SAND77-8274
Unlimited Release

USER'S GUIDE TO THE SANDIA MATHEMATICAL PROGRAM LIBRARY AT LIVERMORE

(Supersedes SAND76-8209)

T. H. Jefferson

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

October 1977

Sandia Laboratories

SF 2900 Q(7-73)

Issued by Sandia Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE

SAND77-8274
Unlimited Release
Printed October 1977

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

(Supersedes SAND76-8209)

T. H. Jefferson
Applied Mathematics Division 8325
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

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. THE GENERAL CONSULTANT FOR THESE ROUTINES AT SANDIA LIVERMORE IS

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. E. HUDDLESTON	8325
T. H. JEFFERSON	8325

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 ,WHERE THE ROUTINES ARE LISTED ALPHABETICALLY, FOR AN EXPLANATION OF THE CALLING SEQUENCE. USING THIS EXPLANATION, ONE WRITES THE CALL INTO HIS FORTRAN PROGRAM. MOST USERS WILL PROCESS THEIR DECKS WITH A CONTROL CARD SEQUENCE SUCH AS

```
JOB CARD
ACCOUNT CARD
FTN.
ATTACH(MATHFTN,ID=MATHFTN)
LDSET(LIB=MATHFTN)
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 OR T. H. JEFFERSON.

IMSL -

THE INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARY (IMSL) IS AN EXTENSIVE COLLECTION OF HUNDREDS OF ROUTINES OF A MATHEMATICAL OR STATISTICAL NATURE. THE IMSL LIBRARY IS SUPPORTED BY THE IMSL CORPORATION FROM WHICH WE LEASE THE IMSL LIBRARY, AND THE IMSL CONTACT AT LIVERMORE IS T. H. JEFFERSON. YOU MAY OBTAIN A COPY OF THE COMPREHENSIVE IMSL REFERENCE MANUAL BY CONTACTING THE COMPUTING DIVISION --8323 SECRETARY, ARLINE HARREL.

IF A DESIRED CAPABILITY IS NOT AVAILABLE ON THE SANDIA MATHEMATICAL PROGRAM LIBRARY, THEN IMSL SHOULD BE USED. CONTROL CARDS FOR ACCESSING IMSL ARE SIMILAR TO THOSE FOR USING SMPL.

```
JOB CARD
ACCOUNT CARD
ATTACH(IMSLFTN,ID=IMSLFTN)
FTN.
LDSET(LIB=IMSLFTN)
LGO.
```

BMD/BMDP -

THE BIOMEDICAL COMPUTER PROGRAMS ARE A COLLECTION OF APPROXIMATELY 90 MAIN PROGRAMS OF A STATISTICAL NATURE THAT ORIGINATED AT UCLA. THE BMD/BMDP CONSULTANT AT SANDIA LIVERMORE IS
C. J. DECARLI, DIVISION 8346

THJFTN -

THJFTN CONTAINS THE MAJOR SUBROUTINES FOR SOLVING STIFF ORDINARY DIFFERENTIAL EQUATIONS AT SANDIA, LIVERMORE. FURTHER INFORMATION ON THJFTN CAN BE OBTAINED FROM
T. H. JEFFERSON,DIVISION 8325

AELIB -

AELIB IS THE SUBROUTINE LIBRARY OF ATOMIC ENERGY OF CANADA LIMITED. FURTHER INFORMATION ON THE CONTENTS AND USE OF AELIB CAN BE OBTAINED FROM T. H. JEFFERSON .

BLAS -

THE BASIC LINEAR ALGEBRA SUBPROGRAMS (BLAS) ARE A COLLECTION OF 38 FORTRAN-CALLABLE SUBPROGRAMS FOR BASIC OPERATIONS OF NUMERICAL LINEAR ALGEBRA. THE SUBROUTINES ARE AVAILABLE ON SMPL BUT, BECAUSE OF THE LARGE NUMBER OF BLAS ROUTINES, THE BLAS PACKAGE IS NOT EXPLAINED IN DETAIL IN THIS DOCUMENT. USERS INTERESTED IN FURTHER INFORMATION ON THIS COLLECTION OF HIGH SPEED ROUTINES SHOULD CONTACT
T. H. JEFFERSON .
THE BLAS HAVE THE CAPABILITY OF PERFORMING THE FOLLOWING OPERATIONS

- DOT PRODUCT
- CONSTANT TIMES A VECTOR PLUS A VECTOR
- GIVENS ROTATION
- MODIFIED GIVENS ROTATION
- COPY VECTOR X INTO VECTOR Y
- SWAP VECTOR X AND VECTOR Y
- 2-NORM (EUCLIDEAN LENGTH)
- SUM OF ABSOLUTE VALUES
- CONSTANT TIMES A VECTOR
- INDEX OF ELEMENT HAVING MAX ABSOLUTE VALUE

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
SEE IMSL LIBRARY 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.

CHBND
: CALCULATES ERROR BOUNDS FOR COMPUTED EIGENVALUES AND EIGENVECTORS OF COMPLEX HERMITIAN MATRICES. CHBND IS A COMPANION ROUTINE FOR CHAA.

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.

RSBND
: CALCULATES ERROR BOUNDS FOR COMPUTED EIGENVALUES AND EIGENVECTORS OF REAL SYMMETRIC MATRICES. RSBND IS A COMPANION ROUTINE FOR RSAA.

****

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.

****

LINEAR LEAST SQUARES

NNLS

COMPUTES THE SOLUTION TO A LINEAR LEAST SQUARES PROBLEM AX=B SUBJECT TO THE CONSTRAINT THAT EVERY COMPONENT OF THE SOLUTION VECTOR X BE NONNEGATIVE.

SODS

SOLVES (IN THE LEAST SQUARES SENSE) AN OVERDETERMINED SYSTEM OF LINEAR EQUATIONS. THAT IS, IF THERE ARE NEQ EQUATIONS IN NUK UNKNOWNS, THEN NEQ.GE.NUK.

SUDS

SOLVES AN UNDERDETERMINED SYSTEM OF LINEAR EQUATIONS. THAT IS, IF THERE ARE NEQ EQUATIONS IN NUK UNKNOWNS, THEN NEQ.LE.NUK.

SVA

COMPUTES THE SINGULAR VALUE DECOMPOSITION OF A LEAST SQUARES PROBLEM AND PRINTS QUANTITIES RELATING TO THIS DECOMPOSITION TO PROVIDE THE USER WITH INFORMATION HELPFUL IN UNDERSTANDING THE PROBLEM.

SVDRS

COMPUTES THE SINGULAR VALUES OF A MATRIX A, AND ALSO COMPUTES AUXILIARY QUANTITIES USEFUL IN ANALYZING AND SOLVING THE LEAST SQUARES PROBLEM AX=B.

****

NONLINEAR EQUATIONS

QN
SOLVES A SYSTEM OF N NONLINEAR EQUATIONS IN N UNKNOWNS. SOSNLE, BELOW, IS THE NEWER ROUTINE.

SOSNLE
SOLVES A SYSTEM OF N NONLINEAR EQUATIONS IN N UNKNOWNS.

TJMAR1
SEE ENTRY UNDER DATA FITTING.

****

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

COLODE
A COLLOCATION CODE DESIGNED FOR SOLVING STIFF SYSTEMS OF ORDINARY SYSTEMS OF DIFFERENTIAL EQUATIONS. COLODE IS MEANT FOR HIGH ACCURACY PROBLEM, AND IS RELATIVELY EXPENSIVE. SEE THE EASIER TO USE DRIVER STFODE BELOW. ALSO SEE NOTE BELOW ON DRODE.

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 FOR NONSTIFF SYSTEMS.

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

STFODE
A DRIVER FOR THE LOWER LEVEL ROUTINE COLODE.

SUPORT
SOLVES A LINEAR TWO-POINT BOUNDARY VALUE PROBLEM.

**** DRODE **** THE ROUTINE THAT SHOULD BE USED FOR SOLVING STIFF SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS WHEN STFODE (OR COLODE) IS TOO EXPENSIVE IN COMPUTER TIME OR MEMORY. DRODE IS ON THE THJFTN LIBRARY MENTIONED IN SECTION 3. FURTHER INFORMATION IS AVAILABLE FROM T. H. JEFFERSON.

****
PRINTER PLOTTERS

QPPLOT
INTERFACE INTO XPPLOT FOR LINE PRINTER PLOTS OF A SINGLE FUNCTION.

QTPLOT
INTERFACE INTO XPPLOT FOR TERMINAL PLOTS OF A SINGLE FUNCTION.

XPPLOT
PLOTS ONE TO FOUR CURVES ON A SINGLE PRINTER-PLOT, WITH VARIOUS SIZE AND LABELING PARAMETERS SPECIFIED BY THE USER.

****
SPECIAL FUNCTIONS

AIRY
COMPUTES AIRY FUNCTION AI(X), X REAL

BAIRY
COMPUTES AIRY FUNCTION BI(X), X REAL

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.

BETAIC
COMPUTES AN N MEMBER SEQUENCE OF BETA DISTRIBUTIONS
Y(K)=I(A+K-1,B,X), K=1,...,N, A.GT.0 , B.GT.0 , AND 0.LE.X.LE.1
WHERE I(A,B,X) IS THE INCOMPLETE BETA FUNCTION NORMALIZED TO 1.

BETALN
EVALUATES THE NATURAL LOG OF THE COMPLETE BETA FUNCTION,
LN BETA(A,B) , WHERE BETA(A,B) IS DEFINED IN TERMS OF THE
GAMMA FUNCTION BY BETA(A,B)=GAMMA(A)*GAMMA(B)/GAMMA(A+B)

BETBIC
COMPUTES AN N MEMBER SEQUENCE OF BETA DISTRIBUTIONS
Y(K)=I(A,B+K-1,X1), K=1,...,N, A.GT.0 , B.GT.0 , AND
0.LEX.LE.1 WHERE I(A,B,X) IS THE INCOMPLETE BETA FUNCTION
NORMALIZED TO 1. AT X=1.

COSH
HYPERBOLIC COSINE FUNCTION.

DAIRY
COMPUTES THE DERIVATIVE OF THE AIRY FUNCTION AI(X), X REAL.

DBAIRY
COMPUTES THE DERIVATIVE OF THE AIRY FUNCTION BI(X), X REAL.

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.

FCENT
COMPUTES THE CUMULATIVE T DISTRIBUTION.

FCHISQ
COMPUTES THE CUMULATIVE CHI-SQUARE DISTRIBUTION.

FCIRCV
CIRCULAR COVERAGE FUNCTION FOR RADIUS A AND OFFSET D.

FFDIST
CUMULATIVE F DISTRIBUTION

FNORM
CUMULATIVE NORMAL DISTRIBUTION

FNORMB
CUMULATIVE BIVARIATE NORMAL DISTRIBUTION

FXXRHO

BIVARIATE NORMAL FOR TRIPLES (XX,0.,RHO)

GAMFN
GAMMA FUNCTION.

GAMIC
N-MEMBER SEQUENCE OF INCOMPLETE GAMMA FUNCTIONS.

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

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

GAMTL
N-MEMBER SEQUENCE OF COMPLEMENTARY GAMMA FUNCTIONS.

RVNORM
GENERATES NORMALLY (RMU,SIG) DISTRIBUTED RANDOM VARIABLE.

SINH
HYPERBOLIC SINE FUNCTION.

THA
COMPUTES THE T(H,A) INTEGRAL OF OWEN.
T(H,A)=INTEGRAL FROM 0. TO A OF EXP(-H*H*(1+X*X)/2)/(1+X*X).

****
VECTOR OPERATIONS
BLAS-BASIC LINEAR ALGEBRA SUBROUTINES
A PACKAGE OF SUBPROGRAMS FOR COMPUTING SEVERAL
VECTOR OPERATIONS FREQUENTLY ENCOUNTERED
IN LINEAR ALGEBRA. ROUTINES ARE AVAILABLE
IN SINGLE PRECISION (PREFIX S), DOUBLE PRECISION(PREFIX C),
AND COMPLEX (PREFIX C).

IF YOU NEED MORE INFORMATION ON THESE ROUTINES, PLEASE INQUIRE.

BLAS CAPABILITIES ARE AS FOLLOWS.

INNER PRODUCT OF TWO VECTORS.
THE OPERATION Y=AX+Y WITH X,Y VECTORS AND A IS A SCALAR.
GIVENS PLANE ROTATION.
MODIFIED GIVENS TRANSFORMATION.
COPY ONE VECTOR INTO ANOTHER.
SWAP TWO VECTORS.
EUCLIDEAN LENGTH OF TWO VECTORS.
SUM OF MAGNITUDES OF VECTOR COMPONENTS.
SCALAR TIMES A VECTOR.
FIND INDEX OF VECTOR COMPONENT WHICH HAS LARGEST MAGNITUDE

****
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 TJMAR1 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)

AIRY AIRY AIRY AIRY AIRY AIRY AIRY AIRY AIRY

```
FUNCTION AIRY(X,KODE,NZ)

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

REFERENCE SAND-75-0147

ABSTRACT
    AIRY COMPUTES THE AIRY FUNCTION AI(X), X REAL, WITH AN
    OPTION FOR SCALED VALUES FOR X.GE.0. CHEBYSHEV SUMS,
    ASYMPTOTICALLY SCALED FOR SMALL AND LARGE ABS(X), ARE USED ON
    INTERVALS X.LT.0 AND X.GE.0 WITH 0.LE.C.LE.5 AND C.GT.5 WHERE
    C=2*(ABS(X)**1.5)/3. THE INTERVAL X.GE.0 AND 0.LE.C.LE.5 IS
    FURTHER SUBDIVIDED AT X=1.2. THE UNDERFLOW TEST IS C.LE.ELIM
    FOR X.GT.0, WHICH CORRESPONDS TO X.LE.100.033330556172 WITH
    ELIM=667.

DESCRIPTION OF ARGUMENTS

    INPUT
      X      - X.LE.100.03333+ FOR KODE=1, UNRESTRICTED FOR KODE=2
      KODE   - A PARAMETER TO INDICATE THE SCALING OPTION
               KODE=1 RETURNS AIRY=AI(X)        ,
                              X.LE.100.033330556172
               KODE=2 RETURNS AIRY=AI(X)        , X.LT.0
                              AIRY=AI(X)*EXP(C), X.GE.0,
                              WHERE C=2*(X**1.5)/3

    OUTPUT
      AIRY   - AIRY FUNCTION AI(X), SCALED ACCORDING TO KODE
      NZ     - UNDERFLOW INDICATOR
               NZ=0    , NORMAL RETURN, COMPUTATION COMPLETED
               NZ.NE.0, AIRY SET TO ZERO DUE TO UNDERFLOW WITH
                        KODE=1 AND X.GT.100.033330556172

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

```
AVINT    AVINT    AVINT    AVINT       AVINT    AVINT    AVINT    AVINT
         ****************************************
              ****************************
                   ********************
                        **********
```

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

BAIRY BAIRY BAIRY BAIRY BAIRY BAIRY BAIRY BAIRY

FUNCTION BAIRY(X,KODE)

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

REFERENCE SAND-75-0150

ABSTRACT
BAIRY COMPUTES THE AIRY FUNCTION BI(X), X REAL, WITH AN OPTION FOR SCALED VALUES FOR X.GE.0. CHEBYSHEV SUMS, ASYMPTOTICALLY SCALED FOR SMALL AND LARGE ABS(X), ARE USED ON INTERVALS X.LT.0 WITH 0.LE.C.LE.5 AND C.GT.5 AND X.GE.0 WITH 0.LE.C.LE.8 AND C.GT.8 WHERE C=2*(ABS(X)**1.5)/3. THE INTERVAL X.GE.0 AND 0.LE.C.LE.8 IS FURTHER SUBDIVIDED AT X=2.5. THE OVERFLOW TEST IS C.LE.ELIM FOR X.GE.0, WHICH CORRESPONDS TO X.LE.100.033330556172 WITH ELIM=667.

DESCRIPTION OF ARGUMENTS

INPUT
X - X.LE.100.03333+ FOR KODE=1, UNRESTRICTED FOR KODE=2
KODE - A PARAMETER TO INDICATE THE SCALING OPTION
KODE=1 RETURNS BAIRY=BI(X) , X.LE.100.033330556172
KODE=2 RETURNS BAIRY=BI(X) , X.LT.0
BAIRY=BI(X)*EXP(-C), X.GE.0,
WHERE C=2*(X**1.5)/3

OUTPUT
BAIRY - AIRY FUNCTION BI(X), SCALED ACCORDING TO KODE

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

BESI BESI BESI BESI BESI BESI BESI BESI BESI

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

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

FUNCTION BESI01(X,NU,KODE)
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.

DESCRIPTION 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 TO 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 FOR KODE=1 - A FATAL ERROR
```

BESJ BESJ BESJ BESJ BESJ BESJ BESJ BESJ BESJ

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

BETAIC BETAIC BETAIC BETAIC BETAIC BETAIC BETAIC

```
    SUBROUTINE BETAIC(X,OMX,A,B,N,Y,NZ)

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

    REFERENCE SC-DR-69 591

    ABSTRACT
        BETAIC COMPUTES AN N MEMBER SEQUENCE OF BETA DISTRIBUTIONS
        Y(K)=I(A+K-1,B,X), K=1,...,N , A.GT.0 , B.GT.0 , AND
        0.LE.X.LE.1, WHERE I(A,B,X) IS THE INCOMPLETE BETA FUNCTION
        NORMALIZED TO 1. AT X=1. THE RELATION OF THE INCOMPLETE BETA
        FUNCTION TO THE GAUSS HYPERGEOMETRIC FUNCTION IS USED OVER
        VARIOUS PARAMETER RANGES WITH SERIES OR ASYMPTOTIC EXPRESSIONS
        USED FOR EVALUATION STARTING AT A+N-1 AND BO.GT.0. WITH
        BO=B-INTEGER PART OF B OR 1. THEN A COMBINATION OF FORWARD
```

RECURSION ON THE PARAMETER TO RAISE BO TO B FOLLOWED BY
BACKWARD RECURSION ON THE FIRST PARAMETER TO DECREASE A+N-1 TO
A GETS THE REQUIRED SEQUENCE. I(A,B,X) SATISFIES A TWO-TERM
RELATION IN BOTH PARAMETERS WHERE ADDITIONS CAN BE USED
EXCLUSIVELY TO RETAIN SIGNIFICANT DIGITS. BOTH X AND OMX=1.-X
ARE ENTERED IN THE CALL LIST TO AVOID LOSSES OF SIGNIFICANCE
IN OMX WHEN AN ANALYTICAL EXPRESSION IS AVAILABLE(SEE THE F
AND T DISTRIBUTIONS). BETAIC USES HYPGEO AND BETALN.

DESCRIPTION OF ARGUMENTS

INPUT
- X - ARGUMENT, 0.LE.X.LE.1.
- OMX - 1.-X
- A - START VALUE OF FIRST PARAMETER, A.GT.0.
- B - VALUE OF SECOND PARAMETER, B.GT.0.
- N - NUMBER OF BETA FUNCTIONS IN THE SEQUENCE, N.GE.1

OUTPUT
- Y - A VECTOR WHOSE FIRST N COMPONENTS CONTAIN
  Y(K)=I(A+K-1,B,X), K=1,...,N.
- NZ - UNDERFLOW FLAG
  NZ.EQ.0, A NORMAL RETURN.
  NZ.NE.0, UNDERFLOW, Y(K)=0.0, K=N-NZ+1,N RETURNED

ERROR CONDITIONS
IMPROPER INPUT - A FATAL ERROR
UNDERFLOW - A NON-FATAL ERROR.

BETALN BETALN BETALN BETALN BETALN BETALN BETALN

FUNCTION BETALN(A,B)

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

REFERENCE SC-DR-69 591 AND SAND-75-0152

ABSTRACT
FUNCTION BETALN COMPUTES THE NATURAL LOG OF THE COMPLETE BETA
FUNCTION, LN BETA(A,B), WHERE BETA(A,B) IS DEFINED IN TERMS OF
THE GAMMA FUNCTION BY BETA(A,B)=GAMMA(A)*GAMMA(B)/GAMMA(A+B).
THE NATURAL LOG OF THE GAMMA FUNCTION IS COMPUTED WITH THE
LOGARITHMIC TERMS COMBINED ANALYTICALLY TO MINIMIZE LOGARITHM
EVALUATIONS. A RATIONAL CHEBYSHEV APPROXIMATION ON (8,1000)
AND THE ASYMPTOTIC EXPANSION FOR X.GT.1000 COMPLETES THE
CALCULATION OF THE NON-LOGARITHMIC TERMS. BACKWARD RECURSION
ON GAMMA(X-1)=GAMMA(X)/(X-1) AND THE LOGARITHM OF THE RESULT
SUFFICES FOR X.LT.8.

DESCRIPTION OF ARGUMENTS

INPUT
- A - ARGUMENT, A.GT.0.
- B - ARGUMENT, B.GT.0.

OUTPUT
- BETALN - VALUE FOR LN BETA(A,B)

ERROR CONDITIONS
INPUT PARAMETER NON-POSITIVE - A FATAL ERROR

BETBIC BETBIC BETBIC BETBIC BETBIC BETBIC BETBIC

SUBROUTINE BETBIC(X,OMX,A,B,N,Y,NZ)

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

REFERENCE SC-DR-69 591

ABSTRACT

BETBIC COMPUTES AN N MEMBER SEQUENCE OF BETA DISTRIBUTIONS Y(K)=I(A,B+K-1,X), K=1,...,N, A.GT.0, B.GT.0, AND 0.LE.X.LE.1, WHERE I(A,B,X) IS THE INCOMPLETE BETA FUNCTION NORMALIZED TO 1. AT X=1. THE RELATION OF THE INCOMPLETE BETA FUNCTION TO THE GAUSS HYPERGEOMETRIC FUNCTION IS USED OVER VARIOUS PARAMETER RANGES WITH SERIES OR ASYMPTOTIC EXPRESSIONS USED FOR EVALUATION STARTING AT AA=A+M, M.GE.0 AND BO.GT.0, BO=B-INTEGER PART OF B OR 1. THEN A COMBINATION OF BACKWARD RECURSION ON AA FOLLOWED BY FORWARD RECURSION ON BB=BO+K GETS CORRECT ARGUMENTS A AND B OR A AND B+K-1, K=1,...,N FOR THE REQUIRED SEQUENCE Y(K). I(A,B,X) SATISFIES A TWO-TERM RELATION IN BOTH PARAMETERS WHERE ADDITIONS CAN BE USED EXCLUSIVELY TO RETAIN SIGNIFICANT DIGITS. BOTH X AND OMX=1.-X ARE ENTERED IN THE CALL LIST TO AVOID LOSSES OF SIGNIFICANCE IN OMX WHEN AN ANALYTICAL EXPRESSION IS AVAILABLE(SEE THE F AND T DISTRIBUTIONS). BETBIC USES HYPGEO AND BETALN.

DESCRIPTION OF ARGUMENTS

INPUT

X	-	ARGUMENT, 0.LE.X.LE.1.
OMX	-	1.-X
A	-	VALUE OF FIRST PARAMETER, A.GT.0.
B	-	START VALUE OF SECOND PARAMETER, B.GT.0.
N	-	NUMBER OF BETA FUNCTIONS IN THE SEQUENCE, N.GE.1

OUTPUT

Y	-	A VECTOR WHOSE FIRST N COMPONENTS CONTAIN Y(K)=I(A,B+K-1,X), K=1,...,N.
NZ	-	UNDERFLOW FLAG NZ.EQ.0, A NORMAL RETURN. NZ.NE.0, UNDERFLOW, Y(K)=0.0, K=1,NZ RETURNED.

ERROR CONDITIONS

IMPROPER INPUT - A FATAL ERROR
UNDERFLOW - A NON-FATAL ERROR.

CAXBI CAXBI CAXBI CAXBI CAXBI CAXBI CAXBI CAXBI

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

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 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 EIGENSYSTEMS 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.
```

```
CHBND    CHBND    CHBND    CHBND       CHBND    CHBND    CHBND    CHBND
            ************************************
               *********************************
                  ******************
                     **********
   SUBROUTINE CHBND(NDIM,N,AR,AI,EV,VECR,VECI,EVIMP,BNDS)

   ABSTRACT

   THIS SUBROUTINE CALCULATES RAYLEIGH QUOTIENT CORRECTIONS FOR
   THE COMPUTED EIGENVALUES OF A COMPLEX HERMITIAN MATRIX AND
   UPPER BOUNDS ON THE ABSOLUTE ERROR OF THE COMPUTED EIGENSYSTEM.
   REASONABLE BOUNDS FOR THE EIGENVECTORS ARE POSSIBLE ONLY
   WHEN THE EIGENVALUES ARE WELL SEPARATED.  WHEN THIS IS NOT THE
   CASE, NO BOUND IS CALCULATED.

   TO COMPUTE ERROR BOUNDS FOR THE EIGENSYSTEMS OF REAL SYMMETRIC
   MATRICES, SEE SUBROUTINE RSBND.  SIMILAR BOUNDS FOR REAL NON-
   SYMMETRIC AND COMPLEX NON-HERMITIAN MATRICES ARE NOT POSSIBLE.

   DESCRIPTION OF ARGUMENTS
     ON INPUT
       NDIM    MUST BE ROW DIMENSION OF AR,AI,VECR,VECI,BNDS
               IN THE CALLING PROGRAM DIMENSION STATEMENT.
       N       IS THE ORDER OF THE MATRIX.  1 .LE. N .LE. NDIM.
       AR,AI   MUST CONTAIN IN THE LEADING N BY N SUBARRAYS
               THE REAL AND IMAGINARY PARTS RESPECTIVELY OF THE
               COMPLEX HERMITIAN MATRIX.  ONLY THE DIAGONALS
               AND LOWER TRIANGLES NEED BE DEFINED.
       EV      MUST CONTAIN IN THE FIRST N ELEMENTS THE REAL
               EIGENVALUES AS COMPUTED, SAY, BY CHAA.
       VECR,VECI  MUST CONTAIN IN THE LEADING N BY N SUBARRAYS
                  THE REAL AND IMAGINARY ORTHONORMAL EIGENVECTORS
                  AS COMPUTED, SAY, BY CHAA.  THE J-TH COLUMNS
                  OF VECR,VECI MUST CORRESPOND TO THE J-TH
                  ELEMENT OF EV.
     ON OUTPUT
       EVIMP   CONTAINS DOUBLE PRECISION IMPROVED EIGENVALUES
               (RAYLEIGH QUOTIENTS) IN THE SAME ORDER AS EV.
       BNDS    CONTAINS UPPER BOUNDS ON THE ABSOLUTE ERRORS OF
               THE COMPUTED EIGENSYSTEM
               BNDS(J,1) -- UPPER BOUND ON ABSOLUTE ERROR IN
```

EVIMP(J).

BNDS(J,2) -- UPPER BOUND ON L-2 NORM OF ERROR IN J-TH COMPUTED EIGENVECTOR. THIS QUANTITY IS SET TO -1.0 WHEN THE EIGENVALUES ARE TOO CLOSE TO PERMIT A REASONABLE BOUND

BNDS(J,3) -- L-2 NORM OF RESIDUAL ASSOCIATED WITH EV(J) AND VEC(*,J).

CNAA CNAA CNAA CNAA CNAA CNAA CNAA CNAA CNAA

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

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 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 DESTROYED.
```

CNAN CNAN CNAN CNAN CNAN CNAN CNAN CNAN CNAN

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

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

```
COLODE    COLODE    COLODE    COLODE          COLODE    COLODE    COLODE
              ****************************************
                   ***********************************
                        ******************************
                             ************************
                                  ******************
     SUBROUTINE COLODE(F,JACSUB,CONSTJ,NEQN,Y,T,TFIN,EPSREL,EPSABS,
    1                  KFLAG,NCOLPT,IQUAD,IRETRN,IWORK,WORK,NWRKD)

     WRITTEN BY B. L. HULME AND S. L. DANIEL.

  ABSTRACT.
          COLODE IS A COLLOCATION CODE FOR SYSTEMS OF ORDINARY
     DIFFERENTIAL EQUATIONS OF THE FORM
               DY/DT=F(T,Y(1),Y(2),...,Y(NEQN)).
     IT IS RECOMMENDED PRIMARILY FOR SOLVING STIFF EQUATIONS, SINCE THE
     IMPLICIT, ONE-STEP COLLOCATION METHODS ARE A-STABLE, HIGHLY
     ACCURATE, AND RELATIVELY EXPENSIVE.  COLODE WILL ALSO SOLVE
     NON-STIFF PROBLEMS, BUT PROBABLY LESS EFFICIENTLY THAN, SAY, A
     GOOD VARIABLE ORDER ADAMS CODE.
          THE SUBROUTINE IS DESIGNED TO INTEGRATE FROM T TO TFIN WITH
     RETURNS TO THE USER EITHER AFTER EACH PAIR OF STEPS, OR ONLY AT
     T=TFIN.  IF IRETRN=.T., THEN EACH CALL OF COLODE RESULTS IN
     INTEGRATION OVER TWO STEPS OF LENGTH H, A RETURN TO THE USER WITH
     T=T+2H, Y=Y(T), WORK(4,...,3+NEQN)=Y(T-H), AND ALL PARAMETERS SET
     FOR CONTINUATION.  THE USER NEEDS ONLY TO CALL COLODE REPEATEDLY
     UNTIL KFLAG=2, INDICATING THAT T=TFIN. IF IRETRN=.F., THEN EACH
     CALL OF COLODE RESULTS IN INTEGRATION OVER (T,TFIN), A RETURN TO
     THE USER WITH T=TFIN, Y=Y(T), AND ALL PARAMETERS SET FOR
     CONTINUATION.  THE STEP SIZE H IS INITIALIZED BY COLODE AND VARIED
     AT EACH PAIR OF STEPS IN AN ATTEMPT TO KEEP THE ESTIMATED LOCAL
     ERROR PER STEP SMALLER THAN THE USER SPECIFIED TOLERANCE.
          TWO DOZEN DIFFERENT COLLOCATION METHODS ARE AVAILABLE TO THE
```

USER. THEY CORRESPOND TO COLLOCATION AT THE ABSCISSAS OF DIFFERENT QUADRATURE FORMULAS MAPPED ONTO EACH STEP. THE METHOD USED IS CHOSEN BY EITHER THE USER OR COLODE FROM AMONG THE GAUSS-LEGENDRE OR RADAU (RIGHT END POINT) QUADRATURES USING ONE TO TWELVE POINTS. IQUAD INDICATES THE TYPE OF QUADRATURE POINTS, AND NCOLPT INDICATES THE NUMBER. THE GAUSS-LEGENDRE METHODS ARE A-STABLE AND ARE OF ORDER 2*NCOLPT, WHILE THE RADAU METHODS ARE STRONGLY A-STABLE AND ARE OF ORDER 2*NCOLPT-1. (SEE REFERENCE.) COLODE MAKES NO ATTEMPT TO VARY THE METHOD DURING INTEGRATION.

THE ITERATIVE SOLUTION OF THE GENERALLY NONLINEAR COLLOCATION EQUATIONS REQUIRES SOME KNOWLEDGE OF THE JACOBIAN OF F WITH RESPECT TO Y. THE USER MAY EITHER SUPPLY A JACOBIAN SUBROUTINE DFDY, OR ELSE LET COLODE APPROXIMATE THE MATRIX BY DIFFERENCES. THE USER*S CHOICE IS INDICATED BY JACSUB. IN ADDITION, CONSTJ INDICATES WHETHER OR NOT THE JACOBIAN IS CONSTANT, SINCE CONSTANT JACOBIAN PROBLEMS CAN BE TREATED MORE EFFICIENTLY. FOR SUCH PROBLEMS THERE IS NO REASON FOR THE USER TO WRITE A JACOBIAN SUBROUTINE, SINCE THE PROBLEM IS NECESSARILY LINEAR AND THE FIRST DIFFERENCES ARE EXACT, EXCEPT FOR ROUNDOFF.

REFERENCE.

B. L. HULME,**DISCRETE GALERKIN AND RELATED ONE-STEP METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS,**MATH.COMP.,V.26,1972,PP.881-891.

SUBROUTINES.

COLODE CALLS TWOSTP, WHICH CALLS F AND POSSIBLY DFDY. COLODE ALSO CONTAINS THREE CALLS,AND TWOSTP ONE CALL,OF ERRCHK,A LOCAL LIBRARY SUBROUTINE FOR PROCESSING ERROR MESSAGES.

DUMMY ARGUMENTS.

F	- THE NAME OF THE DERIVATIVE SUBROUTINE.
JACSUB	- THE JACOBIAN SUBROUTINE INDICATOR.
CONSTJ	- THE CONSTANT JACOBIAN INDICATOR.
NEQN	- THE NUMBER OF EQUATIONS IN THE SYSTEM.
Y(*)	- THE APPROXIMATE SOLUTION VECTOR AT T.
T	- THE INDEPENDENT VARIABLE.
TFIN	- THE FINAL POINT OF THE INTEGRATION INTERVAL.
EPSREL	- THE RELATIVE LOCAL ERROR TOLERANCE PER STEP.
EPSABS	- THE ABSOLUTE LOCAL ERROR TOLERANCE PER STEP.
KFLAG	- THE FLAG FOR COMMUNICATION BETWEEN COLODE AND USER.
NCOLPT	- THE NUMBER OF COLLOCATION POINTS PER STEP.
IQUAD	- THE TYPE OF QUADRATURE ABSCISSAS USED FOR COLLOCATION.
IRETRN	- THE INTERMEDIATE RETURN INDICATOR.
IWORK(*)	- THE INTEGER WORK VECTOR.
WORK(*)	- THE REAL WORK VECTOR.
NWRKD	- THE DIMENSION OF WORK.

ACTUAL ARGUMENTS.

THE ONLY ARGUMENTS WHICH MAY BE CONSTANTS OR EXPRESSIONS ARE THOSE CORRESPONDING TO THE DUMMY ARGUMENTS JACSUB, CONSTJ, NEQN, TFIN, IRETRN, AND NWRKD. ALL OTHERS MUST BE NAMES. THE ARGUMENTS JACSUB, CONSTJ, AND IRETRN ARE TYPE LOGICAL. IF VARIABLES ARE USED FOR THESE PARAMETERS THEY MUST BE DECLARED AS TYPE LOGICAL.

INPUT.

F	- NAME OF SUBROUTINE F(T,Y,YP) TO STORE DY(J)/DT IN YP(J). THIS NAME MUST BE DECLARED IN AN EXTERNAL STATEMENT.
JACSUB	= .T., IF THE USER SUPPLIES A JACOBIAN SUBROUTINE DFDY(T,Y,PD,NPDIM) TO STORE THE PARTIAL DERIVATIVES OF F(I) WITH RESPECT TO Y(J) INTO PD(I,J), FOR 1.LE.I,J.LE.NEQN. PD MUST BE DIMENSIONED PD(NPDIM,NPDIM), REGARDLESS OF THE VALUE OF NEQN.
	= .F., OTHERWISE.
CONSTJ	= .T., IF THE JACOBIAN IS CONSTANT,
	= .F., OTHERWISE.
NEQN	= THE NUMBER OF EQUATIONS.
Y(*)	= THE INITIAL VALUES.
T	= THE INITIAL POINT.
TFIN	= THE FINAL POINT.
EPSREL	= THE RELATIVE LOCAL ERROR TOLERANCE PER STEP.
EPSABS	= THE ABSOLUTE LOCAL ERROR TOLERANCE PER STEP. PRIOR TO ACCEPTANCE OF A PAIR OF STEPS, COLODE TESTS

```
                THE ESTIMATED LOCAL ERROR IN Y(J) RELATIVE TO
                      EPSREL*ABS(Y(J))+EPSABS.
   KFLAG      = 1, START, OR RESTART WITH SOME CHANGE IN THE
                   ARGUMENTS OTHER THAN TFIN.
              = 2, CONTINUE WITH ARGUMENTS UNCHANGED EXCEPT FOR TFIN.
              = 3, CONTINUE WITH ARGUMENTS UNCHANGED.
              = 4, CONTINUE WITH ARGUMENTS UNCHANGED.
              = 5, RESTART WITH ARGUMENTS UNCHANGED EXCEPT FOR TFIN.
   NCOLPT     = 0,        IF COLODE IS TO SELECT NCOLPT AND IQUAD,
              = 1,...,12, IF THE USER SELECTS NCOLPT.
                      GENERALLY, USE BETWEEN 2 AND 7 POINTS.
   IQUAD      = 0, IF COLODE IS TO SELECT IQUAD,
              = 1, FOR GAUSS-LEGENDRE POINTS,
              = 2, FOR RADAU (RIGHT-END) POINTS.
                   GENERALLY, GAUSS-LEGENDRE POINTS ARE BETTER FOR
                   NON-STIFF PROBLEMS.  A SIMPLE RULE FOR STIFF
                   PROBLEMS IS TO USE GAUSS-LEGENDRE POINTS WITH
                   ABSOLUTE ERROR TOLERANCES AND RADAU POINTS WITH
                   RELATIVE ERROR TOLERANCES.  IF THE PROBLEM SEEMS
                   VERY STIFF, USE RADAU POINTS.
   IRETRN     = .T., IF COLODE IS TO RETURN AFTER EACH PAIR OF STEPS,
              = .F., IF COLODE IS TO RETURN ONLY AT T=TFIN.
   IWORK(*)   - THE NAME OF THE INTEGER WORK VECTOR.  THE DIMENSION OF
                IWORK MUST BE .GE. NEQN*NCOLPT.
   WORK(*)    - THE NAME OF THE REAL WORK VECTOR.
   NWRKD      = THE DIMENSION OF WORK.
                NWRKD.GE.(NEQN*NEQN+2)*NCOLPT*NCOLPT
                        +(NEQN*NEQN+5*NEQN+2)*NCOLPT
                                +6*NEQN+3.
                IN ORDER TO DIMENSION IWORK AND WORK WHEN COLODE
                IS TO SELECT NCOLPT, THE USER MAY SIMPLY TAKE
                NCOLPT TO BE 7, THE LARGEST VALUE COLODE WILL
                EVER SELECT.  TO MINIMIZE STORAGE, HOWEVER, THE
                ACTUAL VALUE OF NCOLPT SHOULD BE USED.
                SEE DESCRIPTION OF OUTPUT VALUE OF NCOLPT, BELOW.

OUTPUT.
   Y(*)       = THE APPROXIMATE SOLUTION AT T.
   T          = THE END OF THE LAST SUCCESSFUL PAIR OF STEPS.
   EPSREL     = ABSOLUTE VALUE OF THE INITIAL EPSREL NORMALLY.  IF
                KFLAG=4 AND EPSREL.NE.0., THEN EPSREL=1.42E-14.
   EPSABS     = ABSOLUTE VALUE OF THE INITIAL EPSABS NORMALLY.  IF
                KFLAG=4 AND EPSREL.EQ.0., THEN EPSABS=1.42E-14*NORM Y.
   KFLAG      = 2, INTEGRATION WAS COMPLETED TO T.EQ.TFIN.
              = 3, TWO STEPS WERE TAKEN, AND T.NE.TFIN.
              = 4, EITHER EPSREL OR EPSABS WAS TOO SMALL AND HAS BEEN
                   INCREASED TO TWO UNITS OF ROUNDOFF.  NO STEPS WERE
                   TAKEN, AND COLODE IS SET FOR CONTINUATION.
              = 5, H IS MORE THAN FIVE ORDERS OF MAGNITUDE SMALLER
                   THAN ITS VALUE AT THE LAST START OR RESTART.  NO
                   DIFFICULTY IS EXPECTED.  THE USER MAY RETURN TO
                   COLODE, PERHAPS WITH A NEW TFIN, AND THE
                   INTEGRATION WILL PROCEED.
              = 6, CONVERGENCE AND/OR LOCAL ERROR FAILURES CONTINUED
                   TO OCCUR EVEN AFTER H WAS REDUCED FIVE ORDERS OF
                   MAGNITUDE BELOW ITS VALUE AT THE LAST START OR
                   RESTART.  THE USER MUST CHANGE THE METHOD AND/OR
                   THE TOLERANCE BEFORE RETURNING TO COLODE.  IF
                   NCOLPT IS INCREASED, THE USER MUST BE SURE NWRKD IS
                   LARGE ENOUGH.
              = 7, CANNOT PROCEED BECAUSE OF EITHER ILLEGAL INPUT OR
                   SINGULAR MATRICES.  FOR DETAILS SEE THE DIAGNOSTIC
                   MESSAGE PRINTED BY THE SUBROUTINE ERRCHK.
                   NOTICE THAT THE OUTPUT VALUES KFLAG=2,3,4,5 HAVE
                   BEEN COORDINATED WITH THE INPUT VALUES.  AFTER
                   INITIALIZING KFLAG=1, THE USER NEED NOT RESET
                   KFLAG UNTIL EITHER HE WISHES TO START A NEW
                   INTEGRATION OR HE ENCOUNTERS KFLAG=6,7.
   NCOLPT     = THE NUMBER OF COLLOCATION POINTS USED.
                WHEN THE INPUT VALUE NCOLPT.EQ.0, COLODE RESETS
                NCOLPT AS FOLLOWS.  FOR EPSREL.EQ.0,
                  NCOLPT=MIN0(7,MAX0(2,INT((-ALOG10(EPSABS/A))*.5))),
                WHERE A IS THE MAGNITUDE OF THE LARGEST ELEMENT IN Y
                (OR 1. IF Y IS ZERO), AND FOR EPSREL.NE.0,
```

```
                NCOLPT=MIN0(7,MAX0(3,INT((14.-ALOG10(EPSREL))*.25)).
              ALSO, WHEN NWRKD FAILS TO SATISFY THE ABOVE
              INEQUALITY, NCOLPT IS REDUCED SO THAT IT IS SATISFIED.
    IQUAD     = 1, IF GAUSS-LEGENDRE POINTS WERE USED,
              = 2, IF RADAU POINTS WERE USED.
              IQUAD IS RESET BY COLODE IF NCOLPT.EQ.0 OR IQUAD.EQ.0.
              FOR EPSREL.EQ.0, IQUAD=1, AND FOR EPSREL.NE.0, IQUAD=2
    IWORK(*)  = THE INTEGER WORK VECTOR.
    WORK(*)   = THE REAL WORK VECTOR.

  SPECIAL AUXILIARY OUTPUT.
      CERTAIN USERS MAY WANT SOME OF THE AUXILIARY OUTPUT VARIABLES
    WORK(1)   = H       - STEP SIZE.
    WORK(2)   = ELE     - RATIO OF ESTIMATED LOCAL ERROR TO TOLERANCE.
    WORK(3)   = HMIN    - 1.E-5 TIMES H AT THE LAST START OR RESTART.
    WORK(4)   = YTMH(*) - APPROXIMATE SOLUTION AT T-H, IF KFLAG=2,3.
```

```
COSH    COSH    COSH    COSH    COSH       COSH    COSH    COSH    COSH
              ****************************************
                   ******************************
                        ********************
                             **********
```

```
    FUNCTION COSH(X)
    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)
    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
```

```
DAIRY    DAIRY    DAIRY    DAIRY      DAIRY    DAIRY    DAIRY    DAIRY
            ******************************************
                ********************************
                    ********************
                          **********
      FUNCTION DAIRY(X,KODE,NZ)

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

      REFERENCE SAND-75-0147

      ABSTRACT
          DAIRY COMPUTES THE DERIVATIVE OF THE AIRY FUNCTION AI(X), X
          REAL, WITH AN CPTION FOR SCALED VALUES FOR X.GE.0. CHEBYSHEV
          SUMS, ASYMPTOTICALLY SCALED FOR SMALL AND LARGE ABS(X), ARE
          USED ON INTERVALS X.LT.0 AND X.GE.0 WITH 0.LE.C.LE.5 AND
          C.GT.5 WHERE C=2*(ABS(X)**1.5)/3. THE INTERVAL X.GE.0 AND
          0.LE.C.LE.5 IS FURTHER SUBDIVIDED AT X=1.2. THE UNDERFLOW TEST
          IS C.LE.ELIM FOR X.GT.0, WHICH CORRESPONDS TO
          X.LE.100.033330556172 WITH ELIM=667.

      DESCRIPTION CF ARGUMENTS

          INPUT
            X       - X.LE.100.03333+ FOR KODE=1, UNRESTRICTED FOR KODE=2
            KODE    - A PARAMETER TO INDICATE THE SCALING OPTION
                      KODE=1 RETURNS DAIRY=DERIVATIVE OF AI(X),
                                     X.LE.100.033330556172
                      KODE=2 RETURNS DAIRY=DERIVATIVE OF AI(X), X.LT.0
                                     DAIRY=(DERIVATIVE OF AI(X))*EXP(C),
                                     X.GE.0 WHERE C=2*(X**1.5)/3

          OUTPUT
            DAIRY   - DERIVATIVE OF AI(X), SCALED ACCORDING TO KCDE
            NZ      - UNDERFLOW INDICATOR
                      NZ=C    , NORMAL RETURN, COMPUTATICN COMPLETED
                      NZ.NE.0, DAIRY SET TO ZERO DUE TO UNDERFLOW WITH
                               KODE=1 AND X.GT.100.03333055172

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

DBAIRY DBAIRY DBAIRY DBAIRY DBAIRY DBAIRY DBAIRY

FUNCTION DBAIRY(X,KODE)

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

REFERENCE SAND-75-0150

ABSTRACT

DBAIRY COMPUTES THE DERIVATIVE OF THE AIRY FUNCTION BI(X), X REAL, WITH AN OPTION FOR SCALED VALUES FOR X.GE.0. CHEBYSHEV SUMS, ASYMPTOTICALLY SCALED FOR SMALL AND LARGE ABS(X), ARE USED ON INTERVALS X.LT.0 WITH 0.LE.C.LE.5 AND C.GT.5 AND X.GE.0 WITH 0.LE.C.LE.8 AND C.GT.8 WHERE C=2*(ABS(X)**1.5)/3. THE INTERVAL X.GE.0 AND 0.LE.C.LE.8 IS FURTHER SUBDIVIDED AT X=2.5. THE OVERFLOW TEST IS C.LE.ELIM FOR X.GE.0, WHICH CORRESPONDS TO X.LE.100.033330556172 WITH ELIM=667.

DESCRIPTION OF ARGUMENTS

INPUT

X - X.LE.100.03333+ FOR KODE=1, UNRESTRICTED FOR KODE=2

KODE - A PARAMETER TO INDICATE THE SCALING OPTION
KODE=1 RETURNS DBAIRY=DERIVATIVE OF BI(X), X.LE.100.033330556172
KODE=2 RETURNS DBAIRY=DERIVATIVE OF BI(X), X.LT.0, DBAIRY=(DERIVATIVE OF BI(X))*EXP(-C) X.GE.0 WHERE C=2*(X**1.5)/3

OUTPUT

DBAIRY - DERIVATIVE OF BI(X), SCALED ACCORDING TO KODE

ERROR CONDITIONS

IMPROPER INPUT ARGUMENTS - A FATAL ERROR
OVERFLOW WITH KODE=1 - A FATAL ERROR

ERF ERF ERF ERF ERF ERF ERF ERF ERF ERF

FUNCTION ERF(XX)
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)
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 APPROXIMATIONS. 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)
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)
     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)
     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)
```

```
FCENT     FCENT     FCENT     FCENT          FCENT     FCENT     FCENT     FCENT
                ***************************************
                    *******************************
                        ***********************
                            ***********

      FUNCTION FCENT(X,FN,KODE,NZ)
      CDC 6600 ROUTINE

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

      REFERENCE SC-DR-73 0333

      ABSTRACT
          FCENT COMPUTES THE CUMULATIVE T DISTRIBUTION F(X) OR ITS
          COMPLEMENT 1.-F(X) FOR A RANDOM VARIABLE T=U/SQRT(V/N) WHERE
          U IS NORMAL(0,1) AND V IS CHI-SQUARE(N). THE RELATION OF
          F(X) TO THE INCOMPLETE BETA FUNCTION (NORMALIZED TO 1. AT X=1)
          IS USED FOR THE COMPUTATION IN SUBROUTINE BETAIC.

      DESCRIPTION OF ARGUMENTS

          INPUT
            X       - ARGUMENT, UNRESTRICTED,
            FN      - DEGREES OF FREEDOM OF THE DENOMINATOR CHI-SQUARE(N)
                      RANDOM VARIABLE, FN.GE.1.
            KODE    - A SELECTION PARAMETER
                      KODE=1 RETURNS FCENT=F(X)
                      KODE=2 RETURNS FCENT=1.-F(X).
          OUTPUT
            FCENT   - F(X) OR 1.-F(X) DEPENDING ON KODE,
            NZ      - UNDERFLOW FLAG
                      NZ=0, A NORMAL RETURN
                      NZ=1, UNDERFLOW, FCENT=0.0 RETURNED.

      ERROR CONDITIONS
          IMPROPER INPUT - A FATAL ERROR
          UNDERFLOW - A NON-FATAL ERROR.
```

```
FCHISQ    FCHISQ    FCHISQ    FCHISQ         FCHISQ    FCHISQ    FCHISQ
                ***************************************
                    *******************************
                        ***********************
                            ***********

      FUNCTION FCHISQ(X,FN,KODE,NZ)
      CDC 6600 ROUTINE

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

      REFERENCE SC-DR-73 0333

      ABSTRACT
          FCHISQ COMPUTES THE CUMULATIVE CHI-SQUARE DISTRIBUTION F(X) OR
          ITS COMPLEMENT 1.-F(X) WITH N DEGREES OF FREEDOM, X.GE.0 AND
          N.GE.1. THE RELATION OF F(X) TO THE INCOMPLETE GAMMA FUNCTION
          (NORMALIZED TO 1. AT X=INFINITY) IS USED FOR THE COMPUTATION
          IN SUBROUTINES GAMIC AND GAMTL, WITH THE CHANGE FROM ONE
          SUBROUTINE TO THE OTHER AT ARGUMENT=THE PARAMETER OF THE GAMMA
          FUNCTIONS. THIS CHANGE NOT ONLY ENSURES SIGNIFICANT DIGITS FOR
          BOTH F(X) AND 1.-F(X), BUT MAKES THE COMPUTATION AS FAST AS
          POSSIBLE IN EACH SUBROUTINE. A PARAMETER REL=1.E-8 IS SET IN
          THE PROGRAM FOR 8 SIGNIFICANT DIGITS. REL CAN BE SET AS LOW AS
          1.E-12 FOR 12 SIGNIFICANT DIGITS.
          FCHISQ USES GAMIC, GAMTL, GAMLN.

      DESCRIPTION OF ARGUMENTS

          INPUT
```

```
        X       - ARGUMENT, X.GE.0.0
        FN      - DEGREES OF FREEDOM OF THE CHI-SQUARE(N)
                  DISTRIBUTION, FN.GE.1.
        KODE    - A SELECTION PARAMETER
                  KODE=1 RETURNS FCHISQ=F(X)
                  KODE=2 RETURNS FCHISQ=1.-F(X)
      OUTPUT
        FCHISQ - F(X) OR 1.-F(X), DEPENDING ON KODE.
        NZ      - UNDERFLOW FLAG
                  NZ.EQ.0, A NORMAL RETURN.
                  NZ.NE.0, UNDERFLOW, FCHISQ=0.0 RETURNED.

ERROR CONDITIONS
    IMPROPER INPUT - A FATAL ERROR
    UNDERFLOW - A NON-FATAL ERROR.
```

```
FCIRCV    FCIRCV    FCIRCV    FCIRCV          FCIRCV    FCIRCV    FCIRCV
          ****************************************
               ******************************
                    ****************
                         **********
```

```
    FUNCTION FCIRCV(R,T,KODE)

    CDC 6600 ROUTINE

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

    REFERENCE SLA-73-0333

    ABSTRACT
        THE CIRCULAR COVERAGE FUNCTION GIVES THE PROBABILITY OF
        COVERING A POINT TARGET BY A WEAPON OF RADIUS A WHEN THE
        OFFSET AIM IS D, AND THE DISTRIBUTION OF AIMING ERRORS IS
        CIRCULAR NORMAL WITH STANDARD DEVIATION SIG. THIS PROBABILITY
        CAN BE EXPRESSED IN TERMS OF

        P(R,T)=INTEGRAL ON (0,R) OF X*EXP(-(T*T+X*X)/2)*I/SUB(0)/(XT)

        WHERE R=A/SIG, T=D/SIG AND I IS A MODIFIED BESSEL FUNCTION.
        FCIRCV COMPUTES P(R,T) OR 1-P(R,T) BY SUMMING QUADRATURES OF
        LENGTH 4 TO THE LEFT OR RIGHT OF R. TRADE-OFFS ON THE WORK
        REQUIRED ARE MADE AT XR=T+4 OR XL=T-4, SINCE THE INTEGRAND HAS
        A MAXIMUM NEAR X=T AND THE MAJOR CONTRIBUTIONS TO P(R,T) OR
        1-P(R,T) ARE OBTAINED TO THE LEFT OF XR OR THE RIGHT OF XL.
        ECONOMY IS ACHIEVED FOR P(R,T) WITH R.GT.XR BY COMPUTING
        1-P(R,T) AND USING P(R,T)=1-(1-P(R,T)). SIMILARLY, P(R,T) IS
        COMPUTED FOR R.LT.XL AND THE RESULT SUBTRACTED FROM 1 WHEN
        1-P(R,T) IS REQUESTED. OTHERWISE DIRECT QUADRATURE TO THE LEFT
        OF XR OR TO THE RIGHT OF XL GETS P(R,T) OR 1-P(R,T)
        RESPECTIVELY. THE QUADRATURE SUMS ARE TERMINATED AT X=0 OR
        A RELATIVE ERROR TEST ON A TERM BEING ADDED. IF T=0,
        P(R,0)=1-EXP(-R*R/2) AND THE POWER SERIES FOR EXP FUNCTION IS
        USED FOR R.LE.0.1.  ERR=5.E-8 FOR GAUS8 QUADRATURES.
        FCIRCV CALLS BESI01 AND GAUS8. GAUS8 CALLS FUNCTION FEIZX.

    DESCRIPTION OF ARGUMENTS

        INPUT
          R       - RADIUS, STANDARDIZED BY SIG, R.GE.0
          T       - OFFSET AIM, STANDARDIZED BY SIG, T.GE.0
          KODE    - A SELECTION PARAMETER
                    KODE=1 RETURNS FCIRCV=P(R,T)
                    KODE=2 RETURNS FCIRCV=1.-P(R,T)

        OUTPUT
          FCIRCV - P(R,T) OR 1.-P(R,T) DEPENDING ON KODE
```

```
ERROR CONDITIONS
    IMPROPER INPUT PARAMETERS - A FATAL ERROR
    DIAGNOSTICS FROM GAUS8.
```

FFDIST FFDIST FFDIST FFDIST FFDIST FFDIST FFDIST

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

```
FUNCTION FFDIST(X,FM,FN,KODE,NZ)
CDC 6600 ROUTINE

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

REFERENCE SC-DR-73 0333

ABSTRACT
    FFDIST COMPUTES THE CUMULATIVE F DISTRIBUTION F(X) OR ITS
    COMPLEMENT 1.-F(X) FOR A RANDOM VARIABLE F=(U/M)/(V/N) WHERE
    U IS CHI-SQUARE(M) AND V IS CHI-SQUARE(N). THE RELATION OF
    F(X) TO THE INCOMPLETE BETA FUNCTION (NORMALIZED TO 1. AT X=1)
    IS USED FOR THE COMPUTATION IN SUBROUTINE BETAIC.

DESCRIPTION OF ARGUMENTS

    INPUT
      X      - ARGUMENT, X.GE.0.0.
      FM     - DEGREES OF FREEDOM OF THE NUMERATOR CHI-SQUARE(M)
               RANDOM VARIABLE, FM.GE.1.
      FN     - DEGREES OF FREEDOM OF THE DENOMINATOR CHI-SQUARE(N)
               RANDOM VARIABLE, FN.GE.1.
      KODE   - A SELECTION PARAMETER
               KODE=1 RETURNS FFDIST=F(X)
               KODE=2 RETURNS FFDIST=1.-F(X)
    OUTPUT
      FFDIST - F(X) OR 1.-F(X), DEPENDING ON KODE
      NZ     - UNDERFLOW FLAG
               NZ=0, A NORMAL RETURN,
               NZ=1, UNDERFLOW, FFDIST=0.0 RETURNED

ERROR CONDITIONS
    IMPROPER INPUT - A FATAL ERROR
    UNDERFLOW - A NON-FATAL ERROR.
```

```
FNORM     FNORM     FNORM     FNORM          FNORM     FNORM     FNORM     FNORM
          ****************************************
               ******************************
                    ********************
                         **********
     FUNCTION FNORM(X,KODE,NZ)

     CDC 6600 ROUTINE

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

     REFERENCE SC-DR-72-0918

     ABSTRACT
         FNORM COMPUTES THE CUMULATIVE NORMAL DISTRIBUTION F(X) OR
         ITS COMPLEMENT 1.-F(X). CHEBYSHEV EXPANSIONS FOR ERF(Z) ON
         0.LE.Z.LT.2 AND ERFC(Z) ON 2.LE.Z.LE.4 AND Z.GT.4 ARE
         USED FOR EVALUATION.  THE RELATIONS
            F(X)=.5*ERFC(Z)    , Z=-X/SQRT(2) , X.LT.-2.*SQRT(2)
            F(X)=.5-.5*ERF(Z)  , Z=-X/SQRT(2) ,-2*SQRT(2).LE.X.LT.0
            F(X)=.5+.5*ERF(Z)  , Z= X/SQRT(2) , 0.LE.X.LT.2*SQRT(2)
            F(X)=1.-.5*ERFC(Z) , Z=X/SQRT(2) , 2.LE.Z.LT.6
            F(X)=1. , X.GE.6*SQRT(2)
            F(-X)=1.-F(X) ,
         ARE USED TO COMPLETE THE DEFINITION ON THE REAL LINE SO THAT
         SIGNIFICANT DIGITS ARE RETAINED OVER THE FULL EXPONENT RANGE.

     DESCRIPTION OF ARGUMENTS

         INPUT
           X       - ARGUMENT OF THE DISTRIBUTION
           KODE    - A SELECTION PARAMETER
                     KODE=1 RETURNS FNORM=F(X)
                     KODE=2 RETURNS FNORM=1.-F(X)

         OUTPUT
           FNORM   - ANSWER FOR F(X) OR 1.-F(X) DEPENDING ON KODE.
           NZ      - UNDERFLOW FLAG
                     NZ=0, A NORMAL RETURN
                     NZ=1, UNDERFLOW, FNORM=0.0 RETURNED

     ERROR CONDITIONS
         IMPROPER INPUT FOR KODE- A FATAL ERROR
         UNDERFLOW - A NON-FATAL ERROR, XLIM=-36.5444845898331 IS THE
         CRITICAL VALUE.
```

```
FNORMB    FNORMB    FNORMB    FNORMB         FNORMB    FNORMB    FNORMB
          ****************************************
               ******************************
                    ********************
                         **********
     FUNCTION FNORMB(X1,X2,RHO)
     CDC 6600 ROUTINE

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

     REFERENCE  SLA-73-0334

     ABSTRACT
         FNORMB COMPUTES THE CUMULATIVE BIVARIATE NORMAL PROBABILITY
         P(R1.LE.X1,R2.LE.X2,RHO)=N(X1,X2,RHO) FOR RANDOM VARIABLES
         R1 AND R2 WITH CORRELATION RHO. A VARIETY OF FUNCTIONAL
         RELATIONSHIPS ARE USED TO COVER THE (X1,X2) PLANE. FOR A
         GENERAL TRIPLE, THE PROBLEM IS REDUCED TO N(X,0,ALPHA) SINCE

                   N(X1,X2,RHO)=N(X1,0,GAMMA)+N(X2,0,BETA)+H
```

```
      WHERE GAMMA, BETA, AND H ARE SIMPLE FUNCTIONS OF X1,X2, AND
      RHO. SUBROUTINE FXXRHO COMPUTES N(X,0,ALPHA) BY MEANS OF THE
      THA INTEGRAL OF OWEN. SUBROUTINE THA COMPUTES THIS INTEGRAL
      BY QUADRATURE WITH GAUS8 AND FUNCTION FTHA. FOR X.GT.1, THE
      THA INTEGRAL CAN BE RELATED TO THE SAME INTEGRAL FOR AN
      ARGUMENT LESS THAN 1, KEEPING THE RANGE FOR GAUS8 WELL SCALED.
      FNORMB USES SUBROUTINES FXXRHO, THA AND FUNCTIONS FNORM
      ERF, ERFC, ASIN.  FNORMB RETURNS APPROXIMATELY 8 DECIMAL
      PLACE ACCURACY.

  DESCRIPTION OF ARGUMENTS

      INPUT
        X1      - ARGUMENT, UNRESTRICTED
        X2      - ARGUMENT, UNRESTRICTED
        RHO     - CORRELATION COEFICIENT, -1.LE.RHO.LE.1

      OUTPUT
        FNORMB - CUMULATIVE NORMAL PROBABILITY, P

  ERROR CONDITIONS
      IMPROPER INPUT PARAMETERS- A FATAL ERROR
      DIAGNOSTICS FROM GAUS8
```

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

FXXRHO FXXRHO FXXRHO FXXRHO FXXRHO FXXRHO FXXRHO

```
SUBROUTINE FXXRHO(XB,RHOB,PROB)
FXXRHO COMPUTES THE BIVARIATE NORMAL FOR TRIPLES (XX,0.,RHO) BY
MEANS OF OWENS FORMULA.

CDC 6600 SUBROUTINE
FXXRHO USES ROUTINES FNORM, THA, FTHA, GAUS8 AND ERRCHK.
```

GAMFN GAMFN GAMFN GAMFN GAMFN GAMFN GAMFN GAMFN

```
FUNCTION GAMFN(X,IERR)

    A CDC 6600 SUBROUTINE

WRITTEN BY D.E. AMOS, JANUARY, 1976.

REFERENCE

ABSTRACT
    GAMFN COMPUTES THE GAMMA FUNCTION ON THE INTERVAL  (-169,+170)
    EXCEPT AT ZERO OR THE NEGATIVE INTEGERS WHERE THE GAMMA
    FUNCTION HAS POLES. A RATIONAL CHEBYSHEV APPROXIMATION IS
    USED ON (2,3) WITH FORWARD RECURSION ON

                  GAMMA(X+1) = X*GAMMA(X)

    UP TO X=10 AND BACKWARD RECURSION DOWN TO X=0. FOR X GREATER
    THAN 10, ANOTHER RATIONAL CHEBYSHEV APPROXIMATION IS USED FOR
    THE LOG OF THE GAMMA FUNCTION. FOR X NEGATIVE AND NOT AN
    INTEGER, THE REFLECTION FORMULA

        GAMMA(-X)=-PI/(SIN(PI*X)*GAMMA(1+X)),  X.GT.0
         *
    IS USED. FOR X=1.,2.,...,100., A TABLE LOOK-UP IS PERFORMED
    ON THE GAM VECTOR FOR FACTORIALS. IF LARGE POSITIVE VALUES OF
    X ARE ANTICIPATED, GAMLN FOR THE NATURAL LOG OF THE GAMMA
    FUNCTION MAY BE MORE APPROPRIATE FOR SCALING PURPOSES.

DESCRIPTION OF ARGUMENTS

    INPUT
       X     - ARGUMENT, -169.LT.X.LE.+170

    OUTPUT
       GAMFN - A VALUE FOR GAMMA(X)
       IERR  - AN ERROR FLAG
               IERR = 0 , NORMAL RETURN, AT LEAST 11 SIGNIFICANT
                          DIGITS

               IERR.NE.0, X IS WITHIN 0.01 OF A NEGATIVE INTEGER
                          RESULTING IN REDUCED SIGNIFICANCE BY
                          APPROXIMATELY IERR DIGITS, OR 14-IERR
                          SIGNIFICANT DIGITS RETAINED.

ERROR CONDITIONS
    IMPROPER INPUT ARGUMENT - A FATAL ERROR
    X IS ZERO OR A NEGATIVE INTEGER - A FATAL ERROR
    X WITHIN 0.01 OF A NEGATIVE INTEGER - A NON-FATAL ERROR.
```

GAMIC GAMIC GAMIC GAMIC GAMIC GAMIC GAMIC GAMIC

```
SUBROUTINE GAMIC(X,ALPHA,REL,N,Y,NZ)

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

REFERENCE SC-DR-72 0303

ABSTRACT
    GAMIC COMPUTES AN N MEMBER SEQUENCE OF INCOMPLETE GAMMA
    FUNCTIONS NORMALIZED SO THAT AT X=INFINITY, THE INCOMPLETE
    GAMMA FUNCTION HAS THE VALUE 1. THE SEQUENCE IS DENOTED BY

    Y(K)=INCGAMMA(ALPHA+K-1,X)/GAMMA(ALPHA+K-1),  K=1,2,...,N

    AND IS COMPUTED TO A RELATIVE ERROR REL OR BETTER WHERE ALPHA
    .GT.0. IF ALPHA+N-1.GE.X, THE LAST MEMBER IS COMPUTED BY THE
    CONFLUENT HYPERGEOMETRIC SERIES WITH THE OTHER MEMBERS
    COMPUTED BY BACKWARD RECURSION ON A TWO-TERM FORMULA,

      Y(K-1)=Y(K)+EXP((ALPHA+K-1)*ALOG(X)-X-GAMLN(ALPHA+K)).

    IF ALPHA+N-1.LT.X, AN INTEGER M IS ADDED SO THAT
    ALPHA+N-1+M.GE.X AND THE FIRST PROCEDURE IS APPLIED. SPECIAL
    PROCEDURES APPLY FOR ALPHA.EQ.1 OR AN UNDERFLOW OCCURS OR
    X EXCEEDS A CRITICAL VALUE, APTEST, WHERE ALL MEMBERS ARE 1.
    TO THE WORD LENGTH OF THE CDC 6600. GAMIC USES GAMLN.

DESCRIPTION OF ARGUMENTS

    INPUT
      X      - ARGUMENT, X.GE.0.0
      ALPHA  - PARAMETER, ALPHA.GT.0.0
      REL    - RELATIVE ERROR TOLERANCE, REL=1.E-S FOR S
               SIGNIFICANT DIGITS
      N      - NUMBER OF GAMMA FUNCTIONS IN THE SEQUENCE
               BEGINNING AT PARAMETER ALPHA, N.GE.1

    OUTPUT
      Y      - A VECTOR CONTAINING AN N MEMBER SEQUENCE
               Y(K)= INCGAMMA(ALPHA+K-1,X)/GAMMA(ALPHA+K-1),
               K=1,...,N TO A RELATIVE ERROR REL.
      NZ     - UNDERFLOW FLAG
               NZ.EQ.0, A NORMAL RETURN
               NZ.NE.0, UNDERFLOW, Y(K)=0.0, K=N-NZ+1,N RETURNED

ERROR CONDITIONS
    IMPROPER INPUT PARAMETERS - A FATAL ERROR
    UNDERFLOW - A NON-FATAL ERROR.
```

GAMLN GAMLN GAMLN GAMLN GAMLN GAMLN GAMLN GAMLN

FUNCTION GAMLN(X)

WRITTEN BY D.E. AMOS AND S.L. DANIEL, JANUARY, 1976.

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.,...,100., 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

GAMMAZ GAMMAZ GAMMAZ GAMMAZ GAMMAZ GAMMAZ GAMMAZ

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

```
GAMTL   GAMTL   GAMTL   GAMTL   GAMTL   GAMTL   GAMTL   GAMTL
        ****************************************
          ****************************
            ********************
              **********
```

```
      SUBROUTINE GAMTL(X,B,REL,N,Y,NZ)

      CDC 6600 ROUTINE

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

      REFERENCE SC-DR-72 0303

      ABSTRACT
          GAMTL COMPUTES AN N MEMBER SEQUENCE OF COMPLEMENTARY
          GAMMA FUNCTIONS

          Y(K)=1.-INCGAMMA(B+K-1,X)/GAMMA(B+K-1), K=1,...,N,

          TO A RELATIVE ERROR REL FOR X.GE.0 AND B.GT.0. THE CONTINUED
          FRACTION IS EVALUATED FOR BO.GT.0., BO=B-INTEGER PART OF
          B, FOLLOWED BY FORWARD RECURSION ON ITS TWO TERM RELATION TO
          RAISE BO TO B+N-1.  THE CONVERGENCE IS BEST FOR LARGE
          X .GE. MAX(1,BO). WHERE SPEED IS A CONSIDERATION, EVALUATE
          Y(K) BY SUBTRACTING THE INCOMPLETE GAMMA FUNCTION FROM 1.
          USING SUBROUTINE GAMIC FOR X.LT.MAX(1,B+K-1) AND GAMTL FOR
          X.GE.MAX(1,B+K-1).

      DESCRIPTION OF ARGUMENTS

          INPUT
            X       - ARGUMENT, X.GE.0.0
            B       - PARAMETER, B.GT.0.0
            REL     - RELATIVE ERROR REQUIREMENT,
                      REL=1.E-S FOR S SIGNIFICANT DIGITS, 0.LE.S.LE.12
            N       - NUMBER OF COMPLEMENTARY FUNCTIONS IN THE SEQUENCE
                      BEGINNING AT PARAMETER B, N.GE.1

          OUTPUT
            Y       - A VECTOR CONTAINING AN N MEMBER SEQUENCE
                      Y(K)=1.-INCGAMMA(B+K-1,X)/GAMMA(B+K-1), K=1,...,N
                      TO A RELATIVE ERROR REL.
            NZ      - UNDERFLOW FLAG
                      NZ.EQ.0, A NORMAL RETURN
                      NZ.NE.0, UNDERFLOW, Y(K)=0.0, K=1,NZ RETURNED.

      ERROR CONDITIONS
          IMPROPER INPUT - A FATAL ERROR
          CONTINUED FRACTION DOES NOT CONVERGE - A FATAL ERROR.
          UNDERFLOW - A NON-FATAL ERROR.
```

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

MINA MINA MINA MINA MINA MINA MINA MINA MINA

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

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

NNLS NNLS NNLS NNLS NNLS NNLS NNLS NNLS NNLS

****

SUBROUTINE NNLS (A,MDA,M,N,B,X,RNORM,W,ZZ,INDEX,MODE)
DIMENSION A(MDA,N),B(M),X(N),W(N),ZZ(M),INDEX(N)

WRITTEN BY C. L. LAWSON AND R. J. HANSON, FROM THE BOOK SOLVING LEAST SQUARES PROBLEMS, PRENTICE-HALL, INC. (1974). FOR FURTHER ALGORITHMIC DETAILS SEE ALGORITHM NNLS IN CHAPTER 23.

ABSTRACT

GIVEN AN M BY N MATRIX A AND AN M VECTOR B THIS SUBPROGRAM COMPUTES THE SOLUTION TO THE LEAST SQUARES PROBLEM AX = B SUBJECT TO X.GE.0. THE INEQUALITY CONSTRAINT X.GE.0 MEANS THAT EVERY COMPONENT OF THE SOLUTION WILL BE NONNEGATIVE.

EITHER M.GE.N OR M.LT.N IS PERMITTED. THIS PROBLEM ALWAYS HAS A SOLUTION BUT IT IS NOT UNIQUE IF THE RANK OF A IS .LT. N.

THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST.. A(MDA,N),B(M),X(N),W(N),ZZ(M),INDEX(N). THIS PERMITS THE SOLUTION OF A RANGE OF PROBLEMS IN THE SAME ARRAY SPACE.

THE PARAMETERS FOR NNLS ARE

INPUT..

A(*,*),MDA,M,N	THE ARRAY A(*,*) IS DOUBLY SUBSCRIPTED WITH FIRST DIMENSIONING PARAMETER EQUAL TO MDA. THE ARRAY A(*,*) INITIALLY CONTAINS THE M BY N MATRIX A. EITHER M.GE.N OR M.LT.N IS PERMITTED. THE FIRST DIMENSIONING PARAMETER MUST SATISFY MDA.GE.M. THE CONDITION MDA.LT.M IS CONSIDERED AN ERROR.
B(*)	THE ARRAY B(*) CONTAINS THE M-VECTOR B.

OUTPUT..

A(*,*),B(*)	THE CONTENTS OF THE ARRAYS A(*,*) AND B(*) WILL BE MODIFIED BY THE SUBROUTINE. THESE CONTENTS ARE GENERALLY NOT REQUIRED BY THE USER.
X(*)	ON TERMINATION THIS ARRAY CONTAINS A VECTOR X.GE.0. IF MODE=1, X(*) CONTAINS A SOLUTION VECTOR. IF MODE=3, X(*) CONTAINS A NONNEGATIVE APPROXIMATE SOLUTION AND RNORM CONTAINS THE CORRESPONDING RESIDUAL VECTOR NORM. IF MODE=2, X(*) IS NOT DEFINED.
RNORM	ON TERMINATION RNORM CONTAINS THE EUCLIDEAN

LENGTH OF THE FINAL RESIDUAL VECTOR B-AX.

W(*),ZZ(*),INDEX(*)
ON OUTPUT W(*) CONTAINS THE N-VECTOR OF DUAL COEFFICIENTS W=(TRANSPOSE OF A)(B-AX). THE ARRAYS ZZ(*) AND INDEX(*) ARE WORKING SPACE WHOSE CONTENTS ARE NOT GENERALLY REQUIRED BY THE USER.

MODE
THIS FLAG IS SET BY THE SUBROUTINE TO INDICATE THE STATUS OF THE COMPUTATION ON COMPLETION..

MODE=1, SUCCESSFUL
MODE=2, BAD DIMENSIONS. ONE OF THE CONDITIONS M.LE.0 OR N.LE.0 OCCURRED.
MODE=3, MAXIMUM NUMBER (3*N) ITERATIONS HAS BEEN EXCEEDED.

THE SOLUTION IS USUALLY OBTAINED IN ABOUT N/2 ITERATIONS. A RETURN WITH MODE=3 MAY INDICATE THAT THE DATA MATRIX CONTAINS IMPROPERLY DEFINED ENTRIES.

ODE ODE ODE ODE ODE ODE ODE ODE ODE ODE

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

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 , STEP1 , 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 STEP1 AND INTRP TO ADVANCE THE INTEGRATION AND TO INTERPOLATE AT OUTPUT POINTS. STEP1 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 STEP1 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 ,  STEP1 , 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  STEP1  TO
   ADVANCE THE SOLUTION AND  INTRP  TO INTERPOLATE THE SOLUTION AND
   ITS DERIVATIVE.  STEP1  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  STEP1  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  STEP1 , 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)
      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)
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 POLFIT POLFIT POLFIT POLFIT POLFIT

SUBROUTINE POLFIT (N,X,Y,W,MAXORD,NORD,EPS,R,IERR,A)
WRITTEN BY L. F. SHAMPINE AND S. M. DAVENPORT. THE STATISTICAL OPTIONS PROVIDED WERE WRITTEN BY R. E. HUDDLESTON.

ABSTRACT

GIVEN A COLLECTION 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 POLFIT 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 FOR 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, POLFIT 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 TO 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)
     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)
     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)
      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 (BCT 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)
      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 COMPUTATION.  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 OF 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
                     REASONS 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 ERROR
                              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
                     EQUATIONS 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)
   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)
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.
```

QPPLOT QPPLOT QPPLOT QPPLOT QPPLOT QPPLOT QPPLOT

```
SUBROUTINE QPPLOT (X1,Y1,N1)
```

TRIVIAL INTERFACE INTO ROUTINE XPPLOT FOR LINE PRINTER PLOTS OF A SINGLE FUNCTION, (X1(I),Y1(I),I=1,N1).

QTPLOT QTPLOT QTPLOT QTPLOT QTPLOT QTPLOT QTPLOT

```
SUBROUTINE QTPLOT (X1,Y1,N1)
```

TRIVIAL INTERFACE INTO ROUTINE XPPLOT FOR TELETYPE PLOTS OF A SINGLE FUNCTION (X1(I),Y1(I),I=1,N1).

RBND2 RBND2 RBND2 RBND2 RBND2 RBND2 RBND2 RBND2

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

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)
      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 FCURTR(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)
   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)
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)
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)
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)
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)
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.
```

```
RSBND    RSBND    RSBND    RSBND       RSBND    RSBND    RSBND    RSBND
              ****************************************
                   ****************************
                        ******************
                             **********
     SUBROUTINE RSBND(NDIM,N,A,EV,VEC,EVIMP,BNDS)

     ABSTRACT
       THIS SUBROUTINE CALCULATES RAYLEIGH QUOTIENT CORRECTIONS
       FOR THE COMPUTED EIGENVALUES OF A REAL SYMMETRIC MATRIX
       AND UPPER BOUNDS ON THE ABSOLUTE ERROR OF THE COMPUTED
       EIGENSYSTEM.  REASONABLE BOUNDS FOR THE EIGENVECTORS ARE
       POSSIBLE ONLY WHEN THE EIGENVALUES ARE WELL-SEPARATED.
       WHEN THIS IS NOT THE CASE, NO BOUND IS CALCULATED.

       TO COMPUTE ERROR BOUNDS FOR THE EIGENSYSTEMS OF COMPLEX
       HERMITIAN MATRICES, SEE SUBROUTINE CHBND.  SIMILAR BOUNDS
       FOR REAL NON-SYMMETRIC AND COMPLEX NON-HERMITIAN MATRICES
       ARE NOT POSSIBLE.

     DESCRIPTION OF ARGUMENTS
       ON INPUT
         NDIM  MUST BE THE ROW DIMENSION OF ARRAYS A,VEC,BNDS
               IN THE CALLING PROGRAM DIMENSION STATEMENT.
         N     IS THE ORDER OF THE MATRIX.  1.LE.N.LE.NDIM.
         A     MUST CONTAIN IN THE LEADING N BY N SUBARRAY
               THE REAL SYMMETRIC MATRIX.  ONLY THE DIAGONAL
               AND LOWER TRIANGLE NEED BE DEFINED.
         EV    MUST CONTAIN IN THE FIRST N ELEMENTS THE REAL
               EIGENVALUES AS COMPUTED, SAY, BY RSAA.
         VEC   MUST CONTAIN IN THE LEADING N BY N SUBARRAY
               THE ORTHONORMAL EIGENVECTORS AS COMPUTED, SAY,
               BY RSAA.  THE J-TH COLUMN OF VEC MUST CORRESPOND
               TO THE J-TH ELEMENT OF EV.
       ON OUTPUT
         EVIMP  CONTAINS DOUBLE PRECISION CORRECTED EIGENVALUES
                (RAYLEIGH QUOTIENTS) IN THE SAME ORDER AS EV.
         BNDS   CONTAINS UPPER BOUNDS ON THE ABSOLUTE ERRORS OF
                THE COMPUTED EIGENSYSTEM
                BNDS(J,1)  UPPER BOUND ON ABSOLUTE ERROR IN
                           EVIMP(J).
                BNDS(J,2)  UPPER BOUND ON L-2 NORM OF ERROR IN
                           J-TH COMPUTED EIGENVECTOR.  THIS
                           QUANTITY IS SET TO -1.0 WHEN EIGENVALUES
                           ARE TOO CLOSE TO PERMIT A REASONABLE BOUND.
                BNDS(J,3)  L-2 NORM OF RESIDUAL ASSOCIATED
                           WITH EV(J) AND VEC(*,J).
```

RVNORM RVNORM RVNORM RVNORM RVNORM RVNORM RVNORM

FUNCTION RVNORM(RMU,SIG)

WRITTEN BY D.E. AMOS, JULY, 1976.

REFERENCES
SAND77-0612

HASTINGS,C. JR., APPROXIMATIONS FOR DIGITAL COMPUTERS,
PRINCETON UNIV. PRESS, PRINCETON, N.J., 1955.

ABSTRACT
RVNORM COMPUTES A VALUE FOR A NORMAL (RMU,SIG) RANDOM VARIABLE ON EACH CALL. A VALUE Y OF A UNIFORM RANDOM VARIABLE ON (0,1) IS RETURNED FROM FUNCTION RANF. A TABLE OF 96 PERCENT POINTS FOR A NORMAL (0,1) RANDOM VARIABLE IS LINEARLY INTERPOLATED FOR RVN WHEN Y IS IN THE CLOSED INTERVAL (0.02,0.98). FOR Y OUTSIDE THIS INTERVAL, RVN IS COMPUTED BY A LOW ACCURACY RATIONAL CHEBYSHEV APPROXIMATION FOR THE INVERSE NORMAL. THEN,

RVNORM = RVN*SIG + RMU .

THE MAXIMUM EROR OF THE NORMALIZED VARIABLE RVN IS APPROXIMATELY 0.37 PERCENT.

CAUTION

DO NOT REFERENCE RVNORM WITH THE SAME ARGUMENTS MORE THAN ONCE IN A GIVEN STATEMENT. THE COMPILER MAY INTERPRET THE RETURN FOR EACH CALL TO BE THE SAME, WHEN DIFFERENT VALUES ARE EXPECTED BY THE USER.

DESCRIPTION OF ARGUMENTS
INPUT
RMU - MEAN OF THE NORMAL RANDOM VARIABLE
SIG - STANDARD DEVIATION OF THE NORMAL RANDOM VARIABLE

OUTPUT
RVNORM - VALUE OF NORMAL (RMU,SIG) RANDOM VARIABLE

ERROR CONDITIONS
NONE

SAXB SAXB SAXB SAXB SAXB SAXB SAXB SAXB SAXB

SUBROUTINE SAXB(ND,N,M,A,B,INIT,IN,KER)
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 ECUATIONS 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)
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 (RCW) 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)
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 FICO 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 PORCEEDING 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)
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)
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)
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.
```

```
SODS    SODS    SODS    SODS    SODS        SODS    SODS    SODS    SODS
              ****************************************
                  ******************************
                      ******************
                          **********
    SUBROUTINE SODS(A,X,B,NEQ,NUK,NRDA,IFLAG,WORK,IWORK)

    SODS SOLVES THE OVERDETERMINED SYSTEM OF LINEAR EQUATIONS A X = B,
    WHERE A IS NEQ BY NUK AND NEQ .GE. NUK. IF RANK A = NUK,
    X IS THE UNIQUE LEAST SQUARES SOLUTION VECTOR. THAT IS,
             R(1)**2 + ..... + R(NEQ)**2 = MINIMUM
    WHERE R IS THE RESIDUAL VECTOR  R = B - A X.
    IF RANK A .LT. NUK , THE LEAST SQUARES SOLUTION OF MINIMAL
    LENGTH CAN BE PROVIDED.
    SODS IS AN INTERFACING ROUTINE WHICH CALLS SUBROUTINE LSSODS
    FOR THE SOLUTION. LSSODS IN TURN CALLS SUBROUTINE ORTHOL AND
    POSSIBLY SUBROUTINE OHTROR FOR THE DECOMPOSITION OF A BY
    ORTHOGONAL TRANSFORMATIONS. IN THE PROCESS,ORTHOL CALLS UPON
    SUBROUTINE CSCALE FOR SCALING.

    WRITTEN BY H.A. WATTS , ORG. 2642 , SANDIA LABORATORIES

   REFERENCES
    G.GOLUB, NUMERICAL METHODS FOR SOLVING LINEAR LEAST SQUARES
    PROBLEMS, NUMER. MATH.,V.7,PP.206,H-216,1965.
    P. BUSINGER AND G. GOLUB, LINEAR LEAST SQUARES SOLUTIONS BY
    HOUSEHOLDER TRANSFORMATIONS,NUMER. MATH.,V.7,PP.269-276,1965.

    H.A.WATTS,SOLVING LINEAR LEAST SQUARES PROBLEMS USING
    SODS/SUDS/CODS , SANDIA REPORT SAND77-0683

**************************************************************************
  INPUT
**************************************************************************

    A -- CONTAINS THE MATRIX OF NEQ EQUATIONS IN NUK UNKNOWNS AND MUST
         BE DIMENSIONED NRDA BY NUK. THE ORIGINAL A IS DESTROYED
    X -- SOLUTION ARRAY OF LENGTH AT LEAST NUK
    B -- GIVEN CONSTANT VECTOR OF LENGTH NEQ, B IS DESTROYED
    NEQ -- NUMBER OF EQUATIONS, NEQ GREATER OR EQUAL TO 1
    NUK -- NUMBER OF COLUMNS IN THE MATRIX (WHICH IS ALSO THE NUMBER
           OF UNKNOWNS), NUK NOT LARGER THAN NEQ
    NRDA -- ROW DIMENSION OF A, NRDA GREATER OR EQUAL TO NEQ
    IFLAG -- STATUS INDICATOR
           =0 FOR THE FIRST CALL (AND FOR EACH NEW PROBLEM DEFINED BY
              A NEW MATRIX A) WHEN THE MATRIX DATA IS TREATED AS EXACT
          =-K FOR THE FIRST CALL (AND FOR EACH NEW PROBLEM DEFINED BY
              A NEW MATRIX A) WHEN THE MATRIX DATA IS ASSUMED TO BE
```

```
            ACCURATE TO ABOUT K DIGITS
          =1 FOR SUBSEQUENT CALLS WHENEVER THE MATRIX A HAS ALREADY
            BEEN DECOMPOSED (PROBLEMS WITH NEW VECTORS B BUT
            SAME MATRIX A CAN BE HANDLED EFFICIENTLY)
    WORK(*),IWORK(*) -- ARRAYS FOR STORAGE OF INTERNAL INFORMATION,
                  WORK MUST BE DIMENSIONED AT LEAST  2 + 5*NUK
                  IWORK MUST BE DIMENSIONED AT LEAST NUK+2
    IWORK(2) -- SCALING INDICATOR
              =-1 IF THE MATRIX A IS TO BE PRE-SCALED BY
              COLUMNS WHEN APPROPRIATE
              IF THE SCALING INDICATOR IS NOT EQUAL TO -1
              NO SCALING WILL BE ATTEMPTED
            FOR MOST PROBLEMS SCALING WILL PROBABLY NOT BE NECESSARY

*************************************************************************
  OUTPUT
*************************************************************************

    IFLAG -- STATUS INDICATOR
          =1 IF SOLUTION WAS OBTAINED
          =2 IF IMPROPER INPUT IS DETECTED
             (EXECUTION TERMINATES UNLESS A PRIOR CALL TO
              ERXSET WAS MADE)
          =3 IF RANK OF MATRIX IS LESS THAN NUK
             IF THE MINIMAL LENGTH LEAST SQUARES SOLUTION IS
             DESIRED, SIMPLY RESET IFLAG=1 AND CALL THE CODE AGAIN
             (THE USER MUST MAKE A PRIOR CALL TO THE ERXSET
              ROUTINE IF THIS RETURN IS TO BE NONFATAL WHEN
              THE INPUT MODE IFLAG=0 IS USED. THIS IS NOT
              NECESSARY WHEN THE INPUT MODE IFLAG=-K IS USED.)
    X -- LEAST SQUARES SOLUTION OF  A X = B
    A -- CONTAINS THE STRICTLY UPPER TRIANGULAR PART OF THE REDUCED
         MATRIX AND THE TRANSFORMATION INFORMATION
    WORK(*),IWORK(*) -- CONTAINS INFORMATION NEEDED ON SUBSEQUENT
                        CALLS (IFLAG=1 CASE ON INPUT) WHICH MUST NOT
                        BE ALTERED
                        WORK(1) CONTAINS THE EUCLIDEAN NORM OF
                        THE RESIDUAL VECTOR
                        WORK(2) CONTAINS THE EUCLIDEAN NORM OF
                        THE SOLUTION VECTOR
                        IWORK(1) CONTAINS THE NUMERICALLY DETERMINED
                        RANK OF THE MATRIX A

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

SOSNLE SOSNLE SOSNLE SOSNLE SOSNLE SOSNLE SOSNLE

```
                *************************************
                 ***********************************
                  *********************************
                   *******************
                    *********
    SUBROUTINE SOSNLE(FNC,NEQ,X,RER,AER,IFLG,W,NWD)

    WRITTEN BY H.A.WATTS
    APPLIED MATHEMATICS DIVISION 2642
    SANDIA LABORATORIES,ALBUQUERQUE,NEW MEXICO

   ABSTRACT
    SOSNLE SOLVES A SYSTEM OF NEQ SIMULTANEOUS NONLINEAR EQUATIONS.
    THE ALGORITHM IS BASED ON AN ITERATIVE METHOD WHICH IS A
    VARIATION OF NEWTONS METHOD USING GAUSSIAN ELIMINATION
    IN A MANNER SIMILAR TO THE GAUSS-SEIDEL PROCESS. CONVERGENCE
    IS ROUGHLY QUADRATIC. ALL PARTIAL DERIVATIVES REQUIRED BY
    THE ALGORITHM ARE APPROXIMATED BY FIRST DIFFERENCE QUOTIENTS.

    ACTUALLY,SOSNLE IS MERELY AN INTERFACING ROUTINE FOR
    CALLING SUBROUTINE SNLEQS WHICH EMBODIES THE SOLUTION
    ALGORITHM. THE PURPOSE OF THIS IS TO ADD GREATER
```

```
      FLEXIBILITY AND EASE OF USE FOR THE PROSPECTIVE USER.

      SNLEQS CALLS THE ACCOMPANYING ROUTINE BACSOL WHICH SOLVES A
      TRIANGULAR LINEAR SYSTEM BY BACK-SUBSTITUTION.

      THE USER MUST SUPPLY A FUNCTION SUBPROGRAM WHICH EVALUATES THE
      K-TH EQUATION ONLY (K SPECIFIED BY SNLEQS) FOR EACH CALL
      TO THE SUBPROGRAM.

      SOSNLE REPRESENTS AN IMPLEMENTATION OF THE MATHEMATICAL ALGORITHM
      DESCRIBED IN THE REFERENCES BELOW.
      REFERENCES
        1. K.M.BROWN,ALGORITHM 316,COMM.A.C.M.,VOL.10,1967,PP728-729.
        2. K.M.BROWN,A QUADRATICALLY CONVERGENT NEWTON-LIKE METHOD
           BASED UPON GAUSSIAN ELIMINATION,SIAM J.NUM.ANAL.,VOL.6,1969,
           PP560-569.


*************************************************************************
**** A MACHINE DEPENDENT CONSTANT URO APPEARS IN A DATA STATEMENT
**** IN THE SUBROUTINE SNLEQS.
**** THIS SHOULD BE APPROXIMATELY THE COMPUTER UNIT-ROUNDOFF VALUE.
**** THAT IS, SET URO TO THE VALUE OF THE SMALLEST POSITIVE NUMBER
**** SUCH THAT 1.0+URO .GT. 1.0
**** THE DESIGNATED COMPUTER OUTPUT TAPE UNIT NUMBER ALSO APPEARS
**** IN A DATA STATEMENT.SEE THE IFLG INPUT DESCRIPTION.
*************************************************************************


   -INPUT-

     FNC -NAME OF THE FUNCTION PROGRAM WHICH EVALUATES THE EQUATIONS.
          THIS NAME MUST BE IN A EXTERNAL STATEMENT IN THE CALLING
          PROGRAM. THE USER MUST SUPPLY FNC IN THE FORM FNC(X,K)
          WHERE X IS THE SOLUTION VECTOR (WHICH MUST BE DIMENSIONED
          IN FNC) AND FNC RETURNS THE VALUE OF THE K-TH FUNCTION.
     NEQ -NUMBER OF EQUATIONS TO BE SOLVED.
     X   -SOLUTION VECTOR. INITIAL GUESSES MUST BE SUPPLIED.
     RER -RELATIVE ERROR TOLERANCE USED IN THE CONVERGENCE CRITERIA.
          EACH SOLUTION COMPONENT X(I) IS CHECKED BY AN ACCURACY
          REQUIREMENT OF ABS(X(I)-XOLD(I)) .LE. RE*ABS(X(I))+AER,
          WHERE XOLD(I) REPRESENTS THE PREVIOUS ITERATION VALUE AND
          RE=MAX(RER,URO). URO IS THE MACHINE UNIT-ROUNDOFF VALUE.
     AER -ABSOLUTE ERROR TOLERANCE USED IN THE CONVERGENCE CRITERIA
          IF THE USER SUSPECTS SOME SOLUTION COMPONENT MAY BE ZERO,HE
          SHOULD SET AER TO AN APPROPRIATE (DEPENDS ON THE SCALE OF
          THE REMAINING VARIABLES) POSITIVE VALUE FOR BETTER
          EFFICIENCY OF THE ROUTINE.
     IFLG-ON INPUT, IT IS USED AS AN INTERNAL PRINTING PARAMETER.
          YOU MUST SET IFLG=-1 IF YOU WANT THE INTERMEDIATE
          SOLUTION ITERATES TO BE PRINTED. ANY OTHER VALUE WILL NOT
          CAUSE PRINTING OF THE SUCCESSIVE SOLUTION APPROXIMATIONS.
          IT MAY BE NECESSARY TO ALTER THE FORMAT STATEMENT
          TO CONFORM TO YOUR PARTICULAR MACHINE. ALSO,IT MAY BE
          NECESSARY TO ALTER THE OUTPUT TAPE UNIT NUMBER SPECIFIED
          BY A DATA STATEMENT IN SNLEQS.
     WORK-A WORK ARRAY USED INTERNALLY BY SNLEQS. SOSNLE SPLITS UP
          WORK INTO 8 SUB-ARRAYS FOR LOCAL STORAGE USE BY
          SNLEQS. ONE OF THESE IS USED THERE AS AN INTEGER
          ARRAY AND MAY CAUSE PROBLEMS WITH SOME COMPILERS.
          IF THIS IS THE CASE,ADD THE DECLARATION CARD   REAL IS
          IN SNLEQS.
     NWD -DIMENSION OF THE WORK ARRAY. NWD MUST BE AT LEAST
          7*NEQ+NEQ*(NEQ+1)/2

   -OUTPUT-

     X   -SOLUTION VECTOR.
     IFLG-STATUS INDICATOR
          0 MEANS SATISFACTORY CONVERGENCE TO A SOLUTION WAS ACHIEVED.
            EACH SOLUTION COMPONENT X(I) SATISFIES THE ERROR
            TOLERANCE TEST ABS(X(I)-XOLD(I)) .LE. RE*ABS(X(I))+AER.
            THIS REPRESENTS NORMAL TERMINATION.
          1 MEANS PROCEDURE CONVERGED TO A SOLUTION SUCH THAT ALL
```

```
      RESIDUALS ARE ZERO.THE SITUATION FOR IFLG=0 ALSO HOLDS
      SINCE THE SOLUTION INCREMENT IS ZERO.THIS IS FREQUENTLY
      ASSOCIATED WITH ILL-CONDITIONED ROOTS SHOWING THE EFFECT
      OF LIMITING PRECISION IN THE COMPUTATION OF FNC(X).
    2 MEANS POSSIBLE NUMERICAL CONVERGENCE.BEHAVIOR INDICATES
      LIMITING PRECISION CALCULATIONS AS A RESULT OF USER ASKING
      FOR TOO MUCH ACCURACY OR ELSE CONVERGENCE IS VERY SLOW.
      RESIDUAL NORMS AND SOLUTION INCREMENT NORMS HAVE
      REMAINED ROUGHLY CONSTANT OVER NSRRC CONSECUTIVE
      ITERATIONS. (SOSNLE SETS NSRRC=5.)
    3 MEANS THE ALLOWABLE NUMBER OF ITERATIONS HAS BEEN MET
      WITHOUT OBTAINING A SOLUTION TO THE SPECIFIED ACCURACY.
      VERY SLOW CONVERGENCE IS INDICATED.
      (SOSNLE SETS THE MAXIMUM NUMBER OF ITERATIONS MXIT=100)
    4 MEANS THE ALLOWABLE NUMBER OF ITERATIONS HAS BEEN MET.
      ITERATIVE PROCESS MAY NOT BE CONVERGING SINCE THE FIRST
      SOLUTION INCREMENT NORM,FOLLOWING THE ATTAINMENT OF
      MINIMUM RESIDUAL NORM,EXCEEDS THE PREVIOUS SOLUTION
      INCREMENT NORM BY A FACTOR OF 10. A LOCAL MINIMUM MAY
      HAVE BEEN ENCOUNTERED.
    5 MEANS THAT THE ITERATIVE SCHEME APPEARS TO BE DIVERGING.
      RESIDUAL NORMS AND SOLUTION INCREMENT NORMS HAVE
      INCREASED OVER NSRI CONSECUTIVE ITERATIONS. (SOSNLE
      SETS NSRI=5.)
    6 MEANS THAT A JACOBIAN-RELATED MATRIX WAS SINGULAR,
      PROCESS CANNOT BE CONTINUED.
    7 MEANS IMPROPER INPUT PARAMETERS.
    *** IFLG SHOULD BE EXAMINED AFTER EACH CALL TO      ***
    *** SOSNLE WITH THE APPROPRIATE ACTION BEING TAKEN. ***
```

```
SPLIFT    SPLIFT    SPLIFT    SPLIFT         SPLIFT    SPLIFT    SPLIFT
          ****************************************
               ******************************
                    ********************
                         **********
```

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

      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)
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 SMOO 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 CRDER) THAT DEFINE THE
           SPLINE.  USUALLY X IS THE SAME AS X IN SPLIFT OR SMOO.
```

```
      Y   - ARRAY OF ORDINATES THAT DEFINE THE SPLINE.  USUALLY Y IS
            THE SAME AS Y IN SPLIFT OR AS R IN SMOO.
      YPP - ARRAY OF SECOND DERIVATIVES THAT DEFINE THE SPLINE.
            USUALLY YPP IS THE SAME AS YPP IN SPLIFT OR R2 IN SMOO.
      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)
   THIS ROUTINE WAS WRITTEN BY M. K. GORDON

   ABSTRACT

   SUBROUTINE SPLIQ INTEGRATES A CUBIC SPLINE (GENERATED BY
   SPLIFT, SMOO, 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 SMOO.
      Y   - ARRAY OF ORDINATES THAT DEFINE THE SPLINE.  USUALLY Y IS
            THE SAME AS Y IN SPLIFT OR AS R IN SMOO.
      YP  - ARRAY OF FIRST DERIVATIVES OF THE SPLINE AT ABSCISSAS.
            USUALLY YP IS THE SAME AS YP IN SPLIFT OR R1 IN SMOO.
      YPP - ARRAY OF SECOND DERIVATIVES THAT DEFINE THE SPLINE.
            USUALLY YPP IS THE SAME AS YPP IN SPLIFT OR R2 IN SMOO.
```

```
      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,KFLAG)
      WRITTEN BY RONDALL E JONES
      MODIFIED BY JOHN A. WISNIEWSKI TO USE THE SINGLETON QUICKSORT
      ALGORITHM. DATE 18 NOVEMBER 1976.

      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
          QUICKSORT ALGORITHM IS USED.

      REFERENCE
          SINGLETON,R.C., ALGORITHM 347, AN EFFICIENT ALGORITHM FOR
          SORTING WITH MINIMAL STORAGE, CACM,12(3),1969,185-7.

      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
          KFLAG - 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.
```

```
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)
   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 EQUATICNS 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 CONTRCL 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(*) -- SOLUTICN 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),P(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.

```
STFODE    STFODE    STFODE    STFODE          STFODE    STFODE    STFODE
                ****************************************
                    ******************************
                        ******************
                            **********
      SUBROUTINE STFODE(F,NEQN,Y,T,TFIN,EPSREL,EPSABS,IFLAG,IWORK,WORK,
     1                  NWRKD)

      WRITTEN BY B. L. HULME AND S. L. DANIEL.

  ABSTRACT.
           STFODE SOLVES FIRST ORDER SYSTEMS OF STIFF ORDINARY DIFFEREN-
      TIAL EQUATIONS OF THE FORM
                    DY/DT = F(T,Y(1),Y(2),...,Y(NEQN))
      BY COLLOCATION METHODS.  IT IS INEFFICIENT FOR NON-STIFF PROBLEMS.
           STFODE INTEGRATES FROM T TO TFIN RETURNING WITH T=TFIN,Y=Y(T)
      AND ALL PARAMETERS SET FOR CONTINUATION.  THE USER MAY SIMPLY
      CHANGE TFIN AND CALL STFODE AGAIN.
           SUBROUTINES COLODE AND TWOSTP ARE USED BY STFODE.  COLODE IS
      ASKED TO SELECT AN A-STABLE COLLOCATION METHOD OF FIXED ORDER AND
      TO SUPERVISE THE USE OF TWOSTP, WHICH ADVANCES THE SOLUTION BY TWO
      STEPS OF SIZE H.  TWOSTP CHOOSES THE INITIAL STEP SIZE AND VARIES
      H FOR EACH PAIR OF STEPS TO KEEP THE ESTIMATED LOCAL ERROR PER
      STEP SMALLER THAN THE SPECIFIED TOLERANCE.
           THE USER HAS THE OPTION OF INTERMEDIATE RETURNS AFTER EACH
      PAIR OF STEPS WITH T=T+2H, Y=Y(T) AND Y(T-H) LOCATED IN WORK(4),
      WORK(5),...,WORK(3+NEQN).  THIS IS HELPFUL IN SELECTING OUTPUT
      POINTS FOR LATER RUNS BASED ON THE BEHAVIOR OF THE SOLUTION.
           THE USER ALSO HAS THE OPTION OF TELLING STFODE WHEN THE
      JACOBIAN OF F WITH RESPECT TO Y IS CONSTANT.  THIS WILL REDUCE F
      EVALUATIONS BY A FACTOR OF 3/(NEQN+6).
           COLODE CAN BE USED DIRECTLY FOR SOLVING STIFF EQUATIONS.  IT
      PROVIDES THE ADDITIONAL OPTIONS OF SUPPLYING A JACOBIAN SUBROUTINE
      AND PICKING THE COLLOCATION METHOD FROM AMONG THOSE USING N
      GAUSS-LEGENDRE OR RADAU POINTS, N=1,2,...,12.

  DUMMY ARGUMENTS.
      F         - THE NAME OF THE DERIVATIVE SUBROUTINE.
      NEQN      - THE NUMBER OF EQUATIONS IN THE SYSTEM.
      Y(*)      - THE APPROXIMATE SOLUTION VECTOR AT T.
      T         - THE INDEPENDENT VARIABLE.
      TFIN      - THE FINAL POINT OF THE INTEGRATION INTERVAL.
      EPSREL    - THE RELATIVE LOCAL ERROR TOLERANCE PER STEP.
      EPSABS    - THE ABSOLUTE LOCAL ERROR TOLERANCE PER STEP.
      IFLAG     - THE FLAG FOR COMMUNICATION BETWEEN STFODE AND USER.
      IWORK(*)  - THE INTEGER WORK VECTOR.
      WORK(*)   - THE REAL WORK VECTOR.
      NWRKD     - THE DIMENSION OF WORK.

  ACTUAL ARGUMENTS.
           THE ONLY ARGUMENTS WHICH MAY BE CONSTANTS OR EXPRESSIONS ARE
      THOSE CORRESPONDING TO NEQN, TFIN AND NWRKD.  ALL OTHERS MUST BE
      NAMES.

  FIRST CALL OF STFODE.
           DECLARE F IN AN EXTERNAL STATEMENT, AND SUPPLY A SUBROUTINE
      F(T,Y,YP) WHICH STORES DY(J)/DT IN YP(J).  DIMENSION THE VECTORS
              Y(NEQN), IWORK(7*NEQN+3), WORK(NWRKD),
      WHERE NWRKD IS THE MAXIMUM AMOUNT OF STORAGE AVAILABLE WITHIN
      6*NEQN**2+16*NEQN+15 .LE. NWRKD .LE. 56*NEQN**2+41*NEQN+115.
      THE MORE WORK STORAGE, THE MORE FLEXIBILITY COLODE HAS IN THE
      SELECTION OF AN EFFICIENT METHOD.
           IF INTERMEDIATE RETURNS ARE DESIRED AFTER EACH PAIR OF STEPS,
      DIMENSION YTMH(NEQN) AND EQUIVALENCE (H,WORK(1)),(YTMH,WORK(4))
      SO THAT YTMH(*)=Y(T-H) IS AVAILABLE AS WELL AS Y(*)=Y(T).
           INITIALIZE THE ARGUMENTS
      NEQN   = THE NUMBER OF EQUATIONS.
      Y(*)   = THE INITIAL VALUES.
      T      = THE INITIAL POINT.
      TFIN   = THE FINAL POINT.
      EPSREL = THE RELATIVE LOCAL ERROR TOLERANCE PER STEP.
      EPSABS = THE ABSOLUTE LOCAL ERROR TOLERANCE PER STEP.
               NOTE. A PAIR OF STEPS IS ACCEPTED ONLY IF THE ESTIMATED
                     LOCAL ERROR IN Y(J) .LE. EPSREL*ABS(Y(J))+EPSABS.
```

```
      IFLAG  = 1, VARIABLE JACOBIAN AND RETURN ONLY WHEN T=TFIN.
             = 0, VARIABLE JACOBIAN AND RETURN AFTER EACH PAIR OF STEPS.
             =-1, CONSTANT JACOBIAN AND RETURN ONLY WHEN T=TFIN.
             =-2, CONSTANT JACOBIAN AND RETURN AFTER EACH PAIR OF STEPS.
      NWRKD  = THE DIMENSION OF WORK.

   OUTPUT.
      Y(*)   = THE APPROXIMATE SOLUTION AT T.
      T      = THE END OF THE LAST SUCCESSFUL PAIR OF STEPS.
      EPSREL = ABSOLUTE VALUE OF THE INITIAL EPSREL NORMALLY.
               IF IFLAG=4 AND EPSREL.NE.0., THEN EPSREL=1.42E-14.
      EPSABS = ABSOLUTE VALUE OF THE INITIAL EPSABS NORMALLY.
               IF IFLAG=4 AND EPSREL.EQ.0., THEN EPSABS=1.42E-14*NORM Y.
      IFLAG  = 2, THE INTEGRATION REACHED T=TFIN.  TO CONTINUE CHANGE
                  TFIN AND CALL STFODE AGAIN.
             = 3, THE INTEGRATION ADVANCED TWO STEPS, BUT T.NE.TFIN.
                  THIS VALUE IS POSSIBLE ONLY WHEN IFLAG=0,-2 INITIALLY.
                  CALL STFODE AGAIN TO CONTINUE.
             = 4, WARNING, EITHER EPSREL OR EPSABS HAS BEEN INCREASED TO
                  TWO UNITS OF ROUNDOFF.  CALL STFODE AGAIN TO CONTINUE.
             = 5, WARNING, H HAS DECREASED BY A FACTOR OF 1.E-05 SINCE
                  THE LAST START OR RESTART, BUT THERE ARE NO OTHER
                  DIFFICULTIES.  CALL STFODE AGAIN TO OBTAIN A RESTART,
                  POSSIBLY CHANGING TFIN.
             = 6, CONVERGENCE AND/OR LOCAL ERROR FAILURES CONTINUE TO
                  OCCUR EVEN AFTER H HAS BEEN REDUCED BY A FACTOR OF
                  1.E-05.  THE USER MUST CHANGE THE TOLERANCE AND/OR
                  NWRKD AND REINITIALIZE IFLAG BEFORE RECALLING STFODE.
             = 7, THE INTEGRATION CANNOT PROCEED BECAUSE OF EITHER
                  ILLEGAL INPUT OR A SINGULAR MATRIX.  SEE THE DIAGNOS-
                  TIC MESSAGE.
      WORK(1)= THE STEP SIZE H.
      WORK(4,5,...,3+NEQN)= APPROXIMATE SOLUTION AT T-H, IF IFLAG=2,3.

   SUBSEQUENT CALLS OF STFODE.
         WHEN IFLAG=2,3,4,5, THE USER MAY CALL STFODE AGAIN WITHOUT
      CHANGING IFLAG.  WHEN IFLAG=6,7, IFLAG MUST BE REINITIALIZED
      BEFORE STFODE IS CALLED AGAIN.
```

```
SUDS     SUDS     SUDS     SUDS     SUDS        SUDS     SUDS     SUDS     SUDS
                 ****************************************
                     ****************************
                         ********************
                             **********
      SUBROUTINE SUDS(A,X,B,NEQ,NUK,NRDA,IFLAG,MLSC,WORK,IWORK)

      SUDS SOLVES THE UNDERDETERMINED SYSTEM OF LINEAR EQUATIONS A Z = B
      WHERE A IS NEQ BY NUK AND NEQ .LE. NUK. IN PARTICULAR,IF RANK A
      EQUALS IRA,A VECTOR X AND A MATRIX U ARE DETERMINED SUCH THAT
      X IS THE UNIQUE SOLUTION OF SMALLEST LENGTH,SATISFYING A X = B,
      AND THE COLUMNS OF U FORM AN ORTHONORMAL BASIS FOR THE NULL
      SPACE OF A,SATISFYING A U = 0 . THEN ALL SOLUTIONS Z ARE
      GIVEN BY
               Z = X + C(1)*U(1) + ..... + C(NUK-IRA)*U(NUK-IRA)
      WHERE U(J) REPRESENTS THE J-TH COLUMN OF U AND THE C(J) ARE
      ARBITRARY CONSTANTS.
      IF THE SYSTEM OF EQUATIONS ARE NOT COMPATIBLE,ONLY THE LEAST
      SQUARES SOLUTION OF MINIMAL LENGTH IS COMPUTED.
      SUDS IS AN INTERFACING ROUTINE WHICH CALLS SUBROUTINE LSSUDS
      FOR THE SOLUTION. LSSUDS IN TURN CALLS SUBROUTINE ORTHOR AND
      POSSIBLY SUBROUTINE OHTROL FOR THE DECOMPOSITION OF A BY
      ORTHOGONAL TRANSFORMATIONS. IN THE PROCESS,ORTHOR CALLS UPON
      SUBROUTINE CSCALE FOR SCALING.

      WRITTEN BY H.A. WATTS , ORG. 2642 , SANDIA LABORATORIES
   REFERENCE
      H.A.WATTS,SOLVING LINEAR LEAST SQUARES PROBLEMS USING
```

```
      SODS/SUDS/CODS , SANDIA REPORT SAND77-0683

***********************************************************************
   INPUT
***********************************************************************

      A -- CONTAINS THE MATRIX OF NEQ EQUATIONS IN NUK UNKNOWNS AND MUST
           BE DIMENSIONED NRDA BY NUK. THE ORIGINAL A IS DESTROYED.
      X -- SOLUTION ARRAY OF LENGTH AT LEAST NUK
      B -- GIVEN CONSTANT VECTOR OF LENGTH NEQ, B IS DESTROYED
      NEQ -- NUMBER OF EQUATIONS, NEQ GREATER OR EQUAL TO 1
      NUK -- NUMBER OF COLUMNS IN THE MATRIX (WHICH IS ALSO THE NUMBER
             OF UNKNOWNS), NUK NOT SMALLER THAN NEQ
      NRDA -- ROW DIMENSION OF A, NRDA GREATER OR EQUAL TO NEQ
      IFLAG -- STATUS INDICATOR
            =0  FOR THE FIRST CALL (AND FOR EACH NEW PROBLEM DEFINED BY
                A NEW MATRIX A) WHEN THE MATRIX DATA IS TREATED AS EXACT
            =-K FOR THE FIRST CALL (AND FOR EACH NEW PROBLEM DEFINED BY
                A NEW MATRIX A) WHEN THE MATRIX DATA IS ASSUMED TO BE
                ACCURATE TO ABOUT K DIGITS
            =1  FOR SUBSEQUENT CALLS WHENEVER THE MATRIX A HAS ALREADY
                BEEN DECOMPOSED (PROBLEMS WITH NEW VECTORS B BUT
                SAME MATRIX A CAN BE HANDLED EFFICIENTLY)
      MLSO -- =0 IF ONLY THE MINIMAL LENGTH SOLUTION IS WANTED
              =1 IF THE COMPLETE SOLUTION IS WANTED, INCLUDES THE
                 LINEAR SPACE DEFINED BY THE MATRIX U IN THE ABSTRACT
      WORK(*),IWORK(*) -- ARRAYS FOR STORAGE OF INTERNAL INFORMATION,
                 WORK MUST BE DIMENSIONED AT LEAST
                        NUK + 3*NEQ + MLSO*NUK*(NUK-RANK A)
                 WHERE IT IS POSSIBLE FOR   0 .LE. RANK A .LE. NEQ
                 IWORK MUST BE DIMENSIONED AT LEAST   3 + NEQ
      IWORK(2) -- SCALING INDICATOR
                  =-1 IF THE MATRIX IS TO BE PRE-SCALED BY
                  COLUMNS WHEN APPROPRIATE
                  IF THE SCALING INDICATOR IS NOT EQUAL TO -1
                  NO SCALING WILL BE ATTEMPTED
                FOR MOST PROBLEMS SCALING WILL PROBABLY NOT BE NECESSARY

***********************************************************************
   OUTPUT
***********************************************************************

      IFLAG -- STATUS INDICATOR
             =1 IF SOLUTION WAS OBTAINED
             =2 IF IMPROPER INPUT IS DETECTED
                (EXECUTION TERMINATES UNLESS A PRIOR CALL TO
                 ERXSET WAS MADE)
             =3 IF RANK OF MATRIX IS LESS THAN NEQ
                TO CONTINUE SIMPLY RESET IFLAG=1 AND CALL SUDS AGAIN
                (THE USER MUST MAKE A PRIOR CALL TO THE ERXSET
                 ROUTINE IF THIS RETURN IS TO BE NONFATAL WHEN
                 THE INPUT MODE IFLAG=0 IS USED. THIS IS NOT
                 NECESSARY WHEN THE INPUT MODE IFLAG=-K IS USED.)
             =4 IF THE SYSTEM OF EQUATIONS APPEARS TO BE INCONSISTENT.
                HOWEVER,THE LEAST SQUARES SOLUTION OF MINIMAL LENGTH
                WAS OBTAINED.
      X -- MINIMAL LENGTH LEAST SQUARES SOLUTION OF  A X = B
      A -- CONTAINS THE STRICTLY UPPER TRIANGULAR PART OF THE REDUCED
           MATRIX AND TRANSFORMATION INFORMATION
      WORK(*),IWORK(*) -- CONTAINS INFORMATION NEEDED ON SUBSEQUENT
                          CALLS (IFLAG=1 CASE ON INPUT) WHICH MUST NOT
                          BE ALTERED.
                          THE MATRIX U DESCRIBED IN THE ABSTRACT IS
                          STORED IN THE  NUK*(NUK-RANK A) ELEMENTS OF
                          THE WORK ARRAY BEGINNING AT WORK(1+NUK+3*NEQ).
                          HOWEVER U IS NOT DEFINED WHEN MLSO=0 OR
                          IFLAG=4.
                          IWORK(1) CONTAINS THE NUMERICALLY DETERMINED
                          RANK OF THE MATRIX A

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

```
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)

      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 PDP-10
               U = 7.1E-15    FOR CDC-6600
               U = 2.2E-16    FOR IBM 360/370 DOUBLE PRECISION
**************************************************************************
**************************************************************************
```

SVA SVA SVA SVA SVA SVA SVA SVA SVA SVA

```
                  ****************************************
                       ***********************************
                            ******************************
                                 *************************
                                      ********************
                                           ***************
     SUBROUTINE SVA (A,MDA,M,N,MDATA,B,SING,NAMES,ISCALE,D)
     DIMENSION A(MDA,N),B(M),SING(3*N),NAMES(N),D(N)

     WRITTEN BY C. L. LAWSON AND R. J. HANSON, FROM THE BOOK SOLVING
     LEAST SQUARES PROBLEMS, PRENTICE-HALL, INC. (1974).  FOR AN
     EXAMPLE OF THE ANALYSIS OF A LEAST SQUARES PROBLEM USING SVA, SEE
     CHAPTER 26.

     ABSTRACT

     THE SUBROUTINE SVA USES SUBROUTINE SVDRS TO OBTAIN THE SINGULAR
     VALUE DECOMPOSITION OF A COLUMN SCALED MATRIX C = AD.

     THE SUBROUTINE SVA PRINTS QUANTITIES DERIVED FROM THIS
     DECOMPOSITION TO PROVIDE THE USER WITH INFORMATION USEFUL TO THE
     UNDERSTANDING OF THE GIVEN LEAST SQUARES PROBLEM AX=B.

     THE USER PROVIDES AN M BY N MATRIX A AND AN M-VECTOR B, DEFINING A
```

LEAST SQUARES PROBLEM AX=B. THE USER SELECTS ONE OF THREE OPTIONS REGARDING COLUMN SCALING DESCRIBED BELOW IN THE DEFINITION OF THE PARAMETER ISCALE. THIS SELECTION DEFINES AN N BY N DIAGONAL MATRIX D.

INTRODUCING THE CHANGE OF VARIABLES

$$X = DY$$

THE SUBROUTINE PERFORMS A SINGULAR VALUE ANALYSIS OF THE LEAST SQUARES PROBLEM

$$CY = B$$

WHERE

$$C = AD$$

THE SUBROUTINE SVA USES THE SUBROUTINE SVDRS TO COMPUTE THE SINGULAR VALUE DECOMPOSITION

$$CV = US$$

AND THE TRANSFORMED RIGHT-SIDE VECTOR G DEFINED BY

$$G = (\text{TRANSPOSE OF } U)B$$

LET K DENOTE THE INDEX OF THE LAST NONZERO SINGULAR VALUE OF C, AND DEFINE L = MIN(M,N).

COMPUTE THE VECTOR P WITH COMPONENTS DEFINED BY

COMPONENT I OF P =

(COMPONENT I OF G)/(SINGULAR VALUE I),

I = 1,...,K

COMPONENT I OF P = 0,

I = K+1,...,N

THE J-TH CANDIDATE SOLUTION VECTOR IS DEFINED AS

$$Y = VQ$$
$$X = DY,\ J = 1,\ldots,L$$

HERE Q DEPENDS ON J AND IS THE SAME AS THE VECTOR P IN COMPONENTS 1 THROUGH J, WITH THE OTHER COMPONENTS 0.

ASSOCIATED WITH THE J-TH CANDIDATE SOLUTION VECTOR IS THE J-TH SUM OF SQUARES OF RESIDUALS

$$RS(J+1) = \sum_{I=J+1}^{M} (\text{COMPONENT I OF G})^{**}2$$

FOR VALUES OF J BETWEEN 0 AND K.

IT IS POSSIBLE THAT THE M BY (N+1) DATA MATRIX (A B) PROVIDED TO THIS SUBROUTINE MAY BE A COMPRESSED REPRESENTATION OF A LEAST SQUARES PROBLEM INVOLVING MORE THAN M ROWS. FOR EXAMPLE, (A B) MAY BE THE TRIANGULAR MATRIX RESULTING FROM SEQUENTIAL HOUSEHOLDER TRIANGULARIZATION OF A LARGE SET OF DATA. THE USER PROVIDES AN INTEGER MDATA THAT SPECIFIES THE NUMBER OF ROWS OF DATA IN THE ORIGINAL PROBLEM. OF COURSE, IF (A B) IS THE ORIGINAL DATA, THEN MDATA AND M SHOULD BE SET TO THE SAME VALUE. THE NUMBER MDATA IS USED IN COMPUTING

$$SG(J+1) = SQRT(RS(J+1)/MAX(1,MDATA-J))$$

FOR VALUES OF J BETWEEN 0 AND K.

UNDER APPROPRIATE STATISTICAL HYPOTHESES ON THE DATA (A B), THE

NUMBER SG(J+1) CAN BE INTERPRETED AS AN UNBIASED ESTIMATE OF THE STANDARD DEVIATION OF ERRORS IN THE DATA VECTOR B.

PRINTED OUTPUT..

THIS SUBROUTINES PRINTS THE FOLLOWING..

1. THE QUANTITIES M,N,MDATA,IDENTIFICATION OF THE SCALING OPTION USED, AND THE DIAGONAL ELEMENTS OF THE SCALING MATRIX D.

2. THE MATRIX V, MULTIPLIED BY 10,000 TO FACILITATE SCANNING FOR LARGE AND SMALL ELEMENTS.

3. (A) THE J-TH SINGULAR VALUE, FOR J=1,...,L.
   (B) THE J-TH ENTRY OF THE VECTOR P, THE RECIPROCAL OF THE J-TH SINGULAR VALUE, THE J-TH COMPONENT OF THE VECTOR G, AND ITS SQUARE, FOR J=1,...,K.
   (C) THE QUANTITIES RS(J+1) AND SG(J+1) FOR J=0,...,K.

4. THE QUANTITIES J, THE LENGTH OF THE J-TH Y SOLUTION VECTOR, SQRT(RS(J+1)), ALOG10(LENGTH OF THE J-TH Y SOLUTION VECTOR), AND ALOG10(SQRT(RS(J+1))), FOR J=1,...,K.

5. (THE USER MAY OFTEN IGNORE THESE QUANTITIES, WHICH MAY BE USED FOR LEVENBERG-MARQUARDT ANALYSIS.) THE QUANTITIES LAMBDA,SQRT(EQ. 25.47), SQRT(EQ. 25.48),ALOG10(LAMBDA),ALOG10(SQRT(EQ. 25.47)),ALOG10(SQRT(EQ. 25.48)). THESE QUANTITIES ARE PRINTED FOR 21 VALUES OF LAMBDA, RANGING FROM 10*(LARGEST SINGULAR VALUE) TO (K-TH SINGULAR VALUE)/10, IN INCREMENTS THAT ARE UNIFORMLY SPACED IN ALOG10(LAMBDA). THE ABOVE EQUATION NUMBERS REFER TO THE BOOK, SOLVING LEAST SQUARES PROBLEMS, CHAPTER 25.

6. THE J-TH CANDIDATE SOLUTION X, FOR J=1,...,K.

THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST.. A(MDA,N),B(M),SING(3*N),NAMES(N),D(N). SEE THE ARRAY NAMES (*) DESCRIPTION BELOW FOR A POSSIBLE EXCEPTION TO THIS.

THE PARAMETERS FOR SVA ARE

INPUT..

A(*,*),MDA,M,N	THE ARRAY A(*,*) INITIALLY CONTAINS THE M BY N MATRIX A OF THE LEAST SQUARES PROBLEM AX = B. EITHER M.GE.N OR M.LT.N IS PERMITTED. THE FIRST DIMENSIONING PARAMETER OF THE ARRAY A(*,*) IS MDA, WHICH MUST SATISFY MDA .GE. MAX(M,N). THE CONDITION MDA.LT.MAX(M,N) IS CONSIDERED AN ERROR.
MDATA	THE NUMBER OF ROWS IN THE ORIGINAL LEAST SQUARES PROBLEM, WHICH MAY HAVE HAD MORE ROWS THAN (A B). THIS NUMBER IS USED ONLY IN COMPUTING THE NUMBERS SG(J), J=0,...,K.
B(*)	THE ARRAY B(*) INITIALLY CONTAINS THE M-VECTOR B OF THE LEAST SQUARES PROBLEM AX = B.
SING(*)	THE LOCATIONS, (SING(I), I=1,...,3*N), ARE USED AS TEMPORARY WORKING SPACE.
NAMES(*)	IN LOCATION NAMES(I) THE USER MAY STORE AN ALPHA-NUMERIC NAME AS AN IDENTIFIER FOR THE I-TH COMPONENT OF THE SOLUTION VECTOR, (I = 1,...,N). THESE NAMES WILL BE PRINTED WITH THE APPROPRIATE COMPONENTS OF THE V MATRIX AND WITH THE CANDIDATE SOLUTIONS. EACH NAME IS PRINTED WITH AN A6 FORMAT. IF NAMES(1) HAS THE SAME CONTENTS AS THE QUANTITY IBLANK, DEFINED BY DATA IBLANK/1H / THEN THE REMAINING WORDS OF THE ARRAY NAMES(*)

WILL BE IGNORED AND NO IDENTIFYING NAMES WILL BE PRINTED.

ISCALE,D(*) — THE USER SETS ISCALE = 1,2, OR 3. IF ISCALE = 1, THE SUBROUTINE FUNCTIONS AS THOUGH THE SCALING MATRIX D WERE AN N BY N IDENTITY MATRIX. IN THIS CASE THE SUBROUTINE MAKES NO REFERENCE TO THE ARRAY D(*).

IF ISCALE = 2, THE SUBROUTINE COMPUTES THE LENGTH OF THE J-TH COLUMN VECTOR OF A AND SETS D(J) EQUAL TO THE RECIPROCAL OF THIS LENGTH, IF NONZERO, AND D(J) = 1, IF ZERO. IN THIS CASE ALL NONZERO COLUMNS OF THE SCALED MATRIX C = AD WILL HAVE UNIT EUCLIDEAN LENGTH.

IF ISCALE = 3, THE SUBROUTINE WILL USE THE USER-SUPPLIED VALUES D(J),J = 1,...,N AS THE DIAGONAL ELEMENTS OF THE DIAGONAL SCALING MATRIX D. IN THIS CASE THE USER MUST ASSIGN VALUES TO D(J), J = 1,...,N, AND THE SUBROUTINE WILL NOT MODIFY THESE VALUES. FOR EXAMPLE, THE USER MIGHT SET D(J) EQUAL TO THE A PRIORI STANDARD DEVIATION OF THE J-TH COMPONENT OF THE SOLUTION VECTOR IF SUCH INFORMATION IS KNOWN. AS A FURTHER EXAMPLE, THE USER MIGHT SET D(J) EQUAL TO THE RECIPROCAL OF THE MEAN OF THE M COMPONENTS OF THE J-TH COLUMN VECTOR S UNCERTAINTY. THIS TYPE OF SCALING HAS THE EFFECT OF MAKING THE UNCERTAINTY OF EVERY COLUMN THE SAME MAGNITUDE IN THE MATRIX C.

OUTPUT..

A(*,*) — ON OUTPUT, THE J-TH CANDIDATE SOLUTION WILL BE STORED IN THE FIRST N LOCATIONS OF THE J-TH COLUMN OF A(*,*) FOR J = 1,...,MIN(M,N).

B(*) — ON OUTPUT THE M-VECTOR G IS STORED IN THE ARRAY B(*).

SING(*) — ON OUTPUT THE SINGULAR VALUES OF THE SCALED MATRIX C ARE STORED IN SING(I),I = 1,..., MIN(M,N).

SVDRS SVDRS SVDRS SVDRS SVDRS SVDRS SVDRS SVDRS

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

```
SUBROUTINE SVDRS (A,MDA,MM,NN,B,MDB,NB,S)
SUBROUTINE SVDRS (A,MDA,M,N,B,MDB,NB,S)
DIMENSION A(MDA,N),(B(MDB,NB) OR B(M)),S(3*N)
```

WRITTEN BY C. L. LAWSON AND R. J. HANSON, FROM THE BOOK SOLVING LEAST SQUARES PROBLEMS, PRENTICE-HALL, INC. (1974). FOR FURTHER ALGORITHMIC DETAILS SEE CHAPTER 18.

ABSTRACT

GIVEN AN M BY N MATRIX A AND AN M BY NB MATRIX B, THIS SUBROUTINE COMPUTES THE SINGULAR VALUES OF A AND ALSO COMPUTES AUXILIARY QUANTITIES USEFUL IN ANALYZING AND SOLVING THE MATRIX LEAST SQUARES PROBLEM AZ = B. DENOTE THE SINGULAR VALUE DECOMPOSITION OF A BY

AV = US

IN THIS DECOMPOSITION THE N BY N AND M BY M MATRICES V AND U ARE ORTHOGONAL. THAT IS, V SATISFIES

(TRANSPOSE OF V) V =
V (TRANSPOSE OF V) = I

WHERE I IS THE N BY N IDENTITY MATRIX. A SIMILAR RELATION HOLDS FOR THE MATRIX U. THE M BY N MATRIX S IS DIAGONAL. THAT IS, EVERY ELEMENT OF S IS ZERO EXCEPT POSSIBLY ON THE MAIN DIAGONAL.

THIS SUBROUTINE COMPUTES V,S AND G, THE PRODUCT OF THE TRANSPOSE OF THE MATRIX U AND THE RIGHT-HAND SIDE B.

TO COMPLETE THE SOLUTION OF THE LEAST SQUARES PROBLEM AX = B, THE USER MUST FIRST DECIDE WHICH SMALL SINGULAR VALUES ARE TO BE TREATED AS ZERO. LET T DENOTE THE MATRIX OBTAINED BY TRANSPOSING S AND RECIPROCATING THE SIGNIFICANT SINGULAR VALUES AND SETTING THE OTHERS TO ZERO. THEN THE SOLUTION MATRIX X CAN BE OBTAINED BY COMPUTING P = TG AND X = VP.

EITHER M.GE.N OR M.LT.N IS PERMITTED. NOTE THAT IF B = I, THEN X IS THE PSEUDOINVERSE OF A.

THE USER MUST DIMENSION ALL ARRAYS APPEARING IN THE CALL LIST.. A(MDA,N), (B(MDB,NB) OR B(M)), S(3*N). THIS ALLOWS FOR A SOLUTION OF A RANGE OF PROBLEMS IN THE GIVEN WORKING SPACE.

THE PARAMETERS FOR SVDRS ARE

INPUT..

A(*,*),MDA,M,N	THE ARRAY A(*,*) IS DOUBLY SUBSCRIPTED WITH FIRST DIMENSIONING PARAMETER EQUAL TO MDA. THE ARRAY A(*,*) INITIALLY CONTAINS THE M BY N MATRIX A. EITHER M.GE.N OR M.LT.N IS PERMITTED. THE FIRST DIMENSIONING PARAMETER MDA MUST SATISFY MDA.GE.MAX(M,N). THE CONDITION MDA.LT.MAX(M,N) IS CONSIDERED AN ERROR.
B(*),MDB,NB	NB DENOTES THE NUMBER OF COLUMN VECTORS IN THE MATRIX B. IF NB = 0 THE ARRAY B(*) WILL NOT BE REFERENCED BY THIS SUBROUTINE NOR WILL ITS CONTENTS BE MODIFIED. IF NB.GE.2, THE ARRAY B SHOULD BE DOUBLE SUBSCRIPTED WITH FIRST DIMENSIONING PARAMETER EQUAL TO MDB.GE.M. IF NB = 1, THEN B WILL BE USED AS A SINGLY SUBSCRIPTED ARRAY OF LENGTH M. IN THIS LATTER CASE THE VALUE OF MDB IS ARBITRARY BUT IT SHOULD BE SET TO A DEFINED VALUE SUCH AS MDB = M. THE CONTENTS OF THE ARRAY B(*) MUST CONTAIN THE MATRIX B. THE CONDITION NB.GT.1 AND. MDB.LT.M IS CONSIDERED AN ERROR.
S(*)	THE ARRAY S(*) IS USED AS 3*N LOCATIONS OF TEMPORARY WORKING SPACE BY THE SUBROUTINE.

OUTPUT..

A(*,*)	ON OUTPUT THE ARRAY A(*,*) CONTAINS THE N BY N ORTHOGONAL MATRIX V.
B(*)	ON OUTPUT THE ARRAY B(*),(OR B(*,*)), CONTAINS THE PRODUCT OF THE TRANSPOSE OF THE M BY M MATRIX U AND THE RIGHTHAND SIDE B.
S(*)	ON OUTPUT THE FIRST N LOCATIONS OF S(*) CONTAIN THE DIAGONAL TERMS OF THE M BY N DIAGONAL MATRIX S. THESE NONNEGATIVE TERMS ARE ORDERED FROM LARGEST TO SMALLEST. IF M.LT.N, ENTRIES M+1,...,N OF S(*) ARE ZERO.

THA THA THA THA THA THA THA THA THA THA

```
      SUBROUTINE THA(H,A,REL,ANS)

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

      REFERENCE SLA-73-0333

      ABSTRACT
          SUBROUTINE THA COMPUTES THE T(H,A) INTEGRAL OF OWEN,
          T(H,A)=INTEGRAL ON (0,A) OF EXP(-H*H*(1+X*X)/2)/(1+X*X).
          THE IDENTITIES
                       T(-H,A)=T(H,A) ,  T(H,-A)=-T(H,A)
          T(H,A)=G(U)+G(V)-4.0*G(U)*G(V)-T(V,1/A)   ,   A.GT.1
                    G(X)=0.25*(1+ERF(X/SQRT(2))
                            U=H , V=A*H
          AND THEIR VARIANTS ARE USED TO KEEP THE VARIABLES POSITIVE
          AND LIMIT THE INTERVAL OF INTEGRATION TO A SUBSET OF (0,1).
          THE INTEGRATION IS PERFORMED BY GAUS8 ON FUNCTION FTHA.  THE
          CASES H=0 AND A=0 ARE EVALUATED EXPLICITLY.

      DESCRIPTION OF ARGUMENTS

          INPUT
            H       - ARGUMENT, UNRESTRICTED
            A       - ARGUMENT, UNRESTRICTED
            REL     - RELATIVE ERROR PARAMETER FOR GAUS8
                      REL=5.E-(S+1) FOR S SIGNIFICANT DIGITS
          OUTPUT
            ANS     - A VALUE FOR T(H,A)

      ERROR CONDITIONS
          ERROR MESSAGE FROM GAUS8 - A FATAL ERROR
```

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 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  KDRV    .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 PTP=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   NO 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.
-----
-----
-----
```

```
XPPLOT    XPPLOT    XPPLOT    XPPLOT         XPPLOT    XPPLOT    XPPLOT
          ****************************************
              ******************************
                  ******************
                      **********
```

```
      SUBROUTINE XPPLOT(ISIZE,IPAGE,IBOX,ILABEL,IMARK,IAXES,ITYPE,RANGE,
     1             NUM,X1,Y1,N1,L1,X2,Y2,N2,L2,X3,Y3,N3,L3,X4,Y4,N4,L4)

      ABSTRACT
         XPPLOT PLOTS ONE TO FOUR CURVES ON A SINGLE PRINTER-TYPE PLOT.
      THE PHYSICAL SIZE OF THE PLOT MAY BE VARIED AS APPROPRIATE FOR
      PRINTER OR TERMINAL OUTPUT.  OPTIONS ARE SUPPLIED FOR AXIS
      DRAWING AND LABELLING, SURROUNDING THE PLOT AREA WITH A BOX,
      SPECIFYING SPECIAL PLOT LIMITS, PRODUCING BAR GRAPHS, ETC.
      XPPLOT DOES NO OVER-PRINTING, SO IT MAY BE USED WITH HALF-DUPLEX
      TERMINALS.

      DESCRIPTION OF ARGUMENTS
         ISIZE  - SIZE OF ACTUAL PLOT AREA, NOT INCLUDING LABELS, ETC.
                  IF =2, 4, 6, 8, 10, OR 12, THE PLOT SIZE WILL BE
                     THAT MANY INCHES, HORIZONTALLY AND VERTICALLY.
                     OR, ISIZE MAY BE OF THE FORM 100*IVERT+IHORZ, WHERE
                     IVERT AND IHORZ ARE THE VERTICAL AND HORIZONTAL
                     DIMENSIONS, IN INCHES, OF THE ACTUAL PLOT AREA.
                     IN THIS CASE, IVERT AND IHORZ MUST EACH BE ONE OF
                     THE VALUES 2, 4, 6, 8, 10, OR 12.
                     WHEN USING A TERMINAL, IHORZ (OR ISIZE) SHOULD
                     BE NO LARGER THAN 6.  WHEN USING A LINE PRINTER
                     WITH AUTOMATIC PAGE EJECT AT PAGE BREAKS, IVERT
                     (OR ISIZE) SHOULD BE NO LARGER THAN 8.
                  IF ISIZE.LT.0, ITS FORM MUST BE -(1000*NROW+NCOL),
                     WHERE NROW IS THE NUMBER OF ROWS, OR PRINT LINES,
                     TO BE USED, AND NCOL IS THE NUMBER OF COLUMNS,
                     OR PRINT POSITIONS, PER LINE.  THERE ARE NO
                     RESTRICTIONS ON NROW AND NCOL EXCEPT THAT THEY
                     BE AT LEAST TWO, AND NCOL.LE.121.
                     FOR EXAMPLE, ISIZE= -49101 WOULD PRODUCE A 49 ROW
                     BY 101 COLUMN (8 BY 10 INCH) PLOT.
                     LABELING OF THE AXES MAY NOT APPEAR NICELY REGULAR
                     WHEN UNUSUAL VALUES ARE USED FOR NROW OR NCOL.
                  IF ISIZE=0, A 6 BY 6 INCH PLOT WILL BE DONE.
                  IF ISIZE=1, AN 8 BY 8 INCH PLOT WILL BE DONE.
         IPAGE  - IF .GT.0, A PAGE EJECT WILL BE DONE BEFORE THE PLOT
                     IS STARTED.
                  IF .LE.0, NO PAGE EJECT IS DONE.  THIS WOULD BE THE
                     APPROPRIATE CHOICE FOR TERMINAL OUTPUT, OR IF
                     YOU WISH TO PRINT A TITLE AT THE TOP OF THE PLOT.
         IBOX   - IF =1, A BOX WILL BE DRAWN (WITH ASTERISKS) AROUND
```

```
                THE ACTUAL PLOT AREA (NOT INCLUDING LABELS, ETC.)
                USE OF A BOX IS NOT RECOMMENDED WHEN USING A
                TERMINAL, AS IT MAY SLOW DOWN PLOTTING TOO MUCH.
             IF =2, THE BOX WILL BE DRAWN WITH PLUS SIGNS.
             IF =3, THE BOX WILL BE DRAWN WITH ZEROES.
             IF =4, THE BOX WILL BE DRAWN WITH THE LETTER X.
             IF .LE.0, NO BOX WILL BE DRAWN AROUND THE PLOT AREA.
ILABEL -     IF =0 OR ANY VALUE OTHER THAN 1, 2, 3, OR 6, BOTH
                AXES WILL BE LABELLED EVERY TWO INCHES.
             IF =1, 2, 3, OR 6, THE Y AXIS WILL BE LABELLED EVERY
                WHOLE, HALF, THIRD, OR SIXTH OF AN INCH
                (ASSUMING SIX LINES ARE PRINTED PER VERTICAL INCH),
                WHILE X AXIS LABELLING REMAINS AT TWO INCHES.
             IF .LT.0, LABELS WILL BE SUPPRESSED COMPLETELY.
                HOWEVER, IF IMARK.GE.0, THE MAGNITUDE OF ILABEL
                WILL STILL INDICATE THE TIC MARK SPACING.
IMARK  -     IF .GT.0, TIC MARKS WILL MARK EACH ROW OR COLUMN
                BEING LABELLED.  THE LENGTHS OF THE MARKS WILL
                BE ROUGHLY IMARK/6 INCHES.  THUS, IF IMARK IS SET
                TO A LARGE NUMBER, SAY 100, THESE TIC MARKS WILL
                BECOME FULL GRID LINES.  NOTE -- THE X AXIS MAY
                NOT BE LABELLED MORE DENSELY THAN ONCE EVERY
                TWO INCHES.  HOWEVER, THE SPACING OF TIC MARKS
                ON THE X AXIS WILL EQUAL THAT OF THE Y AXIS
                EXCEPT WHEN IABS(ILABEL)=3 OR 6, IN WHICH CASE
                X AXIS TIC MARKS WILL REMAIN AT HALF INCH
                INTERVALS.
             IF .LT.0, NO TIC MARKS WILL BE USED.  IN THIS CASE,
                X AXIS LABELLING WILL BE SUPPRESSED, AS IT WOULD
                BE AMBIGUOUS.  THIS MODE MAY BE DESIRABLE WHEN
                PLOTTING HISTOGRAMS WITH ITYPE=2.
             IF =0, SINGLE CHARACTER MARKS WILL BE USED.
IAXES  -     IF .GT.0, X- AND Y-AXES WILL BE DRAWN ON THE PLOT, SO
                AS TO PASS AS CLOSELY AS POSSIBLE THROUGH X=0, Y=0.
             IF .LE.0, NO AXES WILL BE DRAWN.
ITYPE  -     IF .LE.1, A NORMAL PLOT WILL BE DONE.
             IF =2, A HORIZONTAL BAR GRAPH WILL BE DRAWN.
                THIS MEANS THAT THE OCCURANCE OF A DATA POINT
                ON A CURVE CAUSES THE PLOT CHARACTER FOR
                THAT CURVE TO BE PRINTED IN THE ROW/COLUMN
                POSITION INDICATED BY THAT POINT, PLUS IN ALL
                COLUMNS OF THAT ROW TO THE LEFT OF THAT POINT.
                HOWEVER, A HIGHER PRIORITY CURVE MAY THEN
                OVERPLOT SOME OR ALL OF THAT ROW OF CHARACTERS.
                SEE NOTE AT BOTTOM.
             IF .GE.3, A VERTICAL BAR GRAPH WILL BE DRAWN.
                THAT IS, A GIVEN POINT CAUSES ITS PLOT CHARACTER
                TO BE PLOTTED IN A GIVEN ROW/COLUMN POSITION,
                PLUS IN ALL ROWS BELOW THAT POSITION.
                HOWEVER, IF ANOTHER DATA POINT APPEARS LATER
                (I.E., LOWER) IN THIS COLUMN, IT WILL TAKE
                PRECEDENCE FOR THAT ROW AND BELOW.
                NOTE -- YOU SHOULD NOT USE IMARK.GT.1 OR
                IAXES.GT.0 WHEN ITYPE=3, AS ANYTHING PRINTED ON THE
                ACTUAL PLOT AREA WILL PROPAGATE BELOW THAT
                POINT JUST AS IF IT WERE DATA.
RANGE  -     IF RANGE=0., THE PLOT SCALES WILL BE DETERMINED BY
                THE DATA.  IN THIS CASE, THE MIDDLE OF THE FIRST
                COLUMN WILL CORRESPOND TO THE MINIMUM X VALUE,
                THE MIDDLE OF THE LAST COLUMN WILL CORRESPOND TO
                THE MAXIMUM X VALUE, THE MIDDLE OF THE BOTTOM ROW
                WILL CORRESPOND TO THE MINIMUM Y VALUE, AND
                THE MIDDLE OF THE TOP ROW WILL CORRESPOND TO THE
                MAXIMUM Y VALUE.
             IF RANGE.NE.0., THEN RANGE IS ASSUMED TO BE AN ARRAY
                OF LENGTH 5, CONTAINING THE FOLLOWING.
                   RANGE(1) - A NON-ZERO VALUE.
                   RANGE(2) - X VALUE OF LEFT-MOST COLUMN
                   RANGE(3) - X VALUE OF RIGHT-MOST COLUMN
                   RANGE(4) - Y VALUE OF BOTTOM ROW
                   RANGE(5) - Y VALUE OF TOP ROW
                THESE RANGE PARAMETERS MAY BE USED EITHER TO
                EXTEND OR RESTRICT THE NORMAL RANGE.  THAT IS,
                THEY MAY BE USED TO ZOOM OUT OR IN.
```

```
      NUM     - NUMBER OF CURVES BEING USED FOR THIS PLOT.
                IF NUM=1, THE CALLING SEQUENCE MAY BE TERMINATED
                AFTER THE PARAMETER L1.  (MOST FORTRAN COMPILERS
                WILL TOLERATE THIS USAGE.)  IF NUM=2, THE CALLING
                SEQUENCE MAY BE TERMINATED AFTER PARAMETER L2, ETC.
                IF NUM=0, ONE CURVE WILL BE ASSUMED.
      X1      - ARRAY OF X-COORDINATES FOR FIRST CURVE.
      Y1      - ARRAY OF Y-COORDINATES FOR FIRST CURVE.
      N1      - NUMBER OF DATA POINTS ON FIRST CURVE (DIMENSION OF
                X1 AND Y1).
      L1      - PLOT CHARACTER TO BE USED FOR FIRST CURVE.  THIS
                CHARACTER SHOULD BE IN A1 FORMAT, SUCH AS 1H*.
      X2,Y2,N2,L2 - CORRESPONDING DATA FOR SECOND CURVE.
      X3,Y3,N3,L3 - CORRESPONDING DATA FOR THIRD CURVE.
      X4,Y4,N4,L4 - CORRESPONDING DATA FOR FOURTH CURVE.
                NOTE -- IN CASE DATA FROM MORE THAN ONE CURVE OCCURS
                   AT A GIVEN ROW/COLUMN POSITION, SYMBOL L1 WILL
                   OVERRIDE SYMBOL L2, WHICH WILL OVERRIDE SYMBOL L3,
                   ETC.

   FORMAT
      THE FORMAT USED FOR THE PLOT AREA IS
      FORMAT (1X,G11.4,A1,NA1,A1)
      WHERE NA1 IS THE ACTUAL PLOT AREA, AND THE TWO A1 FIELDS
      ARE FOR THE BOX AND TIC MARKS.

   SOME SIMPLE EXAMPLES --
      A TYPICAL TERMINAL PLOT OF A SINGLE FUNCTION CAN BE DONE BY
         CALL XPPLOT(0,0,0,0,0,0,0,0.,1,X,Y,N,1H*)
      A TYPICAL LINE PRINTER PLOT OF A SINGLE FUNCTION CAN BE DONE BY
         CALL XPPLOT(1,1,1,1,1,1,1,0.,1,X,Y,N,1H*)
      A TYPICAL LINE PRINTER PLOT OF TWO DEPENDENT VARIABLES
      AGAINST A SINGLE INDEPENDENT VARIABLE CAN BE DONE BY
         CALL XPPLOT(1,1,1,1,1,1,1,0.,2,X,Y1,N,1H1,X,Y2,N,1H2)

   R E JONES  DIVISION 2642  JUNE 1976
```

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

INDEX OF ROUTINES

DISTRIBUTION

LAWRENCE BERKELEY LABORATORY, BUILDING 90, ROOM 3111, KUNG RU-MEI
LAWRENCE LIVERMORE LABORATORY, L310 FRITSCH FREDERICK N
LAWRENCE LIVERMORE LABORATORY, L310, HINDMARSH ALAN C
LAWRENCE LIVERMORE LABORATORY, L310, SMIRIGA NORA G
LAWRENCE LIVERMORE LABORATORY, L402, FONG KIRBY W
LOS ALAMOS SCIENTIFIC LABORATORY, C6 KAHANER DAVID
8000 COOK T B
8100 GUTIERREZ L
8200 DESELM C H
8300 MURPHEY B F
8400 SCRIVNER W C
8010 BLACKWELL ARLYN N
8121 CALLABRESI MELVIN L
8121 DELAMETER WILLIAM R
8121 EASON ERNEST D
8121 JONES ALFRED T
8121 LEONARD CHARLES M JR
8121 PERRA MARK W
8121 VOELKER L EUGENE
8121 WEINGARTEN LAWRENCE I
8122 ELLIS RAYMOND W
8122 GRANT JOSEPH E
8122 HOYLE CHARLES S
8122 KAN YIH-RENN
8122 KEILMAN JOHN C
8122 ROGERS L ALICE
8122 SCOTT CARLTON A
8123 GRISBY SYLVESTER
8123 JACOBSON RICHARD S
8123 JORGENSON WILBUR E
8123 ZINKE WILLIAM D
8124 ABRAMS MARTIN
8124 CHENOWETH DONALD R
8124 CLARK GARY L
8124 GREEN ROBERT M
8124 GRIFFITH LEE V
8124 SIEBERS DENNIS L
8131 DELLIN THEODORE A
8131 HILLER CARL C
8131 WILSON WM G
8132 BROWN MELVIN E
8132 CULL EDWARD T
8132 MAVIS CLAYTON L
8132 MOORE C W
8132 RADOSEVICH LEE G
8141 HAVLIK DAVID J
8141 MCDONALD ALFREDO
8141 WIEMKEN ALBERT J
8142 BAUM SAMUEL
8142 BIRNBAUM MICHAEL R
8142 NG RAYMOND
8143 GILDEA PATRICK D
8143 PENDLEY MICHAEL H
8144 DAHLKE LUTZ W
8144 DOLAN KENNETH W
8144 OIEN CHARLES T
8144 TICHENOR DANIEL A
8144 WILLIS ALEC R
8156 DEAN DAVID K
8156 DIGHTON LEONARD E
8156 FURNBERG CARLTON M
8156 STARKEY DONALD B
8156 TALBOT EDWARD B
8156 WRIGHT JAMES B
8157 BOLEN JACK L
8157 LOLL MARVIN B
8158 BOLTON WILLARD R JR
8158 EVERETT ROGER N
8158 HILES LEONARD A
8158 HUMPHREY ROBERT E

8158 RIVENES ARNOLD S
8158 ROGERS MICHAEL H
8159 BRAY DAVID N
8159 CASHEN JERRY J
8159 KLEIN NORMAN G
8159 LEEPER JOHN E
8159 SKOOG CLIFFORD D
8159 SODERSTRAND MICHAEL A
8159 TANNER DUNCAN N
8159 THOMMEN JEFFERY
8159 WATKINS LAURENCE M
8161 MARTIN GERTRUDE A
8166 ANDERSON JOHN S
8167 DERICKSON LINN W
8167 HENDERSON GERALD W
8168 HICKS MEARLE G
8168 MAUCK CHARLES J
8181 CHRISMAN SANDRA E
8181 COOLEY NOEL H
8181 HAWLEY J DALE
8181 KELLOM ARTHUR W
8181 LUCERO L ANTONIO
8181 SWANSON DONALD H
8181 TOCKEY ROBERT J
8183 COFIELD CURTIS H
8183 HUNT CHARLES T
8185 AMARAL RONALD J
8185 POTTHOFF C M
8212 JOINER EMILY A
8265 CUPPS FRANK J
8313 BRADSHAW ROBERT W
8313 BRAMLETTE T TAZWELL
8313 CARLING ROBERT W
8313 KRAMER CAROLYN M
8313 NICHOLS MONTE C
8314 DAWSON DANIEL B
8314 ESTILL WESLEY B
8314 ROBINSON STEVEN L
8314 WEST ANTON J
8315 BARTEL JAMES J
8315 MILLS BERNICE E
8315 WEST LLOYD A
8320 GOLD THEODORE S
8321 BERKBIGLER KATHRYN P
8321 DELAQUILL PASCAL III
8321 GALLAGHER ROBERT J
8321 KEETON STEWART C
8321 STIMMELL KATHLEEN
8321 STRANDIN GERALD E
8321 WAGNER NORMAN R
8322 BASINGER RICHARD C
8322 LEARY PATRICIA L
8322 MILLER GORDON J
8322 MITCHELL DONNA L
8322 NEIGHBORS PAULA K
8323 BARNHOUSE JOHN N JR
8323 HEIDELBERG S T
8323 ISLER RICHARD E
8323 KOOPMANN BRUCE E
8323 QUOCK HANLOY
8323 SCHUKNECHT ARNOLD G
8323 WHITWORTH FREDDY L
8324 BREAZEAL NORMAN L
8324 BROWN GARRY S
8324 COLL CORNELIUS F III
8324 GRIESEL M ANN
8324 KERSTEIN ALAN R
8324 ROGERS JAMES N
8324 STRUVE JAMES E
8324 WOOLERY EDGAR F
8325 BISSON CHARLES L
8325 GABRIELSON VERLAN K
8325 HUDDLESTON ROBERT E
8325 JEFFERSON THOMAS H JR

8325 MANSFIELD JUANITA
8325 MANTEUFFEL THOMAS A
8325 MARGOLIS STEPHEN B
8326 EICKER PATRICK J
8326 HANKINS JOE D
8326 IANNUCCI JOSEPH J
8326 WOODARD JAMES B JR
8327 BARKER BERTON E
8327 JONES HILARY D
8327 LATHROP JAMES F
8327 LEE ROY Y
8340 WIRTH JOHN L
8341 BASKES MICHAEL I
8341 HAGGMARK LEROY G
8341 HICKS ANNETTE M
8341 LOOK GEORGE W
8341 MELIUS CARL F
8341 WILSON WILLIAM D
8342 GAY RICHARD C
8342 GRAY STEPHEN C
8342 HARTWIG CHARLES M
8342 MATTERN PETER L
8342 RAHN LARRY A
8342 SCHMIEDER ROBERT W
8342 VASEY STEPHEN J
8342 VITKO JOHN JR
8345 MARTINELL RONALD E
8346 CHILDERS CARL W
8346 DECARLI CHARLES J
8347 BROWN LESLIE A
8347 CARON JAMES E
8347 MALINOWSKI MICHAEL E
8347 SWANSIGER WILLIAM A
8347 THOMAS GEORGE J
8347 VER BERKMOES ALFRED A
8351 COLEMAN HUGH W
8351 MITCHELL REGINALD E
8352 ROBINSON CLARENCE W
8352 WITZE PETER O
8353 CATTOLICA ROBERT J
8353 HARDESTY DONALD R
8354 ASHURST WILLIAM T
8354 KEE ROBERT J JR
8354 SANDERS BILLY R
8365 BARR VERNON C
8365 BARSIS EDWIN H
8365 BECKER JOHN A
8365 HALL R MICHAEL
8365 MCALLISTER DAN R
8365 PEGLOW STEVEN G
8365 WITEK HENRY M
8366 CLARK RALPH E
8366 DIDLAKE JOHN E JR
8366 HENSON DOUGLAS R
8366 MITCHELL KENNETH A
8366 NORRIS HAROLD F JR
8366 SCHAFER CLIFFORD T
8410 BARODDY ROGER A
8411 BENTON JOE D
8411 CONVERSE LOUISE S
8411 HOLBROOK ELMOND D
8411 MACMILLAN DOUGLAS C
8411 MANROW BRITT MARIE
8411 POLIAK RALPH S
8411 SWAN HERBERT W
8412 ANGVICK GENE L
8412 WHITNEY WILLIAM L
8413 DANNENBERG DAVID J
8413 GREEN DURWOOD P
8413 NELSON DENNIS B
8413 SKINROOD DONALD A
8423 AFFELDT BRUCE E
8423 JACKSON ROBERT W
8423 MADSEN JOAN F

8423 REIS HERMANN L
8423 SCHMEDDING WILLIAM
8423 VON STEEG HERMAN J
8423 WACKERLY CARL A
8432 BYFIELD VERNON E
8432 COOK RICHARD S JR
8432 FINN RICHARD W
8432 KEIFER PATRICK
8434 EICHERT FRED F JR
3141 TECHNICAL LIBRARY SYSTEMS AND PROCESSES DIVISION
8265 CUPPS F J / TECHNICAL LIBRARY SYSTEMS AND PROCESSES DIVISION
8265 TECHNICAL PUBLICATIONS AND ART DIVISION, FOR TIC (2)
8266 LIBRARY AND SECURITY CLASSIFICATION DIVISION (5)
8325 JEFFERSON T H / SANDIA LABORATORIES (LIVERMORE)
MATHEMATICAL SUBROUTINE LIBRARY (30)