

NCR 76-217  
Contract NAS8-31375

# Listings, Dense FORMA Subroutines

May 1976

Volume IIA

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

(NAS4-31-14431B) EXPANSION AND IMPROVEMENT  
OF THE FORMA SYSTEM FOR RESPONSE AND LOAD  
ANALYSIS. VOLUME 2A: LISTINGS, DENSE FORMA  
SUBROUTINES (MARTIN MARIETTA CORP.) 289 P  
UNCLAS  
N76-25591  
GPO: 1976 O-39-43036

**MARTIN MARIETTA**

MGR-76-217

Contract NAS8-31376

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

Volume IIA - Listings, Dense FORMA Subroutines

May 1976

Author:

Richard L. Wohlen

Approved by:



Richard L. Wohlen  
Program Manager

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

MARTIN MARLETTA 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

	<u>Page</u>
Foreword . . . . .	ii
Contents . . . . .	iii
Abstract . . . . .	iv
Acknowledgements . . . . .	v
I. Introduction . . . . .	I-1
II. Subroutine Listings (Subroutine Names in Alphabetical Order, Numbers Coming Before Letters) . . . . .	II-1

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 dense FORMA subroutine is given in this volume to remove the "black-box" aura of the subroutines so that the analyst may better understand the detail operations of each subroutine.

The FORTRAN IV programming language is used throughout with the exception of MSFC UNIVAC 1108 systems subroutines used in FORMA subroutines START, PLOT1, PLOT2, PLOT3 and ZZBOMB.

II. SUBROUTINE LISTINGS

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



AABB  
-----

SUBROUTINE 'AABB (ALPHA,A,BETA,B,Z,NR,NC,KR)  
DIMENSION A(KR,1), B(KR,1), Z(KR,1)

C  
C MATRIX SUMMATION. (ALPHA \* A + BETA \* B = Z).  
C MATRICES A,Z OR B,Z MAY SHARE SAME CORE LOCATIONS.  
C CODED BY RL WOHLER. FEBRUARY 1965.  
C  
C SUBROUTINE ARGUMENTS  
C ALPHA = INPUT SCALAR.  
C A = INPUT MATRIX. SIZE(NR,NC).  
C BETA = INPUT SCALAR.  
C B = INPUT MATRIX. SIZE(NR,NC).  
C Z = OUTPUT RESULT MATRIX. SIZE(NR,NC).  
C NR = INPUT NUMBER OF ROWS IN MATRICES A,B,Z.  
C NC = INPUT NUMBER OF COLS IN MATRICES A,B,Z.  
C KR = INPUT ROW DIMENSION OF A,B,Z IN CALLING PROGRAM.  
C  
DO 10 I=1,NR  
DO 10 J=1,NC  
10 Z(I,J) = ALPHA\*A(I,J) + BETA\*B(I,J)  
RETURN  
END

```

SUBROUTINE AB1(A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)
DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.0/

```

```

C
C AB1 PERFORMS THE MATRIX OPERATION (Z)=(A)*(B) .
C AB1 CAN ALSO PERFORM THE OPERATIONS
C (Z)=(A)*(A) BY CALL AB1(A,A,Z,--ETC--)
C (A)=(A)*(B) BY CALL AB1(A,B,A,--ETC--)
C

```

```

C IF NRA IS NEGATIVE AND ABS(NRA) IS EQUAL TO NCB
C A SQUARE, SYMMETRIC (Z) IS CALCULATED.
C

```

```

C MAXIMUM SIZE NCA=500
C

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C

```

```

C FORMA SUBROUTINE ZZBOMB IS CALLED .
C CODED BY JOHN ADMIRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLN. MARCH 1976.
C

```

ARGUMENTS

```

C A - INPUT MATRIX (A) SIZE(NRA BY NCA)
C B - INPUT MATRIX (B) SIZE(NCA BY NCB)
C Z - OUTPUT MATRIX (Z) SIZE(NRA BY NCB)
C NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)
C NCA - INPUT NUMBER OF COLUMNS IN (A)
C NCB - INPUT NUMBER OF COLUMNS IN (B)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM
C

```

NERROR EXPLANATIONS

```

C 1 = SIZE EXCEEDANCE.
C 2 = NON-SQUARE RESULT ASKED FOR.
C

```

```

NR=IABS(NRA)

```

NERROR = 1

```

IF(NCA .GT. 500 .OR. NR .GT. KRA .OR. NCA .GT. KRB
* .OR. NR .GT. KRZ) GO TO 999
IF(NRA .GT. 0) GO TO 40

```

NERROR = 2

```

IF(NR .NE. NCB) GO TO 999
DO 30 I=1,NR
DO 10 K=1,NCA
10 V(K)=A(I,K)
DO 30 J=1,NCB
SUM=ZERO
DO 20 K=1,NCA
SS=V(K)*B(K,J)
20 SUM=SUM+SS
30 Z(I,J)=SUM
DO 33 I=1,NR
DO 33 J=1,NCB

```

```
33 Z(J,I)=Z(I,J)
RETURN
40 DO 70 I=1,NRA
DO 50 K=1,NCA
50 V(K)=A(I,K)
DO 70 J=1,NCB
SUM=ZERO
DO 60 K=1,NCA
SS=V(K)*B(K,J)
60 SUM=SUM+SS
70 Z(I,J)=SUM
RETURN
999 CALL ZZBOMB(6HAB1 ,NERROR)
END
```

```

SUBROUTINE AB2(A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)
DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.D/

```

```

AB2 PERFORMS THE MATRIX OPERATION (Z)=(A)*(B) .
AB2 CAN ALSO PERFORM THE OPERATIONS
(Z)=(A)*(A) BY CALL AB2(A,A,Z,--ETC--)
(B)=(A)*(B) BY CALL AB2(A,B,B,--ETC--)

```

```

IF (B) AND (Z) DO NOT SHARE THE SAME STORAGE
IT WOULD BE MORE EFFICIENT TO USE SUBROUTINE
AB1 TO PERFORM THIS OPERATION.

```

```

IF NRA IS NEGATIVE AND ABS(NRA) IS EQUAL TO NCB
A SQUARE, SYMMETRIC (Z) IS CALCULATED.

```

```

MAXIMUM SIZE NCA=500

```

```

INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

```

```

FORMA SUBROUTINE ZZBOMB IS CALLED.
CODED BY JOHN ADMIRE *NASA* JULY 1972 .
LAST REVISION BY RL WOHLER. MARCH 1976.

```

ARGUMENTS

```

A - INPUT MATRIX (A) SIZE(NRA BY NCA)
B - INPUT MATRIX (B) SIZE(NCA BY NCB)
Z - OUTPUT MATRIX (Z) SIZE(NRA BY NCB)
NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)
NCA - INPUT NUMBER OF COLUMNS IN (A)
NCB - INPUT NUMBER OF COLUMNS IN (B)
KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

NERROR EXPLANATIONS

```

1 = SIZE EXCEEDANCE.
2 = NON-SQUARE RESULT ASKED FOR.

```

```

NR=IABS(NRA)

```

NERROR = 1

```

IF(NCA .GT. 500 .OR. NCA .GT. KRB .OR. NR .GT. KRA
* .OR. NR .GT. KRZ) GO TO 999
IF(NRA .GT. 0) GO TO 40

```

NERROR = 2

```

IF(NR .NE. NCB) GO TO 999
DO 30 J=1,NCB
DO 10 K=1,NCA
10 V(K)=B(K,J)
DO 30 I=1,J
SUM=ZERO
DO 20 K=1,NCA
SS=A(I,K)*V(K)

```

```
20 SUM=SUM+SS
30 Z(I,J)=SUM
   DO 33 I=1,NR
   DO 33 J=I,NR
33 Z(J,I)=Z(I,J)
   RETURN
40 DO 70 J=1,NCR
   DO 50 K=1,NCA
50 V(K)=B(K,J)
   DO 70 I=1,NRA
   SUM=ZERO
   DO 60 K=1,NCA
   SS=A(I,K)*V(K)
60 SUM=SUM+SS
70 Z(I,J)=SUM
   RETURN
999 CALL ZZBOCB(6HAB2 ,NERROR)
   END
```

```

SUBROUTINE ABC1(A,B,C,Z,NRA,NCA,NCB,KRA,KRB,KRC,KRZ)
DIMENSION A(KRA,1),B(KRB,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKVI / V(500)
DOUBLE PRECISION SUM,SS

```

ABC1 PERFORMS THE MATRIX OPERATION  $(Z)=(A)*(B)+(C)$  .

ABC1 CAN ALSO PERFORM THE OPERATIONS

$(Z)=(A)*(B)+(A)$  BY CALL ABC1(A,B,A,Z,--ETC--)

$(Z)=(A)*(B)+(B)$  BY CALL ABC1(A,B,B,Z,--ETC--)

$(Z)=(A)*(A)+(C)$  BY CALL ABC1(A,A,C,Z,--ETC--)

$(Z)=(A)*(A)+(A)$  BY CALL ABC1(A,A,A,Z,--ETC--)

$(A)=(A)*(B)+(C)$  BY CALL ABC1(A,B,C,A,--ETC--)

$(C)=(A)*(B)+(C)$  BY CALL ABC1(A,B,C,C,--ETC--)

$(A)=(A)*(B)+(A)$  BY CALL ABC1(A,B,A,A,--ETC--)

IF NRA IS NEGATIVE AND ABS(NRA) IS EQUAL TO NCB  
A SQUARE, SYMMETRIC (Z) IS CALCULATED.

MAXIMUM SIZE NCA=500

INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

FORMA SUBROUTINE ZZBOMB IS CALLED.

CODED BY JOHN ADMIRE \*NASA\* JULY 1972 .

LAST REVISION BY RL WOHLER. MARCH 1976.

#### ARGUMENTS

A - INPUT MATRIX (A) SIZE(NRA BY NCA)  
B - INPUT MATRIX (B) SIZE(NCA BY NCB)  
C - INPUT MATRIX (C) SIZE(NRA BY NCB)  
Z - OUTPUT MATRIX (Z) SIZE(NRA BY NCB)  
NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)  
NCA - INPUT NUMBER OF COLUMNS IN (A)  
NCB - INPUT NUMBER OF COLUMNS IN (B)  
KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM  
KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM  
KRC - INPUT ROW DIMENSION OF (C) IN IN CALLING PROGRAM  
KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

#### NERROR EXPLANATIONS

1 = SIZE EXCEEDANCE.  
2 = NON-SQUARE RESULT ASKED FOR.

NR=IABS(NRA)

NERROR = 1

IF(NCA .GT. 500 .OR. NR .GT. KRA .OR. NR .GT. KRC  
\* .OR. NCA .GT. KRB .OR. NR .GT. KRZ) GO TO 999  
IF(NRA .GT. 0) GO TO 40

NERROR = 2

IF(NR .NE. NCB) GO TO 999

```

DO 30 I=1,NR
DO 10 K=1,NCA
10 V(K)=A(I,K)
DO 30 J=1,NCB
SUM=C(I,J)

```

```
      DO 20 K=1,NCA
      SS=V(K)*B(K,J)
20    SUM=SUM+SS
30    Z(I,J)=SUM
      DO 33 I=1,NR
      DO 33 J=I,NR
33    Z(J,I)=Z(I,J)
      RETURN
40    DO 70 I=1,NRA
      DO 50 K=1,NCA
50    V(K)=A(I,K)
      DO 70 J=1,NCB
      SUM=C(I,J)
      DO 60 K=1,NCA
      SS=V(K)*B(K,J)
60    SUM=SUM+SS
70    Z(I,J)=SUM
      RETURN
999  CALL ZZBOMB(6HABC1 ,NERROR)
      END
```

```

SUBROUTINE ABC2(A,B,C,Z,NRA,NCA,NCB,KRA,KRB,KRC,KRZ)
DIMENSION A(KRA,1),B(KRB,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS

```

ABC2 PERFORMS THE MATRIX OPERATION  $(Z)=(A)*(B)+(C)$  .

ABC2 CAN ALSO PERFORM THE OPERATIONS

$(Z)=(A)*(B)+(B)$  BY CALL ABC2(A,B,B,Z,--ETC--)

$(Z)=(A)*(B)+(A)$  BY CALL ABC2(A,B,A,Z,--ETC--)

$(Z)=(A)*(A)+(C)$  BY CALL ABC2(A,A,C,Z,--ETC--)

$(Z)=(A)*(A)+(A)$  BY CALL ABC2(A,A,A,Z,--ETC--)

$(B)=(A)*(B)+(C)$  BY CALL ABC2(A,B,C,B,--ETC--)

$(C)=(A)*(B)+(C)$  BY CALL ABC2(A,B,C,C,--ETC--)

$(B)=(A)*(B)+(B)$  BY CALL ABC2(A,B,B,B,--ETC--)

IF (B) DOES NOT SHARE STORAGE WITH (C) OR (Z) IT WOULD BE MORE EFFICIENT TO USE ABC1 TO PERFORM THIS OPERATION .

IF NRA IS NEGATIVE AND ABS(NRA) IS EQUAL TO NCB A SQUARE, SYMMETRIC (Z) IS CALCULATED.

MAXIMUM SIZE NCA=500

INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

FORMA SUBROUTINE ZZBOMB IS CALLED .

CODED BY JOHN ADMIRE \*NASA\* JULY 1972 .

LAST REVISION BY RL WOHLN. MARCH 1976.

#### ARGUMENTS

A - INPUT MATRIX (A) SIZE(NRA BY NCA)  
 B - INPUT MATRIX (B) SIZE(NCA BY NCB)  
 C - INPUT MATPIX (C) SIZE(NRA BY NCB)  
 Z - OUTPUT MATRIX (Z) SIZE(NRA BY NCB)  
 NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)  
 NCA - INPUT NUMBER OF COLUMNS IN (A)  
 NCB - INPUT NUMBER OF COLUMNS IN (B)  
 KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM  
 KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM  
 KRC - INPUT ROW DIMENSION OF (C) IN CALLING PROGRAM  
 KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

#### NERROR EXPLANATIONS

1 = SIZE EXCEEDANCE .

2 = NON-SQUARE RESULT ASKED FOR.

NR=IABS(NRA)

NERROR = 1

IF(NCA .GT. 500 .OR. NCA .GT. KRB .OR. NR .GT. KRC  
 \* .OR. NR .GT. KRA .OR. NR .GT. KRZ) GO TO 999

IF(NRA .GT. 0) GO TO 40

NERROR = 2

IF(NR .NE. NCB) GO TO 999

DO 30 J=1,NCB

DO 10 K=1,NCA



```
10 V(K)=B(K,J)
   DO 30 I=1,J
   SUM=C(I,J)
   DO 20 K=1,NCA
   SS=A(I,K)*V(K)
20 SUM=SUM+SS
30 Z(I,J)=SUM
   DO 33 I=1,NR
   DO 33 J=I,NR
33 Z(J,I)=Z(I,J)
   RETURN
40 DO 70 J=1,NCB
   DO 50 K=1,NCA
50 V(K)=B(K,J)
   DO 70 I=1,NRA
   SUM=C(I,J)
   DO 60 K=1,NCA
   SS=A(I,K)*V(K)
60 SUM=SUM+SS
70 Z(I,J)=SUM
   RETURN
999 CALL ZZBOMB(6HABC2 ,NERROR)
   END
```

```

SUBROUTINE ALOD1 (PP,DIST,CONC,CONVRT, Z, NPP,ND,NC, KD,KC)
DIMENSION PP(1),DIST(KD,1),CONC(KC,1),Z(1)
COMMON /LLINE /NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C REPLACE DISTRIBUTED AND CONCENTRATED LATERAL FORCES ON A BEAM
C WITH REPRESENTATIVE CONCENTRATED FORCES AT THE PANEL POINTS.
C THIS ENTAILS BEAMING BAY FORCE TO ADJACENT PANEL POINTS.
C THE DISTRIBUTED FORCE MAY NOT EXCEED THE PANEL POINT LIMITS.
C THE CONCENTRATED FORCES MAY BE OUTSIDE THE PANEL POINT LIMITS.
C OPTION TO OMIT FORCE DATA BY ND OR NC EQUAL ZERO.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C CODED BY RL WOHLER. FEBRUARY 1970.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C PP = INPUT VECTOR OF PANEL POINTS. SIZE(NPP).
C DIST = INPUT MATRIX OF DISTRIBUTED FORCE STRAIGHT LINE
C SEGMENT DATA. SIZE(ND,4).
C COL 1 = X AT SEGMENT END 1.
C COL 2 = X AT SEGMENT END 2.
C COL 3 = FORCE AT SEGMENT END 1.
C COL 4 = FORCE AT SEGMENT END 2.
C CONC = INPUT MATRIX OF CONCENTRATED FORCE DATA. SIZE(NC,2).
C COL 1 = X COORDINATE.
C COL 2 = FORCE.
C CONVRT = INPUT CONVERSION SCALAR BY WHICH COLS 3,4 OF DIST AND
C COL 2 OF CONC WILL BE MULTIPLIED.
C Z = OUTPUT VECTOR OF CONCENTRATED PANEL POINT FORCES. SIZE(NPP).
C NPP = INPUT NUMBER OF PANEL POINTS. SIZE OF VECTORS PP,Z.
C ND = INPUT NUMBER OF SEGMENTS (ROWS) IN DIST. CAN BE ZERO.
C NC = INPUT NUMBER OF FORCES (ROWS) IN CONC. CAN BE ZERO.
C KD = INPUT ROW DIMENSION OF DIST IN CALLING PROGRAM.
C KC = INPUT ROW DIMENSION OF CONC IN CALLING PROGRAM.

```

```

C ERROR EXPLANATION
C 1 = LESS THAN 2 PANEL POINTS.
C 2 = PANEL POINTS NOT IN INCREASING ORDER.
C 3 = INCORRECT DISTRIBUTED DATA.

```

```

2001 FORMAT (3//,30X,31H SUBROUTINE ALOD1 USES CONVRT = E15.8, /
* 57X,33H AND COMPUTES THE TOTAL PROPERTIES /
* 45X,16HLATERAL FORCE = E15.8, /
* 40X,21HCENTER OF PRESSURE = E15.8)

```

```

2002 FORMAT(/1X131(1H-))

```

```

C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER.
IF (NPP .LT. 2) GO TO 999
NERROR = 1
DO 5 K=2,NPP
5 IF (PP(K-1) .GE. PP(K)) GO TO 999
NERROR = 2

```

```

C INITIALIZE DATA.
DO 10 I=1,NPP

```

```
10 Z(I) = 0.0
   NBAYS = NPP-1
```

```
C
C BRANCH TO APPROPRIATE SECTION.
  IF(ND .EQ. 0) GO TO 100
```

```
C
C SOLVE FOR DISTRIBUTED INPUT.
```

```
DO 90 I=1,ND
  X1 = DIST(I,1)
  X2 = DIST(I,2)
  F1 = DIST(I,3)*CONVRT
  F2 = DIST(I,4)*CONVRT
```

NERROR = 3

```
  IF (X1 .LT. PP(1) .OR. X2 .GT. PP(NPP) .OR. X1 .GE. X2) GO TO 999
  DO 32 K=1,NBAYS
32 IF (X1 .LT. PP(K+1)) GO TO 34
34 XP = X1
   FP = F1
36 IF (X2 .LE. PP(K+1)) GO TO 38
   XQ = PP(K+1)
   FQ = F1 + (XQ-X1)*(F2-F1)/(X2-X1)
   GO TO 39
38 XQ = X2
   FQ = F2
39 BAYL = PP(K+1)-PP(K)
   SEGL = XQ-XP
   Z(K)  = Z(K)  + SEGL*(FP*(3.*(PP(K+1)-XP)-SEGL)
*           +FQ*(3.*(PP(K+1)-XP)-2.*SEGL))/(6.*BAYL)
   Z(K+1) = Z(K+1) + SEGL*(FP*(3.*(XP-PP(K))+SEGL)
*           +FQ*(3.*(XP-PP(K))+2.*SEGL))/(6.*BAYL)
   IF (X2 .LE. PP(K+1)) GO TO 90
   K = K+1
   XP = XQ
   FP = FQ
   GO TO 36
90 CONTINUE
```

```
C
C SOLVE FOR CONCENTRATED FORCE.
```

```
100 IF(NC .EQ. 0) GO TO 200
   DO 103 I=1,NC
   XC = CONC(I,1)
   FC = CONC(I,2)*CONVRT
   DO 101 K=1,NBAYS
101 IF (XC .LE. PP(K+1)) GO TO 102
   K = NBAYS
102 BAYL = PP(K+1) - PP(K)
   Z(K)  = Z(K)  + FC*(PP(K+1)-XC)/BAYL
103 Z(K+1) = Z(K+1) + FC*(XC-PP(K))/BAYL
```

```
C
C COMPUTE AND PRINT TOTAL PROPERTIES.
```

```
200 TF = 0.0
   TP = 0.0
   DO 201 I=1,NPP
   TF = TF + Z(I)
201 TP = TP + Z(I)*PP(I)
```

```
CP = TP/TF
IF(MINI .NE. 4HMINI) GO TO 300
IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 300
IF(NLINE +9 .GT. MAXLIN) GO TO 300
WRITE(NOT,2002)
NLINE=NLINE+2
GO TO 310
300 CALL PAGEHD
310 WRITE(NOT,2001) CONVRT,TF,CP
NLINE=NLINE+7
RETURN
```

```
C
999 CALL ZZBOMB (6HALOD1 ,NERRDR)
END
```

```

SUBROUTINE ALOD2 (PP,DIST,CONC,CONVRT, Z, NPP,ND,NC, KD,KC)
DIMENSION PP(1),DIST(KD,1),CONC(KC,1),Z(1)
COMMON /LLINE /NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C REPLACE DISTRIBUTED AND CONCENTRATED AXIAL FORCES ON A BEAM WITH
C REPRESENTATIVE CONCENTRATED FORCES AT THE PANEL POINTS.
C THIS ENTAILS PLACING BAY FORCE AT AFT (+X IS AFT) PANEL POINT OF BAY.
C THE DISTRIBUTED FORCE MAY NOT EXCEED THE PANEL POINT LIMITS.
C THE CONCENTRATED FORCES MAY BE OUTSIDE THE PANEL POINT LIMITS.
C OPTION TO OMIT FORCE DATA BY ND OR NC EQUAL ZERO.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C CODED BY RL WOHLER. FEBRUARY 1970.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C PP = INPUT VECTOR OF PANEL POINTS. SIZE(NPP).
C DIST = INPUT MATRIX OF DISTRIBUTED FORCE STRAIGHT LINE
C SEGMENT DATA. SIZE(ND,4).
C COL 1 = X AT SEGMENT END 1.
C COL 2 = X AT SEGMENT END 2.
C COL 3 = FORCE AT SEGMENT END 1.
C COL 4 = FORCE AT SEGMENT END 2.
C CONC = INPUT MATRIX OF CONCENTRATED FORCE DATA. SIZE(NC,2).
C COL 1 = X COORDINATE.
C COL 2 = FORCE.
C CONVRT = INPUT CONVERSION SCALAR BY WHICH COLS 3,4 OF DIST AND
C COL 2 OF CONC WILL BE MULTIPLIED.
C Z = OUTPUT VECTOR OF CONCENTRATED PANEL POINT FORCES. SIZE(NPP).
C NPP = INPUT NUMBER OF PANEL POINTS. SIZE OF VECTORS PP,Z.
C ND = INPUT NUMBER OF SEGMENTS (ROWS) IN DIST. CAN BE ZERO.
C NC = INPUT NUMBER OF FORCES (ROWS) IN CONC. CAN BE ZERO.
C KD = INPUT ROW DIMENSION OF DIST IN CALLING PROGRAM.
C KC = INPUT ROW DIMENSION OF CONC IN CALLING PROGRAM.
C CONVRT = INPUT CONVERSION SCALAR BY WHICH COLS 3,4 OF DIST AND

```

```

C
C ERROR EXPLANATION
C 1 = LESS THAN 2 PANEL POINTS.
C 2 = PANEL POINTS NOT IN INCREASING ORDER.
C 3 = INCORRECT DISTRIBUTED DATA.

```

```

2001 FORMAT ( 3(/),30X,31H SUBROUTINE ALOD2 USES CONVRT = E15.8, /
* 30X,37HAND COMPUTES THE TOTAL AXIAL FORCE = E15.8)
2002 FORMAT(/1X131(1H-))

```

```

C
C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER.

```

```

IF (NPP .LT. 2) GO TO 999

```

NERROR = 1

```

DO 5 K=2,NPP
5 IF (PP(K-1) .GE. PP(K)) GO TO 999

```

NERROR = 2

```

INITIALIZE DATA.

```

```

DO 10 I=1,NPP
10 Z(I) = 0.0

```

NBAYS = NPP-1

C BRANCH TO APPROPRIATE SECTION.  
IF(ND .EQ. 0) GO TO 100

C SOLVE FOR DISTRIBUTED INPUT.

DO 90 I=1,ND  
X1 = DIST(I,1)  
X2 = DIST(I,2)  
F1 = DIST(I,3)\*CONVRT  
F2 = DIST(I,4)\*CONVRT

NERROR = 3

IF (X1 .LT. PP(1) .OR. X2 .GT. PP(NPP) .OR. X1 .GE. X2) GO TO 999  
DO 32 K=1,NBAYS

32 IF (X1 .LT. PP(K+1)) GO TO 34

34 XP = X1  
FP = F1

36 IF (X2 .LE. PP(K+1)) GO TO 38  
XQ = PP(K+1)  
FQ = F1 + (XQ-X1)\*(F2-F1)/(X2-X1)  
GO TO 39

38 XQ = X2  
FQ = F2

39 Z(K+1) = Z(K+1) + .5\*(FP+FQ)\*(XQ-XP)  
IF (X2 .LE. PP(K+1)) GO TO 90  
K = K+1  
XP = XQ  
FP = FQ  
GO TO 36

90 CONTINUE

C SOLVE FOR CONCENTRATED FORCE.

100 IF(NC .EQ. 0) GO TO 200  
DO 103 I=1,NC  
XC = CONC(I,1)  
FC = CONC(I,2)\*CONVRT  
IF (XC .LE. PP(1)) Z(1) = Z(1)+FC  
IF (XC .LE. PP(1)) GO TO 103  
DO 101 K=1,NBAYS

101 IF (XC .LE. PP(K+1)) GO TO 102  
K = NBAYS

102 Z(K+1) = Z(K+1) + FC  
103 CONTINUE

C COMPUTE AND PRINT TOTAL PROPERTIES.

200 TF = 0.0  
DO 201 I=1,NPP  
201 TF = TF + Z(I)  
IF(MINI .NE. 4HMINI) GO TO 300  
IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 300  
IF(NLINE +7 .GT. MAXLIN) GO TO 300  
WRITE(ND,2002)  
NLINE=NLINE+2  
GO TO 310  
300 CALL PAGEHD

```
310 WRITE(NDT,2001) CONVRT,TF  
    NLINE=NLINE+5  
    RETURN
```

C

```
999 CALL ZZBOMB (6HALOD2 ,NERROR)  
    END
```

ALPHAA

---

SUBROUTINE ALPHAA (ALPHA,A,Z,NR,NC,KR)  
DIMENSION A(KR,1), Z(KR,1)

C  
C SCALAR ALPHA TIMES MATRIX A. (ALPHA \* A = Z).  
C MATRICES A,Z MAY SHARE SAME CORE LOCATIONS.  
C CODED BY RL WOHLER. FEBRUARY 1965.  
C  
C SUBROUTINE ARGUMENTS  
C ALPHA = INPUT SCALAR.  
C A = INPUT MATRIX. SIZE(NR,NC).  
C Z = OUTPUT RESULT MATRIX. SIZE(NR,NC).  
C NR = INPUT NUMBER OF ROWS IN MATRICES A,Z.  
C NC = INPUT NUMBER OF COLS IN MATRICES A,Z.  
C KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.  
C  
DO 10 I=1,NR  
DO 10 J=1,NC  
10 Z(I,J) = ALPHA \* A(I,J)  
RETURN  
END



ASSEM

SUBROUTINE ASSEM (A,IRZ,JCZ,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)  
DIMENSION A(KRA,1), Z(KRZ,1)

C MATRIX ASSEMBLY. (MATRIX A INTO MATRIX Z ).  
C BE SURE MATRIX Z IS DEFINED BEFORE CALLING THIS SUBROUTINE. FOR  
C EXAMPLE, CALL ZERO TO CLEAR MATRIX Z.  
C CALLS FORMA SUBROUTINE ZZBOMB.  
C CODED BY RL WOHLER. FEB 1965.  
C LAST REVISION BY RL WOHLER. MARCH 1976.

C SUBROUTINE ARGUMENTS

C A = INPUT MATRIX. SIZE(NRA,NCA).  
C IRZ = INPUT ROW NUMBER IN MATRIX Z OF FIRST ROW OF MATRIX A.  
C JCZ = INPUT COL NUMBER IN MATRIX Z OF FIRST COL OF MATRIX A.  
C Z = OUTPUT RESULT MATRIX. SIZE(NRZ,NCZ).  
C NRA = INPUT NUMBER OF ROWS OF MATRIX A.  
C NCA = INPUT NUMBER OF COLS OF MATRIX A.  
C NRZ = INPUT NUMBER OF ROWS OF MATRIX Z.  
C NCZ = INPUT NUMBER OF COLS OF MATRIX Z.  
C KRA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.  
C KRZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.

C NERROR EXPLANATION

C 1 = MATRIX A EXCEEDS MATRIX Z - ROWS.  
C 2 = MATRIX A EXCEEDS MATRIX Z - COLUMNS.

IF ((IRZ-1+NRA) .GT. NRZ) GO TO 999

NERROR = 1

IF ((JCZ-1+NCA) .GT. NCZ) GO TO 999

NERROR = 2

DO 10 IA=1,NRA  
IZ = IA + IRZ - 1  
DO 10 JA=1,NCA  
JZ = JA + JCZ - 1  
10 Z(IZ,JZ) = A(IA,JA)  
RETURN

999 CALL ZZBOMB (6HASSEM ,NERROR)  
END

ATI

SUBROUTINE ATI(A,Z,NR,NC,KRA,KRZ)  
 DIMENSION A(KRA,1),Z(KRZ,1)  
 COMMON / LWRKV1 / V(500)

ATI PERFORMS THE OPERATION (Z)=((A)TRANPOSE)  
 ATI CAN ALSO PERFORM THE OPERATION  
 (A)=((A)TRANPOSE) BY CALL ATI(A,A,---ETC---)

MAXIMUM SIZE NC=500

FORMA SUBROUTINE ZZBOMB IS CALLED  
 CODED BY JOHN ADMIRE \*NASA\* JULY 1972 .

ARGUMENTS

A - INPUT MATRIX (A) SIZE(NR BY NC)  
 Z - OUTPUT MATRIX (Z) SIZE(NC BY NR)  
 NR - INPUT NUMBER OF ROWS IN (A)  
 NC - INPUT NUMBER OF COLUMNS IN (A)  
 KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM  
 KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

NERROR=1

IF(NC .GT. 500 .OR. NC .GT. KRZ .OR. NR .GT. KRA) GO TO 999  
 N=NR

IF(NC .LT. NR) N=NC

DO 40 K=1,N

DO 10 J=K,NC

10 V(J)=A(K,J)

DO 20 I=K,NR

20 Z(K,I)=A(I,K)

DO 30 J=K,NC

30 Z(J,K)=V(J)

40 CONTINUE

RETURN

999 CALL ZZBOMB(6HATI ,NERROR)

END

```

SUBROUTINE ATB1(A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)
DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.0/

```

```

C
C ATB1 PERFORMS THE OPERATION (Z)=((A)TRANPOSE)*(B) .
C ATB1 CAN ALSO BE USED TO PERFORM THE OPERATIONS
C (A)=((A)TRANPOSE)*(B) BY CALL ATB1(A,B,A,--ETC--)
C (Z)=((A)TRANPOSE)*(A) BY CALL ATB1(A,A,Z,--ETC--) .
C

```

```

C IF NRA IS NEGATIVE AND NCA IS EQUAL TO NCB
C A SQUARE, SYMMETRIC (Z) IS CALCULATED.
C

```

```

C IF (A) DOES NOT SHARE STORAGE WITH (Z) IT WOULD
C BE MORE EFFICIENT TO USE ATB2 TO PERFORM THIS OPERATION .
C

```

```

C MAXIMUM SIZE NRA=500
C

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C

```

```

C FORMA SUBROUTINES ZZBOMB AND AT1 ARE CALLED.
C CODED BY JOHN ADMJRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLER. MARCH 1976.
C

```

ARGUMENTS

```

C A - INPUT MATRIX (A) SIZE(NRA BY NCA)
C B - INPUT MATRIX (B) SIZE(NRA BY NCB)
C Z - OUTPUT MATRIX (Z) SIZE(NCA BY NCB)
C NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)
C NCA - INPUT NUMBER OF COLUMNS IN (A)
C NCB - INPUT NUMBER OF COLUMNS IN (B)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM
C

```

NERROR EXPLANATIONS

```

C 1 = SIZE EXCEEDANCE.
C 2 = NON-SQUARE RESULT ASKED FOR.
C

```

```

NR=IABS(NRA)

```

NERROR = 1

```

IF(NR .GT. 500 .OR. NR .GT. KRA .OR. NR .GT. KRB
* .OR. NCB .GT. KRZ .OR. NCA .GT. KRZ) GO TO 999
IF(NRA .GT. 0) GO TO 40

```

NERROR = 2

```

IF(NCA .NE. NCB) GO TO 999
DO 30 I=1,NCA
DO 10 K=1,NR
10 V(K)=A(K,I)
DO 30 J=1,NCB
SUM=ZERO
DO 20 K=1,NR
SS=V(K)*B(K,J)
20 SUM=SUM+SS

```

```
30 Z(J,I)=SUM
   DO 33 I=1,NR
   DO 33 J=1,NR
33 Z(I,J)=Z(J,I)
   RETURN
40 DO 70 I=1,NCA
   DO 50 K=1,NRA
50 V(K)=A(K,I)
   DO 70 J=1,NCB
   SUM=ZERO
   DO 60 K=1,NRA
   SS=V(K)*B(K,J)
60 SUM=SUM+SS
70 Z(J,I)=SUM
   CALL AT1(Z,Z,NCB,NCA,KRZ,KRZ)
   RETURN
999 CALL ZZBOMB(6HATB1 ,NERROR)
   END
```

```

SUBROUTINE ATB2(A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)
DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.D/

```

```

C
C ATB2 PERFORMS THE OPERATION (Z)=((A)TRANPOSE)*(B)
C ATB2 CAN ALSO BE USED TO PERFORM THE OPERATIONS
C (B)=((A)*TRANPOSE)*(B) BY CALL ATB2(A,B,B,--ETC--)
C (Z)=((A)*TRANPOSE)*(A) BY CALL ATB2(A,A,Z,--ETC--)
C

```

```

C IF NRA IS NEGATIVE AND NCA IS EQUAL TO NCB
C A SQUARE, SYMMETRIC (Z) IS CALCULATED.
C

```

```

C MAXIMUM SIZE NRA=500
C

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C

```

```

C FORMA SUBROUTINE ZZROMB IS CALLED .
C CODED BY JOHN ADMIRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLN. MARCH 1976.
C

```

ARGUMENTS

```

C A - INPUT MATRIX (A) SIZE(NRA BY NCA)
C B - INPUT MATRIX (B) SIZE(NRA BY NCB)
C Z - OUTPUT MATRIX (Z) SIZE(NCA BY NCB)
C NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)
C NCA - INPUT NUMBER OF COLUMNS IN (A)
C NCB - INPUT NUMBER OF COLUMNS IN (B)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM
C

```

NERROR EXPLANATIONS

```

C 1 = SIZE EXCEEDANCE.
C 2 = NON-SQUARE RESULT ASKED FOR.
C

```

```

NR=IABS(NRA)

```

NERROR = 1

```

IF(NR .GT. 500 .OR. NR .GT. KRB .OR. NR .GT. KRA
* .OR. NCA .GT. KRZ) GO TO 999
IF(NRA .GT. 0) GO TO 40

```

NERROR = 2

```

IF(NCA .NE. NCB) GO TO 999
DO 30 J=1,NCB
DO 10 K=1,NR
10 V(K)=B(K,J)
DO 30 I=1,J
SUM=ZERO
DO 20 K=1,NR
SS=A(K,I)*V(K)
20 SUM=SUM+SS
30 Z(I,J)=SUM
DO 33 I=1,NR
DO 33 J=1,NR

```

```
33 Z(J,I)=Z(I,J)
   RETURN
40 DO 70 J=1,NCB
   DO 50 K=1,NRA
50 V(K)=B(K,J)
   DO 70 I=1,NCA
   SUM=ZERO
   DO 60 K=1,NRA
   SS=A(K,I)*V(K)
60 SUM=SUM+SS
70 Z(I,J)=SUM
   RETURN
999 CALL ZZBOMB(6HATB2 ,NERROR)
   END
```

```

SUBROUTINE ATBC1(A,B,C,Z,NRA,NCA,NCB,KRA,KRB,KRC,KRZ)
DIMENSION A(KRA,1),B(KRB,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKVI / V(500)
DOUBLE PRECISION SUM,SS

```

```

C
C ATBC1 PERFORMS THE OPERATION (Z)=((A)TRANPOSE)*(B)+(C) .
C ATBC1 CAN ALSO BE USED TO PERFORM THE OPERATIONS
C (Z)=((A)TRANPOSE)*(B)+(A) BY CALL ATBC1(A,B,A,Z,--ETC--)
C (Z)=((A)TRANPOSE)*(B)+(B) BY CALL ATBC1(A,B,B,Z,--ETC--)
C (Z)=((A)TRANPOSE)*(A)+(C) BY CALL ATBC1(A,A,C,Z,--ETC--)
C (Z)=((A)TRANPOSE)*(A)+(A) BY CALL ATBC1(A,A,A,Z,--ETC--)
C (A)=((A)TRANPOSE)*(B)+(C) BY CALL ATBC1(A,B,C,A,--ETC--)
C (A)=((A)TRANPOSE)*(B)+(B) BY CALL ATBC1(A,B,B,A,--ETC--)

```

```

C IF NRA IS NEGATIVE AND NCA IS EQUAL TO NCB
C A SQUARE, SYMMETRIC (Z) IS CALCULATED.

```

```

C IF (A) DOES NOT SHARE STORAGE WITH (Z) OR (C) IT WOULD
C BE MORE EFFICIENT TO USE ATBC2 TO PERFORM THIS OPERATION .

```

```

C MAXIMUM SIZE NRA=500

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

```

```

C FORMA SUBROUTINES ZZBOMB AND AT1 ARE CALLED .
C CODED BY JOHN ADMIRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

ARGUMENTS

```

C A - INPUT MATRIX (A) SIZE(NRA BY NCA)
C B - INPUT MATRIX (B) SIZE(NPA BY NCB)
C C - INPUT MATRIX (C) SIZE(NCA BY NCB)
C Z - OUTPUT MATRIX (Z) SIZE(NCA BY NCB)
C NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)
C NCA - INPUT NUMBER OF COLUMNS IN (A)
C NCB - INPUT NUMBER OF COLUMNS IN (B)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
C KRC - INPUT ROW DIMENSION OF (C) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

NERROR EXPLANATIONS

```

C 1 = SIZE EXCEEDANCE.
C 2 = NON-SQUARE RESULT ASKED FOR.

```

```

C NR=IABS(NRA)

```

NERROR = 1

```

C IF(NR .GT. 500 .OR. NR .GT. KRA .OR. NCA .GT. KRC
C * .OR. NR .GT. KRB .OR. NCB .GT. KRZ .OR. NCA .GT. KRZ) GO TO 999
C IF(NRA .GT. 0) GO TO 40

```

NERROR = 2

```

C IF(NCA .NE. NCB) GO TO 999

```

```

C DO 30 I=1,NCA

```

```

C DO 10 K=1,NR

```

```

C 10 V(K)=A(K,I)

```

```
DO 30 J=1,NCB
SUM=C(I,J)
DO 20 K=1,NR
SS=V(K)*B(K,J)
20 SUM=SUM+SS
30 Z(J,I)=SUM
DO 33 I=1,NR
DO 33 J=I,NR
33 Z(I,J)=Z(J,I)
RETURN
40 DO 70 I=1,NCA
DO 50 K=1,NRA
50 V(K)=A(K,I)
DO 70 J=1,NCB
SUM=C(I,J)
DO 60 K=1,NRA
SS=V(K)*B(K,J)
60 SUM=SUM+SS
70 Z(J,I)=SUM
CALL AT1(Z,Z,NCB,NCA,KRZ,KRZ)
RETURN
999 CALL ZZBOMB(6HATBC1 ,NERROR)
END
```



```

SUBROUTINE ATBC2(A,B,C,Z,NRA,NCA,NCB,KRA,KRB,KRC,KRZ)
DIMENSION A(KRA,1),B(KRB,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS

```

```

C
C ATBC2 PERFORMS THE OPERATION (Z)=((A)TRANPOSE)*(B)+(C) .
C ATBC2 CAN ALSO BE USED TO PERFORM THE OPERATIONS
C (Z)=((A)TRANPOSE)*(B)+(A) BY CALL ATBC2(A,B,A,Z,--ETC--)
C (Z)=((A)TRANPOSE)*(B)+(B) BY CALL ATBC2(A,B,B,Z,--ETC--)
C (Z)=((A)TRANPOSE)*(A)+(C) BY CALL ATBC2(A,A,C,Z,--ETC--)
C (Z)=((A)TRANPOSE)*(A)+(A) BY CALL ATBC2(A,A,A,Z,--ETC--)
C (B)=((A)TRANPOSE)*(B)+(C) BY CALL ATBC2(A,B,C,B,--ETC--)
C (C)=((A)TRANPOSE)*(B)+(C) BY CALL ATBC2(A,B,C,C,--ETC--)
C (B)=((A)TRANPOSE)*(B)+(B) BY CALL ATBC2(A,B,B,B,--ETC--)

```

```

C IF NRA IS NEGATIVE AND NCA IS EQUAL TO NCB
C A SQUARE, SYMMETRIC (Z) IS CALCULATED.

```

```

C MAXIMUM SIZE NRA=500

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

```

```

C FORMA SUBROUTINE ZZEOMB IS CALLED.
C CODED BY JOHN ADMIRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

#### ARGUMENTS

```

C A - INPUT MATRIX (A) SIZE(NRA BY NCA)
C B - INPUT MATRIX (B) SIZE(NRA BY NCB)
C C - INPUT MATRIX (C) SIZE(NCA BY NCB)
C Z - OUTPUT MATRIX (Z) SIZE(NCA BY NCB)
C NRA - INPUT ABS(NRA) IS THE NUMBER OF ROWS IN (A)
C NCA - INPUT NUMBER OF COLUMNS IN (A)
C NCB - INPUT NUMBER OF COLUMNS IN (B)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
C KRC - INPUT ROW DIMENSION OF (C) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

#### NERROR EXPLANATIONS

```

C 1 = SIZE EXCEEDANCE.
C 2 = NON-SQUARE RESULT ASKED FOR.

```

```

C NR=IAES(NRA)

```

```

C NERROR = 1

```

```

C IF(NR .GT. 500 .OR. NR .GT. KRB .OR. NCA .GT. KRC
C * .OR. NR .GT. KRA .OR. NCA .GT. KRZ) GO TO 999
C IF(NRA .GT. 0) GO TO 40

```

```

C NERROR = 2

```

```

C IF(NCA .NE. NCB) GO TO 999

```

```

C DO 30 J=1,NCB

```

```

C DO 10 K=1,NR

```

```

C 10 V(K)=R(K,J)

```

```

C DO 30 I=1,J

```

```

C SUM=C(I,J)

```

```
      DO 20 K=1, NR
      SS=A(K, I)*V(K)
20    SUM=SUM+SS
30    Z(I, J)=SUM
      DO 33 I=1, NR
      DO 33 J=I, NR
33    Z(J, I)=Z(I, J)
      RETURN
40    DO 70 J=1, NCB
      DO 50 K=1, NRA
50    V(K)=B(K, J)
      DO 70 I=1, NCA
      SUM=C(I, J)
      DO 60 K=1, NRA
      SS=A(K, I)*V(K)
60    SUM=SUM+SS
70    Z(I, J)=SUM
      RETURN
999  CALL ZZBOMB(6HATBC2 ,NERROR)
      END
```

ATXBA1

```

SUBROUTINE ATXBA1 (AZ,B,NRB,NCB,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON /LWRKVI/W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/

```

```

C
C SPECIAL MATRIX MULTIPLICATION. A(TRANPOSE) * B = Z.
C Z WILL BE SYMMETRIC.
C USES TWO MATRIX SPACES. RESULT (Z) IS PLACED IN A.
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF A OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NCB = 500
C DEVELOPED BY RL WOHLN. SEPTEMBER 1972.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

SUBROUTINE ARGUMENTS

```

C AZ   = INPUT 1ST MATRIX. SIZE(NRB,NCB).
C      = OUTPUT RESULT MATRIX. SIZE(NCB,NCB).
C B    = INPUT 2ND MATRIX. SIZE(NRB,NCB).
C NRB  = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A(TRANS).
C NCB  = INPUT NUMBER OF COLS OF MATRIX B, ROWS OF MATRIX A(TRANS),
C      SIZE OF MATRIX Z(SQUARE). MAX=500.
C KAZ  = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB   = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.

```

NERROR EXPLANATION

```

C 1 = SIZE LIMITATION EXCEEDED.

```

NERROR = i

```

IF (NCB.GT.500 .OR. NCB.GT.KAZ .OR. NRB.GT.KB) GO TO 999

```

```

DO 40 I=1,NCB
DO 35 J=I,NCB
S = ZERO
DO 30 K=1,NRB
SS = AZ(K,I)*B(K,J)
30 S = S+SS
35 W(J) = S
DO 40 J=I,NCB
40 AZ(J,I) = W(J)
DO 50 I=1,NCB
DO 50 J=I,NCB
50 AZ(I,J) = AZ(J,I)
RETURN

```

```

999 CALL ZZBOMB (6HATXBA1,NERROR)
END

```

ATXBB  
-----

```
SUBROUTINE ATXBB (A,BZ,NRAT,NRB,NCB,KA,KBZ)
DIMENSION A(KA,1), BZ(KBZ,1)
COMMON / LWPKV1 / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/
```

```
C
C MATRIX MULTIPLICATION. A(TRANPOSE) * B = Z.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.
C BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF B OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C NRAT = 500
C DEVELOPED BY W A BENFIELD. NOVEMBER 1971.
C LAST REVISION BY RL WOHLN. MARCH 1976.
```

```
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX. SIZE(NRB,NRAT).
C BZ = INPUT MATRIX. SIZE(NRB,NCB).
C = OUTPUT RESULT MATRIX. SIZE (NRAT,NCB).
C NRAT = INPUT NUMBER OF ROWS OF MATRICES A(TRANS),Z. MAX=500.
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A(TRANS).
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z.
C KA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KBZ = INPUT ROW DIMENSION OF BZ IN CALLING PROGRAM.
```

```
C ERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.
```

NERROR=1

```
IF (NRAT.GT.500 .OR. NRAT.GT.KBZ .OR. NRB.GT.KBZ) GO TO 999
```

```
C
DO 40 J=1,NCB
DO 35 I=1,NRAT
S = ZERO
DO 30 K=1,NRB
SS = A(K,I)*BZ(K,J)
30 S = S+SS
35 W(I) = S
DO 40 I=1,NRAT
40 BZ(I,J) = W(I)
RETURN
```

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

ATXBB1

```
SUBROUTINE ATXBB1 (A,BZ,NRB,NCB,KA,KBZ)
  DIMENSION A(KA,1), BZ(KBZ,1)
  COMMON / LWRKV1 / W(500)
  DOUBLE PRECISION S,SS,ZERO
  DATA ZERO /0.0/
```

```
C
C SPECIAL MATRIX MULTIPLICATION. A(TRANPOSE) * B = Z.
C A IS ASSUMED UPPER TRIANGULAR, SQUARE.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NRB = 500
C DEVELOPED BY R L WOHLER AND W A BENFIELD. MAY 1972.
C LAST REVISION BY RL WOHLER. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C A      = INPUT MATRIX. SIZE(NRB,NRB).
C BZ     = INPUT MATRIX. SIZE(NRB,NCB).
C        = OUTPUT RESULT MATRIX. SIZE (NRB,NCB).
C NRB    = INPUT NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX A(SQUARE).
C        = MAX=500.
C NCB    = INPUT NUMBER OF COLS OF MATRICES B,Z.
C KA     = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KBZ    = INPUT ROW DIMENSION OF BZ IN CALLING PROGRAM.
```

NERROR EXPLANATION

```
C 1 = SIZE LIMITATION EXCEEDED.
```

NERROR=1

```
IF (NRB.GT.500 .OR. NRB.GT.KBZ) GO TO 999
```

```
C
  DO 40 J=1,NCB
  DO 35 I=1,NRB
  S = ZERO
  DO 30 K=1,I
  SS = A(K,I)*BZ(K,J)
30 S = S+SS
35 W(I) = S
  DO 40 I=1,NRB
40 BZ(I,J) = W(I)
  RETURN
```

```
C
999 CALL ZZBOMB (6HATXBB1,NERROR)
  END
```

ATXBB2

-----

```
SUBROUTINE ATXBB2 (A,BZ,NRB,NCB,KA,KBZ)
DIMENSION A(KA,1), BZ(KBZ,1)
COMMON / LWRKV1 / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/
```

```
C
C SPECIAL MATRIX MULTIPLICATION. A(TRANSPOSE) * B = Z.
C Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.
C BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF B OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NCB = 500
C DEVELOPED BY R L WOHLER AND W A BENFIELD. MAY 1972.
C LAST REVISION BY RL WOHLER. MARCH 1976.
```

```
C
C SUBROUTINE ARGUMENTS
```

```
C A = INPUT MATRIX. SIZE(NRB,NCB).
C BZ = INPUT MATRIX. SIZE(NRB,NCB).
C = OUTPUT RESULT MATRIX. SIZE (NCB,NCB).
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A(TRANS).
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z, ROWS OF A(TRANS).
C   MAX=500.
C KA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KBZ = INPUT ROW DIMENSION OF BZ IN CALLING PROGRAM.
```

```
C
C NERROR EXPLANATION
```

```
C 1 = SIZE LIMITATION EXCEEDED.
```

```
C
C NERROR=1
```

```
C IF (NCB.GT.500 .OR. NCB.GT.KBZ .OR. NRB.GT.KBZ) GO TO 999
```

```
C
C DO 40 J=1,NCB
C DO 35 I=1,J
C S = ZERO
C DO 30 K=1,NRB
C SS = A(K,I)*BZ(K,J)
30 S = S+SS
35 W(I) = S
C DO 40 I=1,J
C BZ(J,I) = W(I)
40 BZ(I,J) = W(I)
C RETURN
```

```
C
999 CALL ZZBOMB (6HATXBB2,NERROR)
END
```

AXBA1

-----

```

SUBROUTINE AXBA1 (AZ,B,NRA,NCA,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON / LWRKV1 / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERC /0.0/

```

```

C
C MATRIX MULTIPLICATION. A * B = Z.
C MATRIX P IS ASSUMED UPPER TRIANGULAR, SQUARE.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NCA = 500
C DEVELOPED BY R A PHILIPPUS. MAY 1972.
C LAST REVISION BY RL WOHLN. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C AZ = INPUT MATRIX. SIZE(NRA,NCA).
C   = OUTPUT RESULT MATRIX. SIZE(NRA,NCA).
C B = INPUT MATRIX. SIZE(NCA,NCA).
C NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z.
C NCA = INPUT NUMBER OF COLS OF MATRICES A,Z, SIZE OF B (SQUARE).
C   MAX=500.
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.
C
C : NERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.
C
C
C IF (NCA .GT. 500) GO TO 999
C
C DO 40 I=1,NRA
C DO 20 K=1,NCA
20 W(K) = AZ(I,K)
C DO 40 J=1,NCA
C S = ZERC
C DO 30 K=1,J
C SS = W(K)*B(K,J)
30 S = S + SS
40 AZ(I,J) = S
C RETURN
C
C 999 CALL ZZBOMB (6HAXBA1 ,NERROR)
C END

```

NERROR=1

AXBA2

-----

```

SUBROUTINE AXBA2 (AZ,B,N,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON / LWRKV1 / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/

```

```

C
C MATRIX MULTIPLICATION. A * B = Z.
C B IS ASSUMED UPPER TRIANGULAR, SQUARE.
C Z WILL BE SYMMETRIC. LOWER HALF CALCULATED, REFLECTED TO UPPER HALF.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   N = 500
C DEVELOPED BY R L WOHLN AND W A BENFIELD. MAY 1972.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C AZ = INPUT MATRIX. SIZE(N,N).
C   = OUTPUT RESULT MATRIX. SIZE(N,N).
C B = INPUT MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,B,Z (SQUARE). MAX=500.
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.

```

```

C
C NERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.

```

NERROR=1

```

C
C IF (N .GT. 500) GO TO 999

```

```

C
C DO 40 I=1,N
C DO 20 K=1,I
20 W(K) = AZ(I,K)
C DO 40 J=I,I
C S = ZERO
C DO 30 K=1,J
C SS = W(K)*B(K,J)
30 S = S + SS
C AZ(J,I) = S
40 AZ(I,J) = S
C RETURN

```

```

C
999 CALL ZZBOMB (6HAXBA2 ,NERROR)
END

```



AXBA3

```

SUBROUTINE AXBA3 (AZ,B,NRB,NCB,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON / LWRKVI / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.D/

```

```

C
C MATRIX MULTIPLICATION. A * B = Z.
C A IS ASSUMED UPPER TRIANGULAR, SQUARE.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF A OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NRB = 500
C DEVELOPED BY R L WOHLN AND WA BENFIELD. MAY 1972.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C AZ = INPUT MATRIX. SIZE(NRB,NRB).
C   = OUTPUT RESULT MATRIX. SIZE(NRB,NCB).
C B = INPUT MATRIX. SIZE(NRB,NCB).
C NRB = INPUT NUMBER OF ROWS OF MATRICES B,Z, SIZE OF MATRIX A(SQUARE).
C   MAX=500.
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z.
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.

```

```

C
C ERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.

```

NERROR=1

```

C
C   IF (NRB .GT. 500) GO TO 999
C
C   DO 40 I=1,NRB
C     DO 20 K=I,NRB
20  W(K) = AZ(I,K)
C     DO 40 J=1,NCB
C       S = ZERO
C     DO 30 K=I,NRB
C       SS = W(K)*B(K,J)
30  S = S + SS
40  AZ(I,J) = S
C     RETURN
C
C 999 CALL ZZBOMB (6MAXBA3 ,NERROR)
C     END

```

BABT  
-----

```

SUBROUTINE BABT (A,B,Z,NRB,NCB,KA,KB)
DIMENSION A(KA,1), R(KB,1), Z(KB,1)
COMMON /LWRKVI/ W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.D/

```

```

C
C SPECIAL TRIPLE MATRIX PRODUCT.  $B * A * B^T = Z$ .
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZFORMB.
C THE MAXIMUM SIZE IS
C   NCB = 500
C DEVELOPED BY CARL BODLEY. JANUARY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C A   = INPUT INNER MATRIX. SIZE(NCB,NCB).
C B   = INPUT OUTER MATRIX. SIZE(NRB,NCB).
C Z   = OUTPUT RESULT MATRIX. SIZE(NRB,NRB).
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX Z.
C NCB = INPUT NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX A. MAX=500.
C KA  = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KB  = INPUT ROW DIMENSION OF B,Z IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.
C
C
C                                  NERROR=1
C   IF (NCB .GT. 500 .OR. NCB .GT. KA .OR. NRB .GT. KB) GO TO 999
C
C   DO 40 J=1,NRB
C   DO 20 L=1,NCB
C     S = ZERO
C   DO 10 K=1,NCB
C     SS = A(L,K)*B(J,K)
C 10  S = S+SS
C 20  W(L) = S
C   DO 40 I=1,NRB
C     S = ZERO
C   DO 30 L=1,NCB
C     SS = B(I,L)*W(L)
C 30  S = S+SS
C 40  Z(I,J) = S
C     RETURN
C
C 999 CALL ZZBOMB (6HBAET ,NERROR)
C     END

```

```

SUBROUTINE BABTA (AZ,B,NRB,NCB,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON /LWRKV1/ W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/

```

```

C
C SPECIAL TRIPLE MATRIX PRODUCT. B*A*B(TRANSPCSE) = Z.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF A OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C NCB = 500
C DEVELOPED BY CARL BODLEY. JULY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C AZ = INPUT INNER MATRIX. SIZE(NCB,NCB).
C = OUTPUT RESULT MATRIX. SIZE(NRB,NRB).
C B = INPUT OUTER MATRIX. SIZE(NRB,NCB).
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX Z.
C NCB = INPUT NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX A. MAX=500.
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.

```

```

C
C NERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.
C
C NERROR=1
IF (NCB.GT.500 .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ .OR. NRB .GT. KB)
* GO TO 999

```

```

C
DO 30 I=1,NCB
DO 10 K=1,NCB
10 W(K) = AZ(I,K)
DO 30 J=1,NRB
S = ZERO
DO 20 K=1,NCB
SS = W(K)*B(J,K)
20 S = S + SS
30 AZ(I,J) = S

```

```

C
DO 60 J=1,NRB
DO 40 K=1,NCB
40 W(K) = AZ(K,J)
DO 60 I=1,NRB
S = ZERO
DO 50 K=1,NCB
SS = B(I,K)*W(K)
50 S = S + SS
60 AZ(I,J) = S
RETURN

```

```

C
999 CALL ZZBOMB (6HBAETA ,NERROR)

```

BABTA -- 2/ 2

END

BTAB

```
SUBROUTINE BTAB (A,B,Z,NRB,NCB,KAB,KZ)
DIMENSION A(KAB,1), B(KAB,1), Z(KZ,1)
COMMON /LWRKV1/ W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.D/
```

```
C
C TRIPLE MATRIX PRODUCT. B(TRANPOSE) * A * B = Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NRB = 500
C DEVELOPED BY RL WOHLN. FEBRUARY 1965.
C LAST REVISION BY RL WOHLN. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C A = INPUT INNER MATRIX. SIZE(NRB,NRB).
C B = INPUT OUTER MATRIX. SIZE(NRB,NCB).
C Z = OUTPUT RESULT MATRIX. SIZE(NCB,NCB).
C NRB = INPUT NUMBER OF ROWS OF MATRIX A, SIZE OF MATRIX A. MAX=500.
C NCB = INPUT NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX Z.
C KAB = INPUT ROW DIMENSION OF A,B IN CALLING PROGRAM.
C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.
C
C                                     NERROR=1
C   IF (NRB.GT.500 .OR. NRB.GT.KAB .OR. NCB.GT.KZ) GO TO 999
C
C   DO 40 J=1,NCB
C     DO 20 L=1,NRB
C       S = ZERO
C       DO 10 K=1,NRB
C         SS = A(L,K)*B(K,J)
C       10 S = S + SS
C     20 W(L) = S
C       DO 40 I=1,NCB
C         S = ZERO
C         DO 30 L=1,NRB
C           SS = B(L,I)*W(L)
C         30 S = S + SS
C     40 Z(I,J) = S
C     RETURN
C
C 999 CALL ZZBOMB (6*BTAB ,NERROR)
C   END
```

```

SUBROUTINE RTAB1(A,B,Z,NRB,NCB,KRA,KRB,KRZ)
COMMON / LWRKVI / V(500)
DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.D/

```

```

BTAB1 PERFORMS THE OPERATION (Z)=((B)TRANPOSE)*(A)*(B) .
BTAB1 CAN ALSO PERFORM THE OPERATION
(A)=((B)TRANPOSE)*(A)*(B) BY CALL BTAB1(A,B,A,--ETC--) .

```

```

IF NRB IS NEGATIVE A SYMMETRIC (Z) IS COMPUTED.

```

```

MAXIMUM SIZE NRB=500

```

```

INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

```

```

FORMA SUBROUTINE ZZBOMB IS CALLED.

```

```

CODED BY JOHN ADMIRE *NASA* JULY 1972 .

```

```

LAST REVISION BY RL WOHLER. MARCH 1976.

```

#### ARGUMENTS

```

A - INPUT MATRIX (A) SIZE(NRB BY NRB)
B - INPUT MATRIX (B) SIZE(NRB BY NCB)
Z - OUTPUT MATRIX (Z) SIZE(NCB BY NCB)
NRB - INPUT ABS(NRB) IS THE NUMBER OF ROWS IN (B)
NCB - INPUT NUMBER OF COLUMNS IN (B)
KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

#### NERROR EXPLANATIONS

```

1 = SIZE EXCEEDANCE.

```

```

NR=IABS(NRB)

```

```

NERROR = 1

```

```

IF(NR .GT. 500 .OR. NR .GT. KRA .OR. NR .GT. KRB
* .OR. NR .GT. KRZ .OR. NCB .GT. KRZ) GO TO 999
IF(NRB .GT. 0) GO TO 70
DO 30 I=1,NR
DO 10 K=1,NR
10 V(K)=A(I,K)
DO 30 J=1,NCB
SUM=ZERO
DO 20 K=1,NR
SS=V(K)*B(K,J)
20 SUM=SUM+SS
30 Z(I,J)=SUM
DO 60 J=1,NCB
DO 40 K=1,NR
40 V(K)=Z(K,J)
DO 60 I=1,J
SUM=ZERO
DO 50 K=1,NP
SS=B(K,I)*V(K)
50 SUM=SUM+SS

```

```
60 Z(I,J)=SUM
   DO 63 I=1,NR
   DO 63 J=I,NR
63 Z(J,I)=Z(I,J)
   RETURN
70 DO 100 I=1,NRB
   DO 80 K=1,NRB
80 V(K)=A(I,K)
   DO 100 J=1,NCB
   SUM=ZERO
   DO 90 K=1,NRB
   SS=V(K)*B(K,J)
90 SUM=SUM+SS
100 Z(I,J)=SUM
   DO 130 J=1,NCB
   DO 110 K=1,NRB
110 V(K)=Z(K,J)
   DO 130 I=1,NCB
   SUM=ZERO
   DO 120 K=1,NRB
   SS=B(K,I)*V(K)
120 SUM=SUM+SS
130 Z(I,J)=SUM
   RETURN
999 CALL ZZBOMB(6HBTAB1 ,NERRCR)
   END
```

```

SUBROUTINE BTAPA (AZ,P,NRB,NCB,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON / LWRKVI / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/

```

```

C
C TRIPLE MATRIX PRODUCT. B(TRANPOSE) * A * E = Z.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF A OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZES ARE
C   NRB = 500
C   NCB = 500
C DEVELOPED BY W A BENFIELD. MAY 1972.
C LAST REVISION BY PL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C AZ = INPUT INNER MATRIX. SIZE(NRB,NRB).
C   = OUTPUT RESULT MATRIX. SIZE(NCB,NCB).
C B = INPUT OUTER MATRIX. SIZE(NRB,NCB).
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, SIZE OF MATRIX A. MAX=500.
C NCB = INPUT NUMBER OF COLS OF MATRIX B, SIZE OF MATRIX Z. MAX=500.
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.

```

```

C
C ERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.

```

```

C
C                                     NERROR=1
C   IF (NRB.GT.500 .OR. NCB.GT.500 .OR. NRB.GT.KAZ .OR. NCB.GT.KAZ)
C *   GO TO 999

```

```

C
C   DO 20 I=1,NRB
C     DO 5 K=1,NRB
C   5 W(K) = AZ(I,K)
C     DO 20 J=1,NCB
C       S = ZERO
C       DO 10 K=1,NRB
C         SS = W(K)*B(K,J)
C   10 S = S + SS
C   20 AZ(I,J) = S

```

```

C
C   DO 30 J=1,NCB
C   DO 27 I=1,NRB
C   S = ZERO
C   DO 25 K=1,NRB
C     SS = B(K,I)*AZ(K,J)
C  25 S = S + SS
C  27 W(I) = S
C   DO 30 I=1,NCB
C  30 AZ(I,J) = W(I)
C   RETURN

```

```

C

```



BTABA -- 2/ 2

---

999 CALL ZZBOMB (6HBTABA ,NERROR)  
END

## BTABA2

-----

```
SUBROUTINE BTABA2 (AZ,B,N,KA)
DIMENSION AZ(KA,1), B(KA,1)
COMMON / LWRKVI / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO /0.0/
```

```
C
C TRIPLE MATRIX PRODUCT. B(TRANPOSE) * A * B = Z.
C A MUST BE SYMMETRIC TO GET CORRECT ANSWER.
C B IS ASSUMED UPPER TRIANGULAR.
C Z WILL BE SYMMETRIC. UPPER HALF CALCULATED, REFLECTED TO LOWER HALF.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   N = 500
C DEVELOPED BY R L WOHLN AND W A BENFIELD. MAY 1972.
C LAST REVISION BY RL WOHLN. MARCH 1976.
```

```
C
C SUBROUTINE ARGUMENTS
C AZ = INPUT INNER MATRIX. SIZE(N,N).
C   = OUTPUT RESULT MATRIX. SIZE(N,N).
C B = INPUT OUTER MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,B,Z. MAX=500.
C KA = INPUT ROW DIMENSION OF AZ AND B IN CALLING PROGRAM.
```

```
C
C NERROR EXPLANATION
C 1 = SIZE LIMITATION OR DIMENSION SIZE EXCEEDED.
```

NERROR=1

```
IF (N.GT.500 .OR. N.GT.KA) GO TO 999
```

```
C
DO 20 I=1,N
DO 5 K=1,N
5 W(K) = AZ(I,K)
DO 20 J=1,N
S = ZERO
DO 10 K=1,J
SS = W(K)*B(K,J)
10 S = S + SS
20 AZ(I,J) = S
C
DO 30 J=1,N
DO 28 I=1,J
S = ZERO
DO 25 K=1,I
SS = B(K,I)*AZ(K,J)
25 S = S + SS
28 W(I) = S
DO 30 I=1,J
AZ(I,J) = W(I)
30 AZ(J,I) = W(I)
RETURN
```

```
999 CALL ZZBOMB (6HBTABA2,NERROR)
END
```

```

SUBROUTINE BTABC1(A,B,C,Z,NRB,NCB,KRA,KRB,KRC,KRZ)
DIMENSION A(KRA,1),B(KRB,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKVI / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.0/

```

```

BTABC1 PERFORMS THE OPERATION (Z)=((B)TRANPOSE)*(A)*(B)+(C)
BTABC1 CAN ALSO PERFORM THE OPERATION
(C)=((B)TRANPOSE)*(A)*(B)+(C) BY CALL BTABC1(A,B,C,C,--ETC--)
(A)=((B)TRANPOSE)*(A)*(B)+(C) BY CALL BTABC1(A,B,C,A,--ETC--)

```

IF NRB IS NEGATIVE A SYMMETRIC (Z) IS COMPUTED.

MAXIMUM SIZE NRB=500

INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

FORMA SUBROUTINE ZZBOMB IS CALLED.  
CODED BY JOHN ADMIRE \*NASA\* JULY 1972 .  
LAST REVISION BY RL WOHLN. MARCH 1976.

#### ARGUMENTS

```

A - INPUT MATRIX (A) *DESTROYED* SIZE(NRB BY NRB)
B - INPUT MATRIX (B) SIZE(NRB BY NCB)
C - INPUT MATRIX (C) SIZE(NCB BY NCB)
Z - OUTPUT MATRIX (Z) SIZE(NCB BY NCB)
NRB - INPUT ABS(NRB) NUMBER OF ROWS IN (B)
NCB - INPUT NUMBER OF COLUMNS IN (B)
KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
KRC - INPUT ROW DIMENSION OF (C) IN CALLING PROGRAM
KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

#### NERROR EXPLANATIONS

1 = SIZE EXCEEDANCE.

NR=IABS(NRB)

NERROR = 1

```

IF(NR .GT. 500 .OR. NR .GT. KRA .OR. NR .GT. KRB .OR. NCB
* .GT. KRC .OR. NCB .GT. KRZ) GO TO 999

```

IF(NRB .GT. 0) GO TO 70

DO 30 I=1,NR

DO 10 K=1,NR

10 V(K)=A(I,K)

DO 30 J=1,NCB

SUM=ZERO

DO 20 K=1,NR

SS=V(K)\*B(K,J)

20 SUM=SUM+SS

30 A(I,J)=SUM

DO 60 J=1,NCB

DO 40 K=1,NR

40 V(K)=A(K,J)

DO 60 I=1,J

SUM=C(I,J)

```
DO 50 K=1,NR
SS=B(K,I)*V(K)
50 SUM=SUM+SS
60 Z(I,J)=SUM
DO 63 I=1,NR
DO 63 J=I,NR
63 Z(J,I)=Z(I,J)
RETURN
70 DO 100 I=1,NRB
DO 80 K=1,NRB
80 V(K)=A(I,K)
DO 100 J=1,NCB
SUM=ZERO
DO 90 K=1,NRB
SS=V(K)*B(K,J)
90 SUM=SUM+SS
100 A(I,J)=SUM
DO 130 J=1,NCB
DO 110 K=1,NRB
110 V(K)=A(K,J)
DO 130 I=1,NCB
SUM=C(I,J)
DO 120 K=1,NRB
SS=B(K,I)*V(K)
120 SUM=SUM+SS
130 Z(I,J)=SUM
RETURN
999 CALL ZZBOMB(6HBTABC1,NERROR)
END
```

BTDB1

-----

```
SUBROUTINE BTDB1(D,B,Z,NRB,NCB,KRB,KRZ)
DIMENSION D(1),B(KRB,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.0/
```

```
C
C BTDB1 PERFORMS THE OPERATION (Z)=((B)TRANSP0SE)(-D-)*(B)
C WHERE (-D-) IS A DIAGONAL MATRIX AND THE INPUT VECTOR (D)
C CONTAINS THE DIAGONAL ELEMENTS .
```

```
C
C MAXIMUM SIZE NR=500
```

```
C
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
```

```
C
C FORMA SUBROUTINE ZZBOMB IS CALLED.
C CODED BY JOHN ADMIRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLER. MARCH 1976.
```

#### ARGUMENTS

```
C
C D - INPUT VECTOR CONTAINING THE DIAGONAL ELEMENTS OF (-D-)
C B - INPUT MATRIX (B) SIZE(NRB BY NCB)
C Z - OUTPUT MATRIX (Z) SIZE(NCB BY NCB)
C NR - INPUT ABS(NRB) NUMBER OF ROWS IN (B)
C NCB - INPUT NUMBER OF COLUMNS IN (B)
C KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM
```

#### ERROR EXPLANATIONS

```
C
C 1 = SIZE EXCEEDANCE.
```

```
C
C NR=IABS(NRB)
```

NERROR = 1

```
IF(NR .GT. 500 .OR. NR .GT. KRB .OR. NCB .GT. KRZ) GO TO 999
DO 30 J=1,NCB
DO 10 K=1,NR
10 V(K)=D(K)*B(K,J)
DO 30 I=1,J
SUM=ZERO
DO 20 K=1,NR
SS=B(K,I)*V(K)
20 SUM=SUM+SS
30 Z(I,J)=SUM
DO 33 I=1,NR
DO 33 J=I,NR
33 Z(J,I)=Z(I,J)
RETURN
999 CALL ZZBOMB(6HBTDB1 ,NERROR)
END
```

```

SUBROUTINE BTDBC1(D,B,C,Z,NRB,NCB,KRB,KRC,KRZ)
DIMENSION D(1),B(KRB,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS

```

```

BTDBC1 PERFORMS THE OPERATION (Z)=((B)TRANSPDSE)*(-D-)*(B)+(C)
WHERE (-D-) IS A DIAGONAL MATRIX AND THE INPUT VECTOR (D)
CONTAINS THE DIAGONAL ELEMENTS .
BTDBC1 CAN ALSO PERFORM THE OPERATION
(C)=((B)TRANSPDSE)*(-D-)*(B)+(C) BY CALL BTDBC1(D,B,C,C,--ETC--)
IF NRB IS NEGATIVE A SYMMETRIC (Z) IS COMPUTED.

```

```

MAXIMUM SIZE NRB=500

```

```

INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.

```

```

FORMA SUBROUTINE ZZBOMB IS CALLED.

```

```

CODED BY JOHN ADMIRE *NASA* JULY 1972 .

```

```

LAST REVISION BY RL WOHLN. MARCH 1976.

```

#### ARGUMENTS

```

D - INPUT VECTOR CONTAINING THE DIAGONAL ELEMENTS OF (-D-)
B - INPUT MATRIX (B) SIZE(NRB BY NCB)
C - INPUT MATRIX (C) SIZE(NCB BY NCB)
Z - OUTPUT MATRIX (Z) SIZE(NCB BY NCB)
NRB - INPUT ABS(NRB) NUMBER OF ROWS IN (B)
NCB - INPUT NUMBER OF COLUMNS IN (B)
KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM
KRC - INPUT ROW DIMENSION OF (C) IN CALLING PROGRAM
KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

#### NERROR EXPLANATIONS

```

1 = SIZE EXCEEDANCE.

```

```

NR=IABS(NRB)

```

```

NERROR = 1

```

```

IF(NR .GT. 500 .OR. NR .GT. KRB .OR. NCB .GT. KRC
* .OR. NCB .GT. KRZ) GO TO 999
IF(NRB .GT. 0) GO TO 40
DO 30 J=1,NCB
DO 10 K=1,NR
10 V(K)=D(K)*P(K,J)
DO 30 I=1,J
SUM=C(I,J)
DO 20 K=1,NR
SS=B(K,I)*V(K)
20 SUM=SUM+SS
30 Z(I,J)=SUM
DO 33 I=1,NR
DO 33 J=1,NR
33 Z(J,I)=Z(I,J)
RETURN
40 DO 70 J=1,NCB
DO 50 K=1,NR

```

```
50 V(K)=D(K)*P(K,J)
   DO 70 I=1,NCB
   SUM=C(1,J)
   DO 60 K=1,NP
   SS=B(K,I)*V(K)
60 SUM=SUM+SS
70 Z(I,J)=SUM
   RETURN
999 CALL ZZBOMB(6HBTDBC1,NERROR)
   ENL
```

COLMLT

SUBROUTINE COLMLT (AVEC,B,Z,NR,NC,KR)  
DIMENSION AVEC(1), B(KR,1), Z(KR,1)

C  
C MULTIPLY EACH ELEMENT IN COLUMN(J) OF MATRIX B BY  
C ELEMENT(J) OF VECTOR AVEC.  
C MATRICES B,Z MAY SHARE SAME CORE LOCATIONS.  
C CODED BY RL WOHLER. FEBRUARY 1965.  
C  
C SUBROUTINE ARGUMENTS  
C AVEC = INPUT VECTOR. SIZE(NC).  
C B = INPUT MATRIX. SIZE(NR,NC).  
C Z = OUTPUT RESULT MATRIX. SIZE(NR,NC).  
C NR = INPUT NUMBER OF ROWS IN MATRICES B,Z.  
C NC = INPUT NUMBER OF COLS IN MATRICES B,Z. ELEMENTS IN VECTOR AVEC.  
C KR = INPUT ROW DIMENSION OF B,Z IN CALLING PROGRAM.  
C  
C DO 10 I=1,NR  
C DO 10 J=1,NC  
10 Z(I,J) = AVEC(J) \* B(I,J)  
C RETURN  
C END



COMENT

-----

```

SUBROUTINE COMENT
DIMENSION IREMRK(13)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C READ COMMENT CARDS AND PRINT THEM UNDER PAGE HEADING OF FORMA
C SUBROUTINE PAGEHD. COMMENT CARDS MAY HAVE ANY KEYPUNCH SYMBOL
C IN CARD COLUMNS 1-78.
C IF IT IS DESIRED TO HAVE ANY GIVEN COMMENT CARD PRINT ON A NEW
C PAGE, SUPPLY THE LETTER P IN COLUMN 80 ON THAT CARD.
C ROUTINE IS ENDED BY SUPPLYING A CARD WITH ZEROS IN COLUMNS 1 THRU 10.
C CALLS FORMA SUBROUTINE PAGEHD.
C CODED BY RF HRUDA. MARCH 1966.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
1001 FORMAT (13A6,1X,A1)
2001 FORMAT (///)
2002 FORMAT (22X,13A6)
2050 FORMAT (/ 1X 123(1P-) )

```

```

C
      N = 0
1 READ (NIT,1001) (IREMRK(I),I=1,13),IPGHD
  IF (IREMRK(1) .EQ. 6H000000) RETURN
  N = N+1
  IF (N.NE.1 .AND. IPGHD.NE.1HP) GO TO 2
  IF (MINI .NE. 4HMINI) GO TO 800
  IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
  IF ((NLINE+2) .GT. MAXLIN) GO TO 800
  WRITE (NOT,2050)
  NLINE = NLINE + 2
  GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2001)
  NLINE = NLINE + 3
  N = 1
2 IF ((N+8) .EQ. MAXLIN) N = 0
  WRITE (NOT,2002) (IREMRK(I),I=1,13)
  NLINE = NLINE + 1
  GO TO 1
END

```

```

SUBROUTINE COMPAR (A,R,NR,NC,NDIG,GTOL,ANAME,RNAME,KA,KR)
DIMENSION A(KA,1), R(KR,1)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA EPS/0.0/
DATA NIT,NCT/5,6/

```

```

C
C COMPARE TWO MATRICES ELEMENT BY ELEMENT. PRINT OUT ELEMENT DATA WHEN
C ELEMENTS DO NOT COMPARE TO SPECIFIED NUMBER OF DIGITS (NDIG).
C ELEMENT VALUES BELOW TOLERANCE (GTOL) ARE IGNORED.
C A MAXIMUM OF 1000 NONCOMPARABLE ELEMENTS ARE PRINTED.
C DEVELOPED BY JW ERNST, RL WOHLN. OCTOBER 1971.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX TO BE CHECKED. SIZE(NR,NC).
C R = MATRIX CONTAINING REFERENCE VALUES. SIZE(NR,NC).
C NR = NUMBER OF ROWS OF MATRICES A,R.
C NC = NUMBER OF COLS OF MATRICES A,R.
C NDIG = NUMBER OF DIGITS TO BE COMPARED BETWEEN (A) AND (R).
C GTOL = GARBAGE TOLERANCE. MATRIX ELEMENTS (ABS) LESS THAN OR
C EQUAL TO THIS VALUE WILL BE IGNORED.
C ANAME = NAME OF MATRIX A.
C RNAME = NAME OF MATRIX R.
C KA = ROW DIMENSION OF A IN CALLING PROGRAM.
C KR = ROW DIMENSION OF R IN CALLING PROGRAM.

```

```

C
2001 FORMAT (/// 10X 34H SUBROUTINE COMPAR COMPARES MATRIX ,A6,
*          21H TO REFERENCE MATRIX ,A6,20H ELEMENT BY ELEMENT.
*          / 44X 6H-----, 21X 6H-----,
*          / 10X 25HELEMENTS ARE COMPARED TO ,I2, 10H DIGITS.
*          24HELEMENTS (ABS) LESS THAN,E10.3,13H ARE IGNORED.
*          / 35X 2H--, 35X 9H-----)
2002 FORMAT ( / 15X 48H DISAGREEMENT WAS FOUND AT THE FOLLOWING ELEMENTS
*          / 18X 1HI, 3X 1HJ, 6X 7HMATRIX ,A6, 5X 11HREF MATRIX ,A6)
2003 FORMAT (15X 2I4, 2E19.8)
2004 FORMAT ( / 15X 7HMATRIX ,A6, 30H AGREES WITH REFERENCE MATRIX ,A6)
2005 FORMAT ( / 10X 25H END OF SUBROUTINE COMPAR.)

```

```

C
WRITE (NCT,2001) ANAME,RNAME,NDIG,GTOL
NLINE = NLINE + 7
ATOL = 10.**(-NDIG)
NEL = 0
DO 20 J=1,NC
DO 20 I=1,NR
IF (ABS(A(I,J)).LE.GTOL .AND. ABS(R(I,J)).LE.GTOL) GO TO 20
IF (ABS(R(I,J)) .LE. EPS) GO TO 10
IF (ABS((A(I,J)-R(I,J))/R(I,J)) .LE. ATOL) GO TO 20
10 IF (NEL .EQ. 0) WRITE (NCT,2002) ANAME,RNAME
NLINE = NLINE + 3
NEL = NEL+1
IF (NEL .GT. 1000) GO TO 30
WRITE (NCT,2003) I,J,A(I,J),R(I,J)
NLINE = NLINE + 1
20 CONTINUE
IF (NEL .EQ. 0) WRITE (NCT,2004) ANAME,RNAME

```

```
NLINE = NLINE + 2  
30 WRITE (NOT,2005)  
NLINE = NLINE + 2  
RETURN  
END
```

DB1

SUBROUTINE DB1(D,B,Z,NRB,NCB,KRB,KRZ)  
DIMENSION D(1),B(KRB,1),Z(KRZ,1)

DB1 PERFORMS THE OPERATION  $Z = (-D) * (B)$   
WHERE D IS A VECTOR THAT CONTAINS THE DIAGONAL ELEMENTS OF  $(-D-)$   
DB1 CAN ALSO PERFORM THE OPERATION  
 $(B) = (-D) * (B)$  BY CALL DB1(B,B,--ETC--)

IF NRB IS NEGATIVE AND ABS(NRB) IS EQUAL TO NCB  
A SQUARE, SYMMETRIC (Z) IS CALCULATED.

FORMA SUBROUTINE ZZBOMB IS CALLED .  
CODED BY JOHN ADMIRAL \*NASA\* JULY 1972 .  
LAST REVISION BY RL WOHLER. MARCH 1976.

ARGUMENTS

D - INPUT A VECTOR THAT CONTAINS THE DIAGONAL ELEMENTS OF  $(-D-)$   
B - INPUT MATRIX (B) SIZE(NRB BY NCB)  
Z - OUTPUT MATRIX (Z) SIZE(NRB BY NCB)  
NRB - INPUT ABS(NRB) IS THE NUMBER OF ROWS IN (B)  
KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM  
KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

NERROR EXPLANATIONS

1 = SIZE EXCEEDANCE.  
2 = NON-SQUARE RESULT ASKED FOR.

NR=IABS(NRB)

NERROR = 1

IF(NK .GT. KRB .OR. NR .GT. KRZ) GO TO 999  
IF(NRB .GT. 0) GO TO 20

NERROR = 2

IF(NR .NE. NCB) GO TO 999  
DO 10 J=1,NR  
DO 10 I=1,J  
Z(I,J)=D(I)\*B(I,J)  
10 Z(J,I)=Z(I,J)  
RETURN  
20 DO 30 J=1,NCB  
DO 30 I=1,NRB  
30 Z(I,J)=D(I)\*B(I,J)  
RETURN  
999 CALL ZZBOMB(6HDB1 ,NERROR)  
END

```

SUBROUTINE DCOM1 (A,Z,N,KR)
DIMENSION A(KR,1), Z(KR,1)
DOUBLE PRECISION DM,DS
DATA EPS/ 0.0 /
DATA NIT,NOT/5,6/

```

```

C
C DECOMPOSE MATRIX (A) TO FORM UPPER TRIANGULAR MATRIX (Z) SUCH THAT
C A = Z(TRANS) * Z. CHOLESKI SQUARE ROOT METHOD.
C MATRIX (A) SHOULD BE REAL, SQUARE, SYMMETRIC, POSITIVE DEFINITE.
C UPPER HALF OF MATRIX (A) IS USED.
C MATRICES (A) AND (Z) MAY SHARE SAME CORE LOCATIONS.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY RL WOHLER. OCTOBER 1970.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX TO BE DECOMPOSED. SIZE(N,N).
C Z = OUTPUT MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,Z.
C KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.

```

```

C
C ERROR EXPLANATION
C 1 = MATRIX IS NON-POSITIVE DEFINITE AT A(1,1).
C 2 = MATRIX IS NON-POSITIVE DEFINITE.

```

```

C
C 3001 FORMAT (5H11 = I3)

```

NERROR=1

```

IF (A(1,1) .LE. EPS) GO TO 999
Z(1,1) = SQRT(A(1,1))
IF (N .EQ. 1) RETURN
DO 5 J=2,N
5 Z(1,J) = A(1,J)/Z(1,1)

```

C

NERROR=2

```

DO 30 I=2,N
IM1 = I-1
IP1 = I+1
DS = A(I,I)
DO 10 K=1,IM1
DM = Z(K,I)**2
10 DS = DS - DM
Z(I,I) = DS
IF (Z(I,I) .LE. EPS) GO TO 998
Z(I,I) = SQRT(Z(I,I))
IF (I .EQ. N) GO TO 40
DO 30 J=IP1,N
DS = A(I,J)
DO 20 K=1,IM1
DM = Z(K,I)*Z(K,J)
20 DS = DS - DM
Z(I,J) = DS
30 Z(I,J) = Z(I,J)/Z(I,I)

```

C

```
40 DO 50 I=2,N  
    IM1 = I-1  
    DO 50 J=1,IM1  
50 Z(I,J) = 0.0  
    RETURN
```

C

```
998 WRITE (NOT,3001) I  
999 CALL ZZBOMB (6HDCOM1 ,NERROR)  
    END
```

DIAG

-----

1  
C  
C SUBROUTINE DIAG (AVEC,Z,N,KR)  
C DIMENSION AVEC(1), Z(KR,1)

C  
C DIAGONALIZE A VECTOR (ROW OR COLUMN MATRIX) INTO A SQUARE MATRIX.  
C CODED BY RL WOHLN. FEB 1965.

C  
C SUBROUTINE ARGUMENTS

C AVEC = INPUT VECTOR. SIZE(N).

C Z = OUTPUT RESULT MATRIX. SIZE(N,N).

C N = INPUT SIZE OF MATRIX Z (SQUARE), LENGTH OF VECTOR AVEC.

C KR = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.

C  
DO 20 I=1,N

DO 10 J=1,N

10 Z(I,J) = 0.0

20 Z(I,I) = AVEC(I)

RETURN

END

DIFF1  
-----

SUBROUTINE DIFF1 (XA,XZ,YA,Z,NXA,NXZ,NCA,KA,KZ)  
 DIMENSION XA(1),XZ(1),YA(KA,1),Z(KZ,1)

C  
 C LINEAR DIFFERENTIATION.  
 C VALUES OF XZ MAY BE OUTSIDE OF XA. (EXTRAPOLATION).  
 C CODED BY RF HRUDA. SEPTEMBER 1965.  
 C LAST REVISION BY J ERNST, OCT 1973.  
 C  
 C SUBROUTINE ARGUMENTS  
 C XA = INPUT VECTOR OF X-COORDINATES FOR ROWS OF YA. MUST BE IN  
 C INCREASING ORDER. SIZE(NXA).  
 C XZ = INPUT VECTOR OF X-COORDINATES FOR DERIVATIVES. SIZE(NXZ).  
 C YA = INPUT MATRIX OF Y-COORDINATES TO BE DIFFERENTIATED.  
 C SIZE(NXA,NCA).  
 C Z = OUTPUT MATRIX OF DERIVATIVES. SIZE(NXZ,NCA).  
 C EACH COLUMN OF Z HAS DERIVATIVES OF THE RESPECTIVE  
 C COLUMN OF YA.  
 C NXA = INPUT NUMBER OF XA STATIONS, ROWS OF MATRIX YA.  
 C NXZ = INPUT NUMBER OF XZ STATIONS, ROWS OF MATRIX Z.  
 C NCA = INPUT NUMBER OF COLUMN VECTORS IN MATRICES YA,Z.  
 C KA = INPUT ROW DIMENSION OF YA IN CALLING PROGRAM.  
 C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.  
 C  
 C DO 30 K=1,NXZ  
 C DO 10 I=1,NXA  
 C IF(XZ(K).LE.XA(I+1) .OR. (I+1).EQ.NXA) GO TO 20  
 C 10 CONTINUE  
 C 20 DO 30 J=1,NCA  
 C 30 Z(K,J) = (YA(I+1,J)-YA(I,J))/(XA(I+1)-XA(I))  
 C  
 C RETURN  
 C END



SUBROUTINE DIFF2 (XA,XZ,YA,Z,NXA,NXZ,NCA,KA,KZ)  
 DIMENSION XA(1),XZ(1),YA(KA,1),Z(KZ,1)

C  
 C DIPARABOLIC DIFFERENTIATION.  
 C (PARABOLIC DIFFERENTIATION IN FIRST, LAST BAYS AND OUTSIDE XA).  
 C VALUES OF XZ MAY BE OUTSIDE OF XA. (EXTRAPOLATION).  
 C CALLS FORMA SUBROUTINE ZZBOMB.  
 C CODED BY RF HRUDA. FEBRUARY 1965.  
 C LAST REVISION BY WA BENFIELD. MARCH 1976.

C  
 C SUBROUTINE ARGUMENTS  
 C XA = INPUT VECTOR OF X-COORDINATES FOR ROWS OF YA. MUST BE IN  
 C INCREASING ORDER. SIZE(NXA).  
 C XZ = INPUT VECTOR OF X-COORDINATES FOR DERIVATIVES. SIZE(NXZ).  
 C YA = INPUT MATRIX OF Y-COORDINATES TO BE DIFFERENTIATED.  
 C SIZE(NXA,NCA).  
 C Z = OUTPUT MATRIX OF DERIVATIVES. SIZE(NXZ,NCA).  
 C EACH COLUMN OF Z HAS DERIVATIVES OF THE RESPECTIVE  
 C COLUMN OF YA.  
 C NXA = INPUT NUMBER OF XA STATIONS, ROWS OF MATRIX YA.  
 C NXZ = INPUT NUMBER OF XZ STATIONS, ROWS OF MATRIX Z.  
 C NCA = INPUT NUMBER OF COLUMN VECTORS IN MATRICES YA,Z.  
 C KA = INPUT ROW DIMENSION OF YA IN CALLING PROGRAM.  
 C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.

C  
 C NERROR EXPLANATION  
 C 1 = LESS THAN 3 STATIONS.

NERROR = 1

C  
 C IF (NXA .LT. 3) GO TO 999  
 C  
 C DO 400 K=1,NXZ  
 C IF (XZ(K).LE.XA(2)) GO TO 100  
 C IF (XZ(K).GE.XA(NXA-1)) GO TO 300  
 C DO 50 I=3,NXA  
 C IF (XZ(K).LE.XA(I)) GO TO 200  
 C 50 CONTINUE  
 C  
 C FIRST BAY OR LEFT EXTRAPOLATION.  
 C 100 BAYL = XA(2)-XA(1)  
 C H = (XZ(K)-XA(1))/BAYL  
 C D = (XA(3)-XA(1))/BAYL  
 C DO 102 J=1,NCA  
 C 102 Z(K,J)=( YA(1,J)\*(2.0\*H-1.0-D)/D  
 C \* +YA(2,J)\*(2.0\*H-D)/(1.0-D)  
 C \* +YA(3,J)\*(-2.0\*H+1.0)/(D-D\*\*2) )/BAYL  
 C GO TO 400  
 C  
 C INTERIOR BAY.  
 C 200 BAYL = XA(I)-XA(I-1)  
 C H = (XZ(K) -XA(I-1))/BAYL  
 C C = (XA(I-2)-XA(I-1))/BAYL  
 C D = (XA(I+1)-XA(I-1))/BAYL  
 C DO 202 J=1,NCA  
 C 202 Z(K,J)=( YA(I-2,J)\*(3.0\*H\*\*2-4.0\*H+1.0)/(C-C\*\*2)

```

*      +YA(I-1,J)*(3.0*H**2*(C-D)+2.0*H*(2.0*D-C)-D*(1.+C))/(C*D)
*      +YA(I  ,J)*(3.0*H**2*(D-C)+2.0*H*(1.0-2.0*D+C)-C*(1.0-D))/
*      ((1.0-C)*(1.0-D))
*      +YA(I+1,J)*(-3.0*H**2+2.0*H)/(D-D**2) )/BAYL

```

GO TO 400

C

C LAST BAY OR RIGHT EXTRAPOLATION.

```
300 BAYL = XA(NXA)-XA(NXA-1)
```

```
  H = (XZ(K) -XA(NXA-1))/BAYL
```

```
  C = (XA(NXA-2)-XA(NXA-1))/BAYL
```

```
  DO 302 J=1,NCA
```

```
302 Z(K,J)=( YA(NXA-2,J)*(-2.0*H+1.0)/(C-C**2)
```

```
*      +YA(NXA-1,J)*(2.0*H-1.0-C)/C
```

```
*      +YA(NXA  ,J)*(2.0*H-C)/(1.0-C) )/BAYL
```

C

```
400 CONTINUE
```

```
  RETURN
```

C

```
999 CALL ZZBOMB (6HDIFF2 ,NERROR)
```

```
  END
```

## DISA

```

SUBROUTINE DISA (A,IRA,JCA,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)
DIMENSION A(KRA,1), Z(KRZ,1)
C
C MATRIX DISASSEMBLY. (MATRIX Z FROM MATRIX A).
C CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY PL WOHLER. FEB 1965.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX. SIZE(NRA,NCA).
C IRA = INPUT ROW NUMBER IN MATRIX A OF FIRST ROW OF MATRIX Z.
C JCA = INPUT COL NUMBER IN MATRIX A OF FIRST COL OF MATRIX Z.
C Z = OUTPUT RESULT MATRIX. SIZE(NRZ,NCZ).
C NRA = INPUT NUMBER OF ROWS OF MATRIX A.
C NCA = INPUT NUMBER OF COLS OF MATRIX A.
C NRZ = INPUT NUMBER OF ROWS OF MATRIX Z.
C NCZ = INPUT NUMBER OF COLS OF MATRIX Z.
C KRA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KRZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.
C
C NERROR EXPLANATION
C 1 = LOOKING FOR DATA OUTSIDE OF MATRIX A ROWS.
C 2 = LOOKING FOR DATA OUTSIDE OF MATRIX A COLUMNS.
C
C
C IF ((IRA-1+NRZ) .GT. NRA) GO TO 999
C
C IF ((JCA-1+NCZ) .GT. NCA) GO TO 999
C
C DO 10 IZ=1,NRZ
C IA = IZ + IRA - 1
C DO 10 JZ=1,NCZ
C JA = JZ + JCA - 1
C 10 Z(IZ,JZ) = A(IA,JA)
C RETURN
C
C 999 CALL ZZBOMB (6HDISA ,NERROR)
C END
```

```

SUBROUTINE EIGN1 (A,VAL,VEC,NIN,FODIN,KR)
DIMENSION A(KR,1), VAL(1), VEC(KR,1)
DATA NIT,NOT/5,0/

```

```

C
C CALCULATE EIGENVALUES / EIGENVECTORS OF (A)(VEC) = (VEC)(-VAL-).
C JACOBI METHOD, THRESHOLD VERSION. PROGRESS FROM PIVOT ELEMENT
C (IPIVOT,JPIVOT) TO ELEMENT (IPIVOT,JPIVOT+1) AFTER A PIVOT.
C THE (A) MATRIX SHOULD BE REAL, SYMMETRIC. UPPER HALF IS USED.
C CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY RL WOHLN. APRIL 1969.
C LAST REVISION BY RL WOHLN. JANUARY 1975.
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX TO BE DIAGONALIZED. SIZE(N,N). *DESTROYED*
C VAL = OUTPUT VECTOR OF EIGENVALUES. SIZE(N).
C VEC = OUTPUT MATRIX OF EIGENVECTORS. SIZE(N,N).
C NIN = INPUT ABS(NIN)=N IS THE SIZE OF MATRICES A,VEC, VECTOR VAL.
C IF NIN IS NEGATIVE, INITIAL VEC MATRIX IS ASSUMED TO
C BE SUPPLIED FROM ARGUMENT.
C FODIN = INPUT FINAL OFF-DIAGONAL VALUE FOR DIAGONALIZED A.
C IF FODIN .LE. 0., FOD=TRACE(A)*10**(-21) WILL BE USED.
C KR = INPUT ROW DIMENSION OF A,VEC IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = SUM OF THE DIAGONALS IS NOT POSITIVE.
C
2001 FORMAT (//// 54X,18H(SUBROUTINE EIGN1) )
2002 FORMAT (//41X,26HFINAL OFF-DIAGONAL (FOD) =E10.3, 8H (INPUT))
2003 FORMAT (//39X,26HFINAL OFF-DIAGONAL (FOD) =E10.3,13H (CALCULATED))
C
N = IABS(NIN)
IF (NIN .LT. 0) GO TO 10
C SET INITIAL VEC MATRIX TO UNITY.
DO 6 I=1,N
DO 5 J=1,N
5 VEC(I,J) = 0.0
6 VEC(I,I) = 1.0
C
10 IF (N .EQ. 1) GO TO 60
C FIND LARGEST OFF-DIAGONAL ELEMENT (THRESH) OF A.
C CALCULATE SUM OF DIAGONALS (TRACE) OF A.
TRACE = 0.
THRESH = ABS(A(1,2))
NMI = N-1
DO 15 I=1,NMI
TRACE = TRACE + A(I,I)
IPI = I+1
DO 15 J=IPI,N
15 IF (ABS(A(I,J)) .GT. THRESH) THRESH:=ABS(A(I,J))
TRACE = TRACE + A(N,N)
FOD = FODIN
IF (FODIN .LE. 0.) FOD=TRACE*1.E-21
WRITE (NOT,2001)
IF (FODIN .GT. 0.) WRITE (NOT,2002) FOD

```

```

IF (FODIN .LE. 0.) WRITE (NO1,2003) FOD
                                                    NERROR=1
IF (FOD .LE. 0.) GO TO 999
IF (THRESH .LE. FOD) GO TO 60
C
C SCAN UPPER OFF-DIAGONAL ELEMENTS OF MATRIX A BY ROWS UNTIL A VALUE
C GREATER THAN THRESH IS FOUND. PIVOT ON THIS ELEMENT (IP,JP).
20 THRESH = THRESH/10.
   IF (THRESH .LT. FOD) THRESH=FOD
22 IREDO = 0
   DO 41 IP=1,NM1
     IPM1 = IP-1
     IPP1 = IP+1
     DO 40 JP=IPP1,N
       IF (ABS(A(IP,JP)) .LT. THRESH) GO TO 40
     IREDO = 1
C CALCULATE ROTATION VALUES.
   DEL = A(IP,IP) - A(JP,JP)
   RAD = SQRT (DEL**2 + 4.*A(IP,JP)**2)
   IF (DEL .LT. 0.) RAD=-RAD
   TN = (2. * A(IP,JP)) / (DEL + RAD)
   CS = 1. / SQRT (1. + TN**2)
   SN = TN * CS
C DIAGONALIZE MATRIX (A). ONLY UPPER HALF IS USED.
   JPM1 = JP-1
   JPP1 = JP+1
   IF (IP .EQ. 1) GO TO 33
   DO 32 I=1,IPM1
     AIIP = A(I,IP)*CS + A(I,JP)*SN
     A(I,JP) = -A(I,IP)*SN + A(I,JP)*CS
32 A(I,IP) = AIIP
33 IF (IPP1 .EQ. JP) GO TO 35
   DO 34 I=IPP1,JPM1
     APII = A(IP,I)*CS + A(I,JP)*SN
     A(I,JP) = -A(IP,I)*SN + A(I,JP)*CS
34 A(IP,I) = APII
35 IF (JP .EQ. N) GO TO 37
   DO 36 I=JPP1,N
     APII = A(IP,I)*CS + A(JP,I)*SN
     A(JP,I) = -A(IP,I)*SN + A(JP,I)*CS
36 A(IP,I) = APII
37 AIPIP = A(IP,IP)
   AJPJP = A(JP,JP)
   CS2 = CS**2
   SN2 = SN**2
   ASC = 2.*A(IP,JP)*SN*CS
   A(IP,IP) = AIPIP*CS2 + ASC + AJPJP*SN2
   A(JP,JP) = AIPIP*SN2 - ASC + AJPJP*CS2
   A(IP,JP) = 0.0
C CALCULATE EIGENVECTORS.
   DO 38 I=1,N
     VECIIP = VEC(I,IP)*CS + VEC(I,JP)*SN
     VEC(I,JP) = -VEC(I,IP)*SN + VEC(I,JP)*CS
38 VEC(I,IP) = VECIIP
40 CONTINUE

```

```
41 CONTINUE
   IF (IREDO .EQ. 1) GO TO 22
   IF (THRESH .GT. FOD) GO TO 20
```

```
C
C PLACE DIAGONAL FROM A INTO VAL (EIGENVALUES).
```

```
60 DO 61 I=1,N
61 VAL(I) = A(I,I)
   RETURN
```

```
C
999 CALL ZZBOMB (6HEIGN1 ,NERROR)
   END
```

```

SUBROUTINE EIGNIA (A,VAL,VEC,NIN,CTVIN,KR)
DIMENSION A(KR,1), VAL(1), VEC(KR,1)
C
C CALCULATE EIGENVALUES / EIGENVECTORS OF (A)(VEC) = (VEC)(-VAL-).
C JACOBI METHOD, THRESHOLD VERSION. PROGRESS FROM PIVOT ELEMENT
C (IPIVOT,JPIVOT) TO ELEMENT (IPIVOT,JPIVOT+1) AFTER A PIVOT.
C THE (A) MATRIX SHOULD BE REAL, SYMMETRIC. UPPER HALF IS USED.
C DEVELOPED BY F L WOHLN. AUGUST 1972.
C LAST REVISION BY R A PHILIPPUS. JUNE 1973.
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX TO BE DIAGONALIZED. SIZE(N,N). *DESTROYED*
C VAL = OUTPUT VECTOR OF EIGENVALUES. SIZE(N).
C VEC = OUTPUT MATRIX OF EIGENVECTORS. SIZE(N,N).
C NIN = INPUT ABS(NIN)=N IS THE SIZE OF MATRICES A,VEC, VECTOR VAL.
C IF NIN IS NEGATIVE, INITIAL VEC MATRIX IS ASSUMED TO
C BE SUPPLIED FROM ARGUMENT.
C CTVIN = INPUT CONVERGENCE TOLERANCE ON EIGENVALUES. CONVERGENCE
C ASSUMED IF ABS(EIGENVALUE) LESS THAN CTVIN OR IF
C THE EIGENVALUE RATIO OF (CURRENT-PRECEDING)/CURRENT
C IS LESS THAN CTVIN.
C IF CTVIN .LE. 0.; 10**-6 WILL BE USED.
C KR = INPUT ROW DIMENSION OF A,VEC IN CALLING PROGRAM.
C
N = IABS(NIN)
IF (NIN .LT. 0) GO TO 10
: SET INITIAL VEC MATRIX TO UNITY.
DO 6 I=1,N
DO 5 J=1,N
5 VEC(I,J) = 0.0
6 VEC(I,I) = 1.0
C
10 IF (N .EQ. 1) GO TO 60
C
C SET INITIAL EIGENVALUES, CONVERGENCE TOLERANCE.
DO 12 I=1,N
12 VAL(I) = A(I,I)
CTVAL = CTVIN
IF (CTVIN .LE. 0.) CTVAL=1.E-6
C FIND LARGEST OFF-DIAGONAL ELEMENT (THRESH) OF A.
THRESH = ABS(A(1,2))
NM1 = N-1
DO 15 I=1,NM1
IP1 = I+1
DO 15 J=IP1,N
15 IF (ABS(A(I,J)) .GT. THRESH) THRESH=ABS(A(I,J))
C
C SCAN UPPER OFF-DIAGONAL ELEMENTS OF MATRIX A BY ROWS UNTIL A VALUE
C GREATER THAN THRESH IS FOUND. PIVOT ON THIS ELEMENT (IP,JP).
20 THRESH = THRESH/10.
22 IREDO = 0
DO 41 IP=1,NM1
IPM1 = IP-1
IPP1 = IP+1
DO 40 JP=IPP1,N

```

```

        IF (ABS(A(IP,JP)) .LT. THRESH) GO TO 40
        IREDO = 1
C   CALCULATE ROTATION VALUES.
        DEL = A(IP,IP) - A(JP,JP)
        RAD = SQRT (DEL**2 + 4.*A(IP,JP)**2)
        IF (DEL .LT. 0.) RAD=-RAD
        TN = (2. * A(IP,JP)) / (DEL + RAD)
        CS = 1. / SQRT (1. + TN**2)
        SN = TN * CS
C   DIAGONALIZE MATRIX (A). ONLY UPPER HALF IS USED.
        JPM1 = JP-1
        JPP1 = JP+1
        IF (IP .EQ. 1) GO TO 33
        DO 32 I=1,IPM1
        AIIP = A(I,IP)*CS + A(I,JP)*SN
        A(I,JP) = -A(I,IP)*SN + A(I,JP)*CS
32  A(I,IP) = AIIP
33  IF (IPM1 .EQ. JP) GO TO 35
        DO 34 I=IPM1,JPM1
        AIPI = A(IP,I)*CS + A(I,JP)*SN
        A(I,JP) = -A(IP,I)*SN + A(I,JP)*CS
34  A(IP,I) = AIPI
35  IF (JP .EQ. N) GO TO 37
        DO 36 I=JPP1,N
        AIPI = A(IP,I)*CS + A(JP,I)*SN
        A(JP,I) = -A(IP,I)*SN + A(JP,I)*CS
36  A(IP,I) = AIPI
37  AIPIP = A(IP,IP)
        AJPJP = A(JP,JP)
        CS2 = CS**2
        SN2 = SN**2
        ASC = 2.*A(IP,JP)*SN*CS
        A(IP,IP) = AIPIP*CS2 + ASC + AJPJP*SN2
        A(JP,JP) = AIPIP*SN2 - ASC + AJPJP*CS2
        A(IP,JP) = 0.0
C   CALCULATE EIGENVECTORS.
        DO 38 I=1,N
        VECIIP = VEC(I,IP)*CS + VEC(I,JP)*SN
        VEC(I,JP) = -VEC(I,IP)*SN + VEC(I,JP)*CS
38  VEC(I,IP) = VECIIP
40  CONTINUE
41  CONTINUE
        IF (IREDO .EQ. 1) GO TO 22
C   TEST EIGENVALUES FOR CONVERGENCE.
        DO 52 I=1,N
        IF (ABS(A(I,I)) .LT. CTVAL) GO TO 52
        IF (ABS((A(I,I)-VAL(I))/A(I,I)) .GT. CTVAL) GO TO 55
52  CONTINUE
        GO TO 60
55  DO 56 I=1,N
56  VAL(I) = A(I,I)
        GO TO 20
C   PLACE DIAGONAL FROM A INTO VAL (EIGENVALUES).
60  DO 61 I=1,N

```



```
61 VAL(I) = A(I,I)  
RETURN
```

C

```
END
```

```

SUBROUTINE FRI (A,B,C,D,TABW,TABF,W,NX,NW,NRTABF,NCTABF,
*             KA,KF,WRK,NTAPE)
DIMENSION A(KA,1),E(KA,1),C(KA,1),D(KA,1),TABW(KF,1),TABF(KF,1),
*         W(1),WRK(KA,1),F(80),XR(50),XI(50),GN(50),GNDB(50),
*         PHAS(50),WW(50),1V(50),IRE(50),BIN(50),U(50)
DATA NIT,NOT/5,6/
DATA NLPP / 60 /

```

```

C
C FREQUENCY RESPONSE ROUTINE TO SOLVE THE DIFFERENTIAL EQUATION
C   (-W**2*A + IW*B +C)*X(W) = D*F(W) FOR X(W).
C MATRIX B MUST BE NON-SINGULAR.
C VECTOR F IS OBTAINED BY LINEAR INTERPOLATION USING TABW, TABF.
C THE ANSWERS (F,XREAL,XIMAG,GAIN,GAIN(DECIBELS),PHASE ANGLE) WILL BE
C WRITTEN ON PAPER AND NTAPE EVERY W (OMEGA).
C CALLS FORMA SUBROUTINES INVI,MULT,MULTB,PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE
C   NX      = 50
C   NRTABF  = 80
C CODED BY CARL EODLEY. AUGUST 1965.
C LAST REVISION BY J ERNST, OCT 1973.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A      = MATRIX COEF OF -W**2. SIZE (NX,NX). *DESTROYED*
C B      = MATRIX COEF OF IW.   SIZE (NX,NX). *DESTROYED*
C C      = MATRIX.              SIZE (NX,NX). *DESTROYED*
C D      = MATRIX COEF OF F.    SIZE (NX,NRTABF). *DESTROYED*
C TABW   = TABLE OF OMEGAS FOR FORCE IN TABF. SIZE (NRTABF,NCTABF).
C         OMEGA IS IN RADIAN/SEC.
C TABF   = TABLE OF FORCES AT OMEGA IN TABW. SIZE (NRTABF,NCTABF).
C W      = VECTOR OF FREQUENCIES AT WHICH EQUATION IS SOLVED. SIZE(NW).
C         OMEGA IS IN RADIAN/SEC.
C NX     = SIZE OF MATRICES A,B,C,WRK, (SQUARE). NUMBER OF ROWS IN D.
C         MAX=50.
C NW     = SIZE OF VECTOR W.
C NRTABF = NUMBER OF ROWS IN TABW,TABF. NUMBER OF COLS IN D.
C         MAX=80.
C NCTABF = NUMBER OF COLS IN TABW,TABF.
C KA     = ROW DIMENSION OF A,B,C,D,WRK IN CALLING PROGRAM.
C KF     = ROW DIMENSION OF TABW,TABF IN CALLING PROGRAM.
C WRK    = WORKSPACE MATRIX. SIZE (NX,NX)
C NTAPE  = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 4)
C
C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE AT EACH OMEGA) IS
C W      = OMEGA. SCALAR. RADIAN/SEC.
C F      = FORCE OBTAINED BY LINEAR INTERPOLATION ON TABF.
C         SIZE(NRTABF).
C XR     = X(REAL).           SIZE(NX).
C XI     = X(IMAG).          SIZE(NX).
C GN     = GAIN (SQRT(XR**2+XI**2)). SIZE(NX).
C GNDB   = GAIN(DECIBELS).    SIZE(NX).
C PHAS   = PHASE ANGLE (DEGREES). SIZE(NX).
C
C 2010 FORMAT (//10X, 7HFREQ = F10.4,8H RPS, = ,F10.4,4H CPS)
C 2015 FORMAT (//10X,49HR*A INVERSION CHECK. MAXIMUM DIAGONAL ERROR =
*           E11.3, 6H AT (,I3,1H,,I3,1H),

```

```

          *          /3IX,          28HMAXIMUM OFF-DIAGONAL ERROR =
          *          F11.3, 6H AT (,I3,1H,,I3,1H) )
2020 FORMAT (//10X,14HAPPLIED FORCES / (10X, 5E16.8))
2030 FORMAT (//10X,3HROW,9X,7HX(REAL),13X,7HX(IMAG),12X,9HAMP RATIO,
          *          9X,14HAMP RATIO (DB),6X,11HPHASE (DEG),
          *          //(10X,I3,4E20.8,5X,F11.6))
C
                                                                 NERROR=1
          IF (NX.GT.50 .OR. NRTABF.GT.80) GO TO 999
C
          REWIND NTAPE
          WRITE (NTAPE) ((A(I,J),I=1,NX),J=1,NX)          A=A
          REWIND NTAPE
          CALL INVT (B,A, NX, KA)          A=BI
          CALL MULTB (A,C,NX,NX,NX,KA,KA)          C=BIC
          CALL MULTB (A,D,NX,NX,NRTABF,KA,KA)          D=BID
          DO 15 I=1,NX
          DO 15 J=1,NX
15 B(I,J)=A(I,J)          B=BI
          READ (NTAPE) ((A(I,J),I=1,NX),J=1,NX)          A=A
          CALL MULTB (B,A,NX,NX,NX,KA,KA)          A=BI
          REWIND NTAPE
C
          DO 500 L=1,NW
C CHECK OMEGA TABLE (TABW) AND INTERPOLATE (LINEAR).
          DO 28 I=1,NRTABF
                                                                 NERROR=2
          IF (W(L).LT.TAPW(I,1)) GO TO 999
          DO 22 J=2,NCTABF
                                                                 NERROR=3
          IF (TABW(I,J-1).GE.TABW(I,J) .AND. W(L).GT.TABW(I,J-1)) GO TO 999
          IF (W(L) .LT. TABW(I,J)) GO TO 28
22 CONTINUE
                                                                 NERROR=4
          GO TO 999
          28 F(I) = TABF(I,J-1) + (W(L)-TABW(I,J-1)) * (TABF(I,J)-TABF(I,J-1))/
          1          (TABW(I,J)-TABW(I,J-1))
C
          CALL MULT (D,F,XI,NX,NRTABF,1,KA,KF)          XI=BIDF
          DO 30 I=1,NX
          DO 30 J=1,NX
          B(I,J) = C(I,J) - (W(L)**2)*A(I,J)          B=C-A*W2
30 WRK(I,J)=B(I,J)          WRK=B
          CALL MULT (B,XI,XR,NX,NX,1,KA,KA)          XR=B*X
          DO 40 I=1,NX          XI=-W*
          40 XI(I) = -W(L)*XI(I)          EIDF
          CALL MULTB (WRK,B,NX,NX,NX,KA,KA)          B=B**2
          DO 50 I=1,NX
          50 B(I,I) = B(I,I) + W(L)**2          B=B+W2
C
C BEGIN INVERSION (STATEMENTS FROM INV2).
                                                                 NERROR=5
          IF (NX .EQ. 1 .AND. B(1,1) .EQ. 0.0) GO TO 999
          IF (NX .EQ. 1 .AND. B(1,1) .NE. 0.0) GO TO 153
          IT = 1

```

```
GO TO 60
55 IT = 2
60 DO 65 I=1,NX
   IRE(I) = 1
65 IV(I) = I
   NM1 = NX - 1
   DO 75 L2=1,NM1
     SMAX = 0.0
     DO 70 J=L2,NX
       LA = IRE(J)
       I = L2
       K = LA
       IF (IT .EQ. 2) I = LA
       IF (IT .EQ. 2) K = L2
       IF (ABS(B(K,I)) .LE. SMAX) GO TO 70
       JMAX = J
       SMAX = ABS(B(K,I))
70 CONTINUE
   LS = IRE(L2)
   IRE(L2) = IRE(JMAX)
75 IRE(JMAX) = LS
   DO 80 L2=1,NX
     LA = IRE(L2)
     BIN(L2) = B(LA,L2)
     IF (IT .EQ. 2) BIN(L2) = B(L2,LA)
80 IF (BIN(L2) .EQ. 0.0) BIN(L2) = 1.0
   DO 90 L2=1,NX
     LA = IRE(L2)
     I = L2
     K = LA
     IF (IT .EQ. 2) I = LA
     IF (IT .EQ. 2) K = L2
     DO 85 J=1,NX
       M = J
       M1 = LA
       IF (IT .EQ. 2) M = LA
       IF (IT .EQ. 2) M1 = J
85 WRK(M,M1) = 0.0
     WRK(I,K) = 1.0/BIN(L2)
90 B(K,I) = B(K,I) - BIN(L2)
   DO 120 L2=1,NX
     SMAX = 0.0
     DO 100 J=L2,NX
       LA = IV(J)
       S = 1.0
       DO 95 K=1,NX
95 S = S + B(LA,K)*WRK(K,LA)
       IF (ABS(S) .LE. SMAX) GO TO 100
       LMAX = J
       SMAX = ABS(S)
100 CONTINUE
   IF (SMAX .GT. 1.0E-99) GO TO 105

   IF (IT .EQ. 2) GO TO 999
   GO TO 125
```

```

105 LS = IV(L2)
    IV(L2) = IV(LMAX)
    IV(LMAX) = LS
    LA = IV(L2)
    DO 110 I=1,NX
    WW(I) = 0.0
    DO 110 J=1,NX
110 WW(I) = WW(I) + B(LA,J)*WRK(J,I)
    S = 1.0 + WW(LA)
    DO 115 I=1,NX
115 U(I) = WRK(I,LA)
    DO 120 I=1,NX
    DO 120 J=1,NX
120 WRK(I,J) = WRK(I,J) - U(I)*WW(J)/S
125 DO 130 L2=1,NX
    LA = IRE(L2)
    I = L2
    K = LA
    IF (IT .EQ. 2) I = LA
    IF (IT .EQ. 2) K = L2
130 B(K,3) = B(K,I) - BIN(L2)
    IF (SMAX .LE. 1.0E-99) GO TO 55
    DIAGER = 0.
    IDIAG = 1
    XOFF = 0.0
    IOFF = 1
    JOFF = 1
    DO 150 J=1,NX
    DO 145 I=1,NX
    X = 0.0
    DO 135 K=1,NX
135 X = X + WRK(I,K)*B(K,J)
    IF (I .NE. J) GO TO 140
    IF (ABS(X-1.) .LT. DIAGER) GO TO 145
    DIAGER = ABS(X-1.)
    IDIAG = I
    GO TO 145
140 IF (ABS(X) .LT. ABS(XOFF)) GO TO 145
    XOFF = X
    IOFF = I
    JOFF = J
145 CONTINUE
150 CONTINUE
    GO TO 155
153 WRK(1,1) = 1.0/B(1,1)
    DIAGER = B(1,1)*WRK(1,1) - 1.
    IDIAG = 1
    XOFF = 0.
    IOFF = 0
    JOFF = 0
155 CALL MULTP (WRK,XP,NX,NX,1,KA,KA)
    CALL MULTB (WRK,XI,NX,NX,1,KA,KA)

```

XR=XREAL  
XI=XIMAG

C GET GAIN, PHASE ANGLE.  
DO 180 I=1,NX

```

GN(I) = SQRT(XR(I)**2 + XI(I)**2)
GNDB(I) = 0.0
IF (GN(I) .GT. 0.0) GNDB(I)=20.0*ALOG10(GN(I))
PHAS(I) = 0.0
180 IF (XI(I) .NE. 0.0 .OR. XR(I) .NE. 0.0) PHAS(I)=
1 57.29578*ATAN2(XI(I),XR(I))

```

```

C
C PRINT ANSWERS ON PAPER. HEAD A NEW PAGE EACH OMEGA.
CALL PAGEHD
FREQ= W(L)/6.283185
WRITE (NOT,2010) W(L), FREQ
WRITE (NOT,2015) DIAGER,IDIAG,IDIAG,XOFF,IOFF,JOFF
WRITE (NOT,2020) (F(I),I=1,NRTABF)
NXS = 1
NXE = NX
NFLN = (NRTABF-1)/5 + 1
IF ((NXE + NFLN) .GT. (NLPP-19)) NXE=(NLPP-19)-NFLN
190 WRITE (NOT,2030) (I,XR(I),XI(I),GN(I),GNDB(I),PHAS(I),I=NXS,NXE)
IF (NX .EQ. NXE) GO TO 200
NXS = NXE + 1
NXE = NX
IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
CALL PAGEHD
GO TO 190

```

```

C
C WRITE ANSWERS ON NTAPE FOR SUBSEQUENT USE (SUCH AS
FREQUENCY RESPONSE ADDITIONAL EQUATIONS OR PLOT).
200 WRITE (NTAPE) W(L), (F(J),J=1,NRTABF)
WRITE (NTAPE) (XR(I),XI(I),GN(I),GNDB(I),PHAS(I),I=1,NX)

```

```

C
500 CONTINUE
RETURN

```

```

C
999 CALL ZZBOMB (6HFR1 ,NERROR)
END

```

```

SUBROUTINE FRAEI (A,STA,ZIDENT,NZ,NX,NW,KA,NXTAPE,NZTAPE)
DIMENSION A(KA,1),STA(1),XR(50),XI(50),ZR(80),ZI(80),GN(80),
*         GNDB(80),PHAS(80),ZIDENT(12)
DATA NIT,NOT/5,6/
DATA NLPP / 60 /

C
C FREQUENCY RESPONSE ADDITIONAL EQUATIONS ROUTINE.
C SOLVE THE COMPLEX MATRIX EQUATION  Z(W) = A*X(W).
C W=OMEGA, X(W)=(XREAL,XIMAG) IS OBTAINED FROM NXTAPE (OUTPUT OF
C FREQUENCY RESPONSE SUBROUTINE).
C THE ANSWERS (W,ZREAL,ZIMAG,GAIN,GAIN(DECIBELS),PHASE ANGLE) WILL BE
C WRITTEN ON PAPER AND NZTAPE EVERY W=OMEGA OF FREQ RESP SUBRT.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE
C     NZ = 80
C     NX = 50
C CODED BY RL WOHLN.  AUGUST 1965.
C
C     SUBROUTINE ARGUMENTS (ALL INPUT)
C A      = MATRIX COEF OF X(W).  SIZE(NZ,NX)
C STA    = VECTOR OF STATIONS.  SIZE(NZ).  (A6 FORMAT).
C ZIDENT = HEADING FOR Z IN OUTPUT DATA.  (UP TO 12A6 FORMAT).
C NZ     = NUMBER OF ROWS OF MATRIX A.  MAX=80.
C NX     = NUMBER OF COLS OF MATRIX A.  MAX=50.
C NW     = NUMBER OF OMEGAS TO BE READ FROM NXTAPE.
C KA     = ROW DIMENSION OF A IN CALLING PROGRAM.
C NXTAPE = NUMBER OF TAPE FROM WHICH W,X WILL BE READ.  (E.G. 3).
C NZTAPE = NUMBER OF TAPE ON WHICH W,ZR,ZI,GN,GNDB,PHAS
C        WILL BE WRITTEN (EG 4).  IF NZTAPE=0 BYPASS WRITING ON NZTAPE
C
C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NZTAPE EACH OMEGA) IS
C W      = OMEGA.  SCALAR.  RADIAN/SEC.
C ZR     = Z(REAL).           SIZE(NZ).
C ZI     = Z(IMAG).           SIZE(NZ).
C GN     = GAIN (SQRT(ZR**2+ZI**2)).  SIZE(NZ).
C GNDB   = GAIN(DECIBELS).     SIZE(NZ).
C PHAS   = PHASE ANGLE (DEGREES).  SIZE(NZ).
C
C 2010 FORMAT (/15X, 12A6, //10X, 7HFEQ = F10.4,8H RPS, = F10.4,4H CPS)
C 2030 FORMAT (//10X,3HROW,3X,7HSTATION,9X,7HZ(REAL),13X,7HZ(IMAG),
*           12X,9HAMP RATIO,9X,14HAMP RATIO (DB),6X,11HPHASE (DEG),
*           //(10X,13,4X,A6,4E20.8,5X,F11.6))

C
C                                     NERROR=1

C IF (NX .GT. 50 .OR. NZ .GT. 80) GO TO 999
C REWIND NXTAPE
C IF (NZTAPE .GT. 0) REWIND NZTAPE

C
C DO 100 I=OMEGA = 1,NW
C   READ (NXTAPE) W
C   READ (NXTAPE) (XR(I),XI(I),DUM,DUM,DUM, I=1,NX)
C   DO 10 I=1,NZ
C     ZR(I) = 0.0
C     ZI(I) = 0.0
C   DO 10 J=1,NX

```

```

      ZR(I) = ZR(I) + A(I,J) * XP(J)
10  ZI(I) = ZI(I) + A(I,J) * XI(J)
C
C GET GAIN, PHASE ANGLE.
      DO 55 I=1,NZ
      GN(I) = SQRT(ZR(I)**2 + ZI(I)**2)
      GNDB(I) = 0.0
      IF (GN(I) .GT. 0.0) GNDB(I)=20.0*ALOG10(GN(I))
      PHAS(I) = 0.0
55  IF (ZI(I) .NE. 0.0 .OR. ZR(I) .NE. 0.0) PHAS(I)=
      1                                     57.29578*ATAN2(ZI(I),ZR(I))
C
C PRINT ANSWERS ON PAPER. HEAD A NEW PAGE EACH OMEGA.
      CALL PAGEHD
      FREQ= W/6.283185
      WRITE (NOT,2010) ZIDENT, W, FREQ
      NZS = 1
      NZE = NZ
      IF (NZE .GT. (NLPP-13)) NZE=NLPP-13
60  WRITE (NOT,2030) (I,STA(I),ZR(I),ZI(I),GN(I),GNDB(I),PHAS(I),
      *                               I=NZS,NZE)
      IF (NZ .EQ. NZE) GO TO 80
      NZS = NZE + 1
      NZE = NZ
      IF ((NZE-NZS) .GT. (NLPP-10)) NZE=NZS-(NLPP-10)
      CALL PAGEHD
      GO TO 60
C
C WRITE ANSWERS ON NZTAPE FOR SUBSEQUENT USE (SUCH AS PLOTTING).
80  IF (NZTAPE .GT. 0) WRITE (NZTAPE) W,(STA(I),ZR(I),ZI(I),GN(I),
      *                               GNDB(I),PHAS(I), I=1,NZ)
C
100 CONTINUE
      RETURN
C
999 CALL ZZBOMB (6HFRAE1 ,NERROR)
      END

```



IN

-----

SUBROUTINE IN (NTAPE,Z,N)  
DIMENSION Z(1)

C  
C READ DATA FROM NTAPE INTO CORE SPACE Z.  
C CODED BY RL WOHLER. MARCH 1976.  
C  
C SUBROUTINE ARGUMENTS  
C NTAPE = INPUT NUMBER OF TAPE. (EG 10).  
C Z = OUTPUT DATA READ FROM TAPE.  
C N = INPUT NUMBER OF WORDS OF DATA TO BE READ FROM NTAPE.  
C  
C READ (NTAPE) (Z(1),1=1,N)  
C RETURN  
C END

## INTAPE

-----

```
      SUBROUTINE INTAPE (NTAPE,TAPEID)
      COMMON /LLINE/ NLINE,MAXLIN,MINI
      DATA IZ1,BUF,ECT/1,0.,3HEOT/
      DATA NIT,NOT/5,6/
C
C  INITIALIZE TAPE FOR SUBROUTINE WTAPE.
C  CALLS FORMA SUBROUTINE PAGEHD.
C  CODED BY  RF HRUDA.  JULY 1968.
C  LAST REVISION BY RL WOHLN.  APRIL 1976.
C
C      SUBROUTINE ARGUMENTS (ALL INPUT)
C  NTAPE = NUMBER OF TAPE. (E.G. 10).
C  TAPEID = TAPE IDENTIFICATION. (E.G. T1234). (A6 FORMAT).
C
C 2001 FORMAT (//// 14H LOGICAL UNIT I2, 7H, TAPE A6,
*           23H, HAS BEEN INITIALIZED.)
C 2050 FORMAT (/ 1X 123(1H-) )
C
      REWIND NTAPE
      WRITE (NTAPE) TAPEID, IZ1, ECT, (BUF, I=1, 16)
      ENDFILE NTAPE
      REWIND NTAPE
      IF (MINI .NE. 4*MINI) GO TO 800
      IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
      IF ((NLINE+2+5 ) .GT. MAXLIN) GO TO 800
      WRITE (NOT, 2050)
      NLINE = NLINE + 2
      GO TO 810
C 800 CALL PAGEHD
C 810 WRITE (NOT, 2001) NTAPE, TAPEID
      NLINE = NLINE + 5
C
      RETURN
      END
```

```
SUBROUTINE INVI (A,Z,N,KR)  
DIMENSION A(1), Z(1)  
COMMON /LWRKV1/ G(250), DETR(250)  
COMMON /LWRKV2/ IX(250), B(250)  
COMMON /LLINE/ NLINE,MAXLIN,MINI  
DATA NIT,NOT/5,6/
```

```
C  
C MATRIX INVERSION (A**-1 = Z). BORDERING METHOD.  
C THE DETERMINANT RATIO DET(I+1) / DET(I) IS PRINTED. DET(I) IS THE  
C DETERMINANT OF THE FIRST I BY I SUB-MATRIX OF A.  
C THE INVERSION CHECK Z*A IS CALCULATED AND PRINTED.  
C MATRICES A,Z MAY SHARE SAME CORE LOCATIONS. (Z*A CHECK IS INVALID).  
C CALLS FOR SUBROUTINES PAGEHD,ZZBOMB.  
C THE MAXIMUM SIZE IS  
C N = 250  
C DEVELOPED BY BOB DILLON. FEBRUARY 1965.  
C LAST REVISION BY RL WOHLN. MARCH 1976.
```

```
C  
C SUBROUTINE ARGUMENTS  
C A = INPUT MATRIX TO BE INVERTED. SIZE(N,N).  
C Z = OUTPUT RESULT MATRIX. SIZE(N,N).  
C N = INPUT SIZE OF MATRICES A,Z. MAX=250.  
C KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.
```

```
C  
C ERROR EXPLANATION  
C 1 = SIZE GREATER THAN 250.  
C 2 = FIRST COLUMN IS ZERO.  
C 3 = MATRIX IS SINGULAR.
```

```
C  
2000 FORMAT (// 10X,10(7X,1H(,12,1H)))  
2001 FORMAT (// 10X,45H SUBROUTINE INVI HAS CALCULATED THE DATA BELOW  
* // 10X,44H THE DETERMINANT RATIOS DET(I+1) / DET(I) ARE  
* // (13X,10F11.3))  
2002 FORMAT (/// 10X,37H THE (A**-1)*(A) INVERSION CHECK GIVES  
* // 10X,25H THE DIAGONAL ELEMENTS ARE // (13X,10F11.6))  
2003 FORMAT (// 10X,35H THE MAXIMUM OFF-DIAGONAL ELEMENT IS  
* E11.3, 2X, 4H AT ( 13, 1H, 13, 1H) )  
2050 FORMAT (/ 1X 123(1H-))
```

```
C  
NERROR=1  
IF (N .GT. 250) GO TO 999
```

```
C  
DO 160 I=2,N  
160 IX(I) = 1  
C INVERT FIRST NON-ZERO ELEMENT IN FIRST COLUMN.  
DO 190 I=1,N  
IF (A(I) .NE. 0.) GO TO 220  
190 CONTINUE  
GO TO 999
```

```
NERROR=2
```

```
C  
START INVERSION WITH ROW 1.  
220 DETR(1) = A(1)  
Z(1) = 1. / A(1)  
IF (N .EQ. 1) RETURN
```

```

IX(I) = 1
IX(1) = I
C BORDERING LOOP.
DO 630 L=2,N
  K = L
  LI = L - 1
250 S = 0.
      MIXL = KR * (IX(L) - 1)
      LL = IX(L) + MIXL
DO 450 I=1,LI
      MIXI = KR * (IX(I) - 1)
      LI = IX(L) + MIXI
B(I) = 0.
G(I) = 0.
DO 440 J=1,LI
      MIXJ = KR * (IX(J) - 1)
      IJ = IX(I) + MIXJ
      JL = IX(J) + MIXL
B(I) = B(I) - Z(IJ)* A(IJ)
      JI = IX(I) + MIXI
      LJ = IX(L) + MIXJ
440 G(I) = G(I) - A(LJ)* Z(JI)
450 S = S + A(LI)* B(I)
AL = A(LL) + S
IF (A(LL) .EQ. 0.) GO TO 480
ALBAR = ABS (AL / A(LL))
GO TO 490
480 ALBAR = ABS (AL)
490 IF (ALBAR .GE. .1E-6) GO TO 550
C
C INTERCHANGE ROWS AND COLUMNS.
K = K + 1
IF (K .GT. N) GO TO 540
IX L = IX(L)
IX(L) = IX(K)
IX(K) = IX L
GO TO 250
540 IF (ALBAR .GE. .1E-8) GO TO 550
GO TO 999
C
550 Z(LL) = 1. / AL
DETR(L) = AL
DO 570 I=1,LI
      IL = IX(I) + MIXL
      LI = IX(L) + KR * (IX(I) - 1)
      Z(IL) = B(I) * Z(LL)
      Z(LI) = G(I) * Z(LL)
DO 570 J=1,LI
      IJ = IX(I) + KR * (IX(J) - 1)
570 Z(IJ) = Z(IJ) + G(J) * Z(IL)
630 CONTINUE
C
C COMPUTE INVERSION CHECK Z* A.

```

NERROR=3

```

XOFF = 0.0
DO 720 I=1,N
DO 710 J=1,N
X = 0.0
KJA = KR * (J-1)
DO 703 K=1,N
IK = I + KR*(K-1)
KJ = K + KJA
703 X = X + Z(IK) * A(KJ)
IF (I .NE. J) GO TO 705
G(I) = X
GO TO 710
705 IF (ABS(X) .LT. ABS(XOFF)) GO TO 710
XOFF = X
IOFF = I
JOFF = J
710 CONTINUE
720 CONTINUE

```

C

C PRINT THE DETERMINANT RATIO AND INVERSION CHECK.

```

NPL = N/10
IF ((NPL*10) .NE. N) NPL = NPL+1
NNL = 2*NPL + 21
IF (MINI .NE. 4*MINI) GO TO 800
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
IF ((NLINE+2+NNL) .GT. MAXLIN) GO TO 800
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2000) (JC, JC=1,10)
WRITE (NOT,2001) (DETR(I), I=1,N)
WRITE (NOT,2002) ( G (I), I=1,N)
WRITE (NOT,2003) XOFF,IOFF,JOFF
NLINE = NLINE +>NNL
RETURN

```

C

```

999 CALL ZZBOR. (6HINVI ,NERROR)
END

```

\*

```

SUBROUTINE INV2 (A,Z,N,KR)
DIMENSION A(KR,1), Z(KR,1), W(250)
COMMON /LWRKV1/ IRE(250), BIN(250)
COMMON /LWPKV2/ U(250), IV(250)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION DM,DS,ZERO,ONE
DATA ZERO/0.0/, ONE/1.0/
DATA NIT,NOT/5,6/

C
C MATRIX INVERSION (A**=-1 = Z). RANK ANNIHILATION METHOD.
C ALGORITHM FORMULATED BY CARL BODLEY.
C THE INVERSION CHECK Z*A IS CALCULATED AND PRINTED.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C THE MAXIMUM SIZE IS
C N = 250
C DEVELOPED BY CARL BODLEY. JANUARY 1967.
C LAST REVISION BY RL WOHLN. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX TO BE INVERTED. SIZE(N,N).
C Z = OUTPUT RESULT MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,Z. MAX=250.
C KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = SIZE GREATER THAN 250.
C 2 = MATRIX IS SINGULAR. SIZE = 1.
C 3 = MATRIX IS SINGULAR.
C
2000 FORMAT (// 10X,10(7X,1H(,I2,1H)))
2001 FORMAT (// 10X,45H SUBROUTINE INV2 HAS CALCULATED THE DATA BELOW )
2002 FORMAT (///10X,37H THE (A**=-1)*(A) INVERSION CHECK GIVES
* //10X,25H THE DIAGONAL ELEMENTS ARE // (13X,10F11.8))
2003 FORMAT (// 10X,35H THE MAXIMUM OFF-DIAGONAL ELEMENT IS
* F11.3, 2X, 4H AT ( I3, 1H, I3, 1H )
2050 FORMAT (/ 1X 123(1H-))
C
IF (N .GT. 250) GO TO 999
NERROR=1
IF (N .EQ. 1 .AND. A(1,1) .EQ. 0.0) GO TO 999
IF (N .EQ. 1 .AND. A(1,1) .NE. 0.0) GO TO 98
NERROR=2
C
C GENERATE INITIAL ROW INDICES.
IT = 1
GO TO 90
91 IT = 2
90 DO 5 I=1,N
IRF(I) = I
5 IV(I) = I
C
CONDITION A FOR MAXIMUM DIAGONAL ELEMENTS.
NM1 = N - 1
DO 6 L=1,NM1

```

```

SMAX = 0.0
DO 8 J=L,N
LA = IRE(J)
I = L
K = LA
IF (IT .EQ. 2) I = LA
IF (IT .EQ. 2) K = L
IF (ABS(A(K,I)) .LE. SMAX) GO TO 8
JMAX = J
SMAX = ABS(A(K,I))
8 CONTINUE
LS = IRE(L)
IRE(L) = IRE(JMAX)
6 IRE(JMAX) = LS
DO 7 L=1,N
LA = IRE(L)
BIN(L) = A(LA,L)
IF (IT .EQ. 2) BIN(L) = A(L,LA)
7 IF (BIN(L) .EQ. 0.0) BIN(L) = 1.0
C
C GENERATE INITIAL Z AND ABAR.
DO 10 L=1,N
LA = IRE(L)
I = L
K = LA
IF (IT .EQ. 2) I = LA
IF (IT .EQ. 2) K = L
DO 15 J=1,N
M = J
MI = LA
IF (IT .EQ. 2) M = LA
IF (IT .EQ. 2) MI = J
15 Z(M,MI) = 0.0
Z(I,K) = 1.0/BIN(L)
10 A(K,I) = A(K,I) - BIN(L)
C
C INVERSION LOOP, USES ROW OF ABAR WITH MAXIMUM S.
DO 35 L=1,N
SMAX = 0.0
DO 23 J=L,N
LA = IV(J)
DS = ONE
DO 26 K=1,N
DM = A(LA,K)*Z(K,LA)
26 DS = DS + DM
S = DS
IF (ABS(S) .LE. SMAX) GO TO 23
LMAX = J
SMAX = ABS(S)
23 CONTINUE
IF (SMAX .GT. 1.0E-99) GO TO 60

IF (IT .EQ. 2) GO TO 999
GO TO 65
60 LS = IV(L)

```

NERROR=3

```
      IV(L) = IV(LMAX)
      IV(LMAX) = LS
      LA = IV(L)
      DO 25 I=1,N
      DS = ZERO
      DO 24 J=1,N
      DM = A(LA,J)*Z(J,I)
24  DS = DS + DM
25  W(I) = DS
      S = 1.0 + W(LA)
      DO 30 I=1,N
30  U(I) = Z(I,LA)
      DO 35 I=1,N
      DO 35 J=1,N
35  Z(I,J) = Z(I,J) - U(I)*W(J)/S
C
C  RESTORE A.
65  DO 40 L=1,N
      LA = IRE(L)
      I = L
      K = LA
      IF (IT .EQ. 2) I = LA
      IF (IT .EQ. 2) K = L
40  A(K,I) = A(K,I) + BIN(L)
      IF (SMAX .LE. 1.0E-99) GO TO 91
C
C  COMPUTE INVERSION CHECK Z*A.
      XOFF = 0.0
      DO 50 J=1,N
      DO 45 I=1,N
      DS = ZERO
      DO 46 K=1,N
      DM = Z(I,K)*A(K,J)
46  DS = DS + DM
      X = DS
      IF (I .NE. J) GO TO 47
      U(I) = X
      GO TO 45
47  IF(ABS(X) .LT. XOFF) GO TO 45
      XOFF = X
      IOFF = I
      JOFF = J
45  CONTINUE
50  CONTINUE
C
C  PRINT INVERSION CHECK AND MAXIMUM OFF-DIAGONAL ELEMENT.
      NPL = N/10
      IF ((NPL*10) .NE. N) NPL = NPL+1
     >NNL = NPL + 17
      IF (MINI .NE. 4HMINI) GO TO 800
      IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
      IF ((NLINE+2>NNL) .GT. MAXLIN) GO TO 800
      WRITE (NC7,2050)
     >NNLINE =>NNLINE + 2
      GO TO 810
```



```

800 CALL PAGEHD
810 WRITE (NOT,2000) (JC, JC=1,10)
    WRITE (NOT,2001)
    WRITE (NOT,2002) ( U (I), I=1,N)
    WRITE (NOT,2003) XOFF,I0FF,J0FF
    NLINE = NLINE +>NNL
    RETURN

```

```

C
98 Z(1,1) = 1.0/A(1,1)
    RETURN

```

```

C
999 CALL ZZBOMB (6HINV2 ,NERROR)
    END

```

\*

```

SUBROUTINE INV3 (A,Z,N,KR)
DIMENSION A(KR,1), Z(KR,1), W1(250)
COMMON /LWPKV1/ W2(250), DETR(250)
COMMON /LLINF/ NLINE,MAXLIN,MINI
DOUBLE PRECISION DM,DS,ZERO
DATA ZERO /0.0/
DATA NIT,NCT/5,6/

```

```

C
C MATRIX INVERSION (A*-1 = Z). METHOD USES TRIANGULAR DECOMPOSITION
C AND TRIANGULAR INVERSION. MATRIX A SHOULD BE SYMMETRIC, POSITIVE
C DEFINITE. UPPER HALF OF MATRIX A IS USED TO CALCULATE Z. FULL
C MATRICES A,Z ARE USED FOR INVERSION CHECK.
C THE DETERMINANT RATIO DET(I+1) / DET(I) IS PRINTED. DET(I) IS THE
C DETERMINANT OF THE FIRST I BY I SUB-MATRIX OF A.
C THE INVERSION CHECK Z*A IS CALCULATED AND PRINTED.
C MATRICES A,Z MAY SHARE SAME CORE LOCATIONS. (Z*A CHECK IS INVALID).
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES DCOM1,INV4,PAGEHD,ZZBOMB.
C THE MAXIMUM SIZE IS
C   N = 250
C DEVELOPED BY CARL BODLEY. MARCH 1969.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX TO BE INVERTED. SIZE(N,N).
C Z = OUTPUT RESULT MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,Z. MAX=250.
C KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.

```

```

C ERROR EXPLANATION
C I = SIZE GREATER THAN 250.

```

```

C
2000 FORMAT (// 10X,10(7X,1H(,I2,1H)))
2001 FORMAT (// 10X,45H SUBROUTINE INV3 HAS CALCULATED THE DATA BELOW
*          // 10X,44H THE DETERMINANT RATIOS DET(I+1) / DET(I) ARE
*          // (13X,10E11.3))
2002 FORMAT (/// 10X,37H THE (A*-1)*(A) INVERSION CHECK GIVES
*          // 10X,25H THE DIAGONAL ELEMENTS ARE // (13X,10F11.8))
2003 FORMAT (// 10X,35H THE MAXIMUM OFF-DIAGONAL ELEMENT IS
*          F11.3, 2X, 4H AT ( I3, 1H, I3, 1H )
2050 FORMAT (/ 1X 123(1H-) ,

```

NERROR=1

```

IF (N .GT. 250) GO TO 999

```

```

C
CALL DCOMDM (A,Z,N,KR)
DO 5 I=1,N
5 DETR(I) = Z(I,I)*Z(I,I)
CALL INV4DM (Z,Z,N,KR)
DO 40 L=1,N
DO 20 I=L,N
20 W1(I) = Z(L,I)
DO 35 I=1,N
DS = ZERO
DO 30 K=1,N

```

```

      DM = Z(I,K)*W1(K)
30 DS = DS + DM
35 W2(I) = DS
      JO 40 K=1,N
40 Z(K,I) = W2(K)

```

C

C CALCULATE INVERSION CHECK Z\*A.

```

      XOFF = 0.0
      DO 120 I=1,N
      DO 110 J=1,N
      DS = ZERO
      DO 105 K=1,N
      DM = Z(I,K)*A(K,J)
105 DS = DS + DM
      X = DS
      IF (I .NE. J) GO TO 108
      W1(I) = X
      GO TO 110
108 IF (ABS(X) .LT. ABS(XOFF)) GO TO 110
      XOFF = X
      IOFF = I
      JOFF = J
110 CONTINUE
120 CONTINUE

```

C

C PRINT THE DETERMINANT RATIOS AND INVERSION CHECK.

```

      NPL = N/10
      IF ((NPL*10) .NE. N) NPL = NPL+1
     >NNL = 2*NPL + 21
      IF (MINI .NE. 4*MINI) GO TO 800
      IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
      IF ((NLINE+2>NNL) .GT. MAXLIN) GO TO 800
      WRITE (NOT,2050)
      NLINE = NLINE + 2
      GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2000) (JC, JC=1,10)
      WRITE (NOT,2001) (DETR(I), I=1,N)
      WRITE (NOT,2002) (W1 (I), I=1,N)
      WRITE (NOT, 003) XOFF,IOFF,JOFF
      NLINE = NLINE +>NNL
      RETURN

```

C

```

999 CALL ZZPOMP (6HINV3 ,NERRO)
      END

```

INV4  
-----

```

SUBROUTINE INV4 (A,Z,N,KR)
DIMENSION A(KR,1), Z(KR,1)
DOUBLE PRECISION DM,DS
DATA EPS/1.E-35/

```

```

C
C MATRIX INVERSION (A**-1 = Z). MATRIX A IS ASSUMED TO BE
C UPPER TRIANGULAR.
C MATRICES A,Z MAY SHARE SAME CORE LOCATIONS.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY RL WOHLER. JANUARY 1971.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX TO BE INVERTED. SIZE(N,N).
C Z = OUTPUT RESULT MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,Z.
C KR = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.

```

```

C
C ERROR EXPLANATION
C 1 = A DIAGONAL ELEMENT IS LESS THAN 1.E-35.

```

NERROR = 1

```

DO 10 I=1,N
  IF (ABS(A(I,I)) .LT. EPS) GO TO 999
10 Z(I,I) = 1./A(I,I)
  IF (N .EQ. 1) RETURN

```

```

C
  NM1 = N-1
  DO 25 I=1,NM1
    IP1 = I+1
    DO 25 J=IP1,N
      Z(I,J) = Z(I,1)*A(I,J)
      IF (J .EQ. IP1) GO TO 23
      JM1 = J-1
      DS = Z(I,J)
      DO 20 K=IP1,JM1
        DM = Z(I,K)*A(K,J)
      20 DS = DS + DM
      Z(I,J) = DS
      23 Z(I,J) = -Z(I,J)*Z(J,J)
      25 CONTINUE

```

```

C
  DO 30 I=2,N
    IM1 = I-1
    DO 30 J=1,IM1
      30 Z(I,J) = 0.
  RETURN

```

```

C
999 CALL ZZBOMB (6HINV4 ,NERROR)
END

```

SUBROUTINE LTAPE (NTAPE)  
DATA NIT,NOT/5,6/

```

C
C LIST HEADINGS OF MATRICES ON TAPE.
C CALLS FORMA SUBROUTINE PAGEHD.
C CODED BY RF HRUDA. JULY 1967. REVISED NOVEMBER 1970.
C REVISED BY R A PHILIPPUS. APRIL 1969.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C NTAPE = NUMBER OF TAPE. (E.G. 10).
C
2001 FORMAT (//36X35HLISTING OF MATRICES ON LOGICAL UNIT13,7H, TAPE A6)
2002 FORMAT (//30X35HLISTING OF MATRICES ON LOGICAL UNIT13,7H, TAPE A6,
* 12H (CONTINUED))
2003 FORMAT (27X69(1H-)/27X3HNO.3X7HRUN NO.4X4HNAME5X5HNROWS4X5HNCOLS4X
* 4HDATE6X3HNNZ3X9HPARTITION/
* 27X3H---3X6H-----4X6H-----4X5H-----
* 4X5H-----3X6H-----5X3H---3X9H----- / )
2004 FORMAT (25X15,3XA6,4XA6,3XI5,4XI5,4XA6,3XI5,3XI4,1H/I4)
2005 FORMAT (/27X12HEND OF LIST.)
C
REWIND NTAPE
READ (NTAPE) TAPEID
REWIND NTAPE
L=0
C
12 CALL PAGEHD
IF(L .EQ. 0) WRITE (NOT,2001) NTAPE,TAPEID
IF(L .NE. 0) WRITE (NOT,2002) NTAPE,TAPEID
WRITE (NOT,2003)
NLINE=1
13 L=L+1
READ (NTAPE) TAPEID,LN,IEOTCK,IRUNNO,ANAME,NR,NC,DATE,ITYPE,NNZ,
* NP,NPT
IF (L .EQ. 1) ICHK = IRUNNO
IF (ICHK .EQ. IRUNNO) GO TO 15
NLINE=NLINE+1
WRITE (NOT,2004)
ICHK = IRUNNO
15 IF (IEOTCK .EQ. 3HEOT) GO TO 30
READ (NTAPE)
IF (ITYPE .EQ. 6HDENSE ) WRITE (NOT,2004)
* LN,IRUNNO,ANAME,NR,NC,DATE
IF (ITYPE .EQ. 6HDENSE ) GO TO 20
IF (ITYPE .EQ. 6HSPARSE) WRITE (NOT,2004)
* LN,IRUNNO,ANAME,NR,NC,DATE,NNZ
IF (ITYPE .EQ. 6HSPARSE) GO TO 20
IF (ITYPE .EQ. 6HSPART ) WRITE (NOT,2004)
* LN,IRUNNO,ANAME,NR,NC,DATE,NNZ,NP,NPT
IF (ITYPE .EQ. 6HSPART ) GO TO 20
WRITE (NOT,2004) LN,IRUNNO,ANAME,NR,NC,ITYPE
20 NLINE=NLINE+1
IF(NLINE.GT.43) GO TO 12
GO TO 13
C

```

LTape -- 2 / 2

---

```
30 WRITE (NOT,2004) LN,IECTCK
   WRITE (NOT,2005)
   REWIND NTAPE
   RETURN
   END
```

\*

```

SUBROUTINE MASS1 (PP,DMASS,DRIN,CONC,CONVRT,Z,NPP,NDM,NDI,NC,
*           KDM,KDI,KC,KZ)
DIMENSION PP(1), DMASS(KDM,1), DRIN(KDI,1), CONC(KC,1), Z(KZ,1)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C CALCULATE MASS MATRIX FOR A BEAM. ASSUMES LINEAR VELOCITY FUNCTION
C BETWEEN CONSECUTIVE PANEL POINTS.
C TRANSLATION AT EACH PANEL POINT ARE THE GENERALIZED COORDINATES.
C INPUT IS DISTRIBUTED MASS, DISTRIBUTED ROTARY INERTIA, CONCENTRATED
C ITEMS. THE DISTRIBUTED DATA MAY NOT EXCEED THE PANEL POINT LIMITS.
C THE ATTACH POINT FOR CONCENTRATED ITEMS MAY NOT EXCEED THE PANEL
C POINT LIMITS. OPTION TO OMIT DATA BY NDM,NDI, OR NC EQUAL ZERO.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C CODED BY RL WOHLER. DECEMBER 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C PP = INPUT VECTOR OF PANEL POINTS. SIZE(NPP).
C DMASS = INPUT MATRIX OF DISTRIBUTED MASS STRAIGHT LINE
C SEGMENT DATA. SIZE(NDM,4).
C COL 1 = X AT SEGMENT END 1.
C COL 2 = X AT SEGMENT END 2.
C COL 3 = MASS AT SEGMENT END 1.
C COL 4 = MASS AT SEGMENT END 2.
C DRIN = INPUT MATRIX OF DISTRIBUTED ROTARY INERTIA STRAIGHT LINE
C SEGMENT DATA. SIZE(NDI,4).
C COLUMNS ARE SIMILAR TO DMASS.
C CONC = INPUT MATRIX OF CONCENTRATED ITEM DATA. SIZE(NC,4).
C COL 1 = ATTACH STATION.
C COL 2 = MASS OF ITEM.
C COL 3 = CENTER OF GRAVITY OF ITEM.
C COL 4 = MOMENT OF INERTIA ABOUT CG OF ITEM.
C CONVRT = INPUT CONVERSION SCALAR BY WHICH COL 3,4 OF DMASS,DRIN AND
C COL 2,4 OF CONC WILL BE MULTIPLIED.
C Z = OUTPUT MASS MATRIX. SIZE(NPP,NPP).
C NPP = INPUT NUMBER OF PANEL POINTS. SIZE OF VECTOR PP, MATRIX Z.
C NDM = INPUT NUMBER OF SEGMENTS (ROWS) IN DMASS. CAN BE ZERO.
C NDI = INPUT NUMBER OF SEGMENTS (ROWS) IN DRIN. CAN BE ZERO.
C NC = INPUT NUMBER OF ITEMS (ROWS) IN CONC. CAN BE ZERO.
C KDM = INPUT ROW DIMENSION OF DMASS IN CALLING PROGRAM.
C KDI = INPUT ROW DIMENSION OF DRIN IN CALLING PROGRAM.
C KC = INPUT ROW DIMENSION OF CONC IN CALLING PROGRAM.
C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.

```

```

C
C NERROR EXPLANATION
C 1 = LESS THAN 2 PANEL POINTS.
C 2 = PANEL POINTS NOT IN INCREASING ORDER.
C 3 = INCORRECT DISTRIBUTED DATA.
C 4 = CONCENTRATED MASS ATTACH STATION OUTSIDE PANEL
C POINT BOUNDS.

```

```

2001 FORMAT ( 3//,30X,31HSUBROUTINE MASS1 USES CONVRT = E15.8, ///
?           37X,33HAND COMPUTES THE TOTAL PROPERTIES ///
*           43X,6HM = E15.8,///43X,6HXCG = E15.8,///43X,6HXCG = E15.8)

```

2050 FORMAT (/ 1X 123(1H-))

C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER.

IF (NPP .LT. 2) GO TO 999

NERROR = 1

DO 5 K=2,NPP

IF (PP(K-1) .GE. PP(K)) GO TO 999

NERROR = 2

5 CONTINUE

C

C INITIALIZE DATA.

DO 10 I=1,NPP

DO 10 J=1,NPP

10 Z(I,J) = 0.0

NBAYS = NPP-1

C

C DISTRIBUTED MASS (MIC=1), DISTRIBUTED ROTARY INERTIA (MIC=2),

C CONCENTRATED ITEM (MIC=3).

DO 95 MIC=1,3

IF (MIC .EQ. 1) NSEGS = NDM

IF (MIC .EQ. 2) NSEGS = NDI

IF (MIC .EQ. 3) NSEGS = NC

IF (NSEGS .EQ. 0) GO TO 95

C

DO 90 I=1,NSEGS

GO TO (21,22,70),MIC

21 X1 = DMASS(I,1)

X2 = DMASS(I,2)

V1 = DMASS(I,3) \* CONVRT

V2 = DMASS(I,4) \* CONVRT

GO TO 30

22 X1 = DRIN(I,1)

X2 = DRIN(I,2)

V1 = DRIN(I,3) \* CONVRT

V2 = DRIN(I,4) \* CONVRT

30

IF (X1 .LT. PP(1) .OR. X2 .GT. PP(NPP) .OR. X1 .GE. X2) GO TO 999

NERROR = 3

DO 32 K=1,NBAYS

IF (X1 .LT. PP(K+1)) GO TO 34

32 CONTINUE

34 XP = X1

VP = V1

36 IF (X2 .LE. PP(K+1)) GO TO 38

XQ = PP(K+1)

VQ = V1 + (XQ-X1)\*(V2-V1)/(X2-X1)

GO TO 39

38 XQ = X2

VQ = V2

39 BAYL = PP(K+1) - PP(K)

SEGL = XQ-XP

HP = (XP-PP(K)) / BAYL

HQ = (XQ-PP(K)) / BAYL

VPVQ = VP + VQ

GO TO (50,60),MIC

C



```

50 F1 = SEGL*VPVQ/2.
   F2 = SEGL*(VPVQ*(HP+HQ) + VP*HP + VQ*HQ)/6.
   F3 = SEGL*(VPVQ*(HP+HQ)**2 + 2.*(VP*HP**2 + VQ*HQ**2))/12.
   GO TO 80

```

```

C
60 F1 = 0.
   F2 = 0.
   F3 = SEGL*VPVQ/(2.*BAYL**2)
   GO TO 80

```

```

C
70 XA = CONC(I,1)
                                     NERROR = 6
   IF (XA .LT. PP(1) .OR. XA .GT. PP(NPP)) GO TO 999
   CM = CONC(I,2) * CONVRT
   DO 72 K=1,NBAYS
   IF (XA .LT. PP(K+1)) GO TO 75
72 CONTINUE
   K = NBAYS
75 BAYL = PP(K+1) - PP(K)
   HC = (CONC(I,3) - PP(K))/BAYL
   F1 = CM
   F2 = CM*HC
   F3 = CM*HC**2 + CONC(I,4)*CONVRT/BAYL**2

```

```

C
80 L = K+1
   Z(K,K) = Z(K,K) + F1 - 2.*F2 + F3
   Z(K,L) = Z(K,L) + F2 - F3
   Z(L,L) = Z(L,L) + F3

```

```

C
   IF (MIC .EQ. 3 .OR. X2 .LE. PP(K+1)) GO TO 90
   K = K+1
   XP = X0
   VP = VQ
   GO TO 36
90 CONTINUE
95 CONTINUE

```

```

C
C SYMMETRIZE.
   DO 110 K=1,NBAYS
110 Z(K+1,K) = Z(K,K+1)

```

```

C
C COMPUTE AND PRINT TOTAL MASS PROPERTIES.
   TM = 0.
   TP = 0.
   TI = 0.
   DO 201 I=1,NPP
   DO 201 J=1,NPP
   TM = TM + Z(I,J)
   TP = TP + Z(I,J)*PP(J)
201 TT = TI + PP(I)*Z(I,J)*PP(J)
   CG = TP/TM
   TI = TI - TM*CG**2
   IF (MINI .NE. 4*MINI) GO TO 800
   IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
   IF ((NLINE+16) .GT. MAXLIN) GO TO 800

```

```
WRITE (NOT,2050)  
NLINE = NLINE + 2  
GO TO 810  
800 CALL PAGEHD  
810 WRITE (NOT,2001) CONVRT, TM, CG, TI  
NLINE = NLINE + 14  
RETURN
```

```
C  
999 CALL ZZBOMB (6HM, 991 ,NERROR)  
END
```

```

SUBROUTINE MASS2 (PP,DMASS,DRIN,CONC,CONVRT,Z,NPP,NDM,NDI,NC,NZ,
*           KDM,KDI,KC,KZ)
DIMENSION PP(1), DMASS(KDM,1), DRIN(KDI,1), CONC(KC,1), Z(KZ,1),
*           F(7), F(10)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C CALCULATE MASS MATRIX FOR A BEAM. ASSUMES CUBIC VELOCITY FUNCTION
C BETWEEN CONSECUTIVE PANEL POINTS.
C LATERAL TRANSLATION AND ROTATION AT EACH PANEL POINT ARE THE
C GENERALIZED COORDINATES. TRANSLATION COORDINATES ARE GROUPED FIRST
C SIGN CONVENTION IS ROTATION = -D(LATERAL DISP)/D(AXIAL COORDINATE).
C INPUT IS DISTRIBUTED MASS, DISTRIBUTED ROTARY INERTIA, CONCENTRATED
C ITEMS. THE DISTRIBUTED DATA MAY NOT EXCEED THE PANEL POINT LIMITS.
C THE ATTACH POINT FOR CONCENTRATED ITEMS MAY NOT EXCEED THE PANEL
C POINT LIMITS. OPTION TO OMIT DATA BY NDM,NDI, OR NC EQUAL ZERO.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C CODED BY RL WOHLER. DECEMBER 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

SUBROUTINE ARGUMENTS

```

C PP      = INPUT  VECTOR OF PANEL POINTS. SIZE(NPP).
C DMASS   = INPUT  MATRIX OF DISTRIBUTED MASS STRAIGHT LINE
C          SEGMENT DATA. SIZE(NDM,4).
C          COL 1 = X AT SEGMENT END 1.
C          COL 2 = X AT SEGMENT END 2.
C          COL 3 = MASS AT SEGMENT END 1.
C          COL 4 = MASS AT SEGMENT END 2.
C DRIN    = INPUT  MATRIX OF DISTRIBUTED ROTARY INERTIA STRAIGHT LINE
C          SEGMENT DATA. SIZE(NDI,4).
C          COLUMNS ARE SIMILAR TO DMASS.
C CONC    = INPUT  MATRIX OF CONCENTRATED ITEM DATA. SIZE(NC,4).
C          COL 1 = ATTACH STATION.
C          COL 2 = MASS OF ITEM.
C          COL 3 = CENTER OF GRAVITY OF ITEM.
C          COL 4 = MOMENT OF INERTIA ABOUT CG OF ITEM.
C CONVRT  = INPUT  CONVERSION SCALAR BY WHICH COL 3,4 OF DMASS,DRIN AND
C          COL 2,4 OF CONC WILL BE MULTIPLIED.
C Z       = OUTPUT MASS MATRIX. SIZE(NZ,NZ).
C NPP     = INPUT  NUMBER OF PANEL POINTS. SIZE OF VECTOR PP.
C NDM     = INPUT  NUMBER OF SEGMENTS (ROWS) IN DMASS. CAN BE ZERO.
C NDI     = INPUT  NUMBER OF SEGMENTS (ROWS) IN DRIN. CAN BE ZERO.
C NC      = INPUT  NUMBER OF ITEMS (ROWS) IN CONC. CAN BE ZERO.
C NZ      = OUTPUT SIZE OF MATRIX Z. (NZ=2*NPP).
C KDM     = INPUT  ROW DIMENSION OF DMASS IN CALLING PROGRAM.
C KDI     = INPUT  ROW DIMENSION OF DRIN  IN CALLING PROGRAM.
C KC      = INPUT  ROW DIMENSION OF CONC  IN CALLING PROGRAM.
C KZ      = INPUT  ROW DIMENSION OF Z    IN CALLING PROGRAM.

```

ERROR EXPLANATION

```

C 1 = LESS THAN 2 PANEL POINTS.
C 2 = PANEL POINTS NOT IN INCREASING ORDER.
C 3 = INCORRECT DISTRIBUTED DATA.
C 4 = CONCENTRATED MASS ATTACH STATION OUTSIDE PANEL
C    POINT BOUNDS.

```

```

2001 FORMAT ( 3(/),30X,3iHSUBROUTINE MASS2 USES CONVRT = E15.8, ///
*          37X,33HAND COMPUTES THE TOTAL PROPERTIES ///
*          43X,6HM   = E15.8,///43X,6HXCG = E15.8,///43X,6HICG = E15.8)
2050 FORMAT (/ 1X 123(1H-))

```

```

C
C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER.
                                NERROR = 1
    IF (NPP .LT. 2) GO TO 999
                                NERROR = 2
    DO 5 K=2,NPP
    IF (PP(K-1) .GE. PP(K)) GO TO 999
    5 CONTINUE
C
C INITIALIZE DATA.
    NZ = 2*NPP
    DO 10 I=1,NZ
    DO 10 J=1,NZ
    10 Z(I,J) = 0.0
    NBAYS = NPP-1
C
C DISTRIBUTED MASS (MIC=1), DISTRIBUTED ROTARY INERTIA (MIC=2),
C CONCENTRATED ITEM (MIC=3).
    DO 95 MIC=1,3
    IF (MIC .EQ. 1) NSEGS = NDM
    IF (MIC .EQ. 2) NSEGS = NDI
    IF (MIC .EQ. 3) NSEGS = NC
    IF (NSEGS .EQ. 0) GO TO 95
C
    DO 90 I=1,NSEGS
    GO TO (21,22,70),MIC
21 X1 = DMASS(I,1)
    X2 = DMASS(I,2)
    V1 = DMASS(I,3) * CONVRT
    V2 = DMASS(I,4) * CONVRT
    GO TO 30
22 X1 = DRIN(I,1)
    X2 = DRIN(I,2)
    V1 = DRIN(I,3) * CONVRT
    V2 = DRIN(I,4) * CONVRT
30
                                NERROR = 3
    IF (X1 .LT. PP(1) .OR. X2 .GT. PP(NPP) .OR. X1 .GE. X2) GO TO 999
    DO 32 K=1,NBAYS
    IF (X1 .LT. PP(K+1)) GO TO 34
32 CONTINUE
34 XP = X1
    VP = V1
36 IF (X2 .LE. PP(K+1)) GO TO 38
    XQ = PP(K+1)
    VQ = V1 + (XQ-X1)*(V2-V1)/(X2-X1)
    GO TO 39
38 XQ = X2
    VQ = V2
39 BAYL = PP(K+1) - PP(K)
    HP = (XP-PP(K)) / BAYL

```

```

HQ = (XQ-PP(K)) / BAYL
W = (VQ-VP)/(HQ-HP)
DO 44 J=1,7
JP1 = J+1
RJ = J
RJP1 = JP1
44 E(J) = W*(HQ**JP1 - HP**JP1)/RJP1 + (VP-W*HP)*(HQ**J - HP**J)/RJ
GO TO (50,60),MIC

```

C

```

50 DO 55 J=1,7
55 F(J) = BAYL * E(J)
F( 8) = BAYL * E(3)
F( 9) = BAYL * E(4)
F(10) = BAYL * E(5)
GO TO 80

```

C

```

60 F( 1) = 0.
F( 2) = 0.
F( 3) = 0.
F( 4) = 0.
F( 5) = 3.*E(3) / BAYL
F( 6) = 6.*E(4) / BAYL
F( 7) = 9.*E(5) / BAYL
F( 8) = E(1) / BAYL
F( 9) = 2.*E(2) / BAYL
F(10) = 4.*E(3) / BAYL
GO TO 80

```

```

70 XA = CONC(I,1)

```

NERROR = 4

```

IF (XA .LT. PP(1) .OR. XA .GT. PP(NPP)) GO TO 999
CM = CONC(I,2) * CONVRT
DO 72 K=1,NBAYS
IF (XA .LT. PP(K+1)) GO TO 75

```

```

72 CONTINUE

```

```

K = NBAYS

```

```

75 BAYL = PP(K+1) - PP(K)
CI = (CONC(I,4)*CONVRT) / BAYL**2
HA = (XA - PP(K)) / BAYL
HC = (CONC(I,3) - PP(K))/BAYL
P1 = 2.*HC - HA
P2 = 3.*HC - 2.*HA
F( 1) = CM
F( 2) = CM*HC
F( 3) = HA * CM*P1
F( 4) = HA**2 * CM*P2
F( 5) = HA**2 * (CM*P2*HC + 3.*CI)
F( 6) = HA**3 * (CM*P1*P2 + 6.*CI)
F( 7) = HA**4 * (CM*P2**2 + 9.*CI)
F( 8) = CM*HC**2 + CI
F( 9) = HA * (CM*P1*HC + 2.*CI)
F(10) = HA**2 * (CM*P1**2 + 4.*CI)

```

```

80 L = K+1
M = K+NPP

```

```

N = K+NPP+1
P1 = -12.*F(6) + 4.*F(7) + 9.*F(10)
P2 = 2.*F(5) - 7.*F(6) + 2.*F(7) - 3.*F(9) + 6.*F(10)
P3 = -5.*F(6) + 2.*F(7) + 3.*F(10)
BAYL2 = BAYL**2
Z(K,K) = Z(K,K) + F(1)-6.*F(3)+4.*F(4)+P1
Z(K,L) = Z(K,L) + 3.*F(3)-2.*F(4)-P1
Z(K,M) = Z(K,M) + (-F(2)+2.*F(3)-F(4)-P2) * BAYL
Z(K,N) = Z(K,N) + (F(3)-F(4)-P3) * BAYL
Z(L,L) = Z(L,L) + P1
Z(L,M) = Z(L,M) + P2*BAYL
Z(L,N) = Z(L,N) + P3*BAYL
Z(M,M) = Z(M,M) + (2.*F(5)-4.*F(6)+F(7)+F(8)-4.*F(9)+4.*F(10) )
1      * BAYL2
Z(M,N) = Z(M,N) + (F(5)-3.*F(6)+F(7)-F(9)+2.*F(10) ) * BAYL2
Z(N,N) = Z(N,N) + (-2.*F(6)+F(7)+F(10) ) * BAYL2

```

```

C
  IF (MIC .EQ. 3 .OR. X2 .LE. PP(K+1)) GO TO 90
  K = K+1
  XP = XQ
  VP = VQ
  GO TO 36
90 CONTINUE
95 CONTINUE

```

```

C
C SYMMETRIZE.
  DO 110 I=1,NZ
  DO 110 J=I,NZ
110 Z(J,I) = Z(I,J)

```

```

C
C COMPUTE AND PRINT TOTAL MASS PROPERTIES.
  TM = 0.
  TP = 0.
  TI = 0.
  DO 201 I=1,NPP
  K = I + NPP
  DO 201 J=1,NPP
  L = J + NPP
  TM = TM + Z(I,J)
  TP = TP - Z(K,J) + Z(I,J)*PP(J)
201 TI = TI + PP(I)*Z(I,J)*PP(J) - 2.*Z(L,I)*PP(I) + Z(K,L)
  CG = TP/TM
  TI = TI - TM*CG**2
  IF (MINI .NE. 4HMINI) GO TO 800
  IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
  IF ((NLINE+16) .GT. MAXLIN) GO TO 800
  WRITE (NOT,2050)
  NLINE = NLINE + 2
  GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2001) CONVRT, TM, CG, TI
  NLINE = NLINE + 14
  RETURN

```

```

C
999 CALL ZZBOMB (6HMASS2 ,NERROR)

```

MASS2 -- 5/ 5

END

```

SUBROUTINE MASS2A (PP,DMASS,SMASS,FLEVEL,CONVRT,Z,NPP,NDM,NZ,
*           KDM,KZ)
DIMENSION PP(1), DMASS(KDM,1), Z(KZ,1), F(7)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C CALCULATE MASS MATRIX FOR FLUID IN A CONTAINER. INCLUDES COUPLING
C BETWEEN DISTRIBUTED MASS AND SLOSH MASS. ASSUMES CUBIC FUNCTION
C BETWEEN CONSECUTIVE PANEL POINTS TO DESCRIBE CONTAINER LATERAL
C VELOCITY AND ASSUMES UNIFORM SLOSH MOTION RELATIVE TO THE CONTAINER.
C LATERAL TRANSLATION AND ROTATION AT EACH PANEL POINT AND THE SLOSH
C AMPLITUDE ARE THE GENERALIZED COORDINATES. TRANSLATION COORDINATES
C ARE FIRST, ROTATION COORDINATES NEXT, THE SLOSH COORDINATE LAST.
C SIGN CONVENTION IS ROTATION = -D(LATERAL DISP)/D(AXIAL COORDINATE).
C INPUT IS THE FLUID DISTRIBUTED MASS, THE FLUID SLOSH MASS, AND THE
C FLUID LEVEL. THE DISTRIBUTED MASS MAY NOT EXCEED THE PANEL POINT
C LIMITS. THE FLUID LEVEL MAY NOT EXCEED THE DISTRIBUTED MASS LIMITS.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C CODED BY C BODLEY. MAY 1966.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

SUBROUTINE ARGUMENTS

```

C PP      = INPUT  VECTOR OF PANEL POINTS. SIZE(NPP).
C DMASS   = INPUT  MATRIX OF DISTRIBUTED MASS STRAIGHT LINE
C          SEGMENT DATA. SIZE(NDM,4).
C          COL 1 = X AT SEGMENT END 1.
C          COL 2 = X AT SEGMENT END 2.
C          COL 3 = MASS AT SEGMENT END 1.
C          COL 4 = MASS AT SEGMENT END 2.
C          DMASS(I,2) MUST EQUAL DMASS(I+1,1), ETC.
C SMASS   = INPUT  SLOSH MASS. Z(NZ,NZ) OF OUTPUT MASS MATRIX.
C FLEVEL  = INPUT  FLUID LEVEL. MUST BE WITHIN DMASS LIMITS.
C CONVRT  = INPUT  CONVERSION SCALAR BY WHICH COL 3,4 OF DMASS WILL BE
C          MULTIPLIED.
C Z       = OUTPUT MASS MATRIX. SIZE(NZ,NZ).
C NPP     = INPUT  NUMBER OF PANEL POINTS. SIZE OF VECTOR PP.
C NDM     = INPUT  NUMBER OF SEGMENTS (ROWS) IN DMASS.
C NZ      = OUTPUT SIZE OF MATRIX Z. (NZ=2*NPP+1).
C KDM     = INPUT  ROW DIMENSION OF DMASS IN CALLING PROGRAM.
C KZ      = INPUT  ROW DIMENSION OF Z      IN CALLING PROGRAM.

```

NERROR EXPLANATION

```

C 1 = LESS THAN 2 PANEL POINTS.
C 2 = PANEL POINTS NOT IN INCREASING ORDER.
C 3 = FLUID LEVEL OUTSIDE DISTRIBUTED MASS BOUNDS.
C 4 = DISTRIBUTED MASS HAS GAPS.
C 5 = DISTRIBUTED MASS EXCEEDS PANEL POINT BOUNDS.

```

```

2001 FORMAT ( 3(/),30X,32H SUBROUTINE MASS2A USES CONVRT = E15.8, //
*          49X,13HSLOSH MASS = E15.8, //48X,14HFLUID LEVEL = E15.8, ///
*          37X,33HAND COMPUTES THE TOTAL PROPERTIES ///
*          43X,6HM      = E15.8, //43X,6HXCG = E15.8, //43X,6HICG = E15.8)
2050 FORMAT (/ 1X 123(1H-))

```

```

C
C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER.

```



IF (NPP .LT. 2) GO TO 999

NERROR = 1

DO 5 K=2,NPP  
 IF (PP(K-1) .GE. PP(K)) GO TO 999  
 5 CONTINUE

NERROR = 2

C  
 C CHECK DISTRIBUTED MASS MATRIX.

IF (FLEVEL.LT.DMASS(1,1) .OR. FLEVEL.GE.DMASS(NDM,2)) GO TO 999  
 IF (NDM .EQ. 1) GO TO 9  
 NDM1 = NDM-1

NERROR = 3

DO 7 I=1,NDM1  
 IF (DMASS(I,2) .NE. DMASS(I+1,1)) GO TO 999  
 7 CONTINUE

NERROR = 4

C  
 C INITIALIZE DATA.

9 NZ = 2\*NPP+1  
 DO 10 I=1,NZ  
 DO 10 J=1,NZ  
 10 Z(I,J) = 0.0  
 NBAYS = NPP-1

C  
 DO 15 JM = 1,NDM  
 IF (FLEVEL .LT. DMASS(JM,2)) GO TO 16  
 15 CONTINUE

16 DMJ1 = DMASS(JM,1)  
 DMJ3 = DMASS(JM,3)  
 DMASS(JM,1) = FLEVEL  
 DMASS(JM,3) = DMJ3 + (FLEVEL - DMJ1) \* (DMASS(JM,4) - DMJ3) /  
 \* (DMASS(JM,2) - DMJ1)

C  
 DO 90 I = JM,NDM  
 X1 = DMASS(I,1)  
 X2 = DMASS(I,2)  
 V1 = DMASS(I,3) \* CONVRT  
 V2 = DMASS(I,4) \* CONVRT

NERROR = 5

IF (X1 .LT. PP(1) .OR. X2 .GT. PP(NPP) .OR. X1 .GE. X2) GO TO 999  
 DO 32 K=1,NBAYS  
 IF (X1 .LT. PP(K+1)) GO TO 34  
 32 CONTINUE

34 XP = X1  
 VP = V1  
 36 IF (X2 .LE. PP(K+1)) GO TO 38  
 XQ = PP(K+1)  
 VQ = V1 + (XQ-X1)\*(V2-V1)/(X2-X1)  
 GO TO 39

38 XQ = X2  
 VQ = V2  
 39 BAYL = PP(K+1) - PP(K)  
 HP = (XP-PP(K)) / BAYL  
 HQ = (XQ-PP(K)) / BAYL  
 W = (VQ-VP)/(HQ-HP)

```

DO 44 J=1,7
  JPI = J+1
  RJ = J
  RJP1 = JP1
44 F(J) = (W*(HQ**JP1 - HP**JP1)/RJP1 + (VP-W*HP)*(HQ**J - HP**J)/RJ)
      *
      * BAYL

```

C

```

L = K+1
M = K+NPP
N = K+NPP+1
P1 = 9.*F(5) - 12.*F(6) + 4.*F(7)
P2 = -3.*F(4) + 8.*F(5) - 7.*F(6) + 2.*F(7)
P3 = 3.*F(5) - 5.*F(6) + 2.*F(7)
BAYL2 = BAYL**2
Z(K,K) = Z(K,K) + F(1)-6.*F(3)+4.*F(4)+P1
Z(K,L) = Z(K,L) + 3.*F(3)-2.*F(4)-P1
Z(K,M) = Z(K,M) + (-F(2)+2.*F(3)-F(4)-P2) * BAYL
Z(K,N) = Z(K,N) + (F(3)-F(4)-P3) * BAYL
Z(L,L) = Z(L,L) + P1
Z(L,M) = Z(L,M) + P2*BAYL
Z(L,N) = Z(L,N) + P3*BAYL
Z(M,M) = Z(M,M) + (F(3)-4.*F(4)+6.*F(5)-4.*F(6) +F(7)) * BAYL2
Z(M,N) = Z(M,N) + (-F(4) +3.*F(5)-3.*F(6)+F(7)) * BAYL2
Z(N,N) = Z(N,N) + (F(5)-2.*F(6)+F(7)) * BAYL2
Z(K,NZ) = Z(K,NZ) + F(1)-3.*F(3)+2.*F(4)
Z(L,NZ) = Z(L,NZ) + 3.*F(3)-2.*F(4)
Z(M,NZ) = Z(M,NZ) + (-F(2)+2.*F(3)-F(4)) * BAYL
Z(N,NZ) = Z(N,NZ) + (F(3)-F(4)) * BAYL
Z(NZ,NZ) = Z(NZ,NZ) + F(1)

```

C

```

IF (X2 .LE. PP(K+1)) GO TO 90
K = K+1
XP = XQ
VP = VQ
GO TO 36
90 CONTINUE

```

C

```

DMASS(JM,1) = DMJ1
DMASS(JM,3) = DMJ3

```

C

```

R = SMASS/Z(NZ,NZ)
DO 95 I=1,NZ
95 Z(I,NZ) = Z(I,NZ)*R
Z(NZ,NZ) = SMASS

```

C

C SYMMETRIZE.

```

DO 110 I=1,NZ
DO 110 J=I,NZ
110 Z(J,I) = Z(I,J)

```

C

C COMPUTE AND PRINT TOTAL MASS PROPERTIES.

```

TM = 0.
TP = 0.
TI = 0.
DO 201 I=1,NPP

```

```

K = I + NPP
DO 201 J=1,NPP
L = J + NPP
TM = TM + Z(I,J)
TP = TP - Z(K,J) + Z(I,J)*PP(J)
201 TI = TI + PP(I)*Z(I,J)*PP(J) - 2.*Z(L,I)*PP(I) + Z(K,L)
CG = TP/TM
TI = TI - TM*CG**2
IF (MINI .NE. 4HMINI) GO TO 800
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
IF ((NLINE+20) .GT. MAXLIN) GO TO 800
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2001) CONVRT,SMASS,FLEVEL,TM,CG,TI
NLINE = NLINE + 18
RETURN

```

C

```

999 CALL Z2BOMB (6HMASS2A,NERROR)
END

```

```

SUBROUTINE MODE1 (A,S,W2,W,FREQ,N,FOD,KR,NUT1)
DIMENSION A(KR,1), S(KR,1), W2(1), W(1), FREQ(1)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C CALCULATE MODE SHAPES (PHI) AND NATURAL FREQUENCIES OF
C (A)**-1(S)(PHI) = (PHI)(-W2-) USING METHOD OF JACOBI.
C THE MASS (A) MATRIX MUST BE REAL, SYMMETRIC, POSITIVE DEFINITE.
C THE STIF (S) MATRIX MUST BE REAL, SYMMETRIC.
C THE FIRST ELEMENT OF EACH MODE SHAPE IS MADE POSITIVE.
C ORTHOGONALITY CHECKS --(PHI)T*(MASS)*(PHI) AND (PHI)T*(STIF)*(PHI)--
C ARE CALCULATED AND PRINTED.
C CALLS FORMA SUBROUTINES BTABA,BTABA2,DCOM1,EIGN1,INV4,PAGEHD,(ZZBOMB).
C THE MAXIMUM SIZE IS
C N = 500 (BASED ON BTABA, BTABA2).
C DEVELOPED BY RL WOHLN. APRIL 1969.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS

```

```

C A = INPUT MASS MATRIX. SIZE(N,N). *DESTROYED*
C S = INPUT STIFFNESS MATRIX. SIZE(N,N). *DESTROYED*
C W2 = OUTPUT VECTOR OF EIGENVALUES (OMEGA SQUARED). SIZE(N).
C W = OUTPUT VECTOR OF CIRCULAR FREQUENCY (OMEGA). SIZE(N).
C FREQ = OUTPUT VECTOR OF FREQUENCY (OMEGA/2PI). SIZE(N).
C N = INPUT SIZE OF MATRICES A,S AND VECTORS W2,W,FREQ. MAX=500.
C FOD = INPUT FINAL OFF-DIAGONAL VALUE FOR DYNAMIC MATRIX.
C IF FOD .LE. ZERO, THE VALUE OF FOD WILL BE CALCULATED
C AUTOMATICALLY IN SUBROUTINE EIGN1.
C KR = INPUT ROW DIMENSION OF A,S IN CALLING PROGRAM.
C NUT1 = INPUT NUMBER OF UTILITY TAPE. (EG 4).

```

```

C
2001 FORMAT ( 3(//) 54X,18H(SUBROUTINE MODE1)
* // 47X,34HTHE FOLLOWING ORTHOGONALITY CHECKS
* // 52X,23H(MODES)T*(MASS)*(MODES)
* // 52X,23H(MODES)T*(STIF)*(MODES)
* // 48X,32HARE A RESULT OF THIS SUBROUTINE.)
2002 FORMAT (// 10X,10(7X,1H(, I2, 1H)))
2011 FORMAT (///10X,39HTHE (MODES)T*(MASS)*(MODES) CHECK GIVES
* //10X,25HTHE DIAGONAL ELEMENTS ARE // (13X,10F11.8))
2012 FORMAT (// 10X,35HTHE MAXIMUM OFF-DIAGONAL ELEMENT IS
* E11.3, 2X, 4HAT ( I3, 1H, I3, 1H) )
2020 FORMAT (///10X,28HTHE OMEGA SQUARED VALUES ARE // (13X,10E11.3))
2021 FORMAT (///10X,39HTHE (MODES)T*(STIF)*(MODES) CHECK GIVES
* //10X,48HTHE ABSOLUTE PERCENT DIFFERENCE IN THE DIAGONAL
* 31HELEMENTS FROM OMEGA SQUARED ARE //(13X,10F11.8))
2022 FORMAT (// 10X,48HTHE LARGEST OFF-DIAGONAL ELEMENT IN EACH ROW ARE
* // (13X,10E11.3))
2050 FORMAT (/ 1X 123(1H-) )

```

```

C
IF (MINI .NE. 4HMINI) GO TO 810
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 810
IF ((NLINE+2+12) .GT. MAXLIN) GO TO 810
WRITE (NOT,2050)
NLINE = NLINE + 2

```

```

GO TO 811
810 CALL PAGEHD
811 WRITE (NOT,2001)
    NLINE = NLINE + 12
C
    CALL RWND    (NUT1)
    CALL OUT    (NUT1,A,KR*N)
    CALL OUT    (NUT1,S,KR*N)
C
    CALL DCOM1  (A,A,N,KR)
    CALL INV4   (A,A,N,KR)
    CALL BTABA2 (S,A,N,KR)
    CALL EIGN1  (S,W2,A,-N,FOD,KR)
C
C
C ALIGN THE CIRCULAR FREQUENCY SQUARED (W2) INTO INCREASING ORDER AND
C THE MODE SHAPES CORRESPONDINGLY.
    IF (N .EQ. 1) GO TO 40
    NM1 = N-1
    DO 35 J=1,NM1
    W2MIN = W2(J)
    IMIN = J
    JP1 = J+1
    DO 30 I=JP1,N
    IF (W2MIN .LE. W2(I)) GO TO 30
    W2MIN = W2(I)
    IMIN = I
30 CONTINUE
    IF (IMIN .EQ. J) GO TO 35
    W2(IMIN) = W2(J)
    W2(J) = W2MIN
    DO 34 K=1,N
    AKJ = A(K,J)
    A(K,J) = A(K,IMIN)
34 A(K,IMIN) = AKJ
35 CONTINUE
C
C MAKE THE FIRST ELEMENT OF EACH MODE SHAPE POSITIVE.
40 DO 45 J=1,N
    IF (A(1,J) .GE. 0.) GO TO 45
    DO 42 I=1,N
42 A(I,J) = -A(I,J)
45 CONTINUE
C
C CALCULATE (PHI)T*(MASS)*(PHI) ORTHOGONALITY CHECK.
    CALL RWND    (NUT1)
    CALL IN      (NUT1,S,KR*N)
    CALL BTABA   (S,A,N,N,KR,KR)
    XOFF = 0.
    IOFF = 1
    JOFF = 2
    DO 54 I=1,N
    DO 52 J=1,N
    IF (I .EQ. J) GO TO 52
    IF (ABS(XOFF) .GE. ABS(S(I,J))) GO TO 52
    A=MASS
    S=STIF
    A=U
    A=U**-1
    S=DYMMAT
    W2=W2
    A=PHI
    S=MASS
    S=PMP

```

```

XOFF = S(I,J)
IOFF = I
JOFF = J
52 CONTINUE
54 CONTINUE
NPL = N/10
IF ((NPL*10) .NE. N) NPL = NPL+1
IF (MINI .NE. 4HMINI) GO TO 820
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 820
IF ((NLINE+2+14+NPL) .GT. MAXLIN) GO TO 820
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 821
820 CALL PAGEHD
821 WRITE (NOT,2002) (JC,JC=1,10)
WRITE (NOT,2011) (S(I,I),I=1,N)
WRITE (NOT,2012) XOFF,IOFF,JOFF
NLINE = NLINE + 14 + NPL

```

C  
C CALCULATE (PHI)T\*(STIF)\*(PHI) ORTHOGONALITY CHECK.

```

CALL IN (NUT1,S,KR*N)
CALL BTABA (S,A,N,N,KR,KR)
DO 64 I=1,N
W(I) = 0.
DO 62 J=1,N
IF (I .EQ. J) GO TO 62
IF (ABS(S(I,J)) .GT. ABS(W(I))) W(I)=S(I,J)
62 CONTINUE
64 CONTINUE
IF (MINI .NE. 4HMINI) GO TO 830
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 830
IF ((NLINE+2+20+3*NPL) .GT. MAXLIN) GO TO 830
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 831
830 CALL PAGEHD
831 WRITE (NOT,2002) (JC,JC=1,10)
WRITE (NOT,2020) (W2(I), I=1,N)
NLINE = NLINE + 8 + NPL
DO 68 I=1,N
IF (W2(I) .LE. 0.) GO TO 68
S(I,I) = ABS(S(I,I)-W2(I))*100./W2(I)
68 CONTINUE
WRITE (NOT,2021) (S(I,I), I=1,N)
WRITE (NOT,2022) (W(I), I=1,N)
NLINE = NLINE + 12 + 2*NPL

```

S=STIF  
S=PSP

C  
DO 72 I=1,N  
W(I) = SQRT (ABS(W2(I)))  
72 FREQ(I)= .15915494 \* W(I)

C  
RETURN  
END

```

SUBROUTINE MODE1A (A,S,W2,W,FREQ,N,FOD,KR,NUT1)
DIMENSION A(KR,1), S(KR,1), W2(1), W(1), FREQ(1)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DATA EPS/1.E-30/
DATA NIT,NOT/5,6/

```

```

C
C CALCULATE MODE SHAPES (PHI) AND NATURAL FREQUENCIES OF
C  $(C(A)+(S))^{*-1}(A)(PHI) = (PHI)(-1/(C+W2)-)$  USING METHOD OF JACOBI.
C THE MASS (A) MATRIX MUST BE REAL, SYMMETRIC, POSITIVE DEFINITE.
C THE STIF (S) MATRIX MUST BE REAL, SYMMETRIC.
C THE FIRST ELEMENT OF EACH MODE SHAPE IS MADE POSITIVE.
C ORTHOGONALITY CHECKS  $-(PHI)T*(MASS)*(PHI)$  AND  $(PHI)T*(STIF)*(PHI)-$ 
C ARE CALCULATED AND PRINTED.
C CALLS FORMA SUBROUTINES BTABA,BTABA2,DCOM1,EIGN1,INV4,PAGEHD,(ZZBOMB).
C THE MAXIMUM SIZE IS
C N = 500 (BASED ON BTABA,BTABA2).
C DEVELOPED BY RL WOHLN. APRIL 1969.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C A = INPUT MASS MATRIX. SIZE(N,N). *DESTROYED*
C S = OUTPUT MODE SHAPES. SIZE(N,N).
C S = INPUT STIFFNESS MATRIX. SIZE(N,N). *DESTROYED*
C W2 = OUTPUT VECTOR OF CIRCULAR FREQUENCY SQUARED. SIZE(N).
C W = OUTPUT VECTOR OF CIRCULAR FREQUENCY (OMEGA). SIZE(N).
C FREQ = OUTPUT VECTOR OF FREQUENCY (OMEGA/2PI). SIZE(N).
C N = INPUT SIZE OF MATRICES A,S AND VECTORS W2,W,FREQ. MAX=500.
C FOD = INPUT FINAL OFF-DIAGONAL VALUE FOR DYNAMIC MATRIX.
C IF FOD .LE. ZERO, THE VALUE OF FOD WILL BE CALCULATED
C AUTOMATICALLY IN SUBROUTINE EIGN1.
C KR = INPUT ROW DIMENSION OF A,S IN CALLING PROGRAM.
C NUT1 = INPUT NUMBER OF UTILITY TAPE. (EG 4).

```

```

2001 FORMAT ( 3(//) 54X,19H(SUBROUTINE MODE1A)
*          /// 38X,37HTHE CALCULATED COMBINATION VALUE = E15.8,
*          /// 47X,34HTHE FOLLOWING ORTHOGONALITY CHECKS
*          // 52X,23H(MODES)T*(MASS)*(MODES)
*          / 52X,23H(MODES)T*(STIF)*(MODES)
*          // 48X,32HARE A RESULT OF THIS SUBROUTINE.)
2002 FORMAT (// 10X,10(7X,1H(,I2,1H)))
2011 FORMAT (///10X,39HTHE (MODES)T*(MASS)*(MODES) CHECK GIVES
*          ///10X,25HTHE DIAGONAL ELEMENTS ARE // (13X,10F11.8))
2012 FORMAT (// 10X,35HTHE MAXIMUM OFF-DIAGONAL ELEMENT IS
*          E11.3, 2X, 4HAT ( I3, 1H, I3, 1H ) )
2020 FORMAT (///10X,28HTHE OMEGA SQUARED VALUES ARE // (13X,10E11.3))
2021 FORMAT (///10X,39HTHE (MODES)T*(STIF)*(MODES) CHECK GIVES
*          ///10X,48HTHE ABSOLUTE PERCENT DIFFERENCE IN THE DIAGONAL
*          31HELEMENTS FROM OMEGA SQUARED ARE //(13X,10F11.8))
2022 FORMAT (// 10X,48HTHE LARGEST OFF-DIAGONAL ELEMENT IN EACH ROW ARE
*          // (13X,10E11.3))
2050 FORMAT (/ 1X 123(1H- ) )

```

```

ANORM = 0.0
SNORM = 0.0
DO 5 I=1,N

```

```

DO 5 J=1,N
ANORM = ANORM + ABS(A(I,J))
5 SNORM = SNORM + ABS(S(I,J))
C = SNORM/ANORM

```

```

C
IF (MINI .NE. 4HMINI) GO TO 810
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 810
IF ((NLINE+2+15) .GT. MAXLIN) GO TO 810
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 811
810 CALL PAGEHD
811 WRITE (NOT,2001) C
NLINE = NLINE + 15

```

```

C
CALL RWND (NUT1)
CALL OUT (NUT1,A,KR*N)
CALL OUT (NUT1,S,KR*N)
A=MASS
S=STIF

```

```

C
DO 12 I=1,N
DO 12 J=1,N
12 A(I,J) = S(I,J) + C*A(I,J)
A=SCM

```

```

C
CALL DCOM1 (A,A,N,KR)
CALL INV4 (A,A,N,KR)
CALL RWND (NUT1)
CALL IN (NUT1,S,KR*N)
CALL BTABA2 (S,A,N,KR)
CALL EIGN1 (S,W2,A,-N,FOD,KR)
DO 28 J=1,N
C1 = SQRT(W2(J))
DO 28 I=1,N
28 A(I,J) = A(I,J)/C1
A=U
A=U**-1
S=MASS
S=DYNMAT
W2=VAL
A=PHI

```

```

C
C CALCULATE W2.
DO 29 I=1,N
IF (W2(I) .LT. EPS) W2(I)=EPS
29 W2(I) = 1./W2(I) - C
W2=W2

```

```

C
C ALIGN THE CIRCULAR FREQUENCY SQUARED (W2) INTO INCREASING ORDER AND
C THE MODE SHAPES CORRESPONDINGLY.
IF (N .EQ. 1) GO TO 40
NM1 = N-1
DO 35 J=1,NM1
W2MIN = W2(J)
IMIN = J
JP1 = J+1
DO 30 I=JP1,N
IF (W2MIN .LE. W2(I)) GO TO 30
W2MIN = W2(I)
IMIN = I
30 CONTINUE
IF (IMIN .EQ. J) GO TO 35
W2(IMIN) = W2(J)
W2(J) = W2MIN

```



```

DO 34 K=1,N
  AKJ = A(K,J)
  A(K,J) = A(K,IMIN)
34 A(K,IMIN) = AKJ
35 CONTINUE

```

C  
C MAKE THE FIRST ELEMENT OF EACH MODE SHAPE POSITIVE.

```

40 DO 45 J=1,N
  IF (A(I,J) .GE. 0.) GO TO 45
  DO 42 I=1,N
42 A(I,J) = -A(I,J)
45 CONTINUE

```

C  
C CALCULATE (PHI)T\*(MASS)\*(PHI) ORTHOGONALITY CHECK.

```

CALL RWND (NUT1)
CALL IN (NUT1,S,KR*N)
CALL BTABA (S,A,N,N,KR,KR)
XOFF = 0.
IOFF = 1
JOFF = 2
DO 54 I=1,N
DO 52 J=I,N
IF (I .EQ. J) GO TO 52
IF (ABS(XOFF) .GE. ABS(S(I,J))) GO TO 52
XOFF = S(I,J)
IOFF = I
JOFF = J
52 CONTINUE
54 CONTINUE
NPL = N/10
IF ((NPL*10) .NE. N) NPL = NPL+1
IF (MINI .NE. 4HMINI) GO TO 820
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 820
IF ((NLINE+2+14*NPL) .GT. MAXLIN) GO TO 820
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 821
820 CALL PAGEHD
821 WRITE (NOT,2002) (JC,JC=1,10)
WRITE (NOT,2011) (S(I,I),I=1,N)
WRITE (NOT,2012) XOFF,IOFF,JOFF
NLINE = NLINE + 14 + NPL

```

S=MASS  
S=PMP

C  
C CALCULATE (PHI)T\*(STIF)\*(PHI) ORTHOGONALITY CHECK.

```

CALL IN (NUT1,S,KR*N)
CALL BTABA (S,A,N,N,KR,KR)
DO 64 I=1,N
W(I) = 0.
DO 62 J=1,N
IF (I .EQ. J) GO TO 62
IF (ABS(S(I,J)) .GT. ABS(W(I))) W(I)=S(I,J)
62 CONTINUE
64 CONTINUE
IF (MINI .NE. 4HMINI) GO TO 830
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 830

```

S=STIF  
S=PSP

```

      IF ((NLINE+2+20+3*NPL) .GT. MAXLIN) GO TO 830
      WRITE (NOT,2050)
      NLINE = NLINE + 2
      GO TO 831
830 CALL PAGEHD
831 WRITE (NOT,2002) (JC,JC=1,10)
      WRITE (NOT,2020) (W2(I), I=1,N)
      NLINE = NLINE + 8 + NPL
      DO 68 I=1,N
      IF (W2(I) .LE. 0.) GO TO 68
      S(I,I) = ABS(S(I,I)-W2(I))*100./W2(I)
68 CONTINUE
      WRITE (NOT,2021) (S(I,I), I=1,N)
      WRITE (NOT,2022) (W(I), I=1,N)
      NLINE = NLINE + 12 +2*NPL
C
      DO 72 I=1,N
      W(I) = SQRT (ABS(W2(I)))
72 FREQ(I)= .15915494 * W(I)
C
      RETURN
      END

```

```

SUBROUTINE MODE1B (A,E,W2,W,FREQ,N,FOD,KR,NUT1)
DIMENSION A(KR,1), E(KR,1), W2(1), W(1), FREQ(1)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA EPS/1.E-30/, ZERO/0.0/
DATA N11,NOT/5,6/

```

```

C
C CALCULATE MODE SHAPES (PHI) AND NATURAL FREQUENCIES OF
C (E)(A)(PHI)=(PHI)(-1/W2-) USING METHOD OF JACOBI.
C THE MASS (A) MATRIX MUST BE REAL, SYMMETRIC, POSITIVE DEFINITE.
C THE FLEX (E) MATRIX MUST BE REAL, SYMMETRIC.
C THE FIRST ELEMENT OF EACH MODE SHAPE IS MADE POSITIVE.
C RIGID BODY MODES WILL BE IN THE LAST POSITIONS.
C ORTHOGONALITY CHECKS -- (PHI)T*(MASS)*(PHI) AND
C (PHI)T*(MASS)*(FLEX)*(MASS)*(PHI) -- ARE CALCULATED AND PRINTED.
C CALLS FORMER SUBROUTINES BTABA,DCOM1,EIGN1,INV4,MULTA,PAGEHD,(ZZBOMB).
C THE MAXIMUM SIZE IS
C N = 500 (BASED ON BTABA,MULTA).
C DEVELOPED BY RL WOHLN. APRIL 1969.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C A = INPUT MASS MATRIX. SIZE(N,N). *DESTROYED*
C E = INPUT FLEXIBILITY MATRIX. SIZE(N,N). *DESTROYED*
C W2 = OUTPUT VECTOR OF CIRCULAR FREQUENCY SQUARED. (INVERTED
: EIGENVALUES). SIZE(N).
C W = OUTPUT VECTOR OF CIRCULAR FREQUENCY (OMEGA). SIZE(N).
C FREQ = OUTPUT VECTOR OF FREQUENCY (OMEGA/2PI). SIZE(N).
C N = INPUT SIZE OF MATRICES A,E AND VECTORS W2,W,FREQ. MAX=500.
C FOD = INPUT FINAL OFF-DIAGONAL VALUE FOR DYNAMIC MATRIX.
C IF FOD .LE. ZERO, THE VALUE OF FOD WILL BE CALCULATED
C AUTOMATICALLY BY SUBROUTINE EIGN1.
C KR = INPUT ROW DIMENSION OF A,E IN CALLING PROGRAM.
C NUT1 = INPUT NUMBER OF UTILITY TAPE. (EG 4).

```

```

C
2001 FORMAT ( 31// 54X,19H(SUBROUTINE MODE1B)
* // 47X,14H(THE FOLLOWING ORTHOGONALITY CHECKS
* // 52X,23H(MODES)T*(MASS)*(MODES)
* // 45X,37H(MODES)T*(MASS)*(FLEX)*(MASS)*(MODES)
* // 48X,32H(ARE A RESULT OF THIS SUBROUTINE.)
2002 FORMAT (// 10X,10(7X,1H(,I2,1H)))
2011 FORMAT ( // 10X,39H(THE (MODES)T*(MASS)*(MODES) CHECK GIVES
* // 10X,25H(THE DIAGONAL ELEMENTS ARE // (13X,10F11.8))
2012 FORMAT (// 10X,35H(THE MAXIMUM OFF-DIAGONAL ELEMENT IS
* F11.3, 2X, 4HAT ( 13, 1H, 13, 1H) )
2020 FORMAT (/// 10X,32H(THE 1/(OMEGA SQUARED) VALUES ARE//(13X,10E11.3))
2021 FORMAT (/// 10X,48H(THE (MODES)T*(MASS)*(FLEX)*(MASS)*(MODES) CHECK
* 54GIVES /// 10X,39H(THE ABSOLUTE PERCENT DIFFERENCE IN THE
* 44HDIAGONAL ELEMENTS FROM 1/(OMEGA SQUARED) ARE
* //(13X,10F11.8))
2022 FORMAT (// 10X,48H(THE LARGEST OFF-DIAGONAL ELEMENT IN EACH ROW ARE
* // (13X,10E11.3))
2050 FORMAT (/ 1X 123(1H- )

```

C

```

IF (MINI .NE. 4HMINI) GO TO 810
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 810
IF ((NLINE+2+12 ) .GT. MAXLIN) GO TO 810
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 811
810 CALL PAGEHD
811 WRITE (NOT,2001)
NLINE = NLINE + 12
C
CALL PWND (NUT1)
CALL OUT (NUT1,A,KR*N)
CALL OUT (NUT1,E,KR*N)
C
CALL DCOM1 (A,A,N,KR)
C CALCULATE DYNAMIC MATRIX = (U)*(FLEX)*(U**T). STATEMENTS FROM
C SUBROUTINE BABA MODIFIED BECAUSE LOWER (U) IS ZERO AND ONLY
C UPPER HALF OF DYNAMIC MATRIX IS USED IN SUBROUTINE EIGN1.
DO 24 I=1,N
DO 22 J=1,N
S = ZERO
DO 21 K=J,N
SS = E(I,K)*A(J,K)
21 S = S + SS
22 W(J) = S
DO 24 J=1,N
24 E(I,J) = W(J)
DO 28 J=1,N
DO 26 I=1,J
S = ZERO
DO 25 K=I,N
SS = A(I,K)*E(K,J)
25 S = S + SS
26 W(I) = S
DO 28 I=1,J
28 E(I,J) = W(I)
CALL INV4 (A,A,N,KR)
CALL EIGN1 (E,W2,A,-N,FOD,KR)
C
C CALCULATE W2.
DO 29 I=1,N
IF (W2(I) .LT. EPS) W2(I)=EPS
29 W2(I) = 1./W2(I)
C
C ALIGN THE CIRCULAR FREQUENCY SQUARED (W2) INTO INCREASING ORDER AND
C THE MODE SHAPES CORRESPONDINGLY.
IF (N .EQ. 1) GO TO 40
NM1 = N-1
DO 35 J=1,NM1
W2MIN = W2(J)
IMIN = J
JP1 = J+1
DO 30 I=JP1,N
IF (W2MIN .LE. W2(I)) GO TO 30
W2MIN = W2(I)

```

A=MASS  
E=FLEX

A=U

E=DYNMAT;  
A=U\*\*-1  
W2=VAL  
A=PHI

W2=W2

```

    IMIN = I
30 CONTINUE
    IF (IMIN .EQ. J) GO TO 35
    W2(IMIN) = W2(J)
    W2(J) = W2MIN
    DO 34 K=1,N
    AKJ = A(K,J)
    A(K,J) = A(K,IMIN)
34 A(K,IMIN) = AKJ
35 CONTINUE

```

C  
C MAKE THE FIRST ELEMENT OF EACH MODE SHAPE POSITIVE.

```

40 DO 45 J=1,N
    IF (A(I,J) .GE. 0.) GO TO 45
    DO 42 I=1,N
42 A(I,J) = -A(I,J)
45 CONTINUE

```

C  
C CALCULATE (PHI)T\*(MASS)\*(PHI) ORTHOGONALITY CHECK.

```

    CALL OUT (NUT1,A,KR*N)
    CALL RWND (NUT1)
    CALL IN (NUT1,E,KR*N)
    CALL RTABA (E,A,N,N,KR,KR)
    XOFF = 0.
    IOFF = 1
    JOFF = 2
    DO 54 I=1,N
    DO 52 J=I,N
    IF (I .EQ. J) GO TO 52
    IF (ABS(XOFF) .GE. ABS(E(I,J))) GO TO 52
    XOFF = E(I,J)
    IOFF = I
    JOFF = J

```

E=MASS  
E=PMP

```

52 CONTINUE
54 CONTINUE
    NPL = N/10
    IF ((NPL*10) .NE. N) NPL = NPL+1
    IF (MINI .NE. 4HMINI) GO TO 820
    IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 820
    IF ((NLINE+2+14+NPL) .GT. MAXLIN) GO TO 820
    WRITE (NOT,2050)
    NLINE = NLINE + 2
    GO TO 821

```

```

820 CALL PAGEHD
821 WRITE (NOT,2002) (JC,JC=1,10)
    WRITE (NOT,2011) (E(I,I),I=1,N)
    WRITE (NOT,2012) XOFF,IOFF,JOFF
    NLINE = NLINE + 14 + NPL

```

C  
C CALCULATE (PHI)T\*(MASS)\*(FLEX)\*(MASS)\*(PHI) ORTHOGONALITY CHECK.

```

    CALL RWND (NUT1)
    CALL IN (NUT1,E,KR*N)
    CALL MULTA (E,A,N,N,N,KR,KR)
    CALL IN (NUT1,A,KR*N)
    CALL RTABA (A,E,N,N,KR,KR)

```

E=MASS  
E=MP  
A=FLEX  
A=PTMEMP

-----

```

DO 64 I=1,N
W(I) = 0.
DO 62 J=1,N
IF (I .EQ. J) GO TO 62
IF (ABS(A(I,J)) .GT. ABS(W(I))) W(I)=A(I,J)
62 CONTINUE
64 CONTINUE
IF (MINI .NE. 4HMINI) GO TO 830
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 830
IF ((NLINE+2+20+3*NPL) .GT. MAXLIN) GO TO 830
WRITE (NOT,2050)
NLINE = NLINE + 2
GO TO 831
830 CALL PAGEHD
831 WRITE (NOT,2002) (JC,JC=1,10)
DO 66 I=1,N
66 FREQ(I) = 1./W2(I)
WRITE (NOT,2020) (FREQ(I),I=1,N)
NLINE = NLINE + 8 + NPL
DO 68 I=1,N
68 A(I,I) = ABS(A(I,I)-FREQ(I))*100./FREQ(I)
WRITE (NOT,2021) (A(I,I), I=1,N)
WRITE (NOT,2022) (W(I), I=1,N)
NLINE = NLINE + 12 +2*NPL
C
CALL IN      (NUT1,A,KR*N)
A=PHI

DO 72 I=1,N
W(I) = SQRT (ABS(W2(I)))
72 FREQ(I)= .15915494 * W(I)
C
RETURN
END

```

```

SUBROUTINE MODEIX (A,S,W2,N,CTW2,KR)
DIMENSION A(KR,1), S(KR,1), W2(1)
DOUBLE PRECISION DM,DS

```

```

C
C CALCULATE MODE SHAPES (PHI) AND NATURAL FREQUENCIES OF
C (A)**-1(S)(PHI) = (PHI)(-W2-) USING METHOD OF JACOBI.
C MODIFICATION OF SUBROUTINE MODE1 TO ALLOW NON-POSITIVE DEFINITE MASS
C MATRIX, REMOVE ORTHOGONALITY CHECKS, AND USE W2 CONVERGENCE TOLERANCE.
C THE MASS (A) MATRIX SHOULD BE REAL, SYMMETRIC.
C THE STIF (S) MATRIX SHOULD BE REAL, SYMMETRIC.
C UPPER HALF OF MATRIX (A) IS USED TO CALCULATE MODE SHAPES AND
C FREQUENCIES. FULL MATRIX (S) IS USED.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES BTABA2,EIGN1A,INV4,(ZZBOMB).
C THE MAXIMUM SIZE IS
C   N = 500 (BASED ON BTABA2).
C DEVELOPED BY RL WOHLN. JANUARY 1972.
C LAST REVISION BY RL WOHLN. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS

```

```

C A   = INPUT MASS MATRIX. SIZE(N,N). *DESTROYED*
C     = OUTPUT MODE SHAPES. SIZE(N,N).
C S   = INPUT STIFFNESS MATRIX. SIZE(N,N). *DESTROYED*
C W2  = OUTPUT VECTOR OF EIGENVALUES (OMEGA SQUARED). SIZE(N).
C N   = INPUT SIZE OF MATRICES A,S AND VECTOR W2.
C CTW2 = INPUT CONVERGENCE TOLERANCE ON W2. IF CTW2 .LE. 0.,
C         10**-6 WILL BE USED. CONVERGENCE ASSUMED
C         IF W2 .LT. CTW2 OR IF THE W2 RATIO OF
C         (CURRENT-PRECEDING)/CURRENT .LT. CTW2.
C KR  = INPUT ROW DIMENSION OF A,S IN CALLING PROGRAM.

```

```

C DECOMPOSE MASS MATRIX (A) = (U)**T * (U).
C MODIFICATION OF SUBROUTINE DCOM1 TO USE SQRT(ABS(A(I,I))) AND A = Z.
  A(1,1) = SQRT(ABS(A(1,1)))
  IF (N .EQ. 1) GO TO 28
  DO 5 J=2,N
  5 A(1,J) = A(1,J)/A(1,1)
  DO 18 I=2,N
  IM1 = I-1
  IP1 = I+1
  DS = A(I,I)
  DO 10 K=1,IM1
  DM = A(K,I)**2
 10 DS = DS - DM
  A(I,I) = DS
  A(I,I) = SQRT(ABS(A(I,I)))
  IF (I .EQ. N) GO TO 20
  DO 18 J=IP1,N
  DS = A(I,J)
  DO 15 K=1,IM1
  DM = A(K,I)*A(K,J)
 15 DS = DS - DM
  A(I,J) = DS
 18 A(I,J) = A(I,J)/A(I,I)
 20 DO 25 I=2,N

```

```

      IM1 = I-1
      DO 25 J=1,IM1
25  A(I,J) = 0.0
C
28  CALL INV4  (A,A,N,KR)
      CALL BTABA2 (S,A,N,KR)
      CALL EIGN1A (S,W2,A,-N,CTW2,KR)
C
C  ALIGN THE CIRCULAR FREQUENCY SQUARED (W2) INTO INCREASING ORDER AND
C  THE MODE SHAPES CORRESPONDINGLY.
      IF (N .EQ. 1) RETURN
      NM1 = N-1
      DO 35 J=1,NM1
      W2MIN = W2(J)
      IMIN = J
      JP1 = J+1
      DO 30 I=JP1,N
      IF (W2MIN .LE. W2(I)) GO TO 30
      W2MIN = W2(I)
      IMIN = I
30  CONTINUE
      IF (IMIN .EQ. J) GO TO 35
      W2(IMIN) = W2(J)
      W2(J) = W2MIN
      DO 34 K=1,N
      AKJ = A(K,J)
      A(K,J) = A(K,IMIN)
34  A(K,IMIN) = AKJ
35  CONTINUE
C
      RETURN
      END

```

A=U  
A=U\*\*--1  
S=DYNNAT  
W2=W2  
A=PHI

\*



MULT

```
SUBROUTINE MULT (A,B,Z,NRA,NRB,NCB,KRA,KRB)
DIMENSION A(KRA,1), B(KRB,1), Z(KRA,1)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.D/
```

```
C
C MATRIX MULTIPLICATION. A * B = Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C DEVELOPED BY R L WOHLER. FEBRUARY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.
```

```
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX. SIZE(NRA,NRB).
C B = INPUT MATRIX. SIZE(NRB,NCB).
C Z = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z.
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A.
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z.
C KRA = INPUT ROW DIMENSION OF A,Z IN CALLING PROGRAM.
C KRB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.
```

```
C
DO 20 I=1,NRA
DO 20 J=1,NCB
S = ZERO
DO 10 K=1,NRB
SS = A(I,K)*B(K,J)
10 S = S + SS
20 Z(I,J) = S
RETURN
END
```

MULTA

```

SUBROUTINE MULTA (AZ,B,NRA,NRB,NCB,KAZ,KB)
DIMENSION AZ(KAZ,1), B(KB,1)
COMMON / LWRKV1 / W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.0/

```

```

C
C MATRIX MULTIPLICATION. A * B = Z.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN A.
C AZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF A OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NRB = 500
C DEVELOPED BY C S BODLEY. JANUARY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS
C AZ = INPUT MATRIX. SIZE(NRA,NRB).
C   = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C B = INPUT MATRIX. SIZE(NRB,NCB)
C NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z.
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=500.
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z.
C KAZ = INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM.
C KB = INPUT ROW DIMENSION OF B IN CALLING PROGRAM.

```

```

C NERROR EXPLANATION
C 1 = MORE THAN 500 ROWS IN MATRIX B.

```

NERROR=1

```

C IF (NRB .GT. 500) GO TO 999

```

```

C DO 40 I=1,NRA
DO 20 K=1,NRB
20 W(K) = AZ(I,K)
DO 40 J=1,NCB
S = ZERO
DO 30 K=1,NRB
SS = W(K)*B(K,J)
30 S = S + SS
40 AZ(I,J) = S
RETURN

```

```

C 999 CALL ZZBOMB (6HMULTA ,NERROR)
END

```

## MULTB

-----

```
SUBROUTINE MULTB (A,BZ,NRA,NRB,NCB,KA,KBZ)
DIMENSION A(KA,1),BZ(KBZ,1)
COMMON /LWRKV1/ W(500)
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.0/
```

```
C
C MATRIX MULTIPLICATION. A * B = Z.
C USES TWO WORK SPACES. RESULT (Z) IS PLACED IN B.
C BZ MUST BE DIMENSIONED LARGE ENOUGH IN MAIN PROGRAM TO CONTAIN THE
C LARGER OF B OR Z.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS
C   NRB = 500
C DEVELOPED BY CARL RODLEY. JANUARY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C A = INPUT MATRIX. SIZE(NRA,NRB).
C BZ = INPUT MATRIX. SIZE(NRB,NCB).
C   = OUTPUT RESULT MATRIX. SIZE(NRA,NCB).
C NRA = INPUT NUMBER OF ROWS OF MATRICES A,Z.
C NRB = INPUT NUMBER OF ROWS OF MATRIX B, COLS OF MATRIX A. MAX=500.
C NCB = INPUT NUMBER OF COLS OF MATRICES B,Z.
C KA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KBZ = INPUT ROW DIMENSION OF BZ IN CALLING PROGRAM.
C
C NERROR EXPLANATION
C 1 = SIZE LIMITATION EXCEEDED.
C
C NERROR=1
C IF (NRB.GT.500 .OR. NRA.GT.KBZ .OR. NRB.GT.KBZ) GO TO 999
C
C DO 40 J=1,NCB
C DO 20 K=1,NRB
20 W(K) = BZ(K,J)
C DO 40 I=1,NRA
C S = ZERO
C DO 30 K=1,NRB
C SS = A(I,K)*W(K)
30 S = S + SS
40 BZ(I,J) = S
RETURN
C
999 CALL ZZBOMB (6HMULTB ,NERROR)
END
```

```

FUNCTION NAME (NAMEIN,NUMIN)
DIMENSION FMT1(5),FMT2(3)
DATA FMT1/2H(A,1H ,2H,I,1H ,1H)/
DATA FMT2/2H(I,1H ,1H)/

```

```

C
C FUNCTION TO MERGE NAMEIN AND NUMIN INTO ONE VARIABLE (NAME) WHICH MAY
C BE USED AS AN A6 OUTPUT NAME IN ROUTINES SUCH AS FORMA SUBROUTINES
C WRITE, WTape, CKSTF1, PLOT1, ETC. (SEE EXAMPLES BELOW.)
C NOTE...IF THE SUM OF THE NON-BLANK CHARACTERS IN NAMEIN-MERGED-WITH-
C NUMIN EXCEEDS 6, THE RIGHT MOST CHARACTERS OF NAMEIN WILL BE
C DROPPED TO MAKE ROOM FOR NUMIN. (THE NUMBER ZERO IS NOT
C CONSIDERED A BLANK.)
C NOTE...THE INTRINSIC FUNCTION FLD IS USED. IT IS NOT
C AVAILABLE ON ALL COMPUTERS.

```

```

C DEVELOPED BY R. RUDA APRIL 01,1972
C LAST REVISION BY JOHN ADMIRE *NASA* JAN 1974.

```

#### FUNCTION ARGUMENTS

```

C NAMEIN = INPUT ALPHANUMERIC NAME. MUST BE SUPPLIED IN CALLING
C PROGRAM AS A 6H , OR BY A VARIABLE DEFINED
C WITH AN A6 FORMAT.
C NUMIN = INPUT A POSITIVE INTEGER NUMBER TO BE MERGED INTO NAMEIN.
C NAME = OUTPUT ALPHANUMERIC NAME WHICH MAY BE USED IN AN A6
C OUTPUT FORMAT.

```

#### EXAMPLES,

```

C CALL WRITE (A,NR,NC,NAME(3HABC,69),KA)
C WOULD YIELD AN OUTPUT NAME ABC69 (LEFT JUSTIFIED).
C CALL WRITE (A,NA,NA,NAME(1HK,NA),KA) , WHERE NA = 124,
C WOULD YIELD AN OUTPUT NAME K124 (LEFT JUSTIFIED).
C DO 5 I=1,N
C 5 CALL WRITE (A,NA,NA,NAME(4HSTIF,I),KA)
C WOULD YIELD OUTPUT NAMES STIF1 ,STIF2 ,STIF3 ,...

```

#### 4000 FORMAT (I1)

```

C IF (NUMIN.LT.0 .OR. NUMIN.GT.999999) RETURN

```

```

C FIND NUMBER OF DIGITS IN NUMIN.
C DO 10 ND=1,6
C IF (NUMIN.LT.10**ND) GO TO 20
C 10 CONTINUE
C 20 IF (ND.EQ.6) GO TO 50

```

```

C FIND NUMBER OF LETTERS IN NAMEIN.
C DO 30 I=1,6
C NL = 7-I
C N1 = (NL-1)*6
C IF (FLD(N1,6,NAMEIN) .NE. 6H( ( ( ( ( ) GO TO 40
C 30 CONTINUE
C GO TO 50
C 40 IF (NL+ND.GT.6) NL = 6-ND

```

```

C MERGE NAMEIN AND NUMIN INTO NAME.

```

```
NTOT = NL+ND  
ENCODE (4000,FMT1(2)) NL  
ENCODE (4000,FMT1(4)) ND  
ENCODE (FMT1,NAME) NAMEIN,NUMIN  
RETURN
```

C

```
50 ENCODE (4000,FMT2(2)) ND  
ENCODE (FMT2,NAME) NUMIN  
RETURN  
END
```

ONES

SUBROUTINE ONES (Z,NR,NC,KR)  
DIMENSION Z(KR,1)

C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C

GENERATE A MATRIX OF ONES.  
CODED BY RL WOHLER. FEB 1965.  
SUBROUTINE ARGUMENTS  
Z = OUTPUT MATRIX GENERATED. SIZE(NR,NC).  
NR = INPUT NUMBER OF ROWS IN MATRIX Z.  
NC = INPUT NUMBER OF COLS IN MATRIX Z.  
KR = INPUT ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.

DO 10 I=1,NR  
DO 10 J=1,NC  
10 Z(I,J) = 1.0  
RETURN  
END

ONRBM  
-----

```

SUBROUTINE ONRBM (RBM,AMASS, N ,NRBM, K)
DIMENSION RBM(K,1),AMASS(K,1),EVAL(6),EVEC(6,6),B(6,6)

```

```

C ORTHONORMALIZES THE RIGID BODY MODE(S).
C NORMALIZATION IS RBM(TRANSPOSE)*AMASS*RBM = UNITY.
C CALLS FORMA SUBROUTINES BTAB,EIGN1,MULTA,PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE
C   N   = 250   (BASED ON ETAB)
C   NRBM =   6
C DEVELOPED BY CS BODLEY AND RF HRUDA. DECEMBER 1965.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS
C RBM   = INPUT ANY RIGID BODY MODES. SIZE(N,NRBM). *DESTROYED*
C       = OUTPUT ORTHONORMAL RIGID BODY MODES. SIZE(N,NRBM).
C AMASS = INPUT CORRESPONDING MASS MATRIX. SIZE(N,N).
C N     = INPUT SIZE OF MASS MATRIX, NUMBER OF ROWS IN RBM. MAX=250.
C NRBM  = INPUT NUMBER OF RIGID BODY MODES, COLUMNS IN RBM. MAX=6.
C K     = INPUT ROW DIMENSION SIZE OF RBM AND AMASS IN CALLING PROGRAM.

```

```

C ERROR EXPLANATION
C 1 = MORE THAN 6 RIGID BODY MODES.

```

NERROR=1

```

IF(NRBM .GT. 6) GO TO 999

```

```

DO 20 J=1,NRBM
RMAX = ABS(RBM(1,J))
DO 10 I=2,N
10 IF( ABS(RBM(I,J)) .GT. RMAX ) RMAX = ABS(RBM(I,J))
DO 20 I=1,N
20 RBM(I,J) = RBM(I,J)/RMAX

```

```

CALL BTAB (AMASS,RBM,B , N ,NRBM, K,6)
CALL EIGN1 (B ,EVAL,EVEC, NRBM,1.E-10, 6)
DO 30 J=1,NRBM
DO 30 I=1,NRBM
30 EVEC(I,J) = EVEC(I,J)/SQRT(EVAL(J))
CALL MULTA (RBM,EVEC, N ,NRBM,NRBM, K,6)
RETURN

```

```

999 CALL ZZBOMB (6HONRBM ,NERROR)
END

```

```

SUBROUTINE ORDALP(IMAT, NR, NC, NCAL, IWMAT, KR, KCW)
DIMENSION IMAT(KR,1), IWMAT(KR,1), IVA(37)
COMMON / LWRKVI / IV(180)
DATA IVA/6H'0',6H'1',6H'2',6H'3',6H'4',
*        6H'5',6H'6',6H'7',6H'8',6H'9',6H'A',
*        6H'R',6H'C',6H'D',6H'E',6H'F',6H'G',
*        6H'H',6H'I',6H'J',6H'K',6H'L',6H'M',
*        6H'N',6H'O',6H'P',6H'Q',6H'R',6H'S',
*        6H'T',6H'U',6H'V',6H'W',6H'X',6H'Y',
*        6H'Z/

```

```

C THIS ROUTINE REORDERS THE ROWS OF A MATRIX ALPHABETICALLY ACCORDING
C TO THE FIRST NCAL COLUMNS OF THE MATRIX. EACH ELEMENT IN THE FIRST
C NCAL COLUMNS IS ASSUMED TO CONTAIN SIX CHARACTERS. THE CHARACTERS
C ARE RESTRICTED TO LETTERS, NUMBERS AND SPACES.

```

```

C NOTE< IF THE FIRST NCAL COLUMNS ARE THE SAME FOR TWO ROWS THEIR
C ORDER MAY BE REVERSED AFTER CALLING THIS ROUTINE.

```

#### SUBROUTINE ARGUMENTS

```

C IMAT - INPUT MATRIX TO BE REORDERED
C NR - INPUT NUMBER OF ROWS IN IMAT
C NC - INPUT NUMBERS OF COLUMNS IN IMAT
C NCAL - INPUT NUMBER OF COLUMNS IN IMAT TO BE USED FOR REORDERING
C MAXIMUM VALUE OF NCAL=30.
C IWMAT - MATRIX WORKING SPACE (KR BY KCW)
C KR - INPUT ROW DIMENSION IN CALLING PROGRAM FOR IMAT AND IWMAT
C KCW - INPUT COLUMN DIMENSION IN CALLING PROGRAM FOR IWMAT
C KCW .GE. NCAL

```

```

C CODED BY JOHN ADMIRE *NASA* DEC 1974.

```

```

C I108 SYSTEM ROUTINE FLD IS CALLED.

```

```

C FORMA ROUTINE ZZBOMB IS CALLED.

```

```

C
NCW=NCAL*6
NERROR=1
IF(NCW .GT. KCW) GO TO 999
NERROR=2
IF(NCW .GT. 180) GO TO 999
DO 20 I=1, NR
DO 20 J=1, NCAL
DO 20 L=1, 6
JJ=(J-1)*6+L
II=IABS((L-1)*6)
IWMAT(I, JJ)=FLD(II, 6, IMAT(I, J))
DO 10 K=1, 37
IF(IWMAT(I, JJ) .NE. IVA(K)) GO TO 10
IWMAT(I, JJ)=K
GO TO 20
10 CONTINUE
NERROR=3
GO TO 999
20 CONTINUE
NRM=NR-1

```



```
DO 100 I=1,NRM
DO 30 L=1,NCW
30 IV(L)=IWMAT(I,L)
II=I
IP=I+1
DO 70 J=IP,NR
DO 40 L=1,NCW
IF(IWMAT(J,L)-IV(L))50,40,70
40 CONTINUE
GO TO 70
50 DO 60 L=1,NCW
60 IV(L)=IWMAT(J,L)
II=J
70 CONTINUE
IF(II .EQ. J) GO TO 100
DO 80 L=1,NC
IA=IMAT(II,L)
IMAT(II,L)=IMAT(I,L)
80 IMAT(I,L)=IA
DO 90 L=1,NCW
IA=IWMAT(II,L)
IWMAT(II,L)=IWMAT(I,L)
90 IWMAT(I,L)=IA
100 CONTINUE
RETURN
999 CALL ZZBOMB(6HORDALP,NERROR)
END
```

OUT

-----

SUBROUTINE OUT (NTAPE,A,N)  
DIMENSION A(1)

C  
C WRITE DATA FROM CORE SPACE A OUT TO NTAPE.  
C CODED BY RL WOHLER. MARCH 1976.

C  
C SUBROUTINE ARGUMENTS (ALL INPUT)  
C NTAPE = NUMBER OF TAPE. (EG 10).  
C A = DATA TO BE WRITTEN ON NTAPE.  
C N = NUMBER OF WORDS OF DATA TO BE WRITTEN ON NTAPE.

C  
C WRITE (NTAPE) (A(I),I=1,N)  
C RETURN  
C END

```

SUBROUTINE PA(P,A,Z,NR,NC,KRA,KRZ)
DIMENSION A(KRA,1),Z(KRZ,1)

```

```

C PA PERFORMS THE OPERATION (Z)=P*(A)
C WHERE (Z) AND (A) ARE MATRICES AND P IS A SCALAR.
C PA CAN ALSO PERFORM THE OPERATION
C (A)=P*(A) BY CALL PA(P,A,A,--ETC--)

```

```

C IF NR IS NEGATIVE AND ABS(NR) IS EQUAL TO NC
C A SQUARE, SYMMETRIC (Z) IS COMPUTED USING THE UPPER HALF OF (A).

```

```

C FORMA SUBROUTINE ZZBOMB IS CALLED .
C CODED BY JOHN ADMIRE *NASA* JULY 1972 .
C LAST REVISION BY RL WOHLER. APRIL 1976.

```

ARGUMENTS

```

C P - INPUT SCALAR P
C A - INPUT MATRIX (A) SIZE(NR BY NC)
C Z - OUTPUT MATRIX (Z) SIZE(NR BY NC)
C NR - INPUT ABS(NR) IS THE NUMBER OF ROWS IN (A)
C NC - INPUT NUMBER OF COLUMNS IN (A)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

```

NERROR EXPLANATIONS

```

C 1 = SIZE EXCEEDS DIMENSIONS.
C 2 = NON-SQUARE (Z) WANTED.

```

```

N=IABS(NR)

```

NERROR = 1

```

IF(N .GT. KRA .OR. N .GT. KRZ) GO TO 999
IF(NR .LT. 0) GO TO 40
C=ABS(P-1.)
IF(C .GT. 1.0E-7) GO TO 20
DO 10 I=1,NR
DO 10 J=1,NC
10 Z(I,J)=A(I,J)
RETURN

```

```

20 DO 30 I=1,NR
DO 30 J=1,NC
30 Z(I,J)=P*A(I,J)
RETURN

```

NERROR = 2

```

40 IF(N .NE. NC ) GO TO 999
C=ABS(P-1.)
IF(C .GT. 1.0E-7) GO TO 60
DO 50 I=1,N
DO 50 J=I,N
Z(I,J)=A(I,J)
50 Z(J,I) = Z(I,J)
RETURN
60 DO 70 I=1,N
DO 70 J=I,N
Z(I,J)=P*A(I,J)
70 Z(J,I) = Z(I,J)
RETURN

```

PA -- 2/ 2  
-----

999 CALL ZZBOMB(6HPA ,NERROR)  
END

PAGEHD

```
-----  
SUBROUTINE PAGEHD  
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)  
COMMON /LLINE / NLINE,MAXLIN,MINI  
DATA NIT,NOT/5,6/  
C  
C BRINGS UP NEW PAGE AND PUTS HEADING AT TOP.  
C INCREASES PAGE NUMBER BY ONE AND SETS LINE NUMBER EQUAL TO FIVE.  
C  
C INTERNAL VARIABLES  
C IRUNNO = RUN NUMBER (A6 FORMAT)  
C DATE = DATE (A6 FORMAT)  
C NPAGE = PAGE NUMBER  
C UNAME = USERS NAME (3A6 FORMAT)  
C TITLE1 = FIRST TITLE (12A6 FORMAT)  
C TITLE2 = SECOND TITLE (12A6 FORMAT)  
C NLINE = LINE NUMBER  
C MAXLIN = MAXIMUM NUMBER OF LINES PER PAGE  
C MINI = PRINT OPTION (A4 FORMAT)  
C  
C MODIFIED AUG 1973 BY JOHN ADMIRE *NASA*  
C  
C 2001 FORMAT(9H1RUN NO. A6,32X,5HDATE A6,12H CPU TIME=I4,  
* 4H SEC,32X,9HPAGE NO. I4,/55X7HRUN BY 3A6//10X,  
* 12A6/10X,12A6)  
C  
C CALL CPUTIM(ISEC)  
C ISEC=ISEC/1000000  
C NPAGE=NPAGE+1  
C NLINE=5  
C WRITE (NOT,2001) IRUNNO,DATE,ISEC,NPAGE,UNAME,TITLE1,TITLE2  
C RETURN  
C END
```

SUBROUTINE PAQB(P,A,Q,B,Z,NR,NC,KRA,KRB,KRZ)  
 DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)

PAQB PERFORMS THE OPERATION  $Z = P * (A) + Q * (B)$   
 WHERE (A),(B) AND (Z) ARE MATRICES AND  
 P AND Q ARE SCALARS .

PAQB CAN ALSO PERFORM THE OPERATIONS  
 (A)=P\*(A)+Q\*(B) BY CALL PAQB(P,A,Q,B,A,—ETC—)  
 (B)=P\*(A)+Q\*(B) BY CALL PAQB(P,A,Q,B,B,—ETC—)  
 (Z)=P\*(A)+Q\*(A) BY CALL PAQB(P,A,Q,A,Z,—ETC—)  
 (A)=P\*(A)+Q\*(A) BY CALL PAQB(P,A,Q,A,A,—ETC—)

IF NR IS NEGATIVE AND ABS(NR) IS EQUAL TO NC  
 A SQUARE, SYMMETRIC (Z) IS COMPUTED USING THE UPPER HALF OF (A),(B).

FORMA SUBROUTINE ZZBOMB IS CALLED .  
 CODED BY JOHN ADMIRE \*NASA\* JULY 1972 .  
 LAST REVISION BY RL WOHLN. APRIL 1976.

ARGUMENTS

P - INPUT SCALAR P  
 A - INPUT MATRIX (A) SIZE(NR BY NC)  
 Q - INPUT SCALAR Q  
 B - INPUT MATRIX (B) SIZE(NR BY NC)  
 Z - OUTPUT MATRIX (Z) SIZE(NR BY NC)  
 NR - INPUT ABS(NR) IS THE NUMBER ROWS IN (A), (B) AND (Z)  
 NC - INPUT NC IS THE NUMBER OF COLUMNS IN (A), (B) AND (Z)  
 KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM  
 KRB - INPUT ROW DIMENSION OF (B) IN CALLING PROGRAM  
 KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM

NERROR EXPLANATIONS

1 = SIZE EXCEEDS DIMENSIONS.  
 2 = NON-SQUARE (Z) WANTED.  
 3 = NON-SQUARE (Z) WANTED.

N=IABS(NR)

NERROR = 1

IF(N .GT. KRA .OR. N .GT. KRB .OR. N .GT. KRZ) GO TO 999

CP=ABS(P-1.)

CQ=ABS(Q-1.)

IF(CP .GT. 1.0E-7 .OR. CQ .GT. 1.0E-7) GO TO 40

IF(NR .LT. 0) GO TO 20

DO 10 I=1,NR

DO 10 J=1,NC

10 Z(I,J)=A(I,J)+B(I,J)

RETURN

20

NERROR = 2

IF(N .NE. NC) GO TO 999

DO 30 I=1,N

DO 30 J=I,N

Z(I,J) = A(I,J) + B(I,J)

30 Z(J,I) = Z(I,J)

RETURN

40 IF(NR .LT. 0) GO TO 60

```
DO 50 I=1,NR
DO 50 J=1,NC
50 Z(I,J)=P*A(I,J)+Q*B(I,J)
RETURN
60
IF(N .NE. NC) GO TO 999
DO 70 I=1,N
DO 70 J=I,N
Z(I,J)=P*A(I,J)+Q*B(I,J)
70 Z(J,I) = Z(I,J)
RETURN
999 CALL ZZBOMB(6HPAQB ,NERROR)
END
```

NERROR = 3

```

SUBROUTINE PLOT1 (XVEC,YMAT,NR,NC,IXNAME,IYNAME,ITITLE,IFCURV,K)
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
DIMENSION XVEC(1),YMAT(K,1),IYNAME(1),ITITLE(1),ITITL(12)
DATA NIT,NOT/5,6/
EQUIVALENCE (IDATE,DATE)

```

```

C
C PLOTS FROM 1 TO 3 VECTORS PER FRAME. X-AXIS AND Y-AXIS ARE LINEAR.
C CALLS FORMA SUBROUTINES PLOTSS, ZZBOMB.
C THE MAXIMUM SIZE IS
C   NC=3
C CODED BY RF HRUDA 01,JULY 1968
C MODIFIED FOR CONTRACT NAS8-25922, MAY 1971.
C
C NOTE...FORTRAN STATEMENT -CALL IDENT (1)- MUST BE IN LOGIC OF MAIN
C PROGRAM PRIOR TO CALLING THIS ROUTINE. IT MUST BE EXECUTED ONLY
C ONCE (INDEPENDENTLY OF NUMBER OF TIMES MAIN BODY OF THE PROGRAM
C IS EXECUTED).
C
C FORTRAN STATEMENT -CALL ENDJOB- MUST BE IN LOGIC OF MAIN PROGRAM
C SUBSEQUENT TO CALLING THIS ROUTINE. IT MUST BE EXECUTED ONLY ONCE.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C XVEC = VECTOR OF X-AXIS COORDINATES. SIZE(NR).
C YMAT = SET OF NC VECTORS TO BE PLOTTED SIMULTANECUSLY. SIZE(NR,NC).
C NR = NO. OF ROWS IN YMAT (AND XVEC)
C NC = NO. OF COLS (OR VECTORS) IN YMAT
C IXNAME = AN A6 NAME FOR X-AXIS
C IYNAME = A 12A6 NAME FOR Y-AXIS
C (CAN BE READ IN MAIN PROGRAM WITH A FORMAT (12A6))
C ITITLE = A 6A6 DEFINED IN THE CALLING PROGRAM WHICH WILL BE
C ASSEMBLED WITH IRUNNO AND DATE TO FORM TITLE. TITLE WILL
C APPEAR AT BOTTOM OF PLOT SHEET.
C IFCURV = 1 IF CONNECTED CURVE PLOT IS DESIRED
C = 0 IF DOT PLOT IS DESIRED
C K = ROW DIMENSION SIZE OF YMAT IN MAIN PROGRAM
C
C NERROR EXPLANATIONS
C 1 = MORE THAN 3 COLUMNS IN (YMAT).
C 2 = IFCURV IS NOT 0 OR 1.
C
C CHECK ON NO. OF VECTORS
C
C IF (NC.GT.3) GO TO 999
C
C FORM TITLE FROM ITITLE AND COMMON.
C ITITL(1) = IXNAME
C ITITL(2) = 6H
C ITITL(3) = IRUNNO
C ITITL(4) = 6H
C ITITL(5) = IDATE
C ITITL(6) = 6H
C DO 5 I=1,6
C 5 ITITL(I+6) = ITITLE(I)
C

```

NERROR=1



```
-----  
FIND MAX. AND MIN. OF YMAT , XVEC  
  YMAX=YMAT(1,1)  
  YMIN=YMAT(1,1)  
  XMAX = XVEC(1)  
  XMIN = XVEC(1)  
  DO 10 I=1,NR  
    IF (XVEC(I) .GT. XMAX) XMAX = XVEC(I)  
    IF (XVEC(I) .LT. XMIN) XMIN = XVEC(I)  
  DO 10 J=1,NC  
    IF(YMAT(I,J) .GT. YMAX) YMAX=YMAT(I,J)  
  10 IF(YMAT(I,J) .LT. YMIN) YMIN=YMAT(I,J)  
C FIND TOP AND BOTTOM VALUES FOR PLOT FRAME (YT AND YB)  
  CALL PLOTSS (YMAX,YMIN,YT,YB)  
C  
  NP=0  
  IF (IFCURV .EQ. 0) NP=+NR  
  IF (IFCURV .EQ. 1) NP=-NR  
  IF(IFCURV .EQ. 0) JI=40  
  IF(IFCURV .EQ. 1) JI=61  
NERROR=2  
  IF (NP .EQ. 0) GO TO 999  
  NEWGRD = -1  
  DO 20 I=1,NC  
    IF (I .GT. 1) NEWGRD=0  
  20 CALL QUIK3L (NEWGRD,XMIN,XMAX,YB,YT,JI,ITITL,IYNAME,NP,XVEC,  
  *           YMAT(1,I))  
  RETURN  
999 CALL ZZBOMB (6HPLOT1 ,NERROR)  
  END
```

```

SUBROUTINE PLOT2 (XVEC,YMAT,NR,NC,IXNAME,IYNAME,ITITLE,IPLOT,
*                YTOP,YBOT,XLEFT,XRIGHT,KR)
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
DIMENSION XVEC(1),YMAT(KR,1),ITITLE(1),ITITL(12),IND(10)
DIMENSION IYNAME (1)
DATA NIT,NOT/5,6/
DATA IND / 2H 1,2H 2,2H 3,2H 4,2H 5,2H 6,2H 7,2H 8,2H 9,2H10 /
EQUIVALENCE (DATE,IDATE)

```

C SUBROUTINE PRODUCES LOG-LOG, SEMILOG-LINEAR, LINEAR-SEMILOG PLOTS.  
C WILL PLOT UP TO 10 CURVES PER GRID. ALL CURVES WILL BE PLOTTED  
C VERSUS XVEC AND WILL BE PLOTTED WITH THE SAME Y AXIS SCALE.

C NOTE...FORTRAN STATEMENT -CALL IDENT (1)- MUST BE IN LOGIC OF MAIN  
C PROGRAM PRIOR TO CALLING THIS ROUTINE. IT MUST BE EXECUTED ONLY  
C ONCE (INDEPENDENTLY OF NUMBER OF TIMES MAIN PROGRAM IS  
C EXECUTED).

C FORTRAN STATEMENT -CALL ENDJOB- MUST BE IN LOGIC OF MAIN PROGRAM  
C SUBSEQUENT TO CALLING THIS ROUTINE. IT MUST BE EXECUTED  
C ONLY ONCE.

C THE MAXIMUM SIZE IS

C NC = 10

C CALLS FOR A SUBROUTINE ZIBOMB.

C CODED BY R L BERRY. MAY 1969.

C MODIFIED FOR CONTRACT NAS8-25922, MAY 1971.

C SUBROUTINE ARGUMENTS (ALL INPUT)

C XVEC = VECTOR OF X-AXIS COORDINATES. SIZE(NR).

C YMAT = MATRIX OF Y-AXIS COORDINATES TO BE PLOTTED. SIZE(NR,NC).  
C MAY BE DESTROYED.

C NR = NUMBER OF ROWS IN XVEC AND YMAT.

C NC = NUMBER OF COLUMNS IN YMAT. MAX=10.

C IXNAME= AN A6 NAME FOR X-AXIS COORDS.

C IYNAME= A 12A6 NAME FOR Y-AXIS COORDS.

C ITITLE= A 6A6 DEFINED IN THE CALLING PROGRAM WHICH WILL BE  
C ASSEMBLED WITH IRUNNO AND DATE TO FORM TITLE. TITLE  
C WILL APPEAR AT BOTTOM OF PLOT SHEET.

C IPLOT = THE TYPE OF PLOT DESIRED.

C = 1 ESTABLISHES A SEMI-LOG MAPPING WITH Y-AXIS LINEAR.

C = 2 ESTABLISHES A SEMI-LOG MAPPING WITH X-AXIS LINEAR.

C = 3 ESTABLISHES A LOG-LOG MAPPING.

C YTOP = MAXIMUM VALUE OF Y-AXIS SCALE.

C IF .LE. 0. AND LOG AXIS, MAXIMUM VALUE WILL BE COMPUTED.

C YBOT = MINIMUM VALUE OF Y-AXIS SCALE.

C IF .LE. 0. AND LOG AXIS, MINIMUM VALUE WILL BE COMPUTED.

C XLEFT = MINIMUM VALUE OF X-AXIS SCALE.

C IF .LE. 0. AND LOG AXIS, MINIMUM VALUE WILL BE COMPUTED.

C XRIGHT= MAXIMUM VALUE OF X-AXIS SCALE.

C IF .LE. 0. AND LOG AXIS, MAXIMUM VALUE WILL BE COMPUTED.

C KR = ROW DIMENSION OF YMAT IN CALLING PROGRAM.

C ERROR EXPLANATIONS

C 1 = MORE THAN 10 COLUMNS IN (YMAT).

CHECK SIZE LIMITATION OF PROGRAM—NUMBER OF VECTORS TO BE PLOTTED  
NERROR=1

IF (NC .GT. 10) GO TO 999

C  
C FORM TITLE FROM ITITLE AND COMMON

ITITL(1) = IXNAME

ITITL(2) = 6H

ITITL(3) = IRUNNO

ITITL(4) = 6H

ITITL(5) = IDATE

ITITL(6) = 6H

DO 5 I=1,6

5 ITITL(I+6) = ITITLE(I)

C  
C FIND MAX AND MIN OF YMAT,XVEC.

XMAX=XVEC(1)

XMIN=XVEC(1)

YMAX=YMAT(1,1)

YMIN=YMAT(1,1)

DO 12 I=1,NR

IF(XVEC(I).GT. XMAX) XMAX=XVEC(I)

IF(XVEC(I).LT. XMIN) XMIN=XVEC(I)

DO 12 J=1,NC

IF(YMAT(I,J) .GT. YMAX) YMAX=YMAT(I,J)

IF(YMAT(I,J) .LT. YMIN) YMIN=YMAT(I,J)

12 CONTINUE

XL = XLEFT

XR = XRIGHT

YE = YBOT

YT = YTOP

IF (IPLOT .EQ. 2) GO TO 60

C  
C X LOG SCALE DETERMINATION SECTION.

IF (XLEFT.GT.0) GO TO 50

X = ALOG10 (XMIN)

IF (X .LT. 0.) GO TO 45

I = X + 1.

GO TO 48

45 I = X

Y = I - 1

IF (X-Y .GE. 1.) I = I+1

48 XL = 10.\*\*(I - 1)

50 IF (XRIGHT .GT.0) GO TO 60

X = ALOG10 (XMAX)

IF (X .GE. 0.) GO TO 52

I = X

GO TO 53

52 I = X + 1.

Y = I

IF (Y-X .GE. 1.) I = I - 1

53 XR = 10.\*\*I

C  
C Y LOG SCALE DETERMINATION SECTION  
60 IF (IPLOT .EQ. 1) GO TO 13

```
      IF (YTOP .GT. 0.) GO TO 70
      X=ALOG10(YMAX)
      IF (X.GE.0.) GO TO 103
      I=X
      GO TO 104
103  I=X+1.
      Y=I
      IF (Y-X.GE.1.) I=I-1
104  Y=10.**I
      70 IF (YBOT .GT. 0.) GO TO 13
      X = ALOG10 (YMIN)
      IF (X .LT. 0.) GO TO 75
      I = X + 1.
      GO TO 80
      75 I = X
      Y = I - 1
      IF (X-Y .GE. 1.) I = I+1
      80 YB = 10. **(I-1)
C
C  PRODUCE APPROPRIATE GRID
      13 IF (IPILOT .EQ. 1) CALL SMXYV (1,0)
      IF (IPILOT .EQ. 2) CALL SMXYV (0,1)
      IF (IPILOT .EQ. 3) CALL SMXYV (1,1)
      DO 130 I=1,NR
      IF (XVEC(I) .GT. XR) XVEC(I) = XR
      IF (XVEC(I) .LT. XL) XVEC(I) = XL
      DO 130 J=1,NC
      IF (YMAT(I,J) .GT. YT) YMAT(I,J) = YT
130  IF( YMAT(I,J) .LT. YB ) YMAT(I,J)=YB
C
C  PLOT CURVES
      NEWGRD = -1
      DO 40 I=1,NC
      CALL XSCLV1 (XVEC(I),IXRAS,IXERR)
      CALL YSCLV1 (YMAT(I,I),IYRAS,IYERR)
      CALL PRINTV (2,IND(I),IXRAS,IYRAS)
      CALL XSCLV1 (XVEC(NR),IXRAS,IXERR)
      CALL YSCLV1 (YMAT(NR,I),IYRAS,IYERR)
      CALL PRINTV (2,IND(I),IXRAS,IYRAS)
      IF (I.GT.1) NEWGRD = 0
      40 CALL QUKLOG (NEWGRD,XL ,XR ,YB ,YT ,61,ITITL,IYNAME,-NR,XVEC,
      *           YMAT(I,I))
C
C  RETURN TO MAIN PROGRAM
      CALL SMXYV (0,0)
      RETURN
999  CALL ZZBOMB (6HPLOT2 ,NERROR)
      END
```

SUBROUTINE PLOT3 (CLOC,MLOC,COELOC,VPLOC,RANGLE,CANGLE,EED,  
\* IFJNUM,LREYE,NVIEW,IFFA,ITITLE,NC,NM,KC,KM)

C

COMMON /LSTART/ IFUNNO,DATE,NFAGE,UNAME(3),TITLE1(12),TITLE2(12)  
DIMENSION CLOC(KC,1),MLOC(KM,1),COELOC(1),VPLOC(1),ITITLE(1)  
DIMENSION ABE(3),AQ1E(3),AQ2E(3),EA(3),EAE(3),ER(3),EBE(3),  
\* ECE(3),EO(3),EP(3),EPE(3),EQ1E(3),EQ2E(3),RE(3),  
\* UX(3),UY(3),UZ(3), ITITLX(13)  
DIMENSION NUMBER(100),IDATA(3)  
DATA NIT,NOT/5,6/  
DATA TOLRNC / 1.E-08 /  
DATA NUMBER / 3H 1,3H 2,3H 3,3H 4,3H 5,3H 6,3H 7,3H 8,  
\* 3H 9,3H 10,3H 11,3H 12,3H 13,3H 14,3H 15,3H 16,  
\* 3H 17,3H 18,3H 19,3H 20,3H 21,3H 22,3H 23,3H 24,  
\* 3H 25,3H 26,3H 27,3H 28,3H 29,3H 30,3H 31,3H 32,  
\* 3H 33,3H 34,3H 35,3H 36,3H 37,3H 38,3H 39,3H 40,  
\* 3H 41,3H 42,3H 43,3H 44,3H 45,3H 46,3H 47,3H 48,  
\* 3H 49,3H 50,3H 51,3H 52,3H 53,3H 54,3H 55,3H 56,  
\* 3H 57,3H 58,3H 59,3H 60,3H 61,3H 62,3H 63,3H 64,  
\* 3H 65,3H 66,3H 67,3H 68,3H 69,3H 70,3H 71,3H 72,  
\* 3H 73,3H 74,3H 75,3H 76,3H 77,3H 78,3H 79,3H 80,  
\* 3H 81,3H 82,3H 83,3H 84,3H 85,3H 86,3H 87,3H 88,  
\* 3H 89,3H 90,3H 91,3H 92,3H 93,3H 94,3H 95,3H 96,  
\* 3H 97,3H 98,3H 99,3H 100 /  
EQUIVALENCE (DATE,IDATE)

C

PLOTS PERSPECTIVE OR STEREO-PAIR VIEW(S).

C

CODED BY R F HRUDA. OCTOBER 1968.

C

MODIFIED FOR CONTRACT NAS8-25922, MAY 1971.

C

C

NOTE...FORTRAN STATEMENT -CALL IDENT (1)- MUST BE IN LOGIC OF MAIN  
PROGRAM PRIOR TO CALLING THIS ROUTINE. IT MUST BE EXECUTED  
ONLY ONCE (INDEPENDENTLY OF THE NUMBER OF TIMES THE MAIN  
PROGRAM IS EXECUTED).

C

C

C

FORTRAN STATEMENT -CALL ENDJOB- MUST BE IN LOGIC OF MAIN  
PROGRAM SUBSEQUENT TO CALLING THIS ROUTINE. IT MUST BE EXECUTED  
ONLY ONCE.

C

C

THIS ROUTINE CALLS FORMA SUBROUTINES VCROSS, VDOT, ZZBOMB.

C

C

SUBROUTINE ARGUMENTS (ALL INPUT)

C

CLOC = NC-BY-3 MATRIX. THE I-TH ROW WOULD DEFINE THE X,Y,Z  
COORDINATE LOCATION OF THE I-TH JOINT OF A STRUCTURE.

C

MLOC = NM-BY-2 MATRIX. THIS MATRIX SPECIFIES WHICH COORDINATES  
IN THE CLOC ARE TO BE CONNECTED BY A STRAIGHT LINE PLGT.  
(I.E. PLOT FROM CLOC(MLOC(I,1)) TO CLOC(MLOC(I,2)).)

C

C

COELOC = A VECTOR DEFINING THE X,Y,Z COORDINATES OF THE  
CENTER-OF-EYES-LOCATION IN THE REFERENCE COORDINATE SYSTEM.  
(WHERE YOU VIEW THE OBJECT FROM.)

C

C

VPLOC = A VECTOR DEFINING THE X,Y,Z COORDINATES OF THE  
VIEW-POINT-LOCATION. (A POINT YOU WISH TO LOOK AT  
FROM THE COELOC.)

C

C

RANGLE = ROLL-ANGLE (IN DEGREES) YOU WISH TO ROLL YOUR HEAD ABOUT  
THE LINE OF SIGHT PRESCRIBED BY COELOC AND VPLOC.

C

C

(CLOCKWISE = +) (THE INITIAL LINE OF SIGHT FROM COELOC IS DOWN THE X-AXIS WITH THE Y-AXIS TO THE RIGHT AND THE Z-AXIS DOWN. TO LOOK AT THE VPLOC, THE LOCAL COORDINATE SYSTEM AT COELOC IS FIRST ROTATED ABOUT THE Z-AXIS AND THEN ABOUT THE Y-AXIS. RANGLE WOULD THEN PRESCRIBE A +THETA-X ROTATION ABOUT THE NEW ORIENTATION OF THE LOCAL COORDINATE SYSTEM.)

CANGLE = CONE ANGLE OF VISION. (A SCALING TYPE OF VARIABLE THAT IS DEPENDENT ON THE VIEWER. FOR MOST VIEWERS ABOUT 60 DEGREES IS USED. IF NO VIEWER IS USED, ABOUT 20 DEGREES IS ACCEPTABLE.) MAX = 80 DEGREES.

EED = EYE-TO-EYE DISTANCE (USUALLY 3.0 INCHES). A VARIATION OF THIS PARAMETER WILL CAUSE A DEPTH PERCEPTION DISTORTION.

IFJNUM = 0, NO JOINT NUMBERS WILL APPEAR ON THE STRUCTURE.

= 1, JOINT NUMBERS WILL BE PUT ON THE STRUCTURE.

LREYE = 1, A PERSPECTIVE (LEFT EYE) VIEW WILL BE PRODUCED.

= 2, COMPANION RIGHT EYE VIEW FOR STEREO WILL BE PRODUCED.

NVIEW = 1, THE PLOTTED IMAGE WILL BE FULL SIZE, AND ONLY ONE VIEW WILL APPEAR ON ONE PLOT FRAME.

= 2, THE PLOTTED IMAGE WILL BE HALF-SIZE SUCH THAT BOTH VIEWS OF A STEREO PAIR MAY BE PUT ON ONE PLOT FRAME. (TO BE USED IN CONJUNCTION WITH LREYE AND IFFA. FOR A STEREO PAIR, THE LEFT EYE VIEW MUST BE PLOTTED FIRST.)

IFFA = 0, FRAME ADVANCE WILL NOT BE EXECUTED AFTER PLOTTING A FRAME.

= 1, FRAME ADVANCE WILL BE EXECUTED AFTER PLOTTING IS COMPLETED. (MUST USE IFFA=1 ON LAST PLOT EXECUTION IN EACH FRAME.)

ITITLE = 13A6 PLOT TITLE. (CANNOT USE TITLE1 OR TITLE2 FROM SUBROUTINE START)

NC = NO. OF ROWS IN CLOC.

NM = NO. OF ROWS IN MLOC.

KC = ROW DIMENSION SIZE IN CALLING PROGRAM OF MATRIX CLOC.

KM = ROW DIMENSION SIZE IN CALLING PROGRAM OF MATRIX MLOC.

#### ERROR EXPLANATIONS

1 = CONE ANGLE GREATER THAN 80 DEGREES.

2 = EYE-TO-EYE DISTANCE LESS THAN 1.E-8.

3 = JOINT NUMBER OPTION (IFJNUM) MUST BE 0 OR 1.

4 = PERSPECTIVE OR STEREO OPTION (LREYE) MUST BE 1 OR 2.

5 = VIEW OPTION (NVIEW) MUST BE 1 OR 2.

6 = FRAME ADVANCE OPTION MUST BE 0 OR 1.

7 = CENTER OF EYES IS TOO CLOSE TO VIEW POINT.

8 = MATRIX (MLOC) DATA EXCEEDS MATRIX (CLOC) SIZE.

IF (CANGLE .GT. 80.) GO TO 999 NERROR=1

IF (EED.LE.TOLRNC) GO TO 999 NERROR=2

IF (IFJNUM.NE.0 .AND. IFJNUM.NE.1) GO TO 999 NERROR=3

IF (LREYE.NE.1 .AND. LREYE.NE.2) GO TO 999 NERROR=4

IF (NVIEW.NE.1 .AND. NVIEW.NE.2) GO TO 999 NERROR=5

NERROR=6

```

IF (IFFA.NE.0 .AND. IFFA.NE.1) GO TO 999

```

L  
C

```

C FORM SINES AND COSINES.

```

```

DX = VPLOC(1)-COELOC(1)
DY = VPLOC(2)-COELOC(2)
DZ = VPLOC(3)-COELOC(3)
OPM = SQRT(DX**2+DY**2+DZ**2)

```

NERROR=7

```

IF (OPM .LE. TOLRNC) GO TO 999

```

```

THETAX = RANGLE/57.2957
THETAY = ATAN2(-DZ,SQRT(DX**2+DY**2))
THETAZ = 0.

```

```

IF (ABS(DY).GT.TOLRNC .OR. ABS(DX).GT.TOLRNC) THETAZ = ATAN2(DY,DX)

```

```

S1 = SIN(THETAZ)
S2 = SIN(THETAY)
S3 = SIN(THETAX)
SEYE = SIN(ATAN(.5*EED/OPM))
SCONE = SIN(.5*CANGLE/57.2957)
C1 = COS(THETAZ)
C2 = COS(THETAY)
C3 = COS(THETAX)
CEYE = COS(ATAN(.5*EED/OPM))
CCONE = COS(.5*CANGLE/57.2957)

```

C

```

C FORM CONVERSION FACTOR (UNIT RASTERS/LENGTH).

```

```

IF (NVIEW.EQ.1) SCALE = 0.4399
IF (NVIEW.EQ.2) SCALE = 0.2499
CONVRT = SCALE/((OPM/CEYE)*TAN(.5*CANGLE/57.2957))

```

C

```

C SET UP DATA FOR PLOTTING TITLES.

```

```

DC 100 I=1,13
100 ITITLX(I) = ITITLE(I)

```

C

```

C PLOT TITLE DATA.

```

C

```

IF (IFFA.EQ.0) GO TO 105
IF (NVIFW.EQ.2) GO TO 103
IF (LPEYE .EQ. 1) IDATA(1) = 6H LEFT
IF (LREYE .EQ. 2) IDATA(1) = 6HRIGHT
IDATA(2) = 6HEYE VI
IDATA(3) = 2HEW
CALL RITE2V (10,512,1024,180,1,14,1,IDATA,IFPR)
103 CALL PRINTV (78,ITITLX,203,107)
CALL PRINTV (30,30HCENTER OF EYES LOCATION ,203,82)
CALL PRINTV (38,38HVIEW POINT LOCATION ROLL ANGLE =,443,82)
CALL LABLV (RANGLE,747,82,6,1,3)
CALL PRINTV (3,3HDEG,803,82)
CALL PRINTV (6,6H X =,203,61)
CALL LABLV (COELOC(1),259,61,-6,1,1)
CALL PRINTV (3,3HX =,451,61)
CALL LABLV (VPLOC(1),486,61,-6,1,1)
CALL PRINTV (19,19H CONE ANGLE =,595,61)
CALL LABLV (CANGLE,747,61,6,1,3)
CALL PRINTV (3,3HDEG,803,61)

```

```

CALL PRINTV (6,6H  Y =,203,41)
CALL LABLV (COELOC(2),259,41,-6,1,1)
CALL PRINTV (3,3HY =,451,41)
CALL LABLV (VPLOC(2),486,41,-6,1,1)
CALL PRINTV (19,19H      EYE TO EYE =,595,41)
CALL LABLV (EED,747,41,6,1,3)
CALL PRINTV (3,3H IN,803,41)
CALL PRINTV (6,6H  Z =,203,20)
CALL LABLV (COELOC(3),259,20,-6,1,1)
CALL PRINTV (3,3HZ =,451,20)
CALL LABLV (VPLOC(3),486,20,-6,1,1)
CALL PRINTV (10,10HRUN NO. = ,203,4)
CALL PRINTV (6,IRUNNO,284,4)
CALL PRINTV (17,17H      DATE = ,364,4)
CALL PRINTV (6,IDATE,500,4)

```

105 CONTINUE

C  
C TRANSFORM VECTORS FROM COE SYSTEM TO REFERENCE SYSTEM AND TAKE  
C ADVANTAGE OF ZEROS IN ORIGINAL VECTORS.

```

SIGN = +1.0
IF (LREYE.EQ.2) SIGN = -1.0
EO(1) = SIGN*.5*EED*(C1*S2*S3-S1*C3)
EO(2) = SIGN*.5*EED*(S1*S2*S3+C1*C3)
EO(3) = SIGN*.5*EED*(C2*S3)
EP(1) = OPM*(C1*C2)+EO(1)
EP(2) = OPM*(S1*C2)+EO(2)
EP(3) = OPM*(-S2) +EO(3)
RE(1) = COELOC(1)-EO(1)
RE(2) = COELOC(2)-EO(2)
RE(3) = COELOC(3)-EO(3)
CALL VCROSS (EO,EP,UX,EOM,EPM,UXM,SINAB)
CALL VCROSS (EP,UX,UY,EPM,UXM,UYM,SINAB)
CALL VCROSS (UX,UY,UZ,UXM,UYM,UZM,SINAB)

```

```

DO 140 NI=1,3
UX(NI) = SIGN*UX(NI)/UXM
UY(NI) = SIGN*UY(NI)/UYM
140 UZ(NI) = +1.0*UZ(NI)/UZM
EPE(1) = 0.
EPE(2) = 0.
EPE(3) = EPM

```

C  
C LOOP FOR PLOTTING THE NM MEMBERS.

C -----  
C DO 330 NMEM=1,NM  
C SET UP VECTORS FROM EYE TO MEMBER ENDS.  
NA = MLOC(NMEM,1)  
NB = MLOC(NMEM,2)

```

IF (NA.GT.NC .OR . NB.GT.NC) GO TO 999
DO 160 NI=1,3
EA(NI) = CLOC(NA,NI)-RE(NI)
160 EB(NI) = CLOC(NB,NI)-RE(NI)
EAE(1) = UX(1)*EA(1)+UX(2)*EA(2)+UX(3)*EA(3)
EAE(2) = UY(1)*EA(1)+UY(2)*EA(2)+UY(3)*EA(3)
EAE(3) = UZ(1)*EA(1)+UZ(2)*EA(2)+UZ(3)*EA(3)

```

NERRGR=8



```

EBE(1) = UX(1)*EB(1)+UX(2)*EB(2)+UX(3)*EB(3)
EBE(2) = UY(1)*EB(1)+UY(2)*EB(2)+UY(3)*EB(3)
EBE(3) = UZ(1)*EB(1)+UZ(2)*EB(2)+UZ(3)*EB(3)

```

```

C CHECK IF BOTH ENDS ARE BEHIND THE EYE.
  IF (EAE(3).LE.TOLRNC .AND. EEE(3).LE.TOLRNC) GO TO 330

```

```

C CHECK IF BOTH ENDS ARE IN CONE OF VISION.
  CALL VDOT (EPE,EAEM,PRODCT,EPEM,EAEM,COSPA)
  CALL VDOT (EPE,EBE,PRODCT,EPEM,EBEM,COSPB)
  IF (EAEM.LE.TOLRNC .OR. EBEM.LE.TOLRNC) GO TO 330
  IFNUM = 0
  IF (COSPA.LT.CCONE .OR. COSPB.LT.CCONE) GO TO 170
  IFNUM = 1
  PAX = (EPEM/COSPA)*(EAE(1)/EAEM)
  PAY = (EPEM/COSPA)*(EAE(2)/EAEM)
  PBX = (EPEM/COSPB)*(EBE(1)/EBEM)
  PBY = (EPEM/COSPB)*(EBE(2)/EBEM)
  GO TO 320

```

```

C
C FIND INTERSECTION OF LINE AND CONE AND DETERMINE WHICH SOLUTIONS
C ARE VALID FOR POINTS TO BE PROJECTED ONTO VIEWING PLANE.

```

```

170 CALL VCROSS (EAE,EBE,ECE,EAEM,EBEM,ECEM,SINAB)
  IF (ECEM.LE.TOLRNC) GO TO 330
  CALL VDOT (EPE,ECE,PRODCT,EPEM,ECEM,COSPC)
  C9MC = COS( (90.0-.5*CANGLE)/57.2957 )
  C9PC = COS( (90.0+.5*CANGLE)/57.2957 )
  IF (COSPC.GE.C9MC .OR. COSPC.LE.C9PC) GO TO 330
  BETA = ATAN2(ECE(1),ECE(2))
  SINPPB = (-1.0/TAN(.5*CANGLE/57.2957))*
*      (ECE(3)/SQRT(ECE(1)**2+ECE(2)**2))
  IF (SINPPB**2.GE..995) GO TO 330
  DENOM = SQRT(1.0-SINPPB**2)
  PHI1 = ATAN2(SINPPB,+DENOM)-BETA
  PHI2 = ATAN2(SINPPB,-DENOM)-BETA
  DO 180 N1=1,3

```

```

180 ABE(N1) = EBE(N1)-EAE(N1)
  IFUSE1 = 1
  IFUSE2 = 1
  R1 = 0.
  R2 = 0.
  IF (ABS(ECE(2)).GT.ABS(ECE(1))) GO TO 190
  DENOM1 = ABE(3)*SCONE*SIN(PHI1)-ABE(2)*CCONE
  DENOM2 = ABE(3)*SCONE*SIN(PHI2)-ABE(2)*CCONE
  IF (ABS(DENOM1).LE.TOLRNC) IFUSE1 = 0
  IF (ABS(DENOM2).LE.TOLRNC) IFUSE2 = 0
  IF (IFUSE1.EQ.1) R1 = ECE(1)/DENOM1
  IF (IFUSE2.EQ.1) R2 = ECE(1)/DENOM2
  GO TO 200

```

```

190 DENOM1 = ABE(1)*CCONE-ABE(3)*SCONE*COS(PHI1)
  DENOM2 = ABE(1)*CCONE-ABE(3)*SCONE*COS(PHI2)
  IF (ABS(DENOM1).LE.TOLRNC) IFUSE1 = 0
  IF (ABS(DENOM2).LE.TOLRNC) IFUSE2 = 0
  IF (IFUSE1.EQ.1) R1 = ECE(2)/DENOM1
  IF (IFUSE2.EQ.1) R2 = ECE(2)/DENOM2

```

```

200 IF (R1.LE.TOLRNC) IFUSE1 = 0
  IF (R2.LE.TOLRNC) IFUSE2 = 0

```

IF (IFUSE1.EQ.0 .AND. IFUSE2.EQ.0) GO TO 330

C FORM VECTORS FROM EYE TO PROJECTED POINT AND FORM X,Y COMPONENTS IN  
C THE P-SYSTEM.

IF (IFUSE1.EQ.0) GO TO 210  
EQ1E(1) = R1\*COS(PHI1)\*SCONE  
EQ1E(2) = R1\*SIN(PHI1)\*SCONE  
EQ1E(3) = R1\*CCONE

210 IF (IFUSE2.EQ.0) GO TO 220  
EQ2E(1) = R2\*COS(PHI2)\*SCONE  
EQ2E(2) = R2\*SIN(PHI2)\*SCONE  
EQ2E(3) = R2\*CCONE

220 IF (IFUSE1.EQ.1 .AND. IFUSE2.EQ.1) GO TO 260

C IF (COSPA.LE.CCONE .AND. COSPB.LE.CCONE) GO TO 330  
IF (IFUSE1.EQ.0) GO TO 230

EQ1EM = SQRT(EQ1E(1)\*\*2+EQ1E(2)\*\*2+EQ1E(3)\*\*2)  
PAX = (EPM/CCONE)\*(EQ1E(1)/EQ1EM)  
PAY = (EPM/CCONE)\*(EQ1E(2)/EQ1EM)  
GO TO 240

230 EQ2EM = SQRT(EQ2E(1)\*\*2+EQ2E(2)\*\*2+EQ2E(3)\*\*2)  
PAX = (EPM/CCONE)\*(EQ2E(1)/EQ2EM)  
PAY = (EPM/CCONE)\*(EQ2E(2)/EQ2EM)

240 IF (COSPA.LT.CCONE) GO TO 250  
PBX = (EPM/COSPA)\*(EAE(1)/EAEM)  
PBY = (EPM/COSPA)\*(EAE(2)/EAEM)  
GO TO 320

250 PBX = (EPM/COSPB)\*(EBE(1)/EBEM)  
PBY = (EPM/COSPB)\*(EBE(2)/EBEM)  
GO TO 320

C TWO INTERSECTION POINTS.  
C SEE IF BOTH ARE INSIDE OR BOTH ARE OUTSIDE OF AB.

260 DO 270 N1=1,3  
AQ1E(N1) = EQ1E(N1)-EAE(N1)  
270 AQ2E(N1) = EQ2E(N1)-EAE(N1)  
CALL VDOT (ABE,AQ1E,ABQ1,ABEM,AQ1EM,COSAQ1)  
CALL VDOT (ABE,AQ2E,ABQ2,ABEM,AQ2EM,COSAQ2)  
RATIO1 = ABQ1/ABEM\*\*2  
RATIO2 = ABQ2/ABEM\*\*2  
IF ((RATIO1.GE.1.0 .AND. RATIO2.GE.1.0) .OR.  
\* (RATIO1.LE.0.0 .AND. RATIO2.LE.0.0) ) GO TO 330  
IF ((RATIO1.GT.0.0 .AND. RATIO1.LT.1.0) .AND.  
\* (RATIO2.GT.0.0 .AND. RATIO2.LT.1.0) ) GO TO 310

C ONE POINT INSIDE AND ONE POINT OUTSIDE OF AB.  
IF (RATIO2.GT.0.0 .AND. RATIO2.LT.1.0) GO TO 280

EQ1EM = SQRT(EQ1E(1)\*\*2+EQ1E(2)\*\*2+EQ1E(3)\*\*2)  
PAX = (EPM/CCONE)\*(EQ1E(1)/EQ1EM)  
PAY = (EPM/CCONE)\*(EQ1E(2)/EQ1EM)  
GO TO 290

280 EQ2EM = SQRT(EQ2E(1)\*\*2+EQ2E(2)\*\*2+EQ2E(3)\*\*2)  
PAX = (EPM/CCONE)\*(EQ2E(1)/EQ2EM)  
PAY = (EPM/CCONE)\*(EQ2E(2)/EQ2EM)

290 IF (COSPA.LT.CCONE) GO TO 300

```
PBX = (EPM/COSPA)*(EAE(1)/EAEM)
PBY = (EPM/COSPA)*(EAE(2)/EAEM)
GO TO 320
300 PBX = (EPM/COSPB)*(EBE(1)/EBEM)
PBY = (EPM/COSPB)*(EBE(2)/EBEM)
GO TO 320
C
C BOTH POINTS INSIDE OF AB
310 EQ1EM = SQRT(EQ1E(1)**2+EQ1E(2)**2+EQ1E(3)**2)
EQ2EM = SQRT(EQ2E(1)**2+EQ2E(2)**2+EQ2E(3)**2)
PAX = (EPM/CCONE)*(EQ1E(1)/EQ1EM)
PAY = (EPM/CCONE)*(EQ1E(2)/EQ1EM)
PBX = (EPM/CCONE)*(EQ2E(1)/EQ2EM)
PBY = (EPM/CCONE)*(EQ2E(2)/EQ2EM)
C
C CONVERT TO 0.-TO-1. GRID VALUES, AND PLOT.
320 CONTINUE
IF (NVIEW.EQ.1) BIAS = 0.500
IF (NVIEW.EQ.2 .AND. LREYE.EQ.1) BIAS = 0.250
IF (NVIEW.EQ.2 .AND. LREYE.EQ.2) BIAS = 0.750
PAX = PAX*CONVRT+0.560
PAY = PAY*CONVRT+BIAS
PBX = PBX*CONVRT+0.560
PBY = PBY*CONVRT+BIAS
IPAX = PAX * FLOAT (1024)
IPAY = PAY * FLOAT (1024)
IPBX = PBX * FLOAT (1024)
IPBY = PBY * FLOAT (1024)
CALL LINEV (IPAY,IPAX,IPBY,IPBX)
IF (IFNUM.EQ.0 .OR. IFJNUM.EQ.0) GO TO 330
IF (NA.GT.100 .OR. NB.GT.100) GO TO 330
CALL PRINTV (3,NUMBER(NA),IPAY,IPAX)
CALL PRINTV (3,NUMBER(NB),IPBY,IPBX)
330 CONTINUE
C
C CLEAR PLOT BUFFER FOR THE FRAME JUST COMPLETED.
IF (IFFA.EQ.1) CALL FRAMEV (3)
C
RETURN
999 CALL ZZPOMB (6HPLOT3 ,NERROR)
END
```

SUBROUTINE PLOTSS (YMAXIN,YMININ,YTOP,YBOT)

C SELECT PLOT SCALE AND CALCULATE TOP,BOTTOM VALUES OF 10 SQUARE  
 C LINEAR PLOT GRID FROM YMAXIN,YMININ.  
 C CALLS FORMA SUBROUTINE ZZBOMB.  
 C CODED BY RF HRUDA. SEPTEMBER 1967.  
 C LAST REVISION BY WA BENFIELD. MARCH 1976.

C  
 C SUBROUTINE ARGUMENTS  
 C YMAXIN = INPUT MAXIMUM VALUE TO BE PLOTTED.  
 C YMININ = INPUT MINIMUM VALUE TO BE PLOTTED.  
 C YTOP = OUTPUT TOP LIMIT OF GRID.  
 C YBOT = OUTPUT BOTTOM LIMIT OF GRID.

C  
 C ERROR EXPLANATION  
 C 1 = YMAX IS LESS THAN YMIN.  
 C 2 = SCALE CANNOT BE CALCULATED.

YMAX = YMAXIN  
 YMIN = YMININ

NERROR = 1

IF (YMAX .LT. YMIN) GO TO 999  
 IF (YMAX .GT. YMIN) GO TO 21  
 11 IF (YMAX .LT. 0.00) GO TO 13  
 YMAX = 1.001\*YMAX  
 YMIN = .999\*YMIN  
 GO TO 15  
 13 YMAX = .999\*YMAX  
 YMIN = 1.001\*YMIN  
 15 IF (YMAX .NE. 0.) GO TO 21  
 YMAX = +.3  
 YMIN = -.3

C  
 21 VALUE = (YMAX-YMIN)/10.  
 IF (VALUE .LT. ABS(YMIN/100000.)) GO TO 11  
 DO 23 I=1,66  
 DO 23 J=1,3  
 SCALE = 2.\*\*(J-2) \*10.\*\*(I-33)  
 IF (SCALE .GE. VALUE) GO TO 31  
 23 CONTINUE

NERROR = 2

GO TO 999

C  
 31 NSTEPS = YMIN/SCALE  
 YBOT = FLOAT(NSTEPS)\*SCALE  
 32 IF (YMIN) 34,38,36  
 33 YBOT = YBOT-SCALE  
 34 IF (YBOT .LE. YMIN) GO TO 38  
 GO TO 33  
 35 YBOT = YBOT+SCALE  
 36 IF (YBOT-YMIN) 35,38,37  
 37 YBOT = YBOT-SCALE  
 38 YTOP = YBOT+10.\*SCALE  
 IF (YTOP .GE. YMAX) RETURN  
 IF (J .LT. 3) GO TO 39

```
J = 0  
I = I+1  
39 J = J+1  
SCALE = 2.**(J-2) *10.**(I-33)  
GO TO 32
```

```
C  
999 CALL ZZBOMB (6HPLOTSS, NERROR)  
END
```

PUNCAN

-----

SUBROUTINE PUNCAN(IA,NR,NC,ANAME,KR)  
 DIMENSION IA(KR,1)

C  
 C PRODUCES PUNCHED CARD OUTPUT USABLE FOR SUBROUTINE READAN.  
 C CODED BY JOHN ADMIRE \*NASA\* OCT 1974.

C  
 C SUBROUTINE ARGUMENTS (ALL INPUT)  
 C IA = MATRIX TO BE PUNCHED. SIZE (NR,NC).  
 C NR = NUMBER OF ROWS IN MATRIX A.  
 C NC = NUMBER OF COLS IN MATRIX A.  
 C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).  
 C KR = ROW DIMENSION OF A IN CALLING PROGRAM.

C  
 4010 FORMAT (A6,I4,I5)  
 4020 FORMAT (2I5,10A6)  
 4030 FORMAT (10H0000000000)

C  
 C PUNCH 4010, ANAME,NR,NC

C  
 DO 60 I=1,NR  
 JS = 1  
 10 JE = JS+9  
 IF (JE .GT. NC) JE=NC  
 C SEE IF ELEMENTS ARE ZERO.  
 DO 20 J=JS,JE  
 20 IF(IA(I,J) .NE. 6H ) GO TO 35  
 GO TO 40  
 35 PUNCH 4020, I,JS,(1A(I,J), J=JS,JE)  
 40 IF (JE .EQ. NC) GO TO 30  
 JS = JS+10  
 GO TO 10  
 60 CONTINUE

C  
 C PUNCH 4030  
 C RETURN  
 C END

PUNCH

-----

SUBROUTINE PUNCH (A,NR,NC,ANAME,KR)  
DIMENSION A(KR,1)

C  
C PRODUCES PUNCHED CARD OUTPUT USABLE FOR SUBROUTINE READ.  
C CODED BY RL WOHLER. DECEMBER 1966.

C  
C SUBROUTINE ARGUMENTS (ALL INPUT)  
C A = MATRIX TO BE PUNCHED. SIZE (NR,NC).  
C NR = NUMBER OF ROWS IN MATRIX A.  
C NC = NUMBER OF COLS IN MATRIX A.  
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).  
C KR = ROW DIMENSION OF A IN CALLING PROGRAM.

C  
4010 FORMAT (A6,I4,I5)  
4020 FORMAT (2I5,4E17.8)  
4030 FORMAT (10H0000000000)

C  
C PUNCH 4010, ANAME,NR,NC

C  
DO 60 I=1,NR  
JS = 1  
10 JE = JS+3  
IF (JE .GT. NC) JE=NC  
C SEE IF ELEMENTS ARE ZERO.  
DO 20 J=JS,JE  
20 IF (A(I,J) .NE. 0.) GO TO 35  
GO TO 40  
35 PUNCH 4020, I,JS,(A(I,J), J=JS,JE)  
40 IF (JE .EQ. NC) GO TO 60  
JS = JS+4  
GO TO 10  
60 CONTINUE

C  
C PUNCH 4030  
RETURN  
END

PUNCHO

-----

```

SUBROUTINE PUNCHO (A,NR,NC,ANAME,KR)
DIMENSION A(KR,1)
C
C PRODUCES PUNCHED CARD OUTPUT IN OCTAL, USABLE FOR SUBROUTINE READO.
C CODED BY CHRIS CHASE. MARCH 1969.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX TO BE PUNCHED. SIZE (NR,NC).
C NR = NUMBER OF ROWS IN MATRIX A.
C NC = NUMBER OF COLS IN MATRIX A.
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C KR = ROW DIMENSION OF A IN CALLING PROGRAM.
C
4010 FORMAT (A6,I4,I5)
4020 FORMAT (2I5,3(3X,0I2))
4030 FORMAT (10H0000000000)
C
C PUNCH 4010, ANAME,NR,NC
C
DO 60 I=1,NR
JS = 1
10 JE = JS+2
IF (JE .GT. NC) JE=NC
C SEE IF ELEMENTS ARE ZERO.
DO 20 J=JS,JE
20 IF (A(I,J) .NE. 0.) GO TO 35
GO TO 40
35 PUNCH 4020, I,JS,(A(I,J), J=JS,JE)
40 IF (JE .EQ. NC) GO TO 60
JS = JS+3
GO TO 10
60 CONTINUE
C
C PUNCH 4030
RETURN
END
```



PUNCIM

```

SUBROUTINE PUNCI(IA,NR,NC,ANAME,KR)
DIMENSION IA(KR,1)
C
C PRODUCES PUNCHED CARD OUTPUT USABLE FOR SUBROUTINE READIM.
C CODED BY JOHN ADMIRE *NASA* OCT 1974.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C IA = MATRIX TO BE PUNCHED. SIZE (NR,NC).
C NR = NUMBER OF ROWS IN MATRIX A.
C NC = NUMBER OF COLS IN MATRIX A.
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C KR = ROW DIMENSION OF A IN CALLING PROGRAM.
C
4010 FORMAT (A6,I4,I5)
4020 FORMAT (I6I5)
4030 FORMAT (1CH00000000C0)
C
PUNCH 4010, ANAME,NR,NC
C
DO 60 I=1,NR
JS = 1
10 JE = JS+13
IF (JE .GT. NC) JE=NC
C SEE IF ELEMENTS ARE ZERO.
DO 20 J=JS,JE
20 IF(IA(I,J) .NE. 0) GO TO 35
GO TO 40
35 PUNCH 4020, I,JS,(IA(I,J), J=JS,JE)
40 IF (JE .EQ. NC) GO TO 60
JS = JS+14
GO TO 10
60 CONTINUE
C
PUNCH 4030
RETURN
END
```

```

SUBROUTINE RBTG1 (XYZ,XYZREF,JDOF,JVEC,RBT,NNODES,NRRBT,NCRBT,
*                KXJ,KR)
* DIMENSION XYZ(KXJ,1),XYZREF(1),JDOF(KXJ,1),JVEC(1),RBT(KR,1),
*                IVEC(6),W(6,6)

```

```

C
C GENERATES A RIGID-BODY-TRANSFORMATION IN CARTESIAN COORDINATES.
C CALLS FORMA SUBROUTINES REVADD,ZZBOMB.
C DEVELOPED BY RF HRUDA. APRIL 1969
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C XYZ = INPUT MATRIX OF X,Y,Z COORDINATE LOCATIONS FOR EACH NODE
C POINT. SIZE(NNODES,3).
C XYZREF = INPUT VECTOR OF X,Y,Z COORDINATE LOCATIONS FOR THE
C REFERENCE POINT. SIZE(3).
C JDOF = INPUT MATRIX. EACH ROW IS USED AS AN IVEC TO REVADD X,Y,Z,
C TX,TY,TZ NODE DEGREES OF FREEDOM INTO ROWS OF RBT. EACH
C OF THESE DEGREES OF FREEDOM ARE ASSUMED TO BE IN THE
C SAME DIRECTION AS ITS CORRESPONDING REFERENCE DEGREE
C OF FREEDOM. A NEGATIVE VALUE IN JDOF CAUSES THE CORRES-
C PONDING ROW OF RBT TO BE ZERO. SIZE (NNODES,6).
C JVEC = INPUT VECTOR. USED AS A JVEC TO REVADD X,Y,Z, TX, TY, TZ REFERENCE
C DEGREES OF FREEDOM INTO COLUMNS OF RBT. NEGATIVE SIGNS
C ENABLES CHANGE FROM ASSUMED RIGHT HAND SYSTEM TO ONE YOU
C WISH TO SPECIFY. SIZE (6).
C RBT = OUTPUT RIGID BODY TRANSFORMATION MATRIX. SIZE(NRRBT,NCRBT).
C NNODES = INPUT NUMBER OF NODES. ROW SIZE OF MATRICES XYZ,JDOF.
C NRRBT = OUTPUT NUMBER OF ROWS IN RBT. EQUAL TO NON-ZEROS IN JDOF.
C NCRBT = OUTPUT NUMBER OF COLS IN RBT. EQUAL TO NON-ZEROS IN JVEC.
C KXJ = INPUT ROW DIMENSION OF XYZ,JDOF IN THE CALLING PROGRAM.
C KR = INPUT ROW DIMENSION OF RBT IN THE CALLING PROGRAM.
C
C ERROR EXPLANATION
C I = NUMBER OF NON-ZEROS IN MATRIX IMAT EXCEEDS
C ROW DIMENSION OF MATRIX RBT.
C
C NRRBT = 0
C NCRBT = 0
C DO 10 J=1,6
10 IF (JVEC(J).NE.0) NCRBT = NCRBT+1
C DO 20 I=1,NNODES
C DO 20 J=1,6
20 IF (JDOF(I,J).NE.0) NRRBT = NRRBT+1
C
C IF (NRRBT.GT.KR) GO TO 999
C
C 30 DO 40 I=1,NRRBT
C DO 40 J=1,NCRBT
40 RBT(I,J) = 0.0
C DO 60 I=1,6
C DO 50 J=1,6
50 W(I,J) = 0.0
60 W(I,I) = 1.0
C DO 80 I=1,NNODES
W(I,5) = (XYZ(I,3)-XYZREF(3))

```

ERROR=1

```
W(1,6) = -(XYZ(I,2)-XYZREF(2))
W(2,4) = -(XYZ(I,3)-XYZREF(3))
W(2,6) = (XYZ(I,1)-XYZREF(1))
W(3,4) = (XYZ(I,2)-XYZREF(2))
W(3,5) = -(XYZ(I,1)-XYZREF(1))
```

```
DC 70 J=1,6
```

```
IVEC(J)=JDCF(I,J)
```

```
IF (IVEC(J).LT.0) IVEC(J)=0
```

```
IF (JVEC(J).LT.0) IVEC(J) = -IVEC(J)
```

```
70 CONTINUE
```

```
80 CALL REVADD (I.,W,IVEC,JVEC,RBT,6,6,NRRBT,NCRBT,6,KR)
```

```
RETURN
```

C

```
999 CALL ZZBOMB (6HRBTG1 ,NERROR)
```

```
END
```

```

SUBROUTINE RBTG2 (XRT,XYZREF,JDOF,JVEC,RBT,NNODES,NRRBT,NCRBT,
*           KXJ,KR)
DIMENSION XRT(KXJ,1),XYZREF(1),JDOF(KXJ,1),JVEC(1),RBT(KR,1),
*           IVEC(6),W(6,6)

```

```

C
C GENERATES A RIGID-BODY-TRANSFORMATION FROM
C CYLINDRICAL TO CARTESIAN COORDINATES
C CALLS FORMA SUBROUTINES REVADD,ZZBOMB.
C DEVELOPED BY RF HRUDA. JAN 1970.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C XRT = INPUT MATRIX OF X,R,THETA(DEGREES) COORDINATE LOCATIONS
C FOR EACH NODE POINT. SIZE(NNODES,3).
C XYZREF = INPUT VECTOR OF X,Y,Z COORDINATE LOCATIONS FOR THE
C REFERENCE POINT. SIZE(3).
C JDOF = INPUT MATRIX. EACH ROW IS USED AS AN IVEC TO REVADD X,R,T,
C TX,TR,TT NODE DEGREES OF FREEDOM INTO ROWS OF RBT. EACH
C OF THESE DEGREES OF FREEDOM ARE ASSUMED TO BE IN THE
C SAME DIRECTION AS ITS CORRESPONDING REFERENCE DEGREE
C OF FREEDOM. A NEGATIVE VALUE IN JDOF CAUSES THE CORRES-
C PONDING ROW OF RBT TO BE ZERO. SIZE (NNODES,6).
C JVEC = INPUT VECTOR. USED AS A JVEC TO REVADD X,Y,Z,TX,TY,TZ REFERENCE
C DEGREES OF FREEDOM INTO COLUMNS OF RBT. NEGATIVE SIGNS
C ENABLES CHANGE FROM ASSUMED RIGHT HAND SYSTEM TO ONE YOU
C WISH TO SPECIFY. SIZE (6).
C RBT = OUTPUT RIGID BODY TRANSFORMATION MATRIX. SIZE(NRRBT,NCRBT).
C NNODES = INPUT NUMBER OF NODES. ROW SIZE OF MATRICES XRT,JDOF.
C NRRBT = OUTPUT NUMBER OF ROWS IN RBT. EQUAL TO NON-ZEROS IN JDOF.
C NCRBT = OUTPUT NUMBER OF COLS IN RBT. EQUAL TO NON-ZEROS IN JVEC.
C KXJ = INPUT ROW DIMENSION OF XRT,JDOF IN THE CALLING PROGRAM.
C KR = INPUT ROW DIMENSION OF RBT IN THE CALLING PROGRAM.
C
C NERROR EXPLANATION
C 1 = NUMBER OF NON-ZEROS IN MATRIX IMAT EXCEEDS
C ROW DIMENSION OF MATRIX RBT.
C
C NRRBT = 0
C NCRBT = 0
C DO 10 J=1,6
10 IF (JVEC(J).NE.0) NCRBT = NCRBT+1
C DO 20 I=1,NNODES
C DO 20 J=1,6
20 IF (JDOF(I,J).NE.0) NRRBT = NRRBT+1
C
C IF (NRRBT.GT.KR) GO TO 999
C
C DO 40 I=1,NRRBT
C DO 40 J=1,NCRBT
40 RBT(I,J) = 0.0
C DO 50 I=1,6
C DO 50 J=1,6
50 W(I,J) = 0.0
C
C RPD = 3.1415926535898 / 180.

```

NERROR=1

YD = XYZREF(2)  
ZD = XYZREF(2)

DO 80 I=1,NNODES  
XD = (XRT(I,1) - XYZREF(1))  
RI = XRT(I,2)  
SI = SIN(XRT(I,3)\*RPD)  
CI = COS(XRT(I,3)\*RPD)

W(1,1) = 1.  
W(1,5) = (SI\*RI) -ZD  
W(1,6) = -(CI\*RI) +YD  
W(2,2) = CI  
W(2,3) = SI  
W(2,4) = -(SI\*YD) +(CI\*ZD)  
W(2,5) = -(SI\*XD)  
W(2,6) = CI\*XD  
W(3,2) = -SI  
W(3,3) = CI  
W(3,4) = -(CI\*YD) -(SI\*ZD) +RI  
W(3,5) = -(CI\*XD)  
W(3,6) = -(SI\*XD)  
W(4,4) = 1.  
W(5,5) = CI  
W(5,6) = SI  
W(6,5) = -SI  
W(6,6) = CI

DO 70 J=1,6  
IVEC(J)=JDOF(I,J)  
IF (IVEC(J).LT.0) IVEC(J)=0  
IF (JVEC(J).LT.0) IVEC(J) = -IVEC(J)  
70 CONTINUE  
80 CALL REVADD (1.,W,IVEC,JVEC,RBT,6,6,NRRBT,NCRBT,6,KR)  
RETURN

999 CALL ZZBOMB (6HRBTG2 ,NERROR)  
END

SUBROUTINE READ (A, NR, NC, KR, KC)  
 DIMENSION A(KR, 1), X(4), IREMRK(9)  
 COMMON / LLINE/ NLINE, MAXLIN, MINI  
 DATA NIT, NCT/5, 6/

C  
 C READ MATRIX OF REAL NUMBERS FROM CARDS OR TAPE AND PRINT IT. WRITE  
 C MATRIX ON TAPE IF SO INDICATED (BY HAVING THE WRITE-TAPE NUMBER IN  
 C COLUMNS 79-80).  
 C THE EXPLANATION OF FORMATS USED BELOW IS ...  
 C A - DENOTES ANY KEY PUNCH SYMBOL. (EG, A1/\*C).  
 C I - DENOTES AN INTEGER NUMBER. (EG, 436).  
 C E - DENOTES A REAL NUMBER. (EG, 24.963).  
 C \*\*\*\* CARD INPUT \*\*\*\*  
 C FIRST CARD - MATRIX NAME, NUMBER OF ROWS, NUMBER OF COLUMNS  
 C WITH A6, I4, I5 FORMAT.  
 C - REMARKS IN COLUMNS 16-69. A-TYPE FORMAT.  
 C - \$ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.  
 C - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR  
 C THE WORDS REWIND OR LIST, OR (WHEN \$ IN COLUMN 72)  
 C THE WRITE-TAPE-ID (EG, T1234).  
 C - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).  
 C MIDDLE CARDS - DATA WITH FORMAT (2I5, 4E17).  
 C - 1-ST I5 IS THE ROW NUMBER.  
 C - 2-ND I5 IS THE COL NUMBER OF THE NEXT E17 FIELD.  
 C - NEXT 4E17 ARE ELEMENTS OF THE MATRIX.  
 C LAST CARD - TEN ZEROS IN COLUMNS 1-10.  
 C \*\*\*\* TAPE INPUT \*\*\*\*  
 C ONE CARD - MATRIX NAME, ZERO OR MINUS THE LOCATION NUMBER OF MATRIX  
 C ON READ-TAPE, READ-TAPE NUMBER (IF MINUS, NO PRINTOUT),  
 C MATRIX RUN NUMBER WITH A6, I4, I5, A6 FORMAT.  
 C - READ-TAPE CONTROL IN COLUMNS 22-27. MAY BE BLANK, OR THE  
 C WORDS REWIND OR LIST.  
 C - REMARKS IN COLUMNS 28-69. A-TYPE FORMAT.  
 C - \$ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.  
 C - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR  
 C THE WORDS REWIND OR LIST, OR (WHEN \$ IN COLUMN 72)  
 C THE WRITE-TAPE-ID (EG, T1234).  
 C - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).  
 C CALLS FOR MA SUPROUTINES INTAPE, LTAPE, PAGEHD, RTAPE, WRITE, WTAPE, ZZBOMB.  
 C CODED BY RF HRUDA. JULY 1968.  
 C MODIFIED FOR CONTRACT NAS8-25922, OCTOBER 1970.  
 C MODIFIED BY JOHN ADMIR \*NASA\* SEPT 1973  
 C LAST REVISION BY RL WOHLN. APRIL 1976.  
 C  
 C SUBROUTINE ARGUMENTS  
 C A = OUTPUT MATRIX READ FROM CARDS OR TAPE.  
 C NR = OUTPUT NUMBER OF ROWS IN MATRIX A.  
 C NC = OUTPUT NUMBER OF COLS IN MATRIX A.  
 C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.  
 C KC = INPUT COL DIMENSION OF A IN CALLING PROGRAM.  
 C  
 C ERROR EXPLANATION  
 C 1 = ROW SIZE EXCEEDS ROW DIMENSION OR  
 C COLUMN SIZE EXCEEDS COLUMN DIMENSION.  
 C 2 = ROW OR COLUMN VALUE OF ELEMENT EXCEEDS MATRIX SIZE.

3 = DATA ON CARD PAST MATRIX COLUMN SIZE.  
 4 = LOCATION ON TAPE PAST END-OF-TAPE MARK.  
 5 = 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 (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )  
 \* 2X 9A6,2X A1,A6,I4//)  
 2002 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )  
 \* 3X 9HCONTINUED //)  
 2003 FORMAT (// 1XA6,I4,I5,5X 9A6,2X A1,A6,I4)  
 2004 FORMAT (1X 2I5,1P4E17.8)  
 2005 FORMAT (13HOEND OF READ.)  
 2006 FORMAT (25HOSIZE OF MATRIX READ IS (I4,2H X I4,2H ) )  
 2007 FORMAT(/,1X,123(1H-))

C  
 C READ IN HEADER CARD.  
 READ (NIT,1001) ANAME,N1,N2,IEMRK,IZ1,IZ2,NWTAPE  
 NR = N1  
 NC = N2

C  
 IF(N1 .GT. 0) GO TO 50  
 IF(MINI .NE. 4HMINI) GO TO 40  
 IF(NLINE .LE. 5) GO TO 40  
 IF(NLINE+9 .GT. MAXLIN) GO TO 40  
 WRITE(NOT,2007)  
 NLINE=NLINE+2  
 GO TO 200  
 40 CALL PAGEHD  
 GO TO 200

C  
 C CARD READING SECTION.  
 50 IF(MINI .NE. 4HMINI) GO TO 60  
 IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 60  
 NBC=NC/ 4  
 IF( 4\*NBC .NE. NC) NBC=NBC+1  
 NN=9+NR\*NBC  
 IF(NN +NLINE .GT. MAXLIN) GO TO 60  
 WRITE(NOT,2007)  
 NLINE=NLINE+2  
 GO TO 70  
 60 CALL PAGEHD  
 70 CONTINUE  
 WRITE (NOT,2001) ANAME,NR,NC,IEMRK,IZ1,IZ2,NWTAPE  
 NLINE=NLINE+5

NERROR = 1

IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999  
 DO 105 I=1,NR  
 DO 105 J=1,NC  
 105 A(I,J) = 0.  
 110 READ (NIT,1002) I,JS,X  
 IF (I.EQ.0 .AND. JS.EQ.0) GO TO 300

NERROR = 2

IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998  
 JE = JS+3

READ -- 3/ 4  
-----

IF (JE.LE.NC) GO TO 115  
JX = NC-JS+2

NERROR = 3

DO 112 J=JX,4  
112 IF(ABS(X(J)) .GT. 0.) GO TO 998  
JF = NC  
115 N = 0  
DO 120 J=JS,JE  
N = N+1  
120 A(I,J) = X(N)  
IF(NLINE+1 .LE. MAXLIN) GO TO 125  
CALL PAGEHD  
WRITE (NOT,2002) ANAME,NR,NC  
NLINE=NLINE+5  
125 WRITE (NOT,2004) I,JS,(A(I,J),J=JS,JE)  
NLINE=NLINE+1  
GO TO 110

C

C TAPE READING SECTION.

200 WRITE (NOT,2003) ANAME,N1,N2,IEMRK,IZ1,IZ2,NWTAPE  
NLINE=NLINE+3  
NRTAPE = IABS(N2)  
IF (IEMRK(2) .EQ. 6HREWIND) REWIND NRTAPE  
IF (IEMRK(2) .EQ. 4HLIST) CALL LTAPE (NRTAPE)  
IF (N1.EQ.0) GO TO 250

C POSITION NRTAPE.

READ (NRTAPE) TID,LN,IEOTCK  
NUM = LN+N1  
IF (NUM) 205,220,225

205 IF (IEOTCK .EQ. 3HEOT) GO TO 997  
READ (NRTAPE) DUM  
NUM = -NUM-1  
IF(NUM .EQ. 0) GO TO 240  
DO 210 L=1,MIM  
READ (NRTAPE) TID,LN,IEOTCK

NERROR = 4

IF (IEOTCK .EQ. 3HEOT) GO TO 997

210 READ (NRTAPE) DUM  
GO TO 240

220 BACKSPACE NRTAPE  
GO TO 240

225 REWIND NRTAPE  
NUM = (-N1-1)\*2  
IF (NUM .EQ. 0) GO TO 240  
DO 230 L=1,NUM

230 READ (NRTAPE) DUM

240 IF(IPEMRK(1) .NE. 6H ) GO TO 250  
READ(NRTAPE) TID,LN,DUM,IEMRK(1),ANAM  
NERROR=6

IF(LN+N1 .NE. 0) GO TO 999  
NERROR=7

IF(ANAM .NE. ANAME) GO TO 999  
BACKSPACE NRTAPE

250 CALL RTAPE (IEMRK(1),ANAME,A,NR,NC,KR,KC,NRTAPE)

NERROR = 5



```

WRITE (NOT,2006) NR,NC
NLINE=NLINE+2
IF (N2 .GT. 0) CALL WRITE (A,NR,NC,ANAME,KR)

```

C

C TAPE WRITING SECTION.

```

300 IF (NWTAPE.LE.0) GO TO 400
IF (IZ1 .EQ. 1H$) CALL INTAPE (NWTAPE,IZ2)
IF (IZ2 .EQ. 6HREWIND) REWIND NWTAPE
CALL WTAPE (A,NR,NC,ANAME,KR,NWTAPE)
IF (IZ2 .EQ. 4HLIST) CALL LTAPE (NWTAPE)

```

C

```

400 WRITE (NOT,2005)
NLINE=NLINE+2
RETURN

```

C

```

997 CALL LTAPE (NRTAPE)
GO TO 999
998 WRITE (NOT,2004) I,JS,X
999 CALL ZZBOMB (6HREAD ,NERROR)
END

```

```

SUBROUTINE READAN (IA,NR,NC,KR,KC)
DIMENSION IA(KR,1),IX(10),IREMRK(9)
COMMON / LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NCT/5,6/

```

```

C
C READ MATRIX OF ALPHA-NUMERIC CHARACTERS (A6) FROM CARDS OR TAPE AND
C PRINT IT. WRITE MATRIX ON TAPE IF SO INDICATED (BY HAVING THE
C WRITE-TAPE NUMBER IN COLUMNS 79-80).
C THE EXPLANATION OF FORMATS USED BELOW IS ...
C   A - DENOTES ANY KEY PUNCH SYMBOL. (EG, A1/*C).
C   I - DENOTES AN INTEGER NUMBER. (EG, 436).
C **** CARD INPUT ****
C FIRST CARD   - MATRIX NAME, NUMBER OF ROWS, NUMBER OF COLUMNS
C               WITH A6,I4,I5 FORMAT.
C               - REMARKS IN COLUMNS 16-69. A-TYPE FORMAT.
C               - $ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.
C               - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR
C               THE WORDS REWIND OR LIST, OR (WHEN $ IN COLUMN 72)
C               THE WRITE-TAPE-ID (EG, T1234).
C               - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).
C MIDDLE CARDS - DATA WITH FORMAT (2I5, 10A6).
C               - 1-ST I5 IS THE ROW NUMBER.
C               - 2-ND I5 IS THE COL NUMBER OF THE NEXT I5 FIELD.
C               - NEXT 10A6 ARE ELEMENTS OF THE MATRIX.
C LAST CARD    - TEN ZEROS IN COLUMNS 1-10.
C **** TAPE INPUT ****
C ONE CARD    - MATRIX NAME, ZERO OR MINUS THE LOCATION NUMBER OF MATRIX
C               ON READ-TAPE, READ-TAPE NUMBER (IF MINUS, NO PRINTOUT),
C               MATRIX RUN NUMBER WITH A6,I4,I5,A6 FORMAT.
C               - READ-TAPE CONTROL IN COLUMNS 22-27. MAY BE BLANK, OR THE
C               WORDS REWIND OR LIST.
C               - REMARKS IN COLUMNS 28-69. A-TYPE FORMAT.
C               - $ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.
C               - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR
C               THE WORDS REWIND OR LIST, OR (WHEN $ IN COLUMN 72)
C               THE WRITE-TAPE-ID (EG, T1234).
C               - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).
C CALLS FORMA SUBROUTINES INTAPE,LTAPE,PAGEHD,RTAPE,WRITAN,WTAPE,ZZBOMB.
C CODED BY JOHN ADMIRE *NASA* OCT 1974.
C LAST REVISION BY RL WOHLN. APRIL 1976.

```

SUBROUTINE ARGUMENTS

```

C IA  = OUTPUT MATRIX READ FROM CARDS OR TAPE.
C NR  = OUTPUT NUMBER OF ROWS IN MATRIX IA.
C NC  = OUTPUT NUMBER OF COLS IN MATRIX IA.
C KC  = INPUT COL DIMENSION OF IA IN CALLING PROGRAM.
C KR  = INPUT ROW DIMENSION OF IA IN CALLING PROGRAM.

```

ERROR EXPLANATION

```

C 1 = ROW SIZE EXCEEDS ROW DIMENSION OR
C     COLUMN SIZE EXCEEDS COLUMN DIMENSION.
C 2 = ROW OR COLUMN VALUE OF ELEMENT EXCEEDS MATRIX SIZE.
C 3 = DATA ON CARD PAST MATRIX COLUMN SIZE.
C 4 = LOCATION ON TAPE PAST END-OF-TAPE MARK.
C 5 = LOCATION ON TAPE PAST END-OF-TAPE MARK.

```

```

1001 FORMAT (A6,I4,I5,9A6, 2XA1,A6,I2)
1002 FORMAT (2I5,10A6)
2001 FORMAT (//33H CARD INPUT ALPHA-NUMERIC MATRIX A6,
*
*           2X 1H( I4,2H X I4,2H )
*           2X 9A6,2X A1,A6,I4//)
2002 FORMAT (//33H CARD INPUT ALPHA-NUMERIC MATRIX A6,
*
*           2X 1H( I4,2H X I4,2H )
*           3X 9HCONTINUED //)
2003 FORMAT (// 1XA6,I4,I5,5X 9A6,2X A1,A6,I4)
2004 FORMAT (1X 2I5,10A6)
2005 FORMAT (15HOEND OF READAN.)
2006 FORMAT (25HOSIZE OF MATRIX READ IS (I4,2H X I4,2H ) )
2007 FORMAT(/,1X,123(1H-))

```

```

C
C READ IN HEADER CARD.
  READ (NIT,1001) ANAME,N1,N2,IREMRK,IZ1,IZ2,NWTAPE
  NR = N1
  NC = N2

```

```

C
  IF(N1 .GT. 0) GO TO 50
  IF(MINI .NE. 4HMINI) GO TO 40
  IF(NLINE .LE. 5) GO TO 40
  IF(NLINE+9 .GT. MAXLIN) GO TO 40
  WRITE(NOT,2007)
  NLINE=NLINE+2
  GO TO 200
40 CALL PAGEHD
  GO TO 200
50 IF(MINI .NE. 4HMINI) GO TO 60
  IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 60
  NBC=NC/10
  IF(10*NBC .NE. NC) NBC=NBC+1
  NN=9+NR*NBC
  IF(NN +NLINE .GT. MAXLIN) GO TO 60
  WRITE(NOT,2007)
  NLINE=NLINE+2
  GO TO 70
60 CALL PAGEHD
70 CONTINUE

```

```

C
C CARD READING SECTION.
  WRITE (NOT,2001) ANAME,NR,NC,IREMRK,IZ1,IZ2,NWTAPE
  NLINE=NLINE+5

```

NERROR = 1

```

  IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999
  DO 105 I=1,NR
  DO 105 J=1,NC
105 IA(I,J) = 6H
110 READ (NIT,1002) I,JS,IX
  IF (I.EQ.0 .AND. JS.EQ.0) GO TO 300

```

NERROR = 2

```

  IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998
  JE = JS+9
  IF (JF.LE.NC) GO TO 115
  JX = NC-JS+2

```

-----  
NERROR = 3

```

      DO 112 J=JX,10
112 IF (IX(J) .NE. 6H      ) GO TO 998
      JE = NC
115 N = 0
      DO 120 J=JS,JE
      N = N+1
120 IA(I,J) = IX(N)
      IF(NLINE+1 .LE. MAXLIN) GO TO 125
      CALL PAGEHD
      WRITE (NOT,2002) ANAME,NR,NC
      NLINE=NLINE+5
125 WRITE (NOT,2004) I,JS,(IA(I,J),J=JS,JE)
      NLINE=NLINE+1
      GO TO 110

```

C

C TAPE READING SECTION.

```

200 WRITE (NOT,2003) ANAME,N1,N2,IEMRK,IZ1,IZ2,NWTAPE
      NLINE=NLINE+3
      NRTAPE = IABS(N2)
      IF (IEMRK(2) .EQ. 6HREWIND) REWIND NRTAPE
      IF (IEMPK(2) .EQ. 4HLIST) CALL LTAPE (NRTAPE)
      IF (N1.EQ.0) GO TO 250

```

C POSITION NRTAPE.

```

      READ (NRTAPE) TID,LN,IEOTCK
      NUM = LN+N1
      IF (NUM) 205,220,225

```

205

```

      IF (IEOTCK .EQ. 3HEOT) GO TO 997
      READ (NRTAPE) DUM
      NUM = -NUM-1
      IF(NUM .EQ. 0) GO TO 240
      DO 210 L=1,NUM
      READ (NRTAPE) TID,LN,IEOTCK

```

NERROR = 4

```

      IF (IEOTCK .EQ. 3HEOT) GO TO 997

```

```

210 READ (NRTAPE) DUM
      GO TO 240

```

```

220 BACKSPACE NRTAPE
      GO TO 240

```

```

225 REWIND NRTAPE
      NUM = (-N1-1)*2
      IF (NUM .EQ. 0) GO TO 240
      DO 230 L=1,NUM

```

```

230 READ (NRTAPE) DUM

```

```

240 IF (IEMRK(1) .NE. 6H      ) GO TO 250
      READ(NRTAPE) TID,LN,DUM,IEMRK(1),ANAM
      NERROR=6

```

```

      IF(LN+N1 .NE. 0) GO TO 999
      NERROR=7

```

```

      IF(ANAM .NE. ANAME) GO TO 999
      BACKSPACE NRTAPE

```

```

250 CALL PTAPE (IEMRK(1),ANAME,IA,NR,NC,KR,KC,NRTAPE)
      WRITE (NOT,2006) NR,NC
      NLINE=NLINE+2

```

NERROR = 5

IF (N2.GT.0) CALL WRITAN (IA, NR, NC, ANAME, KR)

C TAPE WRITING SECTION.

300 IF (NWTAPE.LE.0) GO TO 400  
IF (IZ1 .EQ. 1H\$) CALL INTAPE (NWTAPE, IZ2)  
IF (IZ2 .EQ. 6HREWIND) REWIND NWTAPE  
CALL WTAPE (IA, NR, NC, ANAME, KR, NWTAPE)  
IF (IZ2 .EQ. 4HLIST) CALL LTAPE (NWTAPE)

C  
400 WRITE (NGT, 2005)  
NLINE=NLINE+2  
RETURN

C  
997 CALL LTAPE (NRTAPE)  
GO TO 999  
998 WRITE (NGT, 2004) I, JS, IX  
999 CALL ZZBOMB (6HREADAN, NERROR)  
END

```

SUBROUTINE READIM (IA,NR,NC,KR,KC)
DIMENSION IA(KP,1),IX(14),IREMRK(9)
COMMON / LLINE/ NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C READ MATRIX OF INTEGER NUMBERS FROM CARDS OR TAPE AND PRINT IT. WRITE
C MATRIX ON TAPE IF SO INDICATED (BY HAVING THE WRITE-TAPE NUMBER IN
C COLUMNS 79-80).
C THE EXPLANATION OF FORMATS USED BELOW IS ...
C   A - DENOTES ANY KEY PUNCH SYMBOL. (EG, A1/*C).
C   I - DENOTES AN INTEGER NUMBER. (EG, 436).
C **** CARD INPUT ****
C FIRST CARD   - MATRIX NAME, NUMBER OF ROWS, NUMBER OF COLUMNS
C               WITH A6,I4,I5 FORMAT.
C               - REMARKS IN COLUMNS 16-69. A-TYPE FORMAT.
C               - $ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.
C               - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR
C               THE WORDS REWIND OR LIST, OR (WHEN $ IN COLUMN 72)
C               THE WRITE-TAPE-ID (EG, T1234).
C               - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).
C MIDDLE CARDS - DATA WITH FORMAT (2I5, 14I5).
C               - 1-ST I5 IS THE ROW NUMBER.
C               - 2-ND I5 IS THE COL NUMBER OF THE NEXT 15 FIELD.
C               - NEXT 14I5 ARE ELEMENTS OF THE MATRIX.
C LAST CARD    - TEN ZEROS IN COLUMNS 1-10.
C **** TAPE INPUT ****
C ONE CARD    - MATRIX NAME, ZERO OR MINUS THE LOCATION NUMBER OF MATRIX
C              ON READ-TAPE, READ-TAPE NUMBER (IF MINUS, NO PRINTOUT),
C              MATRIX RUN NUMBER WITH A6,I4,I5,A6 FORMAT.
C              - READ-TAPE CONTROL IN COLUMNS 22-27. MAY BE BLANK, OR THE
C              WORDS REWIND OR LIST.
C              - REMARKS IN COLUMNS 28-69. A-TYPE FORMAT.
C              - $ IN COLUMN 72 FOR WRITE-TAPE INITIALIZATION.
C              - WRITE-TAPE CONTROL IN COLUMNS 73-78. MAY BE BLANK, OR
C              THE WORDS REWIND OR LIST, OR (WHEN $ IN COLUMN 72)
C              THE WRITE-TAPE-ID (EG, T1234).
C              - WRITE-TAPE NUMBER IN COLUMNS 79-80. (EG, 21).
C CALLS FORMA SUBROUTINES INTAPE,LTAPE,PAGEHD,RTAPE,WRITIM,WTAPE,ZZBOMB.
C CODED BY RF HRUDA. JULY 1968.
C MODIFIED FOR CONTRACT NAS8-25922, OCTOBER 1970.
C MODIFIED BY JOHN ADMIRE *NASA* SEPT 1973
C LAST REVISION BY RL WOHLER. APRIL 1976.
C
C SUBROUTINE ARGUMENTS
C IA  = OUTPUT MATRIX READ FROM CARDS OR TAPE.
C NR  = OUTPUT NUMBER OF ROWS IN MATRIX IA.
C NC  = OUTPUT NUMBER OF COLS IN MATRIX IA.
C KR  = INPUT  ROW DIMENSION OF +A IN CALLING PROGRAM.
C KC  = INPUT  COL DIMENSION OF IA IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = ROW SIZE EXCEEDS ROW DIMENSION OR
C   COLUMN SIZE EXCEEDS COLUMN DIMENSION.
C 2 = ROW OR COLUMN VALUE OF ELEMENT EXCEEDS MATRIX SIZE.
C 3 = DATA ON CARD PAST MATRIX COLUMN SIZE.
C 4 = LOCATION ON TAPE PAST END-OF-TAPE MARK.

```

5 = LOCATION ON TAPE PAST END-OF-TAPE MARK.

```

C
1001 FORMAT (A6,I4,I5,9A6, 2XA1,A6,I2)
1002 FORMAT (16I5)
2001 FORMAT (//27H CARD INPUT INTEGER MATRIX A6, 2X 1H( I4,2H X I4,2H )
*          2X 9A6,2X A1,A6,I4//)
2002 FORMAT (//27H CARD INPUT INTEGER MATRIX A6, 2X 1H( I4,2H X I4,2H )
*          3X 9HCONTINUED //)
2003 FORMAT (// 1XA6,I4,I5,5X 9A6,2X A1,A6,I4)
2004 FORMAT (1X 16I5)
2005 FORMAT (15HOEND OF READIM.)
2006 FORMAT (25HOSIZE OF MATRIX READ IS (I4,2H X I4,2H ) )
2007 FORMAT(/,1X,123(1H-))

```

```

C
C READ IN HEADER CARD.
  READ (NIT,1001) ANAME,N1,N2,IEMRK,IZ1,IZ2,NWTAPE
  NR = N1
  NC = N2

```

```

C
  IF(N1 .GT. 0) GO TO 50
  IF(MINI .NE. 4HMINI) GO TO 40
  IF(NLINE .LE. 5) GO TO 40
  IF(NLINE+9 .GT. MAXLIN) GO TO 40
  WRITE(NOT,2007)
  NLINE=NLINE+2
  GO TO 200
40 CALL PAGEHD
  GO TO 200
50 IF(MINI .NE. 4HMINI) GO TO 60
  IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 60
  NBC=NC/14
  IF(14*NBC .NE. NC) NBC=NBC+1
  NN=9+NR*NBC
  IF(NN +NLINE .GT. MAXLIN) GO TO 60
  WRITE(NOT,2007)
  NLINE=NLINE+2
  GO TO 70
60 CALL PAGEHD
70 CONTINUE

```

```

C
C CARD READING SECTION.
  WRITE (NOT,2001) ANAME,NR,NC,IEMRK,IZ1,IZ2,NWTAPE
  NLINE=NLINE+5

```

NERKOR = 1

```

  IF (NR.GT.KR .OR. NC.GT.KC) GO TO 99
  DO 105 I=1,NR
  DO 105 J=1,NC
105 IA(I,J) = 0
110 READ (NIT,1002) I,JS,IX
  IF (I.EQ.0 .AND. JS.EQ.0) GO TO 300

```

NERKOR = 2

```

  IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998
  JE = JS+13
  IF (JE.LE.NC) GO TO 115

```

JX = NC-JS+2

NERROR = 3

```

DO 112 J=JX,14
112 IF (IX(J) .NE. 0) GO TO 998
    JE = NC
115 N = 0
    DO 120 J=JS,JE
        N = N+1
120 IA(I,J) = IX(N)
    IF(NLINE+1 .LE. MAXLIN) GO TO 125
    CALL PAGEHD
    WRITE (NOT,2002) ANAME,NR,NC
    NLINE=NLINE+5
125 WRITE (NOT,2004) I,JS,(IA(I,J),J=JS,JE)
    NLINE=NLINE+1
    GO TO 110

```

C

C TAPE READING SECTION.

```

200 WRITE (NOT,2003) ANAME,N1,N2,IEMRK,IZ1,IZ2,NWTAPE
    NLINE=NLINE+3
    NRTAPE = IABS(N2)
    IF (IEMRK(2) .EQ. 6HREWIND) REWIND NRTAPE
    IF (IEMRK(2) .EQ. 4HLIST) CALL LTAPE (NRTAPE)
    IF (N1.EQ.0) GO TO 250

```

C POSITION NRTAPE.

```

    READ (NRTAPE) TID,LN,IEOTCK
    NUM = LN+N1
    IF (NUM) 205,220,225

```

```

205 IF (IEOTCK .EQ. 3HEGT) GO TO 997
    READ (NRTAPE) DUM
    NUM = -NUM-1
    IF(NUM .EQ. 0) GO TO 240
    DO 210 L=1,NUM
    READ (NRTAPE) TID,LN,IEOTCK

```

NERROR = 4

```

    IF (IEOTCK .EQ. 3HEOT) GO TO 997

```

```

210 READ (NRTAPE) DUM
    GO TO 240

```

```

220 BACKSPACE NRTAPE
    GO TO 240

```

```

225 REWIND NRTAPE
    NUM = (-N1-1)*2
    IF (NUM .EQ. 0) GO TO 240
    DO 230 L=1,NUM

```

```

230 READ (NRTAPE) DUM

```

```

240 IF(IEMRK(1) .NE. 6H      ) GO TO 250
    READ(NRTAPE) TID,LN,DUM,IEMRK(1),ANAM
    NERROR=6

```

```

    IF(LN+N1 .NE. 0) GO TO 999

```

```

    NERROR=7

```

```

    IF(ANAM .NE. ANAME) GO TO 999

```

```

    BACKSPACE NRTAPE

```

```

250 CALL RTAPE (IEMRK(1),ANAME,IA,NR,NC,KR,KC,NRTAPE)
    WRITE (NOT,2006) NR,NC

```

NERROR = 5



```
      NLINE=NLINE+2
      IF (N2.GT.0) CALL WRITIM (IA, NR, NC, ANAME, KR)
C
C  TAPE WRITING SECTION.
300 IF (NWTAPE.LE.0) GO TO 400
      IF (I21.EQ.1H$) CALL INTAPE (NWTAPE, I22)
      IF (I22.EQ.6HREWIND) REWIND NWTAPE
      CALL WTAPE (IA, NR, NC, ANAME, KR, NWTAPE)
      IF (I22.EQ.4HLIST) CALL LTAPE (NWTAPE)
C
400 WRITE (NOT, 2005)
      NLINE=NLINE+2
      RETURN
C
997 CALL LTAPE (NRTAPE)
      GO TO 999
998 WRITE (NOT, 2004) I, JS, IX
999 CALL ZZBOMB (6HREADIM, NERROR)
      END
```

SUBROUTINE READO (A,NR,NC,KR,KC)  
 DIMENSION A(KR,1),X(3),IREMRK(9)  
 DATA NIT,NOT/5,6/

C  
 C READ MATRIX OF OCTAL NUMBERS FROM CARDS (PUNCHED BY SUBROUTINE  
 C PUNCHO) AND PRINT IT SIDE BY SIDE IN OCTAL AND DECIMAL.  
 C THE EXPLANATION OF FORMATS USED BELOW IS ...  
 C A - DENOTES ANY KEY PUNCH SYMBOL. (EG, A1/\*C).  
 C I - DENOTES AN INTEGER NUMBER. (EG, 436).  
 C O - DENOTES AN OCTAL NUMBER.  
 C \*\*\*\* CARD INPUT \*\*\*\*  
 C FIRST CARD - MATRIX NAME, NUMBER OF ROWS, NUMBER OF COLUMNS  
 C WITH A6,I4,I5 FORMAT.  
 C - REMARKS IN COLUMNS 16-69. A-TYPE FORMAT.  
 C MIDDLE CARDS - DATA WITH FORMAT (2I5,3(3X,0I2)).  
 C - 1-ST I5 IS THE ROW NUMBER.  
 C - 2-ND I5 IS THE COL NUMBER OF THE NEXT 0I2 FIELD.  
 C - NEXT 3 0I2 ARE THE ELEMENTS OF THE MATRIX.  
 C LAST CARD - TEN ZEROS IN COLUMNS 1-10.  
 C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.  
 C CODED BY CHRIS CHASE. MARCH 1969.  
 C MODIFIED FOR CONTRACT NAS8-25922, OCTOBER 1970.  
 C LAST REVISION BY RL WOHLN. APRIL 1976.

C SUBROUTINE ARGUMENTS  
 C A = OUTPUT MATRIX READ FROM CARDS.  
 C NR = OUTPUT NUMBER OF ROWS IN MATRIX A.  
 C NC = OUTPUT NUMBER OF COLS IN MATRIX A.  
 C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.  
 C KC = INPUT COL DIMENSION OF A IN CALLING PROGRAM.

C NERROR EXPLANATION  
 C 1 = ROW SIZE EXCEEDS ROW DIMENSION OR  
 C COLUMN SIZE EXCEEDS COLUMN DIMENSION.  
 C 2 = ROW OR COLUMN VALUE OF ELEMENT EXCEEDS MATRIX SIZE.  
 C 3 = DATA ON CARD PAST MATRIX COLUMN SIZE.

C  
 C 1001 FORMAT (A6,I4,I5,9A6)  
 C 1002 FORMAT (2I5,3(3X,0I2))  
 C 2001 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )  
 C \* 2X 9A6//)  
 C 2002 FORMAT (//19H CARD INPUT MATRIX A6, 2X 1H( I4,2H X I4,2H )  
 C \* 3X 9HCONTINUED //)  
 C 2014 FORMAT (1X 2I5, 3X,0I2, 35X, 2X,1PE10.3 )  
 C 2024 FORMAT (1X 2I5,2(3X,0I2),20X,2(2X,1PE10.3))  
 C 2034 FORMAT (1X 2I5,3(3X,0I2), 5X,3(2X,1PE10.3))  
 C 2005 FORMAT (I4H\$END OF READO.)

C  
 C READ IN HEADER CARD.  
 C READ (NIT,1001) ANAME,NR,NC,IREMRK  
 C CALL PAGEHD

WRITE (NOT,2001) ANAME,NR,NC,IREMRK

NERROR = 1

IF (NR.GT.KR .OR. NC.GT.KC) GO TO 999

```

NLINE = 0
DO 105 I=1,NR
DO 105 J=1,NC
105 A(I,J) = 0.
110 READ (NIT,1002) I,JS,X
    IF (I.EQ.0 .AND. JS.EQ.0) GO TO 400
                                NERROR = 2
    IF (I.LE.0 .OR. I.GT.NR .OR. JS.LE.0 .OR. JS.GT.NC) GO TO 998
    JE = JS+2
    IF (JE.LE.NC) GO TO 115
    JX = NC-JS+2
                                NERROR = 3
    DO 112 J=JX,3
112 IF (X(J) .NE. 0.) GO TO 998
    JE = NC
115 N = 0
    DO 120 J=JS,JE
    N = N+1
120 A(I,J) = X(N)
    NLINE = NLINE+1
    IF (NLINE.LE.47) GO TO 125
    CALL PAGEHD
    WRITE (NOT,2002) ANAME,NR,NC
    NLINE = 1
125 NF = JE+1-JS
    IF (NF.EQ.1) WRITE(NOT,2014) I,JS,(A(I,J),J=JS,JE),(A(I,J),J=JS,JE)
    IF (NF.EQ.2) WRITE(NOT,2024) I,JS,(A(I,J),J=JS,JE),(A(I,J),J=JS,JE)
    IF (NF.EQ.3) WRITE(NOT,2034) I,JS,(A(I,J),J=JS,JE),(A(I,J),J=JS,JE)
    GO TO 110
C
400 WRITE (NOT,2005)
    RETURN
C
998 WRITE (NOT,2034) I,JS,X,X
999 CALL ZZBOMB (6HREADO ,NERROR)
    END

```

```

SUBROUTINE REVADD (ALPHA,A,IVEC,JVEC,Z,NRA,NCA,NRZ,NCZ,KRA,KRZ)
DIMENSION A(KRA,1), IVEC(1), JVEC(1), Z(KRZ,1)

```

```

C
C REARRANGE AND ADD ROWS AND COLUMNS OF ALPHA * MATRIX A INTO
C MATRIX Z.
C BE SURE MATRIX Z IS DEFINED BEFORE CALLING THIS SUBROUTINE. FOR
C EXAMPLE, CALL ZERC TO CLEAR MATRIX Z.
C CALLS FORMA SUBROUTINE ZBOMB.
C CODED BY RF HRUDA. JULY 1965.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C ALPHA = INPUT SCALAR THAT MULTIPLIES MATRIX A.
C A = INPUT MATRIX TO BE ARRANGED AND ADDED. SIZE(NRA,NCA).
C IVEC = INPUT 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 = INPUT VECTOR. SIZE(NCA).
C JVEC(J)=COL 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 Z = INPUT/OUTPUT MATRIX TO WHICH ALPHA*A IS ADDED. SIZE(NRZ,NCZ).
C NRA = INPUT NUMBER OF ROWS IN MATRIX A.
C NCA = INPUT NUMBER OF COLS IN MATRIX A.
C NRZ = INPUT NUMBER OF ROWS IN MATRIX Z.
C NCZ = INPUT NUMBER OF COLS IN MATRIX Z.
C KRA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KRZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.
C
C NERROR EXPLANATION
C 1 = ROW LOCATION OUTSIDE MATRIX Z.
C 2 = COLUMN LOCATION OUTSIDE MATRIX Z.
C
DO 30 IA=1,NRA
IZ = IABS(IVEC(IA))
IF (IZ .EQ. 0) GO TO 30
NERROR = 1
IF (IZ .GT. NRZ) GO TO 999
DO 25 JA=1,NCA
JZ = IABS(JVEC(JA))
IF (JZ .EQ. 0) GO TO 25
NERROR = 2
IF (JZ .GT. NCZ) GO TO 999
SIGN = +1.
IF (IVEC(IA).LT.0 .AND. JVEC(JA).GT.0 .OR.
* IVEC(IA).GT.0 .AND. JVEC(JA).LT.0) SIGN=-1.
Z(IZ,JZ) = Z(IZ,JZ) + SIGN*ALPHA*A(IA,JA)
25 CONTINUE
30 CONTINUE
RETURN
C
999 CALL ZBOMB (6HREVADD,NERROR)

```

REVADD-- 2/ 2

---

END

```

SUBROUTINE REVIJ(AZ,IVEC,JVEC,NRA,NCA,NRZ,NCZ,KRAZ)
DIMENSION AZ(KRAZ,1),IVEC(1),JVEC(1)
COMMON / LWRKV1 / V(500)

```

```

C
C REARRANGE AND ADD ROWS AND COLUMNS OF MATRIX (A) INTO
C MATRIX (Z). BOTH MATRIX (A) AND (Z) SHARE THE SAME STORAGE.
C MATRIX (Z) IS SET EQUAL TO ZERO INITIALLY BY THE ROUTINE.
C CALLS FORMA ROUTINE ZZBOMB.
C CODED BY JOHN ADMIRE *NASA* AUG 1973.

```

```

C
C SUBROUTINE ARGUMENTS

```

```

C AZ - INPUT MATRIX (A) TO BE ARRANGED AND ADDED TO MATRIX (Z).
C AZ - OUTPUT MATRIX (Z) RESULT OF ARRANGING AND ADDING (A) TO (Z).
C IVEC - INPUT INTEGER VECTOR (NRA)
C ABS(IVEC(I))=ROW OF (Z) TO WHICH ROW I OF (A) IS ADDED.
C IF(IVEC(I) NEGATIVE) THE SIGNS IN ROW I OF (A) ARE
C CHANGED BEFORE BEING ADDED INTO (Z).
C IF(IVEC(I) ZERO) THE ROW I IN (A) IS OMITTED FROM (Z).
C JVEC - INPUT INTEGER VECTOR (NCA)
C ABS(JVEC(J))=COLUMN OF (Z) TO WHICH COLUMN J OF (A) IS ADDED
C IF(JVEC(J) NEGATIVE) THE SIGNS IN COLUMN J OF (A) ARE
C CHANGED BEFORE BEING ADDED INTO (Z).
C IF(JVEC(J) ZERO) THE COLUMN J IN (A) IS OMITTED FROM (Z)
C NRA - INPUT NUMBER OF ROWS IN MATRIX (A)
C NCA - INPUT NUMBER OF COLUMNS IN MATRIX (A)
C NRZ - OUTPUT NUMBER OF ROWS IN MATRIX (Z)
C NCZ - OUTPUT NUMBER OF COLUMNS IN MATRIX (Z)
C KRAZ - INPUT ROW DIMENSION OF AZ IN CALLING PROGRAM

```

```

C
C NERROR=1
C IF(KRAZ .GT. 500 .OR. NRA .GT. KRAZ .OR. NCA .GT. 500) GO TO 999
C NRZ=0
C NCZ=0
C DO 10 I=1,NRA
C IF(IABS(IVEC(I)) .GT. NRZ) NRZ=IABS(IVEC(I))
10 CONTINUE
C DO 20 J=1,NCA
C IF(IABS(JVEC(J)) .GT. NCZ) NCZ=IABS(JVEC(J))
20 CONTINUE
C NERROR=?
C IF(NRZ .GT. KRAZ .OR. NCZ .GT. 500) GO TO 999
C MAXI=NRA
C MAXJ=NCA
C IF(NRZ .GT. NRA) MAXI=NRZ
C IF(NCZ .GT. NCA) MAXJ=NCZ
C DO 70 J=1,NCA
C DO 30 I=1,NRA
30 V(I)=AZ(I,J)
C DO 40 I=1,MAXI
40 AZ(I,J)=0.
C DO 70 I=1,NRA
C II=IABS(IVEC(I))
C IF(IVEC(I))50,70,60
50 AZ(II,J)=AZ(II,J)-V(I)
C GO TO 70

```

```
60 AZ(II,J)=AZ(II,J)+V(I)
70 CONTINUE
   DO 120 I=1,NRZ
   DO 80 J=1,NCA
80  V(J)=AZ(I,J)
   DO 90 J=1,MAXJ
90  AZ(I,J)=0.
   DO 120 J=1,NCA
   JJ=IABS(JVEC(J))
   IF(JVEC(J))100,120,110
100 AZ(I,JJ)=AZ(I,JJ)-V(J)
   GO TO 120
110 AZ(I,JJ)=AZ(I,JJ)+V(J)
120 CONTINUE
   RETURN
999 CALL ZZBOMB(6HREVIJ ,NERROR)
   END
```

ROWMLT

```
      SUBROUTINE ROWMLT (AVEC,B,Z,NR,NC,KR)
      DIMENSION AVEC(1), B(KR,1), Z(KR,1)
C
C  MULTIPLY EACH ELEMENT IN ROW(I) OF MATRIX B BY ELEMENT(I) OF
C  VECTOR AVEC.
C  MATRICES B,Z MAY SHARE SAME CORE LOCATIONS.
C  CODED BY  RL WOHLER.  FEBRUARY 1965.
C
C  SUBROUTINE ARGUMENTS
C  AVEC = INPUT  VECTOR.  SIZE(NR).
C  B    = INPUT  MATRIX.  SIZE(NR,NC).
C  Z    = OUTPUT RESULT MATRIX.  SIZE(NR,NC).
C  NR   = INPUT  NUMBER OF ROWS IN MATRICES B,Z.  ELEMENTS IN VECTOR AVEC.
C  NC   = INPUT  NUMBER OF COLS IN MATRICES B,Z.
C  KR   = INPUT  ROW DIMENSION OF B,Z IN CALLING PROGRAM.
C
      DO 10 I=1,NR
      DO 10 J=1,NC
10  Z(I,J) = AVEC(I) * B(I,J)
      RETURN
      END
```



```

SUBROUTINE RTAPE (IARUNO,IANAME, A,NRA,NCA, KR,KC,NTAPE)
DIMENSION A(KR,1)
DATA NIT,NOT/5,6/

```

```

C
C READ MATRIX A FROM TAPE BY IDENTIFICATION OF IARUNO, IANAME.
C CALLS FORMA SUBROUTINES LTAPE,PAGEHD,ZZBOMB.
C CODED BY WA BENFIELD. JUNE 1966.
C REVISED BY RF HRUDA. JULY 1968.
C REVISED BY R A PHILIPPUS. APRIL 1969.
C MODIFIED FOR CONTRACT NAS8-25922, OCTOBER 1970.
C
C SUBROUTINE ARGUMENTS
C IARUNO = INPUT RUN NUMBER OF MATRIX A. (A6 FORMAT).
C IANAME = INPUT MATRIX IDENTIFICATION. (A6 FORMAT).
C A = OUTPUT MATRIX READ FROM TAPE. SIZE(NRA,NCA).
C NRA = OUTPUT NUMBER OF ROWS OF MATRIX A. WILL BE READ FROM TAPE.
C NCA = OUTPUT NUMBER OF COLS OF MATRIX A. WILL BE READ FROM TAPE.
C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C KC = INPUT COL DIMENSION OF A IN CALLING PROGRAM.
C NTAPE = INPUT NUMBER OF TAPE. (E.G. 10).
C
C NERROR EXPLANATION
C 1 = MATRIX REQUESTED IS NOT DENSE.
C 2 = ROW OR COLUMN SIZE EXCEEDS DIMENSION SIZE.
C 3 = MATRIX/RUNNO REQUESTED NOT FOUND ON TAPE.
C
C 3001 FORMAT (29H1RTAPE CANNOT FIND RUNNO = A6 /
* 21X 8HANAME = A6 / 29X 6H----- )
C
C NTIME = 0
C SEARCH TAPE FOR CORRECT HEADING.
5 READ (NTAPE) TAPEID,LN,IEOTCK,ITRUNO,ITNAME,NRA,NCA,DATE,ITYPE,NNZ
IF (ITRUNO .EQ. IARUNO .AND. ITNAME .EQ. IANAME) GO TO 10
IF (IEOTCK .EQ. 3HEOT) GO TO 20
READ (NTAPE) DUM
GO TO 5
C
C MATRIX HAS BEEN FOUND.
10 NERROR = 1
IF (ITYPE .NE. 5HDENSE .AND. NNZ .NE. 0) GO TO 999 NERROR = 2
IF (NRA.GT.KR .OR. NCA.GT.KC) GO TO 999
READ (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
RETURN
C
C MATRIX CANNOT BE FOUND. SEARCH TAPE ONCE MORE.
20 NTIME = NTIME+1 NERROR = 3
IF (NTIME .EQ. 2) GO TO 998
REWIND NTAPE
GO TO 5
998 WRITE (NOT,3001) IARUNO,IANAME
CALL LTAPE (NTAPE)
999 CALL ZZBOMB (6HRTAPE ,NERROR)

```

RTAPE -- 2/ 2

END

RWND

-----

SUBROUTINE RWND (NTAPE)

C REWIND TAPE.  
C CODED BY RL WOHLER. MARCH 1976.  
C  
C SUBROUTINE ARGUMENT (INPUT)  
C NTAPE = NUMBER OF TAPE. (EG 10).  
C  
REWIND NTAPE  
RETURN  
END

SIGMA  
-----

SUBROUTINE SIGMA (Z,N,KR)  
DIMENSION Z(KR,1)

C  
C GENERATE A MATRIX OF ONES ON AND BELOW THE DIAGONAL.  
C CODED BY RL WOHLER. FEB 1965.  
C  
C SUBROUTINE ARGUMENTS  
C Z = OUTPUT MATRIX GENERATED. SIZE(N,N).  
C N = INPUT SIZE OF MATRIX Z (SQUARE).  
C KR = INPUT ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.  
C  
DO 10 I=1,N  
DO 10 J=I,N  
Z(I,J) = 0.0  
10 Z(J,I) = 1.0  
RETURN  
END

SKPR

-----

SUBROUTINE SKPR (NTAPE,NREC)

```
C SKIP NREC LOGICAL RECOPIES (FORWARD OR BACKWARD) ON NTAPE.  
C CODED BY RL WOHLER. MARCH 1976.  
C  
C SUBROUTINE ARGUMENTS (ALL INPUT)  
C NTAPE = NUMBER OF TAPE. (EG 10).  
C NREC = NUMBER OF LOGICAL RECORDS TO SKIP (FORWARD OR BACKWARD).  
C  
C IF (NREC .EQ. 0) RETURN  
C IF (NREC .LT. 0) GO TO 20  
C DO 15 IREC=1,NREC  
15 READ (NTAPE)  
C RETURN  
C  
20 LREC = IABS(NREC)  
C DO 25 IREC=1,LREC  
25 BACKSPACE NTAPE  
C RETURN  
C END
```

```

SUBROUTINE SMEQ1 (A,B,Z,N,KR)
DIMENSION A(KR,1),B(1),Z(1)
DATA TOL/1.E-15/

```

```

C
C SOLUTION OF LINEAR SIMULTANEOUS ALGEBRAIC EQUATIONS, A*Z = B.
C GAUSS ELIMINATION METHOD. FORWARD SOLUTION TRANSFORMS ORIGINAL SYSTEM
C INTO TRIANGULAR FORM. BACK SOLUTION THEN GIVES RESULT.
C LARGEST PIVOTAL DIVISOR IS USED TO AVOID DIVISION BY SMALL NUMBERS.
C THE ROWS ARE INTERCHANGED WHEN NECESSARY TO ACCOMPLISH THIS.
C IF NO PIVOT CAN BE FOUND EXCEEDING 1.E-15, THE MATRIX IS CONSIDERED
C SINGULAR AND THE PROGRAM STOPPED.
C CALLS FORMA SUBROUTINE ZZBOMB.
C DEVELOPED BY CARL BOOLEY. AUGUST 1968.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS

```

```

C A = INPUT SQUARE MATRIX OF COEFFICIENTS. SIZE(N,N). *DESTROYED*
C B = INPUT RIGHT HAND SIDE VECTOR. SIZE(N). *DESTROYED*
C Z = OUTPUT RESULT VECTOR. SIZE(N).
C N = INPUT NUMBER OF EQUATIONS.
C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.

```

```

C NERPOR EXPLANATION

```

```

C 1 = MATRIX IS NON-POSITIVE DEFINITE FOR SIZE = 1.
C 2 = MATRIX IS NON-POSITIVE DEFINITE.

```

```

C IF (N .GT. 1) GO TO 5

```

```

NERROR = 1

```

```

C IF (ABS(A(1,1)) .LE. TOL) GO TO 999
C Z(1) = B(1)/A(1,1)
C RETURN

```

```

C FORWARD SOLUTION.

```

```

5 DO 25 L=1,N
  AMAX = TOL
  DO 10 I=L,N
    IF (ABS(A(I,L)) .LT. ABS(AMAX)) GO TO 10
    AMAX = A(I,L)
    IMAX = I
  10 CONTINUE

```

```

NERROR = 2

```

```

C IF (ABS(AMAX) .LE. TOL) GO TO 999
C DO 15 J=L,N
C SAVE = A(IMAX,J)
C A(IMAX,J) = A(L,J)
15 A(L,J) = SAVE/AMAX
  SAVE = B(IMAX)
  B(IMAX) = B(L)
  B(L) = SAVE/AMAX
  IF (L .EQ. N) GO TO 40
  LPI = L + 1
  DO 25 I=LPI,N
    DO 20 J=LPI,N
20 A(I,J) = A(I,J) - A(I,L)*A(L,J)
25 B(I) = B(I) - A(I,L)*B(L)

```

```
BACK SOLUTION.  
40 Z(N) = B(N)  
   NM1 = N - 1  
   DO 45 L=1,NM1  
     I = N - L  
     Z(I) = B(I)  
     IP1 = I+1  
     DO 45 J=IP1,N  
45 Z(I) = Z(I) - A(I,J)*Z(J)  
   RETURN
```

```
C  
999 CALL Z2BOMB (6HSMEQ1 ,NERROR)  
   END
```

```

SUBROUTINE SRED1 (A,R,T,N,NR,IFT,KART)
DIMENSION A(KART,1),R(KART,1),T(KART,1)
DATA EPS/1.E-15/

```

```

C
C REDUCE STIFFNESS MATRIX (A) TO FORM REDUCED STIFFNESS MATRIX (R) AND
C REDUCING TRANSFORMATION (T).
C DEGREES OF FREEDOM TO BE REDUCED MUST BE POSITIONED LAST IN MATRIX A.
C MATRIX (A) SHOULD BE POSITIVE DEFINITE, SYMMETRIC. LOWER HALF OF
C MATRIX (A) IS USED.
C MATRIX (T) MAY BE A SCALAR ARGUMENT IF THE REDUCING TRANSFORMATION
C IS NOT FORMED.
C MATRICES (A), (R), AND (T) MAY SHARE THE SAME CORE LOCATIONS IN ANY
C COMBINATIONS. POSSIBLE COMBINATIONS OF INPUT ARGUMENTS ARE SHOWN
C BELOW WITH THE RESULTING OUTPUT FROM THE SUBROUTINE.
C
C      *CALLING ARGUMENTS*      *RESULTING OUTPUT*
C      CALL SRED1 (A,R,T,N,NR,1,KART)      A=A, R=R, T=T
C      CALL SRED1 (A,A,T,N,NR,1,KART)      A=R, T=T
C      CALL SRED1 (A,T,T,N,NR,1,KART)      A=A, T=T
C      CALL SRED1 (A,A,A,N,NR,1,KART)      A=T
C      CALL SRED1 (A,R,A,N,NR,1,KART)      R=R, A=T
C      CALL SRED1 (A,R,T,N,NR,0,KART)      A=A, R=R
C      CALL SRED1 (A,A,T,N,NR,0,KART)      A=R
C CALLS FORMA SUBROUTINE ZZBOMB.
C DEVELOPED BY CS BODLEY AND WA BENFIELD. OCTOBER 1971.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

#### SUBROUTINE ARGUMENTS

```

C A = INPUT STIFFNESS MATRIX TO BE REDUCED. SIZE(N,N).
C R = OUTPUT REDUCED STIFFNESS MATRIX. SIZE(NR,NR).
C T = OUTPUT REDUCING TRANSFORMATION MATRIX. SIZE(N,NR).
C N = INPUT SIZE OF MATRIX A.
C NR = INPUT SIZE OF REDUCED MATRIX R.
C IFT = INPUT =0, TRANSFORMATION MATRIX T WILL NOT BE CALCULATED AND
C T NEED NOT BE DIMENSIONED IN CALLING PROGRAM.
C =1, TRANSFORMATION MATRIX T WILL BE CALCULATED.
C KART = INPUT ROW DIMENSION OF A,R,T IN CALLING PROGRAM.

```

#### ERROR EXPLANATION

```

C I = MATRIX IS NON-POSITIVE DEFINITE.

```

```

DO 5 I=1,N
DO 5 J=1,I
5 R(I,J) = A(I,J)
NRPI = NR+1

```

NERROR=1

```

C CALCULATE REDUCED STIFFNESS MATRIX.
DO 10 L=NRPI,N
K = N-L+NRPI
KMI = K-1
IF (R(K,K) .LT. EPS) GO TO 999
DO 10 I=1,KMI
S = R(K,I)/R(K,K)
DO 10 J=1,I
10 R(I,J) = R(I,J) - S*R(K,J)
DO 15 I=1,NR

```



```
      DO 15 J=I, NR
15 R(I,J) = R(J,I)
C
C CALCULATE REDUCTION TRANSFORMATION MATRIX.
      IF (IFT .EQ. 0) RETURN
      DO 29 L=NRPI, N
      S = R(L,L)
      DO 25 K=1, NR
25 R(L,K) = R(L,K)/S
      IF(L .GE. N) GO TO 29
      LPI = L+1
      DO 27 I=LPI, N
      DO 27 J=1, NR
27 R(I,J) = R(I,J) - R(I,L)*R(L,J)
29 CONTINUE
      DO 35 I=1, NR
      DO 32 J=1, NR
32 T(I,J) = 0.0
35 T(I,I) = 1.0
      DO 45 I=NRPI, N
      DO 45 J=1, NR
45 T(I,J) = -R(I,J)
      RETURN
C
999 CALL ZZBOME (6HSRED1 ,NERROR)
      END
```

```

SUBROUTINE SRED2 (A,R,T,N,NR,IFT,KART)
DIMENSION A(KART,1), R(KART,1), T(KART,1)
DATA EPS/1.E-15/

```

```

C
C REDUCE STIFFNESS MATRIX (A) TO FORM REDUCED STIFFNESS MATRIX (R) AND
C REDUCING TRANSFORMATION (T).
C DEGREES OF FREEDOM TO BE REDUCED MUST BE POSITIONED FIRST IN MATRIX (A).
C MATRIX (A) SHOULD BE POSITIVE DEFINITE, SYMMETRIC. UPPER HALF OF
C MATRIX (A) IS USED.
C MATRIX (T) MAY BE A SCALAR ARGUMENT IF THE REDUCING TRANSFORMATION
C IS NOT FORMED.
C MATRICES (A), (R), AND (T) MAY SHARE THE SAME CORE LOCATIONS IN ANY
C COMBINATIONS. POSSIBLE COMBINATIONS OF INPUT ARGUMENTS ARE SHOWN
C BELOW WITH THE RESULTING OUTPUT FROM THE SUBROUTINE.

```

*CALLING ARGUMENTS*	*RESULTING OUTPUT*
CALL SRED2 (A,P,T,N,NR,1,KART)	A=A, R=R, T=T
CALL SRED2 (A,A,T,N,NR,1,KART)	A=R, T=T
CALL SRED2 (A,T,T,N,NP,1,KART)	A=A, T=T
CALL SRED2 (A,A-A,N,NR,1,KART)	A=T
CALL SRED2 (A,R,A,N,NR,1,KART)	R=R, A=T
CALL SRED2 (A,P,T,N,NR,0,KART)	A=A, R=R
CALL SRED2 (A,A,T,N,NP,0,KART)	A=R

```

C CALLS FORMA SUBROUTINE ZZEOME.
C DEVELOPED BY CS BODLEY AND WA BENFIELD. JUNE 1972.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS
C A = INPUT STIFFNESS MATRIX TO BE REDUCED. SIZE(N,N).
C R = OUTPUT REDUCED STIFFNESS MATRIX. SIZE(NR,NR).
C T = OUTPUT REDUCING TRANSFORMATION MATRIX. SIZE(N,NR).
C N = INPUT SIZE OF MATRIX A.
C NP = INPUT SIZE OF REDUCED MATRIX R.
C IFT = INPUT =0, TRANSFORMATION MATRIX T WILL NOT BE CALCULATED AND
C T NEED NOT BE DIMENSIONED IN CALLING PROGRAM.
C =1, TRANSFORMATION MATRIX T WILL BE CALCULATED.
C KART = INPUT ROW DIMENSION OF A,R,T IN CALLING PROGRAM.

```

```

C NEPROR EXPLANATION
C 1 = MATRIX IS NON-POSITIVE DEFINITE.

```

```

ND = N - NR
DO 5 I=1,N
DO 5 J=I,N
5 R(I,J) = A(I,J)
NDP1 = ND+1

```

NERROR=1

```

C CALCULATE REDUCED STIFFNESS MATRIX.
DO 11 L=1,ND
IF (ABS(R(L,L)) .LT. EPS) GO TO 999
IF (L .GE. N) GO TO 11
LP1 = L+1
DO 10 I=LP1,N
S = R(L,I)/R(L,L)
DO 10 J=I,N
R(I,J) = R(I,J) - S*R(L,J)

```

```
10 CONTINUE
11 CONTINUE
    IF (IFT .EQ. 0) GO TO 50
C
C CALCULATE REDUCTION TRANSFORMATION MATRIX.
    DO 20 L=1,ND
    S = R(L,L)
    DO 15 K=L,N
15 R(L,K) = R(L,K)/S
    IF (L .EQ. 1) GO TO 20
    LMI = L-1
    LPI = L+1
    DO 16 I=1,LMI
    DO 16 J=LPI,N
16 R(I,J) = R(I,J) - R(I,L)*R(L,J)
20 CONTINUE
    DO 30 I=NDP1,N
    DO 29 J=1,NR
29 T(I,J) = 0.0
    IMND = I - ND
30 T(I,IMND) = 1.0
50 DO 60 L=ND-1,N
    J = L - ND
C SYMMETRIZE R AND START IN 1,1 LOCATION.
    DO 65 K=NDP1,N
    I = K - ND
    IF (I .GT. J) GO TO 66
    R(I,J) = R(K,L)
    GO TO 65
66 R(I,J) = R(L,K)
65 CONTINUE
C START T IN 1,1 LOCATION.
    IF (IFT .EQ. 0) GO TO 60
    DO 67 I=1,ND
67 T(I,J) = -R(I,L)
60 CONTINUE
    RETURN
C
999 CALL ZZBOMB (6HSRED2 ,NERRGR)
    END
```

```

SUBROUTINE SRED3 (A,IV,R,T,N,NR,IFT,KART)
DIMENSION A(KART,1), R(KART,1), T(KART,1), IV(1)
COMMON /LWRKV1/ W(500)
DATA EPS/1.E-15/

```

```

C
C SUBROUTINE TO REDUCE MATRIX (A) TO FORM REDUCED STIFFNESS MATRIX (R)
C AND REDUCING TRANSFORMATION (T).
C ROWS AND COLUMNS TO BE REDUCED OUT MAY BE ANYWHERE IN (A) AND ARE
C SPECIFIED BY THE INTEGER VECTOR (IV).
C THE ORIGINAL NUMBER OF ROWS AND COLUMNS IN (A) ARE THE SAME FOR (R)
C WITH ZERO ROW AND COLUMN ELEMENTS FOR THE REDUCED ROWS AND COLUMNS.
C THE REDUCING TRANSFORMATION (T) LIKEWISE WILL BE SQUARE.
C MATRIX (A) SHOULD BE POSITIVE DEFINITE, SYMMETRIC.
C ALL OF MATRIX (A) IS USED.
C MATRIX (T) MAY BE A SCALAR ARGUMENT IF THE REDUCING TRANSFORMATION
C IS NOT FORMED.
C MATRICES (A), (R), AND (T) MAY SHARE THE SAME CORE LOCATIONS IN ANY
C COMBINATIONS. POSSIBLE COMBINATIONS OF INPUT ARGUMENTS ARE SHOWN
C BELOW WITH THE RESULTING OUTPUT FROM THE SUBROUTINE.

```

	*CALLING ARGUMENTS*	*RESULTING OUTPUT*
C	CALL SRED3 (A,IV,R,T,N,NR,1,KART)	A=A, R=R, T=T
C	CALL SPED3 (A,IV,A,T,N,NR,1,KART)	A=R, T=T
C	CALL SRED3 (A,IV,R,A,N,NR,1,KART)	R=R, A=T
C	CALL SPED3 (A,IV,A,A,N,NR,1,KART)	A=T
C	CALL SRED3 (A,IV,R,R,N,NR,1,KART)	A=A R=T
C	CALL SRED3 (A,IV,R,T,N,NR,0,KART)	A=A, R=R
C	CALL SRED3 (A,IV,A,T,N,NR,0,KART)	A=R

```

C CALLS FORMA SUBROUTINE ZZBCME.
C DEVELOPED BY WA BENFIELD. JANUARY 1974.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS
C A = INPUT STIFFNESS MATRIX TO BE REDUCED. SIZE(N,N).
C IV = INPUT INTEGER ROW MATRIX CONTAINING THE ROW-COLUMN
C LOCATIONS TO BE REDUCED. SIZE(NR).
C R = OUTPUT REDUCED STIFFNESS MATRIX. SIZE(N,N).
C T = OUTPUT REDUCING TRANSFORMATION MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRICES A,R,T. MAX=500.
C NR = INPUT NUMBER OF ROW-COLUMNS TO BE REDUCED.
C IFT = INPUT =0. TRANSFORMATION MATRIX T WILL NOT BE CALCULATED
C AND T NEED NOT BE DIMENSIONED IN CALLING PROGRAM.
C =1. TRANSFORMATION MATRIX T WILL BE CALCULATED.
C KART = INPUT ROW DIMENSION OF A,R,T IN CALLING PROGRAM.

```

```

C
C ERROR EXPLANATION
C 1 = ROW NUMBER IS NEGATIVE.
C 2 = MATRIX IS NON-POSITIVE DEFINITE.

```

```

DO 5 I=1,N
DO 5 J=1,N
5 R(I,J) = A(I,J)
DO 35 K=1,NR
IR = IV(K)

```

```

IF (IR .LE. 0) GO TO 999

```

ERROR=1

-----  
NERROR=2

```
IF (R(IR,IR) .LT. EPS) GO TO 999
C = R(IR,IR)
DO 10 J=1,N
10 R(IR,J) = P(IR,J)/C
DO 30 I=1,N
IF (I .EQ. IR) GO TO 30
C = R(I,IR)
DO 20 J=1,N
20 R(I,J) = R(I,J) - C*R(IR,J)
30 CONTINUE
35 CONTINUE
DO 90 I=1,N
DO 37 K=1,N
37 W(K) = R(I,K)
DO 40 K=1,NR
IF (IV(K) .EQ. I) GO TO 60
40 CONTINUE
IF (IFT .EQ. 0) GO TO 90
DO 50 J=1,N
50 T(I,J) = 0.0
T(I,I) = 1.0
GO TO 90
60 DO 70 J=1,N
R(J,I) = 0.0
70 R(I,J) = 0.0
IF (IFT .EQ. 0) GO TO 90
DO 75 J=1,N
75 T(I,J) = -W(J)
T(I,I) = 0.0
90 CONTINUE
RETURN
999 CALL ZZBOMB (6HSRED3 ,NERROR)
END
```

```

SUBROUTINE START
COMMON /LSTART/IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE /NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C EACH TIME THIS ROUTINE IS CALLED THE COMPUTER IS INTEROGRATED TO
C OBTAIN THE DATE, TIME OF DAY, AND THE CPU TIME. TIME OF DAY AND CPU ARE
C PRINTED ON A NEW PAGE. NPAGE AND NLINE ARE THEN SET EQUAL TO ZERO.
C CARD 1 IS READ NEXT TO OBTAIN IRUNNO, MINI, AND UNAME.
C IF IRUNNO=4HSTOP THE PROGRAM IS STOPPED AT THIS POINT.
C IF IRUNNO=6HRETURN A RETURN IS MADE TO THE CALLING PROGRAM.
C IF IRUNNO IS NOT EQUAL TO STOP OR RETURN TWO ADDITIONAL CARDS ARE READ.
C CARD 2 IS READ NEXT FOR TITLE1.
C CARD 3 IS READ NEXT FOR TITLE2.
C LAST A RETURN IS MADE TO THE CALLING PROGRAM.

```

## CARD INPUT

```

C
C      IRUNNO,MINI,UNAME      FORMAT (A6,A4,3A6)
C      TITLE1                FORMAT (12A6)
C      TITLE2                FORMAT (12A6)

```

## DEFINITIONS

```

C      IRUNNO      = RUN NUMBER
C      DATE        = DATE
C      NPAGE       = PAGE NUMBER
C      UNAME       = USEPS NAME
C      TITLE1      = FIRST TITLE
C      TITLE2      = SECOND TITLE
C      NLINE       = LINE NUMBER
C      MAXLIN      = MAXIMUM NUMBER OF LINES PRINTED PER PAGE
C      MINI        = PRINT OPTION (IF MINI=4HMINI OTHER FORMA ROUTINES WILL
C                   ATTEMPT TO MINIMIZE THE NUMBER OF PAGES PRINTED BY PRINTING
C                   MORE THAN ONE SET OF DATA PER PAGE)

```

```

C      MODIFIED AUG. 1973 BY JOHN ADMIRE *NASA*

```

```

C
1001 FORMAT (A6,A4,3A6)
1002 FORMAT (12A6)
2002 FORMAT (1H1 6(//) 55X 10HTIME SHFET / 38X 45(1H-) //
*          38X 30HCURRENT TIME OF DAY IN H,M,S = A 6 //
*          38X 26HTOTAL CPTIME USED TO NOW = I5, 9H SECONDS. // )
2003 FORMAT (36HIEND OF INPUT DATA HAS BEEN REACHED.)

```

```

C      CALL SCLOCK (DATE,TIME,ESEC,E60SEC)
C      CALL CPUTIM (ICTIME)
C      ICTIM=ICTIME/1000000
C      WRITE (NOT,2002) TIME,ICTIM

```

```

C      NPAGE=0
C      NLINE=0
C      MAXLIN=52

```

```

C      READ (NIT,1001) IPUNNO,MINI,UNAME
C      IF (IRUNNO .NE. 4HSTOP .AND. IRUNNO .NE. 6HRETURN) GO TO 10
C      WRITE (NOT,2003)

```

```
IF(IRUNNO .EQ. 4HSTOP) STOP  
IF(IRUNNO .EQ. 6HRETURN) RETURN
```

C

```
10 READ (NIT,1002) TITLE1  
   READ (NIT,1002) TITLE2  
   RETURN  
   END
```

```

SUBROUTINE STIF1 (PP,DAE,Z,NPP,NDAE,KDAE,KZ)
DIMENSION PP(1), DAE(KDAE,1), Z(KZ,1)
C
C CALCULATE STIFFNESS MATRIX (FREE-FREE) FOR A LONGITUDINAL ROD.
C ASSUMES CONSTANT FORCE BETWEEN CONSECUTIVE PANEL POINTS.
C TRANSLATION AT EACH PANEL POINT ARE THE GENERALIZED COORDINATES
C INPUT IS DISTRIBUTED STIFFNESS (AE).
C SUBROUTINE IS ALSO APPLICABLE FOR TORSIONAL ROD. THEN ROTATION AT
C EACH PANEL POINT ARE THE GENERALIZED COORDINATES, DISTRIBUTED
C STIFFNESS IS GJ.
C CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY C LODLEY. FEBRUARY 1966.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C PP = INPUT VECTOR OF PANEL POINTS. SIZE(NPP).
C DAE = INPUT MATRIX OF DISTRIBUTED STIFFNESS STRAIGHT LINE
C SEGMENT DATA. SIZE(NDAE,4).
C COL 1 = X AT SEGMENT END 1.
C COL 2 = X AT SEGMENT END 2.
C COL 3 = STIFFNESS AT SEGMENT END 1.
C COL 4 = STIFFNESS AT SEGMENT END 2.
C Z = OUTPUT STIFFNESS MATRIX. SIZE(NPP,NPP).
C NPP = INPUT NUMBER OF PANEL POINTS. SIZE OF VECTOR PP, MATRIX Z.
C NDAE = INPUT NUMBER OF SEGMENTS (ROWS) IN DAE.
C KDAE = INPUT ROW DIMENSION OF DAE IN CALLING PROGRAM.
C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = LESS THAN 2 PANEL POINTS.
C 2 = PANEL POINTS NOT IN INCREASING ORDER.
C 3 = DISTRIBUTED DATA MUST START AND END ON FIRST
C AND LAST PANEL POINTS.
C 4 = DISTRIBUTED DATA HAS GAPS.
C 5 = NEGATIVE STIFFNESS IS NOT ALLOWED.
C
C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER.
C
C IF (NPP .LT. 2) GO TO 999
C
C DO 5 K=2,NPP
C IF (PP(K-1) .GE. PP(K)) GO TO 999
C 5 CONTINUE
C
C CHECK FIRST AND LAST POINTS OF DISTRIBUTED STIFFNESS MATRIX.
C
C IF (DAE(1,1) .NE. PP(1) .OR. DAE(NDAE,2) .NE. PP(NPP)) GO TO 999
C
C INITIALIZE DATA.
C DO 10 I=1,NPP
C DO 10 J=1,NPP
C 10 Z(I,J) = 0.0
C NBEAMS = NPP-1
C X2SAVE = DAE(1,1)

```

NERROR = 1

NERROR = 2

NERROR = 3



```

DO 90 I=1,NDAE
X1 = DAE(I,1)
X2 = DAE(I,2)
V1 = DAE(I,3)
V2 = DAE(I,4)

```

NERROR = 4

```

NERROR = 4
IF (X1 .GE. X2 .OR. X1 .NE. X2SAVE) GO TO 999

```

NERROR = 5

```

IF (V1 .LE. 0. .OR. V2 .LE. 0.) GO TO 999

```

```

X2SAVE = X2

```

```

DO 32 K=1,NBAYS

```

```

IF (X1 .LT. PP(K+1)) GO TO 34

```

```

32 CONTINUE

```

```

34 XP = X1

```

```

VP = V1

```

```

36 IF (X2 .LT. PP(K+1)) GO TO 38

```

```

XQ = PP(K+1)

```

```

VQ = V1 + (XQ-X1)*(V2-V1)/(X2-X1)

```

```

GO TO 39

```

```

38 XQ = X2

```

```

VQ = V2

```

```

39 B = (VQ-VP)/(XQ-XP)

```

C

```

IF (B .EQ. 0.) GO TO 55

```

```

Z(K,K) = Z(K,K) + ALOG(VQ/VP) / B

```

```

GO TO 70

```

```

55 Z(K,K) = Z(K,K) + (XQ-XP)/VP

```

C

```

70 IF (X2 .LE. PP(K+1)) GO TO 90

```

```

K = K+1

```

```

XP = XQ

```

```

VP = VQ

```

```

GO TO 34

```

```

90 CONTINUE

```

C

```

STOR2 = Z(1,1)

```

```

Z(1,1) = 0.0

```

```

DO 120 K=1,NSAYS

```

```

L = K+1

```

```

STOR1 = 1./STOR2

```

```

STOR2 = Z(L,L)

```

```

Z(K,K) = Z(K,K) + STOR1

```

```

Z(K,L) = -STOR1

```

```

Z(L,K) = -STOR1

```

```

120 Z(L,L) = STOR1

```

```

RETURN

```

C

```

999 CALL ZZBOMB (6HSTIF1 ,NERROR)

```

```

END

```

SUBROUTINE STIF2 (PP,DKAG,DEI,Z,NPP,NDKAG,NDEI,NZ,KDKAG,KDEI,KZ)  
 DIMENSION PP(1),DKAG(KDKAG,1),DEI(KDEI,1),Z(KZ,1)

C CALCULATE STIFFNESS MATRIX (FREE-FREE) FOR A BEAM. ASSUMES CONSTANT  
 C SHEAR AND LINEARLY VARYING BENDING MOMENT BETWEEN CONSECUTIVE PANEL  
 C POINTS. LATERAL TRANSLATION AND ROTATION AT EACH PANEL POINT ARE THE  
 C GENERALIZED COORDINATES. TRANSLATION COORDINATES ARE GROUPED FIRST  
 C FOLLOWED BY ROTATION COORDINATES.  
 C SIGN CONVENTION IS ROTATION = -D(LATERAL DISP)/D(AXIAL COORDINATE).  
 C INPUT IS DISTRIBUTED FLEXURE STIFFNESS, EI, AND ON OPTION  
 C (NDKAG .GT. 0) DISTRIBUTED SHEAR STIFFNESS, KAG.  
 C CALLS FOR A SUBROUTINE ZZBOMB.  
 C CODED BY C RODLEY. FEBRUARY 1966.  
 C LAST REVISION BY WA BENFIELD. MARCH 1976.

C  
 C SUBROUTINE ARGUMENTS  
 C PP = INPUT VECTOR OF PANEL POINTS. SIZE(NPP).  
 C DKAG = INPUT MATRIX OF DISTRIBUTED SHEAR STIFFNESS STRAIGHT LINE  
 C SEGMENT DATA. SIZE(NDKAG,4).  
 C COL 1 = X AT SEGMENT END 1.  
 C COL 2 = X AT SEGMENT END 2.  
 C COL 3 = STIFFNESS AT SEGMENT END 1.  
 C COL 4 = STIFFNESS AT SEGMENT END 2.  
 C DEI = INPUT MATRIX OF DISTRIBUTED FLEXURE STIFFNESS STRAIGHT LINE  
 C SEGMENT DATA. SIZE(NDEI,4).  
 C COLUMNS ARE SIMILAR TO DKAG.  
 C Z = OUTPUT STIFFNESS MATRIX. SIZE(NZ,NZ).  
 C NPP = INPUT NUMBER OF PANEL POINTS. SIZE OF VECTOR PP.  
 C NDKAG = INPUT NUMBER OF SEGMENTS (ROWS) IN DKAG. CAN BE ZERO.  
 C NDEI = INPUT NUMBER OF SEGMENTS (ROWS) IN DEI.  
 C NZ = OUTPUT SIZE OF MATRIX Z. (NZ=2\*NPP).  
 C KDKAG = INPUT ROW DIMENSION OF DKAG IN CALLING PROGRAM.  
 C KDEI = INPUT ROW DIMENSION OF DEI IN CALLING PROGRAM.  
 C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.

C  
 C ERROR EXPLANATION  
 C 1 = LESS THAN 2 PANEL POINTS.  
 C 2 = PANEL POINTS NOT IN INCREASING ORDER.  
 C 3 = DISTRIBUTED DATA MUST START AND END ON FIRST  
 C AND LAST PANEL POINTS.  
 C 4 = DISTRIBUTED DATA HAS GAPS.  
 C 5 = NEGATIVE STIFFNESS IS NOT ALLOWED.

C  
 C CHECK THAT PANEL POINTS ARE IN INCREASING ORDER. NERROR = 1  
 IF (NPP .LT. 2) GO TO 999 NERROR = 2  
 DO 5 K=2,NPP  
 IF (PP(K-1) .GE. PP(K)) GO TO 999  
 5 CONTINUE

C  
 C CHECK FIRST AND LAST POINTS OF DISTRIBUTED STIFFNESS MATRICES. NERROR = 3  
 IF (NDKAG .EQ. 0) GO TO 7  
 IF (DKAG(1,1) .NE. PP(1) .OR. DKAG(NDKAG,2) .NE. PP(NPP)) GO TO 999

7 IF (DEI (1,1) .NE. PP(1) .OR. DEI (NDEI ,2) .NE. PP(NPP))GO TO 999

C INITIALIZE DATA.

NZ = 2\*NPP

DO 10 I=1,NZ

DO 10 J=1,NZ

10 Z(I,J) = 0.0

NBAYS = NPP-1

C

DO 95 NT = 1,2

IF (NT .EQ. 1 .AND. NDKAG .EQ. 0) GO TO 95

IF (NT .EQ. 1) NSEGS = NDKAG

IF (NT .EQ. 1) X2SAVE = DKAG(1,1)

IF (NT .EQ. 2) NSEGS = NDEI

IF (NT .EQ. 2) X2SAVE = DEI(1,1)

C

DO 90 I=1,NSEGS

GO TO (21,22), NT

21 X1 = DKAG(I,1)

X2 = DKAG(I,2)

V1 = DKAG(I,3)

V2 = DKAG(I,4)

GO TO 30

22 X1 = DEI(I,1)

X2 = DEI(I,2)

V1 = DEI(I,3)

V2 = DEI(I,4)

30

IF (X1 .GE. X2 .OR. X1 .NE. X2SAVE) GO TO 999

NERROR = 4

IF (V1 .LE. 0. .OR. V2 .LE. 0.) GO TO 999

NERROR = 5

X2SAVE = X2

DO 32 K=1,NBAYS

IF (X1 .LT. PP(K+1)) GO TO 34

32 CONTINUE

34 XP = X1

VP = V1

36 IF (X2 .LE. PP(K+1)) GO TO 38

XQ = PP(K+1)

VQ = V1 + (XQ-X1)\*(V2-V1)/(X2-X1)

GO TO 39

38 XQ = X2

VQ = V2

39 PL = XQ-XP

HP = XP-PP(K)

HQ = XQ-PP(K)

A = (VP\*HQ - VQ\*HP)/PL

B = (VQ-VP)/PL

VLOG = ALOG(VQ/VP)

GO TO (50,60),NT

C

50 IF (B .EQ. 0.) GO TO 55

Z(K,K) = Z(K,K) + VLOG/b

GO TO 70

55 Z(K,K) = Z(K,K) + PL/VP

```

GO TO 70
60 L = K+NPP
   IF (B .EQ. 0.) GO TO 65
   Z(K,K) = Z(K,K) + (HQ**2-HP**2)/(2.*B) - A*PL/B**2 +A**2*VLOG/B**3
   Z(K,L) = Z(K,L) + PL/B - A*VLOG/B**2
   Z(L,L) = Z(L,L) + VLOG/B
   GO TO 70
65 Z(K,K) = Z(K,K) + (HQ**3-HP**3)/(3.*VP)
   Z(K,L) = Z(K,L) + (HQ**2-HP**2)/(2.*VP)
   Z(L,L) = Z(L,L) + PL/VP

```

```

C
70 IF (X2 .LE. PP(K+1)) GO TO 90
   K = K+1
   XP = XQ
   VP = VQ
   GO TO 36
90 CONTINUE
95 CONTINUE

```

```

C
   NPP1 = NPP + 1
   STR21 = Z(1,1)
   STR22 = Z(1,NPP1)
   STR23 = Z(NPP1,NPP1)
   Z(1,1) = 0.
   Z(1,NPP1) = 0.0
   Z(NPP1,NPP1) = 0.0
   DO 120 K=1,NBAYS
   L = K + 1
   M = K + NPP
   N = M + 1
   D = STR21*STR23 - STR22**2
   BL = PP(K+1) - PP(K)
   STR11 = STR23/D
   STR12 = -STR22/D
   STR13 = STR21/D
   STR21 = Z(L,L)
   STR22 = Z(L,N)
   STR23 = Z(N,N)
   Z(K,K) = Z(K,K) + STR11
   Z(K,L) = -STR11
   Z(K,M) = Z(K,M) + STR12
   Z(K,N) = -(BL*STR11 + STR12)
   Z(L,L) = STR11
   Z(L,M) = -STR12
   Z(L,N) = -Z(K,N)
   Z(M,M) = Z(M,M) + STR13
   Z(M,N) = -(BL*STR12 + STR13)
120 Z(N,N) = B.**2*STR11 + 2.*BL*STR12 + STR13

```

```

C
C SYMMETRIZE.
   DO 160 I=1,NZ
   DO 160 J=I,NZ
160 Z(J,I) = Z(I,J)
RETURN

```

```

C

```

STIF2 -- 4/ 4

999 CALL ZZBOMB (6HSTIF2 ,NERROR)  
END

SYMLH

```
SUBROUTINE SYMLH (A,N,KR)  
DIMENSION A(KR,1)
```

```
C  
C SYMMETRIZE MATRIX A BY PLACING VALUES FROM  
C ABOVE THE DIAGONAL BELOW THE DIAGONAL.  
C CODED BY RL WOHLN. FEB 1965.
```

```
C  
C SUBROUTINE ARGUMENTS  
C A = INPUT, OUTPUT SUPPLIED AND RESULT MATRIX. SIZE(N,N).  
C N = INPUT SIZE OF MATRIX A (SQUARE).  
C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
```

```
C  
DO 10 I=1,N  
DO 10 J=I,N  
10 A(J,I) = A(I,J)  
RETURN  
END
```

SYMUH

SUBROUTINE SYMUH (A,N,KR)  
DIMENSION A(KR,1)

C  
C SYMMETRIZE MATRIX A BY PLACING VALUES FROM  
C BELOW THE DIAGONAL ABOVE THE DIAGONAL.  
C CODED BY RL WOHLER. FEB 1965.  
C  
C SUBROUTINE ARGUMENTS  
C A = INPUT, OUTPUT SUPPLIED AND RESULT MATRIX. SIZE(N,N).  
C N = INPUT SIZE OF MATRIX A (SQUARE).  
C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.  
C  
DO 10 I=1,N  
DO 10 J=I,N  
10 A(I,J) = A(J,I)  
RETURN  
END

TERP1  
-----

SUBROUTINE TERP1 (XA,XZ,YA,YZ,NXA,NXZ,NCA,KA,KZ)  
DIMENSION XA(1),XZ(1),YA(KA,1),YZ(KZ,1)

```

C
C LINEAR INTERPOLATION.
C VALUES OF XZ MAY BE OUTSIDE OF XA. (EXTRAPOLATION).
C CODED BY RF HRUDA. SEPTEMBER 1965.
C LAST REVISION BY J ERNST, OCT 1973.
C
C SUBROUTINE ARGUMENTS
C XA = INPUT VECTOR OF X-COORDINATES FOR ROWS OF YA. MUST BE IN
C INCREASING ORDER. SIZE(NXA).
C XZ = INPUT VECTOR OF X-COORDINATES FOR INTERPOLATED VALUES.
C SIZE(NXZ).
C YA = INPUT MATRIX OF Y-COORDINATES TO BE INTERPOLATED.
C SIZE(NXA,NCA).
C YZ = OUTPUT MATRIX OF INTERPOLATED Y-COORDINATES. SIZE(NXZ,NCA).
C EACH COLUMN OF YZ HAS INTERPOLATED VALUES OF THE
C RESPECTIVE COLUMN OF YA.
C NXA = INPUT NUMBER OF XA STATIONS, ROWS OF MATRIX YA.
C NXZ = INPUT NUMBER OF XZ STATIONS, ROWS OF MATRIX YZ.
C NCA = INPUT NUMBER OF COLUMN VECTORS IN MATRICES YA,YZ.
C KA = INPUT ROW DIMENSION OF YA IN CALLING PROGRAM.
C KZ = INPUT ROW DIMENSION OF YZ IN CALLING PROGRAM.
C
DO 30 K=1,NXZ
DO 10 I=1,NXA
IF(XZ(K).LE.XA(I+1) .OR. (I+1).EQ.NXA) GO TO 20
10 CONTINUE
20 DO 30 J=1,NCA
30 YZ(K,J) = YA(I,J) + (XZ(K)-XA(I))*(YA(I+1,J)-YA(I,J))/
* (XA(I+1)-XA(I))
C
RETURN
END

```



```

SUBROUTINE TERP2 (XA,XZ,YA,YZ,NXA,NXZ,NCA,KA,KZ)
DIMENSION XA(1),XZ(1),YA(KA,1),YZ(KZ,1)

```

```

C
C DIPARABOLIC INTERPOLATION.
C (PARABOLIC INTERPOLATION IN FIRST, LAST BAYS AND OUTSIDE XA).
C VALUES OF XZ MAY BE OUTSIDE OF XA. (EXTRAPOLATION).
C CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY RF HRUDA. FEBRUARY 1965.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C XA = INPUT VECTOR OF X-COORDINATES FOR ROWS OF YA. MUST BE IN
C INCREASING ORDER. SIZE(NXA).
C XZ = INPUT VECTOR OF X-COORDINATES FOR INTERPOLATED VALUES.
C SIZE(NXZ).
C YA = INPUT MATRIX OF Y-COORDINATES TO BE INTERPOLATED.
C SIZE(NXA,NCA).
C YZ = OUTPUT MATRIX OF INTERPOLATED Y-COORDINATES. SIZE(NXZ,NCA).
C EACH COLUMN OF YZ HAS INTERPOLATED VALUES OF THE
C RESPECTIVE COLUMN OF YA.
C NXA = INPUT NUMBER OF XA STATIONS, ROWS OF MATRIX YA.
C NXZ = INPUT NUMBER OF XZ STATIONS, ROWS OF MATRIX YZ.
C NCA = INPUT NUMBER OF COLUMN VECTORS IN MATRICES YA, YZ.
C KA = INPUT ROW DIMENSION OF YA IN CALLING PROGRAM.
C KZ = INPUT ROW DIMENSION OF YZ IN CALLING PROGRAM.

```

```

C ERROR EXPLANATION

```

```

C 1 = LESS THAN 3 STATIONS.

```

```

ERROR = 1

```

```

IF (NXA .LT. 3) GO TO 999

```

```

DO 400 K=1,NXZ
IF (XZ(K).LE.XA(2)) GO TO 100
IF (XZ(K).GE.XA(NXA-1)) GO TO 300
DO 50 I=3,NXA
IF (XZ(K).LE.XA(I)) GO TO 200
50 CONTINUE

```

```

C FIRST BAY OR LEFT EXTRAPOLATION.
100 BAYL = XA(2)-XA(1)
H = (XZ(K)-XA(1))/BAYL
D = (XA(3)-XA(1))/BAYL
DO 102 J=1,NCA
102 YZ(K,J) = YA(1,J)*(H**2-H*(1.0+D)+D)/D
* + YA(2,J)*(H**2-H*D)/(1.0-D)
* + YA(3,J)*(-H**2+H)/(D-D**2)
GO TO 400

```

```

C INTERIOR BAY.
200 BAYL = XA(I)-XA(I-1)
H = (XZ(K) -XA(I-1))/BAYL
C = (XA(I-2)-XA(I-1))/BAYL
D = (XA(I+1)-XA(I-1))/BAYL
DO 202 J=1,NCA

```

```

202 YZ(K,J)= YA(I-2,J)*(H**3-2.0*H**2+H)/(C-C**2)
*      + YA(I-1,J)*(H**3*(C-D)+H**2*(2.0*D-C)-H*(D+C*D)+C*D)/(C*D)
*      + YA(I  ,J)*(H**3*(D-C)+H**2*(1.0-2.0*D+C)-H*C*(1.0-D))/
*                                          ((1.0-C)*(1.0-D))
*      + YA(I+1,J)*(-H**3+H**2)/(D-D**2)
GO TO 400

```

C  
C LAST BAY OR RIGHT EXTRAPOLATION.

```

300 BAYL = XA(NXA)-XA(NXA-1)
H = (XZ(K) -XA(NXA-1))/BAYL
C = (XA(NXA-2)-XA(NXA-1))/BAYL
DO 302 J=1,NCA

```

```

302 YZ(K,J)= YA(NXA-2,J)*(-H**2+H)/(C-C**2)
*      + YA(NXA-1,J)*(H**2-H*(1.0+C)+C)/C
*      + YA(NXA  ,J)*(H**2-H*C)/(1.0-C)

```

C  
400 CONTINUE  
RETURN

C  
999 CALL ZZBOMB (6HTERP2 ,NERROR)  
END

```

SUBROUTINE TIMCHK(NAMCHK)
COMMON /LTIME / NTM,NOCK(10),NAM(10),TOT(10),TLAS(10)
COMMON /LLINE / NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/
DATA KT/10/

```

C THE PURPOSE OF THIS ROUTINE IS TO DETERMINE THE ELAPSED CPU TIME  
C BETWEEN DEFINED POINTS IN A PROGRAM. UPTO 10 DEFINED TIME CHECKS  
C CAN BE OBTAINED USING THIS ROUTINE. THE ROUTINE IS USED BY CALLING  
C IT AT THE POINT IN THE PROGRAM WHERE THE TIME CHECK IS TO START AND  
C THEN CALLING IT AGAIN AT THE POINT WHERE THE TIME CHECK IS TO END.  
C THE TIME CHECK IS IDENTIFIED BY THE ARGUMENT NAMCHK AS AN A6 VARIABLE  
C (IE NAMCHK=6HTIME 1). IF MORE THAN ONE TIME CHECK IS MADE USING THE  
C SAME NAME FOR THE ARGUMENT NAMCHK THE SUM OF THE ELAPSED TIMES WILL BE  
C RECORDED.

C BEFORE THIS ROUTINE CAN BE USED IT IS NECESSARY TO INITIALIZE IT.  
C THIS DONE BY CALLING IT WITH NAMCHK=6HTBEGIN.

C THE RESULTS OF THE TIME CHECKS MADE BY THIS ROUTINE ARE PRINTED  
C BY CALLING IT WITH NAMCHK=6HTPRINT. THE DATA PRINTED CONSISTS OF  
C A TABLE CONTAINING THE NAMES OF THE TIME CHECKS, THE TOTAL ELAPSED  
C CPU TIME FOR EACH NAMED TIME CHECK, THE NUMBER OF TIMES EACH NAMED  
C TIME CHECK WAS MADE, AND THE AVERAGE CPU TIME FOR EACH NAMED TIME  
C CHECK (IE TOTAL ELAPSED TIME DIVIDED BY THE NUMBER TIMES THE TIME  
C CHECK WAS MADE).

ARGUMENT

C NAMCHK - INPUT (A6 FORMAT) TIME CHECK IDENTIFICATION  
C IF(NAMCHK=6HTBEGIN) ROUTINE IS INITIALIZED  
C IF(NAMCHK=6HTPRINT) RESULTS ARE PRINTED

C FORMA SUBROUTINES CALLED ARE PAGEHD AND ZZBOMB.

C CODED BY JOHN ADMIRE \*NASA\* AUG 1973.

```

2000 FORMAT(/1X123(1H-))
2010 FORMAT(//51X24(1H*)/51X24H* CPU TIME CHECK TABLE *,
*/37X50(1H*)/37X,
*50H* NAME OF * TOTAL * NUMBER OF * AVERAGE */37X,
*50H* TIME CHECK * CPU TIME * CHECKS MADE * CPU TIME */37X50(1H*))
2020 FORMAT(37X,
*4H* ,A6,5H * ,F8.2,2H *5X,I3,5X2H* ,F8.2,2H *
*/37X50(1H*))
IF(NAMCHK .EQ. 6HTBEGIN) GO TO 60
IF(NAMCHK .EQ. 6HTPRINT) GO TO 80
L=NTM+1
CALL CPUTIM(ICTIM)
C1=ICTIM
C1=C1/1000000.
IF('NTM .NE. 0) GO TO 30
10 NERROR=]
IF(L .GT. KT) GO TO 999
NAM(L)=NAMCHK
NTM=L

```

```
20 TLAS(L)=C1
   RETURN
30 DO 40 I=1,NTM
   LL=I
   IF(NAMCHK .EQ. NAM(I)) GO TO 50
40 CONTINUE
   GO TO 10
50 L=LL
   IF(TLAS(L) .LT. 0.) GO TO 20
   TOTL(L)=TOTL(L)+(C1-TLAS(L))
   TLAS(L)=-1.
   NOCK(L)=NOCK(L)+1
   RETURN
60 NTM=0
   DO 70 I=1,KT
   NOCK(I)=0
   NAM(I)=6H
   TOTL(I)=0.
70 TLAS(I)=-1.
   RETURN
80 IF(MINI .NE. 4HMINI) GO TO 90
   IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 90
   NN=10+NTM*2
   IF(NLINE+NN .GT. MAXLIN) GO TO 90
   WRITE(NOT,2000)
   NLINE=NLINE+2
   GO TO 100
90 CALL PAGEHD
100 WRITE(NOT,2010)
   NLINE=NLINE+8
   NERROR=2
   IF(NTM .EQ. 0) GO TO 999
   DO 120 I=1,NTM
   IF(NLINE+2 .LE. MAXLIN) GO TO 110
   CALL PAGEHD
   WRITE(NOT,2010)
   NLINE=NLINE+8
110 C2=NOCK(I)
   C1=TOTL(I)/C2
   WRITE(NOT,2020) NAM(I),TOTL(I),NOCK(I),C1
120 NLINE=NLINE+2
   RETURN
999 CALL ZZBOMB(6HTIMCHK,NERROR)
   END
```

```

SUBROUTINE TRAE2 (IXRUNO,IXNAME,IFA,A,IFB,B,IFC,C,IFD,D,IFE,E,
*           ZTMM,STARTT,ENDT,MLTXTP,NWRITE,ZIDENT,STA,
*           ZNAME,NZ,KZ,NXTAPE,NZTAPE,STOREZ)
DIMENSION A(KZ,1),B(KZ,1),C(KZ,1),D(KZ,1),E(1),ZTMM(KZ,1),STA(1),
*           ZIDENT(1),STOREZ(KZ,1)
DIMENSION STORET(6)
COMMON /LWRKV1/ XDD(250), XD(250)
COMMON /LWRKV2/ X(250), Z(250)
COMMON /LWRKV3/ F(500)
COMMON /LSTART/ IZRUNO,ZDATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.D/
DATA NIT,NOT/5,6/
DATA NLPP,BUF, EOT/
*           54 , 0.,3HEOT/

```

```

C
C SOLVE THE MATRIX EQUATION
C  $Z(T) = A \cdot XDD(T) + B \cdot XD(T) + C \cdot X(T) + D \cdot F(T) + E$ 
C THAT IS, THE ADDITIONAL EQUATIONS TO GET SHEAR, BENDING MOMENT, ETC.
C T,XDD,XD,X,F ARE OBTAINED FROM NXTAPE (OUTPUT OF TIME RESP SUBRT).
C NXTAPE IS POSITIONED BY SEARCHING FOR RUN NUMBER (IXRUNO) AND
C NAME (IXNAME).
C THE ANSWER Z(T) WILL BE WRITTEN ON NZTAPE EVERY MLTXTP*XDELTA (OF
C TIME RESP) AND ON PAPER EVERY NWRITE*(MLTXTP*XDELTA).
C NZTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NZTAPE HERE.
C THE MAXIMUM AND MINIMUM VALUES OF Z WILL BE DETERMINED AND OUTPUT
C THRU MATRIX ZTMM.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F,Z)
C   NX = 250
C   NF = 500
C   NZ = 250
C CODED BY RL WOHLFN. MARCH 1965.
C LAST REVISION BY RL WOHLFN. MARCH 1976.

```

```

C SUBROUTINE ARGUMENTS
C IXRUNO = INPUT RUN NUMBER OF TIME RESPONSE DATA TO BE READ FROM
C           NXTAPE. (A6 FORMAT).
C IXNAME = INPUT IDENTIFICATION OF TIME RESPONSE DATA TO BE READ FROM
C           NXTAPE. (A6 FORMAT).
C IFA = INPUT A NEED NOT BE DIMENSIONED IN MAIN PROG FOR IFA = 0
C A = INPUT MATRIX COEFFICIENT OF XDD. SIZE (NZ,NX).
C IFB = INPUT B NEED NOT BE DIMENSIONED IN MAIN PROG FOR IFB = 0
C B = INPUT MATRIX COEFFICIENT OF XD. SIZE (NZ,NX).
C IFC = INPUT C NEED NOT BE DIMENSIONED IN MAIN PROG FOR IFC = 0
C C = INPUT MATRIX COEFFICIENT OF X. SIZE (NZ,NX).
C IFD = INPUT D NEED NOT BE DIMENSIONED IN MAIN PROG FOR IFD = 0
C D = INPUT MATRIX COEFFICIENT OF F. SIZE (NZ,NF).
C IFE = INPUT E NEED NOT BE DIMENSIONED IN MAIN PROG FOR IFE = 0
C E = INPUT VECTOR. SIZE (NZ).
C ZTMM = OUTPUT MATRIX OF Z MAX, MIN AND TIMES. SIZE (NZ,4).

```

COL 1 = Z MAX,      COL 2 = TIME AT Z MAX,  
 COL 3 = Z MIN,      COL 4 = TIME AT Z MIN.  
 C STARTT = INPUT    START TIME FOR ADDITIONAL EQUATIONS. MAY BE GREATER  
 C                    THAN START TIME USED IN TIME RESPONSE. IF LESS,  
 C                    TIME RESPONSE START TIME IS USED.  
 C ENDT    = INPUT    END TIME FOR ADDITIONAL EQUATIONS. MAY BE LESS  
 C                    THAN END TIME USED IN TIME RESPONSE. IF GREATER,  
 C                    TIME RESPONSE END TIME IS USED.  
 C MLTXTP = INPUT    MULTIPLE OF TIME RESPONSE POINTS TO USE FOR ADD. EQS.  
 C                    MLTXTP = 1    USE EVERY TIME RESP POINT (1,2,3,...)  
 C                    MLTXTP = 2    USE EVERY SECOND TIME RESP POINT (1,3,5,...)  
 C                    ETC  
 C NWRITE = INPUT    MULTIPLE OF ADDITIONAL EQS POINTS TO WRITE ON PAPER.  
 C                    NWRITE = 1    WRITE EVERY POINT (1,2,3,...)  
 C                    NWRITE = 2    WRITE EVERY SECOND POINT (1,3,5,...)  
 C                    ETC  
 C ZIDENT = INPUT    HEADING FOR Z IN PRINTED OUTPUT. (12A6 FORMAT).  
 C STA     = INPUT    STATIONS FOR ROWS OF Z IN PRINTED OUTPUT. SIZE (NZ).  
 C                    (A6 FORMAT).  
 C ZNAME   = INPUT    IDENTIFICATION OF ADDITIONAL EQS DATA TO BE WRITTEN  
 C                    ON NZTAPE. (A6 FORMAT).  
 C NZ      = INPUT    NUMBER OF ROWS IN A,B,C,D,E,ZTMM,STOREZ. MAX=250.  
 C KZ      = INPUT    ROW DIMENSION OF A,B,C,D,ZTMM,STOREZ IN CALLING PROG.  
 C NXTAPE = INPUT    NUMBER OF TAPE FROM WHICH T,XDD,XD,X,F WILL BE READ.  
 C                    (EG 1).  
 C NZTAPE = INPUT    NUMBER OF TAPE ON WHICH T,Z WILL BE WRITTEN. (EG 10).  
 C                    IF NZTAPE = 0, BYPASS WRITING Z ON NZTAPE.  
 C STOREZ = INPUT    WORKSPACE MATRIX TO STORE SIX COLUMNS OF Z FOR  
 C                    PRINTING. SIZE (NZ,6).

C                    THE OUTPUT DATA (TO BE WRITTEN ON NZTAPE AND PAPER) IS  
 C T        = TIME  
 C Z        = SHEAR, BENDING MOMENT, ETC.    SIZE(NZ).

C                    NERROR    EXPLANATION  
 C 1 = SIZE EXCEEDANCE.  
 C 2 = X AND Z DATA CANNOT SHARE SAME TAPE.  
 C 3 = REQUESTED RUN NUMBER OR NAME CANNOT BE FOUND.  
 C 4 = NX OR NF EXCEEDS ALLOWABLE SIZE.

C 2010 FORMAT (/ 15X, 12A6 // 9X,6HTIME = F14.6,5F17.6)  
 C 2011 FORMAT (2X,3HROW, 3X,7HSTATION)  
 C 2012 FORMAT (I5,4X,A6,6E17.8)  
 C 2050 FORMAT (/ 1X 123(1H-))

C                    NERROR=1  
 C                    IF (NZ .GT. 250) GO TO 999  
 C                    NERROR=2  
 C                    IF (NXTAPE .EQ. NZTAPE) GO TO 999

C SEARCH NXTAPE FOR CORRECT HEADING.  
 C                    REWIND NXTAPE  
 C                    2 READ (NXTAPE) ITRUND,ITNAME,IFOTCK,XSTART,XDELTA,XEND,NX,NF,NXTP  
 C                    IF (ITRUND.EQ.IXRUND .AND. ITNAME.EQ.IXNAME) GO TO 5  
 C                    NERROR=3

```

IF (IEOTCK .EQ. 3HEOT) GO TO 999
DO 3 IXTP=1,NXTP
3 READ (NXTAPE)
GO TO 2
5
IF (NX .GT. 250 .OR. NF .GT. 500) GO TO 999
ZSTART = STARTT
ZEND = ENDT
IF (ZSTART .LT. XSTART) ZSTART = XSTART
IF (ZEND .GT. XEND ) ZEND = XEND
HXDEL = .5*XDELTA

```

NERROR=4

```

C
C FIND X-TIME POINT NUMBER FOR ZSTART.
DO 6 IXTP=1,NXTP
XTIME = XSTART + FLOAT(IXTP-1)*XDELTA
IF (ABS(ZSTART-XTIME) .LE. HXDEL) GO TO 7
6 CONTINUE
7 IXTPZS = IXTP
ZSTART = XTIME
C
C FIND X-TIME POINT NUMBER FOR ZEND.
IZTP = 1
8 IXTP = IXTP + MLTXTP
XTIME = XSTART + FLOAT(IXTP-1)*XDELTA
IZTP = IZTP+1
IF (XTIME .LE. (ZEND+HXDEL)) GO TO 8
IXTPZE = IXTP-MLTXTP
ZEND = XSTART + FLOAT(IXTPZE-1)*XDELTA
NZTP = IZTP-1
ZDELTA = FLOAT(MLTXTP)*XDELTA
C
C SKIP RECORDS ON NXTAPE UP TO X-TIME POINT NUMBER FOR ZSTART.
IF (IXTPZS .EQ. 1) GO TO 10
IXZSM1 = IXTPZS-1
DO 9 I=1,IXZSM1
9 READ (NXTAPE)
C
C SEARCH NZTAPE FOR END OF WRITTEN DATA.
10 IF (NZTAPE .LE. 0) GO TO 20
REWIND NZTAPE
15 READ (NZTAPE) BUFIN,BUFIN,IFOTCK,(BUFIN,I=1,4),NREC
IF (IEOTCK .EQ. 3HEOT) GO TO 17
DO 16 IREC=1,NKEC
16 READ (NZTAPE)
GO TO 15
17 BACKSPACE NZTAPE
WRITE (NZTAPE) IZRUND,ZNAME,ZDATE,ZSTART,ZDELTA,ZEND,NZ,NZTP,
* (BUF,I=1,11),(ZIDENT(I),I=1,12),(STA(I),I=1,NZ)
C
C ADDITIONAL EQUATIONS LOOP.
20 LXTP = MLTXTP-1
DO 399 IXTP=IXTPZS,IXTPZE
LXTP = LXTP+1
IF (LXTP .EQ. MLTXTP) GO TO 25
READ (NXTAPE)

```

```

GO TO 399
25 READ (NXTAPE) T, (F(J),J=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
*      (X(I),I=1,NX)
    LXTP = 0
    DO 35 I=1,NZ
35  Z(I) = 0.
    IF (IFA .EQ. 0) GO TO 50
    DO 45 I=1,NZ
    SS = ZERO
    DO 44 J=1,NX
    S = A(I,J)*XDD(J)
44  SS = SS + S
45  Z(I) = Z(I) + SS
50  IF (IFB .EQ. 0) GO TO 60
    DO 55 I=1,NZ
    SS = ZERO
    DO 54 J=1,NX
    S = B(I,J)*XD(J)
54  SS = SS + S
55  Z(I) = Z(I) + SS
60  IF (IFC .EQ. 0) GO TO 70
    DO 65 I=1,NZ
    SS = ZERO
    DO 64 J=1,NX
    S = C(I,J)*X(J)
64  SS = SS + S
65  Z(I) = Z(I) + SS
70  IF (IFD .EQ. 0) GO TO 80
    DO 75 I=1,NZ
    SS = ZERO
    DO 74 J=1,NF
    S = D(I,J)*F(J)
74  SS = SS + S
75  Z(I) = Z(I) + SS
80  IF (IFE .EQ. 0) GO TO 100
    DO 85 I=1,NZ
85  Z(I) = Z(I) + E(I)
C
C  WRITE T,Z ON NZTAPE FOR LATER USE.
100 IF (NZTAPE .GT. 0) WRITE (NZTAPE) T,(Z(I),I=1,NZ)
C
C  CALCULATE MAXIMUM AND MINIMUMS OF Z. PLACE IN ZTMM.
C      COL 1 = Z MAX,      COL 2 = TIME OF Z MAX,
C      COL 3 = Z MIN,      COL 4 = TIME OF Z MIN.
C
    IF (IXTP .GT. IXTPZS) GO TO 150
    DO 110 I=1,NZ
    ZTMM(I,1) = Z(I)
    ZTMM(I,2) = T
    ZTMM(I,3) = Z(I)
110  ZTMM(I,4) = T
    NCOL = 0
    GO TO 190
150 DO 155 I=1,NZ
    IF (Z(I) .LE. ZTMM(I,1)) GO TO 152
    ZTMM(I,1) = Z(I)

```



```

      ZTMM(I,2) = T
152 IF (Z(I) .GE. ZTMM(I,3)) GO TO 155
      ZTMM(I,3) = Z(I)
      ZTMM(I,4) = T
155 CONTINUE
C
C SEE IF DATA SHOULD BE PRINTED.
C COLLECT SIX COLUMNS OF T AND Z BEFORE PRINTING.
      IF (IXTP.LT.IXTPZE .AND. NW.LT.NWRITE) GO TO 210
190 NCOL = NCOL + 1
      STORET(NCOL) = T
      DO 200 I=1,NZ
200 STOREZ(I,NCOL) = Z(I)
      NW = 0
210 NW = NW + 1
      IF (IXTP.LT.IXTPZE .AND. NCOL.LT.6) GO TO 399
      NZE = 0
381 NZS = NZE + 1
      NZE = NZ
      IF ((NZE-NZS) .GT. (NLPP-11)) NZE=NZS+(NLPP-11)
      IF (MINI .NE. 4HMINI) GO TO 800
      IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
      IF ((NLINE+2+5+NZ) .GT. MAXLIN) GO TO 800
      WRITE (NOT,2050)
      NLINE = NLINE + 2
      GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2010) (ZIDENT(I), I=1,12), (STORET(I), I=1,NCOL)
      WRITE (NOT,2011)
      NLINE = NLINE + 5
      DO 387 I=NZS,NZE
      NLINE = NLINE + 1
387 WRITE (NOT,2012) I,STA(I),(STOREZ(I,J),J=1,NCOL)
      IF (NZ .GT. NZE) GO TO 381
      NCCL = 0
399 CONTINUE
C
      IF (NZTAPE .LE. 0) RETURN
      WRITE (NZTAPE) BUF,BUF,EOT,(BUF,I=1,16)
      ENDFILE NZTAPE
      RETURN
C
999 CALL ZZBOMB (6HTRAE2 ,NERROR)
      END

```

TRANS

```
SUBROUTINE TRANS (A,Z,NRA,NCA,KRA,KRZ)  
DIMENSION A(KRA,1), Z(KRZ,1)
```

```
C  
C TRANSPOSE MATRIX A INTO MATRIX Z.  
C CODED BY RL WOHLER. FEB 1965.
```

```
C  
C SUBROUTINE ARGUMENTS  
C A = INPUT MATRIX. SIZE(NRA,NCA).  
C Z = OUTPUT RESULT MATRIX. SIZE(NCA,NRA).  
C NRA = INPUT NUMBER OF ROWS OF MATRIX A, COLS OF MATRIX Z.  
C NCA = INPUT NUMBER OF COLS OF MATRIX A, ROWS OF MATRIX Z.  
C KRA = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.  
C KRZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.
```

```
C  
DO 10 I=1,NRA  
DO 10 J=1,NCA  
10 Z(J,I) = A(I,J)  
RETURN  
END
```

```

SUBROUTINE TRMM (IXRUNO,IXNAME,XTMM,STARTT,ENDT,NX,KX,NXTAPE)
DIMENSION XTMM(KX,1)
COMMON /LWRKV1/ XDD(250), XD(250)
COMMON /LWRKV2/ X(500)

```

```

C
C FIND XDD, XD, X MAXIMUMS, MINIMUMS, AND TIME OF OCCURRENCE FROM TIME
C RESPONSE TAPE.
C NXTAPE IS POSITIONED BY SEARCHING FOR RUN NUMBER (IXRUNO) AND
C NAME (IXNAME).
C THE MAXIMUM SIZE IS (BASED ON DIMENSIONS OF XDD,XD,X)
C   NX = 250
C DEVELOPED BY RL WOHLER. NOVEMBER 1975.
C LAST REVISION BY WA BENFIELD. MARCH 1976.

```

#### SUBROUTINE ARGUMENTS

```

C IXRUNO = INPUT  RUN NUMBER OF TIME RESPONSE DATA TO BE READ FROM
C              NXTAPE. (A6 FORMAT).
C IXNAME = INPUT  IDENTIFICATION OF TIME RESPONSE DATA TO BE READ FROM
C              NXTAPE. (A6 FORMAT).
C XTMM    = OUTPUT MATRIX OF MAX, MINS AND TIMES. SIZE(NX,12).
C              COL 1 = XDD MAX      COL 2 = TIME AT XDD MAX
C              COL 3 = XDD MIN      COL 4 = TIME AT XDD MIN
C              COL 5 = XD MAX       COL 6 = TIME AT XD MAX
C              COL 7 = XD MIN       COL 8 = TIME AT XD MIN
C              COL 9 = X MAX        COL 10 = TIME AT X MAX
C              COL 11 = X MIN       COL 12 = TIME AT X MIN
C STARTT = INPUT  START TIME FOR MAXIMUM, MINIMUMS. MAY BE GREATER
C              THAN START TIME USED IN TIME RESPONSE. IF LESS,
C              TIME RESPONSE START TIME IS USED.
C ENDT    = INPUT  END TIME FOR MAXIMUM, MINIMUMS. MAY BE LESS
C              THAN END TIME USED IN TIME RESPONSE. IF GREATER,
C              TIME RESPONSE END TIME IS USED.
C NX      = OUTPUT NUMBER OF ROWS IN MATRIX XTMM. MAX=250.
C KX      = INPUT  ROW DIMENSION OF XTMM IN CALLING PROGRAM.
C NXTAPE = INPUT  NUMBER OF TAPE FROM WHICH T,XDD,XD,X WILL BE READ.
C              (EG 1).

```

#### NEPROR EXPLANATIONS

```

C 1 = REQUESTED RUN NUMBER OR NAME CANNOT BE FOUND.
C 2 = SIZE EXCEEDANCE.
C
C SEARCH NXTAPE FOR CORRECT HEADING.
C   REWIND NXTAPE
C 2 READ (NXTAPE) ITRUNO,ITNAME,IEOTCK,XSTART,XDELTA,XEND,NX,NF,NXTP
C   IF (ITRUNO.EQ.IXRUNO .AND. ITNAME.EQ.IXNAME) GO TO 5
C
C   NERROR=1
C   IF (IEOTCK .EQ. 3HECT) GO TO 999
C   DO 3 IOTP=1,NXTP
C 3 READ (NXTAPE)
C   GO TO 2
C
C 5
C   NERROR=2
C   IF (NX .GT. 250) GO TO 999
C   STRMM = STARTT
C   ENDMM = FNDT
C   IF (STRMM .LT. XSTART) STRMM = XSTART

```

```

IF (ENDMM .GT. XEND ) ENDMM = XEND
HXDEL = .5*XDELTA
C
C FIND X-TIME POINT NUMBER FOR MAX-MIN START.
DO 6 IXTP=1,NXTP
  XTIME = XSTART + FLOAT(IXTP-1)*XDELTA
  IF (ABS(STRMM-XTIME) .LE. HXDEL) GO TO 7
6 CONTINUE
7 IXTPS = IXTP
C
C FIND X-TIME POINT NUMBER FOR MAX-MIN END.
8 IXTP = IXTP + 1
  XTIME = XSTART + FLOAT(IXTP-1)*XDELTA
  IF (XTIME .LE. (ENDMM+HXDEL)) GO TO 8
  IXTP = IXTP - 1
C
C SKIP RECORDS ON NXTAPE UP TO X-TIME POINT NUMBER FOR MAX-MIN START.
IF (IXTPS .EQ. 1) GO TO 100
IXTSM1 = IXTPS-1
DO 9 I=1,IXTSM1
9 READ (NXTAPE)
C
C FIND MAXIMUMS AND MINIMUMS. PLACE IN XTMM.
COL 1 = XDD MAX      COL 2 = TIME AT XDD MAX
COL 3 = XDD MIN      COL 4 = TIME AT XDD MIN
COL 5 = XD MAX       COL 6 = TIME AT XD MAX
COL 7 = XD MIN       COL 8 = TIME AT XD MIN
COL 9 = X MAX        COL 10 = TIME AT X MAX
COL 11 = X MIN       COL 12 = TIME AT X MIN
100 DO 399 IXTP=IXTPS,IXTPE
  READ (NXTAPE) T, (DUM,J=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
  * (X(I),I=1,NX)
  IF (IXTP .GT. IXTPS) GO TO 200
  DO 110 I=1,NX
    XTMM(I, 1) = XDD(I)
    XTMM(I, 2) = T
    XTMM(I, 3) = XDD(I)
    XTMM(I, 4) = T
    XTMM(I, 5) = XD(I)
    XTMM(I, 6) = T
    XTMM(I, 7) = XD(I)
    XTMM(I, 8) = T
    XTMM(I, 9) = X(I)
    XTMM(I,10) = T
    XTMM(I,11) = X(I)
110 XTMM(I,12) = T
  GO TO 399
200 DO 290 I=1,NX
  IF (XDD(I) .LE. XTMM(I,1)) GO TO 215
  XTMM(I, 1) = XDD(I)
  XTMM(I, 2) = T
215 IF (XDD(I) .GE. XTMM(I,3)) GO TO 220
  XTMM(I, 3) = XDD(I)
  XTMM(I, 4) = T
220 IF (XD(I) .LE. XTMM(I,5)) GO TO 225

```

```
      XTMM(I, 5) = XD(I)
      XTMM(I, 6) = T
225  IF (XD (I) .GE. XTMM(I,7)) GO TO 230
      XTMM(I, 7) = XD(I)
      XTMM(I, 8) = T
230  IF (X  (I) .LE. XTMM(I,9)) GO TO 235
      XTMM(I, 9) = X(I)
      XTMM(I,10) = T
235  IF (X  (I) .GE. XTMM(I,11))GO TO 290
      XTMM(I,11) = X(I)
      XTMM(I,12) = T
290  CONTINUE
C
399  CONTINUE
C
      RETURN
C
999  CALL ZZBOMB (6HTRMM ,NERROR)
      END
```

```

SUBROUTINE TRPSD (IXPUNO,IXNAME,IRAE,IEXP,STARTT,MLTXTP,ZPSD,
*              NFREQ,TIMPER,NXTAPE,WRKV)
DIMENSION ZPSD(1), WRKV(1)
COMMON /LWRKV1/ X(500)

```

```

C
C CALCULATE PSD OF ONE ROW OF TIME RESPONSE ADDITIONAL EQUATIONS
C (ADD EQ) DATA FROM SUBROUTINE TRAE2. DEFINE X = ADD EQ, Z = PSD.
C X IS OBTAINED FROM NXTAPE (OUTPUT OF ADD EQ SUBROUTINE TRAE2).
C NXTAPE IS POSITIONED BY SEARCHING FOR RUN NUMBER (IXRUNO) AND
C NAME (IXNAME).
C CALLS FORMA SUBROUTINE ZZBOMB.
C THE MAXIMUM SIZE IS (BASED ON DIMENSION OF X)
C   NX = 500
C CODED BY RL WOHLN. JANUARY 1976.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C IXRUNG = INPUT  RUN NUMBER OF ADD EQ DATA TO BE READ FROM NXTAPE.
C              (A6 FORMAT).
C IXNAME = INPUT  IDENTIFICATION OF ADD EQ DATA TO BE READ FROM
C              NXTAPE. (A6 FORMAT).
C IRAE   = INPUT  ROW NUMBER OF ADD EQ USED IN PSD CALCULATION.
C IEXP   = INPUT  EXPONENT OF 2. GIVES NUMBER OF TIME POINTS USED
C              IN PSD CALCULATION. NZTP=2**IEXP. MAX IEXP=13.
C              EG, IEXP = 5, 10, 11, 12, 13.
C              NZTP = 32, 1024, 2048, 4096, 8192.
C STARTT = INPUT  START TIME FOR PSD CALCULATION. MAY BE GREATER THAN
C              START TIME USED IN ADD EQ. IF LESS, ADD EQ START TIME
C              IS USED.
C MLTXTP = INPUT  MULTIPLE OF ADD EQ POINTS TO USE FOR PSD CALCULATION.
C              MLTXTP = 1  USE EVERY ADD EQ POINT (1,2,3,...)
C              MLTXTP = 2  USE EVERY SECOND ADD EQ POINT (1,3,5,...)
C              ETC
C ZPSD   = INPUT  WORKSPACE VECTOR. MUST BE DIMENSIONED AT LEAST 2*NZTP
C              WHERE NZTP=2**IEXP.
C              = OUTPUT VECTOR OF PSDS AT VARIOUS FREQUENCIES FOR ROW IRAE
C              OF ADD EQ.
C              ZPSD(1) AT FREQ=0
C              ZPSD(2) AT FREQ=1/DATA TIME PERIOD
C              ZPSD(3) AT FREQ=2/DATA TIME PERIOD
C              ETC
C              ZPSD(NZTP/2) AT FREQ=(NZTP/2-1)/DATA TIME PERIOD.
C NFREQ  = OUTPUT NUMBER OF FREQUENCIES AT WHICH PSD IS CALCULATED.
C              NFREQ = NZTP/2 WHERE NZTP=2**IEXP.
C TIMPER = OUTPUT TIME PERIOD OF DATA USED FOR PSD CALCULATION.
C NXTAPE = INPUT  NUMBER OF TAPE FROM WHICH X WILL BE READ. (EG 1).
C WRKV   = INPUT  WORK VECTOR. DIMENSION AT LEAST NZTP/2-1 IN CALLING
C              PROGRAM.
C
C ERROR EXPLANATION
C 1 = MAXIMUM ALLOWABLE EXPONENT SIZE EXCEEDED.
C 2 = REQUESTED RUN NUMBER OR NAME CANNOT BE FOUND.
C 3 = SIZE EXCEEDANCE.
C
C DEFINITION... X IS ADDITIONAL EQUATIONS, Z IS PSD.

```

NERROR=1

```

IF (IEXP .GT. 13) GO TO 999
NZTP = 2**IEXP
NXTP = NZTP + (NZTP-1)*(MLTXTP-1)

```

```

C
C SEARCH NXTAPE FOR CORRECT HEADING.

```

```

REWIND NXTAPE

```

```

2 READ (NXTAPE) ITRUND,ITNAME,IFOTCK,XSTART,XDELTA,XEND,NX,NXREC
IF (ITRUND.EQ.IXRUND .AND. ITNAME.EQ.IXNAME) GO TO 5

```

NERROR=2

```

IF (IEOTCK .EQ. 3HEOT) GO TO 999
DO 3 IXREC=1,NXREC

```

```

3 READ (NXTAPE)
GO TO 2

```

```

5
IF (NX .GT. 500) GO TO 999
HXDEL = .5*XDELTA

```

NERROR=3

```

C
C FIND X-TIME POINT NUMBER FOR ZSTART.

```

```

ZSTART = STARTT
IF (ZSTART .LT. XSTART) ZSTART = XSTART
DO 6 IXTP=1,NXTP
XTIME = XSTART + FLOAT(IXTP-1)*XDELTA
IF (ABS(ZSTART-XTIME) .LE. HXDEL) GO TO 7

```

```

6 CONTINUE

```

```

7 IXTPZS = IXTP
ZSTART = XTIME

```

```

C
C FIND X-TIME POINT NUMBER FOR ZEND.

```

```

ZEND = ZSTART + FLOAT(NXTP-1)*XDELTA

```

NERROR=4

```

IF (ZEND .GT. XEND) GO TO 999
IXTPZE = IXTPZS + NXTP - 1

```

```

C
C SKIP RECORDS ON NXTAPE UP TO X-TIME POINT NUMBER FOR ZSTART.

```

```

IF (IXTPZS .EQ. 1) GO TO 10
IXZSM1 = IXTPZS-1
DO 9 I=1,IXZSM1
9 READ (NXTAPE)

```

```

C
C READ ADDITIONAL EQUATIONS DATA.

```

```

10 SUM = 0.0
IZTP = 0
LXTP = MLTXTP-1
DO 399 IXTP=IXTPZS,IXTPZE
LXTP = LXTP+1
IF (LXTP .EQ. MLTXTP) GO TO 25
READ (NXTAPE)
GO TO 399
25 READ (NXTAPE) T, (X(I),I=1,NX)
LXTP = 0
SUM = SUM + X(IRAE)
IZTP = IZTP+1
ZPSD(IZTP) = X(IRAE)
IZTP = IZTP+1

```

```
      ZPSD(IZTP) = 0.0
399 CONTINUE
C
C SUBTRACT AVERAGE VALUE FROM ORIGINAL DATA.
  AVRG = SUM/FLOAT(NZTP)
  NZTP2 = 2*NZTP
  DO 510 IZ =1,NZTP2,2
510 ZPSD(IZ) = ZPSD(IZ) - AVRG
C CALCULATE FOURIER COEFFICIENTS.
  CALL FORT (ZPSD, IEXP,WRKV,-1,IERR)
C FORM PSD VALUES.
  TIMPER = ZEND-ZSTART
  TWOPER = 2.*TIMPER
  NFREQ = NZTP/2
  DO 520 I=1,NFREQ
  I2 = 2*I
520 ZPSD(I) = TWOPER*(ZPSD(I2-1)**2 + ZPSD(I2)**2)
  RETURN
C
999 CALL ZZBOMB (6HTRPSD ,NERROR)
  END
```



```

SUBROUTINE TRSP1 (A,B,C,D,TABT,TABF,XDO,XO,STARTT,DELTAT,ENDT,
*           NWRITE,NX,NF,NTF,XNAME,KA,KF,NTAPE,NUT1)
DIMENSION A(KA,1),B(KA,1),C(KA,1),D(KA,1),TABT(KF,1),TABF(KF,1),
*           XDO(1),XO(1)
DIMENSION P(4)
COMMON /LWRKV1/ XDD(250),XD(250)
COMMON /LWRKV2/ QD(250),Q(250)
COMMON /LWRKV3/ X(250),AIDF(250)
COMMON /LWRKV4/ F(500)
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.D/
DATA NIT,NOT/5,6/
DATA NLPP,BUF,DIVTOL, EOT/
*           54 , 0.,1.E+35,3HEOT/

```

```

C
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C (A)XDD + (B)XD + (C)X = (D)F FOR XDD, XD, X.
C FOURTH ORDER RUNGE-KUTTA (GILL MODIFICATION) NUMERICAL INTEGRATION
C IS USED.
C VECTOR F IS OBTAINED BY LINEAR INTERPOLATION USING TABT,TABF.
C MATRICES A,B,C,D SHOULD NOT SHARE SAME CORE LOCATION (DUE TO MULTB).
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES INVI,MULTB,PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F)
C NX = 250
C NF = 500
C CODED BY RL WOHLER. MARCH 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX COEFFICIENT OF XDD. SIZE (NX,NX). * DESTROYED *
C B = MATRIX COEFFICIENT OF XD. SIZE (NX,NX). * DESTROYED *
C C = MATRIX COEFFICIENT OF X. SIZE (NX,NX). * DESTROYED *
C D = MATRIX COEFFICIENT OF F. SIZE (NX,NF). * DESTROYED *
C TABT = TABLE OF TIMES FOR FORCE IN TABF. SIZE (NF,NTF).
C TABF = TABLE OF FORCES. SIZE (NF,NTF).
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX).
C XO = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX).
C STARTT= START TIME.
C DELTAT= INTEGRATION STEP SIZE.
C ENDT = END TIME.
C NWRITE= MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C NWRITE = 1 WRITE EVERY POINT (1,2,3,...)
C NWRITE = 2 WRITE EVERY SECOND POINT (1,3,5,...)
C FTC
C NX = SIZE OF MATRICES A,B,C (SQUARE). NUMBER OF ROWS IN D. MAX=250.
C NF = NUMBER OF ROWS IN TABT,TABF. NUMBER OF COLS IN D. MAX=500.
C NTF = NUMBER OF COLS IN TABT,TABF.

```

```

C XNAME = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).
C KA   = ROW DIMENSION OF A,B,C,D IN CALLING PROGRAM.
C KF   = ROW DIMENSION OF TABT,TABF IN CALLING PROGRAM.
C NTAPE = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C NUT1  = NUMBER OF THE UTILITY TAPE. (E.G. 4).

```

```

C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C T     = TIME
C F     = FORCE OBTAINED BY LINEAR INTERPOLATION ON TABF. SIZE (NF).
C XDD   = ACCELERATION. SIZE (NX).
C XD    = VELOCITY. SIZE (NX).
C X     = DISPLACEMENT. SIZE (NX).
C AIDF  = A**(-1)*D*F. SIZE (NX). (WRITTEN ON PAPER ONLY).

```

```

C ERROR EXPLANATION
C 1 = SIZE EXCEEDANCE.
C 2 = START TIME LESS THAN TABLE BOUNDS.
C 3 = END TIME GREATER THAN TABLE BOUNDS.
C 4 = RUN HAS DIVERGED.

```

```

C 2001 FORMAT (////15X,42H THE INPUT SCALARS TO SUBROUTINE TRSP1 ARE,

```

```

1 //23X, 10H STARTT = F10.6,
2 //23X, 10H DELTAT = F10.6,
3 //23X, 10H ENDT = F10.6,
4 //23X, 10H NWRITE = I5 )

```

```

C 2040 FORMAT (//9X,8H TIME = F10.6)

```

```

C 2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))

```

```

C 2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
* 10X,13H DISPLACEMENT, 4X,19H A**(-1) * D * FORCES //
* (10X, I3, 4E20.8))

```

```

C 2250 FORMAT (/ 1X 123(1H-) )

```

```

C NERROR=1
IF (NX .GT. 250 .OR. NF .GT. 500) GO TO 999

```

```

C PRINT INPUT SCALARS.
IF (MINI .NE. 4HMINI) GO TO 10
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
IF ((NLINE+2+13) .GT. MAXLIN) GO TO 10
WRITE (NOT,2250)
NLINE = NLINE + 2
GO TO 11
10 CALL PAGEHD
11 WRITE (NOT,2001) STARTT,DELTAT,ENDT,NWRITE
NLINE = NLINE + 13

```

```

C SEARCH NTAPE FOR END OF WRITTEN DATA.
REWIND NTAPE
5 READ (NTAPE) BUFIN,BUFIN,IENDTCK,(BUFIN,I=1,5),NREC
IF (IENDTCK .EQ. 3HEOT) GO TO 7
DO 6 IRFC=1,NREC
6 READ (NTAPE)
GO TO 5
7 BACKSPACE NTAPE

```

```

C

```

CHECK TIME TABLE (TABT).

DO 18 I=1,NF

IF (STARTT .LT. TABT(I,1)) GO TO 999

DO 12 J=2,NTF

IF (TABT(I,J-1) .GE. TABT(I,J)) GO TO 14

12 CONTINUE

J = NTF+1

14 IF (ENDT .LE. TABT(I,J-1)) GO TO 18

GO TO 999

18 CONTINUE

NERROR=2

NERROR=3

C  
C CALCULATE NUMBER OF TIME POINTS TO BE USED.  
NTP = (ENDT-STARTT)/DELTAT + 1.1

C  
C CALCULATE A\*\*-1\*B, A\*\*-1\*C, A\*\*-1\*D.

REWIND NUT1

WRITE (NUT1) ((B(I,J), I=1,NX), J=1,NX)

CALL INVT (A, B, NX, KA)

DO 45 J=1,NX

DO 45 I=1,NX

45 A(I,J) = B(I,J)

REWIND NUT1

READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)

CALL MULTB (A, B, NX, NX, NX, KA, KA)

CALL MULTB (A, C, NX, NX, NX, KA, KA)

CALL MULTB (A, D, NX, NX, NX, KA, KA)

B=B

B=AI

A=AI

B=B

B=AIB

C=AIC

D=AID

C  
C SET INITIAL VALUES.

WRITE (NTAPE) IRUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,  
\* (BUF,I=1,10)

T = STARTT

NW = NWRITE

DO 80 I=1,NX

QD(I) = 0.0

Q(I) = 0.0

XD(I) = XDO(I)

80 X(I) = XQ(I)

DO 86 I=1,NF

DO 84 J=1,NTF

IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 86

84 CONTINUE

86 F(I) = TABF(I,J) + (T-TABT(I,J)) \* (TABF(I,J+1)-TABF(I,J)) /  
\* (TABT(I,J+1)-TABT(I,J))

DO 96 I=1,NX

SS = ZERO

DO 94 J=1,NF

S = D(I,J)\*F(J)

94 SS = SS + S

96 AIDF(I) = SS

DO 97 I=1,NX

97 XDD(I) = AIDF(I)

DO 99 I=1,NX

SS = ZERO

```

DO 98 J=1,NX
S = B(I,J)*XD(J)
SS = SS + S
S = C(I,J)*X(J)
98 SS = SS + S
99 XDD(I) = XDD(I) - SS
C
C SET INTEGRATION CONSTANTS.
P(1) = .5
P(2) = 1. - SQRT(.5)
P(3) = 1. + SQRT(.5)
P(4) = .5
C
C INTEGRATION LOOP. (K=1,HALF STEP), (K=2,HALF STEP AG/
C (K=3,FULL STEP), (K=4,END OF STEP).
C GILL FACTOR = .5
DO 399 ITP=1,NTP
IF (ITP .EQ. 1) GO TO 340
DO 150 K=1,4
DO 110 I=1,NX
Z = XD (I) * DELTAT
ZD = XDD(I) * DELTAT
IF (K .EQ. 4) GO TO 105
R = P(K) * (Z - Q (I))
RD = P(K) * (ZD - QD(I))
GO TO 107
105 R = (Z - 2.*Q (I))/6.
RD = (ZD - 2.*QD(I))/6.
107 X (I) = X (I) + R
XD(I) = XD(I) + RD
Q (I) = Q (I) + 3.*R - P(K)*Z
110 QD(I) = QD(I) + 3.*RD - P(K)*ZD
IF (K .NE. 1) GO TO 115
T = T + .5*DELTAT
GO TO 130
115 IF (K .NE. 3) GO TO 140
T = STARTT + FLOAT(ITP-1)*DELTAT
130 DO 136 I=1,NF
DO 134 J=1,NTF
IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 136
134 CONTINUE
136 F(I) = TABF(1,J) + (T-TABT(1,J)) * (TABF(I,J+1)-TABF(I,J)) /
* (TABT(I,J+1)-TABT(I,J))
140 DO 146 I=1,NX
SS = ZERO
DO 144 J=1,NF
S = D(I,J)*F(J)
144 SS = SS + S
146 AIDF(I) = SS
DO 147 I=1,NX
147 XDD(I) = AIDF(I)
DO 149 I=1,NX
SS = ZERO
DO 148 J=1,NX
S = B(I,J)*XD(J)

```

```

      SS = SS + S
      S = C(I,J)*X(J)
148 SS = SS + S
149 XDD(I) = XDD(I) - SS
150 CONTINUE

```

```

C
C WRITE ANSWERS ON NTAPE FOR LATER USE.
340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
      *           (X(I),I=1,NX)

```

```

C
C SEE IF DATA SHOULD BE PRINTED.
      IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345
      NFL = NF/5
      IF ((NFL*5) .NE. NF) NFL = NFL+1
      IF (MIN1 .NE. 4HMIN1) GO TO 800
      IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
      IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 800
      WRITE (NOT,2250)
      NLINE = NLINE + 2
      GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2040) T
      WRITE (NOT,2050) (F(I), I=1,NF)
      NLINE = NLINE + 3 + 3 + NFL
      NXS = 1
      NXE = NX
      NFLN = (NF-1)/5+1
      IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
342 WRITE (NOT,2060) (I, XDD(I), XD(I), X(I), AIDF(I), I=NXS,NXE)
      NLINE = NLINE + 4 + (NXE-NXS+1)
      IF (NX .EQ. NXE) GO TO 343
      NXS = NXE + 1
      NXE = NX
      IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
      CALL PAGEHD
      GO TO 342
343 NW = 0
345 NW = NW+1

```

```

C
C SEE IF RUN HAS DIVERGED.

```

NERROR=4

```

      DO 350 I=1,NX
      IF (ABS(X(I)) .GT. DIVTOL) GO TO 999
350 CONTINUE

```

```

C
399 CONTINUE

```

```

C
      WRITE (NTAPE) BUF,BUF,EOT,(BUF,I=1,16)
      ENDFILE NTAPE
      RETURN

```

```

C
999 ENDFILE NTAPE
      CALL ZZBOMB (6HTRSP1 ,NERROR)
      END

```

```

SUBROUTINE TRSP1A (A,B,C,D,FMAG,PP,VEL,GL,XDO,XO,STARTT,DELTAT,
*                ENDT,NWRITE,NX,NF,XNAME,KA,NTAPE,NUT1)
DIMENSION A(KA,1),B(KA,1),C(KA,1),D(KA,1),FMAG(1),PP(1),
*          XDO(1),XO(1)
DIMENSION P(4)
COMMON /LWRKV1/ XDD(250),XD(250)
COMMON /LWRKV2/ QD(250),Q(250)
COMMON /LWRKV3/ X(250),AIDF(250)
COMMON /LWRKV4/ F(500)
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINF,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.0/
DATA N T,NOT/5,6/
DATA NLPP,BUF,DIVTOL, PI , ECT/
*   54 , 0.,1.E+35,3.1415927,3HEOT/

```

```

C
C THIS MODIFICATION OF TRSP1 USES (1-COS)/2 FORCING FUNCTION.
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C (A)XDD + (B)XD + (C)X = (D)F FOR XDD, XD, X.
C FOURTH ORDER RUNGE-KUTTA (GILL MODIFICATION) NUMERICAL INTEGRATION
C IS USED.
C THE FORCING FUNCTION, F , IS A SINGLE PERIOD (1-COS)/2 FUNCTION
C BEGINNING AT T=STARTT AND FORWARD PP. THE COORDINATES ARE FORCED
C SIMULTANEOUSLY (SUDDEN ENVELOPMENT) IF VECTOR PP IS CONSTANT, OR AS
C A PENETRATING FUNCTION (EACH COORDINATE FORCE LAGS ITS PREDECESSOR
C DEPENDING ON PENETRATION RATE AND STATION SPACING) IF VECTOR PP
C IS NOT CONSTANT.
C MATERIALS A,B,C,D SHOULD NOT SHARE SAME CORE LOCATION (DUE TO MULT6).
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES INV1,MULT6,PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F)
C   NX = 250
C   NF = 500
C CODED BY RL WOHLER. APRIL 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX COEFFICIENT OF XDD. SIZE (NX,NX). * DESTROYED *
C B = MATRIX COEFFICIENT OF XD. SIZE (NX,NX). * DESTROYED *
C C = MATRIX COEFFICIENT OF X. SIZE (NX,NX). * DESTROYED *
C D = MATRIX COEFFICIENT OF F. SIZE (NX,NF). * DESTROYED *
C FMAG = VECTOR OF COORDINATE FORCE MAGNITUDES SUBJECT TO (1-COS)/2
C VARIATION. SIZE(NF)
C PP = VECTOR OF COORDINATE STATIONS, (CONSTANT IF SUDDEN
C ENVELOPMENT). POSITIVE DIRECTION FOR STATIONS IS OPPOSITE
C TO VEL DIRECTION. SIZE(NF).
C VEL = PENETRATION RATE.
C GL = GUST LENGTH. PERIOD OF (1-COS)/2 FUNCTION.
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX).

```

```

C XO      = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX).
C STARTT= START TIME. FORCING FUNCTION BEGINS.
C DELTAT= INTEGRATION STEP SIZE.
C ENDT   = END TIME.
C NWRITE= MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C        NWRITE = 1  WRITE EVERY POINT (1,2,3,...)
C        NWRITE = 2  WRITE EVERY SECOND POINT (1,3,5,...)
C        ETC
C NX     = SIZE OF MATRICES A,B,C (SQUARE). NUMBER OF ROWS IN D. MAX=250.
C NF     = SIZE OF VECTOR FMAG, NUMBER OF COLS IN D. MAX=500.
C XNAME  = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).
C KA     = ROW DIMENSION OF A,B,C,D IN CALLING PROGRAM.
C NTAPE  = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C NUT1   = NUMBER OF THE UTILITY TAPE. (E.G. 4).
C
C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C T      = TIME
C F      = FORCE EVALUATED BY (1-COS)/2 EXPRESSION, SIZE (NF).
C XDD    = ACCELERATION. SIZE (NX).
C XD     = VELOCITY. SIZE (NX).
C X      = DISPLACEMENT. SIZE (NX).
C AIDF   = A** -1 * D * F. SIZE (NX). (WRITTEN ON PAPER ONLY).
C
C NERROR EXPLANATION
C 1 = SIZE EXCEEDANCE.
C 2 = RUN HAS DIVERGED.
C
C 2001 FORMAT (///15X,43H THE INPUT SCALARS TO SUBROUTINE TRSP1A ARE ,
C 1          //23X,          10H STARTT = F10.6,
C 2          //23X,          10H DELTAT = F10.6,
C 3          //23X,          10H ENDT   = F10.6,
C 4          //23X,          10H NWRITE = I5
C 5          //23X,          10H VEL    = E15.8,
C 6          //23X,          10H GL     = E15.8 )
C 2040 FORMAT (//9X,8H TIME = F10.6)
C 2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))
C 2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
C *          10X,13H DISPLACEMENT, 4X,19H A** -1 * D * FORCES //
C *          (10X, I3, 4E20.8))
C 2250 FORMAT (/ 1X 123(1H-))
C
C NERROR=1
C
C IF (NX .GT. 250 .OR. NF .GT. 500) GO TO 999
C
C PRINT INPUT SCALARS.
C IF (MINI .NE. 4*MINI) GO TO 10
C IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
C IF ((NLINE+2+17) .GT. MAXLIN) GO TO 10
C WRITE (NOT,2250)
C NLINE = NLINE + 2
C GO TO 11
C 10 CALL PAGEHD
C 11 WRITE (NOT,2001) STARTT,DELTAT,ENDT,NWRITE,VEL,GL
C NLINE = NLINE + 17
C

```

```

SEARCH NTAPE FOR END OF WRITTEN DATA.
REWIND NTAPE
5 READ (NTAPE) BUFIN,BUFIN,IEOTCK,(BUFIN,I=1,5),NREC
  IF (IEOTCK .EQ. 3HEQT) GO TO 7
  DO 6 IREC=1,NREC
6 READ (NTAPE)
  GO TO 5
7 BACKSPACE NTAPE

```

```

C
C CALCULATE NUMBER OF TIME POINTS TO BE USED.
  NTP = (ENDT-STARTT)/DELTAT + 1.1

```

```

C
C CALCULATE A**-1*B, A**-1*C, A**-1*D.
  REWIND NUT1
  WRITE (NUT1) ((B(I,J), I=1,NX), J=1,NX)
  CALL INVL (A, B, NX, KA)
  DO 45 J=1,NX
  DO 45 I=1,NX
45 A(I,J) = B(I,J)
  REWIND NUT1
  READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
  CALL MULTB (A, B, NX, NX, NX, KA, KA)
  CALL MULTB (A, C, NX, NX, NX, KA, KA)
  CALL MULTB (A, D, NX, NX, NF, KA, KA)

```

B=E  
 B=AI  
  
 A=AI  
  
 B=E  
 B=AIB  
 C=AIC  
 D=AID

```

C
C FIND FIRST STATION (FORWARD PP) TO ENTER GUST.
  FWDPP = PP(1)
  DO 50 I=1,NF
  IF (PP(I) .LT. FWDPP) FWDPP = PP(I)
50 CONTINUE

```

```

C
C SET INITIAL VALUES.
  WRITE (NTAPE) IRUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,
  * (BUF,I=1,10)
  T = STARTT
  NW = NWRITE
  TPIGL = 2.*PI/GL
  DO 80 I=1,NX
  QD(I) = 0.0
  Q (I) = 0.0
  XD(I) = XDO(I)
80 X (I) = XO (I)
  DO 84 I=1,NF
84 F(I) = 0.0
  DO 94 I=1,NX
94 AIDF(I) = 0.
  DO 97 I=1,NX
97 XDD(I) = AIDF(I)
  DO 99 I=1,NX
  SS = ZERO
  DO 98 J=1,NX
  S = B(I,J)*XD(J)
  SS = SS + S
  S = C(I,J)*X(J)
98 SS = SS + S

```



99 XDD(I) = XDD(I) - SS

C SET INTEGRATION CONSTANTS.

P(1) = .5

P(2) = 1. - SQRT(.5)

P(3) = 1. + SQRT(.5)

P(4) = .5

C

C INTEGRATION LOOP. (K=1, HALF STEP), (K=2, HALF STEP AGAIN),

C (K=3, FULL STEP), (K=4, END OF STEP).

C GILL FACTOR = .5

DO 399 ITP=1,NTP

IF (ITP .EQ. 1) GO TO 340

DO 150 K=1,4

DO 110 I=1,NX

Z = XD (I) \* DELTAT

ZD = XDD(I) \* DELTAT

IF (K .EQ. 4) GO TO 105

R = P(K) \* (Z - Q (I))

RD = P(K) \* (ZD - QD(I))

GO TO 107

105 R = (Z - 2.\*Q (I))/6.

RD = (ZD - 2.\*QD(I))/6.

107 X (I) = X (I) + F

XD(I) = XD(I) + RD

Q (I) = C (I) + 3.\*R - P(K)\*Z

110 QD(I) = QD(I) + 3.\*RD - P(K)\*ZD

IF (K .NE. 1) GO TO 115

T = T + .5\*DELTAT

GO TO 130

115 IF (Y .NE. 3) GO TO 140

T = STARTT + FLOAT(ITP-1)\*DELTAT

130 FWDGPD = VEL\*(T-STARTT)

DO 136 I=1,NF

F(I) = 0.0

GPD = FWDGPD - (PP(I)-FWDPP)

IF (GPD.GT.0.0 .AND. GPD.LT.GL)F(I)=FMAG(I)\*(1.-COS(GPD\*TPIGL))/2.

136 CONTINUE

140 DO 146 I=1,NX

SS = ZERO

DO 144 J=1,NF

S = D(I,J)\*F(J)

144 SS = SS + S

146 AIDF(I) = SS

DO 147 I=1,NX

147 XDD(I) = AIDF(I)

DO 149 I=1,NX

SS = ZERO

DO 148 J=1,NF

S = B(I,J)\*XD(J)

SS = SS + S

S = C(I,J)\*X(J)

148 SS = SS + S

149 XDD(I) = XDD(I) - SS

150 CONTINUE

WRITE ANSWERS ON NTAPE FOR LATER USE.

340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),  
 \* (X(I),I=1,NX)

C

C SEE IF DATA SHOULD BE PRINTED.

IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345

NFL = NF/5

IF ((NFL\*5) .NE. NF) NFL = NFL+1

IF (MINI .NF. 4\*MINI) GO TO 800

IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800

IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 800

WRITE (NOT,2250)

NLINE = NLINE + 2

GO TO 810

800 CALL PAGEHD

810 WRITE (NOT,2040) T

WRITE (NOT,2050) (F(I), I=1,NF)

NLINE = NLINE + 3 + 3 + NFL

NXS = 1

NXE = NX

NFLN = (NF-1)/5+1

IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN

342 WRITE (NOT,2060) (I, XDD(I), XD(I), X(I), AIDF(I), I=NXS,NXE)

NLINE = NLINE + 4 + (NXE-NXS+1)

IF (NX .EQ. NXE) GO TO 343

NXS = NXE + 1

NXE = NX

IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)

CALL PAGEHD

GO TO 342

343 NW = 0

345 NW = NW+1

C

C SEE IF RUN HAS DIVERGED.

NERROR=2

DO 350 I=1,NX

IF (ABS(X(I)) .GT. DIVTOL) GO TO 999

350 CONTINUE

C

399 CONTINUE

C

WRITE (NTAPE) BUF,BUF,EOT,(BUF,I=1,16)

ENDFILE NTAPE

RETURN

C

999 ENDFILE NTAPE

CALL ZZBOMB (6HTRSP1A,NERROR)

END

```

SUBROUTINE TRSP1B (B,C,D,TABT,TABF,XDO,XO,STARTT,DELTAT,ENDT,
*                NWRITE,NX,NF,NTF,XNAME,KA,KF,NTAPE)
DIMENSION B(KA,1),C(KA,1),D(KA,1),TABT(KF,1),TABF(KF,1),
*          XDO(1),XO(1)
DIMENSION P(4)
COMMON /LWRKV1/ XDD(250),XD(250)
COMMON /LWRKV2/ QD(250),Q(250)
COMMON /LWRKV3/ X(250),AIDF(250)
COMMON /LWRKV4/ F(500)
COMMON /LSTART/ IRUMNC,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERC
DATA ZERC/0.D/
DATA NIT,NCT/5,6/
DATA NLPP,BUF,DIVTOL, ECT/
*    54 , 0.,1.E+35,3HEOT/

```

```

C
C THIS MODIFICATION OF TRSP1 ASSUMES COEFFICIENT OF XDD IS UNITY SO
C THAT ONE LESS MATRIX SPACE IS REQUIRED.
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C  $XDD + (B)XD + (C)X = (D)F$  FOR XDD, XD, X.
C FOURTH ORDER RUNGE-KUTTA (GILL MODIFICATION) NUMERICAL INTEGRATION
C IS USED.
C VECTOR F IS OBTAINED BY LINEAR INTERPOLATION USING TABT,TABF.
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F)
C   NX = 250
C   NF = 500
C CODED BY RL WOHLN. FEBRUARY 1967.
C LAST REVISION BY RL WOHLN. MARCH 1976.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C B = MATRIX COEFFICIENT OF XD. SIZE (NX,NX).
C C = MATRIX COEFFICIENT OF X. SIZE (NX,NX).
C D = MATRIX COEFFICIENT OF F. SIZE (NX,NF).
C TABT = TABLE OF TIMES FOR FORCE IN TABF. SIZE (NF,NTF).
C TABF = TABLE OF FORCES. SIZE (NF,NTF).
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX).
C XO = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX).
C STARTT= START TIME.
C DELTAT= INTEGRATION STEP SIZE.
C ENDT = END TIME.
C NWRITE= MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C   NWRITE = 1 WRITE EVERY POINT (1,2,3,...)
C   NWRITE = 2 WRITE EVERY SECOND POINT (1,3,5,...)
C   ETC
C NX = SIZE OF MATRICES B,C (SQUARE). NUMBER OF ROWS IN D. MAX=250.
C NF = NUMBER OF ROWS IN TABT,TABF. NUMBER OF COLS IN D. MAX=500.
C NTF = NUMBER OF COLS IN TABT,TABF.

```

```

C XNAME = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).
C KA   = ROW DIMENSION OF B,C,D IN CALLING PROGRAM.
C KF   = ROW DIMENSION OF TABT,TABF IN CALLING PROGRAM.
C NTAPE = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C
C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C T     = TIME
C F     = FORCE OBTAINED BY LINEAR INTERPOLATION ON TABF. SIZE (NF).
C XDD   = ACCELERATION. SIZE (NX).
C XD    = VELOCITY. SIZE (NX).
C X     = DISPLACEMENT. SIZE (NX).
C AIDF  = A**-1*D*F. SIZE (NX). (WRITTEN ON PAPER ONLY).
C
C ERROR EXPLANATION
C 1 = SIZE EXCEEDANCE.
C 2 = START TIME LESS THAN TABLE BOUNDS.
C 3 = END TIME GREATER THAN TABLE BOUNDS.
C 4 = RUN HAS DIVERGED.
C
2001 FORMAT (///15X,43H THE INPUT SCALARS TO SUBROUTINE TRSPIB ARE ,
1          //23X,          10H STARTT = F10.6,
2          //23X,          10H DELTAT = F10.6,
3          //23X,          10H ENDT   = F10.6,
4          //23X,          10H NWRITE = I5      )
2040 FORMAT (//9X,8H TIME = F10.6)
2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))
2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
*          10X,13H DISPLACEMENT, 4X,19H A**-1 * D * FORCES //
*          (10X, 13, 4E20.8))
2250 FORMAT (/ 1X 123(1H-) )
C
C NERROR=1
IF (NX .GT. 250 .OR. NF .GT. 500) GO TO 999
C
C PRINT INPUT SCALARS.
IF (MINI .NE. 4*MINI) GO TO 10
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
IF ((NLINE+2+13) .GT. MAXLIN) GO TO 10
WRITE (NOT,2250)
NLINE = NLINE + 2
GO TO 11
10 CALL PAGEHD
11 WRITE (NOT,2001) STARTT,DELTAT,ENDT,NWRITE
NLINE = NLINE + 13
C
C SEARCH NTAPE FOR END OF WRITTEN DATA.
REWIND NTAPE
5 READ (NTAPE) BUFIN,GUFIN,IECTCK,(BUFIN,I=1,5),NREC
IF (IECTCK .EQ. 3HEOT) GO TO 7
DO 6 IPEC=1,NREC
6 READ (NTAPE)
GO TO 5
7 BACKSPACE NTAPE
C
C CHECK TIME TABLE (TABT).

```

```

DO 18 I=1,NF
                                NERROR=2
IF (STARTT .LT. TABT(I,1)) GO TO 999
DO 12 J=2,NTF
IF (TABT(I,J-1) .GE. TABT(I,J)) GO TO 14
12 CONTINUE
J = NTF+1
14 IF (ENDT .LE. TABT(I,J-1)) GO TO 18
                                NERROR=3
GO TO 999
18 CONTINUE
C
C CALCULATE NUMBER OF TIME POINTS TO BE USED.
NTP = (ENDT-STARTT)/DELTAT + 1.1
C
C SET INITIAL VALUES.
WRITE (NTAPE) IKUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,
* (BUF,I=1,10)
NW = NWRITE
T = STARTT
DO 80 I=1,NX
QD(I) = 0.0
Q (I) = 0.0
XD(I) = XDO(I)
80 X (I) = XO (I)
DO 86 I=1,NF
DO 84 J=1,NTF
IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 86
84 CONTINUE
86 F(I) = TARB(I,J) + (T-TABT(I,J)) * (TARF(I,J+1)-TARF(I,J)) /
* (TABT(I,J+1)-TABT(I,J))
DO 96 I=1,NX
SS = ZERO
DO 94 J=1,NF
S = D(I,J)*F(J)
94 SS = SS + S
96 AIDF(I) = SS
DO 97 I=1,NX
97 XDD(I) = AIDF(I)
DO 99 I=1,NX
SS = ZERO
DO 98 J=1,NX
S = B(I,J)*XD(J)
SS = SS + S
S = C(I,J)*X(J)
98 SS = SS + S
99 XDD(I) = XDD(I) - SS
C
C SET INTEGRATION CONSTANTS.
P(1) = .5
P(2) = 1. - SQRT(.5)
P(3) = 1. + SQRT(.5)
P(4) = .5
C
C INTEGRATION LOOP. (K=1,HALF STEP), (K=2,HALF STEP AGAIN),

```

(K=3,FULL STEP), (K=4,END OF STEP).

GILL FACTOR = .5

```

DO 399 ITP=1,NTP
IF (ITP .EQ. 1) GO TO 340
DO 150 K=1,4
DO 110 I=1,NX
Z = XD (I) * DELTAT
ZD = XDD(I) * DELTAT
IF (K .EQ. 4) GO TO 105
R = P(K) * (Z -Q (I))
RD = P(K) * (ZD-QD(I))
GO TO 107
105 R = (Z - 2.*Q (I))/6.
RD = (ZD - 2.*QD(I))/6.
107 X (I) = X (I) + R
XD(I) = XD(I) + RD
Q (I) = Q (I) + 3.*R - P(K)*Z
110 QD(I) = QD(I) + 3.*RD - P(K)*ZD
IF (K .NE. 1) GO TO 115
T = T + .5*DELTAT
GO TO 130
115 IF (K .NE. 3) GO TO 140
T = STARTT + FLOAT(ITP-1)*DELTAT
130 DO 136 I=1,NF
DO 134 J=1,NTF
IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 136
134 CONTINUE
136 F(I) = TABF(I,J) + (T-TABT(1,J)) * (TABF(I,J+1)-TABF(I,J)) /
* (TABT(I,J+1)-TABT(I,J))
140 DO 146 I=1,NX
SS = ZERO
DO 144 J=1,NF
S = D(I,J)*F(J)
144 SS = SS + S
146 AIDF(I) = SS
DO 147 I=1,NX
SS = ZERO
DO 148 J=1,NX
S = B(I,J)*XD(J)
SS = SS + S
S = C(I,J)*X(J)
148 SS = SS + S
149 XDD(I) = XDD(I) - SS
150 CONTINUE
C
C WRITE ANSWERS ON NTAPE FOR LATER USE.
340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
* (X(I),I=1,NX)
C
SEE IF DATA SHOULD BE PRINTED.
IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345
NFL = NF/5
IF ((NFL*5) .NE. NF) NFL = NFL+1

```

```

IF (MINI .NE. 4HMINI) GO TO 800
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 800
WRITE (NOT,2250)
NLINE = NLINE + 2
GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2040) T
WRITE (NOT,2050) (F(I), I=1,NF)
NLINE = NLINE + 3 + 3 + NFL
NXS = 1
NXE = NX
NFLN = (NF-1)/5+1
IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
342 WRITE (NOT,2060) (I, XDD(I), XD(I), X(I), AIDF(I), I=NXS,NXE)
NLINE = NLINE + 4 + (NXE-NXS+1)
IF (NX .EQ. NXE) GO TO 343
NXS = NXE + 1
NXE = NX
IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
CALL PAGEHD
GO TO 342
343 NW = 0
345 NW = NW+1
C
C SEE IF RUN HAS DIVERGED.
NERROR=4
DO 350 I=1,NX
IF (ABS(X(I)) .GT. DIVTOL) GO TO 999
350 CONTINUE
C
399 CONTINUE
C
WRITE (NTAPE) BUF,BUF,EOT,(BUF,I=1,16)
ENDFILE NTAPE
RETURN
C
999 ENDFILE NTAPE
CALL ZZBOMB (6HTRSP1B,NERROR)
END

```

```

SUBROUTINE TRSPIC (B,C,D,FMAG,PP,VEL,GL,XDO,XO,STARTT,DELTAT,
*                ENDT,NWRITE,NX,NF,XNAME,KA,NTAPE)
DIMENSION B(KA,1),C(KA,1),D(KA,1),FMAG(1),PP(1),
*          XDO(1),XO(1)
DIMENSION P(4)
COMMON /LWRKV1/ XDD(250),XD(250)
COMMON /LWRKV2/ QD(250),Q(250)
COMMON /LWRKV3/ X(250),AIDF(250)
COMMON /LWRKV4/ F(500)
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.D/
DATA NIT,NOT/5,6/
DATA NLPP,BUF,DIVTOL,  PI  , EOT/
*    54 , 0.,1.E+35,3.1415927,3HEOT/

```

```

C
C THIS MODIFICATION OF TRSP1 USES (1-COS)/2 FORCING FUNCTION (LIKE
C TRSP1A) AND ASSUMES COEFFICIENT OF XDD IS UNITY SO THAT ONE LESS
C MATRIX SPACE IS REQUIRED THAN TRSP1A.
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C  $XDD + (B)XD + (C)X = (D)F$  FOR XDD, XD, X.
C FOURTH ORDER RUNGE-KUTTA (GILL MODIFICATION) NUMERICAL INTEGRATION
C IS USED.
C THE FORCING FUNCTION, F , IS A SINGLE PERIOD (1-COS)/2 FUNCTION
C BEGINNING AT T=STARTT AND FORWARD PP. THE COORDINATES ARE FORCED
C SIMULTANEOUSLY (SUDDEN ENVELOPMENT) IF VECTOR PP IS CONSTANT, OR AS
C A PENETRATING FUNCTION (EACH COORDINATE FORCE LAGS ITS PREDECESSOR
C DEPENDING ON PENETRATION RATE AND STATION SPACING) IF VECTOR PP
C IS NOT CONSTANT.
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F)
C   NX = 250
C   NF = 500
C CODED BY RL WOHLER. FEBRUARY 1967.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C B   = MATRIX COEFFICIENT OF XD.  SIZE (NX,NX).
C C   = MATRIX COEFFICIENT OF X.  SIZE (NX,NX).
C D   = MATRIX COEFFICIENT OF F.  SIZE (NX,NF).
C FMAG = VECTOR OF COORDINATE FORCE MAGNITUDES SUBJECT TO (1-COS)/2
C VARIATION. SIZE(NF).
C PP  = VECTOR OF COORDINATE STATIONS. (CONSTANT IF SUDDEN
C ENVELOPMENT). POSITIVE DIRECTION FOR STATIONS IS OPPOSITE
C TO VEL DIRECTION. SIZE(NF).
C VEL = PENETRATION RATE.
C GL  = GUST LENGTH. PERIOD OF (1-COS)/2 FUNCTION.
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX).

```



```

C   XO      = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX).
C   STARTT= START TIME. FORCING FUNCTION BEGINS.
C   DELTAT= INTEGRATION STEP SIZE.
C   ENDT   = END TIME.
C   NWRITE= MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C           NWRITE = 1  WRITE EVERY POINT (1,2,3,...)
C           NWRITE = 2  WRITE EVERY SECOND POINT (1,3,5,...)
C           ETC
C   NX     = SIZE OF MATRICES B,C (SQUARE). NUMBER OF ROWS IN D. MAX=250.
C   NF     = SIZE OF VECTOR FMAG, NUMBER OF COLS IN D. MAX=500.
C   XNAME  = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).
C   KA     = ROW DIMENSION OF B,C,D IN CALLING PROGRAM.
C   NTAPE  = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C
C   THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C   T      = TIME
C   F      = FORCE EVALUATED BY (1-COS)/2 EXPRESSION, SIZE (NF).
C   XDD    = ACCELERATION. SIZE (NX).
C   XD     = VELOCITY.      SIZE (NX).
C   X      = DISPLACEMENT. SIZE (NX).
C   AIDF   = A**-1*D*F. SIZE (NX). (WRITTEN ON PAPER ONLY).
C
C   NERROR EXPLANATION
C   1 = SIZE EXCEEDANCE.
C   2 = RUN HAS DIVERGED.
C
C 2001 FORMAT (///15X,43H THE INPUT SCALARS TO SUBROUTINE TRSP1C ARE ,
1          //23X,          10H STARTT = F10.6,
2          //23X,          10H DELTAT = F10.6,
3          //23X,          10H ENDT   = F10.6,
4          //23X,          10H NWRITE = I5   ,
5          //23X,          10H VEL    = E15.8,
6          //23X,          10H GL     = E15.8 )
2040 FORMAT (//9X,8H TIME = F10.6)
2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))
2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
*          10X,13H DISPLACEMENT, 4X,19H A**-1 * D * FORCES //
*          (10X, 13, 4E20.8))
2250 FORMAT (/ 1X 123(1H- ) )
C
C                                     NERROR=1
C   IF (NX .GT. 250 .OR. NF .GT. 500) GO TO 999
C
C PRINT INPUT SCALARS.
C   IF (MINI .NE. 4HMINI) GO TO 10
C   IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
C   IF ((NLINE+2+17) .GT. MAXLIN) GO TO 10
C   WRITE (NOT,2250)
C   NLINE = NLINE + 2
C   GO TO 11
10 CALL PAGEPD
11 WRITE (NOT,2001) STARTT,DELTAT,ENDT,NWRITE,VEL,GL
C   NLINE = NLINE + 17
C
C SEARCH NTAPE FOR END OF WRITTEN DATA.

```

```

REWIND NTAPE
5 READ (NTAPE) BUFIN,BUFIN,IEOTCK,(BUFIN,I=1,5),NREC
  IF (IEOTCK .EQ. 3HEOT) GO TO 7
  DO 6 IREC=1,NREC
6 READ (NTAPE)
  GO TO 5
7 BACKSPACE NTAPE

C
C CALCULATE NUMBER OF TIME POINTS TO BE USED.
  NTP = (ENDT-STARTT)/DELTAT + 1.1
C
C FIND FIRST STATION (FORWARD PP) TO ENTER GUST.
  FWDPP = PP(1)
  DO 50 I=1,NF
  IF (PP(I) .LT. FWDPP) FWDPP = PP(I)
50 CONTINUE

C
C SET INITIAL VALUES.
  WRITE (NTAPE) IRUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,
  *          (BUF,I=1,10)
  T = STARTT
  NW = NWRITE
  TPIGL = 2.*PI/GL
  DO 80 I=1,NX
  QD(I) = 0.0
  Q (I) = 0.0
  XD(I) = XDO(I)
80 X (I) = XO (I)
  DO 84 I=1,NF
84 F(I) = 0.0
  DO 94 I=1,NX
94 AIDF(I) = 0.
  DO 97 I=1,NX
97 XDD(I) = AIDF(I)
  DO 99 I=1,NX
  SS = ZERO
  DO 98 J=1,NX
  S = B(I,J)*XD(J)
  SS = SS + S
  S = C(I,J)*X(J)
98 SS = SS + S
99 XDD(I) = XDD(I) - SS

C
C SET INTEGRATION CONSTANTS.
  P(1) = .5
  P(2) = 1. - SQRT(.5)
  P(3) = 1. + SQRT(.5)
  P(4) = .5

C
C INTEGRATION LOOP. (K=1,HALF STEP), (K=2,HALF STEP AGAIN),
C          (K=3,FULL STEP), (K=4,END OF STEP).
C
C GILL FACTOR = .5
  DO 399 ITP=1,NTP
  IF (ITP .EQ. 1) GO TO 340
  DO 150 K=1,4

```

```

DO 110 I=1,NX
Z = XD (I) * DELTAT
ZD = XDD(I) * DELTAT
IF (K .EQ. 4) GO TO 105
R = P(K) * (Z -Q (I))
RD = P(K) * (ZD-QD(I))
GO TO 107
105 R = (Z - 2.*Q (I))/6.
RD = (ZD - 2.*QD(I))/6.
107 X (I) = X (I) + R
XD(I) = XD(I) + RD
Q (I) = Q (I) + 3.*R - P(K)*Z
110 QD(I) = QD(I) + 3.*RD - P(K)*ZD
IF (K .NE. 1) GO TO 115
T = T + .5*DELTAT
GO TO 130
115 IF (K .NE. 3) GO TO 140
T = STARTT + FLOAT(ITP-1)*DELTAT
130 FWDGPD = VEL*(T-STARTT)
DO 136 I=1,NF
F(I) = 0.0
GPD = FWDGPD - (PP(I)-FWDPP)
IF (GPD.GT.0.0 .AND. GPD.LT.GL)F(I)=FMAG(I)*(1.-COS(GPD*TPIGL))/2.
136 CONTINUE
140 DO 146 I=1,NX
SS = ZERO
DO 144 J=1,NF
S = D(I,J)*F(J)
144 SS = SS + S
146 AIDF(I) = SS
DO 147 I=1,NX
147 XDD(I) = AIDF(I)
DO 149 I=1,NX
SS = ZERO
DO 148 J=1,NX
S = B(I,J)*XD(J)
SS = SS + S
S = C(I,J)*X(J)
148 SS = SS + S
149 XDD(I) = XDD(I) - SS
150 CONTINUE
C
C WRITE ANSWERS ON NTAPE FOR LATER USE.
340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),J=1,NX), (XD(I),I=1,NX),
* (X(I),I=1,NX)
C
C SEE IF DATA SHOULD BE PRINTED.
IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345
NFL = NF/5
IF ((NFL*5) .NE. NF) NFL = NFL+1
IF (MINI .NE. 4*MINI) GO TO 800
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
IF ((NLINE+2+3+NFL+4+NX) .GT. MAXLIN) GO TO 800
WRITE (NOT,2250)
NLINE = NLINE + 2

```

```

      GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2040) T
      WRITE (NOT,2050) (F(I), I=1,NF)
      NLINE = NLINE + 3 + 3 + NFL
      NXS = I
      NXE = NX
      NFLN = (NF-1)/5+1
      IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
342 WRITE (NOT,2060) (I, XDD(I), XD(I), X(I), AIDF(I), I=NXS,NXE)
      NLINE = NLINE + 4 + (NXE-NXS-1)
      IF (NX .EQ. NXE) GO TO 343
      NXS = NXE + 1
      NXE = NX
      IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
      CALL PAGEHD
      GO TO 342
343 NW = 0
345 NW = NW+1
C
C SEE IF RUN HAS DIVERGED.
                                     NERROR=2
      DO 350 I=1,NX
      IF (ABS(X(I)) .GT. DIVTOL) GO TO 999
350 CONTINUE
C
C 399 CONTINUE
      WRITE (NTAPE) BUF,BUF,EOT,(BUF,I=1,16)
      ENDFILE NTAPE
      RETURN
C
999 ENDFILE NTAPE
      CALL ZZBOMB (6HTRSPIC,NERROR)
      FND

```

```

SUBROUTINE TRSP2 (A,B,C,D,TABT,TABF,XDO,XO,STARTT,DELTAT,ENDT,
1          BETA,NWRITE,NX,NF,NTF,XNAME,KA,KF,NTAPE,NUT1)
DIMENSION A(KA,1),B(KA,1),C(KA,1),D(KA,1),TABT(KF,1),TABF(KF,1),
*          XDO(1),XO(1)
COMMON /LWRKV2/ XM1(250),XM2(250)
COMMON /LWRKV3/ X(250),XD(250)
COMMON /LWRKV4/ XDD(250),F(250)
COMMON /LWRKV5/ FM1(250),FM2(250)
COMMON /LSTART/ IPUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.0/
DATA NIT,NOT/5,6/
DATA NLPP,BUF,DIVTOL, EGT/
*          54 , 0.,1.E+35,3HECT/

```

```

C
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C (A)XDD + (B)XD + (C)X = (D)F FOR XDD, XD, X.
C THIRD ORDER NEWMARK-CHAN-BETA NUMERICAL INTEGRATION IS USED.
C VECTOR F IS OBTAINED BY LINEAR INTERPOLATION USING TABT,TABF.
C MATRICES A,B,C,D SHOULD NOT SHARE SAME CORE LOCATION (DUE TO MULT6).
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES INVI,MULT,MULTR,PAGEHD,ZZR0 3.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XL,X,F)
C NX = 250
C NF = 250
C CODED BY RL WOHLER. MAY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX COEFFICIENT OF XDD. SIZE (NX,NX). * DESTROYED *
C B = MATRIX COEFFICIENT OF XD. SIZE (NX,NX). * DESTROYED *
C C = MATRIX COEFFICIENT OF X. SIZE (NX,NX). * DESTROYED *
C D = MATRIX COEFFICIENT OF F. SIZE (NX,NF). * DESTROYED *
C TABT = TABLE OF TIMES FOR FORCE IN TABF. SIZE (NF,NTF).
C TABF = TABLE OF FORCES. SIZE (NF,NTF).
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX).
C XO = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX).
C STARTT = START TIME.
C DELTAT = INTEGRATION STEP SIZE.
C ENDT = END TIME.
C BETA = PARAMETER OF GENERALISED ACCELERATION (BETWEEN .0 AND .25).
C NWRITE = MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C NWRITE = 1 WRITE EVERY POINT (1,2,3,...)
C NWRITE = 2 WRITE EVERY SECOND POINT (1,3,5,...)
C FTC
C NX = SIZE OF MATRICES A,B,C (SQUARE). NUMBER OF ROWS IN D. MAX=250.
C NF = NUMBER OF ROWS IN TABT,TABF. NUMBER OF COLS IN D. MAX=250.
C NTF = NUMBER OF COLS IN TABT,TABF.
C XNAME = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).

```

```

C KA = ROW DIMENSION OF A,B,C,D IN CALLING PROGRAM.
C KF = POW DIMENSION OF TABT,TABF IN CALLING PROGRAM.
C NTAPE = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C NUT1 = NUMBER OF THE UTILITY TAPE. (E.G. 4).
C
C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C T = TIME
C F = FORCE OBTAINED BY LINEAR INTERPOLATION ON TABF. SIZE (NF).
C XDD = ACCELERATION. SIZE (NX).
C XD = VELOCITY. SIZE (NX).
C X = DISPLACEMENT. SIZE (NX).
C
C ERROR EXPLANATION
C 1 = SIZE EXCEEDANCE.
C 2 = START TIME LESS THAN TABLE BOUNDS.
C 3 = END TIME GREATER THAN TABLE BOUNDS.
C 4 = RUN HAS DIVERGED.
C
C 2001 FORMAT (////15X,42H THE INPUT SCALARS TO SUBROUTINE TRSP2 ARE ,
1 //23X, 10H STARTT = F10.6,
2 //23X, 10H DELTAT = F10.6,
3 //23X, 10H ENDT = F10.6,
4 //23X, 10H BETA = F10.6,
5 //23X, 10H NWRITE = 15 )
C 2040 FORMAT (//9X,8H TIME = F10.6)
C 2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))
C 2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
1 10X,13H DISPLACEMENT // (10X, 13, 3E20.8))
C 2250 FORMAT (/ 1X 123(1H-))
C
C NERROR=1
C IF (NX .GT. 250 .OR. NF .GT. 250) GO TO 999
C
C PRINT INPUT SCALARS.
C IF (MINI .NE. 4HMINT) GO TO 10
C IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
C IF ((NLINE+2+15) .GT. MAXLIN) GO TO 10
C WRITE (NOT,2250)
C NLINE = NLINE + 2
C GO TO 11
C 10 CALL PAGEHD
C 11 WRITE (NOT,2061) STARTT,DELTAT,ENDT,BETA,NWRITE
C NLINE = NLINE + 15
C
C SEARCH NTAPE FOR END OF WRITTEN DATA.
C REWIND NTAPE
C 5 READ (NTAPE) EUFIN,BUFIN,IEOTCK,(BUFIN,I=1,5),NREC
C IF (IEOTCK .EQ. 3HEOT) GO TO 7
C DO 6 IREC=1,NREC
C 6 READ (NTAPE)
C GO TO 5
C 7 BACKSPACE NTAPE
C
C CHECK TIME TABLE (TABT).
C DO 18 I=1,NF

```

NERROR=2

```

IF (STARTT .LT. TABT(I,1)) GO TO 999
DO 12 J=2,NTF
IF (TART(I,J-1) .GE. TABT(I,J)) GO TO 14
12 CONTINUE
J = NTF+1
14 IF (ENDT .LE. TABT(I,J-1)) GO TO 18

```

NERROR=3

```

GO TO 999
18 CONTINUE

```

```

C
C CALCULATE NUMBER OF TIME POINTS TO BE USED.
NTP = (ENDT-STARTT)/DELTAT + 1.1

```

```

C
C CALCULATE A**-1*B, A**-1*C, A**-1*D.
REWIND NUT1

```

```

WRITE (NUT1) ((B(I,J), I=1,NX), J=1,NX)
CALL INVT (A, B, NX, KA)

```

B=B  
B=AI

```

DO 45 J=1,NX
DO 45 I=1,NX

```

```

45 A(I,J) = B(I,J)

```

A=AI

```

REWIND NUT1
READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
CALL MULTP (A, B, NX, NX, NX, KA, KA)
CALL MULTF (A, C, NX, NX, NX, KA, KA)
CALL MULTB (A, D, NX, NX, NX, KA, KA)

```

B=B  
B=AIB  
C=AIC  
D=AID

```

C CALCULATE INITIAL FORCE(F), ACCELERATION(XDD).

```

```

DO 55 I=1,NF
DO 53 J=1,NTF
IF (STARTT .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 55

```

```

53 CONTINUE

```

```

55 FM1(I)=TARF(I,J)+(STARTT-TABT(I,J)) * (TARF(I,J+1)-TARF(I,J)) /
I (TABT(I,J+1)-TABT(I,J))

```

```

DO 66 I=1,NX
SS = ZERO
DO 65 J=1,NF
S = D(I,J)*FM1(J)

```

```

65 SS = SS + S

```

```

66 XDD(I) = SS

```

```

DO 69 I=1,NX
SS = ZERO
DO 68 J=1,NX
S = B(I,J)*XDD(J)
SS = SS + S
S = C(I,J)*XO(J)

```

```

68 SS = SS + S

```

```

69 XDD(I) = XDD(I) - SS

```

```

C
C WRITE HEADER AND ANSWERS AT START ON NTAPE FOR LATER USE.
WRITE (NTAPE) IRUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,
* (FUF,I=1,10)
WRITE (NTAPE) STARTT, (FM1(I),I=1,NF), (XDD(I),I=1,NX),
* (XDD(I),I=1,NX), (XO(I),I=1,NX)

```

C

PRINT DATA AT START.

NFL = NF/5

IF ((NFL\*5) .NE. NF) NFL = NFL+1

IF (MINI .NE. 4HMINI) GO TO 70

IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 70

IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 70

WRITE (NOT,2250)

NLINE = NLINE + 2

GO TO 71

70 CALL PAGEHD

71 WRITE (NOT,2040) STARTT

WRITE (NOT,2050) (FM1(I), I=1,NF)

NLINE = NLINE + 3 + 3 + NFL

NXS = 1

NXE = NX

NFLN = (NF-1)/5+1

IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN

82 WRITE (NOT,2060) (I,XDD(I),XDO(I),XO(I), I=NXS,NXE)

NLINE = NLINE + 4 + (NXE-NXS+1)

IF (NX .EQ. NXE) GO TO 83

NXS = NXE + 1

NXE = NX

IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)

CALL PAGEHD

GO TO 82

83 NW = 1

F

C CALCULATE SCALAR CONSTANTS FOR INTEGRATION.

C1 = DELTAT / 2.

C2 = (.5 - FETA) \* DELTAT\*\*2

C3 = (.25- BETA) \* DELTAT\*\*3

C4 = FETA \* DELTAT\*\*2

C5 = 1./DELTAT

C6 = 1./DELTAT\*\*2

C7 = -2. + 1./BETA

C8 = (1. - 2.\*BETA) \* DELTAT\*\*2

C

C CALCULATE AT START TIME + DELTA TIME.

T = STARTT + DELTAT

DO 95 I=1,NF

DO 93 J=1,NTF

IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 95

93 CONTINUE

95 F(I) = TABF(I,J) + (T-TABT(I,J)) \* (TABF(I,J+1)-TABF(I,J)) /  
\* (TABT(I,J+1)-TABT(I,J))

REWIND NUT1

WRITE (NUT1) ((B(I,J), I=1,NX), J=1,NX)

WRITE (NUT1) ((D(I,J), I=1,NX), J=1,NF)

WRITE (NUT1) ((C(I,J), I=1,NX), J=1,NX)

CALL MULT (B,C,A,NX,NX,NX,KA,KA)

DO 101 I=1,NX

DO 100 J=1,NX

100 A(I,J) = C1\*B(I,J) - C2\*C(I,J) - C3\*A(I,J)

101 A(1,I) = 1. + A(1,I)

DO 111 I=1,NX

B=AIP  
D=AID  
C=AIC  
A=AIBAIC

A=P



```

DO 110 J=1,NX
110 B(I,J) = C1*B(I,J) + C4*C(I,J)
111 B(I,I) = 1. + B(I,I)
CALL INVL (B,C,NX,KA)
WRITE (NUT1) ((C(I,J), I=1,NX), J=1,NX)
CALL MULTB (C,A,NX,NX,NX,KA,KA)
CALL MULT (A,X0,XM1,NX,NX,1,KA,KA)
REWIND NUT1
READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
CALL MULT (B,B,A,NX,NX,NX,KA,KA)
DO 121 I=1,NX
DC 120 J=1,NX
120 A(I,J) = -C3*A(I,J)
121 A(I,I) = DELTAT + A(I,I)
CALL MULTB (C,A,NX,NX,NX,KA,KA)
CALL MULT (A,XD0,XM2,NX,NX,1,KA,KA)
CALL MULTP (C,D,NX,NX,NF,KA,KA)
CALL MULT (D,F,FM2,NX,NF,1,KA,KF)
DC 131 I=1,NX
DO 130 J=1,NX
130 B(I,J) = C3*B(I,J)
131 B(I,I) = C2 + B(I,I)
CALL MULTB (C,B,NX,NX,NX,KA,KA)
READ (NUT1) ((D(I,J), I=1,NX), J=1,NF)
CALL MULTB (B,D,NX,NX,NF,KA,KA)
CALL MULT (D,FM1,X,NX,NF,1,KA,KF)
DO 140 I=1,NX
X(I) = XM1(I) + XM2(I) + C4*FM2(I) + X(I)
140 XD(I) = C5 * (X(I) - X0(I))
REWIND NUT1
READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
READ (NUT1) ((D(I,J), I=1,NX), J=1,NF)
READ (NUT1) ((C(I,J), I=1,NX), J=1,NX)
DC 146 I=1,NX
SS = ZERO
DO 144 J=1,NF
S = D(I,J)*F(J)
144 SS = SS + S
146 XDD(I) = SS
DO 149 I=1,NX
SS = ZERO
DO 148 J=1,NX
S = B(I,J)*XD(J)
SS = SS + S
S = C(I,J)*X(J)
148 SS = SS + S
149 XDD(I) = XDD(I) - SS
C
C CALCULATE CONSTANT COEFFICIENT MATRICES FOR TIME T2,T3,ETC.
DO 151 I=1,NX
DO 150 J=1,NX
150 E(I,J) = -C8*C(I,J)
151 B(I,I) = 2. + B(I,I)
READ (NUT1) ((A(I,J), I=1,NX), J=1,NX)
CALL MULTB (A,B,NX,NX,NX,KA,KA)

```

B=S  
 C=SI  
 A=SIP  
 XM1=AXO  
 B=AIB  
 A=AIBAIB  
 A=C  
 A=SIQ  
 XM2=AXD0  
 D=SIAD  
 FM2=DF1  
 B=R  
 B=SIR  
 D=AID  
 D=SIAD  
 X=DF0  
 X=X1  
 XD=XD1  
 B=AIB  
 D=AID  
 C=AIC  
 B=T  
 A=SI  
 B=SIT

```

REWIND NUT1
READ (NUT1) ((A(I,J), I=1,NX), J=1,NX)
DO 161 I=1,NX
DO 160 J=1,NX
160 C(I,J) = -C1*A(I,J) + C4*C(I,J)
161 C(I,I) = 1. + C(I,I)
READ 'NUT1)
READ (NUT1)
READ (NUT1) ((A(I,J), I=1,NX), J=1,NX)
CALL MULTB (A,C,NX,NX,NX,KA,KA)
CALL MULTB (A,D,NX,NX,NF,KA,KA)
DO 180 I=1,NX
DO 180 J=1,NF
180 D(I,J) = (4*D(I,J))
DO 185 I=1,NX
185 XM1(I) = X0(I)
C
C CALCULATE X,XD,XDD FOR TIME = T2,T3,ETC.
DO 399 ITP=2,NTP
IF (ITP .EQ. 2) GO TO 340
DO 191 I=1,NX
XM2(I) = XM1(I)
191 XM1(I) = X (I)
DO 192 I=1,NF
FM2(I) = FM1(I)
192 FM1(I) = F (I)
T = STARTT + FLOAT(ITP-1)*DELTAT
C
DO 194 I=1,NF
DO 193 J=1,NTF
IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 194
193 CONTINUE
194 F(I) = TABF(I,J) + (T-TABT(I,J)) * (TABF(I,J+1)-TABF(I,J)) /
* (TABT(I,J+1)-TABT(I,J))
DO 201 J=1,NF
201 FM2(J) = F(J) + C7*FM1(J) + FM2(J)
DO 206 I=1,NX
SS = ZERO
DO 204 J=1,NF
S = G(I,J)*FM2(J)
204 SS = SS + S
206 X(I) = SS
DO 208 I=1,NX
SS = ZERO
DO 207 J=1,NX
S = B(I,J)*XM1(J)
SS = SS + S
S = C(I,J)*XM2(J)
207 SS = SS - S
208 X(I) = X(I) + SS
DO 209 I=1,NX
XD (I) = C5 * (X(I) - XM1(I))
/ 209 XDD(I) = C6 * (X(I) - 2.*XM1(I) + XM2(I))
C
C WRITE ANSWERS ON NTAPE FOR LATER USE.

```

A=AIB

C=U

A=SI  
C=SIU  
D=SIAD

```

340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
*          (X(I),I=1,NX)
C
C SEE IF DATA SHOULD BE PRINTED.
  IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345
  NFL = NF/5
  IF ((NFL*5) .NE. NF) NFL = NFL+1
  IF (MINI .NE. 4HMINI) GO TO 800
  IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
  IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 800
  WRITE (NOT,2250)
  NLINE = NLINE + 2
  GO TO 810
800 CALL PAGEHD
810 WRITE (NOT,2040) T
  WRITE (NOT,2050) (F(I), I=1,NF)
  NLINE = NLINE + 3 + 3 + NFL
  NXS = 1
  NXE = NX
  NFLN = (NF-1)/5+1
  IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
342 WRITE (NOT,2060) (I, XDD(I), XD(I), X(I), I=NXS,NXE)
  NLINE = NLINE + 4 + (NXE-NXS+1)
  IF (NX .EQ. NXE) GO TO 343
  NXS = NXE + 1
  NXE = NX
  IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
  CALL PAGEHD
  GO TO 342
343 NW = 0
345 NW = NW+1
C
C SEE IF RUN HAS DIVERGED.
  DO 350 I=1,NX
  IF (ABS(X(I)) .GT. DIVTOL) GO TO 999
350 CONTINUE
C
399 CONTINUE
C
  WRITE (NTAPE) BUF,BUF,EOT,(BUF,I=1,16)
  ENDFILE NTAPE
  RETURN
C
999 ENDFILE NTAPE
  CALL ZZBOMB (6HTRSP2 ,NERROR)
  END

```

NERROR=4

```

SUBROUTINE TRSP2A (A,B,C,D,FMAG,PP,VEL,GL,XDO,XO,STARTT,DELTAT,
*           ENDT,BETA,NWRITE,NX,NF,XNAME,KA,NTAPE,NUT1)
DIMENSION A(KA,1),B(KA,1),C(KA,1),D(KA,1),FMAG(1),PP(1),
*           XDO(1),XO(1)
COMMON /LWRKV2/ XM1(250),XM2(250)
COMMON /LWRKV3/ X(250),XD(250)
COMMON /LWRKV4/ XDD(250),F(250)
COMMON /LWRKV5/ FM1(250),FM2(250)
COMMON /LSTART/ IPUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.0/
DATA NIT,NOT/5,6/
DATA NLPP,BUF,DIVTOL,    PI    , EOT/
*           54 , 0.,1.E+35,3.1415927,3HEOT/

```

```

C
C THIS MODIFICATION OF TRSP2 USES (1-COS)/2 FORCING FUNCTION.
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C (A)XDD + (B)XD + (C)X = (D)F FOR XDD, XD, X.
C THIRD ORDER NEWMARK-CHAN-BETA NUMERICAL INTEGRATION IS USED.
C THE FORCING FUNCTION, F , IS A SINGLE PERIOD (1-COS)/2 FUNCTION
C BEGINNING AT T=STARTT AND FORWARD PP. THE COORDINATES ARE FORCED
C SIMULTANEOUSLY (SUDDEN ENVELOPMENT) IF VECTOR PP IS CONSTANT, OR AS
C A PENETRATING FUNCTION (EACH COORDINATE FORCE LAGS ITS PREDECESSOR
C DEPENDING ON PENETRATION RATE AND STATION SPACING) IF VECTOR PP
C IS NOT CONSTANT.
C MATRICES A,B,C,D SHOULD NOT SHARE SAME CORE LOCATION (DUE TO MULTB).
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES INVI,MULT,MULTB,PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F)
C   NX = 250
C   NF = 250
C CODED BY RL WOHLER. MAY 1965.
C LAST REVISION BY RL WOHLER. MARCH 1976.
C
C ALL SUBROUTINE ARGUMENTS ARE INPUT
C A = MATRIX COEFFICIENT OF XDD. SIZE (NX,NX). * DESTROYED *
C B = MATRIX COEFFICIENT OF XD. SIZE (NX,NX). * DESTROYED *
C C = MATRIX COEFFICIENT OF X. SIZE (NX,NX). * DESTROYED *
C D = MATRIX COEFFICIENT OF F. SIZE (NX,NF). * DESTROYED *
C FMAG = VECTOR OF COORDINATE FORCE MAGNITUDES SUBJECT TO (1-COS)/2
C VARIATION. SIZE(NF).
C PP = VECTOR OF COORDINATE STATIONS, (CONSTANT IF SUDDEN
C ENVELOPMENT). POSITIVE DIRECTION FOR STATIONS IS OPPOSITE
C TO VEL DIRECTION. SIZE(NF).
C VEL = PENETRATION RATE.
C GL = GUST LENGTH. PERIOD OF (1-COS)/2 FUNCTION.
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX).
C XO = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX).
C STARTT= START TIME. FORCING FUNCTION BEGINS.

```

```

C DELTAT= INTEGRATION STEP SIZE.
C ENDT = END TIME.
C BETA = PARAMETER OF GENERALISED ACCELERATION (BETWEEN .0 AND .25).
C NWRITE= MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C     NWRITE = 1  WRITE EVERY POINT (1,2,3,...)
C     NWRITE = 2  WRITE EVERY SECOND POINT (1,3,5,...)
C     ETC
C NX    = SIZE OF MATRICES A,B,C (SQUARE). NUMBER OF ROWS IN D. MAX=250.
C NF    = SIZE OF VECTOR FMAG, NUMBER OF COLS IN D. MAX=250.
C XNAME = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).
C KA    = ROW DIMENSION OF A,B,C,D IN CALLING PROGRAM.
C NTAPE = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C NUTI  = NUMBER OF THE UTILITY TAPE. (E.G. 4).
C
C     THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C T     = TIME
C F     = FORCE EVALUATED BY (1-COS)/2 EXPRESSION, SIZE (NF).
C XDD   = ACCELERATION. SIZE (NX).
C XD    = VELOCITY.     SIZE (NX).
C X     = DISPLACEMENT. SIZE (NX).
C
C     NERROR EXPLANATION
C 1 = SIZE EXCEEDANCE.
C 2 = RUN HAS DIVERGED.
C
C 2001 FORMAT (////15X,43H THE INPUT SCALARS TO SUBROUTINE TRSP2A ARE ,
1         //23X,          10H STARTT = F10.6,
2         //23X,          10H DELTAT = F10.6,
3         //23X,          10H ENDT   = F10.6,
4         //23X,          10H BETA   = F10.6,
5         //23X,          10H NWRITE = I5   ,
6         //23X,          10H VEL    = E15.8,
7         //23X,          10H GL     = E15.8 )
C 2040 FORMAT (//9X,8H TIME = F10.6)
C 2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))
C 2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
1         10X,13H DISPLACEMENT // (10X, I3, 3E20.8))
C 2250 FORMAT (/ 1X 123(1H-))
C
C                                     NERROR=1
C     IF (NX .GT. 250 .OR. NF .GT. 250) GO TO 999
C
C PRINT INPUT SCALARS.
C     IF (MINI .NE. 4HMINI) GO TO 10
C     IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
C     IF ((NLINE+2+19) .GT. MAXLIN) GO TO 10
C     WRITE (NOT,2250)
C     NLINE = NLINE + 2
C     GO TO 11
C 10 CALL PAGEHD
C 11 WRITE (NOT,2061) STARTT,DELTAT,ENDT,BETA,NWRITE,VEL,GL
C     NLINE = NLINE + 19
C
C SEARCH NTAPE FOR END OF WRITTEN DATA.
C     REWIND NTAPE

```

```

5 READ (NTAPE) RUFIN,RUFIN,IEOTCK,(BUFIN,I=1,5),NREC
  IF (IEOTCK .EQ. 3HEOT) GO TO 7
  DO 6 IREC=1,NREC
6 READ (NTAPE)
  GO TO 5
7 BACKSPACE NTAPE

C
C CALCULATE NUMBER OF TIME POINTS TO BE USED.
  NTP = (ENDT-STARTT)/DELTAT + 1.1

C
C CALCULATE A**-1*B, A**-1*C, A**-1*D.
  REWIND NUT1
  WRITE (NUT1) ((B(I,J), I=1,NX), J=1,NX)
  CALL INVI (A, B, NX, KA)
  DO 45 J=1,NX
  DO 45 I=1,NX
45 A(I,J) = B(I,J)
  REWIND NUT1
  READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
  CALL MULTB (A, B, NX, NX, NX, KA, KA)
  CALL MULTB (A, C, NX, NX, NX, KA, KA)
  CALL MULTB (A, D, NX, NX, NF, KA, KA)

C
C FIND FIRST STATION (FORWARD PP) ;D ENTER GUST.
  FWDPP = PP(1)
  DO 50 I=1,NF
  IF (PP(I) .LT. FWDPP) FWDPP = PP(I)
50 CONTINUE
  TPIGL = 2.*PI/GL

C
C CALCULATE INITIAL FORCE(F), ACCELERATION(XDD!).
  DO 55 I=1,NF
55 FMI(I) = 0.
  DO 69 I=1,NX
  SS = ZERO
  DO 68 J=1,NX
  S = B(I,J)*XDO(J)
  SS = SS + S
  S = C(I,J)*XG(J)
68 SS = SS + S
69 XDD(I) = XDD(I) - SS

C
C WRITE HEADER AND ANSWERS AT START ON NTAPE FOR LATER USE.
  WRITE (NTAPE) IRUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,
  * (PUF,I=1,10)
  WRITE (NTAPE) STARTT, (FMI(I),I=1,NF), (XDD(I),I=1,NX),
  * (XDO(I),I=1,NX), (XO(I),I=1,NX)

C
C PRINT DATA AT START.
  NFL = NF/5
  IF ((NFL*5) .NE. NF) NFL = NFL+1
  IF (MINI .NF. 4HMINI) GO TO 70
  IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 70
  IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 70
  WRITE (NOT,2250)

```

B=B  
B=AI

A=AI

B=B  
B=AIB  
C=AIC  
D=AID

```

NLINE = NLINE + 2
GO TO 71
70 CALL PAGEHD
71 WRITE (NOT,2040) STARTT
WRITE (NOT,2050) (FM1(I), I=1,NF)
NLINE = NLINE + 3 + 3 + NFL
NXS = 1
NXE = NX
NFLN = (NF-1)/5+1
IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
82 WRITE (NOT,2060) (I,XOD(I),XDO(I),XO(I), I=NXS,NXE)
NLINE = NLINE + 4 + (NXE-NXS+1)
IF (NX .EQ. NXE) GO TO 83
NXS = NXE + 1
NXE = NX
IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
CALL PAGEHD
GO TO 82
83 NW = 1

```

C  
C CALCULATE SCALAR CONSTANTS FOR INTEGRATION.

```

C1 = DELTAT / 2.
C2 = (.5 - BETA) * DELTAT**2
C3 = (.25- BETA) * DELTAT**3
C4 = BETA * DELTAT**2
C5 = 1./DELTAT
C6 = 1./DELTAT**2
C7 = -2. + 1./BETA
C8 = (1. - 2.*BETA) * DELTAT**2

```

C  
C CALCULATE AT START TIME + DELTA TIME.

```

T = STARTT + DELTAT
FWDGPD = VEL*(T-STARTT)
DO 95 I=1,NF
F(I) = 0.0
GPD = FWDGPD - (PP(I)-FWDPP)
IF (GPD.GT.0.0 .AND. GPD.LT.GL)F(I)=FMAG(I)*(1.-COS(GPD*TPIGL))/2.
95 CONTINUE
REWIND NUT1
WRITE (NUT1) ((B(I,J), I=1,NX), J=1,NX)
WRITE (NUT1) ((D(I,J), I=1,NX), J=1,NF)
WRITE (NUT1) ((C(I,J), I=1,NX), J=1,NX)
CALL MULT (B,C,A,NX,NX,NX,KA,KA)
DO 101 I=1,NX
DO 100 J=1,NX
100 A(I,J) = C1*B(I,J) - C2*C(I,J) - C3*A(I,J)
101 A(I,I) = 1. + A(I,I)
DO 111 I=1,NX
DO 110 J=1,NX
110 B(I,J) = C1*B(I,J) + C4*C(I,J)
111 B(I,I) = 1. + B(I,I)
CALL INVI (B,C,NX,KA)
WRITE (NUT1) ((C(I,J), I=1,NX), J=1,NX)
CALL MULTB (C,A,NX,NX,NX,KA,KA)
CALL MULT (A,XO,XM1,NX,NX,1,KA,KA)

```

B=AIB  
D=AID  
C=AIC  
A=AIBAIC

A=P

B=S  
C=SI

A=SIP  
XM1=AXO

```

REWIND NUT1
READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
CALL MULT (B,B,A,NX,NX,NX,KA,KA)
DO 121 I=1,NX
DO 120 J=1,NX
120 A(I,J) = -C3*A(I,J)
121 A(I,I) = DELTAT + A(I,I)
CALL MULTB (C,A,NX,NX,NX,KA,KA)
CALL MULT (A,XD0,XM2,NX,NX,1,KA,KA)
CALL MULTB (C,D,NX,NX,NF,KA,KA)
CALL MULT (D,F,FM2,NX,NF,1,KA,KF)
DO 131 I=1,NX
DO 130 J=1,NX
130 B(I,J) = C3*B(I,J)
131 B(I,I) = C2 + B(I,I)
CALL MULTB (C,B,NX,NX,NX,KA,KA)
READ (NUT1) ((D(I,J), I=1,NX), J=1,NF)
CALL MULTB (B,D,NX,NX,NF,KA,KA)
CALL MULT (D,FM1,X,NX,NF,1,KA,KF)
DO 140 I=1,NX
X(I) = XM1(I) + XM2(I) + C4*FM2(I) + X(I)
140 XD(I) = C5 * (X(I) - X0(I))
REWIND NUT1
READ (NUT1) ((B(I,J), I=1,NX), J=1,NX)
READ (NUT1) ((D(I,J), I=1,NX), J=1,NF)
READ (NUT1) ((C(I,J), I=1,NX), J=1,NX)
DO 146 I=1,NX
SS = ZERO
DO 144 J=1,NF
S = D(I,J)*F(J)
144 SS = SS + S
146 XDD(I) = SS
DO 149 I=1,NX
SS = ZERO
DO 148 J=1,NX
S = B(I,J)*XD(J)
SS = SS + S
S = C(I,J)*X(J)
148 SS = SS + S
149 XDD(I) = XDD(I) - SS
C
C CALCULATE CONSTANT COEFFICIENT MATRICES FOR TIME T2,T3,ETC.
DO 151 I=1,NX
DO 150 J=1,NX
150 B(I,J) = -C8*C(I,J)
151 B(I,I) = 2. + P(I,I)
READ (NUT1) ((A(I,J), I=1,NX), J=1,NX)
CALL MULTB (A,B,NX,NX,NX,KA,KA)
REWIND NUT1
READ (NUT1) ((A(I,J), I=1,NX), J=1,NX)
DO 161 I=1,NX
DO 160 J=1,NX
160 C(I,J) = -C1*A(I,J) + C4*C(I,J)
161 C(I,I) = 1. + C(I,I)
READ (NUT1)

```

E=AIB  
A=AIBAIB

A=Q  
A=SIQ  
XM2=AXD0  
D=SI AID  
FM2=DF1

B=R  
E=SIR  
D=AID  
D=SI R AID  
X=CF0

X=XI  
XD=XDI

B=AIB  
D=AID  
C=AIC

B=T  
A=SI  
B=SI T

A=AIB

C=U



A= I  
C=SIU  
D=SIAID

```

READ (NUT1)
READ (NUT1) ((A(I,J), I=1,NX), J=1,NX)
CALL MULTR (A,C,NX,NX,NX,KA,KA)
CALL MULTP (A,D,NX,NX,NF,KA,KA)
DO 180 I=1,NX
DO 180 J=1,NF
180 D(I,J) = C4*D(I,J)
DO 185 I=1,NX
185 XM1(I) = X0(I)
C
C CALCULATE X,XD,XDD FOR TIME = T2,T3,ETC.
DO 399 ITP=2,NTP
IF (ITP .EQ. 2) GO TO 340
DO 191 I=1,NX
XM2(I) = XM1(I)
191 XM1(I) = X (I)
DO 192 I=1,NF
FM2(I) = FM1(I)
192 FM1(I) = F (I)
T = STARTT + FLOAT(ITP-1)*DELTAT
FWDGPD = VEL*(T-STARTT)
DO 195 I=1,NF
F(I) = 0.0
GPD = FWDGPD - (PP(I)-FWDPP)
IF (GPD.GT.0.0 .AND. GPD.LT.GL)F(I)=FMAG(I)*(1.-COS((PD*TPIGL)))/2.
195 CONTINUE
DO 201 J=1,NF
201 FM2(J) = F(J) + C7*FM1(J) + FM2(J)
DO 206 I=1,NX
SS = ZERO
DO 204 J=1,NF
S = D(I,J)*FM2(J)
204 SS = SS + S
206 X(I) = SS
DO 208 I=1,NX
SS = ZERO
DO 207 J=1,NX
S = B(I,J)*XM1(J)
SS = SS + S
S = C(I,J)*XM2(J)
207 SS = SS - S
208 X(I) = X(I) + SS
DO 209 I=1,NX
XD (I) = C5 * (X(I) - XM1(I))
209 XDD(I) = C6 * (X(I) - 2.*XM1(I) + XM2(I))
C
C WRITE ANSWERS ON NTAPE FOR LATER USE.
340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
* (X(I),I=1,NX)
C
C SEE IF DATA SHOULD BE PRINTED.
IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345
NFL = NF/5
IF ((NFL*5) .NE. NF) NFL = NFL+1
IF (MINI .NE. 4HMINI) GO TO 800

```

```

IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 800
WRITE (NCT,2250)
NLINE = NLINE + 2
GO TO 810
800 CALL PAGEHD
810 WRITE (NCT,2040) T
WRITE (NCT,2050) (F(I), I=1,NF)
NLINE = NLINE + 3 + 3 + NFL
NXS = 1
NXE = NX
NFLN = (NF-1)/5+1
IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
342 WRITE (NCT,2060) (I, XDD(I), XD(I), X(I), I=NXS,NXE)
NLINE = NLINE + 4 + (NXE-NXS+1)
IF (NX .EQ. NXE) GO TO 343
NXS = NXE + 1
NXE = NX
IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
CALL PAGEHD
GO TO 342
343 NW = 0
345 NW = NW+1
C
C SEE IF RUN HAS DIVERGED.
DO 350 I=1,NX
IF (ABS(X(I)) .GT. DIVTOL) GO TO 999
350 CONTINUE
C
399 CONTINUE
C
WRITE (NT,PE) BUF,BUF,EOT,(BUF,I=1,16)
ENDFILE NTAPE
RETURN
C
999 ENDFILE NTAPE
CALL ZZBOMB (6HTRSP2A,NERROR)
END

```

NERROR=2

```

SUBROUTINE TRSP3 (A,B,C,D,TABT,TABF,XDO,XO,STARTT,DELTAT,ENDT,
*           NWRITE,NX,NF,NTF,XNAME,KD,KF,NTAPE)
DIMENSION A(1),B(1),C(1),D(KD,1),TABT(KF,1),TABF(KF,1),
*           XDO(1),XO(1)
COMMON /LWRKV1/ XDD(250), XD(250)
COMMON /LWRKV2/ X(250), TERK(250)
COMMON /LWRKV3/ AIDFO(250), AIDF(250)
COMMON /LWRKV4/ F(500)
COMMON /LSTART/ IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
COMMON /LLINE/ NLINE,MAXLIN,MINI
DOUBLE PRECISION S,SS,ZERO
DATA ZERO/0.0/
DATA NIT,NOT/5,6/
DATA NLPP,BUF,DIVTOL, ECT/
*           54 , 0.,1.E+35,3HEOT/

```

```

C
C RESPONSE ROUTINE TO SOLVE THE SECOND ORDER DIFFERENTIAL EQUATION
C (A)XDD + (B)XD + (C)X = (D)F FOR XDD, XD, X.
C A, B, AND C ARE UNCOUPLED DIAGONAL MATRICES IN VECTOR FORM.
C CLOSED FORM SOLUTION IS USED TO FIND XDD, XD, AND X.
C VECTOR F IS OBTAINED BY LINEAR INTERPOLATION USING TABT,TABF.
C THE ANSWERS (T,F,XDD,XD,X) WILL BE WRITTEN ON NTAPE EVERY DELTAT AND
C ON PAPER EVERY NWRITE * DELTAT.
C NTAPE MUST HAVE BEEN INITIALIZED WITH SUBROUTINE INTAPE. A HEADER,
C TIME POINT DATA, AND END-OF-FILE WILL BE WRITTEN ON NTAPE HERE.
C COMMON /LSTART/ IS DEFINED IN SUBROUTINE START.
C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C CALLS FORMA SUBROUTINES PAGEHD,ZZBOMB.
C THE MAXIMUM SIZES ARE (BASED ON DIMENSIONS OF XDD,XD,X,F)
C   NX = 250
C   NF = 500
C THE MAXIMUM NUMBER OF UNIQUE TIMES PAST STARTT IN TABT = 250.
C CODED BY WA PENFIELD AND RL WOHLER. FEBRUARY 1967.
C LAST REVISION BY RL WOHLER, MARCH 1976.

```

```

C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX COEFFICIENT OF XDD. INPUT AS A VECTOR,
C   USED AS A DIAGONAL MATRIX. SIZE (NX). *DESTROYED*
C B = MATRIX COEFFICIENT OF XD. INPUT AS A VECTOR,
C   USED AS A DIAGONAL MATRIX. SIZE (NX). *DESTROYED*
C C = MATRIX COEFFICIENT OF X. INPUT AS A VECTOR,
C   USED AS A DIAGONAL MATRIX. SIZE (NX). *DESTROYED*
C D = MATRIX COEFFICIENT OF F. SIZE (NX,NF). *DESTROYED*
C TABT = TABLE OF TIMES FOR FORCE IN TABF. SIZE (NF,NTF).
C TABF = TABLE OF FORCES. SIZE (NF,NTF).
C XDO = VECTOR OF INITIAL VELOCITIES. SIZE (NX). *DESTROYED*
C XO = VECTOR OF INITIAL DISPLACEMENTS. SIZE (NX). *DESTROYED*
C STARTT = START TIME.
C DELTAT = INTEGRATION TIME INTERVAL.
C ENDT = END TIME.
C NWRITE = MULTIPLE OF INTEGRATION POINTS TO WRITE ON PAPER.
C           NWRITE = 1 WRITE EVERY POINT (1,2,3,...)
C           NWRITE = 2 WRITE EVERY SECOND POINT (1,3,5,...)
C           ETC
C NX = SIZE OF A, B, AND C (VECTORS). NUMBER OF ROWS IN D. MAX=250.

```

```

C NF      = NUMBER OF ROWS IN TABT,TABF. NUMBER OF COLS IN D. MAX=500.
C NTF     = NUMBER OF COLS IN TABT,TABF.
C XNAME   = IDENTIFICATION OF DATA TO BE WRITTEN ON NTAPE. (A6 FORMAT).
C KD      = ROW DIMENSION OF D IN CALLING PROGRAM.
C KF      = ROW DIMENSION OF TABT, TABF IN CALLING PROGRAM.
C NTAPE   = NUMBER OF TAPE ON WHICH ANSWERS WILL BE WRITTEN. (E.G. 10).
C
C THE OUTPUT DATA (TO BE WRITTEN ON PAPER AND NTAPE) IS
C T       = TIME .
C F       = FORCE OBTAINED BY LINEAR INTERPOLATION ON TABF. SIZE (NF).
C XDD     = ACCELERATION. SIZE (NX).
C XG      = VELOCITY. SIZE (NX).
C X       = DISPLACEMENT. SIZE (NX).
C AIDF    = A** -1 * D * F. SIZE (NX). (WRITTEN ON PAPER ONLY).
C
C ERROR EXPLANATION
C 1 = SIZE EXCEEDS DIMENSION.
C 2 = START TIME LESS THAN TABLE BOUNDS.
C 3 = END TIME GREATER THAN TABLE BOUNDS.
C 4 = MORE THAN 200 TIME BREAKS.
C
2001 FORMAT (////15X,42H THE INPUT SCALARS TO SUBROUTINE TRSP3 ARE ,
1          //23X,          10H STARTT = F10.6,
2          //23X,          10H DELTAT = F10.6,
3          //23X,          10H ENDT   = F10.6,
4          //23X,          10H NWRITE = 15      )
2040 FORMAT (//9X,8H TIME = F10.6)
2050 FORMAT (//9X,15H APPLIED FORCES / (10X, 5E16.8))
2060 FORMAT (// 9X,4H ROW, 6X,13H ACCELERATION, 8X,9H VELOCITY,
*          10X,13H DISPLACEMENT, 4X,19H A** -1 * D * FORCES //
*          (10X, 13, 4E20.8))
2250 FORMAT (/ 1X 123(1H-))
C
C NERROR=1
IF (NX .GT. 250 .OR. NF .GT. 500) GO TO 999
C
C PRINT INPUT SCALARS.
IF (MINI .NE. 4*MINI) GO TO 10
IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
IF ((NLINE+2+12) .GT. MAXLIN) GO TO 10
WRITE (NOT,2250)
NLINE = NLINE + 2
GO TO 11
10 CALL PAGEHD
11 WRITE (NOT,2001) STARTT,DELTAT,ENDT,NWRITE
NLINE = NLINE + 13
C
C SEARCH NTAPE FOR END OF WRITTEN DATA.
REWIND NTAPE
5 READ (NTAPE) PUFIN,EUFIN,IEGTCK,(BUFIN,J=1,5),NREC
IF (IEGTCK .EQ. 3HECT) GO TO 7
DO 6 IREC=1,NREC
6 READ (NTAPE)
GO TO 5
7 BACKSPACE NTAPE

```

CHECK TIME TABLE (TABT).

DO 18 I=1,NF

IF (STARTT .LT. TABT(I,1)) GO TO 999

DO 12 J=2,NTF

IF (TABT(I,J-1) .GE. TABT(I,J)) GO TO 14

12 CONTINUE

J = NTF+1

14 IF (ENDT .LE. TABT(I,J-1)) GO TO 18

GO TO 999

18 CONTINUE

NERROR=2

NERROR=3

C

C CALCULATE NUMBER OF TIME POINTS TO BE USED.

NTP = (ENDT-STARTT)/DELTAT + 1.1

C

C CALCULATE CONSTANTS.

DO 45 I=1,NX

DO 40 J=1,NF

40 D(I,J) = D(I,J)/A(I)

ASTORE = .5\*B(I)/A(I)

C(I) = C(I)/A(I)

B(I) = SQRT(C(I)-ASTORE\*\*2)

45 A(I) = ASTORE

D=AID

C

A = DAMP/(2.\*MASS)

C B = SQRT (STIF/MASS - (DAMP/(2.\*MASS))\*\*2)

C C = STIF/MASS = OMEGA\*\*2

C

C FIND UNIQUE TIME BREAKS (TBRK) IN TABT AFTER STARTT. MAX=250.

NTRK = 0

DO 55 I=1,NF

DO 54 J=2,NTF

IF (TABT(I,J) .LE. STARTT) GO TO 54

IF (TABT(I,J-1) .GE. TABT(I,J)) GO TO 55

IF (NTRK .EQ. 0) GO TO 52

DO 50 K=1,NTRK

IF (TABT(I,J) .EQ. TBRK(K)) GO TO 54

50 CONTINUE

52 NTRK = NTRK+1

IF (NTRK .LE. 250) GO TO 53

GO TO 999

53 TBRK (NTRK) = TABT(I,J)

54 CONTINUE

55 CONTINUE

DO 65 J=1,NTPRK

DO 60 I=J,NTRK

IF (TBRK(I) .GE. TBRK(J)) GO TO 60

TMIN = TBRK(I)

TBRK(I) = TBRK(J)

TBRK(J) = TMIN

60 CONTINUE

65 CONTINUE

NERROR=4

SET INITIAL VALUES.

WRITE (NTAPE) IRUNNO,XNAME,DATE,STARTT,DELTAT,ENDT,NX,NF,NTP,

\* (BUF,I=1,1

T = STARTT

TB = STARTT

NW = NWRITE

IB = 1

DO 86 I=1,NF

DO 84 J=1,NTF

IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 86

84 CONTINUE

86  $F(I) = TABF(I,J) + (T - TABT(I,J)) * (TABF(I,J+1) - TABF(I,J)) /$   
\*  $(TABT(I,J+1) - TABT(I,J))$

DO 88 I=1,NX

SS = ZERO

DO 87 J=1,NF

S = D(I,J)\*F(J)

87 SS = SS + S

88 AIDF(I) = SS

DO 90 I=1,NX

90 AIDFO(I) = AIDF(I)

DO 95 I=1,NX

X(I) = XO(I)

XD(I) = XDO(I)

95 XDD(I) = AIDFO(I) - 2.\*A(I)\*XD(I) - C(I)\*X(I)

C INTEGRATION LOOP.

DO 399 ITP=1,NTP

IF (ITP .EQ. 1) GO TO 340

TX = STARTT + FLOAT(ITP-1)\*DELTAT

105 T = TX

TMTB = T-TB

C

C SEE IF THERE IS A TIME BREAK (TB) IN TBRK BETWEEN PREVIOUS TIME  
C BREAK AND CURRENT TIME (T).

IF (IB .GT. NTERK) GO TO 399

IF (T .LT. TBRK(IB)) GO TO 110

T = TBRK(IB)

TMTB = T-TB

TB = T

IB = IB+1

110 DO 116 I=1,NF

DO 113 J=1,NTF

IF (T .LE. TABT(I,J+1) .OR. (J+1).EQ.NTF) GO TO 116

113 CONTINUE

116  $F(I) = TABF(I,J) + (T - TABT(I,J)) * (TABF(I,J+1) - TABF(I,J)) /$   
\*  $(TABT(I,J+1) - TABT(I,J))$

DO 118 I=1,NX

SS = ZERO

DO 117 J=1,NF

S = D(I,J)\*F(J)

117 SS = SS + S

118 AIDF(I) = SS

C

CALCULATE RESPONSE DUE TO DISPLACEMENT, VELOCITY, FORCE AT  
PREVIOUS TAFT BREAK AND DUE TO CURRENT RAMP.

```

DO 125 I=1,NX
  RS      = (AIDF(I)-AIDFO(I))/TMTB
  IF (C(I) .EQ. 0.0) GO TO 120
  BT      = B(I)*TMTB
  S1      = SIN(BT)
  C1      = COS(BT)
  AS1     = A(I)*S1
  BC1     = B(I)*C1
  EAB     = EXP(-A(I)*TMTB)/B(I)
  XOEAB   = X0(I)*EAB
  XDOEAB  = XDO(I)*EAB
  X(I)    = XOEAB*(AS1+BC1) + XDOEAB*S1
  1      + AIDFO(I)*(1.-EAB*(AS1+BC1))/C(I)
  2      + RS*(TMTB + (-2.*A(I)+EAB*((A(I)**2-B(I)**2)*S1
  3      + 2.*A(I)*BC1))/C(I) )/C(I)
  XD(I)   = -XOEAB*C(I)*S1 + XDOEAB*(-AS1+BC1) + AIDFO(I)*EAB*S1
  1      + RS*(1.-EAB*(AS1+BC1))/C(I)
  XDD(I)  = XOEAB*C(I)*(AS1-BC1)
  1      + XDOEAB*((A(I)**2-B(I)**2)*S1-2.*A(I)*BC1)
  2      + AIDFO(I)*EAB*(-AS1+BC1) + RS*EAB*S1
  GO TO 125
120 X(I)  = X0(I) + XDO(I)*TMTB + .5*AIDFO(I)*TMTB**2 + RS*TMTB**3/6.
  XD(I)  = XDO(I) + AIDFO(I)*TMTB + .5*RS*TMTB**2
  XDD(I) = AIDFO(I) + RS*TMTB
125 CONTINUE
  IF (T .GT. TB) GO TO 340
  DO 130 I=1,NX
  X0(I)  = X(I)
  XDO(I) = XD(I)
130 AIDFO(I) = AIDF(I)
  IF (T .LT. TX) GO TO 105
C
C WRITE ANSWERS ON NTAPE FOR LATER USE.
340 WRITE (NTAPE) T, (F(I),I=1,NF), (XDD(I),I=1,NX), (XD(I),I=1,NX),
  *      (X(I),I=1,NX)
C
C SEE IF DATA SHOULD BE PRINTED.
  IF (ITP.LT.NTP .AND. NW.LT.NWRITE) GO TO 345
  NFL = NF/5
  IF ((NFL*5) .NE. NF) NFL = NFL+1
  IF (MIN1 .NE. 4*MINT) GO TO 800
  IF (NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 800
  IF ((NLINE+2+3+3+NFL+4+NX) .GT. MAXLIN) GO TO 800
  WRITE (NOT,2250)
  NLINE = NLINE + 2
  GO TO P10
800 CALL PAGEHD
810 WRITE (NOT,2040) T
  WRITE (NOT,2050) (F(I), I=1,NF)
  NLINE = NLINE + 3 + 3 + NFL
  NXS = 1
  NXE = NX
  NFLN = (NF-1)/5+1

```

```
IF ((NXE + NFLN) .GT. (NLPP-15)) NXE=(NLPP-15)-NFLN
342 WRITE (NOT,2060) (I, XDD(I), XD(I), X(I), AIDF(I), I=NXS,NXE)
NLIN = NLIN + 4 + (NXE-NXS+1)
IF (NX .EQ. NXE) GO TO 343
NXS = NXE + 1
NXE = NX
IF ((NXE-NXS) .GT. (NLPP- 9)) NXE=NXS+(NLPP- 9)
CALL PAGEHD
GO TO 342
343 NW = 0
345 NW = NW+1
C
399 CONTINUE
C
WRITE (NTAPE) BUF, BUF, ECT, (BUF, I=1, 16)
ENDFILE NTAPE
RETURN
C
999 ENDFILE NTAPE
CALL ZZBOMB (6HTRSP3 ,NERRDR)
END
```



UMAMI

-----

SUBROUTINE UMAMI (A,RBM,UMAM,N,NRBM,K)  
 DIMENSION A(K,1), RBM(K,1), UMAM(K,1), W1(6,6), W2(6,6)

C  
 C GENERATE TRANSFORMATION RELATING INERTIA PLUS APPLIED LOADS TO  
 C APPLIED LOADS FOR INERTIALLY RESTRAINED SYSTEM.  
 C CALLS FORMA SUBROUTINES BABT,BTAB,INVI,MULTB,PAGEHD,ZZBOMB.  
 C THE MAXIMUM SIZES ARE  
 C N = 500 (BASED ON BTAB,MULTB)  
 C NRBM = 6  
 C DEVELOPED BY CARL BODLEY. JANUARY 1965.  
 C LAST REVISION BY WA BENFIELD. MARCH 1976.  
 C  
 C SUBROUTINE ARGUMENTS  
 C A = INPUT MASS MATRIX. SIZE(N,N).  
 C RBM = INPUT MATRIX OF RIGID BODY MODES, (NEEDNT BE ORTHORNORMAL)  
 C SIZE(N,NRBM).  
 C UMAM = OUTPUT (UNITY MINUS A MESS). SIZE(N,N).  
 C N = INPUT SIZE OF SYSTEM (NUMBER OF COORDINATES).  
 C NRBM = INPUT NUMBER OF RIGID BODY MODES. MAX=6.  
 C K = INPUT ROW DIMENSION OF A,RBM,UMAM IN CALLING PROGRAM.

C  
 C NERROR EXPLANATION  
 C 1 = MORE THAN 6 RIGID BODY MODES.

NERROR=1

IF (NRBM .GT. 6) GO TO 999

CALL BTAB (A,RBM,W1,N,NRBM,K,6)  
 CALL INVI (W1,W2,NRBM,6)  
 CALL PART (W2,RBM,UMAM, N,NRBM, 6,K)  
 CALL MULTB (A,UMAM, N,N,N, K,K)  
 DO 60 I=1,N  
 DO 50 J=1,N  
 50 UMAM(I,J) = -UMAM(I,J)  
 60 UMAM(I,I) = 1.0 + UMAM(I,I)  
 RETURN

C  
 999 CALL ZZBOMB (6HUMAMI ,NERROR)  
 END

UNITY

SUBROUTINE UNITY (Z,N,KR)  
DIMENSION Z(KR,1)

C  
C GENERATE A UNITY MATRIX. (ONES ON THE DIAGONAL).  
C CODED BY RL WOHLN. FEB 1965.  
C  
C SUBROUTINE ARGUMENTS  
C Z = OUTPUT MATRIX GENERATED. SIZE(N,N).  
C N = INPUT SIZE OF MATRIX Z (SQUARE).  
C KR = INPUT ROW DIMENSION OF MATRIX Z IN CALLING PROGRAM.  
C  
DO 20 I=1,N  
DO 10 J=1,N  
10 Z(I,J) = 0.0  
20 Z(I,I) = 1.0  
RETURN  
END

SUBROUTINE UPDATE

C UPDATE TAPE PROGRAM (PROGRAMMED TO WORK WITH DISK UNITS)  
 C MAXIMUM SIZE = 40000 ELEMENTS FOR A DENSE MATRIX.  
 C CALLS FORMA SUBROUTINES ... NONE USED.  
 C CODED BY RF HRUDA. APRIL 1969. REVISED NOVEMBER 1970.  
 C MODIFIED FOR CONTRACT NAS8-25922, MAY 1971.

C INPUT

---

C CARD 1 = IFINIT,TAPEID,NT1,NT2                   FORMAT (2A6,I3,I5)  
 C CARD 2 = LNS,LNE                                    FORMAT (2I5)  
 C CARD N = 10 ZEROS (REST OF CARD BLANK)       FORMAT (I10)

C VARIABLES

---

C IFINIT = INITIL , NT2 WILL BE INITIALIZED AND UPDATE WILL  
 C                    START AT BEGINNING OF NT2.  
 C           = NOINIT , UPDATE WILL BEGIN AT END OF DATA ON NT2.  
 C           = STOP , PROGRAM WILL BE STOPPED.  
 C TAPEID = TAPE I.D. FOR TAPE THAT IS TO BE INITIALIZED (EG T1234).  
 C NT1    = LOGICAL NUMBER OF THE TAPE TO BE READ FROM (EG 11).  
 C NT2    = LOGICAL NUMBER OF THE TAPE TO BE WRITTEN ON (EG 10).  
 C LNS    = STARTING LOCATION NUMBER OF AN UPDATE BLOCK.  
 C LNE    = ENDING LOCATION NUMBER OF THE UPDATE BLOCK.  
 C           = 0, INDICATES END OF TAPE.  
 C           (MATRICES WITH LOCATION NUMBERS LNS THRU LNE WILL BE UPDATED)

C EXAMPLE OF INPUT DATA FOR SUBROUTINE UPDATE.

---

CARD	COLUMN	NUMBER	COMMENTS.
12345678901234567890	1	2	
INITILT1234	11	10	UPDATE FROM 11 ONTO 10 AND INITIAL 10 AS T1234.
5	9		UPDATE FROM MATRIX 5 THRU MATRIX 9,
2	2		UPDATE ONLY MATRIX 2,
19	0		UPDATE FROM MATRIX 19 THRU END OF TAPE10.
0000000000			RETURN TO CALLING PROGRAM.
NOINIT	12	10	UPDATE FROM 12 ONTO 10 (IF REQD)
14	24		
3	7		
9	12		
0000000000			
STOP			
BETA-CARD.			

C  
C  
C  
C

```

DIMENSION A(40000)
DATA NIT,NOT/5,6/
DATA Z,NSMAX, EOT, IDENSE/
* 0.,40000,3HEOT,5HDENSE/
DATA NLPP/54/

```

C

```

1001 FORMAT (2A6,I3,I5)
1002 FORMAT (2I5)
2001 FORMAT (1H) 47X 6HUPDATE 2IX 6HPAGE NO. I3)
2002 FORMAT (/ 26X35HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6)
2003 FORMAT (/ 20X35HLISTING OF MATRICES ON LOGICAL UNITI3,7H, TAPE A6,
* 12H (CONTINUED))
2004 FORMAT ( 26X 5I(1H-) / 27X3HNO.3X7HRUN NO.4X4HNAME5X5HNROWS
* 4X5HNCOLS4X4HDATE/
* 27X3H---3X6H-----4X6H-----4X5H-----
* 4X5H-----3X6H----- /)
2005 FORMAT ( 25XI5,3XA6,4XA6,3XI5,4XI5,4XA6)
2006 FORMAT (/ 27X 14HEND OF UPDATE.)
2007 FORMAT (//27X 41HTHE FOLLOWING DATA WAS UPDATED FROM TAPE A6 /)
2008 FORMAT ( 20X 63(1H-) / 27X3HNO.3X7HRUN NO.4X4HNAME5X5HNROWS
* 4X5HNCOLS4X4HDATE/
* 27X3H---3X6H-----4X6H-----4X5H-----
* 4X5H-----3X6H----- /)
3001 FORMAT (//27X 42HTHE FOLLOWING INPUT DATA WAS NOT EXECUTED,
* / 32X 5HLNS = I5,5X 5HLNE = I5,
* / 27X 35HUPDATE CONTINUES FOR REST OF INPUT./)
3002 FORMAT (//27X 42HMAX SIZE EXCEEDED IN THE FOLLOWING MATRIX,
* / 25X I5,3XA6,4XA6,3XI5,4XI5,4XA6,
* / 27X 28HTHIS MATRIX WAS NOT UPDATED.
* / 27X 38HUPDATE CONTINUES FOR REST OF MATRICES./)

```

C  
C  
C

```

C READ IN HEADER CARD AND INITIALIZE CONSTANTS.
READ (NIT,1001) IFINIT,TAPEID,NT1,NT2
IF (IFINIT .EQ. 4HSTOP) STOP
LN1 = 0
LN2 = 1
NPAGE = 1
WRITE (NOT,2001) NPAGE

```

C

```

C REWIND TAPES, DEFINE NT1 TAPEID, AND INITIALIZE NT2 IF NECESSARY.
REWIND NT1
REWIND NT2
READ (NT1) T1
REWIND NT1
IF (IFINIT .NE. 6HINITIL) GO TO 111
WRITE (NT2) TAPEID,LN2,EOT,(Z,I=1,16)
REWIND NT2

```

```

LIST ANY EXISTING MATRICES ON T2.
111 READ (NT2) T2
    REWIND NT2
    WRITE (NOT,2002) NT2,T2
    WRITE (NOT,2004)
    NLINE = 1
    GO TO 113
112 NPAGE = NPAGE+1
    WRITE (NOT,2001) NPAGE
    WRITE (NOT,2003) NT2,T2
    WRITE (NOT,2008)
    NLINE = 1
113 READ (NT2) T2,LN2,IEOTCK,IRUNNO,ANAME,NR,NC,IDATE
    IF (IEOTCK .EQ. 3HEOT) GO TO 116
    READ (NT.)
    IF (IRUNNO .EQ. ICHK) GO TO 115
    ICHK = IRUNNO
    NLINE = NLINE+1
    WRITE (NOT,1001)
115 WRITE (NOT,2005) LN2,IRUNNO,ANAME,NR,NC,IDATE
    NLINE = NLINE+1
    IF(NLINE .GT. (NLPP-7)) GO TO 112
    GO TO 113
116 BACKSPACE NT2
    WRITE (NOT,2007) T1
    NLINE = NLINE+4
C
C
C
C READ IN DATA CARD AND POSITION T1.
200 READ (NT1,1002) LNS,LNE
    IF (LNS.EQ.0) GO TO 500
    IF (LNS.LT.0) GO TO 401
    LNE1 = LNE
    IF (LNE1.LT.LNS) LNE1 = 9000
    NMATS = LNE1-LNS+1
    IF (LNS.EQ.LN1+1) GO TO 300
    IF (LNS.GT.LN1+1) GO TO 201
    REWIND NT1
    LN1 = 0
201 NUM = LNS-LN1-1
    IF (NUM.EQ.0) GO TO 300
    DO 202 NO=1,NUM
    READ (NT1) T1,LN1,IEOTCK
    IF (IEOTCK .EQ. 3HEOT) GO TO 401
202 READ (NT1)
C
C UPDATE A BLOCK OF NMATS MATRICES FROM T1 ONTO T2.
300 DO 305 N=1,NMATS
    IF(NLINE .LT. (NLPP-7)) GO TO 301
    NPAGE = NPAGE+1
    WRITE (NOT,2001) NPAGE
    WRITE (NOT,2003) NT2,T2
    WRITE (NOT,2008)

```

```
      NLINE = 1
301 READ (NT1) T1, LN1, IEOTCK, IRUNNO, ANAME, NR, NC, IDATE
      IF (IEOTCK .EQ. 3HEOT) GO TO 200
      IF (ICLK .EQ. IRUNNO) GO TO 302
      NLINE = NLINE+1
      WRITE (NOT,1001)
      ICHK = IRUNNO
302 NS = NR*NC
      IF (NS.GT.NSMAX) GO TO 304
      WRITE (NT2) T2, LN2, Z, IRUNNO, ANAME, NR, NC, IDATE, IDENSE, (Z, I=1, 10)
      READ (NT1) (A(I), I=1, NS)
      WRITE (NT2) (A(I), I=1, NS)
      WRITE (NOT,2005) LN2, IRUNNO, ANAME, NR, NC, IDATE
      LN2 = LN2+1
      NLINE = NLINE+1
      GO TO 305
304 WRITE (NOT,3002) LN1, IRUNNO, ANAME, NR, NC, IDATE
      READ (NT1)
      NLINE = NLINE+8
305 CONTINUE
      GO TO 200

C
C
C
C ERROR MESSAGE.
401 WRITE (NOT,3001) LNS, LNE
      REWIND NT1
      LN1 = 0
      NLINE = NLINE+5
      GO TO 200

C
C END OF UPDATE.
500 WRITE (NT2) T2, LN2, EOT, (Z, I=1, 16)
      ENDFILE NT2
      REWIND NT2
      WRITE (NOT,1001)
      WRITE (NOT,2005) LN2, EOT
      WRITE (NOT,2006)
      RETURN
      END
```

```

SUBROUTINE UTAU1(A,U,Z,N,KRA,KRU,KRZ)
DIMENSION A(KRA,1),U(KRU,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.0/

```

```

C
C UTAU1 PERFORMS THE OPERATION (Z)=((U)TRANSPOSE)*(A)*(U)
C WHERE U IS AN UPPER TRIANGULAR MATRIX .
C UTAU1 CAN ALSO PERFORM THE OPERATION
C (A)=((U)TRANSPOSE)*(A)*(U) BY CALL UTAU1(A,U,A,--ETC--) .
C

```

```

C IF N IS NEGATIVE A SYMMETRIC (Z) IS COMPUTED.
C

```

```

C MAXIMUM SIZE N=500
C

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C

```

```

C FORMA SUBROUTINE ZZFORM IS CALLED.
C CODED BY JOHN ADMIRE *NASA* AUG 1972 .
C LAST REVISION BY RL WOHLER. APRIL 1976.
C

```

ARGUMENTS

A	- INPUT	MATRIX (A)	SIZE(N BY N)
U	- INPUT	UPPER TRIANGULAR MATRIX (U)	SIZE(N BY N)
Z	- OUTPUT	MATRIX (Z)	SIZE(N BY N)
N	- INPUT	ABS(N): NUMBER OF ROWS AND COLUMNS IN (A), (U) AND (Z)	
KRA	- INPUT	ROW DIMENSION OF (A) IN CALLING PROGRAM	
KRU	- INPUT	ROW DIMENSION OF (U) IN CALLING PROGRAM	
KRZ	- INPUT	ROW DIMENSION OF (Z) IN CALLING PROGRAM	

NERROR EXPLANATIONS

```

1 = SIZE EXCEEDANCE.

```

```

NN=IABS(N)

```

```

NERROR = 1

```

```

IF(NN .GT. 500 .OR. NN .GT. KRA .OR. NN .GT. KRU
* .OR. NN .GT. KRZ) GO TO 999

```

```

DO 30 J=1,NN
DO 10 K=1,NN
10 V(K)=A(I,K)
DO 30 J=1,NN
SUM=ZERO
DO 20 K=1,J
SS=V(K)*U(K,J)
20 SUM=SUM+SS
30 Z(I,J)=SUM
IF(N .GT. 0) GO TO 70
DO 60 J=1,NN
DO 40 K=1,J
40 V(K)=Z(K,J)
DO 60 I=1,J
SUM=ZERO
DO 50 K=1,1
SS=U(K,I)*V(K)
50 SUM=SUM+SS

```

```

60 Z(I,J)=SUM
   DO 63 I=1,NN
   DO 63 J=I,NN
63 Z(J,I)=Z(I,J)
   RETURN
70 DO 100 J=1,NN
   DO 80 K=1,NN
80 V(K)=Z(K,J)
   DO 100 I=1,NN
   SUM=ZERO
   DO 90 K=1,I
   SS=U(K,I)*V(K)
90 SUM=SUM+SS
100 Z(I,J)=SUM
   RETURN
999 CALL ZZBOMB(6HUTAU1 ,NERROR)
   END

```



```

SUBROUTINE UTAUC1(A,U,C,Z,N,KRA,KRU,KRC,KRZ)
DIMENSION A(KRA,1),U(KRU,1),C(KRC,1),Z(KRZ,1)
COMMON / LWRKV1 / V(500)
DOUBLE PRECISION SUM,SS,ZERO
DATA ZERO /0.0/

```

```

C
C UTAUC1 PERFORMS THE OPERATION (Z)=((U)TRANPOSE)*(A)*(U)+(C)
C WHERE (U) IS AN UPPER TRIANGULAR MATRIX .
C UTAUC1 CAN ALSO PERFORM THE OPERATIONS
C (A)=((U)TRANPOSE)*(A)*(U)+(C) BY CALL UTAUC1(A,U,C,A,--ETC--)
C (C)=((U)TRANPOSE)*(A)*(U)+(C) BY CALL UTAUC1(A,U,C,C,--ETC--) .
C

```

```

C IF N IS NEGATIVE A SYMMETRIC (Z) IS COMPUTED.
C

```

```

C MAXIMUM SIZE N=500
C

```

```

C INNER PRODUCT SUMS ARE PERFORMED IN DOUBLE PRECISION.
C

```

```

C FORMA SUBROUTINE ZZEOME IS CALLED.
C CODED BY JOHN ADMIRE *NASA* AUG 1972 .
C LAST REVISION BY RL WOHLER. APRIL 1976.
C

```

```

C
C ARGUMENTS
C A - INPUT MATRIX (A) *DESTROYED* SIZE(N BY N)
C U - INPUT UPPER TRIANGULAR MATRIX (U) SIZE(N BY N)
C C - INPUT MATRIX (C) SIZE(N BY N)
C Z - OUTPUT MATRIX (Z) SIZE(N BY N)
C N - INPUT ABS(N) NUMBER OF ROWS AND COLUMNS IN(A),(U),(C) AND (Z)
C KRA - INPUT ROW DIMENSION OF (A) IN CALLING PROGRAM
C KRU - INPUT ROW DIMENSION OF (U) IN CALLING PROGRAM
C KRC - INPUT ROW DIMENSION OF (C) IN CALLING PROGRAM
C KRZ - INPUT ROW DIMENSION OF (Z) IN CALLING PROGRAM
C

```

NERPOR EXPLANATIONS

```

C I = SIZE EXCEEDANCE.
C

```

```

C NN=IABS(N)
C

```

NERROP = 1

```

IF(NN .GT. 500 .OR. NN .GT. KRA .OR. NN .GT. KRU
* .OR. NN .GT. KRC .OR. NN .GT. KRZ) GO TO 999
DO 30 I=1,NN
DO 10 K=1,NN
10 V(K)=A(I,K)
DO 30 J=1,NN
SUM=ZERO
DO 20 K=1,J
SS=V(K)*U(K,J)
20 SUM=SUM+SS
30 A(I,J)=SUM
IF(N .GT. 0) GO TO 70
DO 60 J=1,NN
DO 40 K=1,J
40 V(K)=A(K,J)
DO 60 I=1,J
SUM=C(I,J)

```

```

    DO 50 K=1,I
    SS=U(K,I)*V(K)
50 SUM=SUM+SS
60 Z(I,J)=SUM
    DO 63 I=1,NN
    DO 63 J=I,NN
63 Z(J,I)=Z(I,J)
    RETURN
70 DO 100 J=1,NN
    DO 80 K=1,NN
80 V(K)=A(K,J)
    DO 100 I=1,NN
    SUM=C(I,J)
    DO 90 K=1,I
    SS=U(K,I)*V(K)
90 SUM=SUM+SS
100 Z(I,J)=SUM
    RETURN
999 CALL ZZBOMB(6HUTAUC1,NERROR)
    END

```

VCROSS

SUBROUTINE VCROSS (VA,VB, VZ,VAMAG,VBMAG,VZMAG,SINAB)  
DIMENSION VA(3),VB(3),VZ(3)

C  
C VECTOR (3-DIMENSIONAL) CROSS PRODUCT. (VA)CROSS(VB) = (VZ).  
C CODED BY RF HRUDA. OCTOBER 1966.

C  
C SUBROUTINE ARGUMENTS  
C VA = INPUT VECTOR A.  
C VB = INPUT VECTOR B.  
C VZ = OUTPUT VECTOR Z.  
C VAMAG = OUTPUT MAGNITUDE OF VA.  
C VB MAG = OUTPUT MAGNITUDE OF VB.  
C VZ MAG = OUTPUT MAGNITUDE OF VZ.  
C SINAB = OUTPUT SINE OF THE ANGLE BETWEEN VA AND VB.

C  
C VZ(1) = VA(2)\*VB(3)-VA(3)\*VB(2)  
C VZ(2) = VA(3)\*VB(1)-VA(1)\*VB(3)  
C VZ(3) = VA(1)\*VB(2)-VA(2)\*VB(1)

C  
C SA = 0.0  
C SB = 0.0  
C SZ = 0.0  
C DO 10 I=1,3  
C SA = SA + VA(I)\*\*2  
C SB = SB + VB(I)\*\*2  
10 SZ = SZ + VZ(I)\*\*2  
C VAMAG = SQRT(SA)  
C VB MAG = SQRT(SB)  
C VZ MAG = SQRT(SZ)  
C IF (VAMAG.LT.1.E-30 .OR. VB MAG.LT.1.E-30) GO TO 20  
C SINAB = VZMAG/(VAMAG\*VB MAG)  
C IF (SINAB.GT.+1.0) SINAB = +1.0  
C IF (SINAB.LT.-1.0) SINAB = -1.0  
C RETURN

C  
20 SINAB = 0.  
C RETURN  
C END

VDOT

```

C
C      SUBROUTINE VDOT (VA,VB, PRODC, VAMAG, VBMAG, COSAB)
C      DIMENSION VA(3),VB(3)
C
C      VECTOR (3-DIMENSIONAL) DOT PRODUCT. (VA)DOT(VB) = PRODC.
C      CODED BY RF HRUDA. OCTOBER 1968.
C
C      SUBROUTINE ARGUMENTS
C      VA      = INPUT VECTOR A.
C      VB      = INPUT VECTOR B.
C      PRODC   = OUTPUT SCALAR PRODUCT OF (VA)DOT(VB).
C      VAMAG   = OUTPUT MAGNITUDE OF VA.
C      VBMAG   = OUTPUT MAGNITUDE OF VB.
C      COSAB   = OUTPUT COSINE OF THE ANGLE BETWEEN VA AND VB.
C
C      SA      = 0.
C      SB      = 0.
C      PRODC   = 0.
C      DO 10 I=1,3
C      SA = SA + VA(I)**2
C      SB = SB + VB(I)**2
10  PRODC = PRODC + VA(I)*VB(I)
C      VAMAG = SQRT(SA)
C      VBMAG = SQRT(SB)
C      IF (VAMAG.LT.1.E-30 .OR. VBMAG.LT.1.E-30) GO TO 20
C      COSAB = PRODC/(VAMAG*VBMAG)
C      IF (COSAB.GT.+1.0) COSAB = +1.0
C      IF (COSAB.LT.-1.0) COSAB = -1.0
C      RETURN
C
C      20 COSAB = 0.
C      RETURN
C      END

```

```

SUBROUTINE VM1 (XVEC,DIS,CON,AMP,TDD,CONVRT,ZV,ZM,
*           NX,ND,NC,NA,NTDD,KDIS,KCON,KAMP,KTDD)
DIMENSION XVEC(1),DIS(KDIS,1),CON(KCON,1),AMP(KAMP,1),TDD(KTDD,1),
*           ZV(1),ZM(1)

```

```

C
C SUBROUTINE TO INTEGRATE PRESSURE OR WEIGHT DISTRIBUTION TO OBTAIN
C SHEAR AND MOMENT AT A SET OF PRESCRIBED STATIONS (XVEC). THE PRESSURE
C OR WT DISTRIBUTION IS AMPLIFIED BY AN AMPLIFICATION DISTRIBUTION (AMP).
C CONCENTRATED MASS ITEMS (CON) USE 2 AMPLIFICATION FUNCTIONS IN GENERAL,
C AMP ALWAYS AND TDD -(THETA DOUBLE DOT)- IN THE EVENT OF THERE BEING
C A NON-ZERO DISTANCE BETWEEN ATTACH POINT AND CG. OR IF THERE IS LOCAL
C CONCENTRATED INERTIA. IN ANY CASE, AMP AND TDD MUST ALWAYS BE DEFINED
C (FOR EXAMPLE - MAY BE UNITY OR ZERO IN COLUMNS 3 AND 4).

```

C NOTES...

- C 1) THE DISTRIBUTED DATA (DIS,AMP,AND TDD) MUST HAVE THEIR SEGMENT  
C LIMITS IN ASCENDING ORDER, THE SEGMENTS MUST NOT OVERLAP AND  
C MUST BE IN ASCENDING ORDER.
  - C 2) ON ANY INTERVAL WHERE DISTRIBUTED DATA IS NOT DEFINED  
C (GAPS BETWEEN SEGMENTS), THE VALUES ON THE INTERVAL ARE ASSUMED  
C TO BE ZERO.
  - C 3) THE CONCENTRATED ITEMS MAY BE SUPPLIED IN ANY ORDER (ROWWISE).
- C CALLS FORMA SUBROUTINE Z2BOMB.  
C CODED BY CARL BODLEY. AUGUST 1966.  
C LAST REVISION BY WA BENFIELD. MARCH 1976.

C  
C SUBROUTINE ARGUMENTS

C	XVEC	= INPUT	VECTOR OF STATIONS WHERE SHEAR AND MOMENT ARE DESIRED. SIZE(NX). STATIONS MUST BE IN ASCENDING ORDER.
C	DIS	= INPUT	MATRIX OF DISTRIBUTED WEIGHT (OR PRESSURE) STRAIGHT LINE SEGMENT DATA. SIZE(ND,4). SEE NOTES 1,2. COL 1 = X AT SEGMENT END 1. COL 2 = X AT SEGMENT END 2. COL 3 = WEIGHT AT SEGMENT END 1. COL 4 = WEIGHT AT SEGMENT END 2.
C	CON	= INPUT	MATRIX OF CONCENTRATED ITEM DATA. SIZE(NC,4). NOTE 3. COL 1 = ATTACH STATION OF ITEM. COL 2 = MASS OF ITEM. COL 3 = CENTER OF GRAVITY OF ITEM. COL 4 = MOMENT OF INERTIA ABOUT CG OF ITEM.
C	AMP	= INPUT	MATRIX OF DISTRIBUTED AMPLIFICATION STRAIGHT LINE SEGMENT DATA. SIZE(NA,4). SEE NOTES 1,2. COLUMNS ARE SIMILAR TO DIS.
C	TDD	= INPUT	MATRIX OF SUPPLEMENTARY DISTRIBUTED AMPLIFICATION STRAIGHT LINE SEGMENT DATA. SIZE(NTDD,4). NOTES 1,2. COLUMNS ARE SIMILAR TO DIS.
C	CONVRT	= INPUT	CONVERSION SCALAR. (MULTIPLIES COL 3,4 OF DIS AND COL 2,4 OF CON).
C	ZV	= OUTPUT	VECTOR OF SHEARS AT THE STATIONS XVEC. SIZE(NX).
C	ZM	= OUTPUT	VECTOR OF MOMENTS AT THE STATIONS XVEC. SIZE(NX).
C	NX	= INPUT	SIZE OF VECTORS XVEC, ZV, AND ZM.
C	ND	= INPUT	NUMBER OF SEGMENTS (ROWS) OF DIS.
C	NC	= INPUT	NUMBER OF CONCENTRATED ITEMS, (ROWS OF CON).
C	NA	= INPUT	NUMBER OF SEGMENTS (ROWS) OF AMP.
C	NTDD	= INPUT	NUMBER OF SEGMENTS (ROWS) OF TDD.
C	KDIS	= INPUT	ROW DIMENSION OF DIS IN CALLING PROGRAM.

```

C KCON = INPUT ROW DIMENSION OF CON IN CALLING PROGRAM.
C KAMP = INPUT ROW DIMENSION OF AMP IN CALLING PROGRAM.
C KTDD = INPUT ROW DIMENSION OF TOD IN CALLING PROGRAM.
C
C ERROR EXPLANATION
C 1 = NON-POSITIVE SIZES.
C 2 = STATIONS NOT IN ASCENDING ORDER.
C 3 = INCORRECT DATA IN MATRIX AMP.
C 4 = INCORRECT DATA IN MATRIX AMP.
C 5 = INCORRECT DATA IN MATRIX TOD.
C 6 = INCORRECT DATA IN MATRIX TOD.
C 7 = INCORRECT DATA IN MATRIX DIS.
C 8 = INCORRECT DATA IN MATRIX DIS.
C
      DO 10 K=1,NX
        ZV(K) = 0.0
10    ZM(K) = 0.0
                                     NERROR=1
      IF (NX .LE. 0 .OR. NA .LE. 0 .OR. NTDD .LE. 0) GO TO 999
      IF (NX .EQ. 1) GO TO 47
                                     NERROR=2
      DO 40 I=2,NX
        K = I - 1
        IF (XVEC(K) .GE. XVEC(I)) GO TO 999
40    CONTINUE
47                                     NERROR=3
      IF (AMP(1,1) .GE. AMP(1,2)) GO TO 999
      IF (NA .EQ. 1) GO TO 48
      NAM1 = NA - 1
                                     NERROR=4
      DO 30 I=1,NAM1
        K = I + 1
        IF (AMP(I,2) .GT. AMP(K,1) .OR. AMP(K,1) .GE. AMP(K,2)) GO TO 999
30    CONTINUE
48                                     NERROR=5
      IF (TDD(1,1) .GE. TDD(1,2)) GO TO 999
      IF (NTDD .EQ. 1) GO TO 49
      NTM1 = NTDD - 1
                                     NERROR=6
      DO 35 I=1,NTM1
        K = I + 1
        IF (TDD(I,2) .GT. TDD(K,1) .OR. TDD(K,1) .GE. TDD(K,2)) GO TO 999
35    CONTINUE
49 IF (ND .EQ. 0) GO TO 85
                                     NERROR=7
      IF (DIS(1,1) .GE. DIS(1,2)) GO TO 999
      IF (ND .EQ. 1) GO TO 51
      NDM1 = ND - 1
                                     NERROR=8
      DO 20 I=1,NDM1
        K = I + 1
        IF (DIS(I,2) .GT. DIS(K,1) .OR. DIS(K,1) .GE. DIS(K,2)) GO TO 999
20    CONTINUE
C
C DISTRIBUTED DATA.

```

```

51 I = 1
   J = 1
   K = 1
   VI = 0.0
   GI = 0.0
   XIM1 = XVEC(1)
   IF (XIM1 .GT. DIS(I,1)) XIM1 = DIS(I,1)
   IF (XIM1 .GT. AMP(J,1)) XIM1 = AMP(J,1)

```

C

```

50 XI = XVEC(K)
   IF (XI .GT. DIS(I,2) .AND. DIS(I,2) .GT. XIM1) XI = DIS(I,2)
   IF (XI .GT. DIS(I,1) .AND. DIS(I,1) .GT. XIM1) XI = DIS(I,1)
   IF (XI .GT. AMP(J,2) .AND. AMP(J,2) .GT. XIM1) XI = AMP(J,2)
   IF (XI .GT. AMP(J,1) .AND. AMP(J,1) .GT. XIM1) XI = AMP(J,1)
   F = ((DIS(I,4)-DIS(I,3))/(DIS(I,2)-DIS(I,1)))*CONVRT
   E = CONVRT*DIS(I,3) - F*DIS(I,1)
   H = (AMP(J,4)-AMP(J,3))/(AMP(J,2)-AMP(J,1))
   G = AMP(J,3) - H*AMP(J,1)
   DX = XI - XIM1
   A = 0.0
   B = 0.0
   C = 0.0
   D = 0.0
   IF (DIS(I,1) .LE. XIM1 .AND. DIS(I,2) .GE. XI) A=E+F*XIM1
   IF (DIS(I,1) .LE. XIM1 .AND. DIS(I,2) .GE. XI) B=F*DX
   IF (AMP(J,1) .LE. XIM1 .AND. AMP(J,2) .GE. XI) C=G+H*XIM1
   IF (AMP(J,1) .LE. XIM1 .AND. AMP(J,2) .GE. XI) D=H*DX
   GI = GI+VI*DX+DX**2*(A*C/2.+(A*D+B*C)/6. +B*D/12.)
   VI = VI + DX*(A*C + (A*D + B*C)/2. + B*D/3.)
   ZV(K) = VI
   ZM(K) = GI
   IF (XI .EQ. XVEC(NX)) GO TO 85
   IF (XI .EQ. XVEC(K)) K=K+1
   IF (XI .EQ. DIS(I,2) .AND. I+1 .LE. ND) I = I+1
   IF (XI .EQ. AMP(J,2) .AND. J+1 .LE. NA) J = J+1
   XIM1 = XI
   GO TO 50

```

C

C

```

CONCENTRATED MASS ITEMS.
85 IF (NC .EQ. 0) RETURN
   DO 102 I=1,NC
   DO 90 J=1,NX
   IF (XVEC(J) .GE. CON(I,1)) GO TO 95
90 CONTINUE
   GO TO 102
95 DO 115 M=1,NA
   IF (CON(I,1) .LE. AMP(M,2)) GO TO 120
115 CONTINUE
   M = NA
120 VT = AMP(M,3)+(CON(I,1)-AMP(M,1))*(AMP(M,4)-AMP(M,3))/
   * (AMP(M,2)-AMP(M,1))
   IF (CON(I,1) .LT. AMP(M,1) .OR. CON(I,1) .GT. AMP(M,2)) VT = 0.
   IF (CON(I,1) .EQ. CON(I,3) .AND. CON(I,4) .EQ. 0.0) GO TO 105
   DO 125 N=1,NTDD
   IF (CON(I,1) .LE. TOD(N,2)) GO TO 130

```

```
125 CONTINUE
    N = NTDD
130 VR = TDD(N,3)+(CON(I,1)-TDD(N,1))*(TDD(N,4)-TDD(N,3))/
    * (TDD(N,2)-TDD(N,1))
    IF (CON(I,1) .LT. TDD(N,1) .OR. CON(I,1) .GT. TDD(N,2)) VR = 0.
105 VVS = CON(I,2)*(VT + (CON(I,1) - CON(I,3))*VR)*CONVRT
    VMS = CON(I,4)*VR*CONVRT
    DO 100 K=J,NX
        ZV(K) = ZV(K) + VVS
100 ZM(K) = ZM(K) + VMS + (XVEC(K) - CON(I,3))*VVS
102 CONTINUE
    RETURN
```

C

```
999 CALL ZZBOMB (6HVM1 ,NERROR)
    END
```



VMTR1

-----

SUBROUTINE VMTR1 (PP,Z,NPP,NZ,KZ)  
 DIMENSION PP(1), Z(KZ,1)

C  
 C GENERATE TRANSFORMATION MATRIX TO GIVE SHEARS AND BENDING MOMENTS IN  
 C TERMS OF FORCES AND MOMENTS.  
 C CALLS FORMA SUBROUTINE ZZBOMB.  
 C CODED BY C BODLEY. JULY 1965.  
 C LAST REVISION BY WA BENFIELD. MARCH 1976.

C  
 C SUBROUTINE ARGUMENTS  
 C PP = INPUT VECTOR OF PANEL POINT STATIONS. SIZE(NPP).  
 C Z = OUTPUT SHEAR,MOMENT TRANSFORMATION. SIZE(2\*NPP,2\*NPP).  
 C NPP = INPUT NUMBER OF PANEL POINTS.  
 C NZ = OUTPUT SIZE OF SHEAR,MOMENT TRANSFORMATION. (NZ=2\*NPP).  
 C KZ = INPUT ROW DIMENSION OF Z IN CALLING PROGRAM.

C  
 C NERROR EXPLANATION  
 C 1 = LESS THAN 2 PANEL POINTS.  
 C 2 = PANEL POINTS NOT IN INCREASING ORDER.

NERROR=1

IF (NPP .LT. 2) GO TO 999

NERROR=2

DO 5 I=2,NPP  
 IF (PP(I-1) .GE. PP(I)) GO TO 999  
 5 CONTINUE

NZ=2\*NPP  
 DO 10 I=1,NZ  
 DO 10 J=1,NZ  
 10 Z(I,J) = 0.  
 DO 25 I=1,NPP  
 K=I+NPP  
 DO 25 J=1,I  
 L= J + NPP  
 Z(I,J)=1.0  
 Z(K,L)=1.0  
 25 Z(K,J)=PP(I)-PP(J)  
 RETURN

C  
 999 CALL ZZBOMB (6HVMTR1 ,NERROR)  
 END

```

SUBROUTINE WRITAN (IA, NR, NC, ANAME, KR)
DIMENSION IA(KR,1)
COMMON /LLINE /NLINE, MAXLIN, MINI
DATA NIT, NOT/5, 6/

C
C WRITE MATRIX OF ALPHA-NUMERIC CHARACTERS (A6) ON PAPER.
C REQUIRES 132 COLUMN (MINIMUM) PRINTER.
C UP TO 20 DATA FIELDS PER LINE. PRINTS ONLY NON-BLANK FIELD ROWS.
C CALLS FORMA SUBROUTINE PAGEHD.
C CODED BY JOHN ADMIRE *NAS* OCT 1974.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C IA = MATRIX TO BE PRINTED. SIZE(NR,NC).
C NR = NUMBER OF ROWS IN MATRIX IA.
C NC = NUMBER OF COLS IN MATRIX IA.
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C KR = ROW DIMENSION OF IA IN CALLING PROGRAM.
C
2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(34,2H X 14,2H ) //
* 12X,20(1X,1H( 12,1H)1X)/)
2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(14,2H X 14,2H )
* 3X, 9HCONTINUED //12X,20(1X,1H( 12,1H),1X)/)
2030 FORMAT (1X,2I4,3X,20A6)
2040 FORMAT (15HEND OF WRITAN.)
2050 FORMAT (/1X131(1H-))

C
C CHECK IF NEW PAGE NEEDED

C
IF(MINI .NE. 4*MINI) GO TO 10
IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
NBC=NC/20
IF(NBC*20 .NE. NC) NBC=NBC+1
NNL=10+NR
IF(NBC .GT. 1) NNL= 9+NR*(NBC+1)
IF(NNL+NLINE .GT. MAXLIN) GO TO 10
WRITE(NOT,2050)
NLINE=NLINE+2
GO TO 20
10 CALL PAGEHD

C
C WRITE MATRIX
C
20 WRITE(NOT,2010) ANAME, NR, NC, (L, L=1, 20)
NLINE=NLINE+6
DO 90 I=1, NR
NZFRQ=0
JS=1
30 JE=JS+19
IF(JE .GT. NC) JE=NC
DO 40 J=JS, JE
40 IF(IA(I, J) .NE. 6H ) GO TO 50
GO TO 70
50 NLINE=NLINE+1
IF(NLINE .LE. MAXLIN) GO TO 60
CALL PAGEHD

```

```
WRITE (NOT,2020) ANAME, NR, NC, (L, L=1, 20)
NLINE=NLINE+6
60 WRITE (NOT,2030) I, JS, (IA(I, J), J=JS, JE)
NZERO=1
70 IF(JE .EQ. NC) GO TO 80
JS=JS+20
GO TO 30
80 IF(NC .LE. 20 .OR. NZERO.EQ. 0 .OR. I .EQ. NR) GO TO 90
NLINE=NLINE+1
WRITE (NOT,2030)
90 CONTINUE
WRITE (NOT,2040)
NLINE=NLINE+2
RETURN
END
```

```

SUBROUTINE WRITE (A,NR,NC,ANAME,KR)
DIMENSION A(KR,1)
COMMON /LLINE /NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C WRITE MATRIX OF REAL NUMBERS ON PAPER.
C REQUIRES 123 COLUMN (MINIMUM) PRINTER.
C UP TO 10 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.
C CALLS FORMA SUBROUTINE PAGEHD.
C CODED BY RL WOHLER. DECEMBER 1968.
C MODIFIED BY JOHN ADMIRE *NASA* SEPT 1973
C

```

```

C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX TO BE PRINTED. SIZE(NR,NC).
C NR = NUMBER OF ROWS IN MATRIX A.
C NC = NUMBER OF COLS IN MATRIX A.
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C KR = ROW DIMENSION OF A IN CALLING PROGRAM.
C

```

```

2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //
* 10X,10(7X,1H( I2,1H))//)
2020 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H )
* 3X, 9HCONTINUED //10X,10(7X,1H( I2,1H))//)
2030 FOPMAT (1X,2I5,2X,1P10E11.4)
2040 FORMAT (14HOEND OF WRITE.)
2050 FORMAT (/1X123(1H-))

```

```

C CHECK IF NEW PAGE NEEDED
C

```

```

IF(MINI .NE. 4HMINI) GO TO 10
IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
NBC=NC/10
IF(NBC*10 .NE. NC) NBC=NBC+1
NNL=10+NR
IF(NBC .GT. 1) NNL= 9+NR*(NBC+1)
IF(NNL+NLINE .GT. MAXLIN) GO TO 10
WRITE(NOT,2050)
NLINE=NLINE+2
GO TO 20

```

```

10 CALL PAGEHD

```

```

C WRITE MATRIX
C

```

```

20 WRITE(NOT,2010) ANAME,NR,NC,(L,L=1,10)
NLINE=NLINE+6
DO 90 I=1,NR
NZERO=0
JS=1
30 JE=JS+9
IF(JE .GT. NC) JE=NC
DO 40 J=JS,JE
40 IF(ABS(A(I,J)) .GT. 0.) GO TO 50
GO TO 70
50 NLINE=NLINE+1
IF(NLINE .LE. MAXLIN) GO TO 60

```

```
CALL PAGEHD
WRITE (NOT,2020) ANAME, NR, NC, (L, L=1, 10)
NLINE=NLINE+6
60 WRITE (NOT,2030) I, JS, (A(I, J), J=JS, JE)
NZERO=1
70 IF(JE .EQ. NC) GO TO 80
JS=JS+10
GO TO 30
80 IF(NC .LE. 10 .OR. NZERO.EQ. 0 .OR. I .EQ. NR) GO TO 90
NLINE=NLINE+1
WRITE (NOT,2030)
90 CONTINUE
WRITE (NOT,2040)
NLINE=NLINE+2
RETURN
END
```

```

SUBROUTINE WRITIM (IA,NR,NC,ANAME,KR)
DIMENSION IA(KR,1)
COMMON /LLINE /NLINE,MAXLIN,MINI
DATA NIT,NOT/5,6/

```

```

C
C WRITE MATRIX OF INTEGER NUMBERS ON PAPER.
C REQUIRES 116 COLUMN (MINIMUM) PRINTER.
C UP TO 20 DATA FIELDS PER LINE. PRINTS ONLY NON-ZERO FIELD ROWS.
C CALLS FORMA SUBROUTINE PAGEHD.
C CODED BY FL WOHLER. JULY 1968.
C MODIFIED BY JOHN ADMIRE *NASA* SEPT 1973

```

```

C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C IA = MATRIX TO BE PRINTED. SIZE(NR,NC).
C NR = NUMBER OF ROWS IN MATRIX IA.
C NC = NUMBER OF COLS IN MATRIX IA.
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C KR = ROW DIMENSION OF IA IN CALLING PROGRAM.

```

```

C
2010 FORMAT (//15H OUTPUT MATRIX A6,2X 1H(I4,2H X I4,2H ) //
* 16X,20(1X,1H( I2,1H))//)
2020 FORMAT (//15H END OF MATRIX A6,2X 1H(I4,2H X I4,2H )
* 3X, 9HCONTINUED //16X,20(1X,1H( I2,1H))//)
2030 FORMAT (1X,215,5X,2015)
2040 FORMAT (15HEND OF WRITIM.)
2050 FORMAT (/1X116(1H-))

```

```

C
C CHECK IF NEW PAGE NEEDED

```

```

IF(MINI .NE. 4*MINI) GO TO 10
IF(NLINE .LE. 5 .OR. NLINE .GE. MAXLIN) GO TO 10
NBC=NC/20
IF(NBC*20 .NE. NC) NBC=NBC+1
NNL=10+NR
IF(NBC .GT. 1) NNL= 9+NR*(NBC+1)
IF(NNL+NLINE .GT. MAXLIN) GO TO 10
WRITE(NOT,2050)
NLINE=NLINE+2
GO TO 20

```

```

10 CALL PAGEHD

```

```

C
C WRITE MATRIX

```

```

20 WRITE(NOT,2010) ANAME,NR,NC,(L,L=1,20)
NLINE=NLINE+6
DO 90 I=1,NR
NZERO=0
JS=1
30 JE=JS+19
IF(JE .GT. NC) JE=NC
DO 40 J=JS,JE
40 IF(IA(I,J) .NE. 0) GO TO 50
GO TO 70
50 NLINE=NLINE+1
IF(NLINE .LE. MAXLIN) GO TO 60

```

```
CALL PAGEHD
WRITE (NOT,2020) ANAME, NR, NC, (L, L=1, 20)
NLINE=NLINE+6
60 WRITE (NOT,2030) I, JS, (IA(I, J), J=JS, JE)
NZERO=1
70 IF (JF .EQ. NC) GO TO 80
JS=JS+20
GO TO 30
80 IF (NC .LE. 20 .OR. NZERO.EQ. 0 .OR. I .EQ. NR) GO TO 90
NLINE= NLINE+1
WRITE (NOT,2030)
90 CONTINUE
WRITE (NOT,2040)
NLINE=NLINE+2
RETURN
END
```

```

SUBROUTINE WTAPE (A,NRA,NCA,ANAME,KR,NTAPE)
DIMENSION A(KR,1)
COMMON /LSTART/IRUNNO,DATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
DATA BUF,EOT,DENSE/0.,3HEOT,5HDENSE/

```

```

C
C WRITE MATRIX A ON TAPE.
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-6000 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 CALLS FORMA SUBROUTINE ZZBOMB.
C CODED BY W A BENFIELD. MARCH 1966.
C REVISED BY RF HRUDA. NOVEMBER 1970.
C MODIFIED FOR CONTRACT NAS8-25922, MAY 1971.
C
C SUBROUTINE ARGUMENTS (ALL INPUT)
C A = MATRIX TO BE WRITTEN ON TAPE. SIZE(NRA,NCA).
C NRA = NUMBER OF ROWS OF MATRIX A.
C NCA = NUMBER OF COLS OF MATRIX A.
C ANAME = MATRIX IDENTIFICATION. (A6 FORMAT).
C KR = ROW DIMENSION OF A IN CALLING PROGRAM.
C NTAPE = NUMBER OF TAPE. (E.G. 10).
C
C NERROP EXPLANATION
C 1 = NON-POSITIVE ROW OR COLUMN SIZE.
C
C INTERNAL VARIABLES THAT ARE PUT ON TAPE (TRANSFERRED THRU COMMON).
C IRUNNO IS RUN NUMBER OF PROBLEM. (A6 FORMAT).
C DATE IS DATE. (A6 FORMAT). FOR EXAMPLE 15FE65.
C
C
C NERROR = 1
C
C IF (NRA .LT. 1 .OR. NCA .LT. 1) GO TO 999
C
C SEARCH TAPE FOR END OF WRITTEN DATA.
10 READ (NTAPE) TAPEID,LN,IEOTCK
IF (IEOTCK .EQ. 3HEOT) GO TO 20
PEAD (NTAPE)
GO TO 10
C
C END OF WRITTEN DATA HAS BEEN FOUND.
20 BACKSPACE NTAPE
WRITE (NTAPE) TAPEID,LN,BUF,IRUNNO,ANAME,NRA,NCA,DATE,DENSE,
* (BUF,I=1,10)
WRITE (NTAPE) ((A(I,J),I=1,NRA),J=1,NCA)
LN = LN + 1
WRITE (NTAPE) TAPEID,LN,EOT,(BUF,I=1,16)
ENDFILE NTAPE
REWIND NTAPE
NREC = 2 * (LN-1)
DO 30 IREC=1,NREC
30 READ (NTAPE)

```



WTAPE -- 2/ 2

RETURN

999 CALL ZZBOMB (6HWTAPE ,NERROR)  
END

```

SUBROUTINE XLORD (V,LV,LAS,NNZA)
DIMENSION V(1),LV(1),W(256),LW(256),IU(16),IL(16)
DATA NIT,NOT/5,6/
DATA LWDIM/256/

C
C ARRANGE ELEMENT LOCATIONS (LV) INTO INCREASING ORDER.
C REARRANGE ELEMENTS (V) ACCORDINGLY.
C DEVELOPED BY R A PHILIPPUS. OCTOBER 1968.
C LAST REVISION BY WA BENFIELD. MARCH 1976.
C
C SUBROUTINE ARGUMENTS
C V = INPUT VECTOR. A ELEMENTS. *DESTROYED*
C = OUTPUT VECTOR. A ELEMENTS. (ARRANGED)
C LV = INPUT VECTOR. ELEMENT LOCATIONS OF A. *DESTROYED*
C = OUTPUT VECTOR. ELEMENT LOCATIONS OF A. (ARRANGED)
C LAS = INPUT START LOCATION OF A IN V.
C NNZA = INPUT NUMBER OF NON-ZEROS IN A.
C
C ERROR EXPLANATION
C 1 = TWO LIKE LOCATION NUMBERS ENCOUNTERED.
C 2 = TWO LIKE LOCATION NUMBERS ENCOUNTERED.
C 3 = TWO LIKE LOCATION NUMBERS ENCOUNTERED.
C
3001 FORMAT (50HITWO LIKE LOCATION NUMBERS ENCOUNTERED IN XLORD AT111/)
3003 FORMAT (5(I12,E12.3))
C
IF (NNZA.LE.1) RETURN
LAE=LAS-1+NNZA
LAEM1=LAE-1
NSEG=1

C
C QUICK SEARCH FOR 1 OR 2 SEGMENTS.
C
DO 5 I=LAS,LAEM1
IF (LV(I).LT.LV(I+1)) GO TO 5
IF (LV(I).EQ.LV(I+1)) GO TO 990
NSEG=NSEG+1
IA=I
IF (NSEG.GT.2) GO TO 6
5 CONTINUE
IF (NSEG.EQ.1) RETURN
NNZS=IA-LAS+1

C
C CHOOSE BETWEEN MESH AND SINGLETON METHODS
FNNZA = NNZA
FNNZS = NNZS
X = FNNZS/FNNZA
DESCID = 6090./(FNNZA + 1420.)
IF (X .GT. DESCID) GO TO 6

C
C MESHING METHOD
C
LBS = LAS
LRE=LBS-1+NNZS
LCS=LRF+1

```

NERROR=1

```

LCE = LAE
IB=LBS
IC=LCS
IW=0
IZ=LBS-1
50
IF (LV(IB)-LV(IC)) 65,992,55
55 IW=IW+1
W(IW)=V(IC)
LW(IW)=LV(IC)
IC=IC+1
NN=1
IF (IW.EQ.LWDIM) GO TO 95
60 IF (IC.GT.LCE) GO TO 75
GO TO 50
65 IW=IW+1
W(IW)=V(IB)
LW(IW)=LV(IB)
IB=IB+1
NN=2
IF (IW.EQ.LWDIM) GO TO 95
70 IF (IB.GT.LBE) GO TO 85
GO TO 50
75 NELTM=LBE-IB+1
I=IC-1
DO 80 J=1,NELTM
V(I)=V(LBE)
LV(I)=LV(LBE)
LBE=LBE-1
80 I=I-1
85 IF (IW.EQ.0) RETURN
DO 90 I=1,IW
IZ=IZ+1
V(IZ)=W(I)
90 LV(IZ)=LW(I)
RETURN
95 NELTM=LBE-IB+1
I=IC-1
DO 100 J=1,NELTM
V(I)=V(LBE)
LV(I)=LV(LBE)
LBE=LBE-1
100 I=I-1
IB=IB+I-LBE
LBE=IC-1
DO 105 I=1,LWDIM
IZ=IZ+1
V(IZ)=W(I)
105 LV(IZ)=LW(I)
IW=0
GO TO (60,70),NN

```

NERROR=2

SINGLETON METHOD

6 M=1

```

I=LAS
J=LAS-1+NNZA
7 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.LAS) GO TO 7
    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 DO 215 I=LAS,LAEM1

    IF (LV(I).EQ.LV(I+1)) GO TO 990
215 CONTINUE
    RETURN

```

NERROR=3

C

C

```

ERROR STATEMENTS
990 WRITE (NOT,3001) LV(I)
    GO TO 995
992 WRITE (NOT,3001) LV(IB)
995 WRITE (NOT,3003) (LV(L),V(L),L=LAS,LAEM)
    CALL ZZBOMB (SHXLORD ,NERROR)
    END

```



ZEROLH

-----

```
SUBROUTINE ZEROLH (A,N,KR)
  DIMENSION A(KR,1)
```

```
C
C SET LOWER HALF OF SQUARE MATRIX A TO ZERO.
C CODED BY RF HRUDA. FEB 1965.
C
C SUBROUTINE ARGUMENTS
C A = INPUT, OUTPUT SUPPLIED AND RESULT MATRIX. SIZE(N,N).
C N = INPUT SIZE OF MATRIX A (SQUARE).
C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.
C
  DO 10 I=2,N
    IM1 = I-1
    DO 10 J=1,IM1
10 A(I,J) = 0.0
  RETURN
  END
```

ZEROUH  
-----

SUBROUTINE ZEROUH (A,N,KR)  
DIMENSION A(KR,1)

C  
C SET UPPER HALF OF SQUARE MATRIX A TO ZERO.  
C CODED BY RF HRUDA. FEB 1965.

C  
C SUBROUTINE ARGUMENTS  
C A = INPUT, OUTPUT SUPPLIED AND RESULT MATRIX. SIZE(N,N).  
C N = INPUT SIZE OF MATRIX A (SQUARE).  
C KR = INPUT ROW DIMENSION OF A IN CALLING PROGRAM.

C  
DO 10 J=2,N  
JMI = J-1  
DO 10 I=1,JMI  
10 A(I,J) = 0.0  
RETURN  
END



ZZBOMB

SUBROUTINE ZZBOMB(SUBNAM,NERROR)  
DATA NIT,NOT/5,6/

C  
C  
C ZZBOMB IS CALLED WHEN AN ERROR HAS BEEN ENCOUNTERED  
C IN A MAIN PROGRAM OR SUBROUTINE.  
C ZZBOMB PERFORMS THE FOLLOWING  
C (1) PRINTS THE PROGRAM NAME AND ERROR NUMBER WHEN THE  
C ERROR OCCURRED.  
C (2) A WALK BACK IS PRODUCED  
C (3) A DUMP IS PRODUCED  
C (4) PROGRAM IS TERMINATED  
C CODED BY JOHN ADMIRE \*NASA\* AUG 1972.  
C MODIFIED BY JOHN ADMIRE \*NASA\* DEC 1975  
C  
C ARGUMENTS  
C SUBNAM - INPUT SUBROUTINE NAME WHERE ERROR OCCURRED.  
C NERROR - INPUT ERROR NUMBER FROM SUBROUTINE WHERE ERROR OCCURRED.  
C  
C 3001 FORMAT(20H1STOP IN SUBROUTINE A6, 13H AT NERROR = I3)  
C  
C WRITE(NOT,3001) SUBNAM,NERROR  
C  
C CALL STPACE  
C CALL DUMP  
C STOP  
C END