CONTINUATION IS POSSIBLE.
      RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
            WHICH MUST BE NON-NEGATIVE BUT MAY BE CONSTANTS. THE CODE
            SHOULD NORMALLY NOT BE USED WITH RELATIVE ERROR TOLERANCES
            SMALLER THAN ABOUT 1.E-8. TO AVOID LIMITING PRECISION
            DIFFICULTIES THE CODE ALWAYS USES THE LARGER OF RELERR
            AND REMIN FOR THE INTERNAL RELATIVE ERROR PARAMETER.
            REMIN IS A MACHINE DEPENDENT CONSTANT WHICH IS SET IN A
            DATA STATEMENT.  (REMIN = 1.E-12 FOR CDC6600)
      IFLAG -- +1,-1  INDICATOR TO INITIALIZE THE CODE FOR EACH NEW
            PROBLEM. NORMAL INPUT IS +1. THE USER SHOULD SET IFLAG=-1
            ONLY WHEN ONE-STEP INTEGRATOR CONTROL IS ESSENTIAL. IN THIS
            CASE, RKF ATTEMPTS TO ADVANCE THE SOLUTION A SINGLE STEP
            IN THE DIRECTION OF XOP EACH TIME IT IS CALLED. SINCE THIS
            MODE OF OPERATION RESULTS IN EXTRA COMPUTING OVERHEAD, IT
            SHOULD BE AVOIDED UNLESS NEEDED.

*********************************************************************
  OUTPUT FROM RKF
*********************************************************************

      Y(*) -- SOLUTION AT X
      X -- LAST POINT REACHED IN INTEGRATION.
      IFLAG = 2 -- INTEGRATION REACHED XOP. INDICATES SUCCESSFUL RETURN
                   AND IS THE NORMAL MODE FOR CONTINUING INTEGRATION.
            =-2 -- A SINGLE SUCCESSFUL STEP IN THE DIRECTION OF XOP HAS
                   BEEN TAKEN. NORMAL MODE FOR CONTINUING INTEGRATION
                   ONE STEP AT A TIME.
            = 3 -- INTEGRATION WAS NOT COMPLETED BECAUSE MORE THAN
                   6000 DERIVATIVE EVALUATIONS WERE NEEDED. THIS
                   IS APPROXIMATELY 1000 STEPS.
            = 4 -- INTEGRATION WAS NOT COMPLETED BECAUSE SOLUTION
                   VANISHED MAKING A PURE RELATIVE ERROR TEST
                   IMPOSSIBLE. MUST USE NON-ZERO ABSERR TO CONTINUE.
                   USING THE ONE-STEP INTEGRATION MODE FOR ONE STEP
                   IS A GOOD WAY TO PROCEED.
            = 5 -- INTEGRATION WAS NOT COMPLETED BECAUSE REQUESTED
```

```
                ACCURACY COULD NOT BE ACHIEVED USING SMALLEST
                ALLOWABLE STEPSIZE. USER MUST INCREASE THE ERROR
                TOLERANCE BEFORE CONTINUED INTEGRATION CAN BE
                ATTEMPTED.
          = 6 -- IT IS LIKELY THAT RKF IS INEFFICIENT FOR SOLVING
                THIS PROBLEM. USE SUBROUTINE ODE FOR NON-STIFF
                EQUATIONS AND SUBROUTINE STIFF FOR STIFF
                DIFFERENTIAL EQUATIONS.
          = 7 -- INVALID INPUT PARAMETERS (FATAL ERROR UNLESS
                OVERRIDDEN BY CALL TO ERXSET)
                THIS INDICATOR OCCURS IF ANY OF THE FOLLOWING IS
                SATISFIED -    NEQ .LE. 0
                               X=XOP  AND  IFLAG .NE. +1 OR -1
                               RELERR OR ABSERR .LT. 0.
                               IFLAG .EQ. 0  OR  .LT. -2  OR  .GT. 7
    WORK(*),IWORK(*) -- INFORMATION WHICH IS USUALLY OF NO INTEREST
                TO THE USER BUT NECESSARY FOR SUBSEQUENT CALLS.
                WORK(1),...,WORK(NEQ) CONTAIN THE FIRST DERIVATIVES
                OF THE SOLUTION VECTOR Y AT X. WORK(NEQ+1) CONTAINS
                THE STEPSIZE H TO BE ATTEMPTED ON THE NEXT STEP.
                IWORK(1) CONTAINS THE DERIVATIVE EVALUATION COUNTER.

*********************************************************************
  SUBSEQUENT CALLS TO RKF
*********************************************************************

    SUBROUTINE RKF RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE THE
    INTEGRATION. IF THE INTEGRATION REACHED XOP,THE USER NEED ONLY
    DEFINE A NEW XOP AND CALL RKF AGAIN. IN THE ONE-STEP INTEGRATOR
    MODE (IFLAG=-2) THE USER MUST KEEP IN MIND THAT EACH STEP TAKEN IS
    IN THE DIRECTION OF THE CURRENT XOP. UPON REACHING XOP (INDICATED
    BY CHANGING IFLAG TO 2),THE USER MUST THEN DEFINE A NEW XOP AND
    RESET IFLAG TO -2 TO CONTINUE IN THE ONE-STEP INTEGRATOR MODE.

    IF THE INTEGRATION WAS NOT COMPLETED BUT THE USER STILL WANTS TO
    CONTINUE (IFLAG=3 CASE), HE JUST CALLS RKF AGAIN. THE FUNCTION
    COUNTER IS THEN RESET TO 0 AND ANOTHER 6000 FUNCTION EVALUATIONS
    ARE ALLOWED.

    HOWEVER,IN THE CASE IFLAG=4, THE USER MUST FIRST ALTER THE ERROR
    CRITERION TO USE A POSITIVE VALUE OF ABSERR BEFORE INTEGRATION CAN
    PROCEED. IF HE DOES NOT,EXECUTION IS TERMINATED.

    ALSO,IN THE CASE IFLAG=5, IT IS NECESSARY FOR THE USER TO RESET
    IFLAG TO 2 (OR -2 WHEN THE ONE-STEP INTEGRATION MODE IS BEING USED)
    AS WELL AS INCREASING EITHER ABSERR,RELERR OR BOTH BEFORE THE
    INTEGRATION CAN BE CONTINUED. IF THIS IS NOT DONE, EXECUTION WILL
    BE TERMINATED. THE OCCURRENCE OF IFLAG=5 INDICATES A TROUBLE SPOT
    (SOLUTION IS CHANGING RAPIDLY,SINGULARITY MAY BE PRESENT) AND IT
    OFTEN IS INADVISABLE TO CONTINUE.

    IF IFLAG=6 IS ENCOUNTERED, THE USER SHOULD CONSIDER SWITCHING TO
    THE ADAMS CODES ODE/STEP,INTRP. IF THE USER INSISTS UPON CONTINUING
    THE INTEGRATION WITH RKF,HE MUST RESET IFLAG TO 2 (OR -2 WHEN THE
    ONE-STEP INTEGRATION MODE IS BEING USED) BEFORE CALLING RKF AGAIN.
    OTHERWISE,EXECUTION WILL BE TERMINATED.

    IF IFLAG=7 IS OBTAINED, INTEGRATION CAN NOT BE CONTINUED UNLESS
    THE INVALID INPUT PARAMETERS ARE CORRECTED.

    IT SHOULD BE NOTED THAT THE ARRAYS WORK,IWORK CONTAIN INFORMATION
    REQUIRED FOR SUBSEQUENT INTEGRATION. ACCORDINGLY, WORK AND IWORK
    SHOULD NOT BE ALTERED.

*********************************************************************
```

```
RNAA    RNAA    RNAA    RNAA    RNAA       RNAA    RNAA    RNAA    RNAA
              ****************************************
                   ******************************
                        ********************
                             **********
SUBROUTINE RNAA (NDIM,N,A,EVR,EVI,VEC,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

EISPACK IS AN EXTENSIVE COLLECTION OF ROUTINES FOR SOLVING
THE ALGEBRAIC EIGENVALUE PROBLEM.  THE ORIGINAL ALGOL ROUTINES
WERE WRITTEN BY J. H. WILKINSON, ET.AL., AND SUBSEQUENTLY WERE
TRANSLATED TO FORTRAN AND TESTED AT ARGONNE NATIONAL LABORATORY.
THIS INTERFACE TO EISPACK WAS WRITTEN BY W. R. GAVIN.

ABSTRACT
  THIS SUBROUTINE COMPUTES ALL EIGENVALUES AND CORRESPONDING
  EIGENVECTORS OF AN ARBITRARY REAL MATRIX.
  THE MATRIX IS BALANCED BY EXACT NORM REDUCING SIMILARITY
  TRANSFORMATIONS AND THEN IS REDUCED TO HESSENBERG FORM BY
  ELEMENTARY SIMILARITY TRANSFORMATIONS.  THE QR ALGORITHM
  IS USED TO COMPUTE THE EIGENSYSTEM OF THE HESSENBERG FORM.

  TO COMPUTE ONLY THE EIGENVALUES OF AN ARBITRARY REAL MATRIX
  SEE SUBROUTINE RNAN.  FOR EIGENSYSTEMS OF REAL SYMMETRIC
  MATRICES SEE SUBROUTINES RSAA AND RSAN.  FOR EIGENSYSTEMS OF
  COMPLEX MATRICES SEE CHAA, CHAN, CNAA, AND CNAN.

DESCRIPTION OF ARGUMENTS
 ON INPUT

    NDIM   MUST BE THE ROW DIMENSION OF THE ARRAYS A AND VEC
           IN THE CALLING PROGRAM DIMENSION STATEMENT.

    N      IS THE ORDER OF THE MATRIX.  N MUST NOT EXCEED NDIM.
           N*NDIM MUST NOT EXCEED 50625 = 225*225 = 142701(OCTAL).
           N MUST NOT EXCEED 225.  N MAY BE 1.

    A      AN ARRAY WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS.
           THE LEADING N BY N SUBARRAY MUST CONTAIN THE ARBITRARY
           REAL MATRIX WHOSE EIGENSYSTEM IS TO BE COMPUTED.

 ON OUTPUT

    EVR    CONTAINS THE REAL PARTS OF THE COMPUTED EIGENVALUES.

    EVI    CONTAINS THE IMAGINARY PARTS OF THE COMPUTED
           EIGENVALUES.  THE EIGENVALUES ARE NOT ORDERED
           IN ANY WAY.  HOWEVER CONJUGATE PAIRS OCCUR
           IN ADJACENT PLACES WITH THE EIGENVALUE OF
           POSITIVE IMAGINARY PART FIRST.

    VEC    CONTAINS THE COMPUTED EIGENVECTORS OF A
           IN THE COLUMNS OF THE N BY N LEADING SUBARRAY OF VEC.
           IF THE J-TH EIGENVALUE IS REAL, COLUMN J OF VEC
           CONTAINS AN EIGENVECTOR CORRESPONDING TO IT.
           IF THE J-TH EIGENVALUE IS COMPLEX WITH POSITIVE
           IMAGINARY PART, THEN COLUMNS J AND J+1 OF VEC
           CONTAIN THE REAL AND IMAGINARY PARTS RESPECTIVELY
           OF A COMPLEX EIGENVECTOR CORRESPONDING TO IT.
           IN THIS CASE, OF COURSE, COLUMN J AND THE NEGATIVE
           OF COLUMN J+1 OF VEC FORM AN EIGENVECTOR
           CORRESPONDING TO THE J+1-ST EIGENVALUE.
           THE EIGENVECTORS ARE NOT NORMALIZED IN ANY WAY.

    IERR   IS A STATUS CODE.
         --NORMAL CODE
           0 MEANS THE QR ITERATIONS CONVERGED.
         --ABNORMAL CODES
           J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN
```

```
                100 ITERATIONS.  THE LAST N-J ELEMENTS OF EVR AND EVI
                CONTAIN THOSE EIGENVALUES ALREADY FOUND.
                NO EIGENVECTORS ARE COMPUTED.
             -1 MEANS N, NDIM, OR N*NDIM IS OUT OF RANGE.

      NOTE--    THE ARRAYS A AND VEC MUST BE DISTINCT. A IS DESTROYED.
```

```
RNAN    RNAN    RNAN    RNAN    RNAN       RNAN    RNAN    RNAN    RNAN
              ***************************************
                   ******************************
                        ********************
                             **********
SUBROUTINE RNAN (NDIM,N,A,EVR,EVI,IERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

EISPACK IS AN EXTENSIVE COLLECTION OF ROUTINES FOR SOLVING
THE ALGEBRAIC EIGENVALUE PROBLEM.  THE ORIGINAL ALGOL ROUTINES
WERE WRITTEN BY J. H. WILKINSON, ET.AL., AND SUBSEQUENTLY WERE
TRANSLATED TO FORTRAN AND TESTED AT ARGONNE NATIONAL LABORATORY.
THIS INTERFACE TO EISPACK WAS WRITTEN BY W. R. GAVIN.

ABSTRACT
  RNAN COMPUTES ALL THE EIGENVALUES OF AN ARBITRARY REAL MATRIX.
  THE MATRIX IS BALANCED BY EXACT NORM REDUCING SIMILARITY
  TRANSFORMATIONS AND THEN IS REDUCED TO HESSENBERG FORM BY
  ELEMENTARY SIMILARITY TRANSFORMATIONS.  THE QR ALGORITHM
  IS USED TO COMPUTE THE EIGENSYSTEM OF THE HESSENBERG FORM.

  TO COMPUTE ALL EIGENVALUES AND EIGENVECTORS OF AN ARBITRARY REAL
  MATRIX SEE SUBROUTINE RNAA.  FOR EIGENSYSTEMS OF REAL SYMMETRIC
  MATRICES SEE SUBROUTINES RSAA AND RSAN.  FOR EIGENSYSTEMS OF
  COMPLEX MATRICES SEE CHAA, CHAN, CNAA, AND CNAN.

DESCRIPTION OF ARGUMENTS
 ON INPUT
    NDIM   MUST BE THE ROW DIMENSION OF THE ARRAY A IN THE
           CALLING PROGRAM DIMENSION STATEMENT.

    N      IS THE ORDER OF THE MATRIX.  N MUST NOT EXCEED NDIM.
           N*NDIM MUST NOT EXCEED 102400 = 320*320 =310000(OCTAL).
           N MUST NOT EXCEED 320.  N MAY BE 1.

    A      AN ARRAY WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS.
           THE LEADING N BY N SUBARRAY MUST CONTAIN THE ARBITRARY
           REAL MATRIX WHOSE EIGENVALUES ARE TO BE COMPUTED.

 ON OUTPUT
    EVR    CONTAINS THE REAL PARTS OF THE COMPUTED EIGENVALUES.

    EVI    CONTAINS THE IMAGINARY PARTS OF THE COMPUTED
           EIGENVALUES.  THE EIGENVALUES ARE NOT ORDERED
           IN ANY WAY.  HOWEVER CONJUGATE PAIRS OCCUR
           IN ADJACENT PLACES WITH THE EIGENVALUE OF
           POSITIVE IMAGINARY PART FIRST.

    IERR   IS A STATUS CODE.
         --NORMAL CODE
           0 MEANS THE QR ITERATIONS CONVERGED.
         --ABNORMAL CODES
           J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN
             100 ITERATIONS.  THE LAST N-J ELEMENTS OF EVR AND EVI
             CONTAIN THOSE EIGENVALUES ALREADY FOUND.
          -1 MEANS N, NDIM, OR N*NDIM IS OUT OF RANGE.
```

A IS DESTROYED.

```
RPQR    RPQR    RPQR    RPQR    RPQR       RPQR    RPQR    RPQR    RPQR
              ************************************
                   ******************************
                        *********************
                             **********
      SUBROUTINE  RPQR(NDEG,COEF,WR,WI,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      THIS ROUTINE IS AN INTERFACE TO AN EIGENVALUE ROUTINE IN EISPACK.
      THIS INTERFACE WAS WRITTEN BY WILLIAM R. GAVIN.

      ABSTRACT

          THIS ROUTINE COMPUTES ALL ROOTS OF A POLYNOMIAL
          OF DEGREE TWENTY OR LESS WITH REAL COEFFICIENTS
          BY COMPUTING THE EIGENVALUES OF THE COMPANION MATRIX.

      DESCRIPTION OF PARAMETERS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               COEF(NDEG+1), WR(NDEG), WI(NDEG)

        INPUT -
        NDEG   DEGREE OF POLYNOMIAL

        COEF   ARRAY OF COEFFICIENTS IN ORDER OF DESCENDING POWERS OF Z.
               I.E. COEF(1)*(Z**NDEG) + ... + COEF(NDEG)*Z+COEF(NDEG+1)

        OUTPUT-
        WR,WI  REAL AND IMAGINARY PARTS OF COMPUTED ROOTS

        IERR   OUTPUT ERROR CODE
              - NORMAL CODE
               0  MEANS THE ROOTS WERE COMPUTED.
              - ABNORMAL CODES
               1  MORE THAN 40 QR ITERATIONS ON SOME EIGENVALUE
                  OF THE COMPANION MATRIX
               2  COEF(1) = 0.0
               3  NDEG GREATER THAN 20 OR LESS THAN 1
```

```
RSAA    RSAA    RSAA    RSAA    RSAA       RSAA    RSAA    RSAA    RSAA
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE RSAA(NDIM,N,A,EV,VEC,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      EISPACK IS AN EXTENSIVE COLLECTION OF ROUTINES FOR SOLVING
      THE ALGEBRAIC EIGENVALUE PROBLEM.  THE ORIGINAL ALGOL ROUTINES
      WERE WRITTEN BY J. H. WILKINSON, ET.AL., AND SUBSEQUENTLY WERE
      TRANSLATED TO FORTRAN AND TESTED AT ARGONNE NATIONAL LABORATORY.
      THIS INTERFACE TO EISPACK WAS WRITTEN BY W. R. GAVIN.

      ABSTRACT
        THIS SUBROUTINE COMPUTES ALL THE EIGENVALUES AND AN
        ORTHONORMAL SET OF EIGENVECTORS OF A REAL SYMMETRIC MATRIX.
        THE SYMMETRIC MATRIX IS REDUCED TO TRIDIAGONAL FORM
        BY ORTHOGONAL SIMILARITY TRANSFORMATIONS.  QL TRANSFORMATIONS
        ARE USED TO FIND THE EIGENSYSTEM OF THE TRIDIAGONAL MATRIX.

        TO COMPUTE ONLY THE EIGENVALUES OF A REAL SYMMETRIC
        MATRIX SEE SUBROUTINE RSAN.  FOR EIGENSYSTEMS OF NON-SYMMETRIC
        MATRICES SEE SUBROUTINES RNAA AND RNAN.  FOR EIGENSYSTEMS OF
        COMPLEX MATRICES SEE CHAA, CHAN, CNAA, AND CNAN.

      DESCRIPTION OF ARGUMENTS
       ON INPUT

          NDIM   MUST BE THE ROW DIMENSION OF THE ARRAYS A AND VEC
                 IN THE CALLING PROGRAM DIMENSION STATEMENT.

          N      IS THE ORDER OF THE MATRIX.  N MUST NOT EXCEED NDIM.
                 N*NDIM MUST NOT EXCEED 50625 = 225*225 = 142701(OCTAL).
                 N MUST NOT EXCEED 225.  N MAY BE 1.

          A      AN ARRAY WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS.
                 THE LEADING N BY N SUBARRAY MUST CONTAIN THE REAL
                 SYMMETRIC MATRIX WHOSE EIGENSYSTEM IS TO BE COMPUTED.
                 ONLY THE DIAGONAL AND LOWER TRIANGLE NEED BE DEFINED.

       ON OUTPUT

          EV     CONTAINS THE EIGENVALUES OF A IN ASCENDING ORDER.

          VEC    CONTAINS AN ORTHONORMAL SET OF EIGENVECTORS OF A
                 IN THE COLUMNS OF THE N BY N LEADING SUBARRAY OF VEC.
                 THE J-TH COLUMN OF VEC CONTAINS AN EIGENVECTOR OF
                 LENGTH ONE CORRESPONDING TO THE EIGENVALUE IN
                 THE J-TH ELEMENT OF EV.

          IERR   IS A STATUS CODE.
                --NORMAL CODE
                  0 MEANS THE QL-ITERATIONS CONVERGED.
                --ABNORMAL CODES
                  J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN
                    30 ITERATIONS.  THE FIRST J-1 ELEMENTS OF EV CONTAIN
                    UNORDERED EIGENVALUES.  THE FIRST J-1 COLUMNS OF VEC
                    CONTAIN THE CORRESPONDING EIGENVECTORS.
                 -1 MEANS N, NDIM, OR N*NDIM IS OUT OF RANGE.
      NOTE --    THE ARRAYS A AND VEC MAY COINCIDE. IF A AND VEC
                 ARE DISTINCT A IS UNALTERED.
```

```
RSAN    RSAN    RSAN    RSAN    RSAN       RSAN    RSAN    RSAN    RSAN
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE RSAN(NDIM,N,A,EV,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      EISPACK IS AN EXTENSIVE COLLECTION OF ROUTINES FOR SOLVING
      THE ALGEBRAIC EIGENVALUE PROBLEM.  THE ORIGINAL ALGOL ROUTINES
      WERE WRITTEN BY J. H. WILKINSON, ET.AL., AND SUBSEQUENTLY WERE
      TRANSLATED TO FORTRAN AND TESTED AT ARGONNE NATIONAL LABORATORY.
      THIS INTERFACE TO EISPACK WAS WRITTEN BY W. R. GAVIN.

      ABSTRACT
        RSAN COMPUTES ALL THE EIGENVALUES OF A REAL SYMMETRIC MATRIX.
        THE SYMMETRIC MATRIX IS REDUCED TO TRIDIAGONAL FORM
        BY ORTHOGONAL SIMILARITY TRANSFORMATIONS.  QL TRANSFORMATIONS
        ARE USED TO FIND THE EIGENVALUES OF THE TRIDIAGONAL MATRIX.

        TO COMPUTE ALL EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC
        MATRIX SEE SUBROUTINE RSAA.  FOR EIGENSYSTEMS OF NON-SYMMETRIC
        MATRICES SEE SUBROUTINES RNAA AND RNAN.  FOR EIGENSYSTEMS OF
        COMPLEX MATRICES SEE CHAA, CHAN, CNAA, AND CNAN.

      DESCRIPTION OF ARGUMENTS
       ON INPUT

          NDIM   MUST BE THE ROW DIMENSION OF THE ARRAY A IN THE
                 CALLING PROGRAM DIMENSION STATEMENT.

          N      IS THE ORDER OF THE MATRIX.  N MUST NOT EXCEED NDIM.
                 N*NDIM MUST NOT EXCEED 102400 = 320*320 =310000(OCTAL).
                 N MUST NOT EXCEED 320.  N MAY BE 1.

          A      AN ARRAY WITH EXACTLY NDIM ROWS AND AT LEAST N COLUMNS.
                 THE LEADING N BY N SUBARRAY MUST CONTAIN THE REAL
                 SYMMETRIC MATRIX WHOSE EIGENVALUES ARE TO BE COMPUTED.
                 ONLY THE DIAGONAL AND LOWER TRIANGLE NEED BE DEFINED.

       ON OUTPUT

          EV     CONTAINS THE EIGENVALUES OF A IN ASCENDING ORDER.

          IERR   IS A STATUS CODE.
                --NORMAL CODE
                  0 MEANS THE QL ITERATIONS CONVERGED.
                --ABNORMAL CODES
                  J MEANS THE J-TH EIGENVALUE HAS NOT BEEN FOUND IN
                    30 ITERATIONS.  THE FIRST J-1 ELEMENTS OF EV CONTAIN
                    UNORDERED EIGENVALUES.
                 -1 MEANS N, NDIM, OR N*NDIM IS OUT OF RANGE.

          A      IS UNALTERED IN ITS DIAGONAL AND UPPER TRIANGLE.
                 ITS LOWER TRIANGLE IS DESTROYED.
```

```
SAXB    SAXB    SAXB    SAXB    SAXB       SAXB    SAXB    SAXB    SAXB
              ***************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE SAXB(ND,N,M,A,B,INIT,IN,KER)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      WRITTEN BY CARL B. BAILEY, NOVEMBER 1973.

      ABSTRACT
          SAXB SOLVES A NONSINGULAR SYSTEM OF REAL LINEAR ALGEBRAIC
          EQUATIONS, AX=B.
          THE METHOD USED IS GAUSSIAN ELIMINATION (LU DECOMPOSITION
          FOLLOWED BY FORWARD-BACKWARD SUBSTITUTION) WITH IMPLICIT ROW
          SCALING AND PARTIAL (ROW) PIVOTING.  SAXB IS ESPECIALLY
          EFFICIENT FOR SOLVING A SEQUENCE OF SYSTEMS OF EQUATIONS ALL
          HAVING THE SAME COEFFICIENT MATRIX -A-.  IN SUCH A CASE, THE
          LU DECOMPOSITION IS PERFORMED ONLY ON THE FIRST CALL AND THE
          LU FACTORS ARE STORED IN -A-.  ON SUBSEQUENT CALLS, FORWARD-
          BACKWARD SUBSTITUTION IS PERFORMED IMMEDIATELY ON -B- USING
          THE PREVIOUSLY COMPUTED LU FACTORS.

          SAXB CALLS THE ROUTINE RLUD TO PERFORM LU DECOMPOSITION AND
          RFBS TO PERFORM FORWARD-BACKWARD SUBSTITUTION.
          FOR GREATER ACCURACY AND AN ERROR ESTIMATE USE SAXBI.

      REFERENCE
        1.  G.E.FORSYTHE AND C.B.MOLER, COMPUTER SOLUTION OF LINEAR
            ALGEBRAIC EQUATIONS, PRENTICE-HALL, 1967

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
               A(ND,N), B(ND,M), IN(N)
          IF M=1 THEN THE DIMENSION OF B MAY BE B(N)

        --INPUT---
          ND      - THE ACTUAL FIRST DIMENSION OF ARRAYS -A- AND -B-.
                    (I.E. THE MAXIMUM NUMBER OF EQUATIONS THAT CAN BE
                    SOLVED USING -A- TO STORE THE COEFFICIENTS.)
          N       - THE NUMBER OF EQUATIONS TO BE SOLVED IN THIS CALL.
                    (1 .LE. N .LE. ND)
          M       - NUMBER OF COLUMNS OF -B-.  (NORMALLY M=1)
          A       - THE LEADING -N- BY -N- SUBARRAY OF -A- MUST CONTAIN
                    THE COEFFICIENT MATRIX ON THE INITIAL CALL FOR EACH
                    SEQUENCE OF RELATED SYSTEMS OF EQUATIONS.  (INIT=0)
                    ON ANY SUBSEQUENT CALL FOR A SYSTEM WITH THE SAME
                    COEFFICIENT MATRIX BUT DIFFERENT VALUES OF -B-, -A-
                    MUST CONTAIN THE LU FACTORS THAT WERE RETURNED IN -A-
                    ON THE FIRST CALL.  (INIT≠0)
          B       - THE LEADING -N- BY -M- SUBARRAY OF -B- MUST CONTAIN
                    THE MATRIX (OR VECTOR) OF CONSTANTS.
          INIT    - IS A FLAG WHICH PROVIDES FOR THE ESPECIALLY EFFICIENT
                    SOLUTION OF A SEQUENCE OF SYSTEMS OF EQUATIONS HAVING
                    THE SAME -A- BUT DIFFERENT -B- VECTORS.
                    ON THE INITIAL CALL FOR A SEQUENCE OF RELATED SYSTEMS
                    OF EQUATIONS, INIT MUST BE ZERO AND THE ARRAY -A-
                    MUST CONTAIN THE COEFFICIENT MATRIX -A-.
                    IN ORDER TO SOLVE ANY RELATED SYSTEM EFFICIENTLY
                    ON ANY SUBSEQUENT CALL FOR A SYSTEM WITH THE SAME
                    COEFFICIENT MATRIX BUT DIFFERENT VALUES FOR -B-,
                    INIT MUST BE NONZERO AND -A- MUST CONTAIN THE LU
                    FACTORS THAT WERE RETURNED IN -A- ON THE FIRST CALL.
          IN      - PROVIDES STORAGE FOR THE ROW INTERCHANGE INDICES.
                    ON THE INITIAL CALL FOR A SEQUENCE OF RELATED SYSTEMS
                    OF EQUATIONS, -IN- IS JUST A WORK ARRAY.  ON ANY
                    SUBSEQUENT CALL FOR A RELATED SYSTEM OF EQUATIONS,
                    -IN- MUST CONTAIN THE INDICES THAT WERE RETURNED IN
                    -IN- ON THE FIRST CALL.
```

```
--OUTPUT--
  A       - THE LEADING -N- BY -N- SUBARRAY WILL CONTAIN L-I+U
            WHERE -L- AND -U- ARE TRIANGULAR FACTORS OF -A-,
            -L- IS UNIT LOWER TRIANGULAR, -I- IS THE IDENTITY.
            (ACTUALLY, IT IS NOT L-I+U WHICH IS STORED IN -A- BUT
            LL-I+U WHERE LL IS A REARRANGEMENT OF ELEMENTS OF L.)
  B       - THE LEADING -N- BY -M- SUBARRAY OF -B- WILL CONTAIN
            THE SOLUTION -X-.
  IN      - WILL CONTAIN THE ROW INTERCHANGE INDICES COMPUTED
            DURING LU DECOMPOSITION.  IN(N) WILL CONTAIN
             +1 IF AN EVEN NUMBER OF INTERCHANGES WERE PERFORMED,
             -1 IF AN ODD NUMBER OF INTERCHANGES WERE PERFORMED,
              0 IF THE MATRIX -A- AND THE FACTOR U ARE SINGULAR.
  KER     - AN ERROR CODE
          --NORMAL CODES
            0 MEANS NO ERRORS WERE DETECTED
          --ABNORMAL CODES
            1 MEANS -ND- WAS NOT IN THE RANGE 1 .LT. ND .LE. 325
            2 MEANS -N- WAS NOT IN THE RANGE 1 .LE. N .LE. ND.
            3 MEANS THE TRIANGULAR FACTOR -U- OF -A- IS SINGULAR.

NOTE --- AFTER SOLVING A SYSTEM OF EQUATIONS USING SAXB
         ONE CAN EASILY COMPUTE THE DETERMINANT OF -A-,
         AT LEAST IN PRINCIPAL.  FOR EXAMPLE,
              DET = IN(N)
              DO 1 I =1,N
            1 DET = DET*A(I,I)
         HOWEVER, THAT COMPUTATION MAY OFTEN RESULT IN EXPONENTIAL
         OVERFLOW OR UNDERFLOW, ESPECIALLY IF THE COEFFICIENTS
         IN -A- WERE VERY LARGE OR VERY SMALL.
```

```
SAXBI    SAXBI    SAXBI    SAXBI       SAXBI    SAXBI    SAXBI    SAXBI
            ***************************************
                 ******************************
                      ********************
                           **********
SUBROUTINE SAXBI(ND,N,M,A,B,X,INIT,RC,W,IN,KER)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

WRITTEN BY CARL B. BAILEY, NOVEMBER 1973.

ABSTRACT
    SAXBI SOLVES A NONSINGULAR SYSTEM OF REAL LINEAR ALGEBRAIC
    EQUATIONS, AX=B, AND COMPUTES AN ERROR BOUND FOR THE SOLUTION.
    THE METHOD USED IS GAUSSIAN ELIMINATION (LU DECOMPOSITION
    FOLLOWED BY FORWARD-BACKWARD SUBSTITUTION) WITH IMPLICIT ROW
    SCALING, PARTIAL (ROW) PIVOTING, AND ITERATIVE CORRECTIONS.
    SAXBI IS ESPECIALLY EFFICIENT FOR SOLVING A SEQUENCE OF
    SYSTEMS OF EQUATIONS HAVING THE SAME COEFFICIENT MATRIX -A-.
    IN SUCH A CASE, THE LU DECOMPOSITION IS PERFORMED ONLY ON THE
    FIRST CALL AND THE LU FACTORS ARE STORED IN -W-. ON SUBSEQUENT
    CALLS, FORWARD-BACKWARD SUBSTITUTION IS PERFORMED IMMEDIATELY
    ON -B- USING THE PREVIOUSLY COMPUTED LU FACTORS.

    SAXBI CALLS THE ROUTINE RLUD TO PERFORM LU DECOMPOSITION, RFBS
    TO PERFORM FORWARD-BACKWARD SUBSTITUTION, AND RIMP TO PERFORM
    THE ITERATIVE CORRECTIONS.
    FOR FASTER EXECUTION WITHOUT AN ERROR ESTIMATE USE SAXB.

REFERENCE
  1.  G.E.FORSYTHE AND C.B.MOLER, COMPUTER SOLUTION OF LINEAR
      ALGEBRAIC EQUATIONS, PRENTICE-HALL, 1967
```

```
DESCRIPTION OF ARGUMENTS
    THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST
         A(ND,N), B(ND,M), X(ND,M), W(ND,N+1), IN(N)
    IF M=1 THEN THE DIMENSION OF B AND X MAY BE B(N), X(N).

  --INPUT---
    ND      - THE ACTUAL FIRST DIMENSION OF -A-, -B-, -X-, AND -W-.
              (I.E. THE MAXIMUM NUMBER OF EQUATIONS THAT CAN BE
              SOLVED USING -A- TO STORE THE COEFFICIENTS.)
    N       - THE NUMBER OF EQUATIONS TO BE SOLVED IN THIS CALL.
              (1 .LE. N .LE. ND)
    M       - NUMBER OF COLUMNS OF -B- AND -X-.  (NORMALLY M=1)
    A       - THE LEADING -N- BY -N- SUBARRAY OF -A- MUST CONTAIN
              THE COEFFICIENT MATRIX -A-.  (FOR ANY VALUE OF INIT)
    B       - THE LEADING -N- BY -M- SUBARRAY OF -B- MUST CONTAIN
              THE MATRIX (OR VECTOR) OF CONSTANTS.
    INIT    - IS A FLAG WHICH PROVIDES FOR THE ESPECIALLY EFFICIENT
              SOLUTION OF A SEQUENCE OF SYSTEMS OF EQUATIONS HAVING
              THE SAME -A- BUT DIFFERENT -B- VECTORS.
              ON THE INITIAL CALL FOR A SEQUENCE OF RELATED SYSTEMS
              OF EQUATIONS, INIT MUST BE ZERO.
              IN ORDER TO SOLVE ANY RELATED SYSTEM EFFICIENTLY
              ON ANY SUBSEQUENT CALL FOR A SYSTEM WITH THE SAME
              COEFFICIENT MATRIX BUT DIFFERENT VALUES FOR -B-,
              INIT MUST BE NONZERO AND -W- MUST CONTAIN THE LU
              FACTORS THAT WERE RETURNED IN -W- ON THE FIRST CALL
              AND -IN- MUST CONTAIN THE ROW INTERCHANGE INDICES
              THAT WERE RETURNED IN -IN- ON THE FIRST CALL.
    W       - PROVIDES STORAGE FOR THE LU FACTORS OF -A-.
              IF INIT IS ZERO, -W- IS JUST A WORK ARRAY.  IF INIT
              IS NONZERO, -W- MUST CONTAIN THE LU FACTORS THAT WERE
              COMPUTED IN THE INITIAL CALL FOR THE MATRIX -A-.
    IN      - PROVIDES STORAGE FOR THE ROW INTERCHANGE INDICES.
              ON THE INITIAL CALL FOR A SEQUENCE OF RELATED SYSTEMS
              OF EQUATIONS, -IN- IS JUST A WORK ARRAY.  ON ANY
              SUBSEQUENT CALL FOR A RELATED SYSTEM OF EQUATIONS,
              -IN- MUST CONTAIN THE INDICES THAT WERE RETURNED IN
              -IN- ON THE FIRST CALL.

  --OUTPUT--
    X       - THE LEADING -N- BY -M- SUBARRAY OF -X- WILL CONTAIN
              THE SOLUTION.
    RC      - WILL BE THE RATIO OF THE MAXIMUM NORM OF THE FIRST
              CORRECTION TO THE MAXIMUM NORM OF THE INITIAL
              APPROXIMATE SOLUTION.  THE CONDITION NUMBER OF -A-
              AND ERROR BOUNDS FOR THE COMPUTED SOLUTION ARE
              RELATED TO -RC-.  A SMALL VALUE FOR -RC- INDICATES
              A WELL-CONDITIONED SYSTEM AND SMALL UNCERTAINTIES
              IN THE SOLUTION.  A LARGE VALUE FOR -RC- INDICATES
              AN ILL-CONDITIONED SYSTEM AND LARGE UNCERTAINTIES
              IN THE SOLUTION.
    W       - THE LEADING -N- BY -N- SUBARRAY WILL CONTAIN L-I+U
              WHERE -L- AND -U- ARE TRIANGULAR FACTORS OF -A-,
              -L- IS UNIT LOWER TRIANGULAR, AND -I- IS IDENTITY.
              (ACTUALLY, IT IS NOT L-I+U WHICH IS STORED IN -A- BUT
              LL-I+U WHERE LL IS A REARRANGEMENT OF ELEMENTS OF L.)
              THE N+1ST COLUMN CONTAINS THE LAST CORRECTION TO -X-.
              IF INIT .EQ. 0, LU FACTORS OF -A- WILL BE COMPUTED
              AND STORED IN -W-.
    KER     - AN ERROR CODE
            --NORMAL CODES
              0 MEANS NO ERRORS WERE DETECTED
            --ABNORMAL CODES
              1 MEANS -ND- WAS NOT IN THE RANGE 1 .LT. ND .LE. 225
              2 MEANS -N- WAS NOT IN THE RANGE 1 .LE. N .LE. ND.
              3 MEANS THE TRIANGULAR FACTOR -U- OF -A- IS SINGULAR.
              4 MEANS -A- IS TOO ILL-CONDITIONED FOR ITERATIVE
                IMPROVEMENT TO BE EFFECTIVE.

NOTE --- AFTER SOLVING A SYSTEM OF EQUATIONS USING SAXBI
         ONE CAN EASILY COMPUTE THE DETERMINANT OF -A-,
         AT LEAST IN PRINCIPAL.  FOR EXAMPLE,
              DET = IN(N)
```

```
      DO 1 I =1,N
    1 DET = DET*W(I,I)
```

HOWEVER, THAT COMPUTATION MAY OFTEN RESULT IN EXPONENTIAL OVERFLOW OR UNDERFLOW, ESPECIALLY IF THE COEFFICIENTS IN -A- WERE VERY LARGE OR VERY SMALL.

```
SICONT    SICONT    SICONT    SICONT        SICONT    SICONT    SICONT
            ****************************************
                 ******************************
                      ********************
                           **********
      SUBROUTINE SICONT (L,U,NN,W,F1,F1CO,F1SI,FUNCT,IERR)
```

SANDIA MATHEMATICAL PROGRAM LIBRARY

CONSULTANTS AT SLL INCLUDE -
R. E. HUDDLESTON - DIVISION 8322
T. H. JEFFERSON - DIVISION 8322

SICONT WAS ORIGINALLY PROGRAMMED BY A R IACOLETTI IN MARCH, 1966. R E JONES MODIFIED SICONT SOMEWHAT AND PREPARED IT FOR INCLUSION IN THE MATHEMATICAL LIBRARY IN MAY 1968.

ABSTRACT

SICONT CALCULATES THE INTEGRAL OVER (L,U) OF F(X)*COS(W*X) AND F(X)*SIN(W*X) USING TABULATED DATA OR A FUNCTION SUBPROGRAM FOR EVALUATION OF THE FUNCTION F.
(L IS FLOATING POINT.)

DISCUSSION ON CALLING SEQUENCE
SICONT HAS TWO MODES OF OPERATION, AS FOLLOWS ---

IF NN IS A POSITIVE, EVEN INTEGER, THEN SICONT EXPECTS F1 TO BE AN ARRAY OF NN+1 EQUALLY SPACED VALUES OF THE FUNCTION F1. THAT IS, F1 MUST CONTAIN F1(X) FOR X = L, L+(U-L)/NN, L+2*(U-L)/NN, ... , L+(NN-1)*(U-L)/NN, U. SICONT USES THESE VALUES TO APPROXIMATE THE INTEGRALS OF F1(X)*COS(W*X) AND F1(X)*SIN(W*X) OVER THE INTERVAL (L,U). THE ANSWERS APPEAR IN F1CO AND F1SI RESPECTIVELY. FUNCT IS A DUMMY PARAMETER IN THIS CASE.

IF NN IS A NEGATIVE, EVEN INTEGER, THEN FUNCT MUST BE THE NAME OF AN EXTERNAL FUNCTION SUBPROGRAM. (THIS NAME MUST APPEAR IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM). SICONT APPROXIMATES THE INTEGRALS OF FUNCT(X)*COS(W*X) AND FUNCT(X)*SIN(W*X) OVER THE INTERVAL (L,U) BY EVALUATING FUNCT AT THE POINTS LISTED IN THE ABOVE PARAGRAPH, STORING THESE VALUES IN THE ARRAY F1, AND PROCEEDING AS IN THE PREVIOUS CASE. NOTE THAT F1 MUST BE DIMENSIONED AT LEAST NN+1 IN THIS CASE ALSO.

IERR WILL NORMALLY BE RETURNED EQUAL TO 1.
IF NN IS NOT EVEN, IERR WILL BE SET EQUAL TO 2.

METHOD USED IN THE INTEGRATION ---
THE FUNCTIONAL VALUES IN F1 ARE FITTED BY SUCCESSIVE MOVING ARC PARABOLAS, AND THE RESULTING POLYNOMIAL*SINUSOIDAL EXPRESSIONS ARE EVALUATED IN CLOSE FORM. BY COMPUTING BASIC COEFFICIENTS DURING THE INITIAL PHASE THESE CALCULATIONS REDUCE TO SUMMATIONS WITH CONCURRENT EVALUATION OF TRIGONOMETRIC FACTORS BY RECURRENCE RELATIONS. DOUBLE PRECISION ARITHMETIC IS USED FOR CERTAIN CALCULATIONS TO PREVENT LOSS OF SIGNIFICANCE.

```
SIMIN    SIMIN    SIMIN    SIMIN      SIMIN    SIMIN    SIMIN    SIMIN
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE SIMIN (F,K,EPS,ANS,S,NEV,ICONT,Y)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      ORIGINAL ROUTINE BY L F SHAMPINE, AS DESCRIBED IN REF.1 BELOW.
      PREPARATION FOR MATH LIBRARY BY R E JONES.

      ABSTRACT
         SIMIN FINDS AN APPROXIMATE MINIMUM OF A REAL FUNCTION OF K
         VARIABLES, GIVEN AN INITIAL ESTIMATE OF THE POSITION OF THE
         MINIMUM. THE SIMPLEX METHOD IS USED. SEE REFERENCE 1 BELOW
         FOR A FULL EXPLANATION OF THIS METHOD.  BRIEFLY, A SET OF
         K+1 POINTS IN K-DIMENSIONAL SPACE IS CALLED A SIMPLEX.
         THE MINIMIZATION PROCESS ITERATES BY REPLACING THE POINT
         WITH THE LARGEST FUNCTION VALUE BY A NEW POINT WITH A
         SMALLER FUNCTION VALUE.  ITERATION CONTINUES UNTIL ALL THE
         POINTS CLUSTER SUFFICIENTLY CLOSE TO A MINIMUM.

      REFERENCES
       1. L F SHAMPINE, A ROUTINE FOR UNCONSTRAINED OPTIMIZATION,
          SC-TM-72130  OR  SC-RR-720657
       2. J A NELDER AND R MEAD, A SIMPLEX METHOD FOR FUNCTION
          MINIMIZATION, COMPUTER JOURNAL, 7(1965) 308-313

      DESCRIPTION OF PARAMETERS
       --INPUT--
         F   - NAME OF FUNCTION OF K VARIABLES TO BE MINIMIZED.
               (THIS NAME MUST APPEAR IN AN EXTERNAL STATEMENT.)
               FORM OF THE CALLING SEQUENCE MUST BE FUNCTION F(X),
               WHERE X IS AN ARRAY OF K VARIABLES.
         K   - THE NUMBER OF VARIABLES.  K MUST BE AT LEAST 2.
               NORMALLY K SHOULD BE LESS THAN ABOUT 10, AS SIMIN
               BECOMES LESS EFFECTIVE FOR LARGER VALUES OF K.
         EPS- THE CONVERGENCE CRITERION.  LET YAVG BE THE AVERAGE
               VALUE OF THE FUNCTION F AT THE K+1 POINTS OF THE
               SIMPLEX, AND LET R BE THEIR STANDARD ERROR.  (THAT IS,
               THE ROOT-MEAN-SQUARE OF THE SET OF VALUES (Y(I)-YAVG),
               WHERE Y(I) IS THE FUNCTION VALUE AT THE I-TH POINT OF
               THE SIMPLEX.)  THEN--
               IF EPS.GT.0, CONVERGENCE IS OBTAINED IF  R.LE.EPS.
               IF EPS.LT.0, CONVERGENCE IS IF  R.LE.ABS(EPS*YAVG).
               IF EPS=0, THE PROCESS WILL NOT CONVERGE BUT INSTEAD WILL
               QUIT WHEN NEV FUNCTION EVALUATIONS HAVE BEEN USED.
         ANS- AN ARRAY OF LENGTH K CONTAINING A GUESS FOR THE LOCATION
               OF A MINIMUM OF F.
         S   - A SCALE PARAMETER, WHICH MAY BE A SIMPLE VARIABLE OR AN
               ARRAY OF LENGTH K.  USE OF AN ARRAY IS SIGNALLED BY
               SETTING S(1) NEGATIVE.
               -SIMPLE VARIABLE CASE.  HERE S IS THE LENGTH OF EACH
               SIDE OF THE INITIAL SIMPLEX.  THUS, THE INITIAL SEARCH
               RANGE IS THE SAME FOR ALL THE VARIABLES.
               -ARRAY CASE.  HERE THE LENGTH OF SIDE I OF THE INITIAL
               SIMPLEX IS ABS(S(I)).  THUS, THE INITIAL SEARCH RANGE
               MAY BE DIFFERENT FOR DIFFERENT VARIABLES.
               NOTE-- THE VALUE(S) USED FOR S ARE NOT VERY CRITICAL.
               ANY REASONABLE GUESS SHOULD DO O.K.
         NEV- THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS TO BE USED.
               (THE ACTUAL NUMBER USED MAY EXCEED THIS SLIGHTLY SO THE
               LAST SEARCH ITERATION MAY BE COMPLETED.)
         ICONT - ICONT SHOULD BE ZERO ON ANY CALL TO SIMIN WHICH
               IS NOT A CONTINUATION OF A PREVIOUS CALL.
               IF ICONT=1 THE PROBLEM WILL BE CONTINUED.  IN THIS
               CASE THE WORK ARRAY Y MUST BE THE SAME ARRAY THAT WAS
               USED IN THE CALL THAT IS BEING CONTINUED (AND THE VALUES
               IN IT MUST BE UNCHANGED).  THE REASON FOR THIS IS THAT
```

```
          IF ICONT=1 THEN THE ARGUMENT S IS IGNORED AND THE SIMPLEX
          AND RELATED FUNCTION VALUES THAT WERE STORED IN ARRAY Y
          DURING A PREVIOUS EXECUTION ARE USED TO CONTINUE THAT
          PREVIOUS PROBLEM.
     Y  - A WORK ARRAY CONTAINING AT LEAST K*K + 5*K + 1 WORDS.
          IF ICONT=1 THIS MUST BE THE SAME ARRAY USED IN THE CALL
          THAT IS BEING CONTINUED.
  --OUTPUT--
     ANS- ANS WILL CONTAIN THE LOCATION OF THE POINT WITH THE
          SMALLEST VALUE OF THE FUNCTION THAT WAS FOUND.
     S  - IN THE SIMPLE VARIABLE CASE S WILL BE RETURNED AS THE
          AVERAGE DISTANCE FROM THE VERTICES TO THE CENTROID OF
          THE SIMPLEX.
          IN THE ARRAY CASE S(I) WILL BE RETURNED AS THE AVERAGE
          DISTANCE IN THE I-TH DIMENSION OF VERTICES FROM
          THE CENTROID.  (S(1) WILL BE NEGATED.)
          NOTE-- THE VALUE(S) RETURNED IN S ARE USEFUL FOR
          ASSESSING THE FLATNESS OF THE FUNCTION NEAR THE
          MINIMUM.  THE LARGER THE VALUE OF S (FOR A GIVEN
          VALUE OF EPS), THE FLATTER THE FUNCTION.
     NEV- NEV WILL BE THE COUNT OF THE ACTUAL NUMBER OF FUNCTION
          EVALUATIONS USED.
     Y  - WILL CONTAIN ALL DATA NEEDED TO CONTINUE THE MINIMIZATION
          SEARCH EFFICIENTLY IN A SUBSEQUENT CALL.
          NOTE -- THE FIRST K+1 ELEMENTS OF Y WILL CONTAIN THE
          FUNCTION VALUES AT THE K+1 POINTS OF THE LATEST SIMPLEX.
          THE NEXT K*(K+1) ELEMENTS OF Y WILL BE THE K+1 POINTS
          OF THE SIMPLEX (IN EXACT CORRESPONDENSE TO THE ARRAY
          P DISCUSSED IN REFERENCE 1 ABOVE).  THE REMAINING 3*K
          WORDS ARE TEMPORARY WORKING STORAGE ONLY.
```

```
SINH    SINH    SINH    SINH    SINH       SINH    SINH    SINH    SINH
              ****************************************
                   ******************************
                        ********************
                             **********
     FUNCTION SINH(X)

     SANDIA MATHEMATICAL PROGRAM LIBRARY

         CONSULTANTS AT SLL INCLUDE -
             R. E. HUDDLESTON  -   DIVISION 8322
             T. H. JEFFERSON   -   DIVISION 8322

     WRITTEN BY CARL B. BAILEY, NOVEMBER 1971

     ABSTRACT

         SINH EVALUATES THE HYPERBOLIC SINE FUNCTION.  THAT IS,
              SINH(X) = (EXP(X) - EXP(-X)) / 2

         FOR ABS(X) .LE. 0.5 AN ECONOMIZED POLYNOMIAL IS USED WHICH
         YIELDS AN ERROR OF NO MORE THAN ONE BIT IN OBSERVED TESTS.
         FOR ABS(X) .GT. 0.5 THE DEFINITION IN TERMS OF EXPONENTIALS
         IS USED WHICH YIELDS AN ACCURACY COMPARABLE TO THE ACCURACY
         OF THE EXPONENTIAL ROUTINE.

     DESCRIPTION OF ARGUMENT

         X - ANY REAL VALUE FOR WHICH EXP(ABS(X)) IS REPRESENTABLE.
```

```
SMOO    SMOO    SMOO    SMOO    SMOO       SMOO    SMOO    SMOO    SMOO
              ************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE SMOO(N,X,Y,DY,S,A,B,C,D,R,R1,R2,T,T1,U,V,IERR)

      SANDIA MATHEMATICAL PROGRAM LIERARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      CONVERSION FROM THE ALGOL BY RONDALL E JONES
      REFERENCE -- NUMERISHE MATHEMATIK 10,177-183 (1967) C H REINSCH

      ABSTRACT
         SMOO FITS A SMOOTH SPLINE THROUGH A GIVEN SET OF DATA POINTS
         BY MINIMIZING THE INTEGRAL OF THE SECOND DERIVATIVE SQUARED,
         SUBJECT TO THE CONSTRAINT THAT
                 N
                SUM ( (R(I)-Y(I))/DY(I) )**2  .LE. S
                I=1
         (WHERE R(I) IS THE ORDINATE OF THE SMOOTH SPLINE AT X(I).)
         SMOO RETURNS THE VALUES OF THE SPLINE FUNCTION, R,
         ITS FIRST DERIVATIVE, R1, AND ITS SECOND DERIVATIVE, R2,
         EVALUATED AT THE ABSCISSAS OF THE GIVEN DATA POINTS.
         THE RESULTING SPLINE, DEFINED BY THE ARRAYS X, R, AND R2,
         MAY THEN BE INTERPOLATED (IF DESIRED) USING SPLINT.
         FOR AN EXACT SPLINE FIT SEE SUBROUTINE SPLIFT.

      DESCRIPTION OF ARGUMENTS
         INPUT ARGUMENTS --
         N  - NUMBER OF DATA VALUES (AT LEAST 3)
         X  - ABSCISSA ARRAY (INCREASING ORDER)
         Y  - ORDINATE ARRAY
         DY - ARRAY OF ERROR ESTIMATES.  DY(I) SHOULD BE AN ESTIMATE
              OF THE ERROR (ACTUALLY, THE STANDARD DEVIATION) IN Y(I).
              THUS, THE UNITS OF DY ARE THE SAME AS THE UNITS OF Y.
              LARGER VALUES OF DY(I) ALLOW A LOOSER, SMOOTHER FIT.
              SMALLER VALUES OF DY(I) CAUSE A TIGHTER FIT.  SETTING
              DY(I)=0 AT ALL POINTS RESULTS IN AN EXACT FIT.(SEE SPLIFT)
              BY APPROPRIATELY ADJUSTING DY(I) AT EACH POINT, THE SPLINE
              CAN BE MADE TIGHT AT CRITICAL POINTS AND LOOSE AT OTHERS.
         S  - SHOULD NORMALLY = N.  (NOTE-- S IS FLOATING POINT - DONT
              USE N DIRECTLY FOR S.)  IF YOU WISH TO TIGHTEN OR LOOSEN
              THE SPLINE FIT BY MULTIPLYING EACH ELEMENT OF DY BY
              SOME FACTOR F, YOU MAY ALTERNATIVELY SIMPLY MULTIPLY
              S BY F**2.
         OUTPUT ARGUMENTS --
         A,B,C,D - CUBIC BETWEEN X(I) AND X(I+1) IS
              A(I) + B(I)*H + C(I)*H**2 + D(I)*H**3
              WHERE H IS DESIRED ABSCISSA MINUS X(I).
         R  - ARRAY OF SMOOTH SPLINE VALUES
         R1 - ARRAY OF SMOOTH SPLINE DERIVATIVES
         R2 - ARRAY OF SMOOTH SPLINE SECOND DERIVATIVES
         T,T1,U,V - WORK ARRAYS
         IERR- A STATUS CODE
              --NORMAL CODE
                =1 MEANS THE REQUESTED SPLINE WAS COMPUTED.
              --ABNORMAL CODE
                =2 MEANS EITHER N IS LESS THAN 3, OR S IS NEGATIVE,
                   OR THE X-AXIS VALUES ARE MISORDERED.

         X,Y,DY,A,B,C,D MUST BE DIMENSIONED AT LEAST N
         R,R1,R2,T,T1,U,V MUST BE DIMENSION AT LEAST N+2

      THE ORIGINAL N1 WAS FIXED AT 1 TO AVOID WASTED WORK ARRAY SPACE
      THE ORIGINAL N2 IS CALLED N HERE
      ALL WORK ARRAY INDICES ARE 1 LARGER THAN IN THE ALGOL,
      TO AVOID A ZERO SUBSCRIPT.
```

```
SPLIFT    SPLIFT    SPLIFT    SPLIFT         SPLIFT    SPLIFT    SPLIFT
              ************************************
                   ****************************
                        ********************
                             **********
    SUBROUTINE SPLIFT (X,Y,YP,YPP,N,W,IERR,ISX,A1,B1,AN,BN)

    SANDIA MATHEMATICAL PROGRAM LIBRARY

        CONSULTANTS AT SLL INCLUDE -
            R. E. HUDDLESTON  -   DIVISION 8322
            T. H. JEFFERSON   -   DIVISION 8322

    WRITTEN BY RONDALL E. JONES

    ABSTRACT
        SPLIFT FITS AN INTERPOLATING CUBIC SPLINE TO THE N DATA POINTS
        GIVEN IN X AND Y AND RETURNS THE FIRST AND SECOND DERIVATIVES
        IN YP AND YPP.  THE RESULTING SPLINE (DEFINED BY X, Y, AND
        YPP) AND ITS FIRST AND SECOND DERIVATIVES MAY THEN BE
        EVALUATED USING SPLINT.  THE SPLINE MAY BE INTEGRATED USING
        SPLIQ.  FOR A SMOOTHING SPLINE FIT SEE SUBROUTINE SMOOTH.

    DESCRIPTION OF ARGUMENTS
        THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST,
        E.G.   X(N), Y(N), YP(N), YPP(N), W(3N)

      --INPUT--

        X     - ARRAY OF ABSCISSAS OF DATA (IN INCREASING ORDER)
        Y     - ARRAY OF ORDINATES OF DATA
        N     - THE NUMBER OF DATA POINTS.  THE ARRAYS X, Y, YP, AND
                YPP MUST BE DIMENSIONED AT LEAST N.  (N .GE. 4)
        ISX   - MUST BE ZERO ON THE INITIAL CALL TO SPLIFT.
                IF A SPLINE IS TO BE FITTED TO A SECOND SET OF DATA
                THAT HAS THE SAME SET OF ABSCISSAS AS A PREVIOUS SET,
                AND IF THE CONTENTS OF W HAVE NOT BEEN CHANGED SINCE
                THAT PREVIOUS FIT WAS COMPUTED, THEN ISX MAY BE
                SET TO ONE FOR FASTER EXECUTION.
        A1,B1,AN,BN - SPECIFY THE END CONDITIONS FOR THE SPLINE WHICH
                ARE EXPRESSED AS CONSTRAINTS ON THE SECOND DERIVATIVE
                OF THE SPLINE AT THE END POINTS (SEE YPP).
                THE END CONDITION CONSTRAINTS ARE
                        YPP(1) = A1*YPP(2) + B1
                AND
                        YPP(N) = AN*YPP(N-1) + BN
                WHERE
                        ABS(A1).LT. 1.0  AND  ABS(AN).LT. 1.0.

                THE SMOOTHEST SPLINE (I.E., LEAST INTEGRAL OF SQUARE
                OF SECOND DERIVATIVE) IS OBTAINED BY A1=B1=AN=BN=0.
                IN THIS CASE THERE IS AN INFLECTION AT X(1) AND X(N).
                IF THE DATA IS TO BE EXTRAPOLATED (SAY, BY USING SPLINT
                TO EVALUATE THE SPLINE OUTSIDE THE RANGE X(1) TO X(N)),
                THEN TAKING A1=AN=0.5 AND B1=BN=0 MAY YIELD BETTER
                RESULTS.  IN THIS CASE THERE IS AN INFLECTION
                AT X(1) - (X(2)-X(1)) AND AT X(N) + (X(N)-X(N-1)).
                IN THE MORE GENERAL CASE OF A1=AN=A  AND B1=BN=0,
                THERE IS AN INFLECTION AT X(1) - (X(2)-X(1))*A/(1.0-A)
                AND AT X(N) + (X(N)-X(N-1))*A/(1.0-A).

                A SPLINE THAT HAS A GIVEN FIRST DERIVATIVE YP1 AT X(1)
                AND YPN AT Y(N) MAY BE DEFINED BY USING THE
                FOLLOWING CONDITIONS.

                A1=-0.5

                B1= 3.0*((Y(2)-Y(1))/(X(2)-X(1))-YP1)/(X(2)-X(1))

                AN=-0.5

                BN=-3.0*((Y(N)-Y(N-1))/(X(N)-X(N-1))-YPN)/(X(N)-X(N-1))

      --OUTPUT--
```

```
YP   - ARRAY OF FIRST DERIVATIVES OF SPLINE (AT THE X(I))
YPP  - ARRAY OF SECOND DERIVATIVES OF SPLINE (AT THE X(I))
IERR - A STATUS CODE
     --NORMAL CODE
        1 MEANS THAT THE REQUESTED SPLINE WAS COMPUTED.
     --ABNORMAL CODES
        2 MEANS THAT N, THE NUMBER OF POINTS, WAS .LT. 4.
        3 MEANS THE ABSCISSAS WERE NOT STRICTLY INCREASING.

--WORK--

W    - ARRAY OF WORKING STORAGE DIMENSIONED AT LEAST 3N.
```

```
SPLINT    SPLINT    SPLINT    SPLINT         SPLINT    SPLINT    SPLINT
               ****************************************
                    ******************************
                         ********************
                              **********
SUBROUTINE SPLINT (X,Y,YPP,N,XI,YI,YPI,YPPI,NI,KERR)

SANDIA MATHEMATICAL PROGRAM LIBRARY

    CONSULTANTS AT SLL INCLUDE -
        R. E. HUDDLESTON  -   DIVISION 8322
        T. H. JEFFERSON   -   DIVISION 8322

WRITTEN BY RONDALL E. JONES

ABSTRACT

    SPLINT EVALUATES A CUBIC SPLINE AND ITS FIRST AND SECOND
    DERIVATIVES AT THE ABSCISSAS IN XI.  THE SPLINE (WHICH
    IS DEFINED BY X, Y, AND YPP) MAY HAVE BEEN DETERMINED BY
    SPLIFT OR SMOOTH OR ANY OTHER SPLINE FITTING ROUTINE THAT
    PROVIDES SECOND DERIVATIVES.

DESCRIPTION OF ARGUMENTS
    THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST,
    E.G.  X(N), Y(N), YPP(N), XI(NI), YI(NI), YPI(NI), YPPI(NI)

  --INPUT--

    X   - ARRAY OF ABSCISSAS (IN INCREASING ORDER) THAT DEFINE THE
          SPLINE.  USUALLY X IS THE SAME AS X IN SPLIFT OR SMOOTH.
    Y   - ARRAY OF ORDINATES THAT DEFINE THE SPLINE.  USUALLY Y IS
          THE SAME AS Y IN SPLIFT OR AS R IN SMOOTH.
    YPP - ARRAY OF SECOND DERIVATIVES THAT DEFINE THE SPLINE.
          USUALLY YPP IS THE SAME AS YPP IN SPLIFT OR R2 IN SMOOTH
    N   - THE NUMBER OF DATA POINTS THAT DEFINE THE SPLINE.
          THE ARRAYS X, Y, AND YPP MUST BE DIMENSIONED AT LEAST N.
          N MUST BE GREATER THAN OR EQUAL TO 2.
    XI  - THE ABSCISSA OR ARRAY OF ABSCISSAS (IN ARBITRARY ORDER)
          AT WHICH THE SPLINE IS TO BE EVALUATED.
          EACH XI(K) THAT LIES BETWEEN X(1) AND X(N) IS A CASE OF
          INTERPOLATION.  EACH XI(K) THAT DOES NOT LIE BETWEEN
          X(1) AND X(N) IS A CASE OF EXTRAPOLATION.  BOTH CASES
          ARE ALLOWED.  SEE DESCRIPTION OF KERR.
    NI  - THE NUMBER OF ABSCISSAS AT WHICH THE SPLINE IS TO BE
          EVALUATED.  IF NI IS GREATER THAN 1, THEN XI, YI, YPI,
          AND YPPI MUST BE ARRAYS DIMENSIONED AT LEAST NI.
          NI MUST BE GREATER THAN OR EQUAL TO 1.

 --OUTPUT--

    YI  - ARRAY OF VALUES OF THE SPLINE (ORDINATES) AT XI.
    YPI - ARRAY OF VALUES OF THE FIRST DERIVATIVE OF SPLINE AT XI.
    YPPI- ARRAY OF VALUES OF SECOND DERIVATIVES OF SPLINE AT XI.
```

```
KERR- A STATUS CODE
    --NORMAL CODES
       1 MEANS THAT THE SPLINE WAS EVALUATED AT EACH ABSCISSA
         IN XI USING ONLY INTERPOLATION.
       2 MEANS THAT THE SPLINE WAS EVALUATED AT EACH ABSCISSA
         IN XI, BUT AT LEAST ONE EXTRAPOLATION WAS PERFORMED.
    -- ABNORMAL CODE
       3 MEANS THAT THE REQUESTED NUMBER OF EVALUATIONS, NI,
         WAS NOT POSITIVE.
```

```
SPLIQ    SPLIQ    SPLIQ    SPLIQ       SPLIQ    SPLIQ    SPLIQ    SPLIQ
              ****************************************
                   ******************************
                        ********************
                             **********
      SUBROUTINE SPLIQ(X,Y,YP,YPP,N,XLO,XUP,NUP,ANS,IERR)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      THIS ROUTINE WAS WRITTEN BY M. K. GORDON

      ABSTRACT

      SUBROUTINE SPLIQ INTEGRATES A CUBIC SPLINE (GENERATED BY
      SPLIFT,SMOOTH,ETC.) ON THE INTERVALS (XLO,XUP(I)), WHERE XUP
      IS A SEQUENCE OF UPPER LIMITS ON THE INTERVALS OF INTEGRATION.
      THE ONLY RESTRICTIONS ON XLO AND XUP(*) ARE
                 XLO .LT. XUP(1),
                 XUP(I) .LE. XUP(I+1)   FOR EACH I .
      ENDPOINTS BEYOND THE SPAN OF ABSCISSAS ARE ALLOWED.
      THE SPLINE OVER THE INTERVAL (X(I),X(I+1)) IS REGARDED
      AS A CUBIC POLYNOMIAL EXPANDED ABOUT X(I) AND IS INTEGRATED
      ANALYTICALLY.

      DESCRIPTION OF ARGUMENTS
          THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST,
          E.G.  X(N), Y(N), YP(N), YPP(N), XUP(NUP), ANS(NUP)

       --INPUT--

         X    - ARRAY OF ABSCISSAS (IN INCREASING ORDER) THAT DEFINE THE
                SPLINE.  USUALLY X IS THE SAME AS X IN SPLIFT OR SMOOTH.
         Y    - ARRAY OF ORDINATES THAT DEFINE THE SPLINE.  USUALLY Y IS
                THE SAME AS Y IN SPLIFT OR AS R IN SMOOTH.
         YP   - ARRAY OF FIRST DERIVATIVES OF THE SPLINE AT ABSCISSAS.
                USUALLY YP IS THE SAME AS YP IN SPLIFT OR R1 IN SMOOTH.
         YPP  - ARRAY OF SECOND DERIVATIVES THAT DEFINE THE SPLINE.
                USUALLY YPP IS THE SAME AS YPP IN SPLIFT OR R2 IN SMOOTH
         N    - THE NUMBER OF DATA POINTS THAT DEFINE THE SPLINE.
         XLO  - LEFT ENDPOINT OF INTEGRATION INTERVALS.
         XUP  - RIGHT ENDPOINT OR ARRAY OF RIGHT ENDPOINTS OF
                INTEGRATION INTERVALS IN ASCENDING ORDER.
         NUP  - THE NUMBER OF RIGHT ENDPOINTS.  IF NUP IS GREATER THAN
                1, THEN XUP AND ANS MUST BE DIMENSIONED AT LEAST NUP.

       --OUTPUT--

         ANS -- ARRAY OF INTEGRAL VALUES, THAT IS,
                ANS(I) = INTEGRAL FROM XLO TO XUP(I)
         IERR -- ERROR STATUS
                 = 1 INTEGRATION SUCCESSFUL
                 = 2 IMPROPER INPUT - N.LT.4 OR NUP.LT.1
                 = 3 IMPROPER INPUT - ABSCISSAS NOT IN
                         STRICTLY ASCENDING ORDER
```

```
= 4 IMPROPER INPUT - RIGHT ENDPOINTS XUP NOT
        IN ASCENDING ORDER
= 5 IMPROPER INPUT - XLO.GT.XUP(1)
= 6 INTEGRATION SUCCESSFUL BUT AT LEAST ONE ENDPOINT
        NOT WITHIN SPAN OF ABSCISSAS
** NOTE.  ERRCHK PROCESSES DIAGNOSTICS FOR CODES 2,3,4,5.
```

```
SSORT    SSORT    SSORT    SSORT       SSORT    SSORT    SSORT    SSORT
              ****************************************
                   ******************************
                        ********************
                             **********
    SUBROUTINE SSORT(X,Y,N,K)

    SANDIA MATHEMATICAL PROGRAM LIBRARY

        CONSULTANTS AT SLL INCLUDE -
            R. E. HUDDLESTON  -   DIVISION 8322
            T. H. JEFFERSON   -   DIVISION 8322

    WRITTEN BY RONDALL E JONES

    ABSTRACT
        SSORT SORTS ARRAY X AND OPTIONALLY MAKES THE SAME
        INTERCHANGES IN ARRAY Y.  THE ARRAY X MAY BE SORTED IN
        INCREASING ORDER OR DECREASING ORDER.  A SLIGHTLY MODIFIED
        SHELL SORT ALGORITHM IS USED.

    REFERENCE
        A HIGH SPEED SORTING PROCEDURE, D L SHELL, CACM JULY 1959

    DESCRIPTION OF PARAMETERS
        X - ARRAY OF VALUES TO BE SORTED   (USUALLY ABSCISSAS)
        Y - ARRAY TO BE (OPTIONALLY) CARRIED ALONG
        N - NUMBER OF VALUES IN ARRAY X TO BE SORTED
        K - CONTROL PARAMETER
            =2  MEANS SORT X IN INCREASING ORDER AND CARRY Y ALONG.
            =1  MEANS SORT X IN INCREASING ORDER (IGNORING Y)
            =-1 MEANS SORT X IN DECREASING ORDER (IGNORING Y)
            =-2 MEANS SORT X IN DECREASING ORDER AND CARRY Y ALONG.

 CHECK INPUT DATA
```

```
STEP1    STEP1    STEP1    STEP1       STEP1    STEP1    STEP1    STEP1
              ****************************************
                   ******************************
                        ********************
                             **********
    SUBROUTINE STEP1(F,NEQN,Y,X,H,EPS,WT,START,
   1 HOLD,K,KOLD,CRASH,PHI,P,YP,PSI,
   2 ALPHA,BETA,SIG,V,W,G,PHASE1,NS,NORND)

    SANDIA MATHEMATICAL PROGRAM LIBRARY

        CONSULTANTS AT SLL INCLUDE -
            R. E. HUDDLESTON  -   DIVISION 8322
            T. H. JEFFERSON   -   DIVISION 8322

  WRITTEN BY L. F. SHAMPINE AND M. K. GORDON

  ABSTRACT
```

SUBROUTINE STEP1 IS NORMALLY USED INDIRECTLY THROUGH SUBROUTINE ODE . BECAUSE ODE SUFFICES FOR MOST PROBLEMS AND IS MUCH EASIER TO USE, USING IT SHOULD BE CONSIDERED BEFORE USING STEP1 ALONE.

SUBROUTINE STEP1 INTEGRATES A SYSTEM OF NEQN FIRST ORDER ORDINAR DIFFERENTIAL EQUATIONS ONE STEP, NORMALLY FROM X TO X+H, USING A MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE FORMULAS. LOCAL EXTRAPOLATION IS USED TO IMPROVE ABSOLUTE STABILITY AND ACCURACY. THE CODE ADJUSTS ITS ORDER AND STEP SIZE TO CONTROL THE LOCAL ERROR PER UNIT STEP IN A GENERALIZED SENSE. SPECIAL DEVICES ARE INCLUDED TO CONTROL ROUNDOFF ERROR AND TO DETECT WHEN THE USER IS REQUESTING TOO MUCH ACCURACY.

THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT, COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS: THE INITIAL VALUE PROBLEM BY L. F. SHAMPINE AND M. K. GORDON.

THE PARAMETERS REPRESENT:

```
   F -- SUBROUTINE TO EVALUATE DERIVATIVES
   NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
   Y(*) -- SOLUTION VECTOR AT X
   X -- INDEPENDENT VARIABLE
   H -- APPROPRIATE STEP SIZE FOR NEXT STEP.  NORMALLY DETERMINED BY
        CODE
   EPS -- LOCAL ERROR TOLERANCE
   WT(*) -- VECTOR OF WEIGHTS FOR ERROR CRITERION
   START -- LOGICAL VARIABLE SET .TRUE. FOR FIRST STEP,  .FALSE.
        OTHERWISE
   HOLD -- STEP SIZE USED FOR LAST SUCCESSFUL STEP
   K -- APPROPRIATE ORDER FOR NEXT STEP (DETERMINED BY CODE)
   KOLD -- ORDER USED FOR LAST SUCCESSFUL STEP
   CRASH -- LOGICAL VARIABLE SET .TRUE. WHEN NO STEP CAN BE TAKEN,
        .FALSE. OTHERWISE.
   YP(*) -- DERIVATIVE OF SOLUTION VECTOR AT  X  AFTER SUCCESSFUL
        STEP
```

THE ARRAYS PHI, PSI ARE REQUIRED FOR THE INTERPOLATION SUBROUTINE INTRP . THE ARRAY P IS INTERNAL TO THE CODE. THE REMAINING NINE VARIABLES AND ARRAYS ARE INCLUDED IN THE CALL LIST ONLY TO ELIMINATE LOCAL RETENTION OF VARIABLES BETWEEN CALLS.

INPUT TO STEP1

FIRST CALL --

THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR ALL ARRAYS IN THE CALL LIST, NAMELY

```
  DIMENSION Y(NEQN),WT(NEQN),PHI(NEQN,16),F(NEQN),YP(NEQN),PSI(12),
 1  ALPHA(12),BETA(12),SIG(13),V(12),W(12),G(13)
```

THE USER MUST ALSO DECLARE START , CRASH , PHASE1 AND NORND LOGICAL VARIABLES AND F AN EXTERNAL SUBROUTINE, SUPPLY THE SUBROUTINE F(X,Y,YP) TO EVALUATE

```
   DY(I)/DX = YP(I) = F(X,Y(1),Y(2),...,Y(NEQN))
```

AND INITIALIZE ONLY THE FOLLOWING PARAMETERS:

```
   NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
   Y(*) -- VECTOR OF INITIAL VALUES OF DEPENDENT VARIABLES
   X -- INITIAL VALUE OF THE INDEPENDENT VARIABLE
   H -- NOMINAL STEP SIZE INDICATING DIRECTION OF INTEGRATION
        AND MAXIMUM SIZE OF STEP.  MUST BE VARIABLE
   EPS -- LOCAL ERROR TOLERANCE PER STEP.  MUST BE VARIABLE
   WT(*) -- VECTOR OF NON-ZERO WEIGHTS FOR ERROR CRITERION
   START -- .TRUE.
```

STEP1 REQUIRES THAT THE L2 NORM OF THE VECTOR WITH COMPONENTS LOCAL ERROR(L)/WT(L) BE LESS THAN EPS FOR A SUCCESSFUL STEP. THE ARRAY WT ALLOWS THE USER TO SPECIFY AN ERROR TEST APPROPRIATE FOR HIS PROBLEM. FOR EXAMPLE,

```
   WT(L) = 1.0  SPECIFIES ABSOLUTE ERROR,
         = ABS(Y(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF THE
              L-TH COMPONENT OF THE SOLUTION,
```

```
= ABS(YP(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF
     THE L-TH COMPONENT OF THE DERIVATIVE,
= AMAX1(WT(L),ABS(Y(L)))  ERROR RELATIVE TO THE LARGEST
     MAGNITUDE OF L-TH COMPONENT OBTAINED SO FAR,
= ABS(Y(L))*RELERR/EPS + ABSERR/EPS  SPECIFIES A MIXED
     RELATIVE-ABSOLUTE TEST WHERE  RELERR  IS RELATIVE
     ERROR,  ABSERR  IS ABSOLUTE ERROR AND  EPS =
     AMAX1(RELERR,ABSERR) .
```

SUBSEQUENT CALLS --

SUBROUTINE STEP1 IS DESIGNED SO THAT ALL INFORMATION NEEDED TO CONTINUE THE INTEGRATION, INCLUDING THE STEP SIZE H AND THE ORDER K , IS RETURNED WITH EACH STEP. WITH THE EXCEPTION OF THE STEP SIZE, THE ERROR TOLERANCE, AND THE WEIGHTS, NONE OF THE PARAMETERS SHOULD BE ALTERED. THE ARRAY WT MUST BE UPDATED AFTER EACH STEP TO MAINTAIN RELATIVE ERROR TESTS LIKE THOSE ABOVE. NORMALLY THE INTEGRATION IS CONTINUED JUST BEYOND THE DESIRED ENDPOINT AND THE SOLUTION INTERPOLATED THERE WITH SUBROUTINE INTRP . IF IT IS IMPOSSIBLE TO INTEGRATE BEYOND THE ENDPOINT, THE STEP SIZE MAY BE REDUCED TO HIT THE ENDPOINT SINCE THE CODE WILL NOT TAKE A STEP LARGER THAN THE H INPUT. CHANGING THE DIRECTION OF INTEGRATION, I.E., THE SIGN OF H , REQUIRES THE USER SET START = .TRUE. BEFORE CALLING STEP1 AGAIN. THIS IS THE ONLY SITUATION IN WHICH START SHOULD BE ALTERED.

OUTPUT FROM STEP1

SUCCESSFUL STEP --

THE SUBROUTINE RETURNS AFTER EACH SUCCESSFUL STEP WITH START AND CRASH SET .FALSE. . X REPRESENTS THE INDEPENDENT VARIABLE ADVANCED ONE STEP OF LENGTH HOLD FROM ITS VALUE ON INPUT AND Y THE SOLUTION VECTOR AT THE NEW VALUE OF X . ALL OTHER PARAMETERS REPRESENT INFORMATION CORRESPONDING TO THE NEW X NEEDED TO CONTINUE THE INTEGRATION.

UNSUCCESSFUL STEP --

WHEN THE ERROR TOLERANCE IS TOO SMALL FOR THE MACHINE PRECISION, THE SUBROUTINE RETURNS WITHOUT TAKING A STEP AND CRASH = .TRUE. . AN APPROPRIATE STEP SIZE AND ERROR TOLERANCE FOR CONTINUING ARE ESTIMATED AND ALL OTHER INFORMATION IS RESTORED AS UPON INPUT BEFORE RETURNING. TO CONTINUE WITH THE LARGER TOLERANCE, THE USER JUST CALLS THE CODE AGAIN. A RESTART IS NEITHER REQUIRED NOR DESIRABLE.

```
SUPORT    SUPORT    SUPORT    SUPORT         SUPORT    SUPORT    SUPORT
                 ****************************************
                      ******************************
                           ********************
                                **********
      SUBROUTINE SUPORT(Y,NROWY,NCOMP,XPTS,NXPTS,A,NROWA,ALPHA,NIC,
     1                  B,NROWB,BETA,NFC,IGOFX,RE,AE,IFLAG,WORK,NDW,
     2                  IWORK,NDIW)
```

SANDIA MATHEMATICAL PROGRAM LIBRARY

CONSULTANTS AT SLL INCLUDE -
R. E. HUDDLESTON - DIVISION 8322
T. H. JEFFERSON - DIVISION 8322

WRITTEN BY MELVIN R. SCOTT AND HERMAN A. (BUDDY) WATTS

****

ABSTRACT

```
SUBROUTINE SUPORT SOLVES A LINEAR TWO-POINT BOUNDARY-VALUE PROBLEM
OF THE FORM
                    DY/DX = MATRIX(X)*Y(X) + G(X)
          A*Y(XINITIAL) = ALPHA,   B*Y(XFINAL) = BETA

   THE METHOD OF SOLUTION USES SUPERPOSITION COUPLED WITH AN
ORTHONORMALIZATION PROCEDURE AND A VARIABLE-STEP RUNGE-KUTTA-
FEHLBERG INTEGRATION SCHEME. EACH TIME THE SUPERPOSITION SOLUTIONS
START TO LOSE THEIR NUMERICAL LINEAR INDEPENDENCE, THE VECTORS ARE
REORTHONORMALIZED BEFORE INTEGRATION PROCEEDS. THE UNDERLYING
PRINCIPLE OF THE ALGORITHM IS THEN TO PIECE TOGETHER THE
INTERMEDIATE (ORTHOGONALIZED) SOLUTIONS, DEFINED ON THE VARIOUS
SUBINTERVALS, TO OBTAIN THE DESIRED SOLUTIONS.

*******************************************************************

REFERENCES
   M.R. SCOTT AND H.A. WATTS, SUPORT - A COMPUTER CODE FOR TWO-
                 POINT BOUNDARY-VALUE PROBLEMS VIA
                 ORTHONORMALIZATION, SAND75-0198, SANDIA LABS.,
                 ALBUQUERQUE, NEW MEXICO, 1975.

   S.K. GODUNOV, ON THE NUMERICAL SOLUTION OF BOUNDARY VALUE
                 PROBLEMS FOR SYSTEMS OF LINEAR ORDINARY
                 DIFFERENTIAL EQUATIONS, USPEKHI. MAT. NAUK.,
                 VOL. 16, 1961, 171-174.

*******************************************************************
INPUT TO SUPORT
*******************************************************************

NROWY = ACTUAL ROW DIMENSION OF Y IN CALLING PROGRAM.
        NROWY MUST BE .GE. NCOMP

NCOMP = NUMBER OF COMPONENTS PER SOLUTION VECTOR.
        NCOMP IS EQUAL TO NUMBER OF ORIGINAL DIFFERENTIAL
        EQUATIONS.  NCOMP = NIC + NFC.

XPTS = DESIRED OUTPUT POINTS FOR SOLUTION. THEY MUST BE MONOTONIC.
       XINITIAL = XPTS(1)
       XFINAL = XPTS(NXPTS)

NXPTS = NUMBER OF OUTPUT POINTS

A(NROWA,NCOMP) = BOUNDARY CONDITION MATRIX AT XINITIAL
                 MUST BE CONTAINED IN (NIC,NCOMP) SUB-MATRIX.

NROWA = ACTUAL ROW DIMENSION OF A IN CALLING PROGRAM,
        NROWA MUST BE .GE. NIC.

ALPHA(NIC) = BOUNDARY CONDITIONS AT XINITIAL.

NIC = NUMBER OF BOUNDARY CONDITIONS AT XINITIAL.

B(NROWB,NCOMP) = BOUNDARY CONDITION MATRIX AT XFINAL.
                 MUST BE CONTAINED IN (NFC,NCOMP) SUB-MATRIX.

NROWB = ACTUAL ROW DIMENSION OF B IN CALLING PROGRAM,
        NROWB MUST BE .GE. NFC.

BETA(NFC) = BOUNDARY CONDITIONS AT XFINAL.

NFC = NUMBER OF BOUNDARY CONDITIONS AT XFINAL

       0 -- THE INHOMOGENEOUS TERM G(X) IS IDENTICALLY ZERO.
IGOFX =
       1 -- THE INHOMOGENEOUS TERM G(X) IS NOT IDENTICALLY ZERO.
            (IF IGOFX=1, THEN SUBROUTINE GVEC MUST BE SUPPLIED.)

RE = RELATIVE ERROR TOLERANCE USED BY THE INTEGRATOR
     (SEE GERK OR RKF)

AE = ABSOLUTE ERROR TOLERANCE USED BY THE INTEGRATOR
```

```
          (SEE GERK OR RKF)
***NOTE-  SINCE THE COST TYPICALLY RISES RAPIDLY FOR TOLERANCES BELOW
          1.0E-08, WE SUGGEST THE USE OF LARGER ERROR TOLERANCES. ALSO,
          RE AND AE SHOULD NOT BOTH BE ZERO.

     WORK(NDW) = FLOATING POINT ARRAY USED FOR INTERNAL STORAGE.

     NDW = ACTUAL DIMENSION OF WORK ARRAY ALLOCATED BY USER.
           AN ESTIMATE FOR NDW CAN BE COMPUTED FROM THE FOLLOWING
                 NDW=NCOMP**2*(10 + NXPTS/2 + EXPECTED NUMBER OF
                                              ORTHONORMALIZATIONS/8)

     IWORK(NDIW) = INTEGER ARRAY USED FOR INTERNAL STORAGE.

     NDIW = ACTUAL DIMENSION OF IWORK ARRAY ALLOCATED BY USER.
            AN ESTIMATE FOR NDIW CAN BE COMPUTED FROM THE FOLLOWING
                 NDIW=11+NCOMP*(1 + EXPECTED NUMBER OF
                                    ORTHONORMALIZATIONS)
***NOTE --  THE AMOUNT OF STORAGE REQUIRED IS PROBLEM DEPENDENT AND MAY
            BE DIFFICULT TO PREDICT IN ADVANCE. EXPERIENCE HAS SHOWN
            THAT FOR MOST PROBLEMS 20 OR FEWER ORTHONORMALIZATIONS
            SHOULD SUFFICE. IF THE PROBLEM CANNOT BE COMPLETED WITH THE
            ALLOTTED STORAGE, THEN A MESSAGE WILL BE PRINTED WHICH
            ESTIMATES THE AMOUNT OF STORAGE NECESSARY.


     THE USER MUST SUPPLY SUBROUTINES FMAT AND GVEC (THEY MUST BE
     NAMED FMAT AND GVEC) TO EVALUATE THE DERIVATIVES AS FOLLOWS

              SUBROUTINE FMAT(X,Y,YP)
              X = INDEPENDENT VARIABLE (INPUT TO FMAT)
              Y = DEPENDENT VARIABLE VECTOR (INPUT TO FMAT)
              YP = DY/DX = DERIVATIVE VECTOR (OUTPUT FROM FMAT)

            COMPUTE THE DERIVATIVES FOR THE HOMOGENEOUS PROBLEM
              YP(I) = DY(I)/DX = MATRIX(X) * Y(I)   , I = 1,...,NCOMP

            SUBROUTINE BVPDER CALLS FMAT NFC TIMES TO EVALUATE THE
            HOMOGENEOUS EQUATIONS AND,IF NECESSARY,IT CALLS FMAT ONCE
            IN EVALUATING THE PARTICULAR SOLUTION. SINCE X REMAINS
            UNCHANGED IN THIS SEQUENCE OF CALLS IT IS POSSIBLE TO
            REALIZE CONSIDERABLE COMPUTATIONAL SAVINGS FOR COMPLICATED
            AND EXPENSIVE EVALUATIONS OF THE MATRIX ENTRIES. TO DO THIS
            THE USER MERELY PASSES A VARIABLE,SAY XS,VIA COMMON WHERE
            XS IS DEFINED IN THE MAIN PROGRAM TO BE ANY VALUE EXCEPT
            THE INITIAL X. THEN THE NON-CONSTANT ELEMENTS OF MATRIX(X)
            APPEARING IN THE DIFFERENTIAL EQUATIONS NEED ONLY BE
            COMPUTED IF X IS UNEQUAL TO XS ,WHEREUPON XS IS RESET TO X.


              SUBROUTINE GVEC(X,G)
              X = INDEPENT VARIABLE (INPUT TO GVEC)
              G = VECTOR OF INHOMOGENEOUS TERMS G(X) (OUTPUT FROM GVEC)

            COMPUTE THE INHOMOGENEOUS TERMS G(X)
                G(I) = G(X) VALUES FOR I = 1,...,NCOMP.

            SUBROUTINE BVPDER CALLS GVEC IN EVALUATING THE PARTICULAR
            SOLUTION PROVIDED G(X) IS NOT IDENTICALLY ZERO. THUS, WHEN
            IGOFX=0, THE USER NEED NOT WRITE A GVEC SUBROUTINE. ALSO,
            THE USER DOES NOT HAVE TO BOTHER WITH THE COMPUTATIONAL
            SAVINGS SCHEME FOR GVEC AS THIS IS AUTOMATICALLY ACHIEVED
            VIA THE BVPDER SUBROUTINE.


     THE FOLLOWING IS OPTIONAL INPUT TO SUPORT TO GIVE USER MORE
     FLEXIBILITY IN USE OF CODE. SEE SAND75-0198 FOR MORE INFORMATION.

*****CAUTION -- THE USER IS ADVISED TO ZERO OUT IWORK(1),...,IWORK(10)
               PRIOR TO CALLING SUPORT.

     IWORK(7) -- IF IWORK(7) = -1, THEN USER CAN INPUT INTO IWORK(8)
                 THE EXPONENT PARAMETER TO BE USED IN TOLERANCE TEST
```

```
                FOR ORTHONORMALIZATION.

     IWORK(8) -- THE VALUE OF THE EXPONENT PARAMETER IN THE TOLERANCE
                 TEST FOR ORTHONORMALIZATION. IF USER HAS NOT SET
                 IWORK(7)=-1, THEN THE DEFAULT VALUE IS 0.
                 DECREASING THE VALUE OF IWORK(8) RESULTS IN MORE
                 FREQUENT ORTHONORMALIZATIONS.

     IWORK(9) -- INTEGRATOR AND ORTHONORMALIZATION PARAMETER
                 (DEFAULT VALUE IS 1)
                 1 = RKF USING GRAM-SCHMIDT TEST.
                 2 = GERK USING GLOBAL ERROR TEST.

     IWORK(10) -- NORMALIZATION OF PARTICULAR SOLUTION
                  (DEFAULT VALUE IS 0)
                  0 - NORMALIZE PARTICULAR SOLUTION TO UNIT LENGTH AT
                      EACH POINT OF ORTHONORMALIZATION.
                  1 - DO NOT NORMALIZE PARTICULAR SOLUTION

*******************************************************************
     OUTPUT FROM SUPORT
*******************************************************************

     Y(NROWY,NXPTS) = SOLUTION AT SPECIFIED OUTPUT POINTS.

     IFLAG OUTPUT VALUES
            =-3 SUPORT WAS CALLED WITH INCORRECT NUMBER OF ARGUMENTS.
            =-2 INVALID INPUT PARAMETERS.
            =-1 INSUFFICIENT NUMBER OF STORAGE LOCATIONS ALLOCATED FOR
                WORK OR IWORK.

            =0 INDICATES SUCCESSFUL SOLUTION

            =1 A COMPUTED SOLUTION IS RETURNED BUT UNIQUENESS OF THE
               SOLUTION OF THE BOUNDARY-VALUE PROBLEM IS QUESTIONABLE.
            =2 A COMPUTED SOLUTION IS RETURNED BUT THE EXISTENCE OF THE
               SOLUTION TO THE BOUNDARY-VALUE PROBLEM IS QUESTIONABLE.
            ***NOTE-WE ATTEMPT TO DIAGNOSE THE CORRECT PROBLEM BEHAVIOR
               AND REPORT POSSIBLE DIFFICULTIES BY THE APPROPRIATE
               ERROR FLAG. HOWEVER,THE USER SHOULD PROBABLY RESOLVE THE
               PROBLEM USING SMALLER ERROR TOLERANCES AND/OR
               PERTURBATIONS IN THE BOUNDARY CONDITIONS OR OTHER
               PARAMETERS. THIS WILL OFTEN REVEAL THE CORRECT
               INTERPRETATION FOR THE PROBLEM POSED.
            =3 NO SOLUTION RETURNED BECAUSE THE BOUNDARY CONDITION
               MATRIX B*Y(XFINAL) IS SINGULAR.
            =4 MAXIMUM NUMBER OF ORTHONORMALIZATIONS ATTAINED BEFORE
               REACHING XFINAL.
            =13 RANK OF BOUNDARY CONDITION MATRIX A IS LESS THAN NIC.
            =20+FLAG FROM INTEGRATOR (GERK OR RKF).
            =30 HOMOGENEOUS VECTORS FORM A DEPENDENT SET.
            =31 HOMOGENEOUS VECTORS PLUS PARTICULAR VECTOR FORM A
                DEPENDENT SET.

     WORK(1),...,WORK(IWORK(1)) = ORTHONORMALIZATION POINTS
                                  DETERMINED BY BVPORT.

     IWORK(1) = NUMBER OF ORTHONORMALIZATIONS PERFORMED BY BVPORT.

     IWORK(2) = MAXIMUM NUMBER OF ORTHONORMALIZATIONS ALLOWED AS
                CALCULATED FROM STORAGE ALLOCATED BY USER.

     IWORK(3),IWORK(4),IWORK(5),IWORK(6)   GIVE INFORMATION ABOUT
                ACTUAL STORAGE REQUIREMENTS FOR WORK AND IWORK
                ARRAYS.  IN PARTICULAR,
                        REQUIRED STORAGE FOR  WORK ARRAY IS
        IWORK(3) + IWORK(4)*(EXPECTED NUMBER OF ORTHONORMALIZATIONS)

                        REQUIRED STORAGE FOR IWORK ARRAY IS
        IWORK(5) + IWORK(6)*(EXPECTED NUMBER OF ORTHONORMALIZATIONS)

     IWORK(8) = FINAL VALUE OF EXPONENT PARAMETER USED IN TOLERANCE
                TEST FOR ORTHONORMALIZATION.
```

```
     IWORK(11) = NUMBER OF INDEPENDENT VECTORS RETURNED FROM MGS.
                 IT IS ONLY OF INTEREST WHEN IFLAG=30 OR 31 IS OBTAINED.

*********************************************************************

*********************************************************************

     THE FOLLOWING ARE MACHINE CONSTANTS AND THE SUBROUTINES IN WHICH
     THEY APPEAR. THESE ARE SET FOR THE CDC-6600 WHICH CARRIES
     APPROXIMATELY FOURTEEN DIGITS.

     SUBROUTINE NAME        VARIABLE NAME        VALUE
     SUPORT                 LPAR                  7
     BVPORT                 EPS                  1.0E-11
     MGS                    EPS                  1.0E-11
     COEFF                  SRU                  1.0E-07
     UCHECK                 URO                  7.1E-15
     RKFS                   U                    7.1E-15
     GERKS                  U                    7.1E-15

     THE COMPUTER UNIT ROUNDOFF ERROR U IS THE SMALLEST POSITIVE VALUE
     REPRESENTABLE IN THE MACHINE SUCH THAT 1.+U .GT. 1.
                    VALUES TO BE USED ARE
               U = 9.5E-07    FOR IBM 360/370
               U = 1.5E-08    FOR UNIVAC 1108
               U = 7.5E-09    FOR POP-10
               U = 7.1E-15    FOR CDC-6600
               U = 2.2E-16    FOR IBM 360/370 DOUBLE PRECISION
*********************************************************************
*********************************************************************


TJMAR1    TJMAR1    TJMAR1    TJMAR1         TJMAR1    TJMAR1    TJMAR1
                 ****************************************
                      ******************************
                           ********************
                                **********
     SUBROUTINE TJMAR1 (B,LIN,XIN,WORK,XDATA,Y,IB,FUNCT,DERIV)
       FOR DOCUMENTATION SEE  SLL-73-0305  TJMAR1, A FORTRAN SUBROUTINE
       FOR NONLINEAR LEAST SQUARES PARAMETER ESTIMATION
     TJMAR1 WRITTEN BY THOMAS H. JEFFERSON JR, SANDIA, LIVERMORE
 * * * * *  TJMAR1 IS NOT A ROUTINE OFFICIALLY ON THE MATH LIBRARY, BUT
 * * * * *          IT IS SUPPORTED BY THE AUTHOR AT LIVERMORE.

         TJMAR1 SOLVES THE NONLINEAR LEAST SQUARES PROBLEM AS FOLLOWS.
       GIVEN N+NCONS FUNCTIONS, RES SUB I(B) , I=1,N+NCONS, OF THE
       K-VECTOR OF PARAMETERS B, FIND THE VALUE OF THE VECTOR B THAT
       MINIMIZES PHI, THE SUM OF SQUARES OF THE N+NCONS RESIDUAL
       FUNCTIONS.
            EXAMPLES OF USE ARE IN DATA FITTING WHERE THE RESIDUAL
       FUNCTIONS MIGHT BE DEFINED AS THE DIFFERENCE BETWEEN THE
       OBSERVED VALUE AND THE PREDICTED VALUE AT EACH DATA POINT.
                      RES SUB I = (Y(I) - F(XDATA(I),B )
            ANOTHER APPLICATION IS IN SOLVING A SYSTEM OF ALGEBRAIC
       EQUATIONS WHERE THE RESIDUAL FUNCTIONS COULD BE DEFINED AS THE
       DIFFERENCE BETWEEN THE RIGHTHAND  AND LEFTHAND SIDES OF
       EACH EQUATION.
         B        INPUT AND OUTPUT ARRAY OF PARAMETERS.
                       INPUT  INITIAL GUESS FOR PARAMETER VALUES.
                       OUTPUT  FINAL VALUES FOR PARAMETERS.
         LIN      INPUT ARRAY OF INTEGER SUBROUTINE CONSTANTS.
         XIN      INPUT ARRAY OF REAL SUBROUTINE CONSTANTS.
         WORK     TEMPORARY WORK ARRAY OF LENGTH AT LEAST
                        6*K+.5*K*(K+1) FOR AUXILIARY FILE MODE OR
                        7*K+K*(K+1) FOR ALL IN CORE MODE.
                  ON OUTPUT
                    WORK(1)=CONV, STOP INDICATOR.
                       CONV=0.  ERROR IN TJMAR1
                       CONV=1.  EPSILON TEST--PARAMETERS DID NOT CHANGE
```

```
                                MUCH FROM ONE ITERATION TO THE NEXT
                    CONV=2.  PHI.LE.PHIMN--SUM OF SQUARES.LE.PHIMN
                    CONV=3.  GAMMA LAMBDA TEST--COULD BE ROUNDOFF
                                PROBLEMS.
                    CONV=4.  GRADIENT OF PHI IS ZERO
                    CONV=5.  FORCE OFF--MAXIMUM NUMBER ITERATIONS
                                EXCEEDED.  SEE LIN(11).
                    CONV=6.  USER ERROR--CHECK INPUT.
                  WORK(2)=PHIZ, SMALLEST SUM OF SQUARES SO FAR.
                  WORK(3)=XL, LAST VALUE OF LAMBDA.
                  WORK(4)=GAMMA, LAST ANGLE(IN DEGREES) BETWEEN
                              GRADIENT AND LAST INCREMENT.
                  WORK(5)=PHIUNZ=SUM OF SQUARES OF UNCONSTRAINED
                          RESIDUALS USING BEST PARAMETERS.
                  WORK(6)=PHICNZ=SUM OF SQUARES OF CONSTRAINT
                          RESIDUALS USING BEST PARAMETERS.
     XDATA    INPUT ARRAY CONTAINING VALUES OF INDEPENDENT VARIABLE
                  XDATA IS NOT USED BY TJMAR1 EXCEPT TO PASS IN
                  ARGUMENT LIST TO FUNCT AND DERIV.
     Y        INPUT ARRAY CONTAINING VALUES OF DEPENDENT VARIABLE
                   AT THE N DATA POINTS.
     IB       INTEGER INPUT ARRAY CONTAINING NUMBERS OF OMITTED
                   PARAMETERS.
     FUNCT    USER DEFINED EXTERNAL SUBROUTINE FOR EVALUATING
              FUNCTION AND CONSTRAINT RESIDUALS.  FORM OF ROUTINE
                       SUBROUTINE FUNCT(I,XDATA,Y,B,F,RES,PRNT)
                       DIMENSION XDATA(  ),Y(1),B(1),PRNT(5)
                  TJMAR1 WILL PASS TO THIS ROUTINE I,XDATA,Y, AND
              B.  F, RES, AND PRNT( ) SHOULD THEN BE DEFINED IN
              FUNCT.  I IS THE DATA POINT NUMBER BETWEEN 1 AND
              N+NCONS.  XDATA AND Y ARE AS IN TJMAR1 CALLING LIST.
              B IS THE CURRENT SET OF PARAMETERS BEING CONSIDERED.
              F SHOULD BE THE FUNCTION VALUE AT DATA POINT I.
              RES SHOULD BE THE RESIDUAL AT DATA POINT I.
              PRNT(J),J=1,NPRNT ARE THE ADDITIONAL ITEMS TO BE
              PRINTED ALONG WITH DATA IF LIN(12) IS POSITIVE.
     DERIV    USER DEFINED EXTERNAL SUBROUTINE FOR EVALUATING
              ANALYTIC DERIVATIVES IF LIN(5)=2.  DERIV IS NOT
              CALLED IF ESTIMATED DERIVATIVES ARE USED, BUT EVEN
              THEN SOME ARGUMENT MUST APPEAR IN ITS PLACE IN
              THE TJMAR1 CALLING LIST.  FORM OF ROUTINE
                       SUBROUTINE DERIV(I,XDATA,Y,B,F,RES,PD)
                       DIMENSION XDATA( ),Y(1),B(1),PD(1)
                  I,XDATA,Y,B,F,RES ARE AS DEFINED ON THE PREVIOUS
              CALL TO FUNCT.  FOR J=1,K PD(J) SHOULD BE DEFINED
              TO BE THE NEGATIVE OF THE DERIVATIVE OF RESIDUAL I
              WITH RESPECT TO PARAMETER B(J).
*****      FUNCT AND DERIV MUST APPEAR IN AN EXTERNAL STATEMENT
*****                 IN THE ROUTINE THAT CALLS TJMAR1.

                 LIN ARRAY BELOW

  1  K        NUMBER OF PARAMETERS.
  2  N        NUMBER OF DATA POINTS.
  3  NCONS    NUMBER OF CONSTRAINTS, I.E. RESIDUALS WITH NO  DATA.
  4  NF       NUMBER OF OMITTED PARAMETERS.
  5  KORV     .EQ.1  ESTIMATED DERIVATIVES.
              .EQ.2   ANALYTIC DERIVATIVES.
  6  LIST1    DETERMINES TYPE OUTPUT BEFORE INIT ITERATIONS
  7  LIST2    DETERMINES OUTPUT AFTER INIT ITERATIONS BUT BEFORE
                  CONVERGENCE OR FORCE OFF.
  8  LIST3    DETERMINES TYPE OUTPUT AFTER CONVERGENCE OR FORCE OFF
  9  INIT     .GT.0  LESS DETAILED PRINT OUT AFTER INIT ITERATIONS.
              .EQ.0  SAME TYPE PRINT OUT FOR ALL ITERATIONS.
 10  LOP      .EQ.1  PRINTER PLOT SCALE DETERMINED BY INPUT YMN AND
                       YMX.
              .EQ.2  PRINTER PLOT AUTOMATICALLY SCALED BY ROUTINE.
              .EQ.3  NO PRINTER PLOT.  DATA LISTED INSTEAD.
 11  KILL     .GE.1  FORCE OFF AFTER KILL ITERATIONS.
              .EQ.0  NO FORCE OFF
 12  NPRNT    NUMBER OF ADDITIONAL WORDS TO BE PRINTED AT EACH
                  DATA POINT.  MUST BE .GE.0  AND .LE.5.
 13  ITAPE    NUMBER OF FILE ON WHICH OUTPUT IS WRITTEN.
                 ITAPE=6 IS DEFAULT SO ON PROGRAM CARD SHOULD BE
```

```
                    DECLARED TAPE6=OUTPUT.
 14  JKTAPE   NUMBER OF TEMPORARY SCRATCH FILE.
                JKTAPE=3 IS DEFAULT, SO IF SCRATCH FILE IS USED
                TAPE3  SHOULD BE DECLARED ON PROGRAM CARD.
 15  LNWORK   .LE. NUMBER OF WORDS AVAILABLE IN TEMPORARY
                 WORK ARRAY WORK.   USED IN DETERMINING WHETHER OR
                  NOT TO WRITE ON SCRATCH FILE.
 16  JCENTD   .EQ.1  DO NOT USE CENTRAL DIFFERENCES IN CALCULATING
                  ESTIMATED PARTIAL DERIVATIVES.
              .EQ.2  USE CENTRAL DIFFERENCES ONLY AFTER CONVERGENCE
                  WITHOUT CENTRAL DIFFERENCES.
              .EQ.3  USE CENTRAL DIFFERENCES FOR ALL ESTIMATED
                  PARTIAL DERIVATIVES.

                        XIN ARRAY BELOW

  1  YMN      MINIMUM SCALE VALUE FOR PRINTER PLOTTING.
  2  YMX      MAXIMUM SCALE VALUE FOR PRINTER PLOTTING.
  3  XLAM     INITIAL VALUE TO BE ADDED TO DIAGONAL OF FTP=A MATRIX
  4  DLT
  5  DEL
              IN ESTIMATED DERIVATIVE ROUTINE  B*DEL + OR - DLT
                  IS THE INCREMENT USED FOR CALCUALTING DERIVATIVE
  6  GAMCR    CRITICAL VALUE OF ANGLE BETWEEN GRADIENT AND DB
                  INCREMENTS IN CONVERGENCE ROUTINE.  ANGLE IS
                  MEASURED IN SCALED SPACE.
  7  E        USED IN EPSILON TEST.
  8  TAU      USED IN EPSILON TEST.
                  EPSILON TEST FOR CONVERGENCE IS SATISFIED
              IF FROM ONE ITERATION TO THE NEXT ALL COMPONENTS OF
              PARAMETER VECTOR B AND CORRESPONDING INCREMENT VECTOR
              DELB SATISFY
              ABS(DELB(J)) .LE. (E*ABS(B(J))+TAU) , J=1,K
  9  PHIMN    END ITERATING AND RETURN IF PHI.LE.PHIMN


IF ANY VALUE IN THE XIN AND LIN ARRAYS IS .LE. 0 THE DEFAULT VALUE
          CORRESPONDING TO THAT INPUT CONSTANT WILL BE USED.
                      THIS DOES NOT APPLY TO XIN(1) AND XIN(2).
     LIN(1),LIN(2),LIN(15) MUST BE SPECIFIED AS POSITIVE
          QUANTITIES OR ERRORS WILL OCCUR.  THE BUILT IN DEFAULT
          VALUES ARE USUALLY SATISFACTORY FOR ALL OTHER XIN
          AND LIN QUANTITIES.

-----FURTHER EXPLANATION OF THE THREE LIST PARAMETERS, LISTV,V=1,2,3
-----    LISTV = 1   NC OUTPUT
-----          = 2   ERRORS
-----          = 3   ERRORS+ABBREVIATED
-----          = 4   ERRORS+ABBREVIATED+DATA(PLOT OR COLUMN DEP ON LOP)
-----          = 5   ERRORS+ABBREV.+CORREL MATRIX
-----          = 6   ERRORS+ABBREV.+CORREL MATRIX+DATA
-----          = 7   ALL EXCEPT FOR BOTH DATA AND CORREL. MATRIX
-----          = 8   ALL EXCEPT DATA
-----          = 9   ALL
-----    LISTV.GE.10      INDIVIDUAL DIGITS ARE EXAMINED TO DETERMINE
-----                     OPTIONS.
-----              DIGITS ARE ORDERED FROM LOW ORDER TO HIGH ORDER.
-----                                 -----  DIGIT ONE.
-----          = 1    HEADER
-----          = 2    TABLE OF INPUT CONSTANTS.
-----          = 3    BOTH HEADER AND TABLE OF INPUT CONSTANTS.
-----                                 -----  DIGIT TWO.
-----          = 1    COLUMNAR DATA LISTING
-----          = 2    PRINTER PLOT OF DATA.
-----          = 3    BOTH COLUMNAR DATA LISTING AND PRINTER PLOT.
-----                                 -----  DIGIT THREE.
-----          = 1    CORRELATION MATRIX.
-----                                 -----  DIGIT FOUR.
-----          = 1    INCREMENTS
-----          = 2    BETTER PHI OR NOT.
-----          = 3    BOTH INCREMENTS AND BETTER PHI OR NOT.
-----                                 -----  DIGIT FIVE.
-----          = 1    PARAMETERS
-----          = 2    PHI,GAMMA,LAMBDA,...
```

```
-----            = 3    BOTH PARAMETERS AND PHI,GAMMA,LAMBDA,...
-----            = 4    HAS STRECHING HELPED.
-----            = 5    BOTH PARAMETERS AND HAS STRECHING HELPED.
-----            = 6    BOTH PHI,GAMMA,LAMBDA,... AND STRECHING HELPED.
-----            = 7    ALL THREE--PARAMETERS,PHI,...., AND STRECHING.
-----      IF ANY DIGIT IS ZERO, THE CORRESPONDING OPTIONS ARE NOT
-----           CHOSEN.
-----      IF LISTV IS .GE.2 THEN ERRORS WILL BE PRINTED.
-----      IF ANY MORE OPTIONS THAN ERRORS ARE CALLED FOR, THEN
-----                HEADER(AND TRAILER) IS CHOSEN AUTOMATICALLY.
-----
-----
-----
```

```
ZEROIN    ZEROIN    ZEROIN    ZERCIN          ZEROIN    ZEROIN    ZEROIN
                 ****************************************
                      ******************************
                           ********************
                                **********
      SUBROUTINE ZEROIN(F,B,C,RE,AE,IFLAG)

      SANDIA MATHEMATICAL PROGRAM LIBRARY

          CONSULTANTS AT SLL INCLUDE -
              R. E. HUDDLESTON  -   DIVISION 8322
              T. H. JEFFERSON   -   DIVISION 8322

      BASED ON A METHOD BY T J DEKKER
      WRITTEN BY L F SHAMPINE AND H A WATTS
      MODIFIED FOR THE MATH LIBRARY BY C B BAILEY

      ABSTRACT
         ZEROIN SEARCHES FOR A ZERO OF A FUNCTION F(X) BETWEEN
         THE GIVEN VALUES B AND C UNTIL THE WIDTH OF THE INTERVAL
         (B,C) HAS COLLAPSED TO WITHIN A TOLERANCE SPECIFIED BY
         THE STOPPING CRITERION, ABS(B-C) .LE. 2.*(RW*ABS(B)+AE).
         THE METHOD USED IS AN EFFICIENT COMBINATION OF BISECTION AND
         THE SECANT RULE.  IN ORDER TO INSURE THAT ZERCIN WILL CONVERGE
         TO A ZERO, THE USER SHOULD PICK VALUES FOR B AND C AT WHICH
         THE FUNCTION DIFFERS IN SIGN.

      DESCRIPTION OF ARGUMENTS
      F,B,C,RE AND AE ARE INPUT PARAMETERS
      B,C AND IFLAG ARE OUTPUT PARAMETERS
         F     - NAME OF THE REAL VALUED EXTERNAL FUNCTION.  THIS NAME
                 MUST BE IN AN EXTERNAL STATEMENT IN THE CALLING
                 PROGRAM.  F MUST BE A FUNCTION OF ONE REAL ARGUMENT.
         B     - ONE END OF THE INTERVAL (B,C).  THE VALUE RETURNED FOR
                 B USUALLY IS THE BETTER APPROXIMATION TO A ZERO OF F.
         C     - THE OTHER END OF THE INTERVAL (B,C)
         RE    - RELATIVE ERROR USED FOR RW IN THE STOPPING CRITERION.
                 IF THE REQUESTED RE IS LESS THAN MACHINE PRECISION,
                 THEN RW IS SET TO APPROXIMATELY MACHINE PRECISION.
         AE    - ABSOLUTE ERROR USED IN THE STOPPING CRITERION.  IF THE
                 GIVEN INTERVAL (B,C) CONTAINS THE ORIGIN, THEN A
                 NONZERO VALUE SHOULD BE CHOSEN FOR AE.
         IFLAG - A STATUS CODE.  USER MUST CHECK IFLAG AFTER EACH CALL.
                 CONTROL RETURNS TO THE USER FROM ZEROIN IN ALL CASES.
                 ERRCHK DOES NOT PROCESS DIAGNOSTICS IN THESE CASES.
                  1 B IS WITHIN THE REQUESTED TOLERANCE OF A ZERO.
                    THE INTERVAL (B,C) COLLAPSED TO THE REQUESTED
                    TOLERANCE, THE FUNCTION CHANGES SIGN IN (B,C), AND
                    F(X) DECREASED IN MAGNITUDE AS (B,C) COLLAPSED.
                  2 F(B) = 0.  HOWEVER, THE INTERVAL (B,C) MAY NOT HAVE
                    COLLAPSED TO THE REQUESTED TOLERANCE.
                  3 B MAY BE NEAR A SINGULAR POINT OF F(X).
                    THE INTERVAL (B,C) COLLAPSED TO THE REQUESTED
```

```
      TOLERANCE AND THE FUNCTION CHANGES SIGN IN (B,C) BUT
      F(X) INCREASED IN MAGNITUDE AS (B,C) COLLAPSED,I.E.
        ABS(F(B OUT)) .GT. MAX(ABS(F(B IN)),ABS(F(C IN)))
    4 NO CHANGE IN SIGN OF F(X) WAS FOUND ALTHOUGH THE
      INTERVAL (B,C) COLLAPSED TO THE REQUESTED TOLERANCE.
      THE USER MUST EXAMINE THIS CASE AND DECIDE WHETHER
      B IS NEAR A LOCAL MINIMUM OF F(X), OR B IS NEAR A
      ZERO OF EVEN MULTIPLICITY, OR NEITHER OF THESE.
    5 TOO MANY (.GT. 500) FUNCTION EVALUATIONS USED.
```

REFERENCES

1. L F SHAMPINE AND H A WATTS, ZEROIN, A ROOT-SOLVING CODE, SC-TM-70-631, SEPT 1970.
2. T J DEKKER, FINDING A ZERO BY MEANS OF SUCCESSIVE LINEAR INTERPOLATION, *CONSTRUCTIVE ASPECTS OF THE FUNDAMENTAL THEOREM OF ALGEBRA*, EDITED BY B DEJON AND P HENRICI, 1969.

UNLIMITED RELEASE

INITIAL DISTRIBUTION

LAWRENCE LIVERMORE LABORATORY, L310 FRITSCH FREDERICK N
LOS ALAMOS SCIENTIFIC LABORATORY, C6 KAHANER DAVID
8110 BLACKWELL ARLYN N
8111 ABRAMS MARTIN
8111 KEE ROBERT J JR
8111 BRAMLETTE T TAZWELL
8111 CHENOWETH DONALD R
8111 BAKER ALVIN F
8111 GREEN ROBERT M
8111 CATTOLICA ROBERT J
8111 CULL EDWARD T
8113 VOELKER L EUGENE
8113 SCOTT CARLTON A
8113 CALLABRESI MELVIN L
8113 KEILMAN JOHN C
8113 GRANT JOSEPH E
8113 HOYLE CHARLES S
8113 BENEDETTI GERALD A
8113 ROGERS L ALICE
8114 JACOBSON RICHARD S
8114 ZINKE WILLIAM D
8114 JONES ALFRED T
8114 GRISBY SYLVESTER
8114 DELAMETER WILLIAM R
8114 MAMAROS THEODORE C
8115 HARDESTY DONALD R
8115 WITZE PETER O
8115 SMITH JAMES R
8115 COLEMAN HUGH W
8115 MITCHELL REGINALD E
8115 ASHURST WILLIAM T
8156 DEAN DAVID K
8156 DIGHTON LEONARD E
8156 FURNBERG CARLTON M
8156 WRIGHT JAMES B
8156 STARKEY DONALD B
8157 PAGE RODGER J
8157 HANSER HENRY
8157 BOLEN JACK L
8157 DREMALAS JAMES F
8158 KRYVORUKA JOHN K
8158 EVERETT ROGER N
8158 HUMPHREY ROBERT E
8158 LOLL MARVIN B
8158 BOLTON WILLARD R JR
8158 RIVENES ARNOLD S
8158 HILES LEONARD A
8158 DUGGAR JAMES A
8158 ROGERS MICHAEL H
8159 WATKINS LAURENCE M
8159 KELLEY RALPH W
8159 CASHEN JERRY J
8159 TANNER DUNCAN N
8159 BRAY DAVID N
8159 SODERSTRAND MICHAEL A
8159 HULL ARTHUR L
8161 MARTIN GERTRUDE A
8161 SHELEY JAMES W
8161 WARE ROBERT A
8165 WILSON WM G
8165 LEONARD CHARLES F JR
8166 ANDERSON JOHN S
8167 VIEIRA JOSEPH R
8168 MAUCK CHARLES J
8168 JONES JEROME F
8181 TOCKEY ROBERT J
8181 COOLEY NOEL H
8181 SWANSON DONALD H
8183 COFIELD CURTIS H
8183 HUNT CHARLES T

8184 MURPHY LAWRENCE M
8184 MAVIS CLAYTON L
8184 RADOSEVICH LEE G
8184 SCHAFER CLIFFORD T
8184 YOKOMIZO CLIFFORD T
8184 MOORE C W
8185 POTTHOFF C M
8185 AMARAL RONALD J
8185 CHRISMAN SANDRA E
8212 JOINER EMILY A
8214 MCCLURE JANE H
8261 CROW ROBERT E
8265 CUPPS FRANK J
8266 AAS EUGENE A
8310 MEINKEN ROBERT H
8312 KRAMER CAROLYN M
8312 GERMAN RANDALL M
8313 DARGINIS JOSEPH P
8313 BRADSHAW ROBERT W
8313 NICHOLS MONTE C
8313 WEST LLOYD A
8313 BROWN LESLIE A
8313 SEIBEL LEON J
8314 WEST ANTON J
8314 ESTILL WESLEY B
8314 ROBINSON STEVEN L
8314 ODEGARD BENNIE C JR
8320 GOLD THEODORE S ATTN: T. B. COOK, 8000; L. GUTIERREZ, 8100;
8321 BIRNBALM MICHAEL R C. H. DESELM, 8200; B. F. MURPHEY, 8300;
8321 WOODARD JAMES B JR W. C. SCRIVNER, 8400
8321 EICKER PATRICK J
8322 JEFFERSON THOMAS H JR
8322 LEARY PATRICIA L
8322 BASINGER RICHARD C
8322 ROGERS JAMES N
8322 MANSFIELD JUANITA
8322 HUDDLESTON ROBERT E
8322 MILLER GORDON J
8322 HANKINS JOE D
8322 GABRIELSON VERLAN K
8322 NEIGHBORS PAULA K
8322 LEE ROY Y
8322 WILLIAMSON LAWRENCE J
8322 BISSON CHARLES L
8322 JONES HILARY D
8322 BARKER BERTON E
8322 LATHROP JAMES F
8323 BALLOU MARLENE A
8323 CROW ANNE E
8323 BRYAN GAIL M
8323 BRINT HAROLD L
8323 CZAPINSKI ROBERT H
8323 CARSON SHIRLEY Y
8323 ISLER RICHARD E
8323 POLIAK RALPH S
8323 HEIDELBERG S T
8323 BROWN CHARLES V
8323 WHITWORTH FREDDY L
8323 WOOD BURTON L
8323 BARNHOUSE JOHN N JR
8323 SCHUKNECHT ARNOLD G
8323 SPRINGER EUGENE B
8323 DRUMMOND G B JR
8323 WILLIAMS GERALD L
8323 HOUK SHANNON P
8323 THOMAS OWEN R
8323 YANO HESA
8323 GUMMUS RANDALL G
8323 SHORT HAROLD G
8323 BENAFFL JOHN S
8323 NERTON GERALDINE S
8323 PORTOLESE JOSEPH A
8323 CODY TIMOTHY J
8323 ALLEN BOBBY G

8323 CELONI CAROLE L
8323 WAGNER NORMAN R
8323 BRAZIL JOSEPH G
8323 SIMMONS THEODORE M
8323 QUOCK HANLOY
8323 INZERILLA JANET K
8324 BROWN GARRY S
8324 STRANDIN GERALD E
8324 HICKS MEARLE G
8324 BREAZEAL NORMAN L
8332 STORY G COOK
8332 HAVLIK DAVID J
8332 MARION JOHN E
8333 GALLAGHER ROBERT J
8333 ROBINSON CLARENCE W
8333 MUNRO DONALD N
8333 WIEMKEN ALBERT J
8333 NG RAYMOND
8334 MATTERN PETER L
8334 MALINOWSKI MICHAEL E
8334 VITKO JOHN JR
8334 SWANSIGER WILLIAM A
8334 MORSE DANIEL H
8334 PENDLEY MICHAEL H
8335 DUFFEY CHARLES T
8335 WALL WILLIAM R
8335 GILDEA PATRICK D
8340 WIRTH JOHN L
8341 WILSON WILLIAM D
8341 IANNUCCI JOSEPH J
8341 COLL CORNELIUS F III
8341 BASKES MICHAEL I
8341 MELIUS CARL F
8341 KEETON STEWART C
8341 HAGGMARK LEROY G
8342 DELLIN THEODORE A
8342 BARSIS EDWIN H
8342 ZUPPERC ANTHONY C JR
8342 SKOOG CLIFFORD D
8342 RAHN LARRY A
8342 HARTWIG CHARLES M
8342 SCHMIEDER ROBERT W
8342 VASEY STEPHEN J
8342 BARR VERNON C
8342 GRAY STEPHEN C
8344 TICHENOR DANIEL A
8344 HENRY KENNETH W
8344 DOLAN KENNETH W
8344 WILLIS ALEC R
8344 DAHLKE LUTZ W
8345 MARTINELL RONALD E
8346 CHILDERS CARL W
8346 DECARLI CHARLES J
8360 BARHAM JAMES F JR
8362 BROWN MELVIN E
8362 NORRIS HAROLD F JR
8362 CLARK RALPH E
8362 HENDERSON GERALD W
8365 WITEK HENRY M
8366 MITCHELL KENNETH A
8366 SCHOOLER CHARLES C
8410 COZINE RALPH D
8411 CONVERSE LOUISE S
8411 BENTON JOE D
8411 HOLBROOK ELMOND D
8411 MANROW BRITT MARIE
8411 MACMILLAN DOUGLAS C
8412 ROSNOSKI LOUIS J
8412 WHITNEY WILLIAM L
8412 HOUSER RICHARD M
8412 ANGVICK GENE L
8413 DANNENBERG DAVID J
8413 WILSON JACK L
8413 SKINROCD DONALC A

8413 NELSON DENNIS B
8413 GREEN DURWOOD P
8413 WITTMAYER NICHOLAS G
8423 REIS HERMANN L
8423 VON STEEG HERMAN J
8423 FORD ALBERT D JR
8423 JACKSON ROBERT W
8423 SCHMEDDING WILLIAM
8423 AFFELDT BRUCE E
8423 WACKERLY CARL A
8431 MCGUIRE WILBUR V
8432 EICHERT FRED F JR
8432 FINN RICHARD W
8432 BORELLO LAWRENCE A
8432 BYFIELD VERNON E
8432 FERRARIO MICHAEL T
8433 COOK RICHARD S JR
2642 M. R. SCOTT
2642 R. E. JONES
5121 D. E. AMOS
5121 L. F. SHAMPINE
5122 R. J. THOMPSON
8265 TECHNICAL PUBLICATIONS AND ART DIVISION, FOR TIC (2)
3141 / TECHNICAL LIBRARY SYSTEMS AND PROCESSES DIVISION (4)
8265 F.J. CUPPS / TECHNICAL LIBRARY SYSTEMS AND PROCESSES DIVISION
8266 LIBRARY AND SECURITY CLASSIFICATION DIVISION (5)
8322 R. E. HUDDLESTON / SANDIA LABORATORIES (LIVERMORE)
MATHEMATICAL SUBROUTINE LIBRARY (50)