MCR-76-217
Contract NAS8-31376

Volume IIB

Listings, Sparse FORMA Subroutines

May 1976

Expansion and Improvement of the FORMA System for Response and Load Analysis

MARTIN MARIETTA

MCR-76-217

Contract NAS8-31376

EXPANSION AND IMPROVEMENT OF THE FORMA SYSTEM FOR RESPONSE AND LOAD ANALYSIS

Volume IIB - Listings, Sparse FORMA Subroutines

May 1976

Author:

Richard L. Wohlen

Approved:

Richard L. Wohlen
Program Manager

Prepared for: National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Huntsville, Alabama 35812

MARTIN MARIETTA CORPORATION
Denver Division
Denver, Colorado 80201

FOREWORD

This report presents results of the expansion and improvement of the FORMA system for response and load analysis. The acronym FORMA stands for FORTRAN Matrix Analysis. The study, performed from 16 May 1975 through 17 May 1976 was conducted by the Analytical Mechanics Department, Martin Marietta Corporation, Denver Division, under the contract NAS8-31376. The program was administered by the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama under the direction of Dr. John R. Admire, Structural Dynamics Division, Systems Dynamics Laboratory.

This report is published in seven volumes:

Volume I - Programming Manual,
Volume IIA - Listings, Dense FORMA Subroutines,
Volume IIB - Listings, Sparse FORMA Subroutines,
Volume IIC - Listings, Finite Element FORMA Subroutines,
Volume IIIA - Explanations, Dense FORMA Subroutines,
Volume IIIB - Explanations, Sparse FORMA Subroutines, and
Volume IIIC - Explanations, Finite Element FORMA Subroutines.

CONTENTS

ABSTRACT

This report presents techniques for the solution of structural dynamic systems on an electronic digital computer using FORMA (FORTRAN Matrix Analysis).

FORMA is a library of subroutines coded in FORTRAN IV for the efficient solution of structural dynamics problems. These subroutines are in the form of building blocks that can be put together to solve a large variety of structural dynamics problems. The obvious advantage of the building block approach is that programming and checkout time are limited to that required for putting the blocks together in the proper order.

The FORMA method has advantageous features such as:

1. subroutines in the library have been used extensively for many years and as a result are well checked out and debugged;

2. method will work on any computer with a FORTRAN IV compiler;

3. incorporation of new subroutines is no problem;

4. basic FORTRAN statements may be used to give extreme flexibility in writing a program.

Two programming techniques are used in FORMA: dense and sparse.

ACKNOWLEDGMENTS

The editor expresses his appreciation to those individuals whose assistance was necessary for the successful completion of this report. Dr. John R. Admire was instrumental in the definition of the program scope and contributed many valuable suggestions. Messrs. Carl Bodley, Wilcomb Benfield, Darrell Devers, Richard Hruda, Roger Philippus, and Herbert Wilkening, all of the Analytical Mechanics Department, Denver Division of Martin Marietta Corporation, have contributed ideas, as well as subroutines, in the formulation of the FORMA library.

The editor also expresses his appreciation to those persons who developed FORTRAN, particularly the subroutine concept of that programming tool.

I. INTRODUCTION

A listing of the source deck of each sparse FORMA subroutine is given in this volume to remove the "black-box" aura of the subroutines so that the analyst may better understand the detailed operations of each subroutine.

The format of a sparse matrix on a utility tape is given in Chapter II.

The FORTRAN IV programming language is used in all sparse FORMA subroutines.

II. FORMAT OF SPARSE (Y....) MATRIX ON UTILITY TAPE (DISK)

end of record ↓

Matrix Header	Partition 1 Header	Partition 1 Numbers (2 records)	Partition 2 Header	Partition 2 Numbers (2 records)	etc

MATRIX HEADER:

NROW	Number of rows in matrix
NCOL	Number of columns in matrix
NPART	Number of partitions of matrix on tape (disk)
NNZA	Number of non-zeroes in matrix
IFORD	Indicator for ordered matrix
KV	Dimension size of work vector when matrix was formed
ISHAPE	Shape indicator (DIAG, LOWER, UPPER, WHOLE)
0	extra
0	
0	

PARTITION 1 HEADER:

NNZP	Number of nonzeroes in partition
LFELP	Location (Row, Column) of first element in partition
LLELP	Location (Row, Column) of last element in partition
0	extra
0	
0	
0	
0	
0	
0	

PARTITION 1 NUMBERS:

(LV(I), I = 1, NNZP)
↑
Location (Row, Column)

(V(I), I = 1, NNZP)
↑
Element Value

III. SUBROUTINE LISTINGS

The subroutines are given in alphabetical order with numbers coming before letters.

YAA

```
      SUBROUTINE YAA (ALPHA,NUTA,NUTZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1)
      DATA NIT,NOT/5,6/
C
C  SCALAR ALPHA TIMES SPARSE MATRIX A. (ALPHA * A = Z).
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,YLORD ,YNOZER,YOUT   ,YOUTI ,
C                          YPART ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1970.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  ALPHA = SCALAR THAT MULTIPLIES MATRIX A.
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C
C
      REWIND NUTA
      REWIND NUTZ
      CALL YINI   (NUTA,LV,1,10)
      CALL YOUTI  (NUTZ,LV,1,10)
      NPART=LV(3)
C
      DO 10 J=1,NPART
      CALL YINI   (NUTA,LV,1,10)
      CALL YOUTI  (NUTZ,LV,1,10)
      NNZ=LV(1)
                                                   NERROR=1
      IF (NNZ.GT.KV) GO TO 999
      IF (NNZ.GT.0) GO TO 3
      CALL YOUTI  (NUTZ,LV,1,1)
      CALL YOUT   (NUTZ, V,1,1)
      GO TO 10
    3 CALL YINI   (NUTA,LV,1,NNZ)
      CALL YIN    (NUTA,V,1,NNZ)
C
      DO 5 I=1,NNZ
    5 V(I)=ALPHA*V(I)
C
      CALL YOUTI  (NUTZ,LV,1,NNZ)
      CALL YOUT   (NUTZ,V,1,NNZ)
   10 CONTINUE
C
      CALL YNOZER (NUTZ,V,LV,KV,NUT1)
      CALL YLORD  (NUTZ,V,LV,KV,NUT1,NUT2)
      RETURN

  999 CALL ZZBOMB (3HYAA   ,NERROR)
      END
```

```
      SUBROUTINE YAABB (ALPHA,NUTA,BETA,NUTB,NUTZ,V,LV,KV,NUT1,NUT2)
      SUBROUTINE YAABB (ALPHA,NUTA,BETA,NUTB,NUTZ,V,LV,KV,NUT1,NUT2)
      SUBROUTINE YAABB (ALPHA,NUTA,BETA,NUTB,NUTZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  MATRIX SUMMATION FOR SPARSE MATRICES. ALPHA * A + BETA * B = Z.
C  CALLS FORMA SUBROUTINES XLORD ,YIN   ,YINI  ,YLORD ,YNOZER,YOUT
C                          YOUTI ,YPART ,YSYMLH,YSYMUH,YZERO.
C  DEVELOPED BY R A PHILIPPUS.  MAY 1969.
C  LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  ALPHA = SCALAR THAT MULTIPLIES MATRIX A.
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  BETA  = SCALAR THAT MULTIPLIES MATRIX B.
C  NUTB  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX B IS STORED.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C  GET (A) HEADER INFORMATION.
      REWIND NUTA
      REWIND NUTB
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NNZA = MHEAD(4)
      IASHAP = MHEAD(7)
C  GET (B) HEADER INFORMATION.
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
      NNZB = MHEAD(4)
      IBSHAP = MHEAD(7)
C
C  ALLOW FOR DIFFERENT SIZE (A) AND (B).
      NRZ=NRA
      NCZ=NCA
      IF (NRB .GT. NRA) NRZ=NRB
      IF (NCB .GT. NCA) NCZ=NCB
      IF (NNZA .GT. 0 .AND. NNZB .GT. 0) GO TO 4
      IF (NNZA .EQ. 0 .AND. NNZB .GT. 0) GO TO 15
      IF (NNZA .GT. 0 .AND. NNZB .EQ. 0) GO TO 15
      CALL YZERO  (NUTZ,NRZ,NCZ)
      RETURN
C  MAKE (A) AND (B) SAME SHAPE.
    4 IF (IASHAP.NE.IBSHAP) GO TO 5
      IZSHAP=IASHAP
      GO TO 15
    5 IZSHAP=5HWHOLE
      IF (IASHAP.EQ.5HWHOLE .OR. IASHAP.EQ.4HDIAG) GO TO 10
```

```
      IF (IASHAP.EQ.5HLOWER) CALL YSYMUH (NUTA,V,LV,KV,NUT1,NUT2)
      IF (IASHAP.EQ.5HUPPER) CALL YSYMLH (NUTA,V,LV,KV,NUT1,NUT2)
   10 IF (IBSHAP.EQ.5HWHOLE .OR. IBSHAP.EQ.4HDIAG) GO TO 15
      IF (IBSHAP.EQ.5HLOWER) CALL YSYMUH (NUTB,V,LV,KV,NUT1,NUT2)
      IF (IBSHAP.EQ.5HUPPER) CALL YSYMLH (NUTB,V,LV,KV,NUT1,NUT2)
C  MAKE CERTAIN ELEMENTS ARE ORDERED.
   15 CALL YLORD  (NUTA,V,LV,KV,NUT1,NUT2)
      CALL YLORD  (NUTB,V,LV,KV,NUT1,NUT2)
C
      LZS=1
      I=0
      IA=0
      IB=0
      NREC=0
      REWIND NUTA
      REWIND NUTB
      REWIND NUTZ
C
      CALL YINI   (NUTA,MHEAD,1,10)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      IASHAP = MHEAD(7)
      CALL YINI   (NUTB,MHEAD,1,10)
      NPARTB = MHEAD(3)
      NNZB = MHEAD(4)
      IBSHAP = MHEAD(7)
      MHEAD(1) = NRZ
      MHEAD(2) = NCZ
      IF (NNZA.GT.0 .AND. NNZB.GT.0) GO TO 35
C
      IF (NNZA.GT.0) GO TO 25
      MHEAD(3) = NPARTB
      MHEAD(4) = NNZB
      MHEAD(7) = IBSHAP
      CALL YOUTI  (NUTZ,MHEAD,1,10)
C
      DO 20 J=1,NPARTB
      CALL YINI   (NUTB,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      NMULT = MHEAD(1)
      CALL YINI   (NUTB,LV,1,MHEAD(1))
      CALL YIN    (NUTB,V,1,MHEAD(1))
      DO 18 IMULT=1,NMULT
   18 V(IMULT) = BETA*V(IMULT)
      CALL YOUTI  (NUTZ,LV,1,MHEAD(1))
   20 CALL YOUT   (NUTZ, V,1,MHEAD(1))
      RETURN
C
   25 MHEAD(3) = NPARTA
      MHEAD(4) = NNZA
      MHEAD(7) = IASHAP
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      DO 30 J=1,NPARTA
      CALL YINI   (NUTA,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
```

```
      NMULT = MHEAD(1)
      CALL YINI    (NUTA,LV,1,MHEAD(1))
      CALL YIN     (NUTA,V,1,MHEAD(1))
      DO 28 IMULT=1,NMULT
   28 V(IMULT) = ALPHA*V(IMULT)
      CALL YOUTI   (NUTZ,LV,1,MHEAD(1))
   30 CALL YOUT    (NUTZ,V,1,MHEAD(1))
      RETURN
C
   35 NREAD=NPARTA+NPARTB
      NZMAX=NNZA+NNZB
      MHEAD(3) = NREAD
      MHEAD(4) = NZMAX
      MHEAD(6) = 0
      MHEAD(7) = IZSHAP
      CALL YOUTI   (NUTZ,MHEAD,1,10)
      CALL YINI    (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
      CALL YINI    (NUTB,MHEAD,1,10)
      NNZPB = MHEAD(1)
      LFELPB = MHEAD(2)
      LLELPB = MHEAD(3)
C
C  READ A PARTITION OF A AND MULTIPLY IT BY ALPHA
   40 I=I+1
      IA=IA+1
      LZE=LZS-1+NNZPA
      CALL YINI    (NUTA,LV,LZS,LZE)
      CALL YIN     (NUTA,V,LZS,LZE)
      IF (IA.GE.NPARTA) GO TO 42
      CALL YINI    (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
   42 IF (ALPHA.EQ.1.) GO TO 50
      DO 45 J=LZS,LZE
   45 V(J)=ALPHA*V(J)
   50 IF (I.GT.1) GO TO 65
      LZS=LZE+1
C
C  READ A PARTITION OF B AND MULTIPLY IT BY BETA
   55 I=I+1
      IB=IB+1
      LZE=LZS-1+NNZPB
      CALL YINI    (NUTB,LV,LZS,LZE)
      CALL YIN     (NUTB,V,LZS,LZE)
      IF (IB.GE.NPARTB) GO TO 57
      CALL YINI    (NUTB,MHEAD,1,10)
      NNZPB = MHEAD(1)
      LFELPB = MHEAD(2)
      LLELPB = MHEAD(3)
   57 IF (BETA.EQ.1.) GO TO 65
      DO 60 J=LZS,LZE
```

```
   60 V(J)=BETA*V(J)
   65 LAE=LZS-1
      IZ=LZS
C
      DO  85 IP=1,LAE
   70 IF (LV(IP)-LV(IZ)) 85,80,75
   75 IZ=IZ+1
      IF (IZ.GT.LZE) GO TO 90
      GO TO 70
   80 V(IP)=V(IP)+V(IZ)
      V(IZ)=0.
      IZ=IZ+1
      IF (IZ.GT.LZE) GO TO 90
   85 CONTINUE
C
   90 NNZW=0
C
      DO 95  IZ=1,LZE
      IF (V(IZ).EQ.0.) GO TO 95
      NNZW=NNZW+1
      V(NNZW)=V(IZ)
      LV(NNZW)=LV(IZ)
   95 CONTINUE
C
      CALL XLORD (V,LV,1,NNZW)
      LZE=NNZW

   WRITE A PARTITION OF Z
  100 MAX=KV/4
      IF (LZE.LT.KV/4*3 .AND. I.LT.NREAD) GO TO 130
      IF (MAX.GT.LZE) MAX=LZE
  105 IF (LV(MAX).LT.LFELPB .OR. IB.EQ.NPARTB) GO TO 110
      MAX=MAX-1
      GO TO 105
  110 IF (LV(MAX).LT.LFELPA .OR. IA.EQ.NPARTA) GO TO 115
      MAX=MAX-1
      GO TO 110
  115 IF (MAX.EQ.0) GO TO 120
      MHEAD(1) = MAX
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(MAX)
      MHEAD(10) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,1,MAX)
      CALL YOUT   (NUTZ,V,1,MAX)
      NREC=NREC+1
C
C  DETERMINE WHETHER AND WHAT TO READ OR WRITE
  120 IF (MAX.EQ.LZE .AND. I.EQ.NREAD) GO TO 135
      K=MAX
      MOVE=LZE-MAX
      DO 125 J=1,MOVE
      K=K+1
      V(J)=V(K)
  125 LV(J)=LV(K)
```

```
      LZF=MOVE
      IF (T.EQ.NREAD) GO TO 160
  130 LZS=LZE+1
      MIN=LFELPA
      IF (MIN.GT.LFELPB .AND. IB.LT.NPARTB) MIN=LFELPB
      IF (MIN.EQ.LFELPA .AND. LZE+NNZPA.GT.KV) GO TO 100
      IF (MIN.EQ.LFELPB .AND. LZE+NNZPB.GT.KV) GO TO 100
      IF (MIN.EQ.LFELPA .AND. IA.LT.NPARTA) GO TO 40
      GO TO 55
C
  135 IF (NREC.EQ.NREAD) GO TO 140
      DO 138 J=1,10
  138 MHEAD(J) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,2)
      CALL YOUTI  (NUTZ,MHEAD,1,2)
      NREC=NREC+1
      GO TO 135
  140 CALL YNOZER (NUTZ,V,LV,KV,NUT1)
      RETURN
      END
```

```
      SUBROUTINE YASSEM (NUTA,IRZ,JCZ,NUTZ,V,LV,KV,NUT1,NUT2,NUT3)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  SPARSE MATRIX ASSEMBLY. (MATRIX A INTO MATRIX Z).
C  BE SURE MATRIX Z IS DEFINED BEFORE CALLING THIS SUBROUTINE. FOR
C  EXAMPLE, CALL YZERO TO CLEAR MATRIX Z.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                              YSYMLH,YSYMUH,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1970.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  IRZ   = ROW NUMBER IN MATRIX Z OF FIRST ROW OF MATRIX A.
C  JCZ   = COLUMN NUMBER IN MATRIX Z OF FIRST COLUMN OF MATRIX A.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT3  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = MATRIX A EXCEEDS MATRIX Z.
C  2 = SIZE LIMITATION EXCEEDED.
C  3 = SIZE LIMITATION EXCEEDED.
C
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,IC)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      IASHAP = MHEAD(7)
      ILIMIT=NRA+IRZ-1
      JLIMIT=NCA+JCZ-1
      REWIND NUTZ
      CALL YINI   (NUTZ,MHEAD,1,IO)
      NRZ = MHEAD(1)
      NCZ = MHEAD(2)
      IZSHAP = MHEAD(7)
      IZSAVE = MHEAD(7)
                                                             NERROR=1
      IF (ILIMIT.GT.NRZ .OR. JLIMIT.GT.NCZ) GO TO 999
      IF (IASHAP.EQ.5HLOWER) CALL YSYMUH (NUTA,V,LV,KV,NUT1,NUT2)
      IF (IZSHAP.EQ.5HLOWER) CALL YSYMUH (NUTZ,V,LV,KV,NUT1,NUT2)
      IF (IASHAP.EQ.5HUPPER) CALL YSYMLH (NUTA,V,LV,KV,NUT1,NUT2)
      IF (IZSHAP.EQ.5HUPPER) CALL YSYMLH (NUTZ,V,LV,KV,NUT1,NUT2)
      IZSHAP=5HWHOLE
      IF (IZSAVE.EQ.4HDIAG .AND. IASHAP.EQ.4HDIAG .AND. IRZ.EQ.JCZ)
     *    IZSHAP=4HDIAG
      REWIND NUTA
      REWIND NUT1
      CALL YINI   (NUTA,MHEAD,1,IC)
      IASHAP = MHEAD(7)
```

```
      MHEAD(1) = NRZ
      MHEAD(2) = NCZ
      MHEAD(7) = IZSHAP
      CALL YOUTI  (NUT1,MHEAD,1,10)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      IF (NNZA.GT.0) GO TO 2
      DO 1 I=1,10
      V(I) = 0.
    1 MHEAD(I) = 0
      CALL YOUTI  (NUT1,MHEAD,1,10)
      CALL YOUTI  (NUT1,MHEAD,1,2)
      CALL YOUT   (NUT1,    V,1,2)
      NPARTA=0
      GO TO 12
    2 LVADD=100000*(IRZ-1)+JCZ-1
                                                           NERROR=2
C
      DO 10 I=1,NPARTA
      CALL YINI   (NUTA,LV,1,10)
      IF (LV(1).GT.KV) GO TO 999
      LV(2)=LV(2)+LVADD
      LV(3)=LV(3)+LVADD
      CALL YOUTI  (NUT1,LV,1,10)
      NNZ=LV(1)
      CALL YINI   (NUTA,LV,1,NNZ)
      CALL YIN    (NUTA,V,1,NNZ)
      DO 5 J=1,NNZ
    5 LV(J)=LV(J)+LVADD
      CALL YOUTI  (NUT1,LV,1,NNZ)
   10 CALL YOUT   (NUT1,V,1,NNZ)
C
   12 REWIND NUTZ
      REWIND NUT2
      CALL YINI   (NUTZ,MHEAD,1,10)
      NPARTZ = MHEAD(3)
      NNZZ = MHEAD(4)
      MHEAD(7) = IZSHAP
      CALL YOUTI  (NUT2,MHEAD,1,10)
      IF (NNZZ.LE.0) GO TO 30
C
                                                           NERROR=3
      DO 25 I=1,NPARTZ
      CALL YINI   (NUTZ,LV,1,10)
      IF (LV(1).GT.KV) GO TO 999
      CALL YOUTI  (NUT2,LV,1,10)
      NNZ=LV(1)
      CALL YINI   (NUTZ,LV,1,NNZ)
      CALL YIN    (NUTZ,V,1,NNZ)
C
      DO 20 J=1,NNZ
      IZ=LV(J)/100000
      IF (IZ.LT.IRZ) GO TO 20
      IF (IZ.GT.ILIMIT) GO TO 20
      JZ=LV(J)-100000*IZ
```

```
      IF (JZ.LT.JCZ) GO TO 20
      IF (JZ.GT.JLIMIT) GO TO 20
      V(J)=0.
   20 CONTINUE
C
      CALL YOUTI  (NUT2,LV,1,NNZ)
   25 CALL YOUT   (NUT2,V,1,NNZ)
C
      CALL YNOZER (NUT2,V,LV,KV,NUT3)
      GO TO 35
   30 REWIND NUTZ
      MHEAD(1) = NRZ
      MHEAD(2) = NCZ
      MHEAD(3) = NPARTA
      MHEAD(4) = NNZA
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = IZSHAP
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      GO TO 45
   35 REWIND NUT2
      CALL YINI   (NUT2,MHEAD,1,10)
      NPARTZ = MHEAD(3)
      NNZZ = MHEAD(4)
      NPART=NPARTA+NPARTZ
      NNZ=NNZA+NNZZ
      REWIND NUTZ
      MHEAD(3) = NPART
      MHEAD(4) = NNZ
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = IZSHAP
      CALL YOUTI  (NUTZ,MHEAD,1,10)
C
      DO 40 I=1,NPARTZ
      CALL YINI   (NUT2,LV,1,10)
      CALL YOUTI  (NUTZ,LV,1,10)
      NNZ=LV(1)
      CALL YINI   (NUT2,LV,1,NNZ)
      CALL YIN    (NUT2,V,1,NNZ)
      CALL YOUTI  (NUTZ,LV,1,NNZ)
   40 CALL YOUT   (NUTZ,V,1,NNZ)
C
   45 REWIND NUT1
      CALL YINI   (NUT1,MHEAD,10,10)
      MHEAD(10) = 0
C
      DO 50 I=1,NPARTA
      CALL YINI   (NUT1,LV,1,10)
      CALL YOUTI  (NUTZ,LV,1,10)
      NNZ=LV(1)
      CALL YINI   (NUT1,LV,1,NNZ)
      CALL YIN    (NUT1,V,1,NNZ)
      CALL YOUTI  (NUTZ,LV,1,NNZ)
   50 CALL YOUT   (NUTZ,V,1,NNZ)
```

```
      CALL YLORD  (NUTZ,V,LV,KV,NUT1,NUT2)
      RETURN
C
  999 CALL ZZBOMB (6HYASSEM,NERROR)
      END
```

```
      SUBROUTINE YBSL3A (NUTU,NUTD,NUTB,NUTZ,V,LUV,KV,NUT1,NUT2)
      DIMENSION V(1), LUV(1), IH(10)
C
C   SPARSE BACK SOLUTION OF (U**T)*(-D-)*(U)*(Z) = (P).
C   (U) IS A BANDED UPPER TRIANGULAR MATRIX WITH ONES ON THE DIAGONAL.
C   (-D-) IS A DIAGONAL MATRIX.  (B), AND THUS (Z) IS A MATRIX OF FULL
C   COLUMNS. ASSUMING (U)*(Z)=(Y) AND (-D-)*(Y)=(G) GIVES (U**T)*(G)=(B)
C   WHICH ARE EASILY SOLVED FOR (G), (Y), AND (Z).
C   (B),(G),(Y),(Z) IN 1ST THIRD OF V.  (U) IN 2ND THIRD OF V.
C   (-D-) IN 3RD THIRD OF V.
C   CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                              YTRANS.
C   DEVELOPED BY R L WOHLEN AND R A PHILIPPUS.  MARCH 1972.
C   LAST REVISION BY R A PHILIPPUS.  MARCH 1975.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C   NUTU  = LOGICAL NUMBER OF UTILITY TAPE WITH MATRIX U.
C   NUTD  = LOGICAL NUMBER OF UTILITY TAPE WITH MATRIX D.
C   NUTB  = LOGICAL NUMBER OF UTILITY TAPE WITH MATRIX B.
C   NUTZ  = LOGICAL NUMBER OF UTILITY TAPE WITH CALCULATED MATRIX Z.
C   V     = VECTOR WORK SPACE.
C   LUV   = VECTOR WORK SPACE.
C   KV    = DIMENSION SIZE OF V,LUV IN CALLING PROGRAM.
C   NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C   NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C   CONVERT B FROM SPARSE NOTATION TO FULL COLUMN NOTATION.
      CALL YTRANS (NUTB,NUT1,V,LUV,KV,NUT2,NUTZ)
      REWIND NUT1
      REWIND NUT2
      CALL YINI   (NUT1,IH,1,10)
      NCB = IH(1)
      NRB = IH(2)
      NPART = IH(3)
      NNZB = IH(4)
      KVMN = KV-NRB
      KVMNO2 = KVMN/2
      NCG = KVMNO2/NRB
      NGB = (NCB-1)/NCG+1
      DO 1 I=4,7
    1 IH(I) = 0
      IH(1) = NRB
      IH(2) = NCB
      IH(3) = NGB
      CALL YOUTI  (NUT2,IH,1,10)
      NNZPB = 1
      IF (NCB.LT.NCG) NCG=NCB
      LBS = KV/4+1
      LBSM1 = LBS-1
      LBE = LBSM1+NCG*NRB
      DO 2 I=LBS,LBE
    2 V(I) = 0.
      JF = 1
      JL = NCG
      DO 8 I=1,NPART
```

```
      CALL YINI   (NUT1,IH,1,10)
      NNZPB = IH(1)
      CALL YINI   (NUT1,LUV,1,NNZPB)
      CALL YIN    (NUT1,  V,1,NNZPB)
      DO 8 LB=1,NNZPB
      JB = LUV(LB)/100000
      IB = LUV(LB)-100000*JB
      IF (JB.LE.JL) GO TO 4
      IH(1) = NCG
      IH(2) = 0
      IH(3) = 0
      CALL YOUTI  (NUT2,IH,1,10)
      CALL YOUT   (NUT2, V,LBS    ,LBE)
      DO 3 J=LBS,LBE
    3 V(J) = 0.
      JF = JL+1
      JL = JF+NCG-1
      IF (JL.GT.NCB) JL=NCB
      NCG = JL-JF+1
      LBE = LBSM1+NCG*NRB
    4 L = (JB-JF)*NRB+IB
      LBSM1L = LBSM1+L
      V(LBSM1L) = V(LB)
      IF (I.LT.NPART .OR. LB.LT.NNZPB) GO TO 8
      IH(1) = NCG
      IH(2) = 0
      IH(3) = 0
      CALL YOUTI  (NUT2,IH,1,10)
      CALL YOUT   (NUT2, V,LBS    ,LBE)
    8 CONTINUE
C
C  V(1 THRU (KV-N)/2) CONTAINS B,G,Y,Z COLUMNS OF A GROUP.
C  V((KV-N)/2+1 THRU KV-N) CONTAINS COLUMNS OF U (FROM DIAGONAL UP TO
C  TOP NON-ZERO) OF A GROUP.
C  V(KV-N+1 THRU KV) CONTAINS D.
C  LUV(I),I=1,N IS NUMBER OF ELEMENTS IN COLUMN I.
C
      REWIND NUTU
      CALL YINI   (NUTU,IH,1,10)
      N   = IH(1)
      NGU = IH(3)
      LSU = (KV-N)/2 + 1
      LSD = KV-N+1
      CALL YINI   (NUTU,LUV,1,N)
      REWIND NUTD
      CALL YIN    (NUTD,V,LSD,LSD+N-1)
      REWIND NUT2
      CALL YINI   (NUT2,IH,1,10)
      NGB = IH(3)
      REWIND NUT1
      CALL YOUTI  (NUT1,IH,1,10)

      DO 89 IGB=1,NGB
      CALL YINI   (NUT2,IH,1,10)
      NCIGB = IH(1)
```

```
      NELIGB = N*NCIGB
      CALL YIN    (NUT2,V,1,NELIGB)
C
C   SOLUTION FOR (G) FROM (U**T)*(G)=(B).
      DO 37 IGU=1,NGU
      CALL YINI   (NUTU,IH,1,10)
      JSU = IH(1)
      JEU = IH(2)
      NELIGU = IH(3)
      CALL YIN    (NUTU,V,LSU,LSU+NELIGU-1)
      DO 37 JB=1,NCIGB
      LBSM1 = (JB-1)*N
      LJJU = LSU-1
      DO 36 JU=JSU,JEU
      LITJU = LJJU+1
      LJJU = LITJU+LUV(JU)-1
      ITJU = JU-LUV(JU)+1
      IF (ITJU .EQ. JU) GO TO 36
      LJJUM1 = LJJU-1
      LGB = LBSM1+JU
      LG = LBSM1+ITJU-1
      DO 34 LU=LITJU,LJJUM1
      LG = LG+1
   34 V(LGB) = V(LGB) - V(LU)*V(LG)
   36 CONTINUE
   37 CONTINUE

    SOLUTION FOR (Y) FROM (-D-)*(Y)=(G).
      LYE = 0
      DO 45 JY=1,NCIGB
      LYS = LYE+1
      LYE = LYS + N - 1
      LD = LSD-1
      DO 45 LY=LYS,LYE
      LD = LD+1
   45 V(LY) = V(LY)/V(LD)
C
C   SOLUTION FOR (Z) FROM (U)*(Z)=(Y).
C   U GROUPS ARE OBTAINED IN BACKWARDS ORDER.
      DO 57 IGUX=1,NGU
      IF (IGUX .EQ. 1) GO TO 55
      BACKSPACE NUTU
      BACKSPACE NUTU
      CALL YINI   (NUTU,IH,1.10)
      JSU = IH(1)
      JEU = IH(2)
      NELIGU = IH(3)
      CALL YIN    (NUTU,V,LSU,LSU+NELIGU-1)
   55 BACKSPACE NUTU
      BACKSPACE NUTU
      DO 57 JB=1,NCIGB
      LZSM1 = (JB-1)*N
      LITJU = LSU+NELIGU
      DO 56 JUX=JSU,JEU
      JU = JSU+JEU-JUX
```

```
      LJJU = LITJU-1
      LITJU = LJJU-LUV(JU)+1
      ITJU = JU-LUV(JU)+1
      IF (ITJU .EQ. JU) GO TO 56
      LJJUM1 = LJJU-1
      LZ = LZSM1+JU
      LZY = LZSM1+ITJU-1
      DO 54 LU=LITJU,LJJUM1
      LZY = LZY+1
   54 V(LZY) = V(LZY) - V(LU)*V(LZ)
   56 CONTINUE
   57 CONTINUE
C
      DO 72 I=1,10
   72 IH(I) = 0
      IH(1) = NCIGB
      CALL YOUTI  (NUT1,IH,1,10)
      CALL YOUT   (NUT1,V,1,NELIGB)
   89 CONTINUE
C
C  CONVERT Z FROM FULL COLUMN NOTATION TO SPARSE NOTATION.
      REWIND NUT1
      REWIND NUTZ
      CALL YINI   (NUT1,IH,1,10)
      IH(4) = NRB*NCB
      IH(5) = 0
      IH(6) = 0
      IH(7) = 5HWHOLE
      CALL YOUTI  (NUTZ,IH,1,10)
      JZ = 0
      DO 110 IGB=1,NGB
      CALL YINI   (NUT1,IH,1,10)
      NC = IH(1)
      NNZPB = NC*NRB
      CALL YIN    (NUT1,V,1,NNZPB)
      LB = 0
      DO 100 J =1,NC
      JZ = JZ+1
      DO 100 IZ=1,NRB
      LB = LB+1
  100 LUV(LB) = 100000*IZ+JZ
      IH(1) = NNZPB
      IH(2) = LUV(1)
      IH(3) = LUV(NNZPB)
      CALL YOUTI  (NUTZ,IH,1,10)
      CALL YOUTI  (NUTZ,LUV,1,NNZPB)
  110 CALL YOUT   (NUTZ,  V,1,NNZPB)
      CALL YLORD  (NUTZ,V,LUV,KV,NUT1,NUT2)
      RETURN
      END
```

```
      SUBROUTINE YBTAB (NUTA,NUTB,NUTZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  TRIPLE MATRIX PRODUCT FOR SPARSE MATRICES.
C  ( B(TRANSPOSE) * A * B = Z )
C  KV/4 MUST BE EQUAL TO OR GREATER THAN,
C      (1) NUMBER OF ROWS OF MATRIX B (NRB=NRA=NCA)
C   AND (2) NUMBER OF COLUMNS OF MATRIX B.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YMULT ,YNOZER,YOUT  ,
C                           YOUTI ,YPART ,YSYMLH,YSYMUH,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JUNE 1969.
C  LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTB  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX B IS STORED.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C  2 = INCOMPATIBLE MATRICES.

      CALL YPART  (NUTA,V,LV,KV,NUT1)
      CALL YPART  (NUTB,V,LV,KV,NUT1)
C
C  GET (A) HEADER INFORMATION.
      REWIND NUTA
      REWIND NUTB
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NRASAV = NRA
      NCA = MHEAD(2)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      IASHAP = MHEAD(7)
C  GET (B) HEADER INFORMATION.
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
      NPARTB = MHEAD(3)
      NNZB = MHEAD(4)
      IBSHAP = MHEAD(7)
      IADENS=100*NNZA/NRA/NCA
      IBDENS=100*NNZB/NRB/NCB
      IF (IASHAP.EQ.5HLOWER) IADENS=100*(2*(NNZA-NRA)+NRA)/NRA/NCA
      IF (IASHAP.EQ.5HUPPER) IADENS=100*(2*(NNZA-NRA)+NRA)/NRA/NCA
      IF (IBSHAP.EQ.5HLOWER) IBDENS=100*(2*(NNZB-NRB)+NRB)/NRB/NCB
      IF (IBSHAP.EQ.5HUPPER) IBDENS=100*(2*(NNZB-NRB)+NRB)/NRB/NCB
      IF (NNZA.EQ.0 .OR. NNZB.EQ.0) GO TO 95
```

```
C
      REWIND NUTZ
      MHEAD(1) = NCB
      MHEAD(2) = NRB
      IF (IBSHAP.EQ.5HWHOLE) MHEAD(5) = 0
      IF (IBSHAP.EQ.5HWHOLE) MHEAD(6) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
C
      DO 10 I=1,NPARTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NNZP = MHEAD(1)
      LFELP = MHEAD(2)
      LLELP = MHEAD(3)
      CALL YINI   (NUTB,LV,1,NNZP)
      CALL YIN    (NUTB,V,1,NNZP)
      IF (IBSHAP.NE.5HWHOLE) GO TO 6
C
      DO 5 J=1,NNZP
      K=LV(J)/100000
    5 LV(J)=100000*(LV(J)-100000*K)+K
C
    6 MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZP)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,1,NNZP)
   10 CALL YOUT   (NUTZ,V,1,NNZP)

      CALL YMULT (NUTZ,NUTA,NUT1,V,LV,KV,NUT2)
      IF (IASHAP.EQ.5HWHOLE) GO TO 85
C
C  SYMMETRY OF A IS USED FROM HERE TO STATEMENT 85
      NPARTZ=0
      NNZZ=0
      NREC=0
      CALL YLORD  (NUTB,V,LV,KV,NUT2,NUTZ)
      REWIND NUT1
      CALL YINI   (NUT1,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
      NPARTB = MHEAD(3)
      NNZB = MHEAD(4)
      ISHAP = MHEAD(7)
      IF (ISHAP.EQ.5HWHOLE .OR. ISHAP.EQ.4HDIAG) GO TO 15
      IF (ISHAP.EQ.5HLOWER) CALL YSYMUH (NUTB,V,LV,KV,NUT2,NUTZ)
      IF (ISHAP.EQ.5HUPPER) CALL YSYMLH (NUTB,V,LV,KV,NUT2,NUTZ)
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
```

```
      NPARTB = MHEAD(3)
      NNZB = MHEAD(4)
   15 IF (NNZA.EQ.0 .OR. NNZB.EQ.0) GO TO 70
                                                              NERROR=1
      IF (NCA.GT.KV/4 .OR. NCB.GT.KV/4) GO TO 999
                                                              NERROR=2
      IF (NRB.NE.NCA) GO TO 999
C
      IZ=0
      LPBS=KV/4+1
      LPBE=LPBS-1
      LCS=KV/2+1
      LCE=LCS-1+NCB
      LCCS=LCE+1
      LCCE=LCE
      NNZ=KV-LCCS+1
      REWIND NUT2
C
      DO 20  I=LCS,LCE
   20 V(I)=0.
C
      DO 55 I=1,NPARTA
      CALL YINI   (NUT1,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
      CALL YINI   (NUT1,LV,1,NNZPA)
      CALL YIN    (NUT1,V,1,NNZPA)
      K=LPBS
      ITRBL=0
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NREAD=0
C
      DO 50 INA=1,NNZPA
      IA=LV(INA)/100000
      JA=LV(INA)-100000*IA
      IF (IA.EQ.IZ .AND. ITRBL.EQ.1) GO TO 50
      ITRBL=0
      IF (IA.EQ.IZ) GO TO 30
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NREAD=0
C
      DO 25  INC=LCS,LCE
      IF (V(INC).EQ.0.) GO TO 25
      LCCE=LCCE+1
      V(LCCE)=V(INC)
      LV(LCCE)=IZZ+INC-KV/2
      V(INC)=0.
      IF (LCCE.LT.KV) GO TO 25
      CALL YOUTI  (NUT2,LV,LCCS,LCCE)
      CALL YOUT   (NUT2,V,LCCS,LCCE)
      NREC=NREC+1
      NNZZ=NNZZ+NNZ
```

```
      LCCE=LCE
   25 CONTINUE
C
      IZ=IA
      IZZ=100000*IZ
      K=LPBS
   30 IF (K.LE.LPBE .AND. NREAD.GT.0) GO TO 40
   35 IF (NREAD.EQ.NPARTB) ITRBL=1
      IF (ITRBL.EQ.1) GO TO 50
      CALL YINI    (NUTB,MHEAD,1,10)
      NNZPB = MHEAD(1)
      LFELPB = MHEAD(2)
      LLELPB = MHEAD(3)
      LPBE=LPBS-1+NNZPB
      CALL YINI    (NUT3,LV,LPBS,LPBE)
      CALL YIN     (NUTB,V,LPBS,LPBE)
      NREAD=NREAD+1
      K=LPBS
C
   40 DO 45  INB=K,LPBE
      K=INB
      IB=LV(INB)/100000
      IF (IB.GT.JA) GO TO 50
      IF (IB.LT.JA) GO TO 45
      JBZ=LV(INB)-100000*IB
      IF (JBZ.GT.IA) GO TO 45
      INZ=KV/2+JBZ
      V(INZ)=V(INZ)+V(INA)*V(INB)
   45 CONTINUE
C
      GO TO 35
   50 CONTINUE
C
   55 CONTINUE
C
      DO 60 I=LCS,LCF
      IF (V(I).EQ.0.) GO TO 60
      LCCE=LCCE+1
      V(LCCE)=V(I)
      LV(LCCE)=IZZ+I-KV/2
      IF (LCCE.LT.KV) GO TO 60
      CALL YOUTI  (NUT2,LV,LCCS,LCCE)
      CALL YOUT   (NUT2,V,LCCS,LCCE)
      NREC=NREC+1
      LCCE=LCE
      NNZZ=NNZZ+NNZ
   60 CONTINUE
C
      IF (LCCE.EQ.LCE) GO TO 70
      NNZ=LCCE-LCCS+1
      NNZZ=NNZZ+NNZ
      NREC=NREC+1
      CALL YOUTI  (NUT2,LV,LCCS,LCCE)
      CALL YOUT   (NUT2,V,LCCS,LCCE)
   70 REWIND NUT2
```

```
      MHEAD(1) = NRA
      MHEAD(2) = NCB
      MHEAD(3) = NREC
      MHEAD(4) = NNZZ
      MHEAD(5) = 5HORDER
      MHEAD(6) = 0
      MHEAD(7) = 5HLOWER
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      IF (NNZZ.GT.0) GO TO 75
      DO 71 I=1,10
      V(I) = 0.
   71 MHEAD(I) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,2)
      CALL YOUT   (NUTZ,    V,1,2)
      RETURN
   75 LZE=KV
      REWIND NUT2
C
      DO 80  I=1,NREC
      IF (I.EQ.NREC) LZE=LCCS-1+NNZ
      NNZP = LZE-LCCS+1
      CALL YINI   (NUT2,LV,LCCS,LZE)
      CALL YIN    (NUT2,V,LCCS,LZE)
      MHEAD(1) = NNZP
      MHEAD(2) = LV(LCCS)
      MHEAD(3) = LV(LZE)
      DO 76 J=4,10
   76 MHEAD(J) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,LCCS,LZE)
   80 CALL YOUT   (NUTZ,V,LCCS,LZE)
C
      CALL YPART  (NUTZ,V,LV,KV,NUT1)
      GO TO 90
   85 CALL YMULT  (NUT1,NUTB,NUTZ,V,LV,KV,NUT2)
   90 RETURN
C
   95 REWIND NUTZ
      NNZZ=0
      MHEAD(1) = NCB
      MHEAD(2) = NCB
      MHEAD(3) = NNZZ
      MHEAD(4) = NNZZ
      MHEAD(5) = 5HORDER
      MHEAD(6) = KV
      MHEAD(7) = 5HWHOLE
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      DO 96 I=1,10
      V(I) = 0.
   96 MHEAD(I) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,2)
      CALL YOUT   (NUTZ,    V,1,2)
      RETURN
```

```
C
  999 CALL ZZBOMB (5HYBTAB ,NERROR)
      END
```

```
      SUBROUTINE YDCM3A (NUTA,NUTU,NUTD,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10),MPHEAD(10)
      DATA EPS/1.E-20/
C
C  DECOMPOSE SPARSE MATRIX (A) TO FORM UPPER TRIANGULAR MATRIX WITH ONES
C  ON DIAGONAL (U) AND DIAGONAL MATRIX (-D-) SUCH THAT
C  (A) = (U)**T * (-D-) * (U). METHOD ATTRIBUTED TO GAUSS.
C  SPECIAL FORM USED FOR FORMA SUBROUTINE YMOD2A.
C  IF THE WHOLE MATRIX (A) IS INPUT, ONLY THE LOWER HALF IS USED.
C  BAND WIDTH (DIAGONAL UP TO TOP NON-ZERO) MUST BE LESS THAN OR EQUAL
C  TO (KV-N)/2, WHERE N IS MATRIX SIZE (SQUARE).
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLCRD ,YOUT  ,YOUTI ,YPART ,
C                          YTRANS,ZZBOMB.
C  DEVELOPED BY R L WOHLEN AND R A PHILIPPUS.  NOVEMBER 1971.
C  LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTU  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH U IS STORED.
C  NUTD  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH D IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = BANDWIDTH LIMITATION EXCEEDED.
C  2 = SIZE LIMITATION EXCEEDED (LV).
C  3 = SIZE LIMITATION EXCEEDED (LV).
C  4 = MATRIX IS SINGULAR.
C
C  CONVERT FROM SPARSE TO BAND NOTATION.
      KV04=KV/4
      KV02 = KV/2
      KV02P1 = KV02+1
      LAS=KV04+1
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      KVMN = KV-NRA
      KVMN02 = KVMN/2
      IASHAP = MHEAD(7)
      NUTS=NUTA
      IF (IASHAP.EQ.5HUPPER) CALL YTRANS (NUTA,NUTU,V,LV,KV,NUT1,NUT2)
      IF (IASHAP.EQ.5HUPPER) NUTS=NUTU
      CALL YLCRD  (NUTS,V,LV,KV,NUT1,NUT2)
      REWIND NUTS
      CALL YINI   (NUTS,MHEAD,1,10)
      REWIND NUT1
      ILV = KV04
      JLV = KV04+NRA
      IF (JLV.LT.KV02) JLV=KV02
      JLVS = JLV
      KP = 1
```

```
      LAMAX = LAS-1+KVMNO2
      LAE = LAS
      JS = 1
      NGROUP = 0
      LAS1 = KV04
      DO 5 I=LAS,KV
      LV(I) = 0
    5 V(I)=0.
      NNZZ = 0
      NPARTA = MHEAD(3)
      NROWS = 1
      DO 20 I=1,NPARTA
      CALL YINI   (NUTS,MPHEAD,1,10)
      NNZPA = MPHEAD(1)
      CALL YINI   (NUTS,LV,1,NNZPA)
      CALL YIN    (NUTS,V,1,NNZPA)
      DO 20 J=1,NNZPA
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (IA.LT.JA) GO TO 20
      IF (IA.EQ.KP) GO TO 15
      LAS1 = LAE
      LAE = LAE+IA-JA+1
      NELR = KP-JS+1
                                                          NERROR=1
      IF (NELR.GT.KVMNO2) GO TO 999
      NNZZ = NNZZ+NELR
      KP = KP+1
      JS = JA
      NROWS = NROWS+1
      ILV = ILV+1
      LV(ILV) = NELR
      IF (LAE.LE.LAMAX) GO TO 15
      JLV = JLV+1
                                                          NERROR=2
      IF (JLV.GT.KV) GO TO 999
      NROWS = NROWS-1
      LV(JLV) = NROWS
      NROWS = 1
      LAE = LAE-IA+JA-1
      NGROUP = NGROUP+1
      CALL YOUT   (NUT1,V,LAS,LAE)
      DO 10 L=LAS,LAE
   10 V(L)=0.
      LAS1 = KV04
      LAE = KV04+IA-JA+1
      KP = IA
   15 LA = LAS1+JA-JS+1
      V(LA)=V(J)
   20 CONTINUE
      IF (LAS.GT.LAE) GO TO 65
      NGROUP = NGROUP+1
      ILV = ILV+1
      LV(ILV) = KP-JS+1
                                                          NERROR=3
```

```
      IF (LV(ILV).GT.KVMNO2) GO TO 999
      NNZZ = NNZZ+LV(ILV)
      JLV = JLV+1
      IF (JLV.GT.KV) GO TO 999
      LV(JLV) = NROWS
      CALL YOUT   (NUT1,V,LAS,LAE)
   65 DO 30 I=1,NRA
   30 LV(I) = LV(KVO4+I)
      DO 40 I=1,NGROUP
   40 LV(KVO2+I) = LV(JLVS+I)
C
C  DECOMPOSITION.
C  D IN V(1 THRU N). A,U GROUP A START AT V(N+1).
C  A,U GROUP B START AT V(N+1+(KV-N)/2).
C  LV(I),I=1,N IS NUMBER OF ELEMENTS IN COLUMN I.
C  LV(KV/2+IG) IS NUMBER OF COLUMNS IN GROUP IG.
      N = NRA
      NG = NGROUP
      LSGA = N+1
      LSGB = LSGA + (KV-N)/2
      REWIND NUTU
      MHEAD(1) = N
      MHEAD(2) = N
      MHEAD(3) = NG
      CALL YOUTI  (NUTU,MHEAD,1,10)
      CALL YOUTI  (NUTU,LV,1,N)
      JEGA = 0
      DO 195 IGA=1,NG
      REWIND NUT1
      REWIND NUT2
      NUTP = NUT1
      NUTQ = NUT2
      IF (2*(IGA/2) .EQ. IGA) NUTP=NUT2
      IF (NUTP .EQ. NUT2)     NUTQ=NUT1
C  OPERATE ON GROUP A ONLY.
      NCGA = LV(KVO2+IGA)
      JSGA = JEGA+1
      JEGA = JSGA+NCGA-1
      LEGA = LSGA-1
      DO 101 J=JSGA,JEGA
  101 LEGA = LEGA + LV(J)
      CALL YIN    (NUTP,V,LSGA,LEGA)
      LJJ = LSGA-1
      DO 140 J=JSGA,JEGA
      JM1 = J-1
      ITOPJ = J-LV(J)+1
      LITOPJ = LJJ+1
      LJJ = LITOPJ+LV(J)-1
      IF (J .EQ. JSGA) GO TO 134
      IF (ITOPJ .EQ. J) GO TO 134
      ISTART = ITOPJ
      LIJ = LITOPJ-1
      IF (ITOPJ .GE. JSGA) GO TO 105
      ISTART = JSGA
      LIJ = LITOPJ+JSGA-ITOPJ-1
```

```
  105 LITOPI = LSGA
      IF (ISTART .EQ. JSGA) GO TO 110
      ISM1 = ISTART-1
      DO 107 I=JSGA,ISM1
  107 LITOPI = LITOPI+LV(I)
  110 DO 128 I=ISTART,JM1
      LIJ = LIJ+1
      IM1 = I-1
      ITOPI = I-LV(I)+1
      IF (ITOPI .LT. ITOPJ) GO TO 115
      KSTART = ITOPI
      IF (I .EQ. KSTART) GO TO 125
      LKI = LITOPI-1
      LKJ = LITOPJ+ITOPI-ITOPJ-1
      GO TO 120
  115 KSTART = ITOPJ
      IF (I .EQ. KSTART) GO TO 125
      LKI = LITOPI+ITOPJ-ITOPI-1
      LKJ = LITOPJ-1
  120 DO 122 K=KSTART,IM1
      LKI = LKI+1
      LKJ = LKJ+1
  122 V(LIJ) = V(LIJ) - V(K)*V(LKI)*V(LKJ)
  125 V(LIJ) = V(LIJ)/V(I)
  128 LITOPI = LITOPI+LV(I)
  134 V(J) = V(LJJ)
      IF (ITOPJ .EQ. J) GO TO 139
      LKJ = LITOPJ-1
      DO 138 K=ITOPJ,JM1
      LKJ = LKJ+1
  138 V(J) = V(J) - V(K)*V(LKJ)**2
  139                                                          NERROR=4
      IF (ABS(V(J)).LT.EPS) GO TO 999
  140 V(LJJ) = 1.0
C   GROUP A OPERATE ON GROUP B.
C   I COLUMNS ARE IN GROUP A, J COLUMNS IN GROUP B.
      IF (IGA .EQ. NG) GO TO 194
      IGAP1 = IGA+1
      JEGB = JEGA
      DO 192 IGB=IGAP1,NG
      NCGB = LV(KV02+IGB)
      JSGB = JEGB+1
      JEGB = JSGB+NCGB-1
      LEGB = LSGB-1
      DO 151 J=JSGB,JEGB
  151 LEGB = LEGB+LV(J)
      CALL YIN     (NUTP,V,LSGB,LEGB)
      LJJ = LSGB-1
      DO 190 J=JSGB,JEGB
      JM1 = J-1
      ITOPJ = J-LV(J)+1
      LITOPJ = LJJ+1
      LJJ = LITOPJ+LV(J)-1
      IF (ITOPJ .GT. JEGA) GO TO 190
      ISTART = ITOPJ
```

```
      LIJ = LITOPJ-1
      IF (ITOPJ .GE. JSGA) GO TO 155
      ISTART = JSGA
      LIJ = LITOPJ+JSGA-ITOPJ-1
  155 LITOPI = LSGA
      IF (ISTART .EQ. JSGA) GO TO 160
      ISM1 = ISTART-1
      DO 157 I=JSGA,ISM1
  157 LITOPI = LITOPI+LV(I)
  160 DO 178 I=ISTART,JEGA
      LIJ = LIJ+1
      IM1 = I-1
      ITOPI = I-LV(I)+1
      IF (ITOPI .LT. ITOPJ) GO TO 165
      KSTART = ITOPI
      IF (I .EQ. KSTART) GO TO 175
      LKI = LITOPI-1
      LKJ = LITOPJ+ITOPI-ITOPJ-1
      GO TO 170
  165 KSTART = ITOPJ
      IF (I .EQ. KSTART) GO TO 175
      LKI = LITOPI+ITOPJ-ITOPI-1
      LKJ = LITOPJ-1
  170 DO 172 K=KSTART,IM1
      LKI = LKI+1
      LKJ = LKJ+1
  172 V(LIJ) = V(LIJ) - V(K)*V(LKI)*V(LKJ)
  175 V(LIJ) = V(LIJ)/V(I)
  178 LITOPI = LITOPI+LV(I)
  190 CONTINUE
  192 CALL YOUT    (NUTQ,V,LSGB,LEGB)
  194 MPHEAD(1) = JSGA
      MPHEAD(2) = JEGA
      MPHEAD(3) = LEGA-LSGA+1
      CALL YOUTI  (NUTU,MPHEAD,1,10)
  195 CALL YOUT    (NUTU,V,LSGA,LEGA)
      REWIND NUTD
      CALL YOUT    (NUTD,V,1,N)
      RETURN
  999 CALL ZZBOMB (6HYDCM3A,NERROR)
      END
```

```
      SUBROUTINE YDCOM3 (NUTA,NUTU,NUTD,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10),MPHEAD(10)
      DATA EPS/1.E-20/
C
C   DECOMPOSE SPARSE MATRIX (A) TO FORM UPPER TRIANGULAR MATRIX WITH ONES
C   ON DIAGONAL (U) AND DIAGONAL MATRIX (-D-) SUCH THAT
C   (A) = (U)**T * (-D-) * (U). METHOD ATTRIBUTED TO GAUSS.
C   IF THE WHOLE MATRIX (A) IS INPUT, ONLY THE LOWER HALF IS USED.
C   BAND WIDTH (DIAGONAL UP TO TOP NON-ZERO) MUST BE LESS THAN OR EQUAL
C   TO (KV-N)/2, WHERE N IS MATRIX SIZE (SQUARE).
C   CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                           YTRANS,ZZBOMB.
C   DEVELOPED BY R L WOHLEN AND R A PHILIPPUS.  DECEMBER 1972.
C   LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C   NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C   NUTU  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH U IS STORED.
C   NUTD  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH D IS STORED.
C   V     = VECTOR WORK SPACE.
C   LV    = VECTOR WORK SPACE.
C   KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C   NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C   NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C      NERROR  EXPLANATION
C   1 = BANDWIDTH LIMITATION EXCEEDED.
C   2 = SIZE LIMITATION EXCEEDED (LV).
C   3 = BANDWIDTH LIMITATION EXCEEDED.
C   4 = MATRIX IS SINGULAR.
C
C   CONVERT FROM SPARSE TO BAND NOTATION.
      KVO4=KV/4
      KVO2 = KV/2
      KVO2P1 = KVO2+1
      LAS=KVO4+1
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      KVMN = KV-NRA
      KVMNO2 = KVMN/2
      IASHAP = MHEAD(7)
      NUTS=NUTA
      IF (IASHAP.EQ.5HUPPER) CALL YTRANS (NUTA,NUTU,V,LV,KV,NUT1,NUT2)
      IF (IASHAP.EQ.5HUPPER) NUTS=NUTU
      CALL YLORD  (NUTS,V,LV,KV,NUT1,NUT2)
      REWIND NUTS
      CALL YINI   (NUTS,MHEAD,1,10)
      REWIND NUT1
      ILV = KVO4
      JLV = KVO4+NRA
      IF (JLV.LT.KVO2) JLV=KVO2
      JLVS = JLV
      KP = 1
      LAMAX = LAS-1+KVMNO2
```

```
      LAE = LAS
      JS = 1
      NGROUP = 0
      LAS1 = KV04
      DO 5 I=LAS,KV
      LV(I) = 0
    5 V(I)=0.
      NNZZ = 0
      NPARTA = MHEAD(3)
      NROWS = 1
      DO 20 I=1,NPARTA
      CALL YINI    (NUTS,MPHEAD,1,10)
      NNZPA = MPHEAD(1)
      CALL YINI    (NUTS,LV,1,NNZPA)
      CALL YIN     (NUTS,V,1,NNZPA)
      DO 20 J=1,NNZPA
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (IA.LT.JA) GO TO 20
      IF (IA.EQ.KP) GO TO 15
      LAS1 = LAE
      LAE = LAE+IA-JA+1
      NELR = KP-JS+1
                                                        NERROR=1
      IF (NELR.GT.KVMNO2) GO TO 999
      NNZZ = NNZZ+NELR
      KP = KP+1
      JS = JA
      NROWS = NROWS+1
      ILV = ILV+1
      LV(ILV) = NELR
      IF (LAE.LE.LAMAX) GO TO 15
      JLV = JLV+1
                                                        NERROR=2
      IF (JLV.GT.KV) GO TO 999
      NROWS = NROWS-1
      LV(JLV) = NROWS
      NROWS = 1
      LAE = LAE-IA+JA-1
      NGROUP = NGROUP+1
      CALL YOUT    (NUT1,V,LAS,LAE)
      DO 10 L=LAS,LAE
   10 V(L)=0.
      LAS1 = KV04
      LAE = KV04+IA-JA+1
      KP = IA
   15 LA = LAS1+JA-JS+1
      V(LA)=V(J)
   20 CONTINUE
      IF (LAS.GT.LAE) GO TO 65
      NGROUP = NGROUP+1
      ILV = ILV+1
      LV(ILV) = KP-JS+1
                                                        NERROR=3
      IF (LV(ILV).GT.KVMNO2) GO TO 999
```

```
      NNZZ = NNZZ+LV(ILV)
      JLV = JLV+1
      IF (JLV.GT.KV) GO TO 999
      LV(JLV) = NROWS
      CALL YOUT    (NUT1,V,LAS,LAE)
   65 DO 30 I=1,NRA
   30 LV(I) = LV(KVO4+I)
      DO 40 I=1,NGROUP
   40 LV(KVO2+I) = LV(JLVS+I)
C
C  DECOMPOSITION.
C  D IN V(1 THRU N). A,U GROUP A START AT V(N+1).
C  A,U GROUP B START AT V(N+1+(KV-N)/2).
C  LV(I),I=1,N IS NUMBER OF ELEMENTS IN COLUMN I.
C  LV(KV/2+IG) IS NUMBER OF COLUMNS IN GROUP IG.
      N = NRA
      NG = NGROUP
      LSGA = N+1
      LSGB = LSGA + (KV-N)/2
      REWIND NUTD
      JEGA = 0
      DO 195 IGA=1,NG
      REWIND NUT1
      REWIND NUT2
      NUTP = NUT1
      NUTQ = NUT2
      IF (2*(IGA/2) .EQ. IGA) NUTP=NUT2
      IF (NUTP .EQ. NUT2)     NUTQ=NUT1
C  OPERATE ON GROUP A ONLY.
      NCGA = LV(KVO2+IGA)
      JSGA = JEGA+1
      JEGA = JSGA+NCGA-1
      LEGA = LSGA-1
      DO 101 J=JSGA,JEGA
  101 LEGA = LEGA + LV(J)
      CALL YIN     (NUTP,V,LSGA,LEGA)
      LJJ = LSGA-1
      DO 140 J=JSGA,JEGA
      JM1 = J-1
      ITOPJ = J-LV(J)+1
      LITOPJ = LJJ+1
      LJJ = LITOPJ+LV(J)-1
      IF (J .EQ. JSGA) GO TO 134
      IF (ITOPJ .EQ. J) GO TO 134
      ISTART = ITOPJ
      LIJ = LITOPJ-1
      IF (ITOPJ .GE. JSGA) GO TO 105
      ISTART = JSGA
      LIJ = LITOPJ+JSGA-ITOPJ-1
  105 LITOPI = LSGA
      IF (ISTART .EQ. JSGA) GO TO 110
      ISM1 = ISTART-1
      DO 107 I=JSGA,ISM1
  107 LITOPI = LITOPI+LV(I)
  110 DO 128 I=ISTART,JM1
```

```
      LIJ = LIJ+1
      S = V(LIJ)
      IM1 = I-1
      ITOPI = I-LV(I)+1
      IF (ITOPI .LT. ITOPJ) GO TO 115
      KSTART = ITOPI
      IF (I .EQ. KSTART) GO TO 125
      LKI = LITOPI-1
      LKJ = LITOPJ+ITOPI-ITOPJ-1
      GO TO 120
  115 KSTART = ITOPJ
      IF (I .EQ. KSTART) GO TO 125
      LKI = LITOPI+ITOPJ-ITOPI-1
      LKJ = LITOPJ-1
  120 DO 122 K=KSTART,IM1
      LKI = LKI+1
      LKJ = LKJ+1
  122 S = S-V(K)*V(LKI)*V(LKJ)
  125 V(LIJ) = S/V(I)
  128 LITOPI = LITOPI+LV(I)
  134 V(J) = V(LJJ)
      IF (ITOPJ .EQ. J) GO TO 139
      LKJ = LITOPJ-1
      DO 138 K=ITOPJ,JM1
      LKJ = LKJ+1
  138 V(J) = V(J) - V(K)*V(LKJ)**2
  139                                                         NERROR=4
      IF (ABS(V(J)).LT.EPS) GO TO 999
  140 V(LJJ) = 1.0
C  GROUP A OPERATE ON GROUP B.
C  I COLUMNS ARE IN GROUP A, J COLUMNS IN GROUP B.
      IF (IGA .EQ. NG) GO TO 195
      IGAP1 = IGA+1
      JEGB = JEGA
      DO 192 IGB=IGAP1,NG
      NCGB = LV(KV02+IGB)
      JSGB = JEGB+1
      JEGB = JSGB+NCGB-1
      LEGB = LSGB-1
      DO 151 J=JSGB,JEGB
  151 LEGB = LEGB+LV(J)
      CALL YIN    (NUTP,V,LSGB,LEGB)
      LJJ = LSGB-1
      DO 190 J=JSGB,JEGB
      JM1 = J-1
      ITOPJ = J-LV(J)+1
      LITOPJ = LJJ+1
      LJJ = LITOPJ+LV(J)-1
      IF (ITOPJ .GT. JEGA) GO TO 190
      ISTART = ITOPJ
      LIJ = LITOPJ-1
      IF (ITOPJ .GE. JSGA) GO TO 155
      ISTART = JSGA
      LIJ = LITOPJ+JSGA-ITOPJ-1
  155 LITOPI = LSGA
```

```
      IF (ISTART .EQ. JSGA) GO TO 160
      ISM1 = ISTART-1
      DO 157 I=JSGA,ISM1
  157 LITOPI = LITOPI+LV(I)
  160 DO 178 I=ISTART,JEGA
      LIJ = LIJ+1
      S = V(LIJ)
      IM1 = I-1
      ITOPI = I-LV(I)+1
      IF (ITOPI .LT. ITOPJ) GO TO 165
      KSTART = ITOPI
      IF (I .EQ. KSTART) GO TO 175
      LKI = LITOPI-1
      LKJ = LITOPJ+ITOPI-ITOPJ-1
      GO TO 170
  165 KSTART = ITOPJ
      IF (I .EQ. KSTART) GO TO 175
      LKI = LITOPI+ITOPJ-ITOPI-1
      LKJ = LITOPJ-1
  170 DO 172 K=KSTART,IM1
      LKI = LKI+1
      LKJ = LKJ+1
  172 S = S-V(K)*V(LKI)*V(LKJ)
  175 V(LIJ) = S/V(I)
  178 LITOPI = LITOPI+LV(I)
  190 CONTINUE
  192 CALL YOUT   (NUTQ,V,LSGB,LEGB)
  195 CALL YOUT   (NUTD,V,LSGA,LEGA)
C
C  CONVERT FROM BAND TO SPARSE NOTATION.
      REWIND NUT2
      CALL YOUT   (NUT2,V,1,N)
      REWIND NUTU
      REWIND NUTD
      LVGS = KV-NGROUP
      LVR = LVGS
      DO 202 IGROUP=1,NGROUP
      LVR = LVR+1
  202 LV(LVR) = LV(KV02+IGROUP)
      LS = LVGS-N
      LVE = LS
      DO 204 I=1,N
      LVE = LVE+1
  204 LV(LVE) = LV(I)
      KVMAX = KV/4
      IF (KVMAX.GT.LS) KVMAX=LS
      MHEAD(1) = N
      MHEAD(2) = N
      MHEAD(3) = NGROUP
      MHEAD(4) = NNZZ
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = 5HWHOLE
      CALL YOUTI  (NUTU,MHEAD,1,10)
      LVI = 0
```

```
      LVR = LVGS
      LVE = LS
      LVEP = LS
      IZ = 0
      DO 250 IGROUP=1,NGROUP
      LVR = LVR+1
      LZ = 0
      NROWS = LV(LVR)
      NELG = 0
      DO 206 I=1,NROWS
      LVE = LVE+1
  206 NELG = NELG+LV(LVE)
      CALL YIN    (NUTD,V,1,NELG)
      DO 208 I=1,NROWS
      IZ = IZ+1
      LVEP = LVEP+1
      JS = IZ-LV(LVEP)+1
      DO 208 JZ=JS,IZ
      LZ = LZ+1
  208 LV(LZ) = 100000*JZ+IZ
      MPHEAD(1) = LZ
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(LZ)
      CALL YOUTI  (NUTU,MPHEAD,1,10)
      CALL YOUTI  (NUTU,LV,1,LZ)
      CALL YOUT   (NUTU,V,1,LZ)
  250 CONTINUE
      REWIND NUT2
      REWIND NUTD
      CALL YIN    (NUT2,V,1,N)
      DO 260 I=1,N
  260 LV(I) = 100000*I+I
      MHEAD(3) = 1
      MHEAD(4) = N
      MHEAD(7) = 4HDIAG
      CALL YOUTI  (NUTD,MHEAD,1,10)
      MPHEAD(1) = N
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(N)
      CALL YOUTI  (NUTD,MPHEAD,1,10)
      CALL YOUTI  (NUTD,LV,1,N)
      CALL YOUT   (NUTD,V,1,N)
      CALL YPART  (NUTU,V,LV,KV,NUTI)
      RETURN
  999 CALL ZZBOMB (6HYDCOM3,NERROR)
      END
```

```
      SUBROUTINE YDIAG (NUTA,NUTZ,V,LV,KV)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  DIAGONALIZE A SPARSE VECTOR (ROW OR COLUMN) INTO A SQUARE MATRIX.
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,YOUT   ,YOUTI ,ZZBOMB.
C  DEVELOPED BY W A BENFIELD.  OCTOBER 1970.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C
C     NERROR  EXPLANATION
C  1 = MATRIX IS NOT A VECTOR.
C  2 = SIZE LIMITATION EXCEEDED.
C
      REWIND NUTZ
      REWIND NUTA
      CALL YINI    (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPART = MHEAD(3)
      MHEAD(7) = 4HDIAG
                                                           RROR=1
      IF (NRA.NE.1 .AND. NCA.NE.1) GO TO 999
      IF (NRA .EQ. 1) NRA = NCA
      IF (NCA .EQ. 1) NCA = NRA
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      CALL YOUTI  (NUTZ,MHEAD,1,10)
                                                         NERROR=2
      DO 30 I=1,NPART
      CALL YINI    (NUTA,MHEAD,1,10)
      NNZP = MHEAD(1)
      IF (NNZP .GT. KV) GO TO 999
      CALL YINI    (NUTA,LV,1,NNZP)
      CALL YIN     (NUTA,V,1,NNZP)
C
      DO 20 K=1,NNZP
      IA = LV(K)/100000
      IF (IA.EQ.1) GO TO 10
      LV(K) = 100000*IA + IA
      GO TO 20
   10 JA=LV(K)-100000*IA
      LV(K) = 100000*JA + JA
   20 CONTINUE
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZP)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,1,NNZP)
   30 CALL YOUT   (NUTZ,V,1,NNZP)
```

```
C
      RETURN
  999 CALL ZZBOMB (5HYDIAG ,NERROR)
      END
```

```
      SUBROUTINE YDISA (NUTA,IRA,JCA,NUTZ,NRZ,NCZ,V,LV,KV,NUT1)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  SPARSE MATRIX DISASSEMBLY. (MATRIX Z FROM MATRIX A).
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                          ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  FEBRUARY 1970.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  IRA   = ROW NUMBER IN MATRIX A OF FIRST ROW OF MATRIX Z.
C  JCA   = COLUMN NUMBER IN MATRIX A OF FIRST COLUMN OF MATRIX Z.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  NRZ   = NUMBER OF ROWS IN MATRIX Z.
C  NCZ   = NUMBER OF COLUMNS IN MATRIX Z.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = LOOKING FOR DATA OUTSIDE OF MATRIX A.
C
      CALL YLORD  (NUTA,V,LV,KV,NUT1,NUTZ)
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      ISHAP  = MHEAD(7)
      IF (IRA.EQ.JCA .OR. ISHAP.EQ.5HWHOLE) GO TO 5
      IF (ISHAP.EQ.5HLOWER) CALL YSYMUH (NUTA,V,LV,KV,NUT1,NUTZ)
      IF (ISHAP.EQ.5HUPPER) CALL YSYMLH (NUTA,V,LV,KV,NUT1,NUTZ)
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      ISHAP = MHEAD(7)
    5 LZS=KV/4+1
      LZF=LZS-1+KV/4
      LZ=LZS-1
      IRAZ=IRA-1+NRZ
      JCAZ=JCA-1+NCZ
                                                       NERROR=1
      IF (IRAZ.GT.NRA .OR. JCAZ.GT.NCA) GO TO 999
      IJZ=100000*(IRA-1)+JCA-1
      NNZZ=0
      NPARTZ=0
      REWIND NUT1
C
```

```
      DO 30 I=1,NPARTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
      CALL YINI   (NUTA,LV,1,NNZPA)
      CALL YIN    (NUTA,V,1,NNZPA)
      IAF=LFELPA/100000
      IF (IAF.GT.IRAZ .AND. I.LT.NPARTA) GO TO 30
      IAL=LLELPA/100000
      IF (IAL.LT.IRA) GO TO 30
C
      DO 20 J=1,NNZPA
      IA=LV(J)/100000
      IF (IA.LT.IRA .OR. IA.GT.IRAZ) GO TO 10
      JA=LV(J)-100000*IA
      IF (JA.LT.JCA .OR. JA.GT.JCAZ) GO TO 10
      IF (V(J).EQ.0.) GO TO 10
      LZ=LZ+1
      V(LZ)=V(J)
      LV(LZ)=LV(J)-IJZ
      NNZZ=NNZZ+1
   10 IF (LZ.GE.LZE) GO TO 15
      IF (I.LT.NPARTA .OR. J.LT.NNZPA) GO TO 20
   15 N=LZ-LZS+1
      IF (LZ.LT.LZS) GO TO 20
      MHEAD(1) = N
      MHEAD(2) = LV(LZS)
      MHEAD(3) = LV(LZ)
      CALL YOUTI  (NUT1,MHEAD,1,10)
      CALL YOUTI  (NUT1,LV,LZS,LZ)
      CALL YOUT   (NUT1,V,LZS,LZ)
      LZ=LZS-1
      NPARTZ=NPARTZ+1
   20 CONTINUE
   30 CONTINUE
C
      REWIND NUT1
      REWIND NUTZ
      MHEAD(1) = NRZ
      MHEAD(2) = NCZ
      MHEAD(3) = NPARTZ
      MHEAD(4) = NNZZ
      MHEAD(5) = 5HORDER
      MHEAD(7) = ISHAP
      CALL YOUTI  (NUTZ,MHEAD,1,10)
C
      DO 40 I=1,NPARTZ
      CALL YINI   (NUT1,LV,1,10)
      CALL YOUTI  (NUTZ,LV,1,10)
      N=LV(1)
      CALL YINI   (NUT1,LV,1,N)
      CALL YIN    (NUT1,V,1,N)
      CALL YOUTI  (NUTZ,LV,1,N)
   40 CALL YOUT   (NUTZ,V,1,N)
```

```
C
      CALL YPART  (NUTZ,V,LV,KV,MUTI)
      RETURN
C
  999 CALL ZZBOMB (5HYDISA ,NERROR)
      END
```

```
      SUBROUTINE YDTOS (A,NUTA,NRA,NCA,KRA,KCA,V,LV,KV,NUT1)
      DIMENSION V(1), LV(1), A(KRA,1), MHEAD(10), MPHEAD(10)
      DATA NIT,NOT/5,6/
C
C  CONVERT A MATRIX FROM DENSE NOTATION TO SPARSE NOTATION.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YOUT  ,YOUTI ,YPART ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1969.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  A     = DENSE MATRIX. SIZE (NRA,NCA).
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH SPARSE MATRIX A WILL
C          BE STORED.
C  NRA   = NUMBER OF ROWS IN A.
C  NCA   = NUMBER OF COLUMNS IN A.
C  KRA   = ROW DIMENSION OF A IN CALLING PROGRAM.
C  KCA   = COLUMN DIMENSION OF A IN CALLING PROGRAM.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED (KA).
C
                                                         NERROR=1
      IF (NRA.GT.KRA .OR. NCA.GT.KCA) GO TO 999
      L=0
      REWIND NUT1
      NNZA=0
      NREC=0
      DO 2 I=1,10
      MHEAD(I) = 0
    2 MPHEAD(I) = 0
C
C  STORE PARTITIONS ON NUT1 TEMPORARILY.
      DO 20 I=1,NRA
      DO 10 J=1,NCA
      IF (A(I,J).EQ.0.) GO TO 3
      L=L+1
      V(L)=A(I,J)
      LV(L)=100000*I+J
    3 IF (L.GE.KV/4) GO TO 5
      IF (I.LT.NRA) GO TO 10
      IF (J.LT.NCA) GO TO 10
      IF (L.EQ.0) GO TO 10
    5 MPHEAD(1) = L
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(L)
      CALL YOUTI  (NUT1,MPHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,L)
      CALL YOUT   (NUT1,V,1,L)
      NREC=NREC+1
      NNZA=NNZA+L
      L=0
```

```
   10 CONTINUE
   20 CONTINUE
C
      REWIND NUTA
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NREC
      MHEAD(4) = NNZA
      MHEAD(5) = 5HORDER
      MHEAD(7) = 5HWHOLE
      CALL YOUTI  (NUTA,MHEAD,1,10)
      REWIND NUT1
C
C  TRANSFER PARTITIONS FROM NUT1 TO NUTA.
      DO 65 I=1,NREC
      CALL YINI   (NUT1,MPHEAD,1,10)
      CALL YOUTI  (NUTA,MPHEAD,1,10)
      NNZP = MPHEAD(1)
      CALL YINI   (NUT1,LV,1,NNZP)
      CALL YIN    (NUT1,V,1,NNZP)
      CALL YOUTI  (NUTA,LV,1,NNZP)
      CALL YOUT   (NUTA,V,1,NNZP)
   65 CONTINUE
      CALL YPART  (NUTA,V,LV,KV,NUT1)
      RETURN
C
  999 CALL ZZBOMB (5HYDTOS ,NERROR)
      END
```

```
      SUBROUTINE YEQUAL(NUT A,NUTB,V,LV,KV)
      DIMENSION V(1),LV(1),MH(10)
C
C   THIS SUBROUTINE COPIES A SPARSE MATRIX ON UTILITY FILE NUTA
C   TO UTILITY FILE NUTB.
C              SUBROUTINE ARGUMENTS
C   NUTA - INPUT  UTILITY FILE CONTAINING A SPARSE MATRIX
C   NUTB - INPUT  UTILITY FILE THE SPARSE MATRIX ON NUTA IS TO BE COPIED
C                 ONTO.
C   V    - INPUT  WORK SPACE.
C   LV   - INPUT  WORK SPACE.
C   KV   - INPUT  DIMENSION OF V AND LV IN THE CALLING PROGRAM.
C
C   FORMA SUBROUTINES YIN,YINI,YOUT,YOUTI AND ZZBOMB ARE CALLED.
C   CODED BY JOHN ADMIRE  *NASA*  DECEMBER 1973.
C
      REWIND NUTA
      REWIND NUTB
      CALL YINI(NUTA,MH,1,10)
      CALL YOUTI(NUTB,MH,1,10)
      NPART=MH(3)
      NNZA=MH(4)
      DO 10 L=1,NPART
      CALL YINI(NUTA,MH,1,10)
      CALL YOUTI(NUTB,MH,1,10)
      NNZP=MH(1)
      NERROR=1
      IF (NNZP .GT. KV) GO TO 999
      CALL YINI(NUTA,LV,1,NNZP)
      CALL YOUTI(NUTB,LV,1,NNZP)
      CALL YIN(NUTA,V,1,NNZP)
      CALL YOUT(NUTB,V,1,NNZP)
   10 CONTINUE
   20 RETURN
  999 CALL ZZBOMB(6HYEQUAL,NERROR)
      END
```

```
      SUBROUTINE YIN (NUT,A,NS,NE)
      DIMENSION A(1)
C
C  READ IN BINARY DATA FROM PERIPHERAL UNIT NUT INTO CORE AREA A.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1971.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUT    = LOGICAL NUMBER OF UTILITY TAPE.
C  A      = VECTOR TO BE READ.
C  NS     = START LOCATION IN VECTOR A.
C  NE     = END LOCATION IN VECTOR A.
C
C      NERROR  EXPLANATION
C  1 = START LOCATION GREATER THAN END LOCATION.
C  2 = END OF FILE ENCOUNTERED.
C  3 = READ PARITY ERROR.
C
                                                                NERROR=1
      IF (NS .LE. 0 .OR. NS .GT. NE) RETURN
                                                                NERROR=2
      READ (NUT,ERR=998,END=999) (A(I),I=NS,NE)
      RETURN
C
  998                                                           NERROR=3
  999 CALL ZZBOMB (3HYIN   ,NERROR)
      END
```

```
      SUBROUTINE YINI (NUT,IA,NS,NE)
      DIMENSION IA(1)
C
C  READ IN BINARY DATA FROM PERIPHERAL UNIT NUT INTO CORE AREA IA.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  MAY 1973.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUT    = LOGICAL NUMBER OF UTILITY TAPE.
C  IA     = VECTOR TO BE READ.
C  NS     = START LOCATION IN VECTOR IA.
C  NE     = END LOCATION IN VECTOR IA.
C
C      NERROR  EXPLANATION
C  1 = START LOCATION GREATER THAN END LOCATION.
C  2 = END OF FILE ENCOUNTERED.
C  3 = READ PARITY ERROR.
C
                                                            NERROR=1
      IF (NS .LE. 0 .OR. NS .GT. NE) RETURN
                                                            NERROR=2
      READ (NUT,ERR=998,END=999) (IA(I),I=NS,NE)
      RETURN
C
  998                                                       NERROR=3
  999 CALL ZZBOMB (4HYINI,NERROR)
      END
```

```
      SUBROUTINE YLORD (NUTA,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),IU(16),IL(16),MHEAD(10),MPHEAD(10),M2HEAD(10)
      DATA NIT,NOT/5,6/
C
C  ARRANGE ELEMENT LOCATIONS OF MATRIX A INTO INCREASING ORDER.
C  ARRANGE ELEMENTS OF MATRIX A ACCORDINGLY.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YOUT  ,YOUTI ,YPART ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  DECEMBER 1968.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = TWO LIKE LOCATION NUMBERS ENCOUNTERED.
C  2 = TWO LIKE LOCATION NUMBERS ENCOUNTERED.
C
 3001 FORMAT (1H1)
 3002 FORMAT (4(I12,3XE17.8))
C
      CALL YPART  (NUTA,V,LV,KV,NUT1)

      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NPART = MHEAD(3)
      NNZA = MHEAD(4)
      IFORD = MHEAD(5)
      IF (IFORD.EQ.5HORDER .OR. NNZA.LT.2) RETURN
      NREC=0
      NNZPT=0
      KVO4=KV/4
      NTI=NUT1
      REWIND NTI
      NDO=(NPART-1)/4+1
C
      DO 225 IJK=1,NDO
      KJI=0
      LPS=1
    5 CALL YINI   (NUTA,MPHEAD,1,10)
      NNZP = MPHEAD(1)
      NNZPT=NNZPT+NNZP
      LPE=LPS-1+NNZP
      CALL YINI   (NUTA,LV,LPS,LPE)
      CALL YIN    (NUTA,V,LPS,LPE)
      KJI=KJI+1
      LPS=LPE+1
      IF (NNZPT.EQ.NNZA) GO TO 10
      IF (KJI.LT.4) GO TO 5
C
C  SINGLETON METHOD
```

```
C
   10 M=1
      LAEM1=LPE-1
      I=1
      J=LPE
   20 IF(I.GE.J) GO TO 170
  110 K=I
      IJ=(J+I)/2
      IT=LV(IJ)
      IF(LV(I).LE.IT) GO TO 120
      LV(IJ)=LV(I)
      LV(I)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(I)
      V(I)=TG
  120 L=J
      IF(LV(J).GE.IT) GO TO 140
      LV(IJ)=LV(J)
      LV(J)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(J)
      V(J)=TG
      IF(LV(I).LE.IT) GO TO 140
      LV(IJ)=LV(I)
      LV(I)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(I)
      V(I)=TG
      GO TO 140
  130 LV(L)=LV(K)
      LV(K)=ITT
      TG=V(L)
      V(L)=V(K)
      V(K)=TG
  140 L=L-1
      IF(LV(L).GT.IT) GO TO 140
      ITT=LV(L)
  150 K=K+1
      IF(LV(K).LT.IT) GO TO 150
      IF(K.LE.L) GO TO 130
      IF(L-I.LE.J-K) GO TO 160
      IL(M)=I
      IU(M)=L
      I=K
      M=M+1
      GO TO 180
  160 IL(M)=K
      IU(M)=J
      J=L
      M=M+1
      GO TO 180
  170 M=M-1
```

```
      IF(M.EQ.0) GO TO 210
      I=IL(M)
      J=IU(M)
  180 IF(J-I.GE.11) GO TO 110
      IF (I.EQ.1) GO TO 20
      I=I-1
  190 I=I+1
      IF(I.EQ.J) GO TO 170
      IT=LV(I+1)
      IF(LV(I).LE.IT) GO TO 190
      TG=V(I+1)
      K=I
  200 LV(K+1)=LV(K)
      V(K+1)=V(K)
      K=K-1
      IF(IT.LT.LV(K)) GO TO 200
      LV(K+1)=IT
      V(K+1)=TG
      GO TO 190
C
  210                                                       NERROR=1
      DO 215 I=1,LAEM1
      IF (LV(I).EQ.LV(I+1)) GO TO 900
  215 CONTINUE
      IF (LPE.LE.KV04*3) KJI=(LPE-1)/KV04+1
      LPS=1

      IF (NPART.GT.4) GO TO 218
      NPART=(LPE-1)/KV04+1
      REWIND NUTA
      MHEAD(3) = NPART
      MHEAD(5) = 5HORDER
      MHEAD(6) = 0
      CALL YOUTI  (NUTA,MHEAD,1,10)
      DO 217 I=1,NPART
      LPPE=LPS-1+KV04
      IF (LPPE.GT.LPE) LPPE=LPE
      NNZ=LPPE-LPS+1
      MPHEAD(1) = NNZ
      MPHEAD(2) = LV(LPS)
      MPHEAD(3) = LV(LPPE)
      CALL YOUTI  (NUTA,MPHEAD,1,10)
      CALL YOUTI  (NUTA,LV,LPS,LPPE)
      CALL YOUT   (NUTA,V,LPS,LPPE)
  217 LPS=LPPE+1
      GO TO 310
  218 DO 220 J=1,KJI
      IF (LPS.GT.LPE) GO TO 225
      LPPE=LPS-1+KV04
      IF (LPPE.GT.LPE) LPPE=LPE
      NNZ=LPPE-LPS+1
      MPHEAD(1) = NNZ
      MPHEAD(2) = LV(LPS)
      MPHEAD(3) = LV(LPPE)
      CALL YOUTI  (NT1,MPHEAD,1,10)
```

```
      CALL YOUTI  (NT1,LV,LPS,LPPE)
      CALL YOUT   (NT1,V,LPS,LPPE)
      NREC=NREC+1
  220 LPS=LPPE+1
C
  225 CONTINUE
C
C  NOW THERE ARE NREC ORDERED PARTITIONS WRITTEN ON NT1
      REWIND NUTA
      MHEAD(3) = NREC
      MHEAD(5) = 5HORDER
      MHEAD(6) = 0
      CALL YOUTI  (NUTA,MHEAD,1,10)
      NT2=NUT2
C
C  MESHING OPERATION
  230 REWIND NT1
      REWIND NT2
      CALL YINI   (NT1,MPHEAD,1,10)
      NNZP1 = MPHEAD(1)
      CALL YINI   (NT1,LV,1,NNZP1)
      CALL YIN    (NT1,V,1,NNZP1)
      IF (NREC.EQ.1) GO TO 305
      LP2S=NNZP1+1
      NREC2=0
C
      DO 300 I=2,NREC
      CALL YINI   (NT1,M2HEAD,1,10)
      NNZP2 = M2HEAD(1)
      LP2E=LP2S-1+NNZP2
      CALL YINI   (NT1,LV,LP2S,LP2E)
      CALL YIN    (NT1,V,LP2S,LP2E)
      IF (LV(LP2S) .GT. LV(NNZP1)) GO TO 295
C
C  MESH TWO PARTITIONS
      I1=1
      I2=NNZP1+1
      IW=2*KV04
      I2=0
  250 IW=IW+1
                                                    NERROR=2
      IF (LV(I1)-LV(I2)) 265,909,255
  255 V(IW)=V(I2)
      LV(IW)=LV(I2)
      I2=I2+1
      IF (I2.GT.LP2E) GO TO 275
      GO TO 250
  265 V(IW)=V(I1)
      LV(IW)=LV(I1)
      I1=I1+1
      IF (I1.GT.NNZP1) GO TO 285
      GO TO 250
  275 NELTM=NNZP1-I1+1
      K=LP2E
      L=NNZP1
```

```
      DO 280 J=1,NELTM
      V(K)=V(L)
      LV(K)=LV(L)
      K=K-1
  280 L=L-1
C
  285 IF (IW.EQ.2*KV04) GO TO 295
      J1=2*KV04+1
C
      DO 290 J=J1,IW
      IZ=IZ+1
      V(IZ)=V(J)
  290 LV(IZ)=LV(J)
C
  295 NREC2=NREC2+1
      M2HEAD(2) = LV(LP2S)
      M2HEAD(3) = LV(LP2E)
      CALL YOUTI  (NT2,M2HEAD,1,10)
      CALL YOUTI  (NT2,LV,LP2S,LP2E)
      CALL YOUT   (NT2,V,LP2S,LP2E)
  300 CONTINUE
C
C  ALL NREC PARTITIONS HAVE BEEN READ FROM NT1
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(NNZP1)
      CALL YOUTI  (NUTA,MPHEAD,1,10)
      CALL YOUTI  (NUTA,LV,1,NNZP1)
      CALL YOUT   (NUTA,V,1,NNZP1)
      NREC=NREC2
      NTS=NT1
      NT1=NT2
      NT2=NTS
      GO TO 230
C
  305 CALL YOUTI  (NUTA,MPHEAD,1,10)
      CALL YOUTI  (NUTA,LV,1,NNZP1)
      CALL YOUT   (NUTA,V,1,NNZP1)
  310 CALL YPART  (NUTA,V,LV,KV,NUT1)
      RETURN
C
  900 WRITE (NOT,3001)
      WRITE (NOT,3002) LV(I)
      WRITE (NOT,3002) (LV(II),V(II),II=1,LPE)
      GO TO 999
  909 WRITE (NOT,3001)
      WRITE (NOT,3002) LV(I1)
      WRITE (NOT,3002) (LV(II),V(II),II=1,LP2E)
  999 CALL ZZEOMB (5HYLORD ,NERROR)
      END
```

```
      SUBROUTINE YMOD2A (NUTM,NUTK,NUTZ,W2,W,FREQ,
     *                   NW,NU,SHIFT,TOLZ,TOLW2,MAXIT,IFPRNT,
     *                   V,LV,A,S,KVIN,KA,NUT1,NUT2,NUT3,NUT4,NUT5,NUT6)
      DIMENSION V(1), LV(1), W2(1), W(1), FREQ(1), A(KA,1), S(KA,1)
      DIMENSION IH(10)
      DATA NIT,NOT/5,6/
C
C  CALCULATE MODE SHAPES (PHI) AND NATURAL FREQUENCIES OF
C  (-W2*(MASS) + (STIF))*(PHI) = (0) USING ITERATIVE RAYLEIGH-RITZ
C  METHOD OF DR. JOHN ADMIRE. COMPOSITE STRUCTURE TECHNIQUE,
C  NON-SWEEPING VERSION, SPARSE PROGRAMMING LOGIC.
C  THE MASS (NUTM) MATRIX SHOULD BE REAL, SYMMETRIC.
C  THE STIF (NUTK) MATRIX SHOULD BE REAL, SYMMETRIC.
C  THE FIRST ELEMENT OF EACH MODE SHAPE IS MADE POSITIVE.
C  MODES ARE NORMALIZED SUCH THAT (PHI)T*(MASS)*(PHI) = 1.
C  CALLS FORMA SUBROUTINES ....
C     BTABA2,EIGNIA,INV4  ,MODEIX,NAME  ,PAGEHD,TIMCHK,WRITE ,WRITIM,
C     XLORD ,YAABB ,YBSL3A,YDCM3A,YDTOS ,YIN   ,YINI  ,YLORD ,YMULT1,
C     YMULT2,YMULT4,YNOZER,YOUT  ,YOUT1 ,YPART ,YRV1  ,YSTOD ,YSYMLH,
C     YSYMUH,YTRANS,YWRITE,ZZBOMB.
C  DEVELOPED BY RL WOHLEN, WA BENFIELD, RA PHILIPPUS.  MARCH 1972.
C  LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS
C  NUTM   = INPUT  LOGICAL NUMBER OF UTILITY TAPE OF MASS MATRIX.
C  NUTK   = INPUT  LOGICAL NUMBER OF UTILITY TAPE OF STIF MATRIX.
C  NUTZ   = INPUT  LOGICAL NUMBER OF UTILITY TAPE OF CALCULATED MODES.
C                  MAY BE USED TO INPUT INITIAL RAYLEIGH VECTORS.
C  W2     = OUTPUT VECTOR OF EIGENVALUES (OMEGA SQUARED). SIZE(NU).
C  W      = OUTPUT VECTOR OF CIRCULAR FREQUENCY (OMEGA). SIZE(NU).
C  FREQ   = OUTPUT VECTOR OF FREQUENCY (OMEGA/2PI). SIZE (NU).
C  NW     = INPUT  NUMBER OF MODES WANTED. ITERATIONS STOP WHEN NW
C                  CONSECUTIVE MODES AROUND SHIFT POINT CONVERGE.
C                  HOWEVER, ALL NU MODES AND FREQUENCIES ARE OUTPUT
C                  FOR LATER SELECTION.
C  NU     = INPUT  NUMBER OF MODES USED. MUST BE .GE. NW.
C  SHIFT  = INPUT  SHIFT IN (STIF)-SHIFT(MASS).
C                  CONVERGENCE WILL BE ABOUT THIS VALUE.
C  TOLZ   = INPUT  TOLERANCE ON ZERO W2.
C  TOLW2  = INPUT  CONVERGENCE TOLERANCE ON NON-ZERO W2.
C  MAXIT  = INPUT  MAXIMUM NUMBER OF ITERATIONS. A GOOD VALUE IS 20.
C  IFPRNT = INPUT  = 1 PRINT INTERMEDIATE RESULTS.
C  V      = INPUT  VECTOR WORK SPACE. DIMENSION GREATER THAN   * (N-1).
C  LV     = INPUT  VECTOR WORK SPACE. DIMENSION GREATER THAN 6 * (N-1).
C  A      = INPUT  MATRIX WORK SPACE. EQUIVALENCE TO LV AT KV/3+1.
C  S      = INPUT  MATRIX WORK SPACE. EQUIVALENCE TO V AT KV/3+1.
C  KVIN   = INPUT  DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  KA     = INPUT  ROW DIMENSION OF A,S IN CALLING PROGRAM.
C  NUT1   = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2   = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C  NUT3   = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C  NUT4   = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C  NUT5   = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C  NUT6   = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C
```

```
C      NERROR  EXPLANATION
C   1 = NUMBER OF MODES USED IS LESS THAN NUMBER WANTED.
C   2 = NUMBER OF MODES USED EXCEEDS DIMENSION SIZE.
C   3 = SIZE EXCEEDS KV.
C
 2010 FORMAT (/// 54X,19H(SUBROUTINE YMOD2A)
     *         /// 52X, 19HNO. MODES WANTED = I3,
     *         /// 52X, 19HNO. MODES USED   = I3,
     *         /// 52X,  8HSHIFT = E15.8,
     *         /// 52X, 11HTOL ZERO = E15.8,
     *         /// 52X, 11HTOL W2   = E15.8,
     *         /// 52X,  8HMAXIT = I3)
 2020 FORMAT (// 7H ITER = I3, 5H  W2. / (10X 10E11.3) )
 2030 FORMAT (/ 12X 49HCONVERGENCE VALUES. EITHER W2(ITER) IF (W2(ITER)
     *                46H.LT. ZERO TOLERANCE) OR (W2(ITER)-W2(ITER-1))/
     *                 9HW2(ITER). / (10X 10E11.3) )
 6000 FORMAT (20H NO. OF MODES BELOW ,E10.3,3H = ,I5)
 6020 FORMAT (17X 16HITERATION TIME = F10.3,7H CP SEC F15.3,7H PP SEC)
C
                                                    CALL TIMCHK (6HTBEGIN)
      KV = 4*(KVIN/4)
      CALL PAGEHD
      WRITE (NOT,2010) NW,NU,SHIFT,TOLZ,TOLW2,MAXIT
                                                                NERROR=1
      IF (NU .LT. NW) GO TO 999
                                                                NERROR=2
      IF (NU.GT.KA) GO TO 999
      REWIND NUTM
      CALL YINI   (NUTM,IH,1,10)
      N = IH(1)
                                                                NERROR=3
      IF (KV.LT.6*(N-1)) GO TO 999
      IF (IFPRNT.EQ.1) CALL YWRITE (NUTM,4HMASS,V,LV,KV)
      IF (IFPRNT.EQ.1) CALL YWRITE (NUTK,4HSTIF,V,LV,KV)
C
C  CALCULATE (K-HAT) = (STIF) - SHIFT*(MASS).
                                                    CALL TIMCHK (6HYAABB )
      CALL YAABB  (1.,NUTK,-SHIFT,NUTM,NUT3, V,LV,KV,NUT4,NUT5)
                                                    CALL TIMCHK (6HYAABB )
      IF (IFPRNT.EQ.1) CALL YWRITE (NUT3,5HK-HAT,V,LV,KV)
C  DECOMPOSE (K-HAT) INTO (U**T)*(-D-)*(U).
                                                    CALL TIMCHK (6HYDCM3A)
      CALL YDCM3A (NUT3,NUT1,NUT2, V,LV,KV,NUT4,NUT5)
                                                    CALL TIMCHK (6HYDCM3A)
C  CALCULATE NUMBER OF NEGATIVE ROOTS (FREQUENCIES) BELOW SHIFT POINT.
      REWIND NUT2
      CALL YIN    (NUT2,V,1,N)
      IF (IFPRNT .EQ. 1) CALL WRITE (V,1,N,1HD,1)
      NKOUNT = 0
      DO 15 MKOUNT=1,N
      IF (V(MKOUNT).LT.0.) NKOUNT=NKOUNT+1
   15 CONTINUE
      WRITE (NOT,6000) SHIFT,NKOUNT
C  GENERATE INITIAL RAYLEIGH VECTORS.
                                                    CALL TIMCHK (6HYRV1  )
```

```
      CALL YRVI    (NUTZ,N,NU,V,LV,KV,NUT4,NUT5,NUT6)                           Z
                                                    CALL TIMCHK (6HYRVI   )
      IF (IFPRNT.EQ.1) CALL YWRITE (NUTZ,6HZ-IN   ,V,LV,KV)
C
                                                    CALL TIMCHK (6HYMULT1)
      CALL YMULT1 (NUT3,NUTZ,NUT4, V,LV,KV,NUT6)                                4
                                                    CALL TIMCHK (6HYMULT1)
      IF (IFPRNT.EQ.1) CALL YWRITE (NUT4,2HKZ,V,LV,KV)
C
C   BEGIN ITERATION LOOP.
      CALL PAGEHD
      ITER = 0
   20 ITER = ITER+1
                                                    CALL TIMCHK (6HYMULT2)
      CALL YMULT2 (NUTZ,NUT4,NUT3,A,S,V,LV,KV,KA,NUT6)                          3
                                                    CALL TIMCHK (6HYMULT2)
                                                    CALL TIMCHK (6HYMULT1)
      CALL YMULT1 (NUTM,NUTZ,NUT5, V,LV,KV.NUT6)                                5
                                                    CALL TIMCHK (6HYMULT1)
      IF (IFPRNT.EQ.1) CALL YWRITE (NUT5,2HMZ,V,LV,KV)
                                                    CALL TIMCHK (6HYMULT2)
      CALL YMULT2 (NUTZ,NUT5,NUT4,A,S,V,LV,KV,KA,NUT6)                          4
                                                    CALL TIMCHK (6HYMULT2)
                                                    CALL TIMCHK (6HYSTOD )
      CALL YSTOD  (NUT4,A,NR,NC,KA,KA,V,LV,KV,NUT6)                             A
      CALL YSTOD  (NUT3,S,NR,NC,KA,KA,V,LV,KV,NUT6)                             S
                                                    CALL TIMCHK (6HYSTOD )
      IF (IFPRNT .EQ. 1) CALL WRITE (A,NU,NU,4HMBAR,KA)
      IF (IFPRNT .EQ. 1) CALL WRITE (S,NU,NU,4HKBAR,KA)
                                                    CALL TIMCHK (6HMODEIX)
      CALL MODEIX (A,S,W2,NU,TOLW2,KA)                                          A
                                                    CALL TIMCHK (6HMODEIX)
      IF (IFPRNT .EQ. 1) CALL WRITE (A, NU,NU, 2HY*, KA)
                                                    CALL TIMCHK (6HCVTEST)
C   UNSHIFT W2.
      DO 25 J=1,NU
      V(J) = ABS(W2(J))
      LV(J) = J
   25 W2(J) = W2(J) + SHIFT
      WRITE (NOT,2020) ITER,(W2(J),J=1,NU)
      IF (ITER .EQ. 1) GO TO 60
C   STORE CONVERGENCE VALUES OF W2 IN W. (LAST ITER OF W2 IS IN FREQ).
      DO 28 J=1,NU
      W(J) = W2(J)
      IF (ABS(W2(J)) .GT. TOLZ) W(J) = (W2(J)-FREQ(J))/W2(J)
   28 CONTINUE
      WRITE (NOT,2030) (W(J),J=1,NU)
      IF (ITER.GE.MAXIT) GO TO 70
C   FIND START AND END LOCATION OF (W2-SHIFT) OF BAND WIDTH NW ABOUT SHIFT
      DO 37 J=1,NW
      IMIN = J
      VMIN = V(J)
      LMIN = LV(J)
      DO 36 I=J,NU
      IF (VMIN .LE. V(I)) GO TO 36
```

```
      IMIN = I
      VMIN = V(I)
      LMIN = LV(I)
   36 CONTINUE
      V(IMIN) = V(J)
      LV(IMIN) = LV(J)
      V(J) = VMIN
   37 LV(J) = LMIN
      JS = LV(1)
      JE = LV(1)
      DO 38 J=1,NW
      IF (LV(J) .LT. JS) JS=LV(J)
      IF (LV(J) .GT. JE) JE=LV(J)
   38 CONTINUE
C  TEST W2 FOR CONVERGENCE OF NW CONSECUTIVE MODES ABOUT SHIFT POINT.
      DO 45 J=JS,JE
      IF (ABS(W2(J)) .LT. TOLZ) GO TO 45
      IF (ABS(W(J)) .GT. TOLW2) GO TO 50
   45 CONTINUE
      GO TO 70
   50 CONTINUE
C  STORE LAST ITERATION VALUE OF W2 IN FREQ.
   60 DO 62 J=1,NU
   62 FREQ(J) = W2(J)
                                               CALL TIMCHK (6HCVTEST)
C  IMPROVE RAYLEIGH VECTORS.
                                               CALL TIMCHK (6HYMULT4)
      CALL YMULT4 (NUT5,A,NUT4,S,V,LV,KV,KA,NUT6)                        4
                                               CALL TIMCHK (6HYMULT4)
      IF (IFPRNT.EQ.1) CALL YWRITE (NUT4,IHG,V,LV,KV)
                                               CALL TIMCHK (6HYBSL3A)
      CALL YBSL3A (NUT1,NUT2,NUT4,NUTZ, V,LV,KV,NUT5,NUT6)                Z
                                               CALL TIMCHK (6HYBSL3A)
      IF (IFPRNT.EQ.1) CALL YWRITE (NUTZ,IHZ,V,LV,KV)
      GO TO 20
C  END ITERATION LOOP.
C
C  GET W,FREQ,MODE SHAPES. MAKE FIRST ELEMENT OF EACH MODE POSITIVE.
C  SAVE ALL MODES AND FREQUENCIES USED (NU) FOR LATER SELECTION.
   70 DO 72 I=1,NU
      W(I)   = SQRT (ABS(W2(I)))
   72 FREQ(I)= .15915494 * W(I)
                                               CALL TIMCHK (6HYMULT4)
      CALL YMULT4 (NUTZ,A,NUT1,S,V,LV,KV,KA,NUT6)                        1
                                               CALL TIMCHK (6HYMULT4)
                                               CALL TIMCHK (6HMDIPOS)
      REWIND NUT1
      REWIND NUTZ
      IVSM1 = KV-N
      IVS = IVSM1+1
      DO 73 I=IVS,KV
   73 V(I) = 1.
      CALL YIN1   (NUT1,IH,1,10)
      CALL YOUT1  (NUTZ,IH,1,10)
      NGZ = IH(3)
```

```
      DO 75 IGZ=1,NGZ
      CALL YINI   (NUT1,IH,1,IC)
      NELGZ = IH(1)
      CALL YINI   (NUT1,LV,1,NELGZ)
      CALL YIN    (NUT1,V,1,NELGZ)
      DO 74 I=1,NELGZ
      IZ = LV(I)/100000
      JZ = LV(I)-100000*IZ
      IF (IZ.EQ.1 .AND. V(I).LT.0.) V(IVSM1+JZ)=-1.
   74 V(I) = V(I)*V(IVSM1+JZ)
      CALL YOUTI  (NUTZ,IH,1,IC)
      CALL YOUTI  (NUTZ,LV,1,NELGZ)
   75 CALL YOUT   (NUTZ,V,1,NELGZ)                                  Z
                                          CALL TIMCHK (6HMD1POS)
                                          CALL TIMCHK (6HTPRINT)
      RETURN
C
  999 CALL ZZBOMB (6HYMODE2,NERROR)
      END
```

```
      SUBROUTINE YMULT (NUTA,NUTB,NUTZ,V,LV,KV,NUT1)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NCT/5,6/
C
C  SPARSE MATRIX MULTIPLICATION (A * B = Z).
C  KV/4 MUST BE EQUAL TO OR GREATER THAN,
C       (1) NUMBER OF COLUMNS OF MATRIX A
C   AND (2) NUMBER OF COLUMNS OF MATRIX B.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YNOZER,YOUT  ,YOUTI ,
C                              YPART ,YSYMLH,YSYMUH,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  APRIL 1969.
C  LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTB  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX B IS STORED.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C  2 = INCOMPATIBLE MATRICES.
C
      NPARTZ=0
      NNZZ=0
      NREC=0
      CALL YLORD  (NUTA,V,LV,KV,NUT1,NUTZ)
      CALL YLORD  (NUTB,V,LV,KV,NUT1,NUTZ)
C  GET (A) HEADER INFORMATION.
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NNZA = MHEAD(4)
      ISHAP = MHEAD(7)
      NPCTA=100*NNZA/NRA/NCA
      IXYZ3=ISHAP
      IF (ISHAP.EQ.5HWHOLE .OR. ISHAP.EQ.4HDIAG) GO TO 5
      IF (ISHAP.EQ.5HLOWER) CALL YSYMUH (NUTA,V,LV,KV,NUT1,NUTZ)
      IF (ISHAP.EQ.5HUPPER) CALL YSYMLH (NUTA,V,LV,KV,NUT1,NUTZ)
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZA = MHEAD(4)
      NPCTA=100*NNZA/NRA/NCA
    5 NPARTA = MHEAD(3)
C  GET (B) HEADER INFORMATION.
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
      NNZB = MHEAD(4)
      ISHAP = MHEAD(7)
```

```
      NPCTB=100*NNZB/NRB/NCB
      IXYZ4=ISHAP
      IF (ISHAP.EQ.5HWHOLE .OR. ISHAP.EQ.4HDIAG) GO TO 10
      IF (ISHAP.EQ.5HLOWER) CALL YSYMUH (NUTB,V,LV,KV,NUT1,NUTZ)
      IF (ISHAP.EQ.5HUPPER) CALL YSYMLH (NUTB,V,LV,KV,NUT1,NUTZ)
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NNZB = MHEAD(4)
      NPCTB=100*NNZB/NRB/NCB
   10 IF (NNZA.EQ.0 .OR. NNZB.EQ.0) GO TO 70
      NPARTB = MHEAD(3)
                                                                 NERROR=1
      IF (NCA.GT.KV/4 .OR. NCB.GT.KV/4) GO TO 999
                                                                 NERROR=2
      IF (NRB.NE.NCA) GO TO 999
C
      IZ=0
      LPBS=KV/4+1
      LPBE=LPBS-1
      LCS=KV/2+1
      LCE=LCS-1+NCB
      LCCS=LCE+1
      LCCE=LCE
      NNZ=KV-LCCS+1
      REWIND NUT1
C
      DO 15 I=LCS,LCE
   15 V(I)=0.
C
C  LOOP ON (A) PARTIONS.
      DO 55 I=1,NPARTA
C  GET (A) PARTITION INFORMATION.
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
      CALL YINI   (NUTA,LV,1,NNZPA)
      CALL YIN    (NUTA,V,1,NNZPA)
      K=LPBS
      ITRBL=0
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,10,10)
      NREAD=0
C
      DO 50 INA=1,NNZPA
      IA=LV(INA)/100000
      JA=LV(INA)-100000*IA
      IF (IA.EQ.IZ .AND. ITRBL.EQ.1) GO TO 50
      ITRBL=0
      IF (IA.EQ.IZ) GO TO 30
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,10,10)
      NREAD=0
C
      DO 25 INC=LCS,LCE
```

```
      IF (V(INC).EQ.0.) GO TO 25
      LCCE=LCCE+1
      V(LCCE)=V(INC)
      LV(LCCE)=IZZ+INC-KV/2
      V(INC)=0.
      IF (LCCE.LT.KV) GO TO 25
      CALL YOUTI  (NUT1,LV,LCCS,LCCE)
      CALL YOUT   (NUT1,V,LCCS,LCCE)
      NREC=NREC+1
      NNZZ=NNZZ+NNZ
      LCCE=LCE
   25 CONTINUE
C
      IZ=IA
      IZZ=100000*IZ
      K=LPBS
   30 IF (K.LE.LPBE .AND. NREAD.GT.0) GO TO 40
   35 IF (NREAD.EQ.NPARTB) ITRBL=1
      IF (ITRBL.EQ.1) GO TO 50
      CALL YINI   (NUTB,MHEAD,1,10)
      NNZPB = MHEAD(1)
      LFELPB = MHEAD(2)
      LLELPB = MHEAD(3)
      LPBE=LPBS-1+NNZPB
      CALL YINI   (NUTB,LV,LPBS,LPBE)
      CALL YIN    (NUTB,V,LPBS,LPBE)
      NREAD=NREAD+1
      K=LPBS
C
   40 DO 45 INB=K,LPBE
      K=INB
      IB=LV(INB)/100000
      IF (IB.GT.JA) GO TO 50
      IF (IB.LT.JA) GO TO 45
      JBZ=LV(INB)-100000*IB
      INZ=KV/2+JBZ
      V(INZ)=V(INZ)+V(INA)*V(INB)
   45 CONTINUE
C
      GO TO 35
   50 CONTINUE
   55 CONTINUE
C
      DO 60 I=LCS,LCE
      IF (V(I).EQ.0.) GO TO 60
      LCCE=LCCE+1
      V(LCCE)=V(I)
      LV(LCCE)=IZZ+I-KV/2
      IF (LCCE.LT.KV) GO TO 60
      CALL YOUTI  (NUT1,LV,LCCS,LCCE)
      CALL YOUT   (NUT1,V,LCCS,LCCE)
      NREC=NREC+1
      LCCE=LCE
      NNZZ=NNZZ+NNZ
   60 CONTINUE
```

```
C
      IF (LCCE.EQ.LCE) GO TO 70
      NNZ=LCCE-LCCS+1
      NNZZ=NNZZ+NNZ
      NREC=NREC+1
      CALL YOUTI  (NUT1,LV,LCCS,LCCE)
      CALL YOUT   (NUT1,V,LCCS,LCCE)
   70 REWIND NUTZ
      MHEAD(1) = NRA
      MHEAD(2) = NCB
      MHEAD(3) = NREC
      MHEAD(4) = NNZZ
      MHEAD(5) = 5HORDER
      MHEAD(6) = 0
      MHEAD(7) = 5HWHOLE
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      IF (NNZZ.GT.0) GO TO 75
      DO 72 I=1,10
   72 MHEAD(I) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,2)
      CALL YOUT   (NUTZ,    V,1,2)
      RETURN
   75 LZE=KV
      REWIND NUT1
C
      DO 100 I=1,NREC
      IF (I.EQ.NREC) LZE=LCCS-1+NNZ
      NNZP = LZE-LCCS+1
      CALL YINI   (NUT1,LV,LCCS,LZE)
      CALL YIN    (NUT1,V,LCCS,LZE)
      MHEAD(1) = NNZP
      MHEAD(2) = LV(LCCS)
      MHEAD(3) = LV(LZE)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,LCCS,LZE)
  100 CALL YOUT   (NUTZ,V,LCCS,LZE)
C
      CALL YPART  (NUTZ,V,LV,KV,NUT1)
      RETURN
C
  999 CALL ZZBOMB (5HYMULT ,NERROR)
      END
```

```
      SUBROUTINE YMULTI (NUTA,NUTB,NUTZ,V,LV,KV,NUTI)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C   SPECIAL SPARSE MATRIX MULTIPLICATION (A * B = Z).
C   B AND Z ARE DENSE MATRICES.
C   KV/4 MUST BE EQUAL TO OR GREATER THAN,
C        (1) NUMBER OF COLUMNS OF MATRIX A
C    AND (2) NUMBER OF COLUMNS OF MATRIX B.
C   CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YNOZER,YOUT  ,YOUTI ,
C                           YPART ,YSYMLH,YSYMUH,ZZBOMB.
C   DEVELOPED BY R A PHILIPPUS.  AUGUST 1972.
C   LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C   NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C   NUTB  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX B IS STORED.
C   NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C   V     = VECTOR WORK SPACE.
C   LV    = VECTOR WORK SPACE.
C   KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C   NUTI  = LOGICAL NUMBER OF UTILITY TAPE.
C
C      NERROR  EXPLANATION
C   1 = SIZE LIMITATION EXCEEDED.
C   2 = INCOMPATIBLE MATRICES.
C
      KV04 = KV/4
      CALL YLORD  (NUTA,V,LV,KV,NUTI,NUTZ)
      CALL YLORD  (NUTB,V,LV,KV,NUTI,NUTZ)
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NNZA = MHEAD(4)
      ISHAP = MHEAD(7)
      IF (ISHAP.EQ.5HWHOLE .OR. ISHAP.EQ.4HDIAG) GO TO 5
      IF (ISHAP.EQ.5HLOWER) CALL YSYMUH (NUTA,V,LV,KV,NUTI,NUTZ)
      IF (ISHAP.EQ.5HUPPER) CALL YSYMLH (NUTA,V,LV,KV,NUTI,NUTZ)
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZA = MHEAD(4)
    5 NPARTA = MHEAD(3)
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
      NPARTB = MHEAD(3)
      NNZB = MHEAD(4)
                                                           NERROR=1
      IF (NCA.GT.KV/4 .OR. NCB.GT.KV/4) GO TO 999
                                                           NERROR=2
      IF (NRB.NE.NCA) GO TO 999
C
      IZ=0
```

```
      LBS=KVO4+1
      LZS = 2*KVO4+1
      NNZZ = NNZB
      NPARTZ = NPARTB
      NNZPZ = KVO4/NCB*NCB
      MHEAD(1) = NRA
      MHEAD(2) = NCB
      MHEAD(3) = NPARTZ
      MHEAD(4) = NNZZ
      MHEAD(5) = 5HORDER
      MHEAD(6) = 4*KVO4
      MHEAD(7) = 5HWHOLE
      REWIND NUTZ
      CALL YOUTI  (NUTZ,MHEAD,1,10)
C
C  READ A ONE TIME, EACH PARTITION AS REQUIRED.
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
      NNZARD = NNZPA
      CALL YINI   (NUTA,LV,1,NNZPA)
      CALL YIN    (NUTA, V,1,NNZPA)
      INA = 0
      NFRPZ = 1
C
      DO 55 IPARTZ=1,NPARTZ
      IF (IPARTZ.EQ.NPARTZ) NNZPZ=NNZZ-(IPARTZ-1)*NNZPZ
      NRPZ = NNZPZ/NCB
      NLRPZ = NFRPZ-1+NRPZ
      LZE = LZS-1+NNZPZ
      INZ = LZS-1
      DO 20 I=NFRPZ,NLRPZ
      DO 20 J=1,NCB
      INZ = INZ+1
      V(INZ) = 0.
   20 LV(INZ) = 100000*I+J
   21 REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
C
C  READ ALL OF B FOR EACH PARTITION OF A OR Z.
      DO 50 IPARTB=1,NPARTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NNZPB = MHEAD(1)
      LFELPB = MHEAD(2)
      LLELPB = MHEAD(3)
      LBE = LBS-1+NNZPB
      CALL YINI   (NUTB,LV,LBS,LBE)
      CALL YIN    (NUTB, V,LBS,LBE)
      INA = 0
      NFRPB = LFELPB/100000
      NLRPB = LLELPB/100000

   25 IF (INA.LT.NNZPA) GO TO 30
      IF (IPARTB.LT.NPARTB) GO TO 50
```

```
      IF (NNZARD.EQ.NNZA) GO TO 50
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      LFELPA = MHEAD(2)
      LLELPA = MHEAD(3)
      CALL YINI   (NUTA,LV,1,NNZPA)
      CALL YIN    (NUTA, V,1,NNZPA)
      NNZARD = NNZARD+NNZPA
      INA = 0
      GO TO 21
C
   30 INA = INA+1
      IA = LV(INA)/100000
      IF (IA.LE.NLRPZ) GO TO 35
      GO TO 50
C
   35 IF (IA.LT.NFRPZ) GO TO 25
      JA = LV(INA)-100000*IA
      IF (JA.LT.NFRPB) GO TO 25
      IF (JA.GT.NLRPB) GO TO 25
      LZ = (IA-NFRPZ)*NCB+LZS-1
      LB = (JA-NFRPB)*NCB+LBS-1
      IF (LB+NCB.GT.LBE) GO TO 50
C
      DO 40 JZ=1,NCB
      LZ = LZ+1
      LB = LB+1
   40 V(LZ) = V(LZ)+V(INA)*V(LB)
      GO TO 25
C
   50 CONTINUE
      NFRPZ = NLRPZ+1
      MHEAD(1) = NNZPZ
      MHEAD(2) = LV(LZS)
      MHEAD(3) = LV(LZE)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,LZS,LZE)
   55 CALL YOUT   (NUTZ, V,LZS,LZE)
      CALL YPART  (NUTZ,V,LV,KV,NUT1)
      RETURN
C
  999 CALL ZZBOMB (6HYMULTI,NERROR)
      END
```

```
      SUBROUTINE YMULT2 (NUTA,NUTB,NUTZ,A,B,V,LV,KV,KR,NUT1)
      DIMENSION V(1), LV(1), A(KR,1), B(KR,1), MHEAD(10)
      COMMON / LWRKV1 / W(500)
      DATA NIT,NOT/5,6/
C
C  SPECIAL SPARSE MATRIX MULTIPLICATION (A**T)*(B)=(Z).
C  A,B,Z ARE DENSE MATRICES.
C  Z IS SYMMETRIC.
C  KV/4 MUST BE EQUAL TO OR GREATER THAN,
C       (1) NUMBER OF COLUMNS OF MATRIX A
C   AND (2) NUMBER OF COLUMNS OF MATRIX B.
C  CALLS FORMA SUBROUTINES YDTOS ,YIN   ,YINI  ,YLORD ,YNOZER,YOUT  ,
C                          YOUT1 ,YPART ,YSYMLH,YSYMUH,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  AUGUST 1972.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTB  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX B IS STORED.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  A     = MATRIX WORK SPACE.
C  B     = MATRIX WORK SPACE.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  KR    = ROW DIMENSION OF A,B IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C  2 = INCOMPATIBLE MATRICES.
C  3 = INCOMPATIBLE MATRICES.
C  4 = INCOMPATIBLE MATRICES.
C  5 = INCOMPATIBLE MATRICES.
C  6 = MORE THAN 500 COLUMNS IN MATRIX B.
C
      CALL YPART  (NUTA,V,LV,KV,NUT1)
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NNZA = MHEAD(4)
      NPARTA = MHEAD(3)
      REWIND NUTB
      CALL YINI   (NUTB,MHEAD,1,10)
      NRB = MHEAD(1)
      NCB = MHEAD(2)
      NNZB = MHEAD(4)
      NPARTB = MHEAD(3)
                                                            NERROR=1
      IF (NCA.GT.KV/4 .OR. NCB.GT.KV/4) GO TO 999
                                                            NERROR=2
      IF (NRB.NE.NRA) GO TO 999
C
                                                            NERROR=3
```

```
      IF (NPARTA.NE.NPARTB) GO TO 999
                                                                  NERROR=4
      IF (NNZA.NE.NNZB) GO TO 999
      LZS = KV/2+1
      LZE = LZS-1+NCA*NCA
      REWIND NUT1
      DO 30 IPART=1,NPARTA
      CALL YINI    (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      IRF = MHEAD(2)/100000
      IRL = MHEAD(3)/100000
      NR = IRL-IRF+1
      CALL YIN     (NUTA,V,1,1)
      CALL YIN     (NUTA,V,1,NNZPA)
      CALL YINI    (NUTB,MHEAD,1,10)
                                                                  NERROR=5
      IF (MHEAD(1).NE.NNZPA) GO TO 999
      LPBS = NNZPA+1
      LPBE = NNZPA+MHEAD(1)
      CALL YIN     (NUTB,V,LPBS,LPBS)
      CALL YIN     (NUTB,V,LPBS,LPBE)
C
C  STATEMENTS FROM ATXBB2
                                                                  NERROR=6
      IF (NCA.GT.500) GO TO 999
      DO 20 J=1,NCA
      DO 10 I=1,J
      W(I) = 0.
      DO 10 K=1,NR
      INA = (K-1)*NCA+I
      INB = NNZPA+(K-1)*NCA+J
   10 W(I) = W(I)+V(INA)*V(INB)
      DO 20 I=1,J
      A(I,J) = W(I)
   20 A(J,I) = W(I)
   30 CALL YOUT    (NUT1,A,1, KR*NCA)
      IF (NPARTA.LE.1) GO TO 40
      REWIND NUT1
      DO 35 IPART=2,NPARTA
      CALL YIN     (NUT1,B,1, KR*NCA)
      DO 35 I=1,NCA
      DO 35 J=1,NCA
   35 A(I,J) = A(I,J)+B(I,J)
   40 CALL YDTOS   (A,NUTZ,NCA,NCA,KR,KR,V,LV,KV,NUT1)
      RETURN
C
  999 CALL ZZBOMB (6HYMULT2,NERROR)
      END
```

```
      SUBROUTINE YMULT4 (NUTA,B,NUTZ,A,V,LV,KV,KR,NUT1)
      DIMENSION V(1), LV(1), A(KR,1), B(KR,1), MHEAD(10)
      COMMON / LWRKV1 / W(500)
      DATA NIT,NOT/5,6/
C
C  SPECIAL SPARSE MATRIX MULTIPLICATION (A * B = Z).
C  A,B,Z ARE DENSE MATRICES.
C  KV/4 MUST BE EQUAL TO OR GREATER THAN NUMBER OF COLUMNS OF MATRIX A.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                              ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  AUGUST 1972.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  B     = MATRIX. SIZE (NCA,NCA).
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  A     = MATRIX WORK SPACE.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  KR    = ROW DIMENSION OF A,B IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C
      REWIND NUTA
      CALL YINI   (NUT, MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NNZA = MHEAD(4)
      NPARTA = MHEAD(3)
      NRB = NCA
      NCB = NCA
                                                        NERROR=1
      IF (NCA.GT.KV/4) GO TO 999
C
      REWIND NUTZ
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      DO 50 IPART=1,NPARTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZPA = MHEAD(1)
      IRF = MHEAD(2)/100000
      IRL = MHEAD(3)/100000
      NR = IRL-IRF+1
      CALL YINI   (NUTA,LV,1,NNZPA)
      CALL YIN    (NUTA,V,1,NNZPA)
      DO 40 I=1,NR
      DO 20 K=1,NCA
      INA = (I-1)*NCA+K
   20 W(K) = V(INA)
      DO 40 J=1,NCA
      S = 0.
      DO 30 K=1,NCA
```

```
   30 S = S+W(K)*B(K,J)
      INZ = (I-1)*NCA+J
   40 V(INZ) = S
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CAL' YOUTI  (NUTZ,LV,1,NNZPA)
   50 CA.. YOUT   (NUTZ, V,1,NNZPA)
      RETURN
C
  999 CALL ZZBOMB (6HYMULT4,NERROR)
      END
```

```
      SUBROUTINE YNOZER (NUTAZ,V,LV,KV,NUT1)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  REMOVE ZERO ELEMENTS FROM SPARSE MATRIX A TO GET SPARSE MATRIX Z.
C  MATRIX A IS REPLACED BY MATRIX Z ON NUTAZ.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YOUT  ,YOUTI ,YPART ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  OCTOBER 1968.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTAZ = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRICES A,Z ARE
C             STORED. MATRIX A IS REPLACED BY MATRIX Z.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C
      REWIND NUTAZ
      REWIND NUT1
      NREC=0
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPART = MHEAD(3)
      NNZA = MHEAD(4)
      MCKORD = MHEAD(5)
      MASHAP = MHEAD(7)
      IF (NNZA.EQ.0) GO TO 25
      NNZCK=NNZA
      NNZA=0
C
      DO 10 I=1,NPART
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NNZP = MHEAD(1)
                                                            NERROR=1
      IF (NNZP.GT.KV) GO TO 999
      IF (NNZP.GT.0) GO TO 5
      CALL YINI   (NUTAZ,MHEAD,10,10)
      CALL YINI   (NUTAZ,MHEAD,10,10)
      MHEAD(10) = 0
      GO TO 10
    5 CALL YINI   (NUTAZ,LV,1,NNZP)
      CALL YIN    (NUTAZ,V,1,NNZP)
      NNZZ=0
C
      DO 8 J=1,NNZP
      IF (V(J).EQ.0.) GO TO 8
      NNZZ=NNZZ+1
      V(NNZZ)=V(J)
      LV(NNZZ)=LV(J)
    8 CONTINUE
```

```
C
      IF (NNZZ.EQ.0) GO TO 10
      NNZA=NNZA+NNZZ
      NREC=NREC+1
      MHEAD(1) = NNZZ
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZZ)
      CALL YOUTI  (NUT1,MHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,NNZZ)
      CALL YOUT   (NUT1,V,1,NNZZ)
   10 CONTINUE
C
      IF (NNZA.EQ.NNZCK) GO TO 25
      REWIND NUTAZ
      REWIND NUT1
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NREC
      MHEAD(4) = NNZA
      MHEAD(5) = MCKORD
      MHEAD(7) = MASHAP
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      IF (NREC.GT.0) GO TO 15
      MHEAD(1) = 0
      MHEAD(2) = 0
      MHEAD(3) = 0
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      CALL YOUTI  (NUTAZ,MHEAD,1,2)
      CALL YOUTI  (NUTAZ,MHEAD,1,2)
      RETURN
C
   15 DO 20 I=1,NREC
      CALL YINI   (NUT1,MHEAD,1,10)
      CALL YINI   (NUT1,LV,1,MHEAD(1))
      CALL YIN    (NUT1,V,1,MHEAD(1))
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      CALL YOUTI  (NUTAZ,LV,1,MHEAD(1))
   20 CALL YOUT   (NUTAZ,V,1,MHEAD(1))
C
   25 CALL YPART  (NUTAZ,V,LV,KV,NUT1)
      RETURN
C
  999 CALL ZZBOMB (6HYNOZER,NERROR)
      END
```

```
      SUBROUTINE YOUT (NUT,A,NS,NE)
      DIMENSION A(1)
C
C  WRITE OUT BINARY DATA FROM CORE AREA A ONTO PERIPHERAL UNIT NUT.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1971.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C        SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUT    = LOGICAL NUMBER OF UTILITY TAPE.
C  A      = VECTOR TO BE WRITTEN.
C  NS     = START LOCATION IN VECTOR A.
C  NE     = END LOCATION IN VECTOR A.
C
C      NERROR  EXPLANATION
C  1 = START LOCATION GREATER THAN END LOCATION.
C  2 = END OF FILE ENCOUNTERED.
C  3 = WRITE PARITY ERROR.
C
                                                                 NERROR=1
      IF (NS .LE. 0 .OR. NS .GT. NE) RETURN
                                                                 NERROR=2
      WRITE (NUT,ERR=998,END=999) (A(I),I=NS,NE)
      RETURN
C
  998                                                            NERROR=3
  999 CALL ZZBOMB (4HYOUT    ,NERROR)
      END
```

```
      SUBROUTINE YOUTI (NUT,IA,NS,NE)
      DIMENSION IA(1)
C  WRITE OUT BINARY DATA FROM CORE AREA IA ONTO PERIPHERAL UNIT NUT.
C  CALLS FORMA SUBROUTINE ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  MAY 1973.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C       SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUT    = LOGICAL NUMBER OF UTILITY TAPE.
C  IA     = VECTOR TO BE WRITTEN.
C  NS     = START LOCATION IN VECTOR IA.
C  NE     = END LOCATION IN VECTOR IA.
C
C      NERROR  EXPLANATION
C  1 = START LOCATION GREATER THAN END LOC 'ION.
C  2 = END OF FILE ENCOUNTERED.
C  3 = WRITE PARITY ERROR.
C
                                                                 NERROR=1
      IF (NS .LE. 0 .OR. NS .GT. NE) RETURN
                                                                 NERROR=2
      WRITE (NUT,ERR=998,END=999) (IA(I),I=NS,NE)
      RETURN
C
  998                                                            NERROR=3
  999 CALL ZZBOMB (5HYOUTI,NERROR)
      END
```

```
      SUBROUTINE YPART (NUTA,V,LV,KV,NUT1)
      DIMENSION V(1),LV(1),MHEAD(10),MPHEAD(10),M2HEAD(10)
      DATA LIT,NOT/5,6/
C
C  REPARTITION SPARSE MATRIX A BY ROWS.
C  USED TO REPARTITION SPARSE MATRIX A WHICH WAS FORMED IN A PROGRAM
C  HAVING A DIFFERENT DIMENSION KV.
C  THE MAXIMUM ALLOWABLE PARTITION SIZE IS KV/4.
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,YOUT   ,YOUTI ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1968.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C  2 = SIZE LIMITATION EXCEEDED.
C
      LAE=0
      NREC=0
      NNZPT=0
      DO 5 I=1,10
    5 M2HEAD(I) = 0
      MAXP=KV/4
      IF (MAXP.EQ.0) MAXP=1
      LPRS=1
      REWIND NUTA
      REWIND NUT1
C
      CALL YINI   (NUTA,MHEAD,1,10)
      KVCHK = MHEAD(6)
      IF (KVCHK.EQ.KV) RETURN
      NNZA = MHEAD(4)
      IF (NNZA.LE.1) RETURN
   10 CALL YINI   (NUTA,MPHEAD,1,10)
      NNZP = MPHEAD(1)
                                                              NERROR=1
      IF (NNZP.GT.KV) GO TO 999
      LAE=LAE+NNZP
      IF (NNZP.GT.0) GO TO 20
      CALL YINI   (NUTA,LV,KV,KV)
      CALL YIN    (NUTA,V,KV,KV)
   20 IF (LAE.GT.KV) GO TO 40
   30 LPRE=LPRS-1+NNZP
                                                              NERROR=2
      IF (LPRE.GT.KV) GO TO 999
      LAE=LPRE
      CALL YINI   (NUTA,LV,LPRS,LPRE)
      CALL YIN    (NUTA,V,LPRS,LPRE)
      NNZPT=NNZPT+NNZP
```

```
      LPRS=LPRE+1
      IF (NNZPT.LT.NNZA) GO TO 10
C
   40 IF (LAE.GT.KV) LAE=LAE-NNZP
      L=MAXP
      LPWS=1
      LPWE=L
      IF (LAE.LE.LPWE) GO TO 80
   50 IF (LPWE.EQ.LAE) GO TO 80
      IF (LV(LPWE)/100000 .LT. LV(LPWE+1)/100000) GO TO 80
      IF (MAXP.LT.2) GO TO 80
C
      DO 60 I=2,MAXP
      L=L-1
      IF (LV(LPWE)/100000 .EQ. LV(L)/100000) GO TO 60
      LPWE=L
      GO TO 80
   60 CONTINUE
C
   80 IF (LPWE.GT.LAE) LPWE=LAE
      L=LPWE
      NNZPW=LPWE-LPWS+1
      M2HEAD(1) = NNZPW
      M2HEAD(2) = LV(LPWS)
      M2HEAD(3) = LV(LPWE)
      CALL YOUTI  (NUTI,M2HEAD,1,10)
      NREC=NREC+1
      CALL YOUTI  (NUTI,LV,LPWS,LPWE)
      CALL YOUT   (NUTI,V,LPWS,LPWE)
      IF (LPWE.EQ.LAE .AND. NNZPT.EQ.NNZA) GO TO 105
      IF (LAE.LT.LPWE+MAXP) GO TO 85
      LPWS=LPWE+1
      LPWE=LPWS-1+MAXP
      L=LPWE
      GO TO 50
   85 MOVE=LAE-LPWE
      LAE=MOVE
      IF (MOVE.EQ.0) GO TO 102
C
      DO 100 I=1,MOVE
      L=L+1
      V(I)=V(L)
  100 LV(I)=LV(L)
C
  102 LPRS=LAE+1
      IF (NNZPT.LT.NNZA) GO TO 30
      LPWS=1
      LPWE=LAE
      GO TO 80
  105 REWIND NUTI
      REWIND NUTA
      MHEAD(3) = NREC
      MHEAD(6) = KV
      CALL YOUTI  (NUTA,MHEAD,1,10)
C
```

```
      DO 110 I=1,NREC
      CALL YINI    (NUT1,MPHEAD,1,10)
      CALL YOUTI   (NUTA,MPHEAD,1,10)
      NNZP = MPHEAD(1)
      CALL YINI    (NUT1,LV,1,NNZP)
      CALL YIN     (NUT1,V,1,NNZP)
      CALL YOUTI   (NUTA,LV,1,NNZP)
      CALL YOUT    (NUTA,V,1,NNZP)
  110 CONTINUE
C
      RETURN
C
  999 CALL ZZBOMB (5HYPART ,NERROR)
      END
```

```
      SUBROUTINE YPUNCH (NUTA,ANAME,V,LV,KV)
      DIMENSION V(1),LV(1),W(4),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  PUNCH SPARSE MATRIX A ON CARDS IN FORMA COMPATIBLE FORMAT FOR
C  SUBROUTINES READ,YREAD,JREAD.
C  CAUTION - ELEMENT LOCATIONS (LV) SHOULD BE IN INCREASING ORDER.
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1969.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  ANAME = MATRIX IDENTIFICATION. (A6 FORMAT)
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C
C      NERROR  EXPLANATION
C  1 = SIZE LIMITATION EXCEEDED.
C
 4001 FORMAT (A6,I4,I5,A6)
 4002 FORMAT (2I5,4E17.8)
 4003 FORMAT (10H0000000000)
C
      REWIND NUTA
      CALL YINI    (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      ISHAPE = MHEAD(7)
      PUNCH 4001, ANAME,NRA,NCA,ISHAPE
      IF (NNZA.EQ.0) GO TO 40
      LPS=1
      IFLAG=0
C
      DO 38 M=1,NPARTA
      CALL YINI    (NUTA,MHEAD,1,10)
      NNZP = MHEAD(1)
      LFELP = MHEAD(2)
      LLELP = MHEAD(3)
      IF (NNZP.GT.0) GO TO 2
      CALL YINI    (NUTA,MHEAD,1,2)
      CALL YIN     (NUTA,    V,1,2)
      GO TO 38
    2 LPE=LPS-1+NNZP
                                                          NERROR=1
      IF (LPE.GT.KV) GO TO 999
      CALL YINI    (NUTA,LV,LPS,LPE)
      CALL YIN     (NUTA,V,LPS,LPE)
C
      DO 35 I=LPS,LPE
      IA=LV(I)/100000
      JA=LV(I)-100000*IA
```

```
      IF (I.EQ.1 .AND. M.EQ.1) GO TO 20
      K=JA-JS+1
      IF (IA.NE.IS .OR. K.LE.0 .OR. K.GT.4) GO TO 5
      W(K)=V(I)
      IF (I.LT.LPE .OR. M.LT.NPARTA) GO TO 35
      IFLAG=1
    5 NJ=4
      IF ((JS+3).GT.NCA) NJ=NCA-JS+1
      IF (JA.GT.NCA) NJ=4
      PUNCH 4002, IS,JS,(W(J),J=1,NJ)
      IF (NNZP.EQ.0) GO TO 38
      IF (IFLAG.EQ.1) GO TO 35
   20 IS=IA
      JS=JA
      W(1)=V(I)
      DO 30 K=2,4
   30 W(K)=0.
      IF (I.LT.LPE .OR. M.LT.NPARTA) GO TO 35
      IFLAG=1
      GO TO 5
   35 CONTINUE
   38 CONTINUE
C
   40 PUNCH 4003
      RETURN
C
  999 CALL ZZBOMB (6HYPUNCH,NERROR)
      END
```

```
      SUBROUTINE YREAD (NUTA,V,LV,KV,NUT1)
      DIMENSION V(1),LV(1),X(4),IREMRK(9),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  READ SPARSE MATRIX FROM CARDS OR TAPE AND PRINT IT. WRITE MATRIX ON
C  TAPE IF SO INDICATED (BY NWTAPE .NE. 0 IN COLUMNS 79-80).
C  CALLS FORMA SUBROUTINES INTAPE,LTAPE ,PAGEHD,YIN   ,YIN1  ,YOUT  ,
C                          YOUT1 ,YPART ,YRTAPE,YWRITE,YWTAPE,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1968.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C  **** CARD INPUT ****
C  FIRST CARD    - ANAME,NROWS,NCOLS WITH A6,I4,I5 FORMAT.
C                - REMARKS IN COLUMNS 16-69.
C                - IF THIS IS THE LOWER (OR UPPER) HALF OF A SYMMETRIC
C                  MATRIX, THE WORD LOWER (OR UPPER) MUST APPEAR IN
C                  COLUMNS 16-20.
C                - $ IN COLUMN 72 FOR NWTAPE INITIALIZATION.
C                - CONTRL IN 73-78. CONTRL MAY BE BLANK, OR THE WORDS
C                  REWIND OR LIST, OR (WHEN $ IN 72) THE NWTAPE TAPE-ID
C                  (EG T1234).
C                - NWTAPE (LOGICAL TAPE NUMBER) IN COL 79-80 (EG 12).
C  MIDDLE CARDS - DATA WITH FORMAT (2I5, 4E17.0).
C                - 1-ST I5 IS THE ROW NUMBER.
C                - 2-ND I5 IS THE COL NUMBER OF THE NEXT E17.0 FIELD.
C                - NEXT 4E17.0 ARE ELEMENTS OF THE MATRIX.
C  LAST CARD     - TEN ZEROS IN COLUMNS 1-10.
C
C  **** TAPE INPUT ****
C  ONE CARD  - ANAME,LOC(ZERO OR MINUS THE LOCATION NUMBER ON NRTAPE),
C              NRTAPE(NUMBER OF READ TAPE, IF MINUS NO PRINTOUT), ARUNNO
C              RCNTRL(BLANK,REWIND,OR LIST) WITH FORMAT (A6,I4,I5,A6,A6)
C            - REMARKS IN 28-69.
C            - $,CONTRL,NWTAPE SAME AS FIRST CARD UNDER CARD INPUT.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = ROW OR COLUMN VALUE OF ELEMENT EXCEEDS MATRIX SIZE.
C  2 = DATA ON CARD PAST MATRIX COLUMN SIZE.
C  3 = MATRIX IS DESIGNATED LOWER HALF SYMMETRIC BUT NON-ZEROES EXIST
C      ONLY IN THE UPPER HALF.
C  4 = MATRIX IS DESIGNATED UPPER HALF SYMMETRIC BUT NON-ZEROES EXIST
C      ONLY IN THE LOWER HALF.
C  5 = LOCATION ON TAPE PAST END OF TAPE MARK.
C  6 = LOCATION ON TAPE PAST END OF TAPE MARK.
C
 1001 FORMAT (A6,I4,I5,9A6, 2XA1,A6,I2)
 1002 FORMAT (2I5,4E17.0)
 2001 FORMAT (//26H CARD INPUT SPARSE MATRIX A6, 2X,1H(,I4,2H X,I4,2H ),
```

```
     *          2X 9A6,2X A1,A6,I4//)
 2002 FORMAT (//26H CARD INPUT SPARSE MATRIX A6, 2X,1H(,I4,2H X,I4,2H )
     *          3X, 9HCONTINUED  //)
 2003 FORMAT (// 1XA6,14,I5,5X 9A6,2X A1,A6,I4)
 2004 FORMAT (1X 2I5,4E17.8)
 2005 FORMAT (14HOEND OF YREAD.
     *         /31H NUMBER OF NON-ZERO ELEMENTS = I5
     *         /24H NUMBER OF PARTITIONS = I3)
 2006 FORMAT (25HOSIZE OF MATRIX READ IS (I4,2H XI4,2H ))
C
C  READ IN HEADER CARD.
      NUT=NUT1
      READ (NIT,1001) ANAME,N1,N2,IREMRK,IZ1,IZ2,NWTAPE
      CALL PAGEHD
C
      IF (N1.LE.0)  GO TO 200
C
C  CARD READING SECTION.
      LOW=0
      LUP=0
      REWIND NUT
      NNZA=0
      NREC=0
      LAE=0
      NRA=N1
      NCA=N2
      WRITE (NOT,2001) ANAME,NRA,NCA,IREMRK,IZ1,IZ2,NWTAPE
      NLINE=0
  110 READ (NIT,1002) I,JS,X
      IF (I.EQ.0 .AND. JS.EQ.0)  GO TO 132
                                                           NERROR=1
      IF (I.LE.0 .OR. I.GT.NRA .OR. JS.LE.0 .OR. JS.GT.NCA) GO TO 990
      JE = JS+3
      IF (JE.LE.NCA) GO TO 115
      JX=NCA-JS+2
                                                           NERROR=2
      DO 112 J=JX,4
      IF (X(J) .NE. 0.) GO TO 990
  112 CONTINUE
      JE=NCA
  115 N = 0
      DO 120 J=JS,JE
      N = N+1
      IF (X(N).EQ.0.) GO TO 120
      IF (I.GT.J) LOW=1
      IF (I.LT.J) LUP=1
      IF (IREMRK(1).EQ.5HLOWER .AND. I.LT.J) GO TO 120
      IF (IREMRK(1).EQ.5HUPPER .AND. I.GT.J) GO TO 120
      LAE=LAE+1
      IF (LAE.LE.KV/4) GO TO 118
      LAE = LAE-1
      MHEAD(1) = LAE
      CALL YOUTI  (NUT1,MHEAD,1,1)
      NREC=NREC+1
      CALL YOUTI  (NUT1,LV,1,LAE)
```

```
      CALL YOUT   (NUT1,V,1,LAE)
      LAE=1
  118 NNZA=NNZA+1
      V(LAE)=X(N)
      LV(LAE)=100000*I+J
  120 CONTINUE
      NLINE = NLINE+1
      IF (NLINE.LE.47) GO TO 125
      CALL PAGEHD
      WRITE (NOT,2002) ANAME,NRA,NCA
      NLINE = 1
  125 WRITE (NOT,2004) I,JS,X
      GO TO 110
  132 IF (LAE.EQ.0) GO TO 135
      MHEAD(1) = LAE
      CALL YOUTI  (NUT1,MHEAD,1,1)
      NREC=NREC+1
      CALL YOUTI  (NUT1,LV,1,LAE)
      CALL YOUT   (NUT1,V,1,LAE)
  135 REWIND NUT
      REWIND NUTA
      IASHAP=IREMRK(1)
      IF (IASHAP.NE.5HWHOLE .AND. IASHAP.NE.5HUPPER .AND.
     *    IASHAP.NE.5HLOWER .AND. IASHAP.NE.4HDIAG) IASHAP=5HWHOLE
      IF (NNZA.EQ.0) GO TO 137
                                                              NERROR=3
      IF (IREMRK(1).EQ.5HLOWER .AND. LOW.EQ.0 .AND. LUP.EQ.1) GO TO 999
                                                              NERROR=4
      IF (IREMRK(1).EQ.5HUPPER .AND. LUP.EQ.0 .AND. LOW.EQ.1) GO TO 999
  137 MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NREC
      MHEAD(4) = NNZA
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = IASHAP
      MHEAD(8) = 0
      MHEAD(9) = 0
      CALL YOUTI  (NUTA,MHEAD,1,10)
      DO 139 I=1,10
  139 MHEAD(I) = 0
      IF (NNZA.GT.0) GO TO 138
      CALL YOUTI  (NUTA,MHEAD,1,10)
      CALL YOUTI  (NUTA,MHEAD,1,2)
      CALL YOUTI  (NUTA,MHEAD,1,2)
      GO TO 300
  138 DO 140 I=1,NREC
      CALL YINI   (NUT1,MHEAD,1,1)
      NNZP = MHEAD(1)
      CALL YINI   (NUT1,LV,1,NNZP)
      CALL YIN    (NUT1,V,1,NNZP)
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZP)
      CALL YOUTI  (NUTA,MHEAD,1,10)
      CALL YOUTI  (NUTA,LV,1,NNZP)
```

```
  140 CALL YOUT    (NUTA,V,I,NNZF)
      CALL YPART   (NUTA,V,LV,KV,NUT1)
      GO TO 300
C
C  TAPE POSITIONING AND READING SECTION.
  200 WRITE (NOT,2003) ANAME,N1,N2,IREMRK,IZ1,IZ2,NWTAPE
      NRTAPE = IABS(N2)
      IF (IREMRK(2) .EQ. 6HREWIND)  REWIND NRTAPE
      IF (IREMRK(2) .EQ.4HLIST)  CALL LTAPE (NRTAPE)
      IF (N1.EQ.0)  GO TO 250
C     POSITION NRTAPE.
      READ  (NRTAPE) TID,LN,IEOTCK
      NUM = LN+N1
                                                                NERROR=5
      IF (NUM) 205,220,225
  205 IF (IEOTCK.EQ.3HEOT)  GO TO 900
      READ  (NRTAPE)
      NUM = -NUM-1
      IF (NUM.EQ.0)  GO TO 250
                                                                NERROR=6
      DO 210 L=1,NUM
      READ  (NRTAPE) TID,LN, ..OTCK
      IF (IEOTCK.EQ.3HEOT)  GO TO 900
  210 READ  (NRTAPE)
      GO TO 250
  220 BACKSPACE NRTAPE
      GO TO 250
C  NOTE...THE FOLLOWING SECTION WAS DESIGNED PRIMARILY TO BE USED ON A
C        DISK. IT WILL WORK ON A TAPE BUT IT WILL NOT BE AS EFFICIENT
C  **********
  225 REWIND NRTAPE
      NUM = (-N1-1)*2
      IF (NUM.EQ.0)  GO TO 250
      DO 230 L=1,NUM
  230 READ  (NRTAPE)
C  **********
  250 CALL YRTAPE (IREMRK(1),ANAME,NUTA,V,LV,KV,NRTAPE,NUT1)
      REWIND NUTA
      CALL YINI    (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPART = MHEAD(3)
      NNZA = MHEAD(4)
      WRITE (NOT,2006) NRA,NCA
      IF (N2.GT.0) CALL YWRITE (NUTA,ANAME,V,LV,KV)
C
C  NWTAPE INITIALIZING, WRITING, AND LISTING SECTION.
  300 IF (NWTAPE.LE.0) GO TO 350
      IF (IZ1.EQ.1H$   )  CALL INTAPE (NWTAPE,IZ2)
      IF (IZ2.EQ.6HREWIND)REWIND NWTAPE
      CALL YWTAPE (NUTA,ANAME,V,LV,KV,NWTAPE)
      IF (IZ2.EQ.4HLIST) CALL LTAPE (NWTAPE)

  350 REWIND NUTA
      CALL YINI    (NUTA,MHEAD,1,10)
```

```
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPART = MHEAD(3)
      NNZA = MHEAD(4)
      WRITE (NOT,2005) NNZA,NPART
      RETURN
C
  900 CALL LTAPE  (NRTAPE)
      CALL ZZBOMB (5HYREAD ,NERROR)
  990 WRITE (NOT,2004) I,JS,X
  999 CALL ZZBOMB (5HYREAD ,NERROR)
      END
```

```
      SUBROUTINE YREVAD (ALPHA,NUTA,IVEC,JVEC,NUTZ,V,LV,KV,NUT1,NUT2,
     *                  NUT3,NUT4)
      DIMENSION V(1),LV(1),IVEC(1),JVEC(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  REARRANGE ROWS AND COLUMNS OF ALPHA * SPARSE MATRIX A AND ADD TO
C  SPARSE MATRIX Z. (ALPHA * A + Z = Z).
C  CALLS FORMA SUBROUTINES XLORD ,YAABB ,YIN   ,YINI  ,YLORD ,YNOZER,
C                          YOUT  ,YOUTI ,YPART ,YSYMLH,YSYMUH,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  DECEMBER 1969.
C  LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  ALPHA = SCALAR THAT MULTIPLIES MATRIX A.
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  IVEC  = VECTOR. (SIZE = NRA)
C          IVEC(I) = ROW POSITION OF A(ROW I) IN Z.
C          IF IVEC(I) IS PLUS , Z = Z(ROW IVEC(I)) + ALPHA * A(ROW I)
C          IF IVEC(I) IS MINUS, Z = Z(ROW IVEC(I)) - ALPHA * A(ROW I)
C          IF (IVEC(I) IS ZERO , A(ROW I) IS OMITTED IN Z.
C  JVEC  = VECTOR. (SIZE = NCA)
C          JVEC(J) = COLUMN POSITION OF A(COL J) IN Z.
C          IF JVEC(J) IS PLUS , Z = Z(COL JVEC(J)) + ALPHA * A(COL J)
C          IF JVEC(J) IS MINUS, Z = Z(COL JVEC(J)) - ALPHA * A(COL J)
C          IF JVEC(J) IS ZERO , A(COL J) IS OMITTED IN Z.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT3  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT4  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = ROW LOCATION OUTSIDE MATRIX Z.
C  2 = COLUMN LOCATION OUTSIDE MATRIX Z.
C
      CALL YPART  (NUTA,V,LV,KV,NUT1)
C  GET (A) HEADER INFORMATION.
      REWIND NUTA
      REWIND NUT1
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      IF (NNZA.EQ.0) RETURN
      ISHAP = MHEAD(7)
C  GET (Z) HEADER INFORMATION.
      REWIND NUTZ
      CALL YINI   (NUTZ,MHEAD,1,2)
      NRZ = MHEAD(1)
      NCZ = MHEAD(2)
C
```

```
C   CHECK SIZES.
                                                                    NERROR=1
       DO 2 I=1,NRA
       IF (IABS(IVEC(I)).GT.NRZ) GO TO 999
     2 CONTINUE
                                                                    NERROR=2
       DO 3 I=1,NCA
       IF (IABS(JVEC(I)).GT.NCZ) GO TO 999
     3 CONTINUE
C
       IF (ISHAP.NE.4HDIAG) GO TO 10
       ISHAP=5HWHOLE
       IF (NRA.NE.NCA) GO TO 10
C
       DO 5 I=1,NRA
       IF (IABS(IVEC(I)).NE.IABS(JVEC(I))) GO TO 10
     5 CONTINUE
C
       ISHAP=4HDIAG
    10 MHEAD(5) = 0
       MHEAD(6) = 0
       CALL YOUTI  (NUT1,MHEAD,1,10)
C
C   BLOW-UP (A) TO (Z) SIZE.
       DO 100 I=1,NPARTA
       CALL YINI   (NUTA,MHEAD,1,10)
       NNZPA = MHEAD(I)
       CALL YINI   (NUTA,LV,1,NNZPA)
       CALL YIN    (NUTA,V,1,NNZPA)
C
       DO 50 J=1,NNZPA
       IA=LV(J)/100000
       JA=LV(J)-100000*IA
       IF (IVEC(IA)) 15,25,35
    15 IF (JVEC(JA)) 20,25,30
    20 LV(J)=-100000*IVEC(IA)-JVEC(JA)
       GO TO 50
    25 V(J)=0.
       GO TO 50
    30 V(J)=-V(J)
       LV(J) =-100000*IVEC(IA)+JVEC(JA)
       GO TO 50
    35 IF (JVEC(JA)) 40,25,45
    40 V(J)=-V(J)
       LV(J) = 100000*IVEC(IA)-JVEC(JA)
       GO TO 50
    45 LV(J)=100000*IVEC(IA)+JVEC(JA)
    50 CONTINUE
C
       IF (ISHAP.EQ.5HWHOLE .OR. ISHAP.EQ.4HDIAG) GO TO 90
       IF (ISHAP.EQ.5HLOWER) GO TO 70
C
       DO 60 K=1,NNZPA
       IA=LV(K)/100000
       JA=LV(K)-100000*IA
```

```
      IF (IA.GT.JA) LV(K)=100000*JA+IA
   60 CONTINUE
C
      GO TO 90
C
   70 DO 80 K=1,NNZPA
      IA=LV(K)/100000
      JA=LV(K)-100000*IA
      IF (IA.LT.JA) LV(K)=100000*JA+IA
   80 CONTINUE
C
   90 MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZPA)
      CALL YOUTI  (NUT1,MHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,NNZPA)
  100 CALL YOUT   (NUT1,V,1,NNZPA)
C
      CALL YNOZER (NUT1,V,LV,KV,NUT2)
      REWIND NUTZ
      REWIND NUT2
      CALL YINI   (NUTZ,LV,1,10)
      CALL YOUTI  (NUT2,LV,1,10)
      NPARTZ=LV(3)
C
C  TRANSFER ORIGINAL (Z) FROM NUTZ TO NUT2.
      DO 110 J=1,NPARTZ
      CALL YINI   (NUTZ,LV,1,10)
      CALL YOUTI  (NUT2,LV,1,10)
      NNZ=LV(1)
  105 CALL YINI   (NUTZ,LV,1,NNZ)
      CALL YIN    (NUTZ,V,1,NNZ)
      CALL YOUTI  (NUT2,LV,1,NNZ)
      CALL YOUT   (NUT2,V,1,NNZ)
  110 CONTINUE
C
C  ADD ALPHA*BLOWN-UP (A) TO ORIGINAL (Z).
      CALL YAABB  (ALPHA,NUT1,1.,NUT2,NUTZ,V,LV,KV,NUT3,NUT4)
      RETURN
C
  999 CALL ZZBOMB (6HYREVAD,NERROR)
      END
```

```
      SUBROUTINE YRTAPE (IARUNO,IANAME,NUTA,V,LV,KV,NTAPE,NUT1)
      DIMENSION V(1),LV(1),MCHECK(2),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  READ SPARSE MATRIX A FROM TAPE (NTAPE) BY IDENTIFICATION OF IARUNO
C  AND IANAME AND STORE IT ON UTILITY TAPE (NUTA).
C  CALLS FORMA SUBROUTINES LTAPE ,YIN   ,YIN1  ,YOUT  ,YOUT1 ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1968.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  IARUNO = RUN NUMBER OF MATRIX A. (A6 FORMAT)
C  IANAME = MATRIX IDENTIFICATION. (A6 FORMAT)
C  NUTA   = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  V      = VECTOR WORK SPACE.
C  LV     = VECTOR WORK SPACE.
C  KV     = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NTAPE  = LOGICAL NUMBER OF TAPE FROM WHICH MATRIX A IS TO BE READ.
C  NUT1   = LOGICAL NUMBER OF UTILITY TAPE.
C
C      NERROR  EXPLANATION
C  1 = INCORRECT MATRIX TYPE.
C  2 = DIMENSION SIZE EXCEEDED (KV).
C  3 = IANAME AND IARUNO NOT FOUND ON NRTAPE.
C  4 = DIMENSION SIZE EXCEEDED (KV).
C
 3001 FORMAT (30H1YRTAPE CANNOT FIND   RUNNO = A6/
     *         22X8HANAME = A6/17X13HPARTITION NO.I5/30X6H------)
C
      NPF=1
      NREC=0
      NNZA=0
      NTIME=0
      REWIND NUTA
      REWIND NUT1
C  SEARCH TAPE FOR CORRECT HEADING.
    5 READ (NTAPE) TAPEID,LN,IEOTCK,ITRUNO,ITNAME,NRA,NCA,DATE,ITYP,NNZP
     *,              NP,NPT,(MCHECK(I),I=1,2),ISHAP
      IF (ISHAP.EQ.0 .AND. NRA.NE.NCA) ISHAP=5HWHOLE
      IF (ITRUNO.EQ.IARUNO .AND. ITNAME.EQ.IANAME) GO TO 10
      IF (IEOTCK.EQ.3HEOT) GO TO 20
      READ (NTAPE)
      GO TO 5
C  MATRIX HAS BEEN FOUND.
   10                                                          NERROR=1
      IF (ITYP.NE.5HSPART .AND. ITYP.NE.6HSPARSE) GO TO 990
      IF (ITYP.NE.5HSPART) GO TO 32
      IF (NP.EQ.NPF) GO TO 12
      READ (NTAPE)
      GO TO 5
   12                                                          NERROR=2
      IF (NNZP.GT.KV) GO TO 990
      IF (NNZP.GT.0 .OR. NPT.EQ.1) GO TO 15
      READ (NTAPE)
      NPF=NPF+1
```

```
      IF (NPF.GT.NPT) GO TO 25
      GO TO 5
   15 READ (NTAPE) (LV(I),V(I),I=1,NNZP)
      MHEAD(1) = NNZP
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZP)
      DO 16 I=4,10
   16 MHEAD(I) = 0
      CALL YOUTI  (NUT1,MHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,NNZP)
      CALL YOUT   (NUT1,V,1,NNZP)
      NTIME=0
      NREC=NREC+1
      NPF=NPF+1
      NNZA=NNZA+NNZP
      IF (NPF.GT.NPT) GO TO 25
      GO TO 5
C  SEE IF MATRIX WAS FOUND.
   20 NTIME=NTIME+1
                                                     NERROR=3
      IF (NTIME.EQ.2) GO TO 900
      REWIND  TAPE
      GO TO 1
   25 MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NREC
      MHEAD(4) = NNZA
      MHEAD(5) = MCHECK(1)
      MHEAD(6) = MCHECK(2)
      MHEAD(7) = ISHAP
      CALL YOUTI  (NUTA,MHEAD,1,10)
      REWIND NUT1
C
      DO 30 I=1,NREC
      CALL YINI   (NUT1,MHEAD,1,10)
      CALL YINI   (NUT1,LV,1,MHEAD(1))
      CALL YIN    (NUT1,V,1,MHEAD(1))
      CALL YOUTI  (NUTA,MHEAD,1,10)
      CALL YOUTI  (NUTA,LV,1,MHEAD(1))
   30 CALL YOUT   (NUTA,V,1,MHEAD(1))
C
      RETURN
   32                                                NERROR=4
      IF (NNZP.GT.KV) GO TO 990
      READ (NTAPE) (LV(I),V(I),I=1,NNZP)
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NPF
      MHEAD(4) = NNZP
      MHEAD(5) = MCHECK(1)
      MHEAD(6) = MCHECK(2)
      MHEAD(7) = ISHAP
      CALL YOUTI  (NUTA,MHEAD,1,10)
      MHEAD(1) = NNZP
      MHEAD(2) = LV(1)
```

```
      MHEAD(3) = LV(NNZP)
      MHEAD(4) = 0
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = 0
      CALL YOUTI  (NUTA,MHEAD,1,10)
      CALL YOUTI  (NUTA,LV,1,NNZP)
      CALL YOUT   (NUTA,V,1,NNZP)
      RETURN
  900 WRITE (NOT,3001) IARUNC,IANAME,NPF
  990 CALL LTAPE  (NTAPE)
      CALL ZZBOMB (6HYRTAPE,NERROR)
      END
```

```
      SUBROUTINE YRV1   (NUTZ,N,NU,V,LV,KV,NUT1,NUT2,NUT3)
      DIMENSION V(1), LV(1)
      DIMENSION IH(10)
      DATA IR,IL / 100, 1000000000000 /
      DATA IMULT / 899999999 /
      DATA SCALE / 9999999999. /
C
C  GENERATE MATRIX OF RAYLEIGH VECTORS.
C  RANDOM NUMBERS BETWEEN -1. AND +1..
C  INPUT NON-ZERO COLUMNS ARE NOT CHANGED.
C  CALLS FORMA SUBROUTINES ...
C     YIN    ,YINI   ,YLORD ,YOUT   ,YOUTI ,YPART ,YTRANS,ZZBOMB.
C  DEVELOPED BY RA PHILIPPUS. MARCH 1972.
C  LAST REVISION BY RL WOHLEN. JUNE 1975.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTZ    = LOGICAL NUMBER OF UTILITY TAPE OF MATRIX OF INITIAL RAYLEIGH
C            VECTORS. NON-ZERO COLUMNS INPUT TO THIS SUBROUTINE ARE NOT
C            CHANGED.
C  N       = NUMBER OF ROWS OF RAYLEIGH VECTORS MATRIX.
C  NU      = NUMBER OF COLS OF RAYLEIGH VECTORS MATRIX.
C  V       = VECTOR WORK SPACE.
C  LV      = VECTOR WORK SPACE.
C  KV      = DIMENSION OF V,LV IN CALLING PROGRAM.
C  NUT1    = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2    = LOGICAL NUMBER OF UTILITY TAPE.
`  NUT3    = LOGICAL NUMBER OF UTILITY TAPE.

      KV14 = KV/4
C  TRANSFER ROWS ON NUTZ TO COLUMNS ON NUT3.
      CALL YTRANS (NUTZ,NUT3,V,LV,KV,NUT1,NUT2)
      REWIND NUT3
      REWIND NUTZ
      WW11 49WV
      NPARTZ = (NNZZ-1)/KV14+1
C  SET (Z) HEADER.
      IH(1) = N
      IH(2) = NU
      IH(3) = NPARTZ
      IH(4) = NNZZ
      IH(5) = 0
      IH(6) = 0
      IH(7) = 5HWHOLE
      IH(8) = 0
      IH(9) = 0
      IH(10) = 0
      CALL YOUTI  (NUTZ,IH,1,10)
      LZ = 0
      IRN = IMULT**2
C  READ DATA THAT ENTERED ON NUTZ BUT TRANSPOSED TO NUT3.
      CALL YINI   (NUT3,IH,1,10)
      NPARTI = IH(3)
      NREAD = 0
      U-0 49T  3
      LIS = LIE+1
```

```
      LI = LIS
      NNZPI = IH(4)
      IZ = 0
      IZP = 0
      IYOUT = 0
      DO 10 I=1,KV14
   10 V(I) = 0.
C  J=COLUMN NUMBER.
      DO 99 J=1,NU
      IF (NPARTI.EQ.0) GO TO 40
   20 IF (NREAD.EQ.NPARTI .AND. LI.GE.LIE) GO TO 40
      IF (LI.LE.LIE) GO TO 30
      CALL YINI   (NUT3,IH,1,10)
      NNZPI = IH(1)
      LIE = LIS-1+NNZPI
      CALL YINI   (NUT3,LV,LIS,LIE)
      CALL YIN    (NUT3, V,LIS,LIE)
      NREAD = NREAD+1
      LI = LIS
   30 IZ = LV(LI)/100000
      IF (IZ.GT.J) GO TO 40
      IF (IZ.LT.J) LI=LI+1
      IF (LI.GT.LIE) GO TO 20
      IF (IZ.LT.J) GO TO 30
      JZ = LV(LI)-100000*IZ
      IF (IZ.EQ.IZP) GO TO 34
      LZP1 = LZ+1
      LZPN = LZ+N
      LZPE = LZPN
      IF (LZPN.GT.KV14) LZPN=KV14
      L = 0
      DO 33 K=LZP1,LZPN
      L = L+1
   33 LV(K) = 100000*L+J
   34 IF (LZ+JZ.LE.KV14) GO TO 38
      IH(1) = KV14
      IH(2) = LV(1)
      IH(3) = LV(KV14)
      CALL YOUTI  (NUTZ,IH,1,10)
      CALL YOUTI  (NUTZ,LV,1,KV14)
      CALL YOUT   (NUTZ, V,1,KV14)
      IYOUT = IYOUT+1
      DO 37 K=1,KV14
      L = L+1
      LV(K) = 100000*L+J
   37 V(K) = 0.
      LZ = 1-JZ
   38 V(LZ+JZ) = V(LI)
      IZP = IZ
      LI = LI+1
      IF (LI.GT.LIE) GO TO 45
      GO TO 30
   40 IF (IZP.NE.J) GO TO 80
   45 LZ = LZ+N
      GO TO 99
```

```
C   PLACE RANDOM NUMBERS IN ONE COLUMN.
   80 DO 85 I=1,N
      IRN = IRN*IMULT
      JRN = (IRN-(IRN/IL)*IL)/IR
      IF ((JRN/2)*2 .EQ. JRN) JRN=-JRN
      RNUM = JRN
      LZ = LZ+1
      V(LZ) = RNUM/SCALE
      LV(LZ) = 100000*I +J
      IF (LZ.LT.KV14) GO TO 85
      IH(1) = LZ
      IH(2) = LV(1)
      IH(3) = LV(LZ)
      CALL YOUTI  (NUTZ,IH,1,10)
      CALL YOUTI  (NUTZ,LV,1,LZ)
      CALL YOUT   (NUTZ,V ,1,LZ)
      IYOUT = IYOUT+1
      LZ = 0
   85 CONTINUE
   99 CONTINUE
      IF (IYOUT.EQ.NPARTZ) GO TO 109
      IH(1) = LZ
      IH(2) = LV(1)
      IH(3) = LV(LZ)
      CALL YOUTI  (NUTZ,IH,1,10)
      CALL YOUTI  (NUTZ,LV,1,LZ)
      CALL YOUT   (NUTZ, V,1,LZ)
  109 CALL YLORD  (NUTZ,V,LV,KV,NUT1,NUT2)
      RETURN
      END
```

```
      SUBROUTINE YRVAD1 (ALPHA,A,IJVEC,NUTZ,NRA,V,LV,KV,KA,NUT1,NUT2,
     *                  NUT3,NUT4)
      DIMENSION V(1),LV(1),IJVEC(1), MHEAD(10), A(KA,1)
      DATA NIT,NOT/5,6/
      DATA EPS / 1.E-25 /
C
C  REARRANGE ROWS AND COLUMNS OF ALPHA * MATRIX A AND ADD TO
C  SPARSE MATRIX Z. (ALPHA * A + Z = Z).
C  MATRICES A,Z ARE ASSUMED SYMMETRIC.
C  CALLS FORMA SUBROUTINES XLORD ,YAMB ,YIN    ,YINI   ,YLORD ,YNOZER,
C                          YOUT   ,YOUTI ,YPART ,YSYMLH,YSYMUH,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1972.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  ALPHA = SCALAR THAT MULTIPLIES MATRIX A.
C  A     = MATRIX A.
C  IJVEC = VECTOR. (SIZE = NRA)
C          IJVEC(I) = ROW POSITION OF A(ROW I) IN Z.
C          IF IJVEC(I) IS PLUS , Z = Z(ROW IJVEC(I)) + ALPHA * A(ROW I)
C          IF IJVEC(I) IS MINUS, Z = Z(ROW IJVEC(I)) - ALPHA * A(ROW I)
C          IF (IJVEC(I) IS ZERO , A(ROW I) IS OMITTED IN Z.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  NRA   = NUMBER OF ROWS AND COLUMNS OF A.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  KA    = ROW DIMENSION OF MATRIX A IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT3  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT4  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = Z MATRIX IS NOT SQUARE.
C  2 = ROW OR COLUMN LOCATION OUTSIDE MATRIX Z.
C  3 = DIMENSION SIZE EXCEEDED (KV).
C
      REWIND NUT1
      REWIND NUTZ
      CALL YINI   (NUTZ,MHEAD,1,10)
      NRZ = MHEAD(1)
      NCZ = MHEAD(2)
                                                           NERROR=1
      IF (NRZ.NE.NCZ) GO TO 999
      ISHAP = MHEAD(7)
      IF (ISHAP.EQ.5HLOWER .OR. ISHAP.EQ.5HUPPER) GO TO 1
      CALL YZERUH (NUTZ,V,LV,KV,NUT1,NUT2)
      REWIND NUT1
      REWIND NUTZ
      CALL YINI   (NUTZ,MHEAD,1,10)
      ISHAP = 5HLOWER
      MHEAD(7) = ISHAP

                                                           NERROR=2
```

```
      DO 2 I=1,NRA
      IF (IABS(IJVEC(I)).GT.NRZ) GO TO 999
    2 CONTINUE
C
      J = 0
                                                          NERROR=3
      DO 50 IA=1,NRA
      DO 50 JA=1,NRA
      IF (IA.GT.JA .AND. ISHAP.EQ.5HUPPER) GO TO 50
      IF (IA.LT.JA .AND. ISHAP.EQ.5HLOWER) GO TO 50
      IF (ABS(A(IA,JA)).LT.EPS) GO TO 50
      J = J+1
      IF (J.GT.KV) GO TO 999
      IF (IJVEC(IA)) 15,25,35
   15 IF (IJVEC(JA)) 20,25,30
   20 LV(J) = -100000*IJVEC(IA)-IJVEC(JA)
      V(J) = A(IA,JA)
      GO TO 50
   25 J = J-1
      GO TO 50
   30 V(J) = -A(IA,JA)
      LV(J) = -100000*IJVEC(IA)+IJVEC(JA)
      GO TO 50
   35 IF (IJVEC(JA)) 40,25,45
   40 V(J) = -A(IA,JA)
      LV(J) = 100000*IJVEC(IA)-IJVEC(JA)
      GO TO 50
   45 LV(J) = 100000*IJVEC(IA)+IJVEC(JA)
      V(J) = A(IA,JA)
   50 CONTINUE
C
      IF (ISHAP.EQ.5HLOWER) GO TO 70
C
      DO 60 K=1,J
      IA = LV(K)/100000
      JA = LV(K)-100000*IA
      IF (IA.GT.JA) LV(K)=100000*JA+IA
   60 CONTINUE
C
      GO TO 90
C
   70 DO 80 K=1,J
      IA = LV(K)/100000
      JA = LV(K)-100000*IA
      IF (IA.LT.JA) LV(K)=100000*JA+IA
   80 CONTINUE
C
   90 MHEAD(3) = 1
      MHEAD(4) = J
      MHEAD(5) = 5HORDER
      MHEAD(6) = KV
      IF (J.GT.KV[illegible]) MHEAD(6) = 0
      CALL XLOR[illegible] [illegible]V,LV,1,J)
      CALL YO[illegible] (NUT1,MHEAD,1,10)
      MHEAD([illegible]) = J
```

```
      MHEAD(2) = LV(I)
      MHEAD(3) = LV(J)
      MHEAD(4) = 0
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = 0
      CALL YOUTI  (NUT1,MHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,J)
      CALL YOUT   (NUT1,V,1,J)
C
      REWIND NUTZ
      REWIND NUT2
      CALL YINI   (NUTZ,LV,1,10)
      CALL YOUTI  (NUT2,LV,1,10)
      NPARTZ=LV(3)
C
      DO 110 J=1,NPARTZ
      CALL YINI   (NUTZ,LV,1,10)
      CALL YOUTI  (NUT2,LV,1,10)
      NNZ=LV(1)
      IF (NNZ.GT.0) GO TO 105
      CALL YINI   (NUTZ,LV,1,2)
      CALL YOUTI  (NUT2,LV,1,2)
      CALL YIN    (NUTZ,V,1,2)
      CALL YOUT   (NUT2,V,1,2)
      GO TO 110
  105 CALL YINI   (NUTZ,LV,1,NNZ)
      CALL YIN    (NUTZ,V,1,NNZ)
      CALL YOUTI  (NUT2,LV,1,NNZ)
      CALL YOUT   (NUT2,V,1,NNZ)
  110 CONTINUE
C
      CALL YAABE  (ALPHA,NUT1,1.,NUT2,NUTZ,V,LV,KV,NUT3,NUT4)
      RETURN
C
  999 CALL ZZBOMB (6HYRVADI,NERROR)
      END
```

```
      SUBROUTINE YRVAD2 (NUTA,NUTZ,NRZ,W,KW,V,LV,KV,NUT1,NUT2,NUT3)
      DIMENSION V(1),LV(1),IU(16),IL(16),MHEAD(10),MPHEAD(10),M2HEAD(10)
      DIMENSION W(KW,1)
      COMMON / LWRKVI / IJVEC(500)
      DATA EPS / 1.E-25 /
C
C   REARRANGE ROWS AND COLUMNS OF SMALL DENSE MATRICES (A) BY SMALL IVECS
C   BOTH FROM NUTA, AND ADD MATRIX ELEMENTS AT LIKE LOCATIONS TO FORM
C   LARGE SPARSE MATRIX (Z) ON NUTZ. NUTZ IS INITIATED IN THIS SUBROUTINE
C   THUS, ANY PREVIOUS DATA ON NUTZ IS DESTROYED.
C   MATRIX (A) ELEMENTS WHOSE ABSOLUTE VALUE IS LESS THAN EPS ARE NOT
C   PLACED INTO MATRIX (Z).
C   A IS ASSUMED SYMMETRIC, ONLY THE LOWER HALF IS USED.
C   ONLY THE LOWER HALF OF Z (ALSO SYMMETRIC) IS FORMED.
C   CALLS FORMA SUBROUTINES YIN   ,YINI  ,YOUT  ,YOUTI ,YPART ,YZERO ,
C                               ZZBOMB.
C   DEVELOPED BY R A PHILIPPUS.  JANUARY 1973.
C   LAST REVISION BY RL WOHLEN FOR NASA.  MAY 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C   NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH SMALL MATRIX DATA IS
C           STORED. DATA CONSISTS OF SIZE, MATRIX ELEMENTS, IVEC.
C           NUTA IS READ IN THIS SUBROUTINE WITH A READ STATEMENT. THUS,
C           IT MUST HAVE BEEN GENERATED WITH A WRITE STATEMENT. NUTA
C           CANNOT BE USED WITH ANY OTHER SPARSE SUBROUTINE.
C   NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C   NRZ   = NUMBER OF ROWS AND COLUMNS OF SPARSE MATRIX Z.
C   W     = MATRIX WORK SPACE.
C   KW    = ROW DIMENSION OF W IN CALLING PROGRAM.
C   V     = VECTOR WORK SPACE.
C   LV    = VECTOR WORK SPACE.
C   KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C   NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C   NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C   NUT3  = LOGICAL NUMBER OF UTILITY TAPE.
C
C      NERROR  EXPLANATION
C   1 = ROW OR COLUMN LOCATION OUTSIDE MATRIX Z.
C   2 = NUTA CONTAINED ONLY NULL MATRICES.
C
      CALL YZERO  (NUTZ,NRZ,NRZ)
      REWIND NUTA
      READ (NUTA) NAME
      IF (NAME.EQ.6H      ) RETURN
      REWIND NUTA
      REWIND NUT1
      REWIND NUTZ
      KVO4=KV/4
C   SET (Z) MATRIX HEADER.
      MHEAD(1) = NRZ
      MHEAD(2) = NRZ
      MHEAD(3) = 0
      MHEAD(4) = 0
      MHEAD(5) = 5HORDER
      MHEAD(6) = KV
```

C12

```
      MHEAD(7) = 5HLOWER
      ISHAP = 5HLOWER
C
      LZ = 0
      NSEG = 0
   10 IRET = 1
      READ (NUTA) NAME,IE,N,(IB,I=1,7),((W(I,J),I=1,N),J=1,N),
     *              (IJVEC(I),I=1,N)
      IF (NAME.EQ.6H      ) GO TO 72
                                                      NERROR=1
      DO 15 I=1,N
      IF (IABS(IJVEC(I)).GT.NRZ) GO TO 999
   15 CONTINUE
      IA = 0
   20 IA = IA+1
      IF (IA.GT.N) GO TO 10
      JA = 0
   30 JA = JA+1
      IF (JA.GT.N) GO TO 20
      IF (IA.LT.JA) GO TO 20
      IF (ABS(W(IA,JA)).LT.EPS) GO TO 30
      IF (IA.EQ.JA) GO TO 35
      IF (IABS(IJVEC(IA)).NE.IABS(IJVEC(JA))) GO TO 35
      W(IA,JA) = 2.*W(IA,JA)
   35 LZ = LZ+1
      IRET = 2
      IF (LZ.GT.KV04) GO TO 72
      IF (IJVEC(IA)) 40,50,60
   40 IF (IJVEC(JA)) 45,50,55
   45 LV(LZ) = -100000*IJVEC(IA)-IJVEC(JA)
       V(LZ) =  W(IA,JA)
      GO TO 30
   50 LZ = LZ-1
      GO TO 30
   55 LV(LZ) = -100000*IJVEC(IA)+IJVEC(JA)
       V(LZ) = -W(IA,JA)
      GO TO 30
   60 IF (IJVEC(JA)) 65,50,70
   65 LV(LZ) =  100000*IJVEC(IA)-IJVEC(JA)
       V(LZ) = -W(IA,JA)
      GO TO 30
   70 LV(LZ) =  100000*IJVEC(IA)+IJVEC(JA)
       V(LZ) =  W(IA,JA)
      GO TO 30
   72 IF (LZ.GT.KV04) LZ=LZ-1
      IF (LZ.EQ.0) GO TO 225
      DO 85 K=1,LZ
      I = LV(K)/100000
      J = LV(K)-100000*I
      IF (I.LT.J) LV(K)=100000*J+I
   85 CONTINUE
C
C  USE YLORD LOGIC BUT ADD VALUES AT LIKE LOCATIONS.
C  SINGLETON METHOD
      M = 1
```

```
      LAEMI = LZ-1
      I=1
      J = LZ
  105 IF (I.GE.J) GO TO 170
  110 K=I
      IJ=(J+I)/2
      IT=LV(IJ)
      IF(LV(I).LE.IT) GO TO 120
      LV(IJ)=LV(I)
      LV(I)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(I)
      V(I)=TG
  120 L=J
      IF(LV(J).GE.IT) GO TO 140
      LV(IJ)=LV(J)
      LV(J)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(J)
      V(J)=TG
      IF(LV(I).LE.IT) GO TO 140
      LV(IJ)=LV(I)
      LV(I)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(I)
      V(I)=TG
      GO TO 140
  130 LV(L)=LV(K)
      LV(K)=ITT
      TG=V(L)
      V(L)=V(K)
      V(K)=TG
  140 L=L-1
      IF(LV(L).GT.IT) GO TO 140
      ITT=LV(L)
  150 K=K+1
      IF(LV(K).LT.IT) GO TO 150
      IF(K.LE.L) GO TO 130
      IF(L-I.LE.J-K) GO TO 160
      IL(M)=I
      IU(M)=L
      I=K
      M=M+1
      GO TO 180
  160 IL(M)=K
      IU(M)=J
      J=L
      M=M+1
      GO TO 180
  170 M=M-1
      IF(M.EQ.0) GO TO 210
      I=IL(M)
```

```
      J=IU(M)
  180 IF(J-I.GE.11) GO TO 110
      IF (I.EQ.1) GO TO 105
      I=I-1
  190 I=I+1
      IF(I.EQ.J) GO TO 170
      IT=LV(I+1)
      IF(LV(I).LE.IT) GO TO 190
      TG=V(I+1)
      K=I
  200 LV(K+1)=LV(K)
      V(K+1)=V(K)
      K=K-1
      IF(IT.LT.LV(K)) GO TO 200
      LV(K+1)=IT
      V(K+1)=TG
      GO TO 190
C
  210 LZE = 1
      LAW = 1
      IF (LZ.EQ.1) GO TO 222
      DO 220 I=2,LZ
      IF(LV(LAW) .EQ. LV(I)) V(LAW)=V(LAW)+V(I)
      V(LZE)=V(LAW)
      LV(LZE)=LV(LAW)
      IF (LV(LAW).EQ.LV(I)) GO TO 220
      LZE=LZE+1
      LAW=I
      IF (I.LT.LZ) GO TO 220
      V(LZE)=V(I)
      LV(LZE)=LV(I)
  220 CONTINUE
  222 NSEG = NSEG+1
      MPHEAD(1) = LZE
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(LZE)
      CALL YOUTI  (NUT1,MPHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,LZE)
      CALL YOUT   (NUT1, V,1,LZE)
      LZ = 0
      GO TO (225,35), IRET
C
  225 IF (NSEG.GT.1) GO TO 228
                                                      NERROR=2
      IF (NSEG.LT.1) GO TO 999
      REWIND NUTZ
      MHEAD(3) = 1
      MHEAD(4) = LZE
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      MPHEAD(1) = LZE
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(LZE)
      CALL YOUTI  (NUTZ,MPHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,1,LZE)
      CALL YOUT   (NUTZ, V,1,LZE)
```

```
      GO TO 410
C  NOW THERE ARE NSEG ORDERED GROUPS WRITTEN ON NUT1.
  228 NT1 = NUT1
      NT2=NUT2
      NREC = NSEG
      NSEG = 0
      NNZZ = 0
      REWIND NUT3
C
C  MESHING OPERATION
  230 REWIND NT1
      REWIND NT2
      IF (NREC.EQ.0) GO TO 305
      CALL YINI   (NT1,MPHEAD,1,10)
      NNZP1 = MPHEAD(1)
      CALL YINI   (NT1,LV,1,NNZP1)
      CALL YIN    (NT1,V,1,NNZP1)
      IF (NREC.EQ.1) GO TO 305
      NREC2=0
      LZE = NNZP1
C
      I = 1
  235 I = I+1
      IF (I.GT.NREC) GO TO 300
      CALL YINI   (NT1,M2HEAD,1,10)
      NNZP2 = M2HEAD(1)
      LP2S = NNZP1+1
      LP2E=LP2S-1+NNZP2
      CALL YINI   (NT1,LV,LP2S,LP2E)
      CALL YIN    (NT1,V,LP2S,LP2E)
      IF (LV(LP2S) .GT. LV(NNZP1)) GO TO 291
C
C  MESH TWO PARTITIONS
      I1=1
      I2 = LP2S
      IW=2*KVD4
      IZ=0
  250 IW=IW+1
      IF (LV(I1)-LV(I2)) 265,265,255
  255 V(IW)=V(I2)
      LV(IW)=LV(I2)
      I2=I2+1
      IF (I2.GT.LP2E) GO TO 275
      GO TO 250
  265 V(IW)=V(I1)
      LV(IW)=LV(I1)
      I1=I1+1
      IF (I1.GT.LP2S-1) GO TO 285
      GO TO 250
  275 NELTM = LP2S-I1
      K=LP2E
      L = LP2S-1
C
      DO 280 J=1,NELTM
```

```
      V(K)=V(L)
      LV(K)=LV(L)
      K=K-1
  280 L=L-1
C
  285 IF (IW.EQ.2*KV04) GO TO 291
      J1=2*KV04+1
C
      DO 290 J=J1,IW
      IZ=IZ+1
      V(IZ)=V(J)
  290 LV(IZ)=LV(J)
  291 LZE = 1
      LAW = 1
      DO 293 J=2,LP2E
      IF (LV(LAW).EQ.LV(J)) V(LAW)=V(LAW)+V(J)
      LV(LZE) = LV(LAW)
       V(LZE) =  V(LAW)
      IF (LV(LAW).EQ.LV(J)) GO TO 293
      LZE = LZE+1
      LAW = J
      IF (J.LT.LP2E) GO TO 293
      LV(LZE) = LV(J)
       V(LZE) =  V(J)
  293 CONTINUE
      IF (LZE.LT.KV04 .AND. NREC2.EQ.0) NNZP1=LZE
      IF (LZE.LT.KV04 .AND. NREC2.EQ.0) GO TO 235
      IF (LZE.EQ.NNZP1) GO TO 235
C
      NREC2 = NREC2+1
      M2HEAD(1) = LZE-NNZP1
      M2HEAD(2) = LV(NNZP1+1)
      M2HEAD(3) = LV(LZE)
      CALL YOUTI  (NT2,M2HEAD,1,10)
      CALL YOUTI  (NT2,LV,NNZP1+1,LZE)
      CALL YOUT   (NT2, V,NNZP1+1,LZE)
      GO TO 235
C
C  ALL NREC PARTITIONS HAVE BEEN READ FROM NT1
  300 MPHEAD(1) = NNZP1
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(NNZP1)
      CALL YOUTI  (NUT3,MPHEAD,1,10)
      CALL YOUTI  (NUT3,LV,1,NNZP1)
      CALL YOUT   (NUT3, V,1,NNZP1)
      NSEG = NSEG+1
      NNZZ = NNZZ+NNZP1
      NREC=NREC2
      NTS=NT1
      NT1=NT2
      NT2=NTS
      GO TO 230

  305 IF (NREC.EQ.0) GO TO 400
      CALL YOUTI  (NUT3,MPHEAD,1,10)
```

```
      CALL YOUTI  (NUT3,LV,1,NNZP1)
      CALL YOUT   (NUT3, V,1,NNZP1)
      NSEG = NSEG+1
      NNZZ = NNZZ+NNZP1
  400 REWIND NUT3
      REWIND NUTZ
      MHEAD(3) = NSEG
      MHEAD(4) = NNZZ
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      DO 405 I=1,NSEG
      CALL YINI   (NUT3,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      K = MHEAD(1)
      CALL YINI   (NUT3,LV,1,K)
      CALL YOUTI  (NUTZ,LV,1,K)
      CALL YIN    (NUT3, V,1,K)
  405 CALL YOUT   (NUTZ, V,1,K)
  410 CALL YNOZER (NUTZ,V,LV,KV,NUT1)
      RETURN
C
  999 CALL ZZBOMB (6HYRVAD2,NERROR)
      END
```

```
      SUBROUTINE YRVAD3 (NUTA,NUTZ,NRZ,NCZ,W,KW,V,LV,KV,NUT1,NUT2,NUT3)
      DIMENSION V(1),LV(1),IU(16),IL(16),MHEAD(10),MPHEAD(10),M2HEAD(10)
      DIMENSION W(KW,1)
      COMMON / LWRKV1 / I VEC(250), JVEC(250)
      DATA EPS / 1.E-25 /
C
C  REARRANGE ROWS AND COLUMNS OF SMALL DENSE MATRICES (A) BY SMALL IVECS
C  AND JVECS, ALL FROM NUTA, AND ADD MATRIX ELEMENTS AT LIKE LOCATIONS
C  TO FORM LARGE SPARSE MATRIX (Z) ON NUTZ. NUTZ IS INITIATED IN THIS
C  SUBROUTINE.  THUS, ANY PREVIOUS DATA ON NUTZ ARE DESTROYED.
C  MATRIX (A) ELEMENTS WHOSE ABSOLUTE VALUE IS LESS THAN EPS ARE NOT
C  PLACED INTO MATRIX (Z).  (A) IS SQUARE.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YOUT  ,YOUTI ,YPART ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  FEBRUARY 1975.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH SMALL MATRIX DATA IS
C          STORED. DATA CONSISTS OF SIZE, MATRIX ELEMENTS, IVEC.
C          NUTA IS READ IN THIS SUBROUTINE WITH A READ STATEMENT. THUS,
C          IT MUST HAVE BEEN GENERATED WITH A WRITE STATEMENT. NUTA
C          CANNOT BE USED WITH ANY OTHER SPARSE SUBROUTINE.
C  NUTZ  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  NRZ   = NUMBER OF ROWS OF SPARSE MATRIX Z.
C  NCZ   = NUMBER OF COLUMNS OF SPARSE MATRIX Z.
C  W     = MATRIX WORK SPACE.
C  KW    = ROW DIMENSION OF W IN CALLING PROGRAM.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT3  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = LOCAL SIZE EXCEEDS FINAL SIZE.
C  2 = NUMBER OF SEGMENTS LESS THAN ONE.
C
      REWIND NUTA
      REWIND NUT1
      REWIND NUTZ
      KVO4=KV/4
      MHEAD(1) = NRZ
      MHEAD(2) = NCZ
      MHEAD(5) = 
      MHEAD(6) = 0
      MHEAD(7) = 6HWHOLE
      ISHAP = 6HWHOLE
C
      LZ = 0
      NSEG = 0
   10 IRET = 1
      READ (NUTA) NAME,IB,N,(IB,I=1,7),((W(I,J),I=1,N),J=1,N),
     *            (I VEC(I),I=1,N), (JVEC(I),I=1,N)
      IF (NAME.EQ.6H      ) GO TO 72
```

```
                                                                  NERROR=1
      DO 15 I=1,N
      IF (IABS(I VEC(I)).GT.NRZ) GO TO 999
      IF (IABS(J VEC(I)).GT.NCZ) GO TO 999
   15 CONTINUE
      IA = 0
   20 IA = IA+1
      IF (IA.GT.N) GO TO 10
      JA = 0
   30 JA = JA+1
      IF (JA.GT.N) GO TO 20
      IF (ABS(W(IA,JA)).LT.EPS) GO TO 30
   35 LZ = LZ+1
      IRET = 2
      IF (LZ.GT.KV04) GO TO 72
      IF (I VEC(IA)) 40,50,60
   40 IF ( JVEC(JA)) 45,50,55
   45 LV(LZ) = -100000*I VEC(IA)- JVEC(JA)
       V(LZ) =  W(IA,JA)
      GO TO 30
   50 LZ = LZ-1
      GO TO 30
   55 LV(LZ) = -100000*I VEC(IA)+ JVEC(JA)
       V(LZ) = -W(IA,JA)
      GO TO 30
   60 IF ( JVEC(JA)) 65,50,70
   65 LV(LZ) =  100000*I VEC(IA)- JVEC(JA)
       V(LZ) = -W(IA,JA)
      GO TO 30
   70 LV(LZ) =  100000*I VEC(IA)+ JVEC(JA)
       V(LZ) =  W(IA,JA)
      GO TO 30
   72 IF (LZ.GT.KV04) LZ=LZ-1
      IF (LZ.EQ.0) GO TO 225
C
C  SINGLETON METHOD
      M = 1
      LAFM1 = LZ-1
      I=1
      J = LZ
  105 IF (I.GE.J) GO TO 170
  110 K=I
      IJ=(J+I)/2
      IT=LV(IJ)
      IF(LV(I).LE.IT) GO TO 120
      LV(IJ)=LV(I)
      LV(I)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(I)
      V(I)=TG
  120 L=J
      IF(LV(J).GE.IT) GO TO 140
      LV(IJ)=LV(J)
      LV(J)=IT
```

```
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(J)
      V(J)=TG
      IF(LV(I).LE.IT) GO TO 140
      LV(IJ)=LV(I)
      LV(I)=IT
      IT=LV(IJ)
      TG=V(IJ)
      V(IJ)=V(I)
      V(I)=TG
      GO TO 140
  130 LV(L)=LV(K)
      LV(K)=ITT
      TG=V(L)
      V(L)=V(K)
      V(K)=TG
  140 L=L-1
      IF(LV(L).GT.IT) GO TO 140
      ITT=LV(L)
  150 K=K+1
      IF(LV(K).LT.IT) GO TO 150
      IF(K.LE.L) GO TO 130
      IF(L-I.LE.J-K) GO TO 160
      IL(M)=I
      IU(M)=L
      I=K
      M=M+1
      GO TO 180
  160 IL(M)=K
      IU(M)=J
      J=L
      M=M+1
      GO TO 180
  170 M=M-1
      IF(M.EQ.0) GO TO 210
      I=IL(M)
      J=IU(M)
  180 IF(J-I.GE.11) GO TO 110
      IF (I.EQ.1) GO TO 105
      I=I-1
  190 I=I+1
      IF(I.EQ.J) GO TO 170
      IT=LV(I+1)
      IF(LV(I).LE.IT) GO TO 190
      TG=V(I+1)
      K=I
  200 LV(K+1)=LV(K)
      V(K+1)=V(K)
      K=K-1
      IF(IT.LT.LV(K)) GO TO 200
      LV(K+1)=IT
      V(K+1)=TG
      GO TO 190
C
```

```
  210 LZE = 1
      LAW = 1
      IF (LZ.EQ.1) GO TO 222
      DO 220 I=2,LZ
      IF(LV(LAW) .EQ. LV(I)) V(LAW)=V(LAW)+V(I)
      V(LZE)=V(LAW)
      LV(LZE)=LV(LAW)
      IF (LV(LAW).EQ.LV(I)) GO TO 220
      LZE=LZE+1
      LAW=I
      IF (I.LT.LZ) GO TO 220
      V(LZE)=V(I)
      LV(LZE)=LV(I)
  220 CONTINUE
  222 NSEG = NSEG+1
      MPHEAD(1) = LZE
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(LZE)
      CALL YOUTI  (NUT1,MPHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,LZE)
      CALL YOUT   (NUT1, V,1,LZE)
      LZ = 0
      GO TO (225,35), IRET
C
  225 IF (NSEG.GT.1) GO TO 228
                                                              NERROR=2
      IF (NSEG.LT.1) GO TO 999
      REWIND NUTZ
      MHEAD(3) = 1
      MHEAD(4) = LZE
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      MPHEAD(1) = LZE
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(LZE)
      CALL YOUTI  (NUTZ,MPHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,1,LZE)
      CALL YOUT   (NUTZ, V,1,LZE)
      GO TO 410
C
C  NOW THERE ARE NSEG ORDERED GROUPS WRITTEN ON NUT1.
  228 NT1 = NUT1
      NT2=NUT2
      NREC = NSEG
      NSEG = 0
      NNZZ = 0
      REWIND NUT3
C
C  MESHING OPERATION
  230 REWIND NT1
      REWIND NT2
      IF (NREC.EQ.0) GO TO 305
      CALL YINI   (NT1,MPHEAD,1,10)
      NNZPI = MPHEAD(1)
      CALL YINI   (NT1,LV,1,NNZPI)
      CALL YIN    (NT1,V,1,NNZPI)
```

```
      IF (NREC.EQ.1) GO TO 305
      NREC2=0
      LZE = NNZP1
C
      I = 1
  235 I = I+1
      IF (I.GT.NREC) GO TO 300
      CALL YINI    (NT1,M2HEAD,I,I0)
      NNZP2 = M2HEAD(1)
      LP2S = NNZP1+1
      LP2E=LP2S-1+NNZP2
      CALL YINI    (NT1,LV,LP2S,LP2E)
      CALL YIN     (NT1,V,LP2S,LP2E)
      IF (LV(LP2S) .GT. LV(NNZP1)) GO TO 291
C
C  MESH TWO PARTITIONS
      I1=1
      I2 = LP2S
      IW=2*KV04
      IZ=0
  250 IW=IW+1
      IF (LV(I1)-LV(I2)) 265,265,255
  255 V(IW)=V(I2)
      LV(IW)=LV(I2)
      I2=I2+1
      IF (I2.GT.LP2E) GO TO 275
      GO TO 250
  265 V(IW)=V(I1)
      LV(IW)=LV(I1)
      I1=I1+1
      IF (I1.GT.LP2S-1) GO TO 285
      GO TO 250
  275 NELTM = LP2S-I1
      K=LP2E
      L = LP2S-1
C
      DO 280 J=1,NELTM
      V(K)=V(L)
      LV(K)=LV(L)
      K=K-1
  280 L=L-1
C
  285 IF (IW.EQ.2*KV04) GO TO 291
      J1=2*KV04+1
C
      DO 290 J=J1,IW
      IZ=IZ+1
      V(IZ)=V(J)
  290 LV(IZ)=LV(J)
  291 LZE = 1
      LAW = 1
      DO 293 J=2,LP2E
      IF (LV(LAW).EQ.LV(J)) V(LAW)=V(LAW)+V(J)
      LV(LZE) = LV(LAW)
       V(LZE) =  V(LAW)
```

```
      IF (LV(LAW).EQ.LV(J)) GO TO 293
      LZE = LZE+1
      LAW = J
      IF (J.LT.LP2E) GO TO 293
      LV(LZE) = LV(J)
       V(LZE) =  V(J)
  293 CONTINUE
      IF (LZE.LT.KV04 .AND. NREC2.EQ.0) NNZP1=LZE
      IF (LZE.LT.KV04 .AND. NREC2.EQ.0) GO TO 235
      IF (LZE.EQ.NNZP1) GO TO 235
C
      NREC2 = NREC2+1
      M2HEAD(1) = LZE-NNZP1
      M2HEAD(2) = LV(NNZP1+1)
      M2HEAD(3) = LV(LZE)
      CALL YOUTI  (NT2,M2HEAD,1,10)
      CALL YOUTI  (NT2,LV,NNZP1+1,LZE)
      CALL YOUT   (NT2, V,NNZP1+1,LZE)
      GO TO 235
C
C  ALL NREC PARTITIONS HAVE BEEN READ FROM NT1
  300 MPHEAD(1) = NNZP1
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(NNZP1)
      CALL YOUTI  (NUT3,MPHEAD,1,10)
      CALL YOUTI  (NUT3,LV,1,NNZP1)
      CALL YOUT   (NUT3, V,1,NNZP1)
      NSEG = NSEG+1
      NNZZ = NNZZ+NNZP1
      NREC=NREC2
      NTS=NT1
      NT1=NT2
      NT2=NTS
      GO TO . 0
C
  305 IF (NREC.EQ.0) GO TO 400
      CALL YOUTI  (NUT3,MPHEAD,1,10)
      CALL YOUTI  (NUT3,LV,1,NNZP1)
      CALL YOUT   (NUT3, V,1,NNZP1)
      NSEG = NSEG+1
      NNZZ = NNZZ+NNZP1
  400 REWIND NUT3
      REWIND NUTZ
      MHEAD(3) = NSEG
      MHEAD(4) = NNZZ
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      DO 405 I=1,NSEG
      CALL YINI   (NUT3,MHEAD,1,10)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      K = MHEAD(1)
      CALL YINI   (NUT3,LV,1,K)
      CALL YOUTI  (NUTZ,LV,1,K)
      CALL YIN    (NUT3, V,1,K)
  405 CALL YOUT   (NUTZ, V,1,K)
  410 CALL YNOZER (NUTZ,V,LV,KV,NUT1)
```

```
      RETURN

  999 CALL ZZBOMB (6HYRVAD3,NERROR)
      END
```

```
      SUBROUTINE YRVIS1 (A,JVEC,NUTZ,NRAZ,NCA,NCZ,V,LV,KV,KA)
      DIMENSION V(1), LV(1), JVEC(1), MHEAD(10), A(KA,1)
      DATA EPS / 1.E-25 /
C
C  REARRANGE COLUMNS OF DENSE MATRIX A TO FORM SPARSE MATRIX Z.
C  ROWS OF A ARE NOT REARRANGED.
C  CALLS FORMA SUBROUTINES XLORD ,YCUT  ,YCUTI ,ZZBCMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1973.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  A      = MATRIX A.
C  JVEC   = VECTOR (SIZE = NCA).
C           JVEC(J) = COLUMN POSITION OF A(COL J) IN Z.
C           IF JVEC(J) IS PLUS , Z =  A(COL J).
C           IF JVEC(J) IS MINUS, Z = -A(COL J).
C           IF JVEC(J) IS ZERO , A(COL J) IS OMITTED IN Z.
C  NUTZ   = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX Z IS STORED.
C  NRAZ   = NUMBER OF ROWS OF A,Z.
C  NCA    = NUMBER OF COLUMNS OF A.
C  NCZ    = NUMBER OF COLUMNS OF Z.
C  V      = VECTOR WORK SPACE.
C  LV     = VECTOR WORK SPACE.
C  KV     = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  KA     = ROW DIMENSION OF MATRIX A IN CALLING PROGRAM.
C
C     NERROR  EXPLANATION
C  1 = COLUMN LOCATION OUTSIDE MATRIX Z.
C  2 = DIMENSION SIZE EXCEEDED (KV).
C
      REWIND NUTZ
C
                                                               NERROR=1
      DO 2 I=1,NCA
      IF (IABS(JVEC(I)).GT.NCZ) GO TO 999
    2 CONTINUE
C
      J = 0
                                                               NERROR=2
      DO 50 IA=1,NRAZ
      IA1 = 100000*IA
      DO 50 JA=1,NCA
      IF (ABS(A(IA,JA)).LT.EPS) GO TO 50
      J = J+1
      IF (J.GT.KV) GO TO 999
      IF (JVEC(JA)) 40,25,45
   25 J = J-1
      GO TO 50
   40 V(J) = -A(IA,JA)
      LV(J) = IA1-JVEC(JA)
      GO TO 50
   45 LV(J) = IA1+JVEC(JA)
      V(J) = A(IA,JA)
   50 CONTINUE
C
```

```
      MHEAD(1) = NRAZ
      MHEAD(2) = NCZ
      MHEAD(3) = 1
      MHEAD(4) = J
      MHEAD(5) = 5HORDER
      MHEAD(6) = KV
      MHEAD(7) = 5HWHOLE
      IF (J.GT.KV/4) MHEAD(6) = 0
      CALL XLORD  (V,LV,1,J)
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      MHEAD(1) = J
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(J)
      MHEAD(4) = 0
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = 0
      CALL YOUTI  (NUTZ,MHEAD,1,10)
      CALL YOUTI  (NUTZ,LV,1,J)
      CALL YOUT   (NUTZ,V,1,J)
C
      RETURN
  999 CALL ZZBOMB (6HYRVIS1,NERROR)
      END
```

```
      SUBROUTINE YRVTOD(NUTA,IVEC,JVEC,Z,NRZ,NCZ,V,LV,KV,KRZ)
      DIMENSION IVEC(1),JVEC(1),Z(KRZ,1),V(1),LV(1),MH(10)
C
C     REARRANGE AND ADD ROWS AND COLUMNS OF SPARSE MATRIX A
C     TO DENSE MATRIX Z.
C     BE SURE TO DEFINE Z BEFORE CALLING THIS SUBROUTINE.
C     FOR EXAMPLE CALL SUBROUTINE ZERO TO SET MATRIX Z TO ZERO.
C     CALLS FORMA SUBROUTINES YINI,YIN AND ZZBOMB.
C     CODED BY JOHN ADMIRE *NASA* DECEMBER 1973.
C
C                SUBROUTINE ARGUMENTS
C     NUTA - INPUT  LOGICAL NUMBER OF UTILITY TAPE ON WHICH
C                   SPARSE MATRIX A IS STORED.
C     IVEC - INPUT  IVEC(I)= ROW POSITION OF A(ROW I) IN Z.
C                   IF IVEC(I) IS PLUS ,Z=Z(ROW IVEC(I))+A(ROW I).
C                   IF IVEC(I) IS MINUS,Z=Z(ROW IVEC(I))-A(ROW I).
C                   IF IVEC(I) IS ZERO ,ROW I OF A DOES NOT APPEAR IN Z.
C     JVEC - INPUT  JVEC(J)= COL POSITION OF A(COL J) IN Z.
C                   IF JVEC(J) IS PLUS ,Z=Z(COL JVEC(J))+A(COL J).
C                   IF JVEC(J) IS MINUS,Z=Z(COL JVEC(J))-A(COL J).
C                   IF JVEC(J) IS ZERO ,COL J OF A DOES NOT APPEAR IN Z.
C     Z    - INPUT/OUTPUT MATRIX TO WHICH ELEMENTS OF A ARE ADDED OR
C                         SUBSTRACTED.
C     NRZ  - INPUT  NUMBER OF ROWS IN Z.
C     NCZ  - INPUT  NUMBER OF COLS IN Z.
C     V    - VECTOR WORK SPACE.
C     LV   - VECTOR WORK SPACE.
C     KV   - INPUT  DIMENSION OF V AND LV IN CALLING PROGRAM.
C     KRZ  - INPUT  DIMENSION OF NUMBER OF ROWS OF Z IN CALLING PROGRAM.
C
      NERROR=1
      IF(NRZ .GT. KRZ) GO TO 999
      REWIND NUTA
      CALL YINI(NUTA,MH,1,10)
      NRA=MH(1)
      NCA=MH(2)
      NPART=MH(3)
      IF(NPART .EQ. 0) RETURN
      NB=0
      DO 10 I=1,NRA
      IF(IABS(IVEC(I)) .GT. NB) NB=IABS(IVEC(I))
   10 CONTINUE
      NERROR=2
      IF(NB .GT. NRZ) GO TO 999
      NB=0
      DO 20 J=1,NCA
      IF(IABS(JVEC(J)) .GT. NB) NB=IABS(JVEC(J))
   20 CONTINUE
      NERROR=3
      IF(NB .GT. NCZ) GO TO 999
      DO 110 K=1,NPART
      CALL YINI(NUTA,MH,1,10)
      NNZP=MH(1)
      CALL YINI(NUTA,LV,1,NNZP)
      CALL YIN (NUTA,V,1,NNZP)
```

```
      DO 100 LL=1,NNZP
      I=LV(LL)/100000
      IF(IVEC(I))30,100,40
   30 II=IABS(IVEC(I))
      NS=-1
      GO TO 50
   40 II=IVEC(I)
      NS=1
   50 J=LV(LL)-100000*I
      IF(JVEC(J))60,100,70
   60 JJ=IABS(JVEC(J))
      NS=-NS
      GO TO 80
   70 JJ=JVEC(J)
   80 IF(NS .LT. 0) GO TO 90
      Z(II,JJ)=Z(II,JJ)+V(LL)
      GO TO 100
   90 Z(II,JJ)=Z(II,JJ)-V(LL)
  100 CONTINUE
  110 CONTINUE
      RETURN
  999 CALL ZZBOMB(GHYRVTOD,NERROR)
      END
```

```
      SUBROUTINE YSRED2 (NUTA,NUTR,NUTT,NR,IFT,V,LV,KV,NUT1,NUT2,NUT3,
     *                   NUT4)
      DIMENSION V(1),LV(1),MHEAD(10),MPHEAD(10),IH(10)
      DATA EPS/1.E-20/
C
C  REDUCE SPARSE STIFFNESS MATRIX (A) TO FORM SPARSE REDUCED STIFFNESS
C  MATRIX (R) AND (ON OPTION) REDUCTION TRANSFORMATION MATRIX (T).
C  COORDINATES TO BE RETAINED ARE NUMBERED LAST.
C  IF THE WHOLE MATRIX (A) IS INPUT, ONLY THE LOWER HALF IS USED.
C  BAND WIDTH (DIAGONAL UP TO TOP NON-ZERO) MUST BE LESS THAN OR EQUAL
C  TO (KV-N)/2, WHERE N IS MATRIX SIZE (SQUARE).
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                          YTRANS,ZZBOMB.
C  DEVELOPED BY R L WOHLEN AND W A BENFIELD.  DECEMBER 1972.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH (A) IS STORED.
C  NUTR  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH (R) IS STORED.
C  NUTT  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH (T) IS STORED.
C  NR    = NUMBER OF ROWS IN THE REDUCED STIFFNESS MATRIX.
C  IFT   = 0, TRANSFORMATION MATRIX (T) WILL NOT BE CALCULATED. NUTT
C              NEED NOT BE DEFINED IN CALLING PROGRAM.
C        = 1, TRANSFORMATION MATRIX (T) WILL BE CALCULATED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT3  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT4  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = BANDWIDTH LIMITATION EXCEEDED (KV).
C  2 = DIMENSION SIZE EXCEEDED (KV).
C  3 = BANDWIDTH LIMITATION EXCEEDED (KV).
C  4 = MATRIX IS SINGULAR.
C
C  CONVERT (A) FROM SPARSE (NUTA) TO BAND (NUT1) NOTATION.
      KVO4=KV/4
      KVO2 = KV/2
      KVO2P1 = KVO2+1
      LAS=KVO4+1
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      ND = NRA - NR
      KVMN = KV-NRA
      KVMNO2 = KVMN/2
      IASHAP = MHEAD(7)
      NUTS=NUTA                                                          S
      IF (IASHAP.EQ.5HUPPER) CALL YTRANS (NUTA,NUTR,V,LV,KV,NUT1,NUT2)
      IF (IASHAP.EQ.5HUPPER) NUTS=NUTR                                   S
      CALL YLORD  (NUTS,V,LV,KV,NUT1,NUT2)
      REWIND NUTS
```

```
      CALL YINI    (NUTS,MHEAD,1,10)
      REWIND NUTI
      ILV = KVO4
      JLV = KVO4+NRA
      IF (JLV.LT.KVO2) JLV=KVO2
      JLVS = JLV
      KP = 1
      LAMAX = LAS-1+KVMNO2
      LAE = LAS
      JS = 1
      NGROUP = 0
      LAS1 = KVO4
      DO 5 I=LAS,KV
      LV(I) = 0
    5 V(I)=0.
      NNZZ = 0
      NPARTA = MHEAD(3)
      NROWS = 1
      DO 20 I=1,NPARTA
      CALL YINI    (NUTS,MPHEAD,1,10)
      NNZPA = MPHEAD(1)
      CALL YINI    (NUTS,LV,1,NNZPA)
      CALL YIN     (NUTS,V,1,NNZPA)
      DO 20 J=1,NNZPA
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (IA.LT.JA) GO TO 20
      IF (IA.EQ.KP) GO TO 15
      LAS1 = LAE
      LAE = LAE+IA-JA+1
      NELR = KP-JS+1
                                                       NERROR=1
      IF (NELR.GT.KVMNO2) GO TO 999
      NNZZ = NNZZ+NELR
      KP = KP+1
      JS = JA
      NROWS = NROWS+1
      ILV = ILV+1
      LV(ILV) = NELR
      IF (LAE.LE.LAMAX) GO TO 15
      JLV = JLV+1
                                                       NERROR=2
      IF (JLV.GT.KV) GO TO 999
      NROWS = NROWS-1
      LV(JLV) = NROWS
      NROWS = 1
      LAE = LAE-IA+JA-1
      NGROUP = NGROUP+1
      CALL YOUT    (NUTI,V,LAS,LAE)                              1
      DO 10 L=LAS,LAE
   10 V(L)=0.
      LAS1 = KVO4
      LAE = KVO4+IA-JA+1
      KP = IA
   15 LA = LAS1+JA-JS+1
```

```
      V(LA)=V(J)
   20 CONTINUE
      IF (LAS.GT.LAE) GO TO 65
      NGROUP = NGROUP+1
      ILV = ILV+1
      LV(ILV) = KP-JS+1
                                                                   NERROR=3
      IF (LV(ILV).GT.KVMNO2) GO TO 999
      NNZZ = NNZZ+LV(ILV)
      JLV = JLV+1
      IF (JLV.GT.KV) GO TO 999
      LV(JLV) = NROWS
      CALL YOUT   (NUT1,V,LAS,LAE)                                             1
   65 DO 30 I=1,NRA
   30 LV(I) = LV(KVO4+I)
      DO 40 I=1,NGROUP
   40 LV(KVO2+I) = LV(JLVS+I)
C
C  REDUCTION.
C  D IN V(1 THRU N). A,U GROUP A START AT V(N+1).
C  A,U GROUP B START AT V(N+1+(KV-N)/2).
C  LV(I),I=1,N IS NUMBER OF ELEMENTS IN COLUMN I.
C  LV(KV/2+IG) IS NUMBER OF COLUMNS IN GROUP IG.
      N = NRA
      NG = NGROUP
      LSGA = N+1
      LSGB = LSGA + (KV-N)/2
      REWIND NUT3
      JEGA = 0
      DO 195 IGA=1,NG
      REWIND NUT1
      REWIND NUT2
      NUTP = NUT1
      NUTQ = NUT2
      IF (2*(IGA/2) .EQ. IGA) NUTP=NUT2
      IF (NUTP .EQ. NUT2)     NUTQ=NUT1
C  OPERATE ON GROUP A ONLY.
      NCGA = LV(KVO2+IGA)
      JSGA = JEGA+1
      JEGA = JSGA+NCGA-1
      LEGA = LSGA-1
      DO 101 J=JSGA,JEGA
  101 LEGA = LEGA + LV(J)
      CALL YIN    (NUTP,V,LSGA,LEGA)
      LJJ = LSGA-1
      DO 140 J=JSGA,JEGA
      JM1 = J-1
      IEND = J - 1
      IF (J .GT. ND) IEND=J
      ITOPJ = J-LV(J)+1
      LITOPJ = LJJ+1
      LJJ = LITOPJ+LV(J)-1
      IF (J .EQ. JSGA .AND. J .LE. ND) GO TO 134
      IF (ITOPJ .EQ. J) GO TO 134
      ISTART = ITOPJ
```

```
      LIJ = LITOPJ-1
      IF (ITOPJ .GE. JSGA) GO TO 105
      ISTART = JSGA
      LIJ = LITOPJ+JSGA-ITOPJ-1
  105 LITOPI = LSGA
      IF (ISTART .EQ. JSGA) GO TO 110
      ISM1 = ISTART-1
      DO 107 I=JSGA,ISM1
  107 LITOPI = LITOPI+LV(I)
  110 DO 128 I=ISTART,IEND
      KEND = I - 1
      IF (I .GT. ND) KEND = ND
      LIJ = LIJ+1
      S = V(LIJ)
      IM1 = I-1
      ITOPI = I-LV(I)+1
      IF (ITOPI .LT. ITOPJ) GO TO 115
      KSTART = ITOPI
      IF (I .EQ. KSTART) GO TO 125
      IF (KSTART .GT. KEND) GO TO 125
      LKI = LITOPI-1
      LKJ = LITOPJ+ITOPI-ITOPJ-1
      GO TO 120
  115 KSTART = ITOPJ
      IF (I .EQ. KSTART) GO TO 125
      IF (KSTART .GT. KEND) GO TO 125
      LKI = LITOPI+ITOPJ-ITOPI-1
      LKJ = LITOPJ-1
  120 DO 122 K=KSTART,KEND
      LKI = LKI+1
      LKJ = LKJ+1
  122 S = S-V(K)*V(LKI)*V(LKJ)
  125 V(LIJ) = S
      IF (I .LE. ND) V(LIJ) = S/V(I)
  128 LITOPI = LITOPI+LV(I)
  134 IF (J .GT. ND) GO TO 140
      V(J) = V(LJJ)
      IF (ITOPJ .EQ. J) GO TO 139
      LKJ = LITOPJ-1
      DO 138 K=ITOPJ,JM1
      LKJ = LKJ+1
  138 V(J) = V(J) - V(K)*V(LKJ)**2
  139                                                         NERROR=4
      IF (ABS(V(J)).LT.EPS) GO TO 999
      V(LJJ) = 1.0
  140 CONTINUE
C  GROUP A OPERATE ON GROUP B.
C  I COLUMNS ARE IN GROUP A, J COLUMNS IN GROUP B.
      IF (IGA .EQ. NG) GO TO 195
      IGAP1 = IGA+1
      JEGB = JEGA
      DO 192 IGB=IGAP1,NG
      NCGB = LV(KVP2+IGB)
      JSGB = JEGB+1
      JEGB = JSGB+NCGB-1
```

```
      LEGB = LSGB-1
      DO 151 J=JSGB,JEGB
  151 LEGB = LEGB+LV(J)
      CALL YIN    (NUTP,V,LSGB,LEGB)
      LJJ = LSGB-1
      DO 190 J=JSGB,JEGB
      JM1 = J-1
      ITOPJ = J-LV(J)+1
      LITOPJ = LJJ+1
      LJJ = LITOPJ+LV(J)-1
      IF (ITOPJ .GT. JEGA) GO TO 190
      ISTART = ITOPJ
      LIJ = LITOPJ-1
      IF (ITOPJ .GE. JSGA) GO TO 155
      ISTART = JSGA
      LIJ = LITOPJ+JSGA-ITOPJ-1
  155 LITOPI = LSGA
      IF (ISTART .EQ. JSGA) GO TO 160
      ISM1 = ISTART-1
      DO 157 I=JSGA,ISM1
  157 LITOPI = LITOPI+LV(I)
  160 DO 178 I=ISTART,JEGA
      KEND = I - 1
      IF (I .GT. ND) KEND=ND
      LIJ = LIJ+1
      S = V(LIJ)
      IM1 = I-1
      ITOPI = I-LV(I)+1
      IF (ITOPI .LT. ITOPJ) GO TO 165
      KSTART = ITOPI
      IF (I .EQ. KSTART) GO TO 175
      IF (KSTART .GT. KEND) GO TO 175
      LKI = LITOPI-1
      LKJ = LITOPJ+ITOPI-ITOPJ-1
      GO TO 170
  165 KSTART = ITOPJ
      IF (I .EQ. KSTART) GO TO 175
      IF (KSTART .GT. KEND) GO TO 175
      LKI = LITOPI+ITOPJ-ITOPI-1
      LKJ = LITOPJ-1
  170 DO 172 K=KSTART,KEND
      LKI = LKI+1
      LKJ = LKJ+1
  172 S = S-V(K)*V(LKI)*V(LKJ)
  175 V(LIJ) = S
      IF (I .LE. ND) V(LIJ)=S/V(I)
  178 LITOPI = LITOPI+LV(I)
  190 CONTINUE
  192 CALL YOUT   (NUTQ,V,LSGB,LEGB)
  195 CALL YOUT   (NUT3,V,LSGA,LEGA)
      REWIND NUT4
      CALL YOUTI  (NUT4,LV,1,KV)

C  CONVERT (U) FROM BAND (NUT3) TO SPARSE (NUT1) NOTATION.
C  DISASSEMBLE TO GET (U22) ON NUTR.
```

```
      REWIND NUT3
      REWIND NUT1
      LVGS = KV-NGROUP
      LVR = LVGS
      DO 202 IGROUP=1,NGROUP
      LVR = LVR+1
  202 LV(LVR) = LV(KVO2+IGROUP)
      LS = LVGS-NRA
      LVE = LS
      DO 204 I=1,NRA
      LVE = LVE+1
  204 LV(LVE) = LV(I)
      KVMAX = KV/4
      IF (KVMAX.GT.LS) KVMAX=LS
      MHEAD(1) = NRA
      MHEAD(2) = NRA
      MHEAD(3) = NGROUP
      MHEAD(4) = NNZZ
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = 5HUPPER
      CALL YOUT,  (NUT1,MHEAD,1,10)
      LVI = 0
      LVR = LVGS
      LVE = LS
      LVEP = LS
      IZ = 0
      DO 250 IGROUP=1,NGROUP
      LVR = LVR+1
      LZ = 0
      NROWS = LV(LVR)
      NELG = 0
      DO 206 I=1,NROWS
      LVE = LVE+1
  206 NELG = NELG+LV(LVE)
      CALL YIN    (NUT3,V,1,NELG)
      DO 208 I=1,NROWS
      IZ = IZ+1
      LVEP = LVEP+1
      JS = IZ-LV(LVEP)+1
      DO 208 JZ=JS,IZ
      LZ = LZ+1
  208 LV(LZ) = 100000*JZ+IZ
      MPHEAD(1) = LZ
      MPHEAD(2) = LV(1)
      MPHEAD(3) = LV(LZ)
      CALL YOUTI  (NUT1,MPHEAD,1,10)
      CALL YOUTI  (NUT1,LV,1,LZ)
      CALL YOUT   (NUT1,V,1,LZ)                                    1
  250 CONTINUE
      CALL YPART  (NUT1,V,LV,KV,NUT2)
      CALL YDISA  (NUT1,ND+1,ND+1,NUTR,NR,NR,V,LV,KV,NUT2)         F
C
C  CALCULATE REDUCTION TRANSFORMATION MATRIX.
      IF (IFT .EQ. 0) RETURN
```

```
C  (U) IS ON NUT3. BANDED, NO MATRIX OR PARTITION HEADERS.
C  TRANSFER U11 TO NUTT (TOP NON-ZERO IN COLUMN DOWN TO DIAGONAL) AND
C  U12 TO NUT1 (INDIVIDUAL FULL COLUMN).
      REWIND NUT4
      CALL YINI   (NUT4,LV,1,KV)
      DO 302 I=1,10
  302 MPHEAD(I) = 0
      REWIND NUT3
      REWIND NUTT
      REWIND NUT1
      JUE = 0
      NGU11 = 0
      DO 335 IG=1,NG
      NCG = LV(KVO2+IG)
      JUS = JUE+1
      JUE = JUS+NCG-1
      NELG = 0
      DO 306 JU=JUS,JUE
  306 NELG = NELG + LV(JU)
      CALL YIN    (NUT3,V,1,NELG)
      IF (ND .GE. JUE) GO TO 310
      IF (ND .GE. JUS) GO TO 320
      IF (ND .LT. JUS) GO TO 330
                                                       NERROR = 5
                                                       GO TO 999
  310 MPHEAD(1) = JUS
      MPHEAD(2) = JUE
      MPHEAD(3) = NELG
      CALL YOUTI  (NUTT,MPHEAD,1,10)
      CALL YOUT   (NUTT,V,1,NELG)
      NGU11 = NGU11+1
      GO TO 335
  320 NEL = 0
      DO 322 JU=JUS,ND
  322 NEL = NEL + LV(JU)
      MPHEAD(1) = JUS
      MPHEAD(2) = ND
      MPHEAD(3) = NEL
      CALL YOUTI  (NUTT,MPHEAD,1,10)
      CALL YOUT   (NUTT,V,1,NEL)
      NGU11 = NGU11+1
      IF (JUS .EQ. JUE) GO TO 335
      LEJU = NEL
      JUSX = ND+1
  323 DO 327 JU=JUSX,JUE
      ITOP = JU-LV(JU)+1
      LSJU = LEJU+1
      LEJU = LSJU+LV(JU)-1
      DO 324 IV=1,ITOP
      LIV = NELG+IV
  324 V(LIV) = 0.0
      IF (ITOP .GT. ND) GO TO 327
      LUV = LSJU-1
      DO 325 IV=ITOP,ND
      LIV = NELG+IV
```

```
      LUV = LUV+1
  325 V(LIV) = V(LUV)
  327 CALL YOUT   (NUT1,V,NELG+1,NELG+ND)                                1
      GO TO 335
  330 LEJU = 0
      JUSX = JUS
      GO TO 323
  335 CONTINUE
C  GROUP (U12) INDIVIDUAL FULL COLUMNS FROM NUT1 ONTO NUT2.
C  USE V(1 THRU (KV-N)/2) TO AGREE WITH YBSL3A.
      REWIND NUT1
      REWIND NUT2
      KV1 = (KV-N)/2
      LE = 0
      NCG = 0
      NGB = 0
      DO 343 J=1,NR
      LS = LE+1
      LE = LS+ND-1
      CALL YIN    (NUT1,V,LS,LE)
      NCG = NCG+1
      IF (J .EQ. NR) GO TO 342
      IF ((LE+ND) .LE. KV1) GO TO 343
  342 MPHEAD(1) = NCG
      CALL YOUTI  (NUT2,MPHEAD,1,10)
      CALL YOUT   (NUT2,V,1,LE)                                          2
      LE = 0
      NCG = 0
      NGB = NGB + 1
  343 CONTINUE
C
C  BACK SOLUTION FOR (T) FROM (U11)*(T)=(U12).
C  (U11) GROUPS ARE OBTAINED IN BACKWARDS ORDER.
C  V(1 THRU (KV-N)/2) CONTAINS Y=U12, Z=T COLUMNS OF A GROUP.
C  V((KV-N)/2+1 THRU KV-N) CONTAINS COLUMNS OF U (FROM TOP NON-ZERO
C  DOWN TO DIAGONAL) OF A GROUP.
C  LV(I),I=1,N IS NUMBER OF ELEMENTS IN COLUMN I.
      LSU = (KV-N)/2 + 1
      REWIND NUT2
      REWIND NUT1
C
      DO 389 IGB=1,NGB
      CALL YINI   (NUT2,IH,1,10)
      NCIGB = IH(1)
      NELIGB = ND*NCIGB
      CALL YIN    (NUT2,V,1,NELIGB)
      DO 357 IGUX=1,NGU11
      BACKSPACE NUTT
      BACKSPACE NUTT
      CALL YINI   (NUTT,IH,1,10)
      JSU = IH(1)
      JEU = IH(2)
      NELIGU = IH(3)
      CALL YIN    (NUTT,V,LSU,LSU+NELIGU-1)
      BACKSPACE NUTT
```

```
      BACKSPACE NUTT
      DO 357 JB=1,NCIGB
      LZSM1 = (JB-1)*ND
      LITJU = LSU+NELIGU
      DO 356 JUX=JSU,JEU
      JU = JSU+JEU-JUX
      LJJU = LITJU-1
      LITJU = LJJU-LV(JU)+1
      ITJU = JU-LV(JU)+1
      IF (ITJU .EQ. JU) GO TO 356
      LJJUM1 = LJJU-1
      LZ = LZSM1+JU
      LZY = LZSM1+ITJU-1
      DO 354 LU=LITJU,LJJUM1
      LZY = LZY+"
  354 V(LZY) = V(LZY) - V(LU)*V(LZ)
  356 CONTINUE
  357 CONTINUE
      DO 359 IGU=1,NGU11
      CALL YINI   (NUTT,IDUM,1,1)
  359 CALL YIN    (NUTT, DUM,1,1)
C
      DO 372 I=1,10
  372 IH(I) = 0
      IH(1) = NCIGB
      CALL YOUTI  (NUT1,IH,1,10)
      DO 375 I=1,NELIGB
  375 V(I) = -V(I)
      CALL YOUT   (NUT1,V,1,NELIGB)                                  1
  389 CONTINUE
C
C  CONVERT (T) FROM FULL COLUMN (NUT1) TO SPARSE (NUT3) NOTATION.
      REWIND NUT1
      REWIND NUT3
      IH(1) = ND
      IH(2) = NR
      IH(3) = NGB
      IH(4) = ND*NR
      IH(5) = 0
      IH(6) = 0
      IH(7) = 5HWHOLE
      CALL YOUTI  (NUT3,IH,1,10)
      JZ = 0
      DO 395 IGB=1,NGB
      CALL YINI   (NUT1,IH,1,10)
      NC = IH(1)
      NNZPB = ND*NC
      CALL YIN    (NUT1,V,1,NNZPB)
      LB = 0
      DO 392 J =1,NC
      JZ = JZ+1
      DO 392 IZ=1,ND
      LB = LB+1
  392 LV(LB) = 100000*IZ+JZ
      IH(1) = NNZPB
```

```
      IH(2) = LV(1)
      IH(3) = LV(NNZPB)
      CALL YOUTI  (NUT3,IH,1,10)
      CALL YOUTI  (NUT3,LV,1,NNZPB)
  395 CALL YOUT   (NUT3, V,1,NNZPB)                                   3=
      CALL YNOZER (NUT3,V,LV,KV,NUT1)
      CALL YLORD  (NUT3,V,LV,KV,NUT1,NUT2)
      CALL YZERO  (NUTT,NRA,NR)
      CALL YASSEM (NUT3,    1,1,NUTT,V,LV,KV,NUT1,NUT2,NUT4)
      CALL YUNITY (NUT3,NR,V,LV,KV)                                   3=
      CALL YASSEM (NUT3,ND+1,1,NUTT,V,LV,KV,NUT1,NUT2,NUT4)           T=
      RETURN
C
  999 CALL ZZE MB (6HYSRED2,NERROR)
      END
```

```
      SUBROUTINE YSTOD (NUTA,A,NRA,NCA,KRA,KCA,V,LV,KV,NUT1)
      DIMENSION V(1),LV(1),A(KRA,1),MHEAD(10),MPHEAD(10)
      DATA NIT,NOT/5,6/
C
C  CONVERT A MATRIX FROM SPARSE NOTATION TO DENSE NOTATION.
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1969.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS
C  NUTA  = INPUT  LOGICAL NUMBER OF UTILITY TAPE ON WHICH SPARSE MATRIX
C                  A IS STORED.
C  A     = OUTPUT DENSE MATRIX. SIZE (NRA,NCA).
C  NRA   = OUTPUT NUMBER OF ROWS IN A.
C  NCA   = OUTPUT NUMBER OF COLUMNS IN A.
C  KRA   = INPUT  ROW DIMENSION OF A IN CALLING PROGRAM.
C  KCA   = INPUT  COLUMN DIMENSION OF A IN CALLING PROGRAM.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = INPUT  DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = INPUT  LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = DIMENSION SIZE EXCEEDED (KRA,KCA).
C  2 = DIMENSION SIZE EXCEEDED (KV).
C  3 = ROW OR COLUMN LOCATION OUTSIDE MATRIX A.
C  4 = ROW OR COLUMN LOCATION OUTSIDE MATRIX A.
C
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
                                                            NERROR=1
      IF (NRA.GT.KRA .OR. NCA.GT.KCA) GO TO 999
C
      DO 10 I=1,NRA
      DO 10 J=1,NCA
   10 A(I,J)=0.
C
      NPART = MHEAD(3)
      ISHAPE = MHEAD(7)
      DO 40 K=1,NPART
      CALL YINI   (NUTA,MPHEAD,1,10)
      NNZP = MPHEAD(1)
                                                            NERROR=2
      IF (NNZP.GT.KV) GO TO 999
      IF (NNZP.GT.0) GO TO 20
      CALL YINI   (NUTA,LV,1,1)
      CALL YIN    (NUTA,V,1,1)
      GO TO 40
C
   20 CALL YINI   (NUTA,LV,1,NNZP)
      CALL YIN    (NUTA,V,1,NNZP)
      DO 30 L=1,NNZP
      I=LV(L)/100000
```

```
      J=LV(L)-100000*I
                                                        NERROR=3
      IF (I.GT.KRA .OR. J.GT.KCA) GO TO 999
      IF (ISHAPE.EQ.5HWHOLE) GO TO 30
                                                        NERROR=4
      IF (J.GT.KRA .OR. I.GT.KCA) GO TO 999
      A(J,I)=V(L)
   30 A(I,J)=V(L)
   40 CONTINUE
      RETURN
C
  999 CALL ZZBOMB (5HYSTOD ,NERROR)
      END
```

```
      SUBROUTINE YSYMLH (NUTAZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C   SYMMETRIZE SPARSE MATRIX AZ BY PLACING VALUES FROM ABOVE THE DIAGONAL
C   BELOW THE DIAGONAL.
C   CALLS FORMA SUBROUTINES YIN    ,YINI   ,YLORD ,YNOZER,YOUT   ,YOUTI ,
C                               YPART.
C   DEVELOPED BY R A PHILIPPUS.  JUNE 1969.
C   LAST REVISION BY R A PHILIPPUS.  JUNE 1973.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C   NUTAZ = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX AZ IS STORED.
C   V     = VECTOR WORK SPACE.
C   LV    = VECTOR WORK SPACE.
C   KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C   NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C   NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
      CALL YPART  (NUTAZ,V,LV,KV,NUT1)
C
      REWIND NUTAZ
      REWIND NUT1
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NNZA = MHEAD(4)
      IF (NNZA.EQ.0) RETURN
      ISHAP = MHEAD(7)
      IF (ISHAP.EQ.4HDIAG) RETURN
      MHEAD(7) = 5HWHOLE
C
      NPARTA = MHEAD(3)
      DO 20 I=1,NPARTA
      CALL YINI   (NUTAZ,MHEAD,10,10)
      NNZP = MHEAD(10)
      CALL YINI   (NUTAZ,LV,1,NNZP)
      CALL YIN    (NUTAZ,V,1,NNZP)
      NNZQ=NNZP
C
      DO 10 J=1,NNZP
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (IA.EQ.JA) GO TO 10
      IF (IA.GT.JA) GO TO 5
      NNZQ=NNZQ+1
      NNZA=NNZA+1
      LV(NNZQ)=100000*JA+IA
      V(NNZQ)=V(J)
      GO TO 10
    5 V(J)=0.
   10 CONTINUE
C
      MHEAD(10) = NNZQ
      CALL YOUTI  (NUT1,MHEAD,10,10)
      CALL YOUTI  (NUT1,LV,1,NNZQ)
   20 CALL YOUT   (NUT1,V,1,NNZQ)
```

```
C
      REWIND NUTAZ
      REWIND NUT1
      MHEAD(4) = NNZA
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(10) = 0
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
C
      DO 25 I=4,10
   25 MHEAD(I) = 0
      DO 30 I=1,NPARTA
      CALL YINI   (NUT1,MHEAD,1,1)
      NNZP = MHEAD(1)
      CALL YINI   (NUT1,LV,1,NNZP)
      CALL YIN    (NUT1,V,1,NNZP)
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZP)
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      CALL YOUTI  (NUTAZ,LV,1,NNZP)
   30 CALL YOUT   (NUTAZ,V,1,NNZP)
C
      CALL YNOZER (NUTAZ,V,LV,KV,NUT1)
      CALL YLORD  (NUTAZ,V,LV,KV,NUT1,NUT2)
      RETURN
      END
```

```
      SUBROUTINE YSYMUH (NUTAZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  SYMMETRIZE SPARSE MATRIX AZ BY PLACING VALUES FROM BELOW THE DIAGONAL
C  ABOVE THE DIAGONAL.
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,YLORD ,YNOZER,YOUT   ,YOUTI ,
C                          YPART .
C  DEVELOPED BY R A PHILIPPUS.  JUNE 1969.
C  LAST REVISION BY R A PHILIPPUS.  JUNE 1973.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTAZ = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX AZ IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
      CALL YPART  (NUTAZ,V,LV,KV,NUT1)
C
      REWIND NUTAZ
      REWIND NUT1
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NNZA = MHEAD(4)
      IF (NNZA.EQ.0) RETURN
      ISHAP = MHEAD(7)
      IF (ISHAP.EQ.4HDIAG) RETURN
      MHEAD(7) = 5HWHOLE
C
      NPARTA = MHEAD(3)
      DO 20 I=1,NPARTA
      CALL YINI   (NUTAZ,MHEAD,10,10)
      NNZP = MHEAD(10)
      CALL YINI   (NUTAZ,LV,1,NNZP)
      CALL YIN    (NUTAZ,V,1,NNZP)
      NNZQ=NNZP
C
      DO 10 J=1,NNZP
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (IA.EQ.JA) GO TO 10
      IF (IA.LT.JA) GO TO 5
      NNZQ=NNZQ+1
      NNZA=NNZA+1
      LV(NNZQ)=100000*JA+IA
      V(NNZQ)=V(J)
      GO TO 10
    5 V(J)=0.
   10 CONTINUE
C
      MHEAD(10) = NNZQ
      CALL YOUTI  (NUT1 ,MHEAD,10,10)
      CALL YOUTI  (NUT1 ,LV,1,NNZQ)
   20 CALL YOUT   (NUT1 ,V,1,NNZQ)
```

```
C
      REWIND NUTAZ
      REWIND NUT1
      MHEAD(4) = NNZA
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(10) = 0
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
C
      DO 25 I=4,10
   25 MHEAD(I) = 0
      DO 30 I=1,NPARTA
      CALL YINI   (NUT1,MHEAD,1,1)
      NNZP = MHEAD(1)
      CALL YINI   (NUT1,LV,1,NNZP)
      CALL YIN    (NUT1,V,1,NNZP)
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZP)
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      CALL YOUTI  (NUTAZ,LV,1,NNZP)
   30 CALL YOUT   (NUTAZ,V,1,NNZP)
C
      CALL YNOZER (NUTAZ,V,LV,KV,NUT1)
      CALL YLORD  (NUTAZ,V,LV,KV,NUT1,NUT2)
      RETURN
      END
```

```
      SUBROUTINE YTRANS (NUTA,NUTAT,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10),MPHEAD(10)
      DATA NIT,NCT/5,6/
C
C  TRANSPOSE SPARSE MATRIX A INTO SPARSE MATRIX AT.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                            ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1969.
C  LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  NUTAT = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX AT IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = DIMENSION SIZE EXCEEDED (KV).
C
      REWIND NUTA
      REWIND NUTAT
      CALL YINI   (NUTA,MHEAD,1,10)
      N = MHEAD(1)
      MHEAD(1) = MHEAD(2)
      MHEAD(2) = N
      IASHAP = MHEAD(7)
      ISHAP=IASHAP
      IF (IASHAP.EQ.5HUPPER) ISHAP=5HLOWER
      IF (IASHAP.EQ.5HLOWER) ISHAP=5HUPPER
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = ISHAP
      CALL YOUTI  (NUTAT,MHEAD,1,10)
      NNZA = MHEAD(4)
      NPART = MHEAD(3)
      IF (NNZA.GT.0) GO TO 3
      DO 7 I=1,10
    7 MPHEAD(I) = 0
      CALL YOUTI  (NUTAT,MPHEAD,1,10)
      CALL YOUTI  (NUTAT,MPHEAD,1,2)
      CALL YOUTI  (NUTAT,MPHEAD,1,2)
      RETURN
C
    3 DO 10 I=1,NPART
      CALL YINI   (NUTA,MPHEAD,1,10)
      NNZP = MPHEAD(1)
                                                         NERROR=1
      IF (NNZP.GT.KV) GO TO 999
      CALL YINI   (NUTA,LV,1,NNZP)
      CALL YIN    (NUTA,V,1,NNZP)
C
      DO 5 J=1,NNZP
```

```
      IA=LV(J)/100000
    5 LV(J)=100000*(LV(J)-100000*IA)+IA
C
      CALL YOUTI  (NUTAT,MPHEAD,1,10)
      CALL YOUTI  (NUTAT,LV,1,NNZP)
      CALL YOUT   (NUTAT,V,1,NNZP)
   10 CONTINUE
C
      CALL YLORD  (NUTAT,V,LV,KV,NUT1,NUT2)
      RETURN
C
  999 CALL ZZBOMB (6HYTRANS,NERROR)
      END
```

```
      SUBROUTINE YUNITY (NUTA,NRA,V,LV,KV)
      DIMENSION V(1),LV(1),MHEAD(10)
C
C  GENERATE SPARSE UNITY MATRIX A. (ONES ON THE DIAGONAL).
C  CALLS FORMA SUBROUTINES YOUT  ,YOUTI .
C  DEVELOPED BY R A PHILIPPUS.  JANUARY 1970.
C  LAST REVISION BY R A PHILIPPUS.  AUGUST 1973.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH UNITY MATRIX A IS
C          STORED.
C  NRA   = SIZE OF UNITY MATRIX A (SQUARE).
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C
      REWIND NUTA
      NPARTA=(NRA-1)/(KV/4)+1
      LV(1) = NRA
      LV(2) = NRA
      LV(3) = NPARTA
      LV(4) = NRA
      LV(5) = 5HORDER
      LV(6) = KV
      LV(7) = 4HDIAG
      LV(8) = 0
      LV(9) = 0
      LV(10) = 0
      CALL YOUTI  (NUTA,LV,1,10)
      DO 5 I=4,10
    5 MHEAD(I) = 0
      LAE=KV/4
      J=0
C
      DO 10 I=1,NRA
      J=J+1
      LV(J)=100000*I+I
      V(J)=1.
      IF (J.LT.LAE .AND. I.LT.NRA) GO TO 10
      MHEAD(1) = J
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(J)
      CALL YOUTI  (NUTA,MHEAD,1,10)
      CALL YOUTI  (NUTA,LV,1,J)
      CALL YOUT   (NUTA,V,1,J)
      J=0
   10 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE YWRITE (NUTA,ANAME,V,LV,KV)
      DIMENSION V(1),LV(1),W(10),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  WRITE SPARSE MATRIX A ON PAPER IN SAME FORMAT AS DENSE FORMA
C  SUBROUTINE WRITE. REQUIRES 132 COLUMN (MINIMUM) PRINTER.
C  UP TO 10 DATA FIELDS PER LINE. PRINT ONLY NON-ZERO FIELD ROWS.
C  CALLS FORMA SUBROUTINES PAGEHD,YIN   ,YINI   ,ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  SEPTEMBER 1968.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C  ANAME = MATRIX IDENTIFICATION. (A6 FORMAT)
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C
C      NERROR  EXPLANATION
C  1 = DIMENSION SIZE EXCEEDED (KV).
C
 2001 FORMAT (//22H OUTPUT SPARSE MATRIX A6,2X1H(I5,2H XI5,6H )  ( I5,
     *           21H NON-ZERO ELEMENTS) (I5,12H PARTITIONS)1XA6//
     *           10X,10(7X,1H(,I2,1H))/)
 2002 FORMAT (//22H OUTPUT SPARSE MATRIX A6,2X1H(I5,2H XI5,6H )  ( I5,
     * 21H NON-ZERO ELEMENTS) (I5,12H PARTITIONS)1XA6,10H CONTINUED//
     *           10X,10(7X,1H(,I2,1H))/)
 2003 FORMAT (1X,2I5,2X,1P10E11.4)
 2004 FORMAT (15H0END OF YWRITE.)
 3001 FORMAT (45H0END OF YWRITE. NRA OR NCA HAS BEEN EXCEEDED I5,
     *          7H TIMES.)
C
C  PULL UP A NEW PAGE FOR MATRIX AND PRINT MATRIX NAME.
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPART = MHEAD(3)
      NNZA = MHEAD(4)
      MCKORD = MHEAD(5)
      KVCHK = MHEAD(6)
      ISHAPE = MHEAD(7)
      CALL PAGEHD
      WRITE (NOT,2001) ANAME,NRA,NCA,NNZA,NPART,ISHAPE,(I,I=1,10)
      IF (NNZA.EQ.0) GO TO 40
      NLINE=0
      IFLAG=0
      IJK=0
C
      DO 38 M=1,NPART
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZP = MHEAD(1)
      LFELP = MHEAD(2)
      LLELP = MHEAD(3)
      IF (NNZP.GT.0) GO TO 2
```

```
      CALL YINI    (NUTA,MHEAD,10,10)
      CALL YIN     (NUTA,     V,KV,KV)
      MHEAD(10) = 0
      GO TO 38
    2                                                      NERROR=1
      IF (NNZP.GT.KV) GO TO 999
      CALL YINI    (NUTA,LV,1,NNZP)
      CALL YIN     (NUTA,V,1,NNZP)
C
      DO 35 I=1,NNZP
      IA=LV(I)/100000
      JA=LV(I)-100000*IA
      IF (IA.GT.NRA .OR. JA.GT.NCA) IJK=IJK+1
      IF (I.EQ.1 .AND. M.EQ.1) GO TO 20
      K=JA-JS+1
      IF (IA.NE.IS .OR. K.LE.0 .OR. K.GT.10) GO TO 5
      W(K)=V(I)
      IF (I.LT.NNZP .OR. M.LT.NPART) GO TO 35
      IFLAG=1
    5 NJ=10
      IF ((JS+9).GT.NCA) NJ=NCA-JS+1
      IF (JA.GT.NCA) NJ=10
      NLINE=NLINE+1
      IF (NLINE.LE.44) GO TO 10
      CALL PAGEHD
      WRITE (NOT,2002) ANAME,NRA,NCA,NNZA,NPART,ISHAPE,(J,J=1,10)
      NLINE = 1
   10 WRITE (NOT,2003) IS,JS,(W(J),J=1,NJ)
      IF (NNZP.EQ.0) GO TO 38
      IF (IFLAG.EQ.1) GO TO 35
C
C  SKIP A SPACE BETWEEN EACH ROW IF THERE ARE MORE THAN 10 COLUMNS
C  AND SOMETHING HAS BEEN WRITTEN.
      IF (NCA.LE.10 .OR. IL.EQ.IA) GO TO 25
      NLINE = NLINE + 1
      WRITE (NOT,2003)
   20 IL=IA
   25 IS=IA
      JS=(JA-1)/10*10+1
      DO 30 L=1,10
   30 W(L)=0.
      K=JA-JS+1
      W(K)=V(I)
      IF (I.LT.NNZP .OR. M.LT.NPART) GO TO 35
      IFLAG=1
      GO TO 5
   35 CONTINUE
   38 CONTINUE
C
      IF (IJK.EQ.0) GO TO 40
      WRITE (NOT,3001) IJK
      RETURN
   40 WRITE (NOT,2004)
      RETURN
C
```

```
999 CALL ZZBOMB (6HYWRITE,NERROR)
    END
```

```
      SUBROUTINE YWTAPE (NUTA,ANAME,V,LV,KV,NTAPE)
      DIMENSION V(1),LV(1),MCHECK(2),MHEAD(10)
      COMMON /LSTART/ IRUNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
      DATA NIT,NOT/5,6/
      DATA BUF,EOT,NONE,SPART / 0.,3HEOT,1,5HSPART /
C
C   WRITE SPARSE MATRIX A ON TAPE (NTAPE).
C   INITIALIZE TAPE WITH SUBROUTINE INTAPE.
C   REWIND TAPE BEFORE FIRST USE OF THIS SUBROUTINE.
C   NOTE...THIS ROUTINE IS DESIGNED SPECIFICALLY FOR WRITING ON A DISK
C          (EG CDC-6400 DISK). USING THIS ROUTINE TO WRITE ON A PHYSICAL
C          TAPE DIRECTLY (IE WITHOUT USING THE DISK AS AN INTERMEDIARY)
C          WILL PROBABLY GIVE POOR RESULTS (DUE TO THE TOLERANCE
C          CHARACTERISTICS OF A TAPE DRIVE) AND SHOULD BE AVOIDED IF AT
C          ALL POSSIBLE.
C       ...THE CDC-6400 DISK IS AUTOMATICALLY ENDFILED AFTER EACH WRITE.
C   CALLS FORMA SUBROUTINES YIN    ,YINI   ,ZZBOMB.
C   DEVELOPED BY R A PHILIPPUS.  NOVEMBER 1968.
C   LAST REVISION BY WA BENFIELD.  MARCH 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C   NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX A IS STORED.
C   ANAME = MATRIX IDENTIFICATION. (A6 FORMAT)
C   V     = VECTOR WORK SPACE.
C   LV    = VECTOR WORK SPACE.
C   KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C   NTAPE = LOGICAL NUMBER OF TAPE ON WHICH MATRIX A IS TO BE WRITTEN.
C
C      NERROR  EXPLANATION
C   1 = DIMENSION SIZE EXCEEDED (KV).
C
C      INTERNAL VARIABLES THAT ARE PUT ON TAPE (TRANSFERRED THRU COMMON).
C   RUNNO IS RUN NUMBER OF PROBLEM. (A6 FORMAT).
C   DATE IS DATE. (A6 FORMAT).  FOR EXAMPLE  15FE65
C
C   SEARCH TAPE FOR END OF WRITTEN DATA.
   10 READ (NTAPE) TAPEID,LN,IEOTCK
      IF (IEOTCK.EQ.3HEOT)GO TO 20
      READ (NTAPE)
      GO TO 10
C
C   END OF WRITTEN DATA HAS BEEN FOUND.
   20 BACKSPACE NTAPE
      REWIND NUTA
      CALL YINI   (NUTA,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
      NPART = MHEAD(3)
      NNZA = MHEAD(4)
      MCHECK(1) = MHEAD(5)
      MCHECK(2) = MHEAD(6)
      MSHAPE = MHEAD(7)
      IF (NPART.GT.0) GO TO 25
      WRITE (NTAPE) TAPEID,LN,BUF,IRUNO,ANAME,NRA,NCA,DATE,SPART,BUF,
     *              NONE,NONE,MSHAPE,(BUF,I=1,6)
```

```
      WRITE (NTAPE) BUF,BUF
      LN=LN+1
      GO TO 40
   25 DO 35 J=1,NPART
      CALL YINI   (NUTA,MHEAD,1,10)
      NNZP = MHEAD(1)
                                                              NERROR=1
      IF (NNZP.GT.KV) GO TO 999
      WRITE (NTAPE) TAPEID,LN,BUF,IRUNO,ANAME,NRA,NCA,DATE,SPART,NNZP,J,
     *              NPART,(MCHECK(I),I=1,2),MSHAPE,(BUF,I=1,4)
      IF (NNZP.GT.0) GO TO 30
      CALL YINI   (NUTA,MHEAD,10,10)
      CALL YINI   (NUTA,    V, 1, 1)
      MHEAD(10) = 0
      WRITE (NTAPE) BUF,BUF
      GO TO 35
   30 CALL YINI   (NUTA,LV,1,NNZP)
      CALL YIN    (NUTA,V,1,NNZP)
      WRITE (NTAPE) (LV(I),V(I),I=1,NNZP)
   35 LN=LN+1
   40 WRITE (NTAPE) TAPEID,LN,EOT,(BUF,I=1,16)
      BACKSPACE NTAPE
      RETURN
C
  999 CALL ZZECMB (6HYWTAPE,NERROR)
      END
```

```
      SUBROUTINE YZERLH (NUTAZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  ZERO THE LOWER HALF OF SPARSE MATRIX AZ.
C  CALLS FORMA SUBROUTINES YIN   ,YINI  ,YLORD ,YOUT  ,YOUTI ,YPART ,
C                              ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JUNE 1969.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTAZ = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX AZ IS STORED.
C  V     = VECTOR WORK SPACE.
C  LV    = VECTOR WORK SPACE.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = MATRIX NOT SQUARE.
C  2 = DIMENSION SIZE EXCEEDED (KV).
C
C  GET (A) HEADER INFORMATION.
      REWIND NUTAZ
      REWIND NUT1
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
                                                            NERROR=1
      IF (NRA.NE.NCA) GO TO 999
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      MCKORD = MHEAD(5)
      IASHAP = MHEAD(7)
      IF (NNZA .EQ. 0) RETURN
      IF (IASHAP.EQ.4HDIAG) RETURN
      IF (IASHAP.EQ.5HUPPER) RETURN
      MSHAPE = 5HUPPER
      IF (IASHAP.EQ.5HLOWER) MSHAPE=4HDIAG
      NNZZ=0
      NPARTZ=0
C
C  LOOP ON (A) PARTITIONS.
      DO 20 I=1,NPARTA
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NNZP = MHEAD(1)
                                                            NERROR=2
      IF (NNZP.GT.KV) GO TO 999
      CALL YINI   (NUTAZ,LV,1,NNZP)
      CALL YIN    (NUTAZ,V,1,NNZP)
C
      DO 10 J=1,NNZP
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (JA.LT.IA) V(J)=0.
```

```
   10 CONTINUE

      NNZQ=0
C
      DO 15 J=1,NNZP
      IF (V(J).EQ.0.) GO TO 15
      NNZQ=NNZQ+1
      V(NNZQ)=V(J)
      LV(NNZQ)=LV(J)
   15 CONTINUE
C
      IF (NNZQ.EQ.0) GO TO 20
      NNZZ=NNZZ+NNZQ
      MHEAD(1) = NNZQ
      CALL YOUTI  (NUT1,MHEAD,1,1)
      CALL YOUTI  (NUT1,LV,1,NNZQ)
      CALL YOUT   (NUT1,V,1,NNZQ)
      NPARTZ=NPARTZ+1
   20 CONTINUE
C
C  TRANSFER DATA FROM NUT1 TO NUTAZ.
      IF (NNZZ.EQ.NNZA) GO TO 40
      REWIND NUTAZ
      REWIND NUT1
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NPARTZ
      MHEAD(4) = NNZZ
      MHEAD(5) = MCKORD
      MHEAD(7) = MSHAPE
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
C
      DO 30 I=1,NPARTZ
      CALL YINI   (NUT1,MHEAD,1,1)
      NNZQ = MHEAD(1)
      CALL YINI   (NUT1,LV,1,NNZQ)
      CALL YIN    (NUT1,V,1,NNZQ)
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZQ)
      MHEAD(4) = 0
      MHEAD(5) = 0
      MHEAD(7) = 0
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      CALL YOUTI  (NUTAZ,LV,1,NNZQ)
   30 CALL YOUT   (NUTAZ,V,1,NNZQ)
C
   40 CALL YLORD  (NUTAZ,V,LV,KV,NUT1,NUT2)
      RETURN
C
  999 CALL ZZBOMB (6HYZERLH,NERROR)
      END
```

```
      SUBROUTINE YZERO (NUTA,NRA,NCA)
      DIMENSION MHEAD(10)
C
C  GENERATE A NULL SPARSE MATRIX A.
C  CALLS FORMA SUBROUTINES YOUT  ,YOUTI .
C  DEVELOPED BY R A PHILIPPUS.  OCTOBER 1969.
C  LAST REVISION BY JOHN ADMIRE  *NASA*  FEB 1974.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTA  = LOGICAL NUMBER OF UTILITY TAPE ON WHICH NULL MATRIX A IS
C           STORED.
C  NRA   = NUMBER OF ROWS IN A.
C  NCA   = NUMBER OF COLUMNS IN A.
C
      REWIND NUTA
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = 0
      MHEAD(4) = 0
      MHEAD(5) = 0
      MHEAD(6) = 0
      MHEAD(7) = 5HWHOLE
      MHEAD(8) = 0
      MHEAD(9) = 0
      MHEAD(10) = 0
      CALL YOUTI  (NUTA,MHEAD,1,10)
      MHEAD(1) = 0
      MHEAD(2) = 0
      MHEAD(7) = 0
      CALL YOUTI  (NUTA,MHEAD,1,10)
      RETURN
      END
```

```
      SUBROUTINE YZERUH (NUTAZ,V,LV,KV,NUT1,NUT2)
      DIMENSION V(1),LV(1),MHEAD(10)
      DATA NIT,NOT/5,6/
C
C  ZERO THE UPPER HALF OF SPARSE MATRIX AZ.
C  CALLS FORMA SUBROUTINES YIN    ,YINI   ,YLORD ,YOUT   ,YOUTI ,YPART ,
C                                ZZBOMB.
C  DEVELOPED BY R A PHILIPPUS.  JUNE 1969.
C  LAST REVISION BY WA BENFIELD FOR NASA.  MAY 1976.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C  NUTAZ = LOGICAL NUMBER OF UTILITY TAPE ON WHICH MATRIX AZ IS STORED.
C  V     = WORK VECTOR.
C  LV    = WORK VECTOR.
C  KV    = DIMENSION SIZE OF V,LV IN CALLING PROGRAM.
C  NUT1  = LOGICAL NUMBER OF UTILITY TAPE.
C  NUT2  = LOGICAL NUMBER OF UTILITY TAPE.
C
C     NERROR  EXPLANATION
C  1 = MATRIX NOT SQUARE.
C  2 = DIMENSION SIZE EXCEEDED (KV).
C
C  GET (A) HEADER INFORMATION.
      REWIND NUTAZ
      REWIND NUT1
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NRA = MHEAD(1)
      NCA = MHEAD(2)
                                                               NERROR=1
      IF (NRA.NE.NCA) GO TO 999
      NPARTA = MHEAD(3)
      NNZA = MHEAD(4)
      MCKORD = MHEAD(5)
      IASHAP = MHEAD(7)
      IF (NNZA .EQ. 0) RETURN
      IF (IASHAP.EQ.4HDIAG) RETURN
      IF (IASHAP.EQ.5HLOWER) RETURN
      MSHAPE = 5HLOWER
      IF (IASHAP.EQ.5HUPPER) MSHAPE = 4HDIAG
      NNZZ=0
      NPARTZ=0
C
C  LOOP ON (A) PARTITIONS.
      DO 20 I=1,NPARTA
      CALL YINI   (NUTAZ,MHEAD,1,10)
      NNZP = MHEAD(1)
                                                               NERROR=2
      IF (NNZP.GT.KV) GO TO 999
      CALL YINI   (NUTAZ,LV,1,NNZP)
      CALL YIN    (NUTAZ,V,1,NNZP)
C
      DO 10 J=1,NNZP
      IA=LV(J)/100000
      JA=LV(J)-100000*IA
      IF (JA.GT.IA) V(J)=0.
```

```
   10 CONTINUE

      NNZQ=0
      DO 15 J=1,NNZP
      IF (V(J).EQ.0.) GO TO 15
      NNZQ=NNZQ+1
      V(NNZQ)=V(J)
      LV(NNZQ)=LV(J)
   15 CONTINUE
C
      IF (NNZQ.EQ.0) GO TO 20
      NNZZ=NNZZ+NNZQ
      MHEAD(1) = NNZQ
      CALL YOUTI  (NUT1,MHEAD,1,1)
      CALL YOUTI  (NUT1,LV,1,NNZQ)
      CALL YOUT   (NUT1,V,1,NNZQ)
      NPARTZ=NPARTZ+1
   20 CONTINUE
C
C  TRANSFER DATA FROM NUT1 TO NUTAZ.
      IF (NNZZ.EQ.NNZA) GO TO 40
      REWIND NUTAZ
      REWIND NUT1
      MHEAD(1) = NRA
      MHEAD(2) = NCA
      MHEAD(3) = NPARTZ
      MHEAD(4) = NNZZ
      MHEAD(5) = MCKORD
      MHEAD(6) = 0
      MHEAD(7) = MSHAPE
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
C
      DO 30 I=1,NPARTZ
      CALL YINI   (NUT1,MHEAD,1,1)
      NNZQ = MHEAD(1)
      CALL YINI   (NUT1,LV,1,NNZQ)
      CALL YIN    (NUT1,V,1,NNZQ)
      MHEAD(2) = LV(1)
      MHEAD(3) = LV(NNZQ)
      MHEAD(4) = 0
      MHEAD(5) = 0
      MHEAD(7) = 0
      CALL YOUTI  (NUTAZ,MHEAD,1,10)
      CALL YOUTI  (NUTAZ,LV,1,NNZQ)
   30 CALL YOUT   (NUTAZ,V,1,NNZQ)
C
   40 CALL YLORD  (NUTAZ,V,LV,KV,NUT1,NUT2)
      RETURN
C
  999 CALL ZZBOMB (6HYZERUH,NERROR)
      END
```