เอกสารอ้างอิง

1. ทักษิณ เทพยาตรี, โปรแกรมไมโครคอมพิวเตอร์สำหรับวิเคราะห์โครงข้อหมุน 2 มิติ, เลขที่ คส 03/2526 ภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย.
2. ทักษิณ เทพยาตรี, โปรแกรมไมโครคอมพิวเตอร์สำหรับวิเคราะห์โครงข้อแข็ง 2 มิติ, เลขที่ คส 04/2526 ภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย.
3. อำนวย พานิชกุล, สมนึก กุลประภา, วินิต ช่อวิเชียร, การวิเคราะห์โครงสร้าง, 2522.
4. Meek, J.L., Matrix Structural Analysis, Mc. Graw-Hill, 1971.
5. Rubinstein, M.F., Matrix Compute Analysis of Structure, Prentice-Hall Inc., Englewood, N.J., 1966.
6. Norris, C.H., Wilbur, J.B., Elementary Structure Analysis, Mc.Graw-Hill, 1960.
7. Wang, Chu-Kia, Statically Indeterminate Structure, Mc. Graw-Hill, 1953.
8. Irons, B.M., "A Frontal Solution Programme for Finite Element Analysis" Int. J. num. Meth. Engng, Vol. 2, No. 1, January 1970, pp. 5-32.
9. M.F. Light and A.R. Luxmore. "Application of the Frontal Solution to Two and Three-Dimensional Elasto-Plastic Crack Problem", Int. J. Num. Meth. Engng, Vol. 11, 393-395, 1977.
10. P. Hood, "Frontal Solution Programme for Unsymmetric Matrices" Int. J. Num. Meth. Engng, Vol. 10, 379-399, 1976.
11. R. Natarajan, "Frontal Solution Programme for Transmission Tower Analysis" Comp. Struct. Vol. 5, 59-64, 1975.
12. A. Alizadeh and G.T. Will, "A Substructured Frontal Solver and Its Application to Localized Material nontinearity" Comp. Struct., Vol. 10, 225-231, 1979.

13. Rajatabhothi, R., "Multi-Level Substructuring of Large Structure Systems for The Eigenproblem"Ph.D. Dissertation Present to The Faculty of Graduate School of The University of Texas at Austin 1982.

14. Felipa, C.A., "Solution of Linear Equations with Skyline Stored Symmetric Matrix" Int. J. Num. Meth. Engng, Vol. 5, 13-29, 1975.

15. Klauss-Jurgen Bathe, Finite Element Procedures in Engineering Analysis, Prentice-Hall, Inc., Englewood cliffs, N.J. 1982.

16. Cook, R.D., Concepts and Applications of Finite Element Analysis, 2nd Edition, John Willeys son 1974.

17. G. Beer and W. HAAS, "A Partitioned Frontal Solver For Finite Element Analysis" Int. J. Num. Meth. Engng, Vol. 18, 1623-1654, 1982.

18. Erik Thompson and Yoh Shimazaki, "A Frontal Procedure Using Skyline Storage" Int. J. Num. Meth. Engng, Vol.15, 889-910, 1980.

19. Johnson, C.P., "A Frontal-Based Solver with Multi-Level Substructuring" Proceedings, North Carolina State University, May 23-25, 1977, pp. 381-384.

20. Gan Hui Tiong, "A Frontal Substructuring Scheme for Analysis of Unbranched Prismatic Structures" Master's Thesis, Asian In Institute of Technology, Bangkok, Thailand, 1980.

21. เริงเดช รัชตโพธิ์, "โปรแกรมไมโครคอมพิวเตอร์สำหรับระบบไฟไนท์ เอเลเมนท์ โดยใช้วิธีโครงสร้างย่อย", รายงานวิจัย, 2526 (กำลังอยู่ในระหว่างดำเนินงาน).

ภาคผนวก ก.

สมมติฐานในการวิจัย

สมมติฐานที่กำหนดขึ้นเพื่อใช้ในการวิเคราะห์มีดังต่อไปนี้

1. ชิ้นส่วนย่อยที่ประกอบเป็นโครงสร้างต้องมีลักษณะเป็นท่อนตรงมีพื้นที่หน้าตัดคงที่และมีโมเมนต์อินเนอเชียเท่ากันตลอดความยาวของชิ้นส่วน

2. วัสดุที่ใช้ประกอบชิ้นเป็นชิ้นส่วนย่อยแต่ละชิ้นส่วน ต้องเป็นวัสดุชนิดเดียวกันและมีคุณสมบัติเหมือนกันตลอดความยาวของชิ้นส่วน

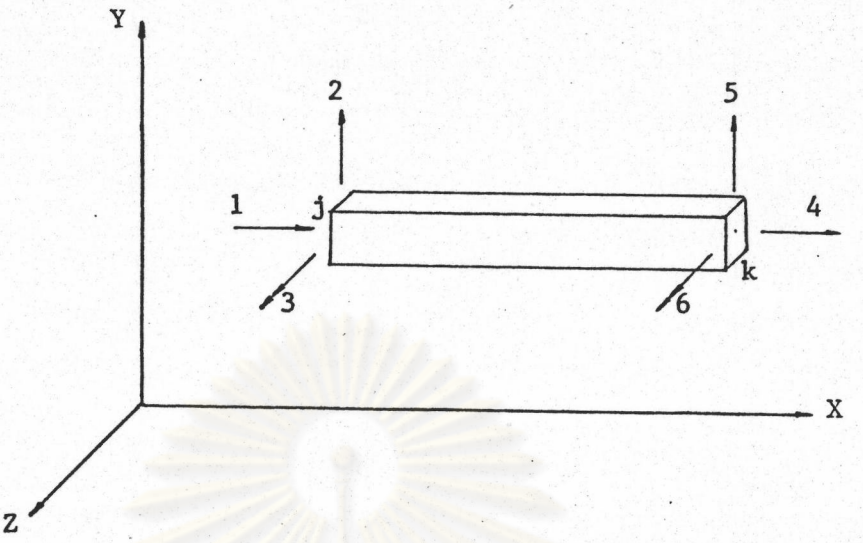
3. พฤติกรรมของชิ้นส่วนย่อยภายใต้การรับน้ำหนักต้องเป็นไปตาม Hooke's Law

4. ที่จุดต่อของชิ้นส่วนต่าง ๆ ของโครงข้อหมุน จะเป็นแบบหมุดยึดหมุน (Pin) ซึ่งหมุนได้ ไม่สามารถรับโมเมนต์

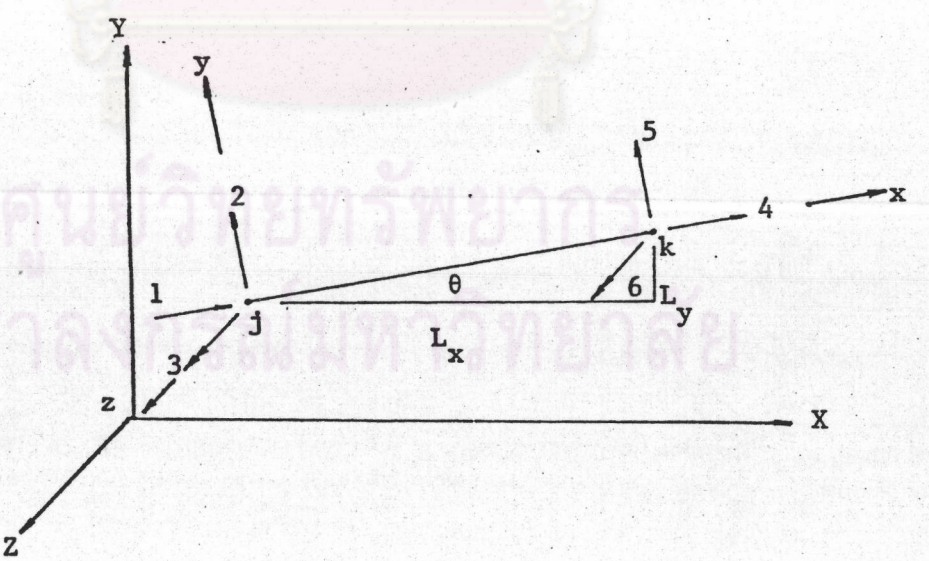
5. ที่จุดต่อของชิ้นส่วนต่าง ๆ ของโครงข้อแข็ง จะเป็นแบบข้อแข็ง (Rigid) สามารถรับโมเมนต์ได้

6. แรงกระทำหรือน้ำหนักบรรทุกที่กระทำบนโครงข้อหมุนจะกระทำที่จุดต่อของชิ้นส่วนย่อยเท่านั้น

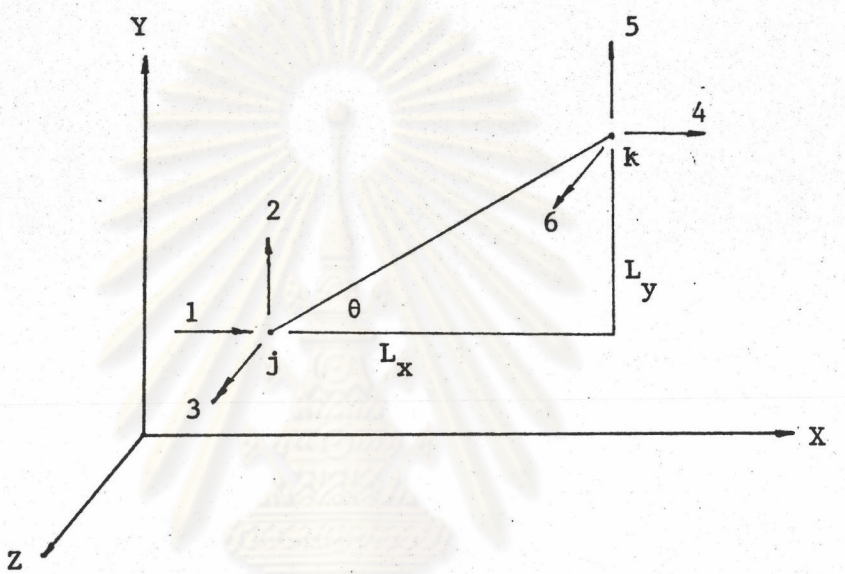
ชิ้นส่วนคาน ทิศทางการเคลื่อนที่ของดีกรีของความอิสระในพิกัดของชิ้นส่วนย่อย และของโครงสร้าง ดังแสดงในรูปที่ ก-1, ก-2, ก-3 ตามลำดับ



รูปที่ ก-1 ยืนส่วนคาน (Beam Element)



รูปที่ ก-2 ทิศทางการเคลื่อนที่ในพิกัดยืนส่วนย่อย



รูปที่ ก-3 ทิศทางการเคลื่อนที่ในพิกัดของโครงสร้าง

ภาคผนวก ข

รายละเอียดโปรแกรมคอมพิวเตอร์

```
1000  REM              ******************
1010  REM              ***   REMARK   ***
1020  REM              ******************
1030  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1040  REM  :  CONTROL IMFORMATION IN THIS PROGRAM            :
1050  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1060  REM
1070  REM     NM      : NUMBER OF MEMBERS
1080  REM     NJ      : NUMBER OF NODES
1090  REM     ML      : NUMBER OF LOADED MEMBERS
1100  REM     JL      : NUMBER OF LOADED NODES
1110  REM     HJ      : NUMBER OF NODES WITH HINGES
1120  REM     BCJ     : NUMBER OF BOUNDARY CONDITION NODES
1130  REM
1140  REM
1150  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1160  REM  :   SUBPROGRAM INPUT DATA, PRINT DATA & PLOT      :
1170  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1180  REM
1190  REM     ND      : NUMBER OF DOF. IN EACH NODE
1200  REM     NODE    : NUMBER OF NODES IN EACH MEMBER
1210  REM     X(I)    : ARRAY STORES X-COORDINATES OF EACH NODE
1220  REM     Y(I)    : ARRAY STORES Y-COORDINATES OF EACH NODE
1230  REM     J(I)    : ARRAY STORES NODES NUMBER OF LEFT END OF
1240  REM               EACH MEMBER
1250  REM     K(I)    : ARRAY STORES NODES NUMBER OF RIGHT END
1260  REM               OF EACH MEMBER
1270  REM     AR(I)   : ARRAY STORES CROSS-SECTIONAL AREA
1280  REM     IZ(I)   : ARRAY STORES MOMENT OF INERTIA
1290  REM     E(I)    : ARRAY STORES YOUNG'S MODULUS
1300  REM     H(I)    : ARRAY STORES NODES NUMBER WITH HINGES
1310  REM     K1(I)   : ARRAY STORES MEMBERS NUMBER OF LOADED MEMBERS
1320  REM     K2(I)   : ARRAY STORES NODES NUMBER OF LOADED NODES
1330  REM     W(I)    : ARRAY STORES UNIFORMLY DISTRIBUTED LOAD
1340  REM     F(I)    : ARRAY STORES NUMBER OF AXIAL LOADS IN EACH
1350  REM               LOADED MEMBER
1360  REM     P(I)    : ARRAY STORES NUMBER OF VERTICAL & HORIZONTAL
1370  REM               POINT LOADS
1380  REM     V1(I,J) : ARRAY STORES VALUE OF VERTICAL POINT LOADS
1390  REM     VA(I,J) : ARRAY STORES DISTANCE OF VERTICAL POINT
1400  REM               LOADS FROM LEFT END
1410  REM     H1(I,J) : ARRAY STORES VALUE OF HORIZONTAL POINT
1420  REM               LOADS
1430  REM     HA(I,J) : ARRAY STORES DISTANCE OF HORIZONTAL POINT
1440  REM               LOADS FROM LEFT END
1450  REM     F1(I,J) : ARRAY STORES VALUE OF AXIAL LOADS
1460  REM     FA(I,J) : ARRAY STORES DISTANCE OF AXIAL LOADS
1470  REM               FROM LEFT END
1480  REM     PJ(I,J) : ARRAY STORES VALUE OF NODAL LOADS
1490  REM     BJ(I)   : ARRAY STORES NUMBER OF BOUNDARY CONDITION
1500  REM               NODES & BOUNDARY CONDITION AT NODES
```

```
1510  REM
1520  REM
1530  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1540  REM  :          -: SUBPROGRAM STIFFNESS :-             :
1550  REM  :  THIS SUBPROGRAM PERSORMS LOCAL & GLOBAL        :
1560  REM  :  STIFFNESS MATRIX AND LOAD VECTOR OF EACH       :
1570  REM  :  MEMBER,  MODIFIES STIFFNESS MATRIX & LOAD      :
1580  REM  :  VECTOR FOR BOUNDARY CONDITION.    SAVE         :
1590  REM  :  LOCAL STIFFNESS & LOAD VECTOR IN FILE           :
1600  REM  :  'LOSTIFF',MODIFIED STIFFNESS & LOAD VECTOR      :
1610  REM  :  IN FILE 'GLOSTIFF'                              :
1620  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1630  REM
1640  REM    PCASE   : PRINT DATA CODE
1650  REM    NK      : NUMBER OF ELEMENTS ON UPPER TRIANGULAR
1660  REM              OF STIFFNESS MATRIX
1670  REM    NW      : NK PLUS NUMBER OF LOAD VECTOR IN ONE MEMBER
1680  REM    MD      : NUMBER OF DOF IN EACH MEMBER
1690  REM    ID(I)   : ARRAY STORES NODES NUMBER IN EACH MEMBER
1700  REM    IH(I)   : ARRAY STORES CONDITIONS OF NODES (HINGE
1710  REM              OR RIGID)
1720  REM    S(I)    : ARRAY STORES ELEMENTS OF STIFFNESS MATRIX
1730  REM    ST(I)   : ARRAY STORES LOADED VECTOR
1740  REM
1750  REM
1760  REM
1770  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1780  REM  :          -: SUBPROGRAM SETFRONT :-              :
1790  REM  :  THIS SUBPROGRAM PERFORMS MOC ARRY (FRONT)      :
1800  REM  :  TO STORE NODES WHICH WILL BE ELIMINATED        :
1810  REM  :  FROM CURRENT FRONT AND NODES WHICH WILL        :
1820  REM  :  REMAIN IN NEXT FRONT, COMPUTES OTHERS          :
1830  REM  :  STORAGE REQUIREMENT. SAVE MOC ARRAY AND        :
1840  REM  :  OTHERS DATA IN FILE 'DFRONT'                   :
1850  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1860  REM
1870  REM     IX     : STORAGE REQUIREMENTS FOR SETFRONT
1880  REM     IY     : STORAGE REQUIREMENTS FOR COMFRONT
1890  REM     NC     : STORAGE REQUIREMENTS FOR SEMRED
1900  REM     M      : NODES WHICH WILL BE ELIMINATED FROM FRONT
1910  REM     N      : NUMBER OF NODE IN FRONT
1920  REM     MQ     : DOF WHICH WILL BE ELIMINATED
1930  REM     NQ     : DOF IN EACH FRONT
1940  REM     NI     : MAX. NUMBER OF ELEMENTS STIFFNESS
1950  REM              IN MAX. FRONT SIZE
1960  REM     MN     : MAX. FRONT SIZE
1970  REM     MF     : MAX. DOF IN MAX. FRONT SIZE
1980  REM     IB(I)  : ARRAY STORES NUMBER OF MEMBERS THAT
1990  REM              CONNECTED AT NODE
2000  REM     IB(I)  : ARRAY STORES NNODES IN EACH FRONT
```

```
2010 REM
2020 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2030 REM  :           -: SUBPROGRAM COMFRONT :-              :
2040 REM  :   THIS SUBPROGRAM COMPARES CURRENT FRONT WITH    :
2050 REM  :   NEXT FRONT, REORDER LOCATION OF NODES IN       :
2060 REM  :   FRONT SO THAT NODES WHICH WILL REMAIN IN       :
2070 REM  :   NEXT FRONT ARE IN THE SAME LOCATION AS IN      :
2080 REM  :   CURRENT FRONT                                  :
2090 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2100 REM
2110 REM     MQ      : NUMBER OF DOF WHICH WILL BE ELIMINATED
2120 REM               FROM CURRENT FRONT
2130 REM     NQ      : NUMBER OF DOF IN CURRENT FRONT
2140 REM     M1      : NUMBER OF NODES WHICH WILL BE ELIMINATED
2150 REM               FROM CURRENT FRONT
2160 REM     N1      : NUMBER OF NODES IN CURRENT FRONT
2170 REM     MR      : NUMBER OF DOF WHICH WILL BE ELIMINATED
2180 REM               FROM NEXT FRONT
2190 REM     NR      : NUMBER OF DOF IN NEXT FRONT
2200 REM     M2      : NUMBER OF NODES WHICH WILL BE ELIMINATED
2210 REM               FROM NEXT FRONT
2220 REM     N2      : NUMBER OF NODES IN NEXT FRONT
2230 REM     KS      : NUMBER OF DOF IN EACH MEMBER
2240 REM     KQ      : MQ PLUS ONE
2250 REM     MK      : RECORD LENGTH FOR FILE 'MQ'
2260 REM     MP      : RECORD LENGTH FOR FILE 'FRONTIQ'
2270 REM     AMOC(I) : ARRAY STORES NODES IN CURRENT FRONT
2280 REM     BMOC(I) : ARRAY STORES NODES IN NEXT FRONT
2290 REM     LE(I)   : ARRAY STORES LOCATION OF DOF IN CURRENT
2300 REM               FRONT
2310 REM     LF(I)   : ARRAY STORES LOCATION OF DOF IN CURRENT
2320 REM               FRONT THAT APPEAR IN NEXT FRONT
2330 REM     IQ(I)   : ARRAY STORES NOBES NUMBER IN BOTH END
2340 REM               OF EACH MEMBER
2350 REM
2360 REM
2370 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2380 REM  :              - SUBPROGRAM SEMRED :-              :
2390 REM  :   THIS DUBPROGRAM ASSEMBLES STIFFNESS MATRIX     :
2400 REM  :   AND LOAD VECTOR OF EACH MEMBER AND REDUCES     :
2410 REM  :   DOF OF NODES WHICH WILL BE ELIMINATED FROM     :
2420 REM  :   FRONT                                          :
2430 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2440 REM
2450 REM     IQ      : NUMBER OF RECORDS IN FILE 'MQ'
2460 REM     A(I)    : ARRAY STORES ELEMENTS OF REDUCED STIFFNESS
2470 REM     S(I)    : ARRAY STORES ELEMENTS OF STIFFNESS
2480 REM               AND LOAD VECTOR
2490 REM     ND(I)   : ARRAY STORES LOCATION OF ELEMENTS AT DIAGONAL
2500 REM               OF STIFFNESS MATRIX
```

```
2510 REM
2520 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2530 REM  :           -: SUBPROGRAM BACKSUB :-               :
2540 REM  :   THIS SUBPROGRAM PERFORMS BACKSUBSTITUTION      :
2550 REM  :   TO FIND DISPLACEMENT IN GLOBAL COORDINATES     :
2560 REM  :   AND MEMBER FORCES IN LOCAL COORDINATES         :
2570 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2580 REM
2590 REM     A(I)    : ARRAY STORES ELEMENTS OF REDUCED STIFFNESS
2600 REM     D(I)    : ARRAY STORES DISPLACEMENT AT NODES
2610 REM     Q(I)    : ARRAY STORES FRONT DISPLACEMENT
2620 REM     DF(I)   : ARRAY STORES REDUCED LOAD VECTOR
2630 REM
2640 REM
2650 REM
2660 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2670 REM  :             -: SUBPROGRAM ERROR :-               :
2680 REM  :   THIS SUBPROGRAM PERFORMS SOLUTION ERROR OF     :
2690 REM  :   STATIC EQUATION :-                             :
2700 REM  :                KU = F                            :
2710 REM  :   USING EUCLIDEAN NORM AND DEVELOP PROGRAM       :
2720 REM  :   FOR FRONTAL METHOD                             :
2730 REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2740 REM
2750 REM     SUM     : SUMMATION OF K*U or (STIFFNESS)*(DISPLACEMENT)
2760 REM     B(I)    : ARRAY STORES LOAD VECTOR
2770 REM     S(I)    : ARRAY STORES GLOBAL STIFFNESS
2780 REM     LE(I)   : ARRAY STORES LOCATION OF NODES IN CURRENT
2790 REM               FRONT
2800 REM     LF(I)   : ARRAY STORES LOCATION OF NODES IN CURRENT
2810 REM               FRONT THAT APPEAR IN NEXT FRONT
```

```
1000  REM       ****************************************
1010  REM       ***                                  ***
1020  REM       ***                                  ***
1030  REM       ***   A FRONTAL PROGRAM FOR ANALYSIS ***
1040  REM       ***                                  ***
1050  REM       ***              OF                  ***
1060  REM       ***                                  ***
1070  REM       ***      PLANE FRAME STRUCTURES      ***
1080  REM       ***                                  ***
1090  REM       ***                                  ***
1100  REM       ****************************************
1110  REM
1120  REM    ::::::::::::::::::::::::::::::::::::::::::::::::::
1130  REM    :   THIS PROGRAM RUNS ON MICROCOMPUTER (Apple II :
1140  REM    :   Apple IIe) FOR ANALYSIS OF PLANE FRAME       :
1150  REM    :   STRUCTURES; LINEAR ELASTIC MATERIAL; BEAM &  :
1160  REM    :   BAR ELEMENTS, AXIAL LOAD, POINT LOAD AND     :
1170  REM    :   UNIFORM DISTRIBUTION LOAD ACT ON THE ELEMENT,:
1180  REM    :   FORCES AND MOMENT ACT AT NODE.               :
1190  REM    :                                                :
1200  REM    :   SUBPROGRAMS: INPUT DATA, PRINT DATA, PLOT,   :
1210  REM    :                STIFFNESS, SETFRONT, COMFRONT,  :
1220  REM    :                SEMRED, BACKSUB, ERROR          :
1230  REM    :                                                :
1240  REM    :   DATA FILES : LOSTIFF, GLOSTIFF, IQ, DFRONT,  :
1250  REM    :                FRONTIQ, MQ, DISPL, FORCE,      :
1260  REM    :                DIMEN                           :
1270  REM    ::::::::::::::::::::::::::::::::::::::::::::::::::
1280  REM
1290  HOME : REM    **  OPTION  **
1300  FLASH : PRINT  TAB( 4);"                                  "
1310  PRINT  TAB( 8);"2 - D FRAME ANALYSIS MENU    "
1320  PRINT  TAB( 4);"                                  ": NORMAL
1330  PRINT : PRINT
1340  PRINT  TAB( 11);"<1> - INPUT DATA": PRINT
1350  PRINT  TAB( 11);"<2> - PLOT": PRINT
1360  PRINT  TAB( 11);"<3> - PRINT DATA": PRINT
1370  PRINT  TAB( 11);"<4> - ANALYZE": PRINT
1380  PRINT  TAB( 11);"<5> - ERROR": PRINT
1390  PRINT  TAB( 11);"<6> - EXIT": PRINT
1400  PRINT : INPUT "     ENTER SELECTION NO.<1-6> : ";A1
1410  IF A1 < 1 GOTO 1400
1420  IF A1 > 6 GOTO 1400
1430  HOME :D$ =  CHR$ (4): ON A1 GOTO 1440,1450,1460,1470,1480,1490
1440  PRINT D$;"RUN INPUT DATA"
1450  PRINT D$;"RUN PLOT"
1460  PRINT D$;"RUN PRINT DATA"
1470  PRINT D$;"RUN STIFFNESS"
1480  PRINT D$;"RUN ERROR"
1490  END
```

```
1000  REM         ********************
1010  REM         *** INPUT DATA ***
1020  REM         ********************
1030  HOME : PRINT  TAB( 16);"***********"
1040  PRINT  TAB( 16);"INPUT DATA": PRINT  TAB( 16);"***********"
1050  PRINT : INPUT "  IS YOUNG'S MODULUS CONSTANT ? (Y/N) ";F$
1060  PRINT : INPUT "  NO. OF MEMBERS.....= ";NM
1070  PRINT : INPUT "  NO. OF NODES.....= ";NJ
1080  PRINT : INPUT "  NO. OF LOADED MEMBERS...= ";ML
1090  PRINT : INPUT "  NO. OF LOADED NODES.....= ";JL
1100  PRINT : INPUT "  NO. OF BOUNDARY NODES...= ";BCJ
1110  PRINT : INPUT "  NO. OF NODES WITH HINGES= ";HJ
1120  IF F$ = "N" GOTO 1140
1130  PRINT : INPUT "  YOUNG'S MODULUS.....= ";E
1140  PRINT : INPUT "  DO YOU ACCEPT THESE DATA ? (Y/N)  ";B$
1150  IF B$ = "N" GOTO 1060
1160  IF B$ <  > "Y" GOTO 1140
1170  LET ND = 3:NODE = 2
1180  LET A$ = "----------------------------------------"
1190  LET BB = 2 * ML
1200  DIM X(NJ),Y(NJ),J(NM),K(NM),AR(NM),IZ(NM),F(ML),H(HJ)
1210  DIM W(BB),P(BB),V1(5,ML),VA(5,ML),H1(5,ML),HA(5,ML),E(NM)
1220  DIM K1(ML),K2(JL),PJ(JL,3),BJ(7 * BCJ),F1(5,ML),FA(5,ML)
1230  IF F$ = "N" GOTO 1250
1240  FOR I = 1 TO NM:E(I) = E: NEXT I
1250  REM    ** INPUT NODAL POINT COORDINATES **
1260  PRINT : INPUT A$: PRINT "     INPUT NODAL POINT COORDINATES"
1270  PRINT A$: PRINT : FOR I = 1 TO NJ
1280  PRINT "  NODE  NO. ";I: PRINT "  ============="
1290  INPUT "  X-COORDINATE.....= ";X(I)
1300  INPUT "  Y-COORDINATE.....= ";Y(I): PRINT
1310  INPUT "  DO YOU ACCEPT THESE DATA ? (Y/N)  ";B$
1320  PRINT : IF B$ = "N" GOTO 1290
1330  IF B$ <  > "Y" GOTO 1310
1340  PRINT : NEXT I
1350  REM    ** INPUT NODE NO. & PROPERTIES OF EACH MEMBERS **
1360  PRINT : PRINT A$: PRINT "   INPUT NODE NO. & PROP. OF MEMBERS"
1370  PRINT A$: PRINT : FOR I = 1 TO NM
1380  PRINT "  MEMBER NO. ";I: PRINT "  =============="
1390  INPUT "  NODE J.....= ";J(I)
1400  INPUT "  NODE K.....= ";K(I)
1410  INPUT "  CROSS SECTIONAL AREA.= ";AR(I)
1420  INPUT "  INERTIA.....= ";IZ(I)
1430  IF F$ = "Y" GOTO 1450
1440  INPUT "  YOUNG'S MODULUS.....= ";E(I)
1450  PRINT : INPUT "  DO YOU ACCEPT THESE DATA ? (Y/N)  ";B$
1460  PRINT : IF B$ = "N" GOTO 1390
1470  IF B$ <  > "Y" GOTO 1450
1480  NEXT I: PRINT
1490  REM   ** INPUT MEMBER LOADS IN LOCAL COORDINATES **
1500  IF ML = 0 GOTO 1750
```

```
1510  PRINT A$: PRINT "INPUT MEMBER LOADS IN GLOBAL COORDINATES"
1520  PRINT A$: PRINT : FOR I = 1 TO ML:I1 = 2 * (I - 1)
1530  INPUT "  MEMBER NO. ";K1(I): PRINT "  =============="
1540  INPUT "  VERT. UNIF. LOAD.....= ";W(I1 + 1)
1550  INPUT "  HORZ. UNIF. LOAD.....= ";W(I1 + 2)
1560  INPUT "  NO. OF AXIAL LOAD....= ";F(I)
1570  INPUT "  NO. OF VERT. PT. LOAD= ";P(I1 + 2)
1580  INPUT "  NO. OF HORZ. PT. LOAD= ";P(I1 + 1)
1590  PRINT : IF F(I) = 0 GOTO 1630
1600  FOR J = 1 TO F(I): PRINT "  AXIAL LOAD NO. ";J
1610  INPUT "    VALUE OF LOAD......= ";F1(J,I)
1620  INPUT "    DIST. FROM LEFT....= ";FA(J,I): NEXT J
1630  IF P(I1 + 2) = 0 GOTO 1670
1640  FOR J = 1 TO P(I1 + 2): PRINT "  VERT. LOAD NO. ";J
1650  INPUT "    VALUE OF LOAD......= ";V1(J,I)
1660  INPUT "    DIST FROM LEFT.....= ";VA(J,I): NEXT J
1670  IF P(I1 + 1) = 0 GOTO 1710
1680  FOR J = 1 TO P(I1 + 1): PRINT "  HORZ. LOAD NO. ";J
1690  INPUT "    VALUE OF LOAD......= ";H1(J,I)
1700  INPUT "    DIST. FORM LEFT....= ";HA(J,I): NEXT J
1710  PRINT : INPUT "  DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1720  IF B$ = "N" GOTO 1530
1730  IF B$ <  > "Y" GOTO 1710
1740  NEXT I: PRINT
1750  IF JL = 0 GOTO 1870
1760  REM   ** INPUT NODAL LOADS IN GLOBAL COORDINATES **
1770  PRINT A$: PRINT "INPUT NODAL LOADS IN GLOBAL COORDINATES"
1780  PRINT A$: PRINT : FOR I = 1 TO JL
1790  INPUT "  NODE NO. ";K2(I): PRINT "  ============="
1800  INPUT "  FORCE IN X-DIRECTION.= ";PJ(I,1)
1810  INPUT "  FORCE IN Y-DIRECTION.= ";PJ(I,2)
1820  INPUT "  MOMENT..............= ";PJ(I,3): PRINT
1830  INPUT "  DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
1840  PRINT : IF B$ = "N" GOTO 1790
1850  IF B$ <  > "Y" GOTO 1830
1860  NEXT I: PRINT
1870  IF BCJ = 0 GOTO 2050
1880  REM    ** INPUT BOUNDARY CONDITIONS AT NODES **
1890  PRINT A$: PRINT " INPUT BC. AT NODES (FREE=0,RESTR.=1)"
1900  PRINT A$: PRINT : FOR I = 1 TO BCJ:I1 = 7 * (I - 1) + 1
1910  INPUT "  NODE NO. ";BJ(I1): PRINT "  ============="
1920  INPUT "  X-CONDITION.........= ";BJ(I1 + 1)
1930  INPUT "  Y-CONDITION.........= ";BJ(I1 + 2)
1940  INPUT "  ROTATE-CONDITION.....= ";BJ(I1 + 3)
1950  PRINT : IF BJ(I1 + 1) = 0 GOTO 1970
1960  INPUT "  X-DISPL.............= ";BJ(I1 + 4)
1970  IF BJ(I1 + 2) = 0 GOTO 1990
1980  INPUT "  Y-DISPL.............= ";BJ(I1 + 5)
1990  IF BJ(I1 + 3) = 0 GOTO 2010
2000  INPUT "  ROTATION............= ";BJ(I1 + 6)
```

```
2010  PRINT : INPUT "  DO YOU ACCEPT THESE DATA ? (Y/N) ";B$
2020  PRINT : IF B$ = "N" GOTO 1910
2030  IF B$ <  > "Y" GOTO 2010
2040  NEXT I
2050  IF HJ = 0 GOTO 2150
2060  IF HJ = NJ GOTO 2150
2070  REM   **  INPUT NODE WITH HINGES  **
2080  PRINT A$: PRINT " INPUT NODE WITH HINGE(HINGE=0,RIGID=1)"
2090  PRINT A$: PRINT
2100  FOR I = 1 TO HJ
2110  INPUT "  NODE NO.............= ";H(I): NEXT I: PRINT
2120  INPUT "  DO YOU ACCEPT THIS DATA ? (Y/N) ";B$
2130  PRINT : IF B$ = "N" GOTO 2100
2140  IF B$ <  > "Y" GOTO 2120
2150  HOME : PRINT "   INPUT DATA DISK NO.1 INTO DRIVE": VTAB 24
2160  INPUT "     PRESS 'RETURN KEY TO CONTINUE";B$: HOME
2170  REM    ** SAVE DATA INTO DISKETTE **
2180  PRINT "    NAME OF DATA FILE TO BE SAVED ": PRINT
2190  INPUT "          FILE NAME..... ";C$:D$ =  CHR$ (4)
2200  PRINT D$;"OPEN";C$: PRINT D$;"DELETE";C$
2210  PRINT D$;"OPEN";C$: PRINT D$;"WRITE";C$
2220  PRINT NM: PRINT NJ: PRINT ML: PRINT JL: PRINT BCJ
2230  PRINT E: PRINT HJ: PRINT NODE: PRINT ND
2240  FOR I = 1 TO NJ: PRINT X(I): PRINT Y(I): NEXT I
2250  FOR I = 1 TO NM: PRINT J(I): PRINT K(I): PRINT AR(I)
2260  PRINT IZ(I): PRINT E(I): NEXT I: IF ML = 0 GOTO 2370
2270  FOR I = 1 TO ML:I1 = 2 * (I - 1)
2280  PRINT K1(I): PRINT W(I1 + 1): PRINT W(I1 + 2): PRINT F(I)
2290  PRINT P(I1 + 1): PRINT P(I1 + 2): IF F(I) = 0 GOTO 2310
2300  FOR J = 1 TO F(I): PRINT F1(J,I): PRINT FA(J,I): NEXT J
2310  IF P(I1 + 2) = 0 GOTO 2340
2320  FOR J = 1 TO P(I1 + 2): PRINT V1(J,I): PRINT VA(J,I)
2330  NEXT J
2340  IF P(I1 + 1) = 0 GOTO 2360
2350  FOR J = 1 TO P(I1 + 1): PRINT H1(J,I): PRINT HA(J,I): NEXT J
2360  NEXT I
2370  IF JL = 0 GOTO 2400
2380  FOR I = 1 TO JL: PRINT K2(I)
2390  FOR L = 1 TO 3: PRINT PJ(I,L): NEXT L: NEXT I
2400  IF BCJ = 0 GOTO 2430
2410  FOR I = 1 TO BCJ: FOR L = 1 TO 7
2420  LET L1 = 7 * (I - 1) + L: PRINT BJ(L1): NEXT L: NEXT I
2430  IF HJ = 0 GOTO 2460
2440  IF HJ = NJ GOTO 2460
2450  FOR I = 1 TO HJ: PRINT H(I): NEXT I
2460  PRINT D$;"CLOSE";C$: HOME
2470  PRINT "     INPUT PROGRAM DISK INTO DRIVE ": VTAB 24
2480  INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
2490  PRINT D$;"RUN OPTION"
2500  END
```

```
1000  REM    ******************
1010  REM    **     PLOT     **
1020  REM    ******************
1030  HOME : PRINT "   INPUT DATA DISK NO.1 INTO DRIVE": VTAB 24
1040  INPUT "   PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1050  PRINT "     NAME OF DATA FILE TO BE PLOTED"
1060  INPUT "          FILE NAME..... ";C$:D$ =  CHR$ (4)
1070  PRINT D$;"OPEN";C$: PRINT D$;"READ";C$
1080  INPUT NM,NJ,ML,JL,BCJ,E,HJ,NODE,ND
1090  DIM X(NJ),Y(NJ),J(NM),K(NM),AR(NM),IZ(NM),E(NM)
1100  FOR I = 1 TO NJ: INPUT X(I),Y(I): NEXT I
1110  FOR I = 1 TO NM: INPUT J(I),K(I),AR(I),IZ(I),E(I): NEXT I
1120  PRINT D$;"CLOSE";C$
1130  HOME : HCOLOR= 7: HGR
1140  LET P = 0:Q = 0:R = 0:S = 0
1150  FOR I = 1 TO NJ
1160  IF X(I) > P THEN P = X(I)
1170  IF X(I) < Q THEN Q = X(I)
1180  IF Y(I) > R THEN R = Y(I)
1190  IF Y(I) < S THEN S = Y(I)
1200  NEXT I:X = P - Q:Y = R - S:Z = X
1210  IF Y > Z THEN Z = Y
1220  LET H = 150 / Z:P = 0:R = 0
1230  IF Q < 0 THEN P =  - Q
1240  IF S < 0 THEN R =  - S
1250  FOR I = 1 TO NM:J = J(I):K = K(I)
1260  LET X1 = (X(J) + P) * H + 20
1270  LET X2 = (X(K) + P) * H + 20
1280  LET Y1 = 155 - (Y(J) + R) * H
1290  LET Y2 = 155 - (Y(K) + R) * H
1300  HPLOT X1,Y1 TO X2,Y2: NEXT I
1310  IF L = 1 GOTO 1390
1320  GOSUB 1420: TEXT : HOME : VTAB 10
1330  PRINT "    DO YOU WANT HARD COPY OF THIS PLOT ?"
1340  VTAB 13: PRINT  TAB( 13);"TYPE Y/ N";
1350  INPUT " ";B$:L = 1
1360  IF B$ = "N" GOTO 1400
1370  HOME : VTAB 10: PRINT  TAB( 8);" PLEASE TURN ON PRINTER"
1380  GOSUB 1420: PRINT  CHR$ (4);"PR#1": GOTO 1130
1390  PRINT  CHR$ (17): PRINT  CHR$ (4);"PR#0": TEXT : HOME
1400  PRINT "    INPUT PROGRAM DISK INTO DRIVE"
1410  GOSUB 1420: PRINT D$;"RUN OPTION"
1420  VTAB 24: PRINT  TAB( 3);
1430  PRINT "  PRESS 'RETURN KEY' TO CONTINUE"
1440  INPUT " ";B$: RETURN
1450  END
```

```
1000  REM   ********************
1010  REM   **  PRINT DATA  **
1020  REM   ********************
1030  HOME : PRINT "   INPUT DATA DISK NO.1 INTO DRIVE": VTAB 24
1040  INPUT "    PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1050  PRINT "    NAME OF DATA FILE TO BE PRINTED"
1060  INPUT "          FILE NAME..... ";C$:D$ =  CHR$ (4)
1070  PRINT D$;"OPEN";C$: PRINT D$;"READ";C$
1080  INPUT NM,NJ,ML,JL,BCJ,E,HJ,NODE,ND:BB = 2 * ML
1090  DIM X(NJ),Y(NJ),J(NM),K(NM),AR(NM),IZ(NM),F(ML),H(HJ)
1100  DIM W(BB),P(BB),V1(5,ML),VA(5,ML),H1(5,ML),HA(5,ML),E(NM)
1110  DIM K1(ML),K2(JL),PJ(JL,3),BJ(7 * BCJ),F1(5,ML),FA(5,ML)
1120  FOR I = 1 TO NJ: INPUT X(I),Y(I): NEXT I
1130  FOR I = 1 TO NM: INPUT J(I),K(I),AR(I),IZ(I),E(I): NEXT I
1140  IF ML = 0 GOTO 1240
1150  FOR I = 1 TO ML:I1 = 2 * (I - 1)
1160  INPUT K1(I),W(I1 + 1),W(I1 + 2),F(I),P(I1 + 1),P(I1 + 2)
1170  IF F(I) = 0 GOTO 1190
1180  FOR J = 1 TO F(I): INPUT F1(J,I),FA(J,I): NEXT J
1190  IF P(I1 + 2) = 0 GOTO 1210
1200  FOR J = 1 TO P(I1 + 2): INPUT V1(J,I),VA(J,I): NEXT J
1210  IF P(I1 + 1) = 0 GOTO 1230
1220  FOR J = 1 TO P(I1 + 1): INPUT H1(J,I),HA(J,I): NEXT J
1230  NEXT I
1240  IF JL = 0 GOTO 1270
1250  FOR I = 1 TO JL: INPUT K2(I)
1260  FOR L = 1 TO 3: INPUT PJ(I,L): NEXT L: NEXT I
1270  IF BCJ = 0 GOTO 1300
1280  FOR I = 1 TO BCJ: FOR L = 1 TO 7
1290  LET L1 = 7 * (I - 1) + L: INPUT BJ(L1): NEXT L: NEXT I
1300  IF HJ = 0 GOTO 1330
1310  IF HJ = NJ GOTO 1330
1320  FOR I = 1 TO HJ: INPUT H(I): NEXT I
1330  PRINT D$;"CLOSE";C$:F$ = "N"
1340  LET A$ = "=================================="
1350  PRINT  TAB( 15);"*************"
1360  PRINT  TAB( 16);"INPUT DATA": PRINT  TAB( 15);"*************"
1370  PRINT : PRINT : PRINT : PRINT
1380  PRINT "  NO. OF MEMBERS..........= ";NM: PRINT
1390  PRINT "  NO. OF NODES............= ";NJ: PRINT
1400  PRINT "  NO. OF LOADED MEMBERS...= ";ML: PRINT
1410  PRINT "  NO. OF LODED NODES......= ";JL: PRINT
1420  PRINT "  NO. OF NODES WITH HINGES= ";HJ: PRINT
1430  PRINT "  NO. OF BOUNDARY NODES...= ";BCJ: PRINT
1440  IF F$ = "Y" GOTO 1460
1450  INPUT "  PRESS 'RETURN KEY' TO CONTINUE";B$
1460  PRINT : PRINT : PRINT A$
1470  PRINT " :NODAL PT. COOR.(HINGE=0,RIGID=1):": PRINT A$: PRINT
1480  PRINT "NODE  CASE     X-COOR.     Y-COOR."
1490  PRINT "----  ----     -------     -------"
1500  FOR I = 1 TO NJ: PRINT  SPC( 4 -  LEN ( STR$ (I)));I;
```

```
1510  LET J = 1
1520  IF HJ = 0 GOTO 1560
1530  FOR L = 1 TO HJ: IF H(L) = I THEN J = 0
1540  NEXT L
1550  IF HJ = NJ THEN J = 0
1560  PRINT  SPC( 4);J; SPC( 12 -  LEN ( STR$ (X(I))));X(I);
1570  PRINT  SPC( 12 -  LEN ( STR$ (Y(I))));Y(I)
1580  NEXT I: PRINT
1590  IF F$ = "Y" GOTO 1610
1600  INPUT "  PRESS 'RETURN KEY' TO CONTINUE";B$
1610  PRINT : PRINT : PRINT A$
1620  PRINT "   :MEMBER NODES & PROPERTIES:": PRINT A$: PRINT
1630  PRINT "MEMBER   NODE-J   NODE-K   AREA       INERTIA     MODULUS"
1640  PRINT "------   ------   ------   ----      -------     -------"
1650  FOR I = 1 TO NM: PRINT  SPC( 5 -  LEN ( STR$ (I)));I;
1660  PRINT  SPC( 8 -  LEN ( STR$ (J(I))));J(I);
1670  PRINT  SPC( 9 -  LEN ( STR$ (K(I))));K(I);
1680  PRINT  SPC( 9 -  LEN ( STR$ (AR(I))));AR(I);
1690  PRINT  SPC( 14 -  LEN ( STR$ (IZ(I))));IZ(I);
1700  PRINT  SPC( 12 -  LEN ( STR$ (E(I))));E(I): NEXT I: PRINT
1710  IF F$ = "Y" GOTO 1730
1720  INPUT "  PRESS 'RETURN KEY' TO CONTINUE";B$
1730  IF ML = 0 GOTO 2090
1740  PRINT : PRINT A$
1750  PRINT "   :MEMBER LOADS IN LOCAL COOR.:": PRINT A$
1760  PRINT "MEM. D-LOAD W-LOAD A-LOAD:DIST.   V-LOAD:DIST.   H-LOAD:DIST."
1770  PRINT "---- ------ ------ ------------   ------------   ------------"
1780  FOR I = 1 TO ML:I1 = 2 * (I - 1)
1790  PRINT  SPC( 3 -  LEN ( STR$ (K1(I))));K1(I);
1800  PRINT  SPC( 7 -  LEN ( STR$ ( INT (W(I1 + 1) * 100 + 0.5) / 100)));
1810  PRINT  INT (W(I1 + 1) * 100 + 0.5) / 100;
1820  PRINT  SPC( 7 -  LEN ( STR$ ( INT (W(I1 + 2) * 100 + 0.5) / 100)));
1830  PRINT  INT (W(I1 + 2) * 100 + 0.5) / 100;
1840  LET LL = F(I): IF LL < P(I1 + 2) THEN LL = P(I1 + 2)
1850  IF LL < P(I1 + 1) THEN LL = P(I1 + 1)
1860  FOR J = 1 TO LL
1870  LET L1 =  INT (F1(J,I) * 100 + 0.5) / 100
1880  LET L2 =  INT (FA(J,I) * 100 + 0.5) / 100
1890  LET L3 =  INT (V1(J,I) * 100 + 0.5) / 100
1900  LET L4 =  INT (VA(J,I) * 100 + 0.5) / 100
1910  LET L5 =  INT (H1(J,I) * 100 + 0.5) / 100
1920  LET L6 =  INT (HA(J,I) * 100 + 0.5) / 100
1930  LET J1 = 6: IF J > 1 THEN J1 = 23
1940  LET J2 = 7: IF J > 1 THEN J2 = 24
1950  IF F1(J,I) = 0 GOTO 1970
1960  PRINT  SPC( J2 -  LEN ( STR$ (L1)));L1;" : ";L2;: GOTO 1980
1970  PRINT  SPC( J1);"- : -";
1980  IF V1(J,I) = 0 GOTO 2010
1990  PRINT  SPC( 12 -  LEN ( STR$ (L2) +  STR$ (L3)));L3;" : ";L4;
2000  GOTO 2020
```

```
2010  PRINT  SPC( 11);"- : -";
2020  IF H1(J,I) = 0 GOTO 2050
2030  PRINT  SPC( 12 -  LEN ( STR$ (L4) +  STR$ (L5)));L5;" : ";L5
2040  GOTO 2060
2050  PRINT  SPC( 8);"- : -"
2060  NEXT J: NEXT I: PRINT
2070  IF F$ = "Y" GOTO 2090
2080  INPUT "  PRESS 'RETURN KEY' TO CONTINUE";B$
2090  IF JL = 0 GOTO 2220
2100  PRINT : PRINT : PRINT A$
2110  PRINT "   :NODAL LOADS IN GLOBAL COOR.:": PRINT A$
2120  PRINT : PRINT "NODE   X-FORCE   Y-FORCE    MOMENT"
2130  PRINT "----   -------   -------    ------"
2140  FOR I = 1 TO JL
2150  PRINT  SPC( 4 -  LEN ( STR$ (K2(I))));K2(I);
2160  PRINT  SPC( 9 -  LEN ( STR$ (PJ(I,1))));PJ(I,1);
2170  PRINT  SPC( 10 -  LEN ( STR$ (PJ(I,2))));PJ(I,2);
2180  PRINT  SPC( 10 -  LEN ( STR$ (PJ(I,3))));PJ(I,3)
2190  NEXT I: PRINT
2200  IF F$ = "Y" GOTO 2220
2210  INPUT "  PRESS 'RETURN KEY' TO CONTINUE";B$
2220  IF BCJ = 0 GOTO 2480
2230  PRINT : PRINT A$: PRINT "  :BOUNDARY CONDITIONS AT NODES:"
2240  PRINT A$: PRINT
2250  PRINT "NODE  X Y Z  X-DISPL     Y-DISPL     ROTATION"
2260  PRINT "----  - - -  -------     -------     --------"
2270  FOR I = 1 TO BCJ:I1 = 7 * (I - 1) + 1
2280  PRINT  SPC( 4 -  LEN ( STR$ (BJ(I1))));BJ(I1);
2290  LET L1 =  INT (BJ(I1 + 4) * 1000 + 0.5) / 1000
2300  LET L2 =  INT (BJ(I1 + 5) * 1000 + 0.5) / 1000
2310  LET L3 =  INT (BJ(I1 + 6) * 1000 + 0.5) / 1000
2320  PRINT "  ";BJ(I1 + 1);" ";BJ(I1 + 2);" ";BJ(I1 + 3);
2330  IF BJ(I1 + 1) = 0 GOTO 2390
2340  PRINT  SPC( 7 -  LEN ( STR$ (L1)));L1;
2350  IF BJ(I1 + 2) = 0 GOTO 2400
2360  PRINT  SPC( 12 -  LEN ( STR$ (L2)));L2;
2370  IF BJ(I1 + 3) = 0 GOTO 2410
2380  PRINT  SPC( 12 -  LEN ( STR$ (L3)));L3: GOTO 2420
2390  PRINT  SPC( 6);"-";: GOTO 2350
2400  PRINT  SPC( 11);"-";: GOTO 2370
2410  PRINT  SPC( 11);"-"
2420  NEXT I: PRINT
2430  IF F$ = "Y" GOTO 2480
2440  INPUT "  PRESS 'RETURN KEY' TO CONTINUE";B$
2450  HOME : VTAB 12: HTAB 7: INPUT "DO YOU WANT HARD COPY ? (Y/N) ";F$
2460  IF F$ = "N" GOTO 2490
2470  PRINT D$;"PR#1": GOTO 1350
2480  PRINT D$;"PR#0"
2490  HOME : PRINT "     INPUT PROGRAM DISK INTO DRIVE": VTAB 24
2500  INPUT "    PRESS 'RETURN KEY' TO CONTINUE";B$
2510  PRINT D$;"RUN OPTION"
2520  END
```

```
1000  REM   ******************
1010  REM   **  STIFFNESS  **
1020  REM   ******************
1030  REM
1040  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1050  REM   :                                                :
1060  REM   :   THIS SUBPROGRAM PERFROMS LOCAL & GLOBAL     :
1070  REM   :   STIFFNESS MATRIX AND LOAD VECTOR OF EACH    :
1080  REM   :   MEMBERS. MODIFIED STIFFNESS MATRIX & LOAD   :
1090  REM   :   VECTOR FOR BOUNDARY NODES. SAVE LOCAL        :
1100  REM   :   STIFFNESS & LOAD VECTOR IN FILE 'LOSTIFF',  :
1110  REM   :   MODIFIED STIFFNESS & LOAD VECTOR IN FILE     :
1120  REM   :   ' GLOSTIFF'.                                :
1130  REM   :                                                :
1140  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1150  REM
1160  REM
1170  HOME : PRINT "   INPUT DATA DISK NO.1 INTO DRIVE ": VTAB 24
1180  INPUT "    PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1190  PRINT "NAME OF DATA FILE TO BE FOUND STIFFNESS"
1200  INPUT "         FILE NAME.....";C$:D$ =  CHR$ (4)
1210  PRINT : PRINT "      ** SELECTED PRINT CASE **"
1220  PRINT "         0-NO PRINT"
1230  PRINT "         1-PRINT EVERY DATA ON MONITOR"
1240  PRINT "         2-PRINT EVERY DATA ON LINE PRINTER"
1250  INPUT "         INPUT PRINT CASE    = ";PCASE: HOME
1260  REM
1270  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1280  REM   :   READ INPUT DATA FROM DISKETTE                :
1290  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1300  REM
1310  PRINT D$;"OPEN";C$: PRINT D$;"READ";C$
1320  INPUT NM,NJ,ML,JL,BCJ,E,HJ,NODE,ND:BB = 2 * ML
1330  DIM X(NJ),Y(NJ),J(NM),K(NM),AR(NM),IZ(NM),H(HJ),E(NM)
1340  DIM F(ML),W(BB),K1(ML),K2(JL),F1(5,ML),FA(5,ML),V1(5,ML)
1350  DIM P(BB),H1(5,ML),HA(5,ML),PJ(JL,3),BJ(7 * BCJ),VA(5,ML)
1360  FOR I = 1 TO NJ: INPUT X(I),Y(I): NEXT I
1370  FOR I = 1 TO NM: INPUT J(I),K(I),AR(I),IZ(I),E(I): NEXT I
1380  IF ML = 0 GOTO 1480
1390  FOR I = 1 TO ML:I1 = 2 * (I - 1)
1400  INPUT K1(I),W(I1 + 1),W(I1 + 2),F(I),P(I1 + 1),P(I1 + 2)
1410  IF F(I) = 0 GOTO 1430
1420  FOR J = 1 TO F(I): INPUT F1(J,I),FA(J,I): NEXT J
1430  IF P(I1 + 2) = 0 GOTO 1450
1440  FOR J = 1 TO P(I1 + 2): INPUT V1(J,I),VA(J,I): NEXT J
1450  IF P(I1 + 1) = 0 GOTO 1470
1460  FOR J = 1 TO P(I1 + 1): INPUT H1(J,I),HA(J,I): NEXT J
1470  NEXT I
1480  IF JL = 0 GOTO 1510
1490  FOR I = 1 TO JL: INPUT K2(I): FOR L = 1 TO 3
1500  INPUT PJ(I,L): NEXT L: NEXT I
```

```
1510  IF BCJ = 0 GOTO 1540
1520  FOR I = 1 TO BCJ: FOR L = 1 TO 7
1530  LET L1 = 7 * (I - 1) + L: INPUT BJ(L1): NEXT L: NEXT I
1540  IF HJ = 0 GOTO 1570
1550  IF HJ = NJ GOTO 1570
1560  FOR I = 1 TO HJ: INPUT H(I): NEXT I
1570  PRINT D$;"CLOSE";C$
1580  HOME : PRINT "  INPUT DATA DISK NO.2 INTO DRIVE ": VTAB 24
1590  INPUT "   PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1600  LET MD = ND * NODE:NW = MD * (MD + 1) / 2 + MD
1610  IF PCASE > 0 GOTO 1630
1620  VTAB 10: HTAB 12: FLASH : PRINT "RUNNING STIFFNESS": NORMAL
1630  DIM IQ(NODE),IH(NODE),S(MD,MD),ST(MD)
1640  PRINT D$;"OPEN IQ": PRINT D$;"DELETE IQ"
1650  PRINT D$;"OPEN IQ":NK = MD * (MD + 1) / 2
1660  PRINT D$;"OPEN LOSTIFF": PRINT D$;"DELETE LOSTIFF"
1670  PRINT D$;"OPEN LOSTIFF"
1680  PRINT D$;"OPEN GLOSTIFF"
1690  PRINT D$;"DELETE GLOSTIFF": PRINT D$;"OPEN GLOSTIFF"
1700  FOR LZ = 1 TO NM
1710  LET L1 = X(K(LZ)) - X(J(LZ))
1720  LET L2 = Y(K(LZ)) - Y(J(LZ)):L =  SQR (L1 ^ 2 + L2 ^ 2)
1730  LET X1 = L1 / L:X2 =  - L2 / L:Y1 = L2 / L:Y2 = L1 / L
1740  LET X3 = 0:Y3 = 0:Z1 = 0:Z2 = 0:Z3 = 1
1750  LET IQ(1) = J(LZ):IQ(2) = K(LZ)
1760  LET IH(1) = 1:IH(2) = 1: IF HJ = 0 GOTO 1840
1770  IF HJ = NJ GOTO 1810
1780  FOR I = 1 TO HJ: IF H(I) = J(LZ) THEN IH(1) = 0
1790  IF H(I) = K(LZ) THEN IH(2) = 0
1800  NEXT I: GOTO 1840
1810  LET IH(1) = 0:IH(2) = 0
1820  REM
1830  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1840  REM  :  CALCULATE LOAD VECTOR IN LOCAL COORDINATES  :
1850  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1860  REM
1870  FOR I = 1 TO MD: FOR J = 1 TO MD:S(I,J) = 0: NEXT J: NEXT I
1880  FOR I = 1 TO MD:ST(I) = 0: NEXT I
1890  LET R1 = 0:R2 = 0:R3 = 0:R4 = 0:R5 = 0:R6 = 0
1900  IF ML = 0 GOTO 2510
1910  FOR I = 1 TO ML: IF K1(I) <  > LZ GOTO 2180
1920  LET I1 = 2 * (I - 1)
1930  LET R1 = (W(I1 + 1) * Y1 + W(I1 + 2) * X1) * L / 2
1940  LET R2 = (W(I1 + 1) * Y2 + W(I1 + 2) * X2) * L / 2
1950  LET R3 = (W(I1 + 1) * Y2 + W(I1 + 2) * X2) * L ^ 2 / 12
1960  LET R4 = R1:R5 = R2:R6 =  - R3
1970  IF F(I) = 0 GOTO 2000
1980  FOR J = 1 TO F(I):R1 = R1 + F1(J,I) * (L - FA(J,I)) / L
1990  LET R4 = R4 + F1(J,I) * FA(J,I) / L: NEXT J
2000  IF P(I1 + 2) = 0 GOTO 2090
```

```
2010  FOR J = 1 TO P(I1 + 2):J1 = VA(J,I):J2 = L - VA(J,I)
2020  LET R1 = R1 + V1(J,I) * Y1 * J2 / L
2030  LET R2 = R2 + V1(J,I) * Y2 * J2 ^ 2 * (3 * J1 + J2) / L ^ 3
2040  LET R3 = R3 + V1(J,I) * Y2 * J1 * J2 ^ 2 / L ^ 2
2050  LET R4 = R4 + V1(J,I) * Y1 * J1 / L
2060  LET R5 = R5 + V1(J,I) * Y2 * J1 ^ 2 * (J1 + 3 * J2) / L ^ 3
2070  LET R6 = R6 - V1(J,I) * Y2 * J2 * J1 ^ 2 / L ^ 2
2080  NEXT J
2090  IF P(I1 + 1) = 0 GOTO 2180
2100  FOR J = 1 TO P(I1 + 1):J1 = HA(J,I):J2 = L - HA(J,I)
2110  LET R1 = R1 + H1(J,I) * X1 * J2 / L
2120  LET R2 = R2 + H1(J,I) * X2 * J2 ^ 2 * (3 * J1 + J2) / L ^ 3
2130  LET R3 = R3 + H1(J,I) * X2 * J1 * J2 ^ 2 / L ^ 2
2140  LET R4 = R4 + H1(J,I) * X1 * J1 / L
2150  LET R5 = R5 + H1(J,I) * X2 * J1 ^ 2 * (J1 + 3 * J2) / L ^ 3
2160  LET R6 = R6 - H1(J,I) * X2 * J2 * J1 ^ 2 / L ^ 2
2170  NEXT J
2180  NEXT I
2190  REM
2200  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2210  REM   :   MODIFIED FIXED END FORCES FOR HINGES         :
2220  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2230  REM
2240  IF IH(1) > 0 AND IH(2) > 0 GOTO 2490
2250  IF IH(1) = 0 AND IH(2) > 0 GOTO 2300
2260  IF IH(1) > 0 AND IH(2) = 0 GOTO 2370
2270  IF IH(1) = 0 AND IH(2) = 0 GOTO 2470
2280  REM
2290  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2300  REM       HINGE AT NODE J
2310  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2320  REM
2330  LET R2 = R2 - R3 * 1.5 / L:R5 = R5 + R3 * 1.5 / L
2340  LET R6 = R6 - R3 * 0.5:R3 = 0: GOTO 2490
2350  REM
2360  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2370  REM       HINGE AT NODE K
2380  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2390  REM
2400  LET R2 = R2 - R6 * 1.5 / L:R5 = R5 + R6 * 1.5 / L
2410  LET R3 = R3 - R6 * 0.5:R6 = 0: GOTO 2490
2420  REM
2430  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2440  REM       HINGES AT NODE J & K
2450  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2460  REM
2470  LET R2 = R2 - (R3 + R6) / L:R5 = R5 + (R3 + R6) / L
2480  LET R3 = 0:R6 = 0
2490  LET ST(1) = R1:ST(2) = R2:ST(3) = R3
2500  LET ST(4) = R4:ST(5) = R5:ST(6) = R6
```

```
2510  REM
2520  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2530  REM   :   CALCULATE LOCAL STIFFNESS                     :
2540  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2550  REM
2560  LET DI = E(LZ) * IZ(LZ) / L ^ 3
2570  REM   **  CALCULATE AXIAL COMPONENT  **
2580  LET S(1,1) = AR(LZ) * E(LZ) / L:S(1,4) =  - S(1,1):S(4,4) = S(1,1)
2590  IF IH(1) > 0 AND IH(2) > 0 GOTO 2650
2600  IF IH(1) = 0 AND IH(2) > 0 GOTO 2740
2610  IF IH(1) > 0 AND IH(2) = 0 GOTO 2820
2620  IF IH(1) = 0 AND IH(2) = 0 GOTO 2870
2630  REM
2640  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2650  REM       NO HINGES
2660  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2670  REM
2680  LET S(2,2) = 12 * DI:S(2,3) = 6 * L * DI:S(2,5) =  - S(2,2)
2690  LET S(2,6) = S(2,3):S(3,3) = 4 * L ^ 2 * DI:S(3,5) =  - S(2,3)
2700  LET S(3,6) = 2 * L ^ 2 * DI:S(5,5) = S(2,2):S(5,6) =  - S(2,3)
2710  LET S(6,6) = S(3,3): GOTO 2870
2720  REM
2730  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2740  REM       HINGE AT NODE J
2750  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2760  REM
2770  LET S(2,2) = 3 * DI:S(2,5) =  - S(2,2):S(2,6) = 3 * L * DI
2780  LET S(5,5) = S(2,2):S(5,6) =  - S(2,6):S(6,6) = 3 * L ^ 2 * DI
2790  GOTO 2870
2800  REM
2810  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2820  REM       HINGE AT NODE K
2830  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2840  REM
2850  LET S(2,2) = 3 * DI:S(2,3) = 3 * L * DI:S(2,5) =  - S(2,2)
2860  LET S(3,3) = 3 * L ^ 2 * DI:S(3,5) =  - S(2,3):S(5,5) = S(2,2)
2870  FOR I = 1 TO MD: FOR J = I TO MD
2880  LET S(J,I) = S(I,J): NEXT J: NEXT I
2890  REM
2900  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2910  REM   :   SAVE LOCAL STIFFNESS & LOAD VECTOR INTO      :
2920  REM   :   DATA FILE ' LOSTIFF'                         :
2930  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2940  REM
2950  PRINT D$;"WRITE LOSTIFF"
2960  PRINT Y1: PRINT Y2: PRINT J(LZ): PRINT K(LZ)
2970  FOR I = 1 TO MD: FOR J = I TO MD
2980  PRINT S(I,J): NEXT J: NEXT I
2990  FOR I = 1 TO MD: PRINT ST(I): NEXT I: PRINT D$
3000  IF PCASE = 0 GOTO 3070
```

```
3010  IF PCASE = 2 THEN  PRINT D$;"PR#1"
3020  PRINT "***  MEMBER NO. ";LZ;"  ***": PRINT
3030  PRINT "LOCAL STIFFNESS": PRINT "==============="
3040  PRINT : GOSUB 3760: PRINT
3050  REM
3060  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3070  REM   :   TRANSFORM FIXED END FORCES INTO GLOBAL       :
3080  REM   :   COORDINATES                                  :
3090  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3100  REM
3110  LET ST(1) = R1 * X1 + R2 * X2:ST(2) = R1 * Y1 + R2 * Y2
3120  LET ST(3) = R3:ST(4) = R4 * X1 + R5 * X2
3130  LET ST(5) = R4 * Y1 + R5 * Y2:ST(6) = R6
3140  IF JL = 0 GOTO 3270
3150  REM
3160  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3170  REM   :   SEMBLE JOINT LOADS INTO LOAD VECTOR          :
3180  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3190  REM
3200  FOR I = 1 TO JL: IF K2(I) <  > K(LZ) GOTO 3230
3210  LET ST(4) = ST(4) + PJ(I,1):ST(5) = ST(5) + PJ(I,2)
3220  LET ST(6) = ST(6) + PJ(I,3):K2(I) =  - K2(I)
3230  IF K2(I) <  > J(LZ) GOTO 3260
3240  LET ST(1) = ST(1) + PJ(I,1):ST(2) = ST(2) + PJ(I,2)
3250  LET ST(3) = ST(3) + PJ(I,3):K2(I) =  - K2(I)
3260  NEXT I
3270  REM
3280  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3290  REM   :   CALCULATE GLOBAL STIFFNESS                   :
3300  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3310  REM
3320  FOR II = 1 TO MD STEP ND: FOR I = II TO MD
3330  LET G1 = S(I,II):G2 = S(I,II + 1):G3 = S(I,II + 2)
3340  LET S(I,II) = G1 * X1 + G2 * X2 + G3 * X3
3350  LET S(I,II + 1) = G1 * Y1 + G2 * Y2 + G3 * Y3
3360  LET S(I,II + 2) = G1 * Z1 + G2 * Z2 + G3 * Z3
3370  NEXT I: NEXT II
3380  FOR II = ND TO MD STEP ND: FOR J = 1 TO II
3390  LET G1 = S(II - 2,J):G2 = S(II - 1,J):G3 = S(II,J)
3400  LET S(II - 2,J) = G1 * X1 + G2 * X2 + G3 * X3
3410  LET S(II - 1,J) = G1 * Y1 + G2 * Y2 + G3 * Y3
3420  LET S(II,J) = G1 * Z1 + G2 * Z2 + G3 * Z3
3430  NEXT J: NEXT II
3440  FOR J = 1 TO MD: FOR I = J TO MD
3450  LET S(J,I) = S(I,J): NEXT I: NEXT J
3460  IF PCASE = 0 GOTO 3500
3470  IF PCASE = 2 THEN  PRINT D$;"PR#1"
3480  PRINT "GLOBAL STIFFNESS": PRINT "================"
3490  PRINT : GOSUB 3760: PRINT
3500  IF BCJ = 0 GOTO 3850
```

```
3510  REM
3520  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3530  REM   :   MODIFIED ROUNDARY CONDITIONS                  :
3540  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3550  REM
3560  FOR I = 1 TO BCJ:I1 = 7 * (I - 1) + 1
3570  IF BJ(I1) <  > J(LZ) GOTO 3590
3580  LET J1 = 0: GOSUB 3640
3590  IF BJ(I1) <  > K(LZ) GOTO 3610
3600  LET J1 = 3: GOSUB 3640
3610  NEXT I
3620  FOR I = 1 TO MD: FOR J = I TO MD
3630  LET S(J,I) = S(I,J): NEXT J: NEXT I: GOTO 3850
3640  FOR J = 1 TO 3:I2 = I1 + J
3650  IF BJ(I2) = 0 GOTO 3730
3660  LET J2 = J1 + J:I3 = I1 + 3 + J
3670  FOR K = 1 TO J2:ST(K) = ST(K) - S(K,J2) * BJ(I3)
3680  LET S(K,J2) = 0: IF K < J2 GOTO 3700
3690  LET ST(K) = BJ(I3):S(K,J2) = 1
3700  NEXT K: IF J2 = MD GOTO 3730
3710  FOR K = J2 + 1 TO MD:ST(K) = ST(K) - S(J2,K) * BJ(I3)
3720  LET S(J2,K) = 0: NEXT K
3730  NEXT J: RETURN
3740  REM
3750  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3760  REM   :   PRINT STIFFNESS MATRIX & LOAD VECTOR          :
3770  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3780  REM
3790  FOR I = 1 TO MD:A2 =  INT (ST(I) * 1000 + 0.5) / 1000
3800  FOR J = 1 TO MD:A1 =  INT (S(I,J) * 1000 + 0.5) / 1000
3810  PRINT  SPC( 10 -  LEN ( STR$ (A1)));A1;: NEXT J
3820  PRINT  SPC( 15 -  LEN ( STR$ (A2)));A2: NEXT I
3830  IF PCASE = 2 THEN  PRINT D$;"PR#0"
3840  RETURN
3850  IF PCASE = 0 GOTO 3950
3860  IF PCASE = 2 THEN  PRINT D$;"PR#1"
3870  PRINT "MODIFIED STIFFNESS": PRINT "=================="
3880  PRINT : GOSUB 3790: PRINT
3890  REM
3900  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3910  REM   :   WRITE MODIFIED STIFFNESS & LOAD VECTOR INTO   :
3920  REM   :   DATA FILE 'GLOSTIFF'                          :
3930  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
3940  REM
3950  PRINT D$;"WRITE GLOSTIFF"
3960  FOR I = 1 TO MD: FOR J = I TO MD
3970  PRINT S(I,J): NEXT J: NEXT I
3980  FOR I = 1 TO MD: PRINT ST(I): NEXT I: PRINT D$
3990  PRINT D$;"WRITE IQ": FOR I = 1 TO NODE: PRINT IQ(I)
4000  NEXT I: PRINT D$: NEXT LZ: PRINT D$;"CLOSE LOSTIFF"
```

```
4010  PRINT D$;"CLOSE GLOSTIFF": PRINT D$;"CLOSE IQ"
4020  PRINT D$;"OPEN DIMEN": PRINT D$;"DELETE DIMEN"
4030  PRINT D$;"OPEN DIMEN": PRINT D$;"WRITE DIMEN": PRINT PCASE
4040  PRINT ND: PRINT NODE: PRINT MD: PRINT NW: PRINT NM: PRINT NJ
4050  PRINT D$;"CLOSE DIMEN": HOME
4060  PRINT "     INPUT PROGRAM DISK INTO DRIVE ": VTAB 24
4070  INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
4080  PRINT D$;"RUN SETFRONT"
4090  END
```

```
1000  REM    ****************
1010  REM    **  SETFRONT  **
1020  REM    ****************
1030  REM
1040  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1050  REM   :   THIS SUBPROGRAM PERFROMS MOC ARRAY (FRONT)    :
1060  REM   :   TO STORE NODES WHICH WILL BE ELIMINATED       :
1070  REM   :   FROM CURRENT FRONT AND NODES WHICH WILL       :
1080  REM   :   ERMAIN IN NEXT FRONT.                         :
1090  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1100  REM
1110  HOME : PRINT "   INPUT DATA DISK NO.2 INTO DRIVE": VTAB 24
1120  INPUT "    PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1130  LET D$ =  CHR$ (4): PRINT D$;"OPEN DIMEN"
1140  PRINT D$;"READ DIMEN": INPUT PCASE,ND,NODE,MD,NW,NM,NJ
1150  PRINT D$;"CLOSE DIMEN":KK =  INT ((6000 - NODE - 2 * NJ) / 2)
1160  DIM IQ(NODE),IC(NJ),IB(NJ),LQ(KK),MOC(KK)
1170  PRINT D$;"OPEN IQ": PRINT D$;"OPEN DFRONT"
1180  PRINT D$;"DELETE DFRONT": PRINT D$;"OPEN DFRONT"
1190  LET MF = 0:MN = 0:ME = MD * (MD + 1) / 2 + MD
1200  LET MI = 0:MJ = 0:KS = NODE * ND:NC = 0:NI = 0
1210  FOR I = 1 TO NJ:IB(I) = 0: NEXT I
1220  VTAB 10: HTAB 15: FLASH : PRINT "RUNNING SETFRONT": NORMAL
1230  IF PCASE = 2 THEN  PRINT D$;"PR#1"
1240  FOR LZ = 1 TO NM: PRINT D$;"READ IQ"
1250  FOR I = 1 TO NODE: INPUT IQ(I): NEXT I: PRINT D$
1260  FOR I = 1 TO NODE:K = IQ(I):IB(K) = IB(K) + 1: NEXT I
1270  NEXT LZ: FOR I = 1 TO NJ:IC(I) = IB(I): NEXT I
1280  PRINT D$;"OPEN IQ": FOR LZ = 1 TO NM: PRINT D$;"READ IQ"
1290  FOR I = 1 TO NODE: INPUT IQ(I): NEXT I: PRINT D$
1300  FOR I = 1 TO NODE:K = IQ(I):IC(K) = IC(K) - 1: NEXT I
1310  GOSUB 1500: NEXT LZ: GOSUB 1500
1320  REM
1330  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1340  REM   :   COMPUTE OTHERS STORAGE REQUIREMENTS          :
1350  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1360  REM
1370  LET IX = 2 * NJ + NODE + 2 * MN
1380  LET IY = 3 * MN + NODE + MD + MF
1390  PRINT D$;"PR#1"
1400  PRINT "MINIMUM STORAGE REQUIREMENTS"
1410  PRINT "  FOR SETFRONT.........= ";IX
1420  PRINT "  FOR COMFRONT.........= ";IY
1430  PRINT "  FOR SEMRED...........= ";NC
1440  PRINT : PRINT D$;"PR#0": GOTO 2000
1450  REM
1460  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1470  REM   :   FIND NODES IN MOC ARRAY (FRONT)               :
1480  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1490  REM
1500  LET M = 0:N = 0: FOR I = 1 TO NJ
```

```
1510  IF IB(I) = IC(I) GOTO 1540
1520  IF IC(I) = 0 THEN IB(I) = 0
1530  LET N = N + 1:LQ(N) = I
1540  NEXT I
1550  IF N = 0 GOTO 1730
1560  REM
1570  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1580  REM   :   STORE NODES WHICH WILL BE ELIMINATE IN MOC    :
1590  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1600  REM
1610  FOR I = 1 TO N:MOC(I) = 0:J = LQ(I): IF IC(J) <  > 0 GOTO 1630
1620  LET LQ(I) = 0:M = M + 1:MOC(M) = J
1630  NEXT I:J = M + 1
1640  REM
1650  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1660  REM   :   STORE NODES WHICH WILL REMAIN IN NEXT FRONT  :
1670  REM   :   IN MOC ARRAY                                 :
1680  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1690  REM
1700  FOR I = 1 TO N: IF LQ(I) = 0 GOTO 1720
1710  LET MOC(J) = LQ(I):LQ(I) = 0:J = J + 1
1720  NEXT I
1730  LET MQ = M * ND:NQ = N * ND
1740  LET NF = NQ * 3 + MD + NW + NQ * (NQ + 1) / 2
1750  LET NZ = NQ * (NQ + 1) / 2
1760  IF NI < NZ THEN NI = NZ
1770  IF NC < NF THEN NC = NF
1780  IF MN < N THEN MN = N
1790  IF MF < NQ THEN MF = NQ
1800  REM
1810  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1820  REM   :   WRITE ELEMENTS FROM MOC ARRAY ON DISKETTE    :
1830  REM   :   IN DATA FILE 'DFRONT'                        :
1840  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1850  REM
1860  PRINT D$;"WRITE DFRONT"
1870  PRINT MQ: PRINT NQ: PRINT M: PRINT N: PRINT N
1880  IF N = 0 GOTO 1900
1890  FOR I = 1 TO N: PRINT MOC(I): NEXT I
1900  PRINT D$: IF PCASE = 0 GOTO 1990
1910  PRINT "MENBER NO. ";LZ: PRINT "=============": PRINT
1920  PRINT "IB    ";: FOR I = 1 TO NJ: PRINT IB(I);
1930  PRINT  SPC( 8 -  LEN ( STR$ (IB(I))));: NEXT I: PRINT
1940  PRINT "IC    ";: FOR I = 1 TO NJ: PRINT IC(I);
1950  PRINT  SPC( 8 -  LEN ( STR$ (IC(I))));: NEXT I: PRINT
1960  PRINT "MQ= ";MQ;"  ";"NQ= ";NQ;"  ";"M= ";M;"  ";"N= ";N
1970  PRINT "MOC   ";: FOR I = 1 TO N
1980  PRINT MOC(I); SPC( 8 -  LEN ( STR$ (MOC(I))));: NEXT I: PRINT
1990  RETURN
2000  PRINT D$;"CLOSE DFRONT": PRINT D$;"CLOSE IQ"
```

```
2010  PRINT D$;"OPEN DIMEN": PRINT D$;"DELETE DIMEN"
2020  PRINT D$;"OPEN DIMEN": PRINT D$;"WRITE DIMEN": PRINT PCASE
2030  PRINT ND: PRINT NODE: PRINT MD: PRINT NW: PRINT NM
2040  PRINT NJ: PRINT MF: PRINT MN: PRINT NI
2050  PRINT D$;"CLOSE DIMEN"
2060  HOME : PRINT "     INPUT PROGRAM DISK INTO DRIVE": VTAB 24
2070  INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
2080  PRINT D$;"RUN COMFRONT"
2090  END
```

```
1000  REM    ****************
1010  REM    **  COMFRONT  **
1020  REM    ****************
1030  REM
1040  REM
1050  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1060  REM   :                                                :
1070  REM   :   THIS PROGRAM COMPARES CURRENT FRONT WITH     :
1080  REM   :   NEXT FRONT, REORDER LOCATION OF NODES IN     :
1090  REM   :   FRONT SO THAT NODES WHICH WILL REMAIN IN     :
1100  REM   :   NEXT FRONT ARE IN THE SAME LOCATION AS IN    :
1110  REM   :   CURRENT FRONT.                               :
1120  REM   :                                                :
1130  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1140  REM
1150  REM
1160  HOME : PRINT "   INPUT DATA DISK NO.2 INTO DRIVE": VTAB 24
1170  INPUT "   PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1180  VTAB 10: HTAB 13: FLASH : PRINT "RUNNING COMFRONT": NORMAL
1190  LET D$ =  CHR$ (4): PRINT D$;"OPEN DIMEN"
1200  PRINT D$;"READ DIMEN": INPUT PCASE,ND,NODE,MD,NW,NM,NJ,MF,MN,NI
1210  PRINT D$;"CLOSE DIMEN": IF PCASE = 2 THEN  PRINT D$;"PR#1"
1220  LET KS = NODE * ND:MP = (KS + MF + 5) * 5
1230  DIM AMOC(MN),BMOC(MN),LE(MD),LF(MF),IQ(NODE),LQ(MN)
1240  PRINT D$;"OPEN DFRONT": PRINT D$;"OPEN IQ"
1250  PRINT D$;"OPEN FRONTIQ": PRINT D$;"DELETE FRONTIQ"
1260  PRINT D$;"OPEN FRONTIQ,L";MP:IM = 0
1270  PRINT D$;"READ DFRONT": INPUT MQ,NQ,M1,N1,NA
1280  FOR I = 1 TO NA: INPUT AMOC(I): NEXT I: PRINT D$
1290  FOR LZ = 1 TO NM
1300  PRINT D$;"READ DFRONT": INPUT MR,NR,M2,N2,NB
1310  IF NB = 0 GOTO 1330
1320  FOR I = 1 TO NB: INPUT BMOC(I): NEXT I
1330  PRINT D$
1340  PRINT D$;"READ IQ": FOR I = 1 TO NODE: INPUT IQ(I)
1350  NEXT I: PRINT D$: IF LZ =  > NM - 1 GOTO 1530
1360  LET K1 = M1 + 1:K2 = M2 + 1:IX = K1:IY = N1
 1370  IF K1 < K2 THEN IX = K2
 1380  IF N1 > N2 THEN IY = N2
 1390  IF IX > IY GOTO 1530
 1400  FOR I = 1 TO N2:LQ(I) = BMOC(I):BMOC(I) = 0: NEXT I
 1410  FOR I = IX TO IY:K = AMOC(I)
 1420  FOR J = K2 TO N2: IF LQ(J) = K GOTO 1440
 1430  NEXT J: GOTO 1450
 1440  LET BMOC(I) = K:LQ(J) = 0
 1450  NEXT I
 1460  FOR I = 1 TO N2: IF LQ(I) = 0 GOTO 1500
 1470  FOR J = 1 TO N2: IF BMOC(J) = 0 GOTO 1490
 1480  NEXT J
 1490  LET BMOC(J) = LQ(I):LQ(I) = 0
 1500  NEXT I
```

```
1510  REM
1520  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1530  REM  :   LOCATE LOCATION OF DOF IN CURRENT FRONT &     :
1540  REM  :   NEXT FRONT                                    :
1550  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1560  REM
1570  LET KS = NODE * ND:KQ = MQ + 1: IF PCASE = 0 GOTO 1610
1580  PRINT "MEMBER NO. ";LZ: PRINT "=============": PRINT
1590  PRINT "KS= ";KS,"M1= ";M1
1600  PRINT "N1= ";N1,"MQ= ";MQ: PRINT "NQ= ";NQ
1610  FOR I = 1 TO NODE: FOR J = 1 TO NA
1620  IF IQ(I) = AMOC(J) THEN LQ(I) = J
1630  NEXT J: NEXT I:L = 0
1640  FOR I = 1 TO NODE:J2 = LQ(I) * ND:J1 = J2 - ND + 1
1650  FOR J = J1 TO J2:L = L + 1:LE(L) = J
1660  NEXT J: NEXT I: IF PCASE = 0 GOTO 1710
1670  PRINT "IQ    ";: FOR I = 1 TO NODE: PRINT IQ(I);
1680  PRINT  SPC( 8 -  LEN ( STR$ (IQ(I))));: NEXT I: PRINT
1690  PRINT "LQ    ";: FOR I = 1 TO NODE: PRINT LQ(I);
1700  PRINT  SPC( 8 -  LEN ( STR$ (LQ(I))));: NEXT I: PRINT
1710  LET L = MQ: IF MQ < 1 GOTO 1730
1720  FOR I = 1 TO MQ:LF(I) = 0: NEXT I
1730  LET K1 = M1 + 1: IF K1 > NA GOTO 1860
1740  FOR I = K1 TO NA: FOR J = 1 TO NB
1750  IF AMOC(I) = BMOC(J) THEN LQ(I) = J
1760  NEXT J: NEXT I
1770  FOR I = K1 TO NA:J2 = LQ(I) * ND:J1 = J2 - ND + 1
1780  FOR J = J1 TO J2:L = L + 1:LF(L) = J
1790  NEXT J: NEXT I
1800  REM
1810  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1820  REM  :   WRITE ELEMENTS OF LE & LF ARRAY ON DISKETTE   :
1830  REM  :   IN DATA FILE 'FRONTIQ'                        :
1840  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1850  REM
1860  PRINT D$;"WRITE FRONTIQ,R";IM
1870  PRINT MQ: PRINT KQ: PRINT NQ: PRINT NR: PRINT KS
1880  FOR I = 1 TO KS: PRINT LE(I): NEXT I
1890  FOR I = 1 TO NQ: PRINT LF(I): NEXT I: PRINT D$
1900  IF PCASE = 0 GOTO 2010
1910  PRINT "AMOC  ";: FOR I = 1 TO NA: PRINT AMOC(I);
1920  PRINT  SPC( 8 -  LEN ( STR$ (AMOC(I))));: NEXT I: PRINT
1930  PRINT "BMOC  ";: FOR I = 1 TO NB: PRINT BMOC(I);
1940  PRINT  SPC( 8 -  LEN ( STR$ (BMOC(I))));: NEXT I: PRINT
1950  PRINT "LQ    ";: FOR I = 1 TO NA: PRINT LQ(I);
1960  PRINT  SPC( 8 -  LEN ( STR$ (LQ(I))));: NEXT I: PRINT
1970  PRINT "LE    ";: FOR I = 1 TO KS: PRINT LE(I);
1980  PRINT  SPC( 8 -  LEN ( STR$ (LE(I))));: NEXT I: PRINT
1990  PRINT "LF    ";: FOR I = 1 TO NQ: PRINT LF(I);
2000  PRINT  SPC( 8 -  LEN ( STR$ (LF(I))));: NEXT I: PRINT
```

```
2010  LET NP = MQ * NQ - (MQ * (MQ - 1)) / 2
2020  IF MI < NP THEN MI = NP
2030  IF MJ < MQ THEN MJ = MQ
2040  LET MK = (MI + MJ + 1) * 15
2050  FOR I = 1 TO NB:AMOC(I) = BMOC(I): NEXT I
2060  LET M1 = M2:N1 = N2:NA = NB:MQ = MR:NQ = NR
2070  LET IM = IM + 1: NEXT LZ: IF PCASE = 2 THEN  PRINT D$;"PR#0"
2080  PRINT D$;"CLOSE IQ": PRINT D$;"CLOSE DFRONT"
2090  PRINT D$;"CLOSE FRONTIQ": PRINT D$;"OPEN DIMEN"
2100  PRINT D$;"DELETE DIMEN"
2110  PRINT D$;"OPEN DIMEN": PRINT D$;"WRITE DIMEN": PRINT PCASE
2120  PRINT ND: PRINT NODE: PRINT MD: PRINT NW: PRINT NM
2130  PRINT NJ: PRINT MF: PRINT MN: PRINT NI: PRINT ME
2140  PRINT MK: PRINT MP: PRINT D$;"CLOSE DIMEN"
2150  PRINT D$;"DELETE DFRONT": PRINT D$;"DELETE IQ"
2160  HOME : PRINT "     INPUT PROGRAM DISK INTO DRIVE": VTAB 24
2170  INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
2180  PRINT D$;"RUN  SEMRED"
2190  END
```

```
1000  REM    ***************
1010  REM    **  SEMRED  **
1020  REM    ***************
1030  REM
1040  REM
1050  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1060  REM   :                                                :
1070  REM   :   THIS PROGRAM ASSEMBLES STIFFNESS MATRIX &    :
1080  REM   :   LOAD VECTOR OF EACH ELEMENTS AND REDUCE      :
1090  REM   :   DOF OF NODES THAT WILL BE ELIMINATED FROM    :
1100  REM   :   FRONT.                                       :
1110  REM   :                                                :
1120  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1130  REM
1140  REM
1150  HOME : PRINT "   INPUT DATA DISK NO.2 INTO DRIVE": VTAB 24
1160  INPUT "   PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1170  VTAB 10: HTAB 14: FLASH : PRINT "RUNNING SEMRED": NORMAL
1180  LET D$ =  CHR$ (4)
1190  PRINT D$;"OPEN DIMEN": PRINT D$;"READ DIMEN"
1200  INPUT PCASE,ND,NODE,MD,NW,NM,NJ,MF,MN,NI,ME,MK,MP
1210  PRINT D$;"CLOSE DIMEN": IF PCASE = 2 THEN  PRINT D$;"PR#1"
1220  PRINT D$;"OPEN GLOSTIFF":NK = MD * (MD + 1) / 2
1230  PRINT D$;"OPEN MQ": PRINT D$;"DELETE MQ"
1240  PRINT D$;"OPEN MQ,L";MK:IM = 0:IQ = 0
1250  PRINT D$;"OPEN FRONTIQ,L";MP
1260  DIM A(NI),B(MF),S(NW),LE(MD),LF(MF),NO(MF)
1270  FOR I = 1 TO NI:A(I) = 0: NEXT I: FOR I = 1 TO MF:B(I) = 0: NEXT I
1280  FOR LZ = 1 TO NM
1290  PRINT D$;"READ FRONTIQ,R";IM
1300  INPUT MQ,KQ,NQ,NR,KS: FOR I = 1 TO KS: INPUT LE(I)
1310  NEXT I: FOR I = 1 TO NQ: INPUT LF(I): NEXT I: PRINT D$
1320  PRINT D$;"READ GLOSTIFF": FOR I = 1 TO NW
1330  INPUT S(I): NEXT I: PRINT D$:NB = 0: GOTO 1380
1340  LET NO(1) = 1:LL = NN + 2: FOR I = 2 TO NN
1350  LET NO(I) = NO(I - 1) + LL - I: NEXT I: RETURN
1360  REM
1370  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1380  REM   :   DETERMINE HOW MANY ROWS CAN BE STORED IN     :
1390  REM   :   FRONT                                        :
1400  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1410  REM
1420  LET NN = NQ: GOSUB 1340
1430  LET NB = NB + 1: IF NB > 1 GOTO 1460
1440  LET IR = 1:NE = MQ * NQ - (MQ * (MQ - 1)) / 2
1450  LET MW = 0:JR = NQ: GOTO 1570
1460  LET IR = JR + 1:JR = IR:II = JR
1470  IF II > NQ GOTO 1490
1480  FOR I = II TO NQ:JR = JR + 1: NEXT I
1490  LET JR = JR - 1: IF NB <  = 1 GOTO 1570
1500  LET MW = (IR - 1) * NQ - ((IR - 2) * (IR - 1)) / 2 - NE
```

```
1510  REM
1520  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1530  REM  :  SEMBLE STIFFNESS MATRIX & LOAD VECTOR WITH    :
1540  REM  :  NEW ELEMENT                                   :
1550  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1560  REM
1570  LET J = NW - KS:L = 0
1580  FOR M = 1 TO KS:J = J + 1:I = LE(M)
1590  IF I < IR OR I > JR GOTO 1610
1600  LET B(I) = B(I) + S(J)
1610  FOR N = M TO KS:L = L + 1:I = LE(M)
1620  IF LE(M) > LE(N) THEN I = LE(N)
1630  IF I < IR OR I > JR GOTO 1680
1640  LET JJ = LE(M)
1650  IF LE(M) < LE(N) THEN JJ = LE(N)
1660  LET K = ND(I) - I + JJ - MW
1670  LET A(K) = A(K) + S(L)
1680  NEXT N: NEXT M: GOTO 1800
1690  PRINT "LZ= ";LZ: PRINT "======"
1700  PRINT "IR= ";IR;"  ";"JR= ";JR;"  ";"NQ= ";NQ;"  ";"MW= ";MW
1710  LET Q = 0: FOR K = IR TO JR:I1 = ND(K) - MW:I2 = I1 + NQ - K
1720  LET L = K - 1: IF L = 0 GOTO 1740
1730  FOR I = 1 TO L: PRINT  SPC( 10);Q;: NEXT I
1740  FOR I = I1 TO I2
1750  PRINT  SPC( 11 -  LEN ( STR$ ( INT (A(I) * 100 + 0.5) / 100)));
1760  PRINT  INT (A(I) * 100 + 0.5) / 100;: NEXT I
1770  PRINT  SPC( 11 -  LEN ( STR$ ( INT (B(K) * 100 + 0.5) / 100)));
1780  PRINT  INT (B(K) * 100 + 0.5) / 100: NEXT K
1790  PRINT : PRINT : RETURN
1800  IF PCASE > 0 THEN  GOSUB 1690
1810  IF NB = 1 THEN IR = 2
1820  IF MQ <  = 0 GOTO 1980
1830  IF IR > JR GOTO 1980
1840  REM
1850  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1860  REM  :  REDUCE DOF OF NODE WHICH WAS ELIMINATED       :
1870  REM  :  FROM FRONT BY GAUSS ELIMINATION               :
1880  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
1890  REM
1900  FOR I = IR TO JR:I0 = ND(I) - I - MW:J2 = I - 1
1910  IF I > MQ THEN J2 = MQ
1920  FOR J = 1 TO J2:J0 = ND(J) - J
1930  IF A(J0 + J) = 0 GOTO 1970
1940  LET C =  - A(J0 + I) / A(J0 + J)
1950  LET B(I) = B(I) + B(J) * C: FOR K = I TO NQ
1960  LET A(I0 + K) = A(I0 + K) + A(J0 + K) * C: NEXT K
1970  NEXT J: NEXT I
1980  IF PCASE = 0 GOTO 2020
1990  IF NB > 1 GOTO 2010
2000  LET IR = 1: GOSUB 1690: GOTO 2020
```

```
2010  GOSUB 1690
2020  IF MQ <  = 0 GOTO 2080
2030  PRINT D$;"WRITE MQ,R";IQ: PRINT NE
2040  FOR I = 1 TO NE: PRINT A(I): NEXT I
2050  FOR I = 1 TO MQ: PRINT B(I): NEXT I: PRINT D$
2060  LET IQ = IQ + 1: FOR I = 1 TO NE:A(I) = 0: NEXT I
2070  FOR I = 1 TO MQ:B(I) = 0: NEXT I
2080  LET IN = NR - NQ: IF IN < 0 GOTO 2500
2090  IF IN = 0 GOTO 2600
2100  IF IN > 0 GOTO 2550
2110  LET KK = 0: IF NQ =  > NT GOTO 2130
2120  LET J = NQ + 1: FOR I = J TO NT:LF(I) = 0: NEXT I
2130  IF KQ > NQ THEN  RETURN
2140  FOR M = KQ TO NQ
2150  LET N = LF(M): IF N = M OR N = 0 GOTO 2440
2160  LET LF(M) = LF(N):LF(N) = N: IF M > N GOTO 2180
2170  LET K = M:L = N: GOTO 2210
2180  LET K = N:L = M
2190  REM
2200  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2210  REM   :   INTERCHANGE ROWS M & N IN CURRENT FRONT      :
2220  REM   :   TO ACCOMODATE NEXT FRONT                    :
2230  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2240  REM
2250  LET C = B(K):B(K) = B(L):B(L) = C
2260  LET MS = NO(K) - K:NS = NO(L) - L:C = A(MS + K)
2270  LET A(MS + K) = A(NS + L):A(NS + L) = C
2280  IF K = 1 GOTO 2310
2290  FOR I = 1 TO K - 1:IS = NO(I) - I:C = A(IS + K)
2300  LET A(IS + K) = A(IS + L):A(IS + L) = C: NEXT I
2310  IF K + 1 = L GOTO 2340
2320  FOR J = K + 1 TO L - 1:JS = NO(J) - J:C = A(MS + J)
2330  LET A(MS + J) = A(JS + L):A(JS + L) = C: NEXT J
2340  IF L = NT GOTO 2420
2350  REM
2360  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2370  REM   :   EXPAND FRONT SPACE IF NEXT FRONT IS BIGGER   :
2380  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2390  REM
2400  FOR J = L + 1 TO NT:C = A(MS + J):A(MS + J) = A(NS + J)
2410  LET A(NS + J) = C: NEXT J
2420  LET KK = KK + 1: IF KK > NT THEN  STOP
2430  GOTO 2150
2440  NEXT M: RETURN
2450  REM
2460  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2470  REM   :   SHRINK FRONT SPACE IF NEXT FRONT IS SMALLER  :
2480  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2490  REM
2500  LET NT = NQ: GOSUB 2110
```

```
2510  LET KL = 0: FOR I = 2 TO NR:KL = KL + IN:M = ND(I)
2520  LET N = M + NR - I: FOR J = M TO N:A(J + KL) = A(J)
2530  LET A(J) = 0: NEXT J: NEXT I
2540  LET NN = NR: GOSUB 1340: GOTO 2610
2550  LET M = ND(NQ) + 1:KL = IN * NQ:L = NQ - 1
2560  FOR I = 1 TO L:M = M - I + 1:KL = KL - IN:N = M + KL
2570  FOR J = 1 TO I:A(N - J) = A(M - J):A(M - J) = 0
2580  NEXT J: NEXT I
2590  LET NN = NR: GOSUB 1340
2600  LET NT = NR: GOSUB 2110
2610  IF PCASE = 0 GOTO 2630
2620  LET IR = 1: GOSUB 1690
2630  LET IM = IM + 1: NEXT LZ: PRINT D$;"CLOSE GLOSTIFF"
2640  IF NI < NW THEN NI = NW
2650  IF PCASE = 2 THEN  PRINT D$;"PR#0"
2660  PRINT D$;"CLOSE FRONTIQ": PRINT D$;"CLOSE MQ"
2670  PRINT D$;"OPEN DIMEN": PRINT D$;"DELETE DIMEN"
2680  PRINT D$;"OPEN DIMEN": PRINT D$;"WRITE DIMEN": PRINT PCASE
2690  PRINT ND: PRINT NODE: PRINT MD: PRINT NW: PRINT NM
2700  PRINT NJ: PRINT MF: PRINT MN: PRINT NI: PRINT ME
2710  PRINT MK: PRINT MP: PRINT NK: PRINT IQ
2720  PRINT D$;"CLOSE DIMEN": PRINT D$;"DELETE GLOSTIFF"
2730  HOME : PRINT "     INPUT PROGRAM DISK INTO DRIVE": VTAB 24
2740  INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
2750  PRINT D$;"RUN BACKSUB"
```

```
1000  REM   ************************
1010  REM   **  BACK SUBSTITUTION  **
1020  REM   ************************
1030  REM
1040  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1050  REM   :                                                :
1060  REM   :   THIS SUBPROGRAM PERFROMS BACKSUBSTITUTION    :
1070  REM   :   TO FIND DISPLACEMENT IN GLOBAL COORDINATE,   :
1080  REM   :   & FIND MEMBER FORCES IN LOCAL COORDINATES.   :
1090  REM   :                                                :
1100  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1110  REM
1120  REM
1130  HOME : PRINT "   INPUT DATA DISK NO.2 INTO DRIVE": VTAB 24
1140  INPUT "   PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1150  VTAB 12: HTAB 13: FLASH : PRINT "RUNNING BACKSUB": NORMAL
1160  IF PCASE = 2 THEN  PRINT D$;"PR#1"
1170  LET D$ =  CHR$ (4): PRINT D$;"OPEN DIMEN"
1180  PRINT D$;"READ DIMEN": INPUT PCASE,ND,NODE,MD,NW,NM,NJ,MF,MN,NI
1190  INPUT ME,MK,MP,NK,IQ: PRINT D$;"CLOSE DIMEN"
1200  PRINT D$;"OPEN FRONTIQ,L";MP:DD = MD * 20
1210  PRINT D$;"OPEN MQ,L";MK: PRINT D$;"OPEN DISPL"
1220  PRINT D$;"DELETE DISPL": PRINT D$;"OPEN DISPL,L";DD
1230  LET II = NM - 1: IF NI < NJ * ND THEN NI = NJ * ND
1240  DIM A(NI),D(MD),LE(MD),LF(MF),DF(MF),Q(MF),IB(NJ)
1250  FOR LZ = 1 TO NM: PRINT D$;"READ FRONTIQ,R";II
1260  INPUT MQ,KQ,NQ,NR,KS: FOR I = 1 TO KS: INPUT LE(I)
1270  NEXT I: FOR I = 1 TO NQ: INPUT LF(I): NEXT I: PRINT D$
1280  IF MQ = NQ GOTO 1410
1290  REM
1300  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1310  REM   :   STORE FRONT DISPLACEMENT IN ARRAY Q( )       :
1320  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1330  REM
1340  FOR I = KQ TO NQ:L = LF(I):Q(I) = DF(L): NEXT I
1350  FOR I = KQ TO NQ:DF(I) = Q(I): NEXT I: GOTO 1390
1360  LET I1 = MX + 1: PRINT "DF  ";: FOR I = I1 TO NQ
1370  PRINT DF(I); SPC( 15 -  LEN ( STR$ (DF(I)))): NEXT I
1380  PRINT : PRINT : RETURN
1390  IF PCASE = 0 GOTO 1410
1400  LET MX = MQ: GOSUB 1360
1410  IF MQ = 0 GOTO 1450
1420  LET IQ = IQ - 1: PRINT D$;"READ MQ,R";IQ: INPUT NE
1430  FOR I = 1 TO NE: INPUT A(I): NEXT I
1440  FOR I = 1 TO MQ: INPUT DF(I): NEXT I: PRINT D$
1450  IF PCASE = 0 GOTO 1510
1460  PRINT "MEMBER NO. ";II + 1: PRINT "=============": PRINT
1470  PRINT "LE    ";: FOR I = 1 TO KS: PRINT LE(I);
1480  PRINT  SPC( 8 -  LEN ( STR$ (LE(I))));: NEXT I: PRINT
1490  PRINT "LF    ";: FOR I = 1 TO NQ: PRINT LF(I);
1500  PRINT  SPC( 8 -  LEN ( STR$ (LF(I))));: NEXT I: PRINT
```

```
1510  IF MQ = 0 GOTO 1630
1520  IF PCASE = 0 GOTO 1630
1530  PRINT :Q = 0: PRINT " A ": PRINT "==="
1540  FOR K = 1 TO MQ:I1 = K - 1
1550  LET I2 = I1 * NQ - (I1 * (I1 - 1)) / 2 + 1
1560  LET I3 = I2 + NQ - K
1570  IF I1 = 0 GOTO 1590
1580  FOR I = 1 TO I1: PRINT  SPC( 9);Q;: NEXT I
1590  FOR I = I2 TO I3:A =  INT (A(I) * 100 + 0.5) / 100
1600  PRINT  SPC( 10 -  LEN ( STR$ (A)));A;: NEXT I
1610  LET A1 =  INT (DF(K) * 100 + 0.5) / 100
1620  PRINT  SPC( 10 -  LEN ( STR$ (A1)));A1: NEXT K: PRINT
1630  IF PCASE = 0 GOTO 1650
1640  LET MX = 0: GOSUB 1360
1650  IF MQ = 0 GOTO 1830
1660  REM
1670  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1680  REM   :   BACKSUBSTITUTION TO FIND DISPLACEMENTS       :
1690  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1700  REM
1710  DEF  FN NO(LL) = NQ * LL - (LL * (LL - 1)) / 2 + 1
1720  IF MQ = NQ GOTO 1750
1730  FOR I = 1 TO MQ:K =  FN NO(I - 1) - I: FOR J = KQ TO NQ
1740  LET DF(I) = DF(I) - A(J + K) * DF(J): NEXT J: NEXT I
1750  LET M = MQ + 1:I =  FN NO(MQ - 1): IF A(I) = 0 GOTO 1770
1760  LET DF(MQ) = DF(MQ) / A(I)
1770  IF MQ = 1 GOTO 1830
1780  FOR L = 2 TO MQ:I = M - L:J1 = I + 1:K =  FN NO(I - 1) - I
1790  FOR J = J1 TO MQ:DF(I) = DF(I) - A(J + K) * DF(J)
1800  NEXT J:K =  FN NO(I - 1): IF A(K) = 0 GOTO 1820
1810  LET DF(I) = DF(I) / A(K)
1820  NEXT L
1830  IF PCASE = 0 GOTO 1850
1840  LET MX = 0: GOSUB 1360
1850  FOR I = 1 TO KS:L = LE(I):D(I) = DF(L): NEXT I
1860  IF PCASE = 0 GOTO 1990
1870  PRINT "==============": PRINT "MEMBER NO. ";II + 1
1880  PRINT "==============": PRINT
1890  PRINT "DISPLACEMENT  ";: FOR I = 1 TO KS
1900  PRINT  INT (D(I) * 1000 + 0.5) / 1000;
1910  PRINT  SPC( 10 -  LEN ( STR$ ( INT (D(I) * 1000 + 0.5) / 1000)));
1920  NEXT I: PRINT
1930  REM
1940  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1950  REM   :   WRITE DISPLACES (ARRAY D) ON DISKETTE IN     :
1960  REM   :   DATA FILE 'DISPL'                            :
1970  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1980  REM
1990  PRINT D$;"WRITE DISPL,R";II: PRINT KS
2000  FOR I = 1 TO KS: PRINT D(I): NEXT I: PRINT D$
```

```
2010  LET II = II - 1: NEXT LZ
2020  PRINT D$;"CLOSE FRONTIQ": PRINT D$;"CLOSE MQ"
2030  PRINT D$;"CLOSE DISPL"
2040  PRINT D$;"DELETE FRONTIQ": PRINT D$;"DELETE MQ"
2050  PRINT D$;"OPEN LOSTIFF":IM = 0
2060  PRINT D$;"OPEN DISPL,L";DD: PRINT D$;"OPEN FORCE"
2070  FOR LZ = 1 TO NM: PRINT D$;"READ LOSTIFF"
2080  INPUT ST,CT,JQ,KQ: FOR I = 1 TO NW: INPUT A(I): NEXT I
2090  PRINT D$:X1 = CT:X2 = ST:X3 = 0:Y1 =  - ST:Y2 = CT
2100  LET Y3 = 0:Z1 = 0:Z2 = 0:Z3 = 1
2110  PRINT D$;"READ DISPL,R";IM: INPUT KS
2120  FOR I = 1 TO KS: INPUT D(I): NEXT I: PRINT D$
2130  REM
2140  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2150  REM   :   TRANSFORM DISPLACEMENTS FROM GLOBAL          :
2160  REM   :   COORDINATES INTO LOCAL COORDINATES           :
2170  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2180  REM
2190  FOR I = 1 TO KS STEP ND:R1 = D(I):R2 = D(I + 1)
2200  LET R3 = D(I + 2)
2210  LET D(I) = R1 * X1 + R2 * X2 + R3 * X3
2220  LET D(I + 1) = R1 * Y1 + R2 * Y2 + R3 * Y3
2230  LET D(I + 2) = R1 * Z1 + R2 * Z2 + R3 * Z3
2240  NEXT I: GOTO 2270
2250  FOR I = 1 TO KS:D =  INT (D(I) * 1000 + 0.5) / 1000
2260  PRINT  SPC( 10 -  LEN ( STR$ (D)));D;: NEXT I: PRINT : RETURN
2270  IF PCASE = 0 GOTO 2350
2280  PRINT "MEMBER NO. ";LZ: PRINT "=============": PRINT
2290  PRINT "LOCAL DISPL     ";: GOSUB 2250
2300  REM
2310  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2320  REM   :   CALCULATE MEMBER FORCES IN LOCAL COORDINATES :
2330  REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
2340  REM
2350  LET LF(1) = 1:L = KS + 2: FOR I = 2 TO KS
2360  LET LF(I) = LF(I - 1) + L - I: NEXT I
2370  FOR I = 1 TO KS:LE(I) = 0:I0 = LF(I) - I
2380  FOR J = 1 TO KS:J0 = LF(J) - J
2390  IF LF(J) =  > LF(I) GOTO 2410
2400  LET LE(I) = LE(I) + A(J0 + I) * D(J): GOTO 2420
2410  LET LE(I) = LE(I) + A(I0 + J) * D(J)
2420  NEXT J:LE(I) = LE(I) - A(NK + I): NEXT I
2430  IF PCASE = 0 GOTO 2470
2440  PRINT "MEMBER FORCE    ";
2450  FOR I = 1 TO KS:D =  INT (LE(I) * 1000 + 0.5) / 1000
2460  PRINT  SPC( 10 -  LEN ( STR$ (D)));D;: NEXT I: PRINT
2470  PRINT D$;"WRITE FORCE": PRINT KS: PRINT JQ: PRINT KQ
2480  FOR I = 1 TO KS: PRINT LE(I): NEXT I: PRINT D$
2490  LET IM = IM + 1: NEXT LZ: PRINT D$;"CLOSE LOSTIFF"
2500  PRINT D$;"CLOSE DISPL": PRINT D$;"CLOSE FORCE"
```

```
2510  PRINT D$;"DELETE LOSTIFF":IM = 0
2520  PRINT D$;"OPEN FORCE": PRINT D$;"OPEN DISPL,L";DD
2530  REM
2540  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2550  REM  :   PRINT NODAL DISPLACEMENTS & MEMBER FORCES    :
2560  REM  ::::::::::::::::::::::::::::::::::::::::::::::::::
2570  REM
2580  LET A$ = "**************************************************"
2590  FOR I = 1 TO NJ:IB(I) = 0: NEXT I
2600  FOR LZ = 1 TO NM: PRINT D$;"READ DISPL,R";IM
2610  INPUT KS: FOR I = 1 TO KS: INPUT D(I): NEXT I: PRINT D$
2620  PRINT D$;"READ FORCE": INPUT KS,JQ,KQ
2630  FOR I = 1 TO KS: INPUT LE(I): NEXT I: PRINT D$
2640  LET L = JQ:IB(L) = IB(L) + 1:M = 0
2650  IF IB(L) = 1 THEN  GOSUB 2690
2660  LET L = KQ:IB(L) = IB(L) + 1:M = ND
2670  IF IB(L) = 1 THEN  GOSUB 2690
2680  GOTO 2710
2690  FOR I = 1 TO ND:I1 = ND * (L - 1) + I:I2 = M + I
2700  LET A(I1) = D(I2): NEXT I: RETURN
2710  LET IM = IM + 1: NEXT LZ: PRINT D$;"CLOSE DISPL"
2720  IF PCASE < 2 THEN  PRINT D$;"PR#1"
2730  PRINT D$;"CLOSE FORCE": PRINT : PRINT : PRINT A$
2740  PRINT "***  NODAL DISPLACEMENTS IN GLOBAL COORDINATES ***"
2750  PRINT A$: PRINT
2760  PRINT "NODE          X-DISPL.        Y-DISDL.       ROTATION"
2770  PRINT "----          --------        --------       --------"
2780  FOR I = 1 TO NJ: PRINT  SPC( 4 -  LEN ( STR$ (I)));I;
2790  FOR J = 1 TO 3:J1 = 3 * (I - 1) + J
2800  LET A =  INT (A(J1) * 1000 + 0.5) / 1000
2810  PRINT  SPC( 15 -  LEN ( STR$ (A)));A;: NEXT J
2820  PRINT : NEXT I: PRINT : PRINT
2830  PRINT D$;"OPEN FORCE"
2840  PRINT : PRINT : PRINT A$
2850  PRINT "*****  MEMBER FORCES IN LOCAL COORDINATES  *****"
2860  PRINT A$: PRINT
2870  PRINT "MEMBER    NODE       AXIAL       SHEAR       MOMENT"
2880  PRINT "------    ----       -----       -----       ------"
2890  FOR LZ = 1 TO NM: PRINT D$;"READ FORCE"
2900  INPUT KS,JQ,KQ: FOR I = 1 TO KS: INPUT LE(I)
2910  NEXT I: PRINT D$: PRINT
2920  LET J = JQ: PRINT  SPC( 5 -  LEN ( STR$ (LZ)));LZ;
2930  PRINT  SPC( 9 -  LEN ( STR$ (J)));J;: GOTO 3000
2940  LET A1 =  INT (LE(J1 + 1) * 1000 + 0.5) / 1000
2950  LET A2 =  INT (LE(J1 + 2) * 1000 + 0.5) / 1000
2960  LET A3 =  INT (LE(J1 + 3) * 1000 + 0.5) / 1000
2970  PRINT  SPC( 12 -  LEN ( STR$ (A1)));A1;
2980  PRINT  SPC( 12 -  LEN ( STR$ (A2)));A2;
2990  PRINT  SPC( 12 -  LEN ( STR$ (A3)));A3: RETURN
3000  LET J1 = 0: GOSUB 2940
```

```
3010  LET K = KQ: PRINT  SPC( 14 -  LEN ( STR$ (K)));K;
3020  LET J1 = 3: GOSUB 2940
3030  NEXT LZ: PRINT D$;"CLOSE FORCE"
3040  PRINT D$;"DELETE FORCE": PRINT D$;"DELETE DISPL"
3050  PRINT D$;"PR#0"
3060  HOME : PRINT "     INPUT PROGRAM DISK INTO DRIVE": VTAB 24
3070  INPUT "     PRESS 'RETURN' KEY TO CONTINUE";B$: HOME
3080  PRINT D$;"RUN OPTION"
3090  END
```

```
1000 REM     ***************
1010 REM     **  ERROR    **
1020 REM     ***************
1030 REM
1040 REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1050 REM   :   THIS SUBPROGRAM PERFORMS SOLUTION ERROR OF    :
1060 REM   :   STATIC EQUATION  :-                           :
1070 REM   :                KU  =  F                         :
1080 REM   :   USING EUCLIDEAN NORM & DEVELOP PROGRAM FOR    :
1090 REM   :   FRONTAL METHOD.                               :
1100 REM   ::::::::::::::::::::::::::::::::::::::::::::::::::
1110 REM
1120 HOME : PRINT "   INPUT DATA DISK NO.2 INTO DRIVE": VTAB 24
1130 INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1140 VTAB 10: HTAB 15: FLASH : PRINT "RUNNING ERROR": NORMAL
1150 LET D$ =  CHR$ (4)
1160 PRINT D$;"OPEN DIMEN": PRINT D$;"READ DIMEN"
1170 INPUT PCASE,ND,NODE,MD,NW,NM,NJ,MF,MN,NI,ME,MK,MP,NK,IQ
1180 PRINT D$;"CLOSE DIMEN":DD = MD * 20
1190 PRINT D$;"OPEN GLOSTIFF":NK = MD * (MD + 1) / 2
1200 PRINT D$;"OPEN DISPL,L";DD:IM = 0
1210 PRINT D$;"OPEN FRONTIQ,L";MP
1220 DIM A(MF),B(MF),S(NW),LE(MD),LF(MF),NO(MD),D(MD),BK(MF),BM(MF)
1230 LET NO(1) = 1:LL = MD + 2: FOR I = 2 TO MD
1240 LET NO(I) = NO(I - 1) + LL - I: NEXT I
1250 LET R1 = 0:R2 = 0
1260 FOR LZ = 1 TO NM
1270 VTAB 12: HTAB 20: PRINT LZ
1280 PRINT D$;"READ FRONTIQ,R";IM
1290 INPUT MQ,KQ,NQ,NR,KS: FOR I = 1 TO KS: INPUT LE(I)
1300 NEXT I: FOR I = 1 TO NQ: INPUT LF(I): NEXT I: PRINT D$
1310 PRINT D$;"READ GLOSTIFF": FOR I = 1 TO NK
1320 INPUT S(I): NEXT I: FOR I = 1 TO KS: INPUT B(I): NEXT I: PRINT D$
1330 PRINT D$;"READ DISPL,R";IM: INPUT KS
1340 FOR I = 1 TO KS: INPUT D(I): NEXT I: PRINT D$
1350 FOR I = 1 TO MD:IO = NO(I) - I:SUM = 0
1360 IF I = 1 GOTO 1390
1370 FOR J = 1 TO I - 1:JO = NO(J) - J
1380 LET SUM = SUM + S(JO + I) * D(J): NEXT J
1390 FOR J = I TO MD:SUM = SUM + S(IO + J) * D(J): NEXT J
1400 LET A(I) = B(I) - SUM: NEXT I
1410 FOR I = 1 TO KS:L = LE(I):BM(L) = BM(L) + A(I)
1420 LET BK(L) = BK(L) + B(I): IF L > MQ GOTO 1440
1430 LET R1 = R1 + BM(L) ^ 2:R2 = R2 + BK(L) ^ 2
1440 NEXT I
1450 IF MQ = NQ GOTO 1500
1460 FOR I = 1 TO NR:B(I) = 0:A(I) = 0: NEXT I
1470 FOR I = KQ TO NQ:L = LF(I):B(L) = BK(I)
1480 LET A(L) = BM(I): NEXT I
1490 FOR I = 1 TO NR:BK(I) = B(I):BM(I) = A(I): NEXT I
1500 LET IM = IM + 1: NEXT LZ
```

```
1510  LET F1 =  SQR (R1):F2 =  SQR (R2):F3 = F1 / F2
1520  PRINT D$;"PR#1"
1530  PRINT "SOLUTION ERROR.....= ";F3: PRINT D$;"PR#0"
1540  PRINT D$;"CLOSE GLOSTIFF": PRINT D$;"CLOSE FRONTIQ"
1550  PRINT D$;"CLOSE DISPL"
1560  HOME : PRINT "     INPUT PROGRAM DISK INTO DRIVE": VTAB 24
1570  INPUT "     PRESS 'RETURN KEY' TO CONTINUE";B$: HOME
1580  PRINT D$;"RUN OPTION"
1590  END
```

ภาคผนวก ค.

วิธีการป้อนข้อมูลและแสดงตัวอย่างข้อมูล

ค-1 ความนำ

โปรแกรมที่ใช้สำหรับวิเคราะห์โครงสร้างชนิดโครงระนาบนี้ เขียนขึ้นเพื่อใช้กับเครื่องไมโครคอมพิวเตอร์ Apple II โดยเขียนโปรแกรมเป็นภาษา แอ๊ปเปิ้ลซอฟท์เบสิค ถ้าใช้กับเครื่องไมโครคอมพิวเตอร์ชนิดอื่น คำสั่งบางคำสั่งที่ใช้ในโปรแกรมอาจจะต้องเปลี่ยนแปลงให้เหมาะสมกับเครื่อง

ในโปรแกรมจะมีโปรแกรม OPTION ให้เลือกว่าต้องการให้โปรแกรมอะไรทำงาน โปรแกรมย่อยต่าง ๆ จะมีดังนี้

1. INPUT DATA
2. PLOT
3. PRINT DATA
4. ANALYZE
5. ERROR
6. EXIT

ถ้าต้องการจะหยุดการวิเคราะห์และการทำงานอื่น ๆ จะเลือกหัวข้อที่ 6

ค-2 การป้อนข้อมูล

โปรแกรมย่อย INPUT DATA เป็นโปรแกรมสำหรับป้อนข้อมูลต่าง ๆ ของชิ้นส่วนย่อยและโครงสร้าง ลำดับการป้อนข้อมูลมีดังนี้

1. ตอบคำถามว่าค่าโมดูลัสของชิ้นส่วนย่อยแต่ละชิ้นส่วนมีค่าเท่ากัน หรือไม่ โดยตอบว่า Y(es) หรือ N(o)

2. ป้อนข้อมูลรายละเอียดของโครงสร้าง

   2.1 จำนวนของชิ้นส่วนย่อย (NM)

   2.2 จำนวนของข้อ (NJ)

2.3 จำนวนชิ้นส่วนย่อยที่มีแรงภายนอกกระทำ (ML)

2.4 จำนวนของข้อที่มีแรงภายนอกกระทำ (JL)

2.5 จำนวนของข้อที่ต้องกำหนดสภาพเงื่อนไข (BCJ)

2.6 จำนวนของข้อที่มีสภาพการยึดแบบหมุดยึดหมุน (HJ)

2.7 ถ้าคำตอบในขั้นตอนที่ 1 คือ Y(es) จะทำการป้อนค่าโมดูลัส (E)

2.8 ตอบคำถามว่ายอมรับข้อมูลชุดนี้หรือไม่โดยตอบ Y(es) หรือ N(o)

3. ป้อนข้อมูลเกี่ยวกับคอออร์ดิเนทของข้อแต่ละข้อ

3.1 X-คอออร์ดิเนท x(I)

3.2 Y-คอออร์ดิเนท Y(I)

3.3 ตอบคำถามว่ายอมรับข้อมูลชุดนี้หรือไม่โดยตอบ Y(es) หรือ N(o)

จำนวนชุดข้อมูลที่ป้อนขึ้นอยู่กับจำนวนข้อของโครงสร้างดังข้อมูลที่ป้อนในขั้นตอนที่ 2.2

4. ป้อนข้อมูลเกี่ยวกับข้อและคุณสมบัติของชิ้นส่วนย่อย

4.1 เลขที่ข้อที่ปลายซ้ายของชิ้นส่วนย่อย J(I)

4.2 เลขที่ข้อที่ปลายขวาของชิ้นส่วนย่อย K(I)

4.3 พื้นที่หน้าตัดของชิ้นส่วนย่อย AR(I)

4.4 โมเมนต์อินเนอเชีย IZ(I)

4.5 ถ้าคำตอบในขั้นตอนที่ 1 คือ N(o) ต้องป้อนข้อมูลของค่าโมดูลัส E(I)

4.6 ตอบคำถามว่ายอมรับข้อมูลชุดนี้หรือไม่โดยตอบ Y(es) หรือ N(o)

จำนวนชุดข้อมูลที่ป้อนขึ้นอยู่กับจำนวนชิ้นส่วนย่อยของโครงสร้าง เท่ากับข้อมูลที่ป้อนในขั้นตอนที่ 2.1

5. ป้อนข้อมูลเกี่ยวกับแรงภายนอกที่กระทำบนชิ้นส่วนย่อย

5.1 เลขที่ชิ้นส่วนย่อย K1(I)

5.2 ค่าของน้ำหนักบรรทุกคงที่ในทิศทางแกน Y W(I)

5.3 ค่าของน้ำหนักบรรทุกคงที่ในทิศทางแกน X W(I + 1)

5.4 จำนวนของแรงตามแนวแกน F(I)

5.5 จำนวนแรงกระทำเป็นจุด (Point Load) ในแนวแกน Y P(I)

5.6 จำนวนแรงกระทำเป็นจุดในแนวแกน X P(I + 1)

5.7 ถ้าค่าในขั้นตอนที่ 5.4 เป็น 0 จะข้ามไปทำในขั้นตอนที่ 5.8

- ค่าของแรงตามแนวแกน F1(I,J)
- ระยะจากปลาย j ถึงจุดที่แรงกระทำ FA(I,J)

5.8 ถ้าค่าในขั้นตอนที่ 5.5 เป็น0จะข้ามไปทำในขั้นตอนที่ 5.9

- ค่าของแรงกระทำเป็นจุดในแนวแกน Y V1(I,J)
- ระยะปลาย j ถึงจุดที่แรงกระทำ VA(I,J)

5.9 ถ้าค่าในขั้นตอนที่ 5.6 เป็น 0 จะข้ามไปทำในขั้นตอนที่ 5.10

- ค่าของแรงกระทำเป็นจุดในแนวแกน X H1(I,J)
- ระยะจากปลาย j ถึงจุดที่แรงกระทำ HA(I,J)

5.10 ตอบคำถามว่ายอมรับข้อมูลชุดนี้หรือไม่โดยตอบ Y(es) หรือ N(o)

จำนวนชุดข้อมูลขึ้นอยู่กับจำนวนขั้นส่วนย่อยที่มีแรงภายนอกกระทำ ตามข้อมูลที่ป้อนในขั้นตอนที่ 2.3

6. ป้อนข้อมูลของแรงภายนอกที่กระทำที่ข้อในพิกัดของโครงสร้าง

6.1 เลขที่ของข้อ K2(I)

6.2 แรงในทิศทางแกน X PJ(I,1)

6.3 แรงในทิศทางแกน Y PJ(I,2)

6.4 โมเมนต์รอบแกน Z PJ(I,3)

6.5 ตอบคำถามยอมรับข้อมูลชุดนี้หรือไม่โดยตอบ Y(es) หรือ N(o)

จำนวนชุดข้อมูลขึ้นอยู่กับจำนวนข้อที่มีแรงกระทำตามข้อมูลที่ป้อนในขั้นตอนที่ 2.4

7. ป้อนข้อมูลเกี่ยวกับสภาพเงื่อนไขที่ข้อ ถ้าดีกรีของความอิสระในทิศทางใดเป็นอิสระ (Free) จะป้อนสภาพเงื่อนไข (Condition) เป็น 0 ถ้าดีกรีของความอิสระในทิศทางใดถูกบังคับหรือถูกกำหนด (Constrain) จะป้อนสภาพเงื่อนไขเป็น 1

7.1 เลขที่ข้อที่กำหนดสภาพเงื่อนไข BJ(I)

7.2 สภาพเงื่อนไขในทิศทาง X BJ(I + 1)

7.3 สภาพเงื่อนไขในทิศทาง Y BJ(I + 2)

7.4 สภาพเงื่อนไขในการหมุนรอบแกน Z BJ(I + 3)

7.5 ป้อนค่าการเคลื่อนที่ในทิศทาง X ถ้าสภาพเงื่อนไขในข้อ 7.2 เท่ากับ 1 BJ(I + 4)

7.6 ป้อนค่าการเคลื่อนที่ในทิศทาง Y ถ้าสภาพเงื่อนไขในข้อ 7.3 เท่ากับ 1 BJ(I + 5)

7.7 ป้อนค่าการหมุนของแกน z ถ้าสภาพเงื่อนไขในข้อ 7.4 เท่ากับ 1 BJ(I + 6)

7.8 ตอบคำถามยอมรับข้อมูลนี้หรือไม่โดยตอบ Y(es) หรือ N(o)

จำนวนชุดข้อมูลขึ้นอยู่กับจำนวนชิ้นที่กำหนดสภาพเงื่อนไขตามข้อมูลที่ป้อนในขั้นตอนที่ 2.5

8. ป้อนเลขที่ของชิ้นที่มีการยึดแบบหมุดยึดหมุน แต่ถ้าโครงสร้างเป็นแบบโครงข้อหมุนก็จะข้ามขั้นตอนนี้ไป

ค-3 <u>ตัวอย่างการป้อนข้อมูล</u>

แสดงตัวอย่างการป้อนข้อมูลในตัวอย่างที่ 1 (หัวข้อ 5.2)

```
**********
INPUT DATA
**********

IS YOUNG'S MODULUS CONSTANT ? (Y/N) Y

NO. OF MEMBERS.....= 15

NO. OF NODES.....= 12

NO. OF LOADED MEMBERS...= 6

NO. OF LOADED NODES.....= 0

NO. OF BOUNDARY NODES...= 3

NO. OF NODES WITH HINGES= 0

YOUNG'S MODULUS.....= 3200

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

INPUT NODAL POINT COORDINATES

```
NODE  NO. 1
==========
X-COORDINATE.....= 0
Y-COORDINATE.....= 0

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 2
==========
X-COORDINATE.....= 384
Y-COORDINATE.....= 0

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 3
==========
X-COORDINATE.....= 672
Y-COORDINATE.....= 0

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 4
==========
X-COORDINATE.....= 0
Y-COORDINATE.....= 156

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

```
NODE  NO. 5
==============
X-COORDINATE.........= 384
Y-COORDINATE.........= 156

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 6
==============
X-COORDINATE.........= 672
Y-COORDINATE.........= 156

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 7
==============
X-COORDINATE.........= 0
Y-COORDINATE.........= 312

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 8
==============
X-COORDINATE.........= 384
Y-COORDINATE.........= 312

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 9
==============
X-COORDINATE.........= 672
Y-COORDINATE.........= 312

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 10
==============
X-COORDINATE.........= 0
Y-COORDINATE.........= 468

DO YOU ACCEPT THESE DATA ? (Y/N)  Y


NODE  NO. 11
==============
X-COORDINATE.........= 384
Y-COORDINATE.........= 468

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

```
NODE  NO. 12
=============
X-COORDINATE.........= 672
Y-COORDINATE.........= 468

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

INPUT NODE NO. & PROP. OF MEMBERS

```
MEMBER NO. 1
==============
NODE J..............= 1
NODE K..............= 4
CROSS SECTIONAL AREA.= 256
INERTIA.............= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 2
==============
NODE J..............= 2
NODE K..............= 5
CROSS SECTIONAL AREA.= 256
INERTIA.............= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 3
==============
NODE J..............= 3
NODE K..............= 6
CROSS SECTIONAL AREA.= 256
INERTIA.............= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 4
==============
NODE J..............= 4
NODE K..............= 5
CROSS SECTIONAL AREA.= 720
INERTIA.............= 34600

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

```
MEMBER NO. 5
==============
NODE J.................= 5
NODE K.................= 6
CROSS SECTIONAL AREA.= 720
INERTIA................= 34600

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 6
==============
NODE J.................= 4
NODE K.................= 7
CROSS SECTIONAL AREA.= 256
INERTIA................= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 7
==============
NODE J.................= 5
NODE K.................= 8
CROSS SECTIONAL AREA.= 256
INERTIA................= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 8
==============
NODE J.................= 6
NODE K.................= 9
CROSS SECTIONAL AREA.= 256
INERTIA................= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 9
==============
NODE J.................= 7
NODE K.................= 8
CROSS SECTIONAL AREA.= 720
INERTIA................= 34600

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 10
==============
NODE J.................= 8
NODE K.................= 9
CROSS SECTIONAL AREA.= 720
INERTIA................= 34600

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

```
MEMBER NO. 11
==============
NODE J.............= 7
NODE K.............= 10
CROSS SECTIONAL AREA.= 256
INERTIA............= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 12
==============
NODE J.............= 8
NODE K.............= 11
CROSS SECTIONAL AREA.= 256
INERTIA............= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 13
==============
NODE J.............= 9
NODE K.............= 12
CROSS SECTIONAL AREA.= 256
INERTIA............= 5460

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 14
==============
NODE J.............= 10
NODE K.............= 11
CROSS SECTIONAL AREA.= 720
INERTIA............= 34600

DO YOU ACCEPT THESE DATA ? (Y/N)  Y

MEMBER NO. 15
==============
NODE J.............= 11
NODE K.............= 12
CROSS SECTIONAL AREA.= 720
INERTIA............= 34600

DO YOU ACCEPT THESE DATA ? (Y/N)  Y
```

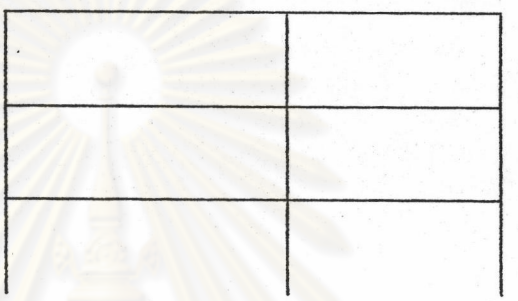
```
----------------------------------------
INPUT MEMBER LOADS IN GLOBAL COORDINATES
----------------------------------------

  MEMBER NO. 4
  ==============
  VERT. UNIF. LOAD.....= -0.05
  HORZ. UNIF. LOAD.....=  0.00
  NO. OF AXIAL LOAD....=  0
  NO. OF VERT. PT. LOAD=  2
  NO. OF HORZ. PT. LOAD=  0

  VERT. LOAD NO. 1
    VALUE OF LOAD......= -88.0
    DIST FROM LEFT.....=  168
  VERT. LOAD NO. 2
    VALUE OF LOAD......= -88.0
    DIST FROM LEFT.....=  336

  DO YOU ACCEPT THESE DATA ? (Y/N) Y


  MEMBER NO. 5
  ==============
  VERT. UNIF. LOAD.....= -0.05
  HORZ. UNIF. LOAD.....=  0
  NO. OF AXIAL LOAD....=  0
  NO. OF VERT. PT. LOAD=  1
  NO. OF HORZ. PT. LOAD=  0

  VERT. LOAD NO. 1
    VALUE OF LOAD......= -88.0
    DIST FROM LEFT.....=  168

  DO YOU ACCEPT THESE DATA ? (Y/N) Y


  MEMBER NO. 9
  ==============
  VERT. UNIF. LOAD.....= -0.05
  HORZ. UNIF. LOAD.....=  0.00
  NO. OF AXIAL LOAD....=  0
  NO. OF VERT. PT. LOAD=  2
  NO. OF HORZ. PT. LOAD=  0

  VERT. LOAD NO. 1
    VALUE OF LOAD......= -88.0
    DIST FROM LEFT.....=  168
  VERT. LOAD NO. 2
    VALUE OF LOAD......= -88.0
    DIST FROM LEFT.....=  336

  DO YOU ACCEPT THESE DATA ? (Y/N) Y
```

```
MEMBER NO. 10
==============
VERT. UNIF. LOAD.....= -0.05
HORZ. UNIF. LOAD.....=  0.00
NO. OF AXIAL LOAD....=  0
NO. OF VERT. PT. LOAD=  1
NO. OF HORZ. PT. LOAD=  0

VERT. LOAD NO. 1
  VALUE OF LOAD......= -88.0
  DIST FROM LEFT.....=  168

DO YOU ACCEPT THESE DATA ? (Y/N) Y


MEMBER NO. 14
==============
VERT. UNIF. LOAD.....= -0.05
HORZ. UNIF. LOAD.....=  0.00
NO. OF AXIAL LOAD....=  0
NO. OF VERT. PT. LOAD=  2
NO. OF HORZ. PT. LOAD=  0

VERT. LOAD NO. 1
  VALUE OF LOAD......= -71.5
  DIST FROM LEFT.....=  168
VERT. LOAD NO. 2
  VALUE OF LOAD......= -71.5
  DIST FROM LEFT.....=  336

DO YOU ACCEPT THESE DATA ? (Y/N) Y


MEMBER NO. 15
==============
VERT. UNIF. LOAD.....= -0.05
HORZ. UNIF. LOAD.....=  0.00
NO. OF AXIAL LOAD....=  0
NO. OF VERT. PT. LOAD=  1
NO. OF HORZ. PT. LOAD=  0

VERT. LOAD NO. 1
  VALUE OF LOAD......= -71.5
  DIST FROM LEFT.....=  168

DO YOU ACCEPT THESE DATA ? (Y/N) Y
```

```
-----------------------------------------
 INPUT BC. AT NODES (FREE=0,RESTR.=1)
-----------------------------------------

  NODE NO. 1
  ==============
  X-CONDITION..........= 1
  Y-CONDITION..........= 1
  ROTATE-CONDITION.....= 1

  X-DISPL..............= 0
  Y-DISPL..............= 0
  ROTATION.............= 0

  DO YOU ACCEPT THESE DATA ? (Y/N) Y

  NODE NO. 2
  ==============
  X-CONDITION..........= 1
  Y-CONDITION..........= 1
  ROTATE-CONDITION.....= 1

  X-DISPL..............= 0
  Y-DISPL..............= 0
  ROTATION.............= 0

  DO YOU ACCEPT THESE DATA ? (Y/N) Y

  NODE NO. 3
  ==============
  X-CONDITION..........= 1
  Y-CONDITION..........= 1
  ROTATE-CONDITION.....= 1

  X-DISPL..............= 0
  Y-DISPL..............= 0
  ROTATION.............= 0

  DO YOU ACCEPT THESE DATA ? (Y/N) Y
```

ค.4 <u>ตัวอย่างข้อมูล</u>

1. ตัวอย่างที่ 1



```
************
 INPUT DATA
************

NO. OF MEMBERS.........= 15
NO. OF NODES...........= 12
NO. OF LOADED MEMBERS...= 6
NO. OF LODED NODES.....= 0
NO. OF NODES WITH HINGES= 0
NO. OF BOUNDARY NODES...= 3
```

:NODAL PT. COOR.(HINGE=0,RIGID=1):

NODE	CASE	X-COOR.	Y-COOR.
1	1	0	0
2	1	384	0
3	1	672	0
4	1	0	156
5	1	384	156
6	1	672	156
7	1	0	312
8	1	384	312
9	1	672	312
10	1	0	468
11	1	384	468
12	1	672	468

:MEMBER NODES & PROPERTIES:

MEMBER	NODE-J	NODE-K	AREA	INERTIA	MODULUS
1	1	4	256	5460	3200
2	2	5	256	5460	3200
3	3	6	256	5460	3200
4	4	5	720	34600	3200
5	5	6	720	34600	3200
6	4	7	256	5460	3200
7	5	8	256	5460	3200
8	6	9	256	5460	3200
9	7	8	720	34600	3200
10	8	9	720	34600	3200
11	7	10	256	5460	3200
12	8	11	256	5460	3200
13	9	12	256	5460	3200
14	10	11	720	34600	3200
15	11	12	720	34600	3200

:MEMBER LOADS IN LOCAL COOR.:

MEM.	D-LOAD	W-LOAD	A-LOAD	DIST.	V-LOAD	DIST.	H-LOAD	DIST.
4	-.05	0	-	-	-88	168	-	-
			-	-	-88	336	-	-
5	-.05	0	-	-	-88	168	-	-
9	-.05	0	-	-	-88	168	-	-
			-	-	-88	336	-	-
10	-.05	0	-	-	-88	168	-	-
14	-.05	0	-	-	-71.5	168	-	-
			-	-	-71.5	336	-	-
15	-.05	0	-	-	-71.5	168	-	-

:BOUNDARY CONDITIONS AT NODES:

NODE	X	Y	Z	X-DISPL	Y-DISPL	ROTATION
1	1	1	1	0	0	0
2	1	1	1	0	0	0
3	1	1	1	0	0	0

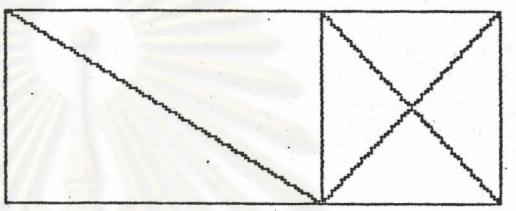
*** NODAL DISPLACEMENTS IN GLOBAL COORDINATES ***

NODE	X-DISPL.	Y-DISDL.	ROTATION
1	0	0	0
2	0	0	0
3	0	0	0
4	9E-03	-.033	-3E-03
5	.011	-.104	1E-03
6	.011	-.024	1E-03
7	.023	-.054	-3E-03
8	.023	-.17	1E-03
9	.023	-.04	1E-03
10	.044	-.063	-4E-03
11	.04	-.201	2E-03
12	.038	-.047	1E-03

****
***** MEMBER FORCES IN LOCAL COORDINATES *****
****

MEMBER	NODE	AXIAL	SHEAR	MOMENT
1	1	170.917	-12.985	-662.707
	4	-170.917	12.985	-1362.947
2	2	543.946	6.987	378.614
	5	-543.946	-6.987	711.31
3	3	128.437	5.998	328.244
	6	-128.437	-5.998	607.486
4	4	-11.682	59.863	3325.274
	5	11.682	135.337	-7256.334
5	5	-5.724	57.781	5521.022
	6	5.724	44.619	-1513.656
6	4	111.055	-24.667	-1962.328
	7	-111.055	24.667	-1885.669
7	5	350.828	12.944	1024.003
	8	-350.828	-12.944	995.327
8	6	83.818	11.722	906.17
	9	-83.818	-11.722	922.497
9	7	-1.737	61.899	3851.862
	8	1.737	133.301	-7000.919
10	8	-1.125	54.512	4972.963
	9	1.125	47.888	-1907.245
11	7	49.155	-26.404	-1966.193
	10	-49.155	26.404	-2152.797
12	8	163.016	13.557	1032.629
	11	-163.016	-13.557	1082.249
13	9	35.929	12.847	984.749
	12	-35.929	-12.847	1019.364
14	10	26.404	49.155	2152.797
	11	-26.404	113.045	-5839.597
15	11	12.847	49.971	4757.348
	12	-12.847	35.929	-1019.364

2. ตัวอย่างที่ 5.2



****
INPUT DATA
****

NO. OF MEMBERS.....= 10

NO. OF NODES.....= 6

NO. OF LOADED MEMBERS...= 0

NO. OF LODED NODES.....= 2

NO. OF NODES WITH HINGES= 6

NO. OF BOUNDARY NODES...= 3

:NODAL PT. COOR.(HINGE=0,RIGID=1):

NODE	CASE	X-COOR.	Y-COOR.
1	0	0	0
2	0	0	12
3	0	16	0
4	0	16	12
5	0	25	0
6	0	25	12

:MEMBER NODES & PROPERTIES:

MEMBER	NODE-J	NODE-K	AREA	INERTIA	MODULUS
1	1	2	4	1	1
2	1	3	2	1	1
3	2	3	5	1	1
4	2	4	4	1	1
5	3	4	6	1	1
6	3	5	3	1	1
7	3	6	5	1	1
8	4	5	3	1	1
9	4	6	3	1	1
10	5	6	3	1	1

:NODAL LOADS IN GLOBAL COOR.:

NODE	X-FORCE	Y-FORCE	MOMENT
2	0	-10	0
6	20	0	0

:BOUNDARY CONDITIONS AT NODES:

NODE	X	Y	Z	X-DISPL	Y-DISPL	ROTATION
1	0	1	0	-	0	-
2	1	0	0	0	-	-
3	1	1	0	0	0	-

****
*** NODAL DISPLACEMENTS IN GLOBAL COORDINATES ***
****

NODE	X-DISPL.	Y-DISDL.	ROTATION
1	0	0	0
2	0	-23.622	0
3	0	0	0
4	80	-4.114	0
5	-4.629	-83.657	0
6	135.371	-91.886	0

****
***** MEMBER FORCES IN LOCAL COORDINATES *****
****

MEMBER	NODE	AXIAL	SHEAR	MOMENT
1	1	7.874	0	0
	2	-7.874	0	0
2	1	0	0	0
	3	0	0	0
3	2	3.543	0	0
	3	-3.543	0	0
4	2	-20	0	0
	4	20	0	0
5	3	2.057	0	0
	4	-2.057	0	0
6	3	1.543	0	0
	5	-1.543	0	0
7	3	-2.571	0	0
	6	2.571	0	0
8	4	-2.571	0	0
	5	2.571	0	0
9	4	-18.457	0	0
	6	18.457	0	0
10	5	2.057	0	0
	6	-2.057	0	0

ค-5 เปรียบเทียบค่าของแรงที่คำนวณได้

1. เปรียบเทียบค่าของแรงตามแนวแกน และแรงเฉือนที่คำนวณได้ของเล่าที่ส่มมาตรกัน ของตัวอย่างที่ 3 ซึ่งผลลัพธ์ที่ได้มีค่าเท่ากับผลจากเครื่องเมนเฟรม และจากเอกสารอ้างอิงที่ 2

ตารางที่ ค-1 ตารางเปรียบเทียบค่าแรงตามแนวแกนและแรงเฉือน(ตัวอย่างที่ 3)

ชั้นที่	ชั้นล่วนที่	แรงตามแนวแกน	แรงเฉือน	ชั้นล่วนที่	แรงตามแนวแกน	แรงเฉือน
1	2	3.892	8.346	3	- 3.895	8.299
2	9	3.802	7.089	10	- 3.798	7.118
3	16	3.700	6.278	17	- 3.697	6.282
4	23	3.539	6.140	24	- 3.536	6.139
5	30	3.283	5.355	31	- 3.280	5.355
6	37	3.029	4.881	38	- 3.026	4.881
7	44	2.773	4.223	45	- 2.770	4.223
8	51	2.479	3.904	52	- 2.476	3.904
9	58	2.162	3.359	59	- 2.159	3.359
10	65	1.865	2.876	66	- 1.862	2.875
11	72	1.605	2.248	73	- 1.602	2.248
12	79	1.256	2.054	80	- 1.252	2.053
13	86	0.761	1.425	87	- 0.758	1.425
14	93	0.404	0.869	94	- 0.400	0.868
15	100	0.156	0.289	101	- 0.154	0.303

2. เปรียบเทียบผลลัพธ์ในตัวอย่างที่ 4 กับเครื่องเมนเฟรม โดยคิดเปอร์เซ็นต์ความแตกต่างโดยใช้ผลจากเครื่องเมนเฟรมเป็นหลัก

ตารางที่ ค-2 เปรียบเทียบค่าแรงตามแนวแกน (ตัวอย่างที่ 4)

ปุ่มที่	ยันล่วนที่	ไมโครฯ	เมนเฟรม	% ผิดพลาด $\times 10^{-4}$	ยันล่วนที่	ไมโครฯ	เมนเฟรม	% ผิดพลาด $\times 10^{-4}$
1	2	161.349	161.347	1.24	3	161.344	161.347	-1.85
2	15	261.344	261.345	-3.83	16	261.345	261.345	0.00
3	28	1010.267	1010.265	1.97	29	1010.263	1010.265	-1.98
4	41	1901.743	1901.744	-0.05	42	1901.744	1901.744	0.00
5	54	2797.410	2797.411	-0.04	55	2797.410	2797.411	-0.04
6	67	3692.456	3692.457	-0.03	68	3692.456	3692.457	-0.03
7	80	4503.682	4503.683	-0.02	81	4503.682	4503.683	-0.02
8	93	4775.046	4775.048	-0.04	94	4775.046	4775.048	-0.04
9	106	2196.225	2196.225	0.00	107	2196.225	2196.225	0.00

ประวัติ

นาย ภาณุวัฒน์ คุรุรัตน์ เกิดเมื่อวันที่ 24 ธันวาคม พ.ศ. 2501 ที่จังหวัดลพบุรี สำเร็จการศึกษาชั้นมัธยมศึกษาตอนปลาย (มศ. 5) จากโรงเรียนส่วนกุหลาบวิทยาลัย ในเดือนมีนาคม พ.ศ. 2520 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโยธา จากมหาวิทยาลัยเชียงใหม่ ในเดือนมีนาคม พ.ศ. 2524 ทำงานครั้งแรกด้านการก่อสร้าง-อาคาร กับบริษัท เนาวรัตน์พัฒนาการ