รายการอ้างอิง

ภาษาไทย

ปราโมทย์ เดชะอำไพ,ไฟไนต์เอลิเมนต์ในงานวิศวกรรม. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2537

ภาษาอังกฤษ

Boley, B. A., and Weiner,J.H. Theory of Thermal Stresses. New York : John Wiley & Sons , 1962

Borse, G. J. Fortran 77 and Numerical Methods for Engineers. 2nd ed. Boston : PWS-KENT , 1991

Chapra, S. C., and Canale,R.P. Numerical Methods for Engineers. 2nd ed. Singapore : McGraw-Hill , 1988

Cook, R. D. Concepts and Applications of Finite Element Analysis. 2nd ed. Canada : John Wiley & Sons , 1981

Crouse, W. H., and Anglin,D.L. Automotive Engines. 8th ed. Singapore : Mc-Graw-Hill Book Co., 1995

Etter, D. M. Structured Fortran 7 for Engineers and Scientists. 3rd ed. Canada : Addison Wesley , 1990

Ferguson, C. R. Internal Combustion Engines, Applied Thermosciences. New York : John Wiley & Sons, 1986

Gatewood, B. E. Thermal Stress. New York : McGraw-Hill, 1957

Greene, A. B., and Lucas, G. G. The Testing of Internal Combustion

Engines. UK : London English Univ. Press, 1969

Holman, J. P. Heat Transfer. 7th ed. New York : McGraw-Hill , 1992

Huobner, K. H., and Thornton, E. A. The Finite Element Method for Engineers. 2nd ed. Canada : John Wiley & Sons , 1982

Incropera, F. P., and DeWitt, D. P. Fundamentals of Heat and Mass Transfer. 2nd ed. Singapore : John Wiley & Sons , 1985

Johns, D. J. Thermal Stress Analyses. UK : Pergamon Press, 1965

Kreyszig, E. Advanced Engineering Mathematics. 7th ed. Canada : John Wiley & Sons , 1993

Mondolfo, L. E. Aluminum Alloys : Structure and Properties. London : Butterworth, 1979

Rao, S. S. The Finite Element Method in Engineering. 2nd ed. Oxford : Pergamon Press , 1989

Singer, F. L., and Pytel, A. Strength of Materials. 3rd ed. New York : Harper & Row, 1980

Stasa, F. L. Applied Finite Element Analysis for Engineers. Japan : CBS, 1986

Timoshenko, S., and Goodier, J. N. Theory of Elasticity. 2nd ed. : McGraw-Hill , 1951

Van Horn, L. F. Aluminum. Ohio : American Society for Metals, 1968

Zienkiewicz, O. C., and Morgan,K. Finite Element and Approximation. Canada : John Wiley & Sons , 1983

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

รายละเอียดของโปรแกรม HEAT3D

รายละเอียดของโปรแกรม HEAT3D

โปรแกรม HEAT3D จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
C    PROGRAM HEAT3D.FOR
C                                                    UPDATE : 13/2/95
C    A FINITE ELEMENT THERMAL ANALYSIS PROGRAM FOR THREE-DIMENSIONAL
C    NONLINEAR TRANSIENT HEAT CONDUCTION WITH INTERNAL HEAT GENERATION,
C    APPLIED SURFACE HEATING  AND SURFACE CONVECTION .
C
C    THE VALUES DECLARED IN THE PARAMETER STATEMENT BELOW SHOULD
C    BE ASSIGNED ACCORDING TO THE SIZE OF THE PROBLEMS
C
C       MXPOI = MAXIMUM NUMBER OF NODES IN THE MODEL
C       MXELE = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C
      PARAMETER (MXPOI=500, MXELE=500)
      IMPLICIT double precision (A-H,O-Z)
      DIMENSION TEXT(20)
      REAL*8 COORD(MXPOI,3),TEMP(MXPOI), SYSK(MXPOI), SYSQ(MXPOI)
      CHARACTER*20  NAME1, NAME2, NAME3, NAME4, NAME5
      INTEGER*2 INTMAT(MXELE,4), IBC(MXPOI), LTYPE(MXELE,3) ,
     *          NFACE(MXELE,4), NGFACE(MXELE,4)
      INTEGER*4 NTS,NTS1,ITV,ITVR,ITS
C
   10 WRITE(6,20)
   20 FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
      READ(5, '(A)', ERR=10)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
   25 WRITE(6,*)
      WRITE(6,*)' DO YOU WANT TO CREAT NEW INPUT FILE FOR NEXT TIME?',
     *       ' ( YES:1 , NO:0 ) : '
      READ(5,*) INEXT
      IF(INEXT.EQ.0) GO TO 50
      IF(INEXT.NE.1) GO TO 25
   30 WRITE(6,40)
   40 FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME FOR NEXT TIME :')
      READ(5, '(A)', ERR=30)  NAME4
      OPEN(UNIT=10, FILE=NAME4, STATUS='NEW', ERR=30)
C
   50 WRITE(6,60)
   60 FORMAT(/, '  PLEASE ENTER FILE NAME FOR TEMPERATURE'
     *          ' SOLUTIONS:'                    )
      READ(5, '(A)', ERR=50)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=50)
C
C     READ OUTPUT-PLOT FILE FOR I-DEAS
C
   85 WRITE(6,*)
      WRITE(6,*)'CREAT THE PLOT FILE NAME ( YES:1 , NO:0 ) :'
      READ(5,*) IPT
      IF(IPT.EQ.0) GO TO 95
      IF(IPT.NE.1) GO TO 85
   87 WRITE(6,90)
   90 FORMAT(/, '  PLEASE INPUT THE PLOT FILE NAME :: ')
      READ(5, '(A)',ERR=87) NAME3
      OPEN(UNIT=9, FILE=NAME3,STATUS='NEW', ERR=87)
C
C
C     READ TITLE OF COMPUTATION:
C
   95 READ(7,*)  NLINES
      DO 100  ILINE=1,NLINES
      READ(7,1)  TEXT
    1 FORMAT(20A4)
  100 CONTINUE
C
C     READ INPUT DATA:
C
      READ(7,1)  TEXT
      READ(7,*)  NPOIN, NELEM
C
      NNODE=4
C
      READ(7,1)  TEXT
      READ(7,*)  DEN, SH, TK, H, Q, QS, TI
C
      WRITE(8,'(A)') '  **** MATERIAL PROPERTIES ****'
      WRITE(8,102) DEN,SH
      WRITE(8,103) H,Q
      WRITE(8,104) QS,TI
      WRITE(8,110) TK
  102 FORMAT(' Density       =',E10.3,'  ,        Specific Heat = '
     *       ,E10.3)
  103 FORMAT(' Convection Coefficient = ',E10.3,' , Internal Heat = '
     *       ,E10.3)
  104 FORMAT(' External Heat = ',E10.3,'  ,   Ambient Temperature = '
     *       ,E10.3)
  110 FORMAT(' Thermal Conductivity Coefficient = ',E10.3)
      READ(7,1)  TEXT
      READ(7,*)  ST,FT,TS
      IF(FT.GE.ST) GO TO 125
      WRITE(6,*)' *** ERROR : TIME DATA IS MISTAKE. ****'
      STOP
C
C     PREPARE DATA FILE FOR THERMAL STRESS ANALYSIS
C
  125 WRITE(6,126)
  126 FORMAT(/, ' DO YOU WANT TO PREPARE DATA FOR THERMAL STRESS',
     *       ' ANALYSIS ?')
  127 WRITE(6,128)
  128 FORMAT(10X,'NO: -2 , MAX HEAT FLUX: -1 , FINAL TIME STEP: 0 ::')
      READ(5,*) LOUT
      IF(LOUT.LT.-2) GO TO 127
      OUT = LOUT
      IF(OUT.GT.FT) GO TO 127
      IF (LOUT.GT.-2) THEN
  129   WRITE(6,130)
  130   FORMAT(/, '  PLEASE ENTER FILE NAME FOR THERMAL STRESS'
     *       ' ANALYSIS:'                    )
        READ(5, '(A)', ERR=129) NAME5
        OPEN(UNIT=11, FILE=NAME5, STATUS='NEW', ERR=129)
      END IF
C
      READ(7,1)  TEXT
      DO 138  IP=1,NPOIN
      READ(7,*)  I, IBC(I), (COORD(I,K), K=1,3), TEMP(I)
      IF(I.NE.IP)  WRITE(6,135)  IP
  135 FORMAT(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP)  STOP
  138 CONTINUE
C
C     CHECK NUMBER OF TIME STEP FOR TRANSIENT PROBLEM
C
      RTS=(FT-ST)/TS
      NTS=INT(RTS)
      NTS1=NTS
      FTS=FT-NTS*TS
      IF(FTS.GT.0.0D+00) NTS=NTS+1
      WRITE(6,*)
      WRITE(6,*)'Number of Time Steps =',NTS
  145 WRITE(6,'(A)')' INPUT RANGE TIME STEP FOR PRINTING OUTPUT DATA'
      READ(5,*, ERR = 145) ITV
      IF(ITV.EQ.0) ITV = NTS
      WRITE(8,*)
      WRITE(8,'(A)') '  *** Time History ***'
      WRITE(8,147) ST,FT
      WRITE(8,148) TS,ITV
  147 FORMAT(' Starting Time =',F10.4,' , Final Time =',F10.4)
  148 FORMAT(' Time Step =',F10.4,' , Range Time Step to print ='
     *       ,I8)
      WRITE(8,*)
C
C     Remarks: FTS is the final time step
C              NTS is the numbers of time step
C              ITV is the interval time for printing results
C
      LT1 = 0
      LT2 = 0
      LT3 = 0
      LT4 = 0
      READ(7,1)  TEXT
      DO 140  IE=1,NELEM
```

```
      READ(7,*)  I, (INTMAT(I,J), J=1,4), ((LTYPE(I,K), K=1,3),
     *             (NGFACE(I,L), L=1,4), (NFACE(I,M), M=1,4)
      IF(I.NE.IE)  WRITE(6,150)  IE
  150 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IE)  STOP
      IF(LTYPE(I,1).EQ.1)  LT1 = 1
      IF(LTYPE(I,2).EQ.1)  LT2 = 1
      IF(LTYPE(I,3).EQ.1)  LT3 = 1
  140 CONTINUE
      WRITE(6,160)
      WRITE(8,160)
  160 FORMAT(/,' THE F.E. MODEL INCLUDES THE FOLLOWING',
     *       ' HEAT TRANSFER MODE(S):',
     *       /,'  -  HEAT CONDUCTION              ')
      IF(LT1.EQ.1)  WRITE(6,170)
      IF(LT1.EQ.1)  WRITE(8,170)
  170 FORMAT( '  -  INTERNAL HEAT GENERATION      ')
      IF(LT2.EQ.1)  WRITE(6,180)
      IF(LT2.EQ.1)  WRITE(8,180)
  180 FORMAT( '  -  SPECIFIED SURFACE HEATING     ')
      IF(LT3.EQ.1)  WRITE(6,190)
      IF(LT3.EQ.1)  WRITE(8,190)
  190 FORMAT( '  -  SURFACE CONVECTION            ')
      WRITE(8,*)
C
      WRITE(6,435)  NPOIN, NELEM
      WRITE(8,435)  NPOIN, NELEM
      WRITE(8,*)
  435 FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I5,
     *       ' NODES AND', I5,' ELEMENTS ***')
C
C     ESTABLISH ALL ELEMENT MATRICES ASSOCIATED WITH THE SPECIFIED
C     HEAT TRANSFER MODES AND ASSEMBLE THEM FOR SYSTEM MATRICES IN
C     THE FORM NEEDED FOR MINIMUM MEMORY REQUIREMENT:
C
      WRITE(6,440)
  440 FORMAT(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *       ' ASSEMBLING ELEMENT EQUATIONS ***'     )
      WRITE(6,*)
      WRITE(6,'(A)')' *** SOLVING ***'
      ITVR = ITV
      JT = 0
C
C     START DO-LOOP OF TIME STEP
C
      DO 800 ITS=1,NTS
C
C     CHANGE SYSQ TO TEMP
C
      IF(ITS.GT.1) THEN
        DO 445 I=1,NPOIN
        TEMP(I) = SYSQ(I)
  445   CONTINUE
      END IF
      DO 450  I=1,NPOIN
      SYSQ(I) = 0.0D+00
      SYSK(I) = 0.0D+00
  450 CONTINUE
C
      CALL TETRA(MXELE, MXPOI, NELEM, INTMAT, COORD,  TS,  DEN,
     *           SH, H, Q, QS, TI, LTYPE, NFACE, NGFACE,  NNODE,
     *           TEMP, TK,                SYSK, SYSQ )
C
C     SOLVE THE SOLUTION OF THE TRANSIENT PROBLEM
C
      CALL SOLVETRA( MXPOI, NPOIN, IBC, TEMP, SYSK,      SYSQ,
     *               TMAX, TMIN, NMAX, NMIN )
      IF(ITS.EQ.NTS1) TS = FTS
C
C     PRINT OUT NODAL TEMPERATURE SOLUTIONS:
C
      IF(JT.EQ.1) NTS = ITS
      IF((ITS.EQ.ITVR).OR.((ITS.EQ.1).OR.((ITS.EQ.NTS)) THEN
        IF(ITS.EQ.1) WRITE(8,490)  NPOIN
  490   FORMAT(' NODAL TEMPERATURE SOLUTIONS (', I5,'):',
     *         //, 2X, 'NODE', 3X, 'TEMPEARTURE',      )
        WRITE(8,495) ITS
  495   FORMAT(/,25X,' TIME STEP is : ', I10)
        WRITE(6,496) ITS
  496   FORMAT(' Time step = ',I6)
        DO 500  IP=1,NPOIN
        WRITE(8,510)  IP, SYSQ(IP)
  510   FORMAT(I6, E14.6)
  500   CONTINUE
        WRITE(8,*)
        WRITE(8,'(A)') '  *** MINIMUM TEMPERATURE ***'
        WRITE(8,510) NMIN, TMIN
        WRITE(8,'(A)') '  *** MAXIMUM TEMPERATURE ***'
        WRITE(8,510) NMAX, TMAX
        IF(ITS.NE.1) ITVR = ITVR + ITV
        IF(ITVR.EQ.1) ITVR = ITVR + ITV
      END IF
C
C     PRINT DATA FILE FOR THERMAL STRESS ANALYSIS
C
      IF ((LOUT.EQ.IFLUX).OR.(LOUT.EQ.-2))GO TO 700
      IF (((ITS.EQ.NTS).AND.(LOUT.EQ.0)).OR.
     *    ((LOUT.GT.0).AND.(LOUT.EQ.ITS))) THEN
        REWIND(11)
        FTIME = ST + ITS*TS
        WRITE(11,550) ITS, FTIME, NPOIN
  550   FORMAT(2X,' AT TIME STEP =',I6,'(',F10.4,' SEC.)',
     *         '  AND ',I6,' NODES')
        DO 600 IU=1,NPOIN
        WRITE(11,*) IU, SYSQ(IU)
  600   CONTINUE
      END IF
C
  800 CONTINUE
C
C     PRINT INPUT FILE FOR NEXT TIME
C
      IF (INEXT.EQ.1)
     *CALL PRNEXT( DEN,  SH, H, Q, QS, TI, ST,  FT,  TS, MXPOI, NPOIN,
     *         MXELE, NELEM, IBC, COORD, SYSQ, INTMAT, LTYPE, NGFACE,
     *         NFACE, NAME1, TK)
C
C     PRINT OUTPUT FILE FOR I-DEAS
C
      IF((IPT.EQ.1).AND.(LOUT.NE.-1))
     *   CALL PRTEMP(MXELE, NPOIN, SYSQ)
C
      STOP
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE SOLVETRA( MXPOI, NPOIN, IBC, TEMP, SYSK,      SYSQ,
     *                     TMAX, TMIN, NMAX, NMIN )
C
C     SOLVE A SET OF SIMULTANEOUS EQUATIONS USING GAUSS ELIMINATION.
C     THIS SOLVER ROUTINE CAN BE DESCRIBED BY USING AN EXAMPLE OF A
C     SET OF FOUR SIMULTANEOUS EQUATIONS (AFTER APPLYING BOUNDARY
C     CONDITIONS) AS SHOWN BELOW:
C
C        [ A11   0    0    0 ]  [ X1 ]      [ F1 ]
C        [                   ]  [  ]       [  ]
C        [  0   A22   0    0 ]  [ X2 ]      [ F2 ]
C        [                   ]  [  ]   =   [  ]
C        [  0    0   A33   0 ]  [ X3 ]      [ F3 ]
C        [                   ]  [  ]       [  ]
C        [  0    0    0   A44]  [ X4 ]      [ F4 ]
C
C     WHERE THE VARIABLE XL IS THE LOAD VECTOR ON RHS OF THE EQUATIONS.
C     THE GLOBAL STIFFNESS MATRIX ABOVE IS STORED IN THE VARIABLE
C     GSTIF IN THE FORMAT SHOWN BELOW:  ( HERE  NROW = 4 )
C
C                     [ A11 ]
C                     [     ]
C                     [ A22 ]
C       [ SYSK ]   =  [     ]
C                     [ A33 ]
C                     [     ]
C                     [ A44 ]
C
C     AND THE OUTPUT SOLUTIONS WILL BE STORED IN THE VARIABLE SYSQ.
C
C
C     APPLY TEMPERATURE BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C          0 = FREE TO CHANGE (TO BE COMPUTED)
C          1 = FIXED AS SPECIFIED
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 SYSK(MXPOI), SYSQ(MXPOI), TEMP(MXPOI)
C
      INTEGER*2 IBC(MXPOI)
C
      TMAX = -1.0D+10
      TMIN =  1.0D+10
      NMAX = 0
      NMIN = 0
      DO 100  IEQ=1,NPOIN
      IF(IBC(IEQ).EQ.0)  GO TO 100
C
      IF(IBC(IEQ).EQ.1)  THEN
        SYSQ(IEQ) = TEMP(IEQ)
        SYSK(IEQ) = 1.0D+00
      ENDIF
C
  100 CONTINUE
C
      DO 200 I=1,NPOIN
      IF(IBC(I).NE.0) GO TO 200
      PIVOT1=SYSK(I)
      IF(ABS(PIVOT1).LT.10.D-10) THEN
      WRITE(6,500) I, PIVOT1
  500 FORMAT(' EQ. NO.', I5, ' HAS NEARLY ZERO PIVOT OF', E14.6,
     *       ' ** STOP **', //,
     *       ' *** CHECK NODE AND ELEMENT NUMBERING IN F.E. MODEL ***')
      STOP
      ENDIF
C
      SYSQ(I)=SYSQ(I)/PIVOT1
C
  200 CONTINUE
C
      DO 300 L=1,NPOIN
      IF(SYSQ(L).GT.TMAX) THEN
        TMAX = SYSQ(L)
        NMAX = L
      ENDIF
      IF(SYSQ(L).LT.TMIN) THEN
        TMIN = SYSQ(L)
        NMIN = L
      ENDIF
  300 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE TETRA(MXELE, MXPOI, NELEM, INTMAT, COORD,  TS,  DEN,
     *           SH, H, Q, QS, TI, LTYPE, NFACE, NGFACE,  NNODE, TEMP,
     *           TK,          SYSK, SYSQ )
C
C     ESTABLISH ELEMENT MATRICES ACCORDING TO THE SPECIFIED HEAT
C     TRANSFER MODES AND ASSEMBLE THEM FOR SYSTEM EQUATIONS
C
      IMPLICIT double precision (A-H,O-Z)
```

```
      REAL*8 COORD(MXPOI,3), SYSK(MXPOI), SYSQ(MXPOI),
     *       TEMP(MXPOI)
      REAL*8 QQ(4), AKE(4,4), QE(4), AKP(4), QP(4)
C
      INTEGER*2  INTMAT(MXELE,4), LTYPE(MXELE,3), NFACE(MXELE,4),
     *       NGFACE(MXELE,4)
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 5000  IE=1,NELEM
C
C     ZERO ALL COEFFICIENTS OF THE FINAL ELEMENT MATRICES:
C
      DO 50  I=1,4
      QE(I) = 0.0D+00
      DO 50  J=1,4
      AKE(I,J) = 0.0D+00
   50 CONTINUE
C
C     FIND THE NEW CONDUCTIVITY COEFFICIENT
C
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
      LL = INTMAT(IE,4)
      TAVG = (TEMP(II)+TEMP(JJ)+TEMP(KK)+TEMP(LL))/4.0D+00
C
C     ELEMENT CONDUCTION MATRIX:
C
      CALL CONDTE( INTMAT, MXPOI, COORD, MXELE, IE, TK, VOLUME,
     *       XG1, XG2, XG3, XG4, YG1, YG2, YG3, YG4, ZG1,
     *       ZG2, ZG3, ZG4,   AKE )
C
C     FIND AREA OF SURFACE
C
C
C       AS IS THE LENGTH OF I,J NODES
C       BS IS THE LENGTH OF I,K NODES
C       CS IS THE LENGTH OF J,K NODES
C       DS IS THE LENGTH OF K,L NODES
C       ES IS THE LENGTH OF I,L NODES
C       FS IS THE LENGTH OF J,L NODES
C
C       AREA1 IS THE AREA OF I,J,K NODES
C       AREA2 IS THE AREA OF I,K,J NODES
C       AREA3 IS THE AREA OF L,J,I NODES
C       AREA4 IS THE AREA OF L,I,K NODES
C
      AS = SQRT((XG2-XG1)*(XG2-XG1)+(YG2-YG1)*(YG2-YG1)
     1          +(ZG2-ZG1)*(ZG2-ZG1))
      BS = SQRT((XG3-XG1)*(XG3-XG1)+(YG3-YG1)*(YG3-YG1)
     1          +(ZG3-ZG1)*(ZG3-ZG1))
      CS = SQRT((XG3-XG2)*(XG3-XG2)+(YG3-YG2)*(YG3-YG2)
     1          +(ZG3-ZG2)*(ZG3-ZG2))
      DS = SQRT((XG4-XG3)*(XG4-XG3)+(YG4-YG3)*(YG4-YG3)
     1          +(ZG4-ZG3)*(ZG4-ZG3))
      ES = SQRT((XG4-XG1)*(XG4-XG1)+(YG4-YG1)*(YG4-YG1)
     1          +(ZG4-ZG1)*(ZG4-ZG1))
      FS = SQRT((XG4-XG2)*(XG4-XG2)+(YG4-YG2)*(YG4-YG2)
     1          +(ZG4-ZG2)*(ZG4-ZG2))
      S = 0.5D+00*(AS+BS+CS)
      AREA1 = SQRT(S*(S-AS)*(S-BS)*(S-CS))
      S = 0.5D+00*(CS+DS+FS)
      AREA2 = SQRT(S*(S-CS)*(S-DS)*(S-FS))
      S = 0.5D+00*(AS+ES+FS)
      AREA3 = SQRT(S*(S-AS)*(S-ES)*(S-FS))
      S = 0.5D+00*(BS+DS+ES)
      AREA4 = SQRT(S*(S-BS)*(S-DS)*(S-ES))
C
C     ELEMENT CONVECTION MATRICES:
C
      IF(LTYPE(IE,3).EQ.1) CALL CONVTE(IE, H, TI, MXELE, NFACE,
     *             AREA1, AREA2, AREA3, AREA4,    AKE, QE )
C
C     ELEMENT HEAT LOAD DUE TO INTERNAL HEAT GENERATION:
C
      IF((LTYPE(IE,1).NE.1).OR.(Q.EQ.0.0D+00)) GO TO 400
      FAC = Q*VOLUME/4.0D+00
      DO 310  I=1,4
      QQ(I) = FAC
  310 CONTINUE
      DO 320  I=1,4
      QE(I) = QE(I) + QQ(I)
  320 CONTINUE
  400 CONTINUE
C
C     ELEMENT HEAT LOAD DUE TO SPECIFIED SURFACE HEATING:
C
      IF(LTYPE(IE,2).EQ.1) CALL EXHTE( IE, MXELE, QS, NGFACE,
     *             AREA1, AREA2, AREA3, AREA4,     QE )
C
C     ELEMENT MASS MATRICES:
C
      CALL TRANTE( DEN, SH, TS, VOLUME, INTMAT, MXELE, MXPOI,
     *       AKE, QE, TEMP, IE,     AKP, QP)
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
C
      DO 500  NR=1,NNODE
      NODR = INTMAT(IE,NR)
C
C     DENOTE:  NSR = ROW POSITION IN THE SYSTEM  EQS.
C              NER = ROW POSITION IN THE ELEMENT EQS.
C
      NSR = (NODR-1) + 1
      NER = (NR  -1) + 1
      SYSQ(NSR) = SYSQ(NSR) + QP(NER)
      SYSK(NSR) = SYSK(NSR) + AKP(NER)
C
  500 CONTINUE
C
 5000 CONTINUE
C
      RETURN
      END
C
C-----------------------------
C
      SUBROUTINE CONDTE( INTMAT, MXPOI, COORD, MXELE, IE,TK, VOLUME,
     *             XG1, XG2, XG3, XG4, YG1, YG2, YG3, YG4, ZG1,
     *             ZG2, ZG3, ZG4,    AKE )
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 COORD(MXPOI,3), AKC(4,4), AKE(4,4),
     *       B(3,4), BT(4,3)
C
      INTEGER*2  INTMAT(MXELE,4)
C
C     FIND ELEMENT LOCAL COORDINATES:
C
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
      LL = INTMAT(IE,4)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      XG4 = COORD(LL,1)
C
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      YG4 = COORD(LL,2)
C
      ZG1 = COORD(II,3)
      ZG2 = COORD(JJ,3)
      ZG3 = COORD(KK,3)
      ZG4 = COORD(LL,3)
C
      DETA11 = XG2*(YG3*ZG4-YG4*ZG3) + XG3*(YG4*ZG2-YG2*ZG4)
     1       + XG4*(YG2*ZG3-YG3*ZG2)
      DETA21 = XG1*(YG4*ZG3-YG3*ZG4) + XG3*(YG1*ZG4-YG4*ZG1)
     1       + XG4*(YG3*ZG1-YG1*ZG3)
      DETA31 = XG1*(YG2*ZG4-YG4*ZG2) + XG2*(YG4*ZG1-YG1*ZG4)
     1       + XG4*(YG1*ZG2-YG2*ZG1)
      DETA41 = XG1*(YG3*ZG2-YG2*ZG3) + XG2*(YG1*ZG3-YG3*ZG1)
     1       + XG3*(YG2*ZG1-YG1*ZG2)
      VOLUME = (DETA11 + DETA21 +DETA31 + DETA41)/6.0D+00
C
      IF(VOLUME.LE.0.0D+00)  WRITE(6,5)  IE
    5 FORMAT(/,'  !!! ERROR !!!  ELEMENT NO.', I5,
     *       ' HAS NEGATIVE OR ZERO VOLUME ', /,
     *       ' -- CHECK F.E. MODEL FOR NODAL COORDINATES',
     *       ' AND ELEMENT NODAL CONNECTIONS --'          )
      IF(VOLUME.LE.0.0D+00)  STOP
C
      B1 = YG2*(ZG4-ZG3)+YG3*(ZG2-ZG4)+YG4*(ZG3-ZG2)
      B2 = YG1*(ZG3-ZG4)+YG3*(ZG4-ZG1)+YG4*(ZG1-ZG3)
      B3 = YG1*(ZG4-ZG2)+YG2*(ZG1-ZG4)+YG4*(ZG2-ZG1)
      B4 = YG1*(ZG2-ZG3)+YG2*(ZG3-ZG1)+YG3*(ZG1-ZG2)
      C1 = XG2*(ZG3-ZG4)+XG3*(ZG4-ZG2)+XG4*(ZG2-ZG3)
      C2 = XG1*(ZG4-ZG3)+XG3*(ZG1-ZG4)+XG4*(ZG3-ZG1)
      C3 = XG1*(ZG2-ZG4)+XG2*(ZG4-ZG1)+XG4*(ZG1-ZG2)
      C4 = XG1*(ZG3-ZG2)+XG2*(ZG1-ZG3)+XG3*(ZG2-ZG1)
      D1 = XG2*(YG4-YG3)+XG3*(YG2-YG4)+XG4*(YG3-YG2)
      D2 = XG1*(YG3-YG4)+XG3*(YG4-YG1)+XG4*(YG1-YG3)
      D3 = XG1*(YG4-YG2)+XG2*(YG1-YG4)+XG4*(YG2-YG1)
      D4 = XG1*(YG2-YG3)+XG2*(YG3-YG1)+XG3*(YG1-YG2)
C
      DO 10  I=1,3
      DO 10  J=1,4
      B(I,J) = 0.0D+00
   10 CONTINUE
C
      B(1,1) = B1
      B(1,2) = B2
      B(1,3) = B3
      B(1,4) = B4
      B(2,1) = C1
      B(2,2) = C2
      B(2,3) = C3
      B(2,4) = C4
      B(3,1) = D1
      B(3,2) = D2
      B(3,3) = D3
      B(3,4) = D4
C
      DO 20  I=1,3
      DO 30  J=1,4
      B(I,J)  = B(I,J)/(6.0D+00*VOLUME)
      BT(J,I) = B(I,J)
   30 CONTINUE
   20 CONTINUE
C
C     ELEMENT CONDUCTION MATRIX:
C
      DO 100  I=1,4
      DO 100  J=1,4
      AKC(I,J) = 0.0D+00
      DO 110  K=1,3
      AKC(I,J) = AKC(I,J) + BT(I,K)*B(K,J)
  110 CONTINUE
      AKC(I,J) = TK*VOLUME*AKC(I,J)
  100 CONTINUE
      DO 120  I=1,4
      DO 120  J=1,4
      AKE(I,J) = AKE(I,J) + AKC(I,J)
  120 CONTINUE
```

```
      RETURN
      END
C
C-----------------------------
C
      SUBROUTINE CONVTE(IE, H, TI, MXELE, NFACE, AREA1, AREA2,
     *                  AREA3, AREA4,        AKE, QE )
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 AKH(4,4), QH(4), AKE(4,4), QE(4)
C
      INTEGER*2  NFACE(MXELE,4)
C
      IF (NFACE(IE,1).EQ.1) THEN
        FAC = H*AREA1/12.0D+00
        DO 230  I=1,4
        DO 230  J=1,4
        IF ((I.EQ.4).OR.(J.EQ.4)) THEN
              AKH(I,J)=0.0D+00
        ELSE
              IF (I.EQ.J) AKH(I,J) = 2.0D+00*FAC
              IF (I.NE.J) AKH(I,J) = FAC
        END IF
  230   CONTINUE
        FAC = H*AREA1*TI/3.0D+00
        DO 250  I=1,4
        IF(I.EQ.4) THEN
              QH(I)=0.0D+00
        ELSE
              QH(I) = FAC
        END IF
  250   CONTINUE
        DO 260  I=1,4
        QE(I) = QE(I) + QH(I)
        DO 260  J=1,4
        AKE(I,J) = AKE(I,J) + AKH(I,J)
  260   CONTINUE
C
      END IF
C
      IF (NFACE(IE,2).EQ.1) THEN
        FAC = H*AREA2/12.0D+00
        DO 270  I=1,4
        DO 270  J=1,4
        IF((I.EQ.3).OR.(J.EQ.3)) THEN
              AKH(I,J)=0.0D+00
        ELSE
              IF(I.EQ.J) AKH(I,J) = 2.0D+00*FAC
              IF(I.NE.J) AKH(I,J) = FAC
        END IF
  270   CONTINUE
        FAC = H*AREA2*TI/3.0D+00
        DO 275  I=1,4
        IF(I.EQ.3) THEN
              QH(I)=0.0D+00
        ELSE
              QH(I) = FAC
        END IF
  275   CONTINUE
        DO 280  I=1,4
        QE(I) = QE(I) + QH(I)
        DO 280  J=1,4
        AKE(I,J) = AKE(I,J) + AKH(I,J)
  280   CONTINUE
        END IF
C
C
      IF (NFACE(IE,3).EQ.1) THEN
        FAC = H*AREA3/12.0D+00
        DO 282  I=1,4
        DO 282  J=1,4
        IF((I.EQ.1).OR.(J.EQ.1)) THEN
              AKH(I,J)=0.0D+00
        ELSE
              IF(I.EQ.J) AKH(I,J) = 2.0D+00*FAC
              IF(I.NE.J) AKH(I,J) = FAC
        END IF
  282   CONTINUE
        FAC = H*AREA3*TI/3.0D+00
        DO 285  I=1,4
        IF(I.EQ.1) THEN
              QH(I)=0.0D+00
        ELSE
              QH(I) = FAC
        END IF
  285   CONTINUE
        DO 290  I=1,4
        QE(I) = QE(I) + QH(I)
        DO 290  J=1,4
        AKE(I,J) = AKE(I,J) + AKH(I,J)
  290   CONTINUE
        END IF
C
      IF (NFACE(IE,4).EQ.1) THEN
        FAC = H*AREA4/12.0D+00
        DO 292  I=1,4
        DO 292  J=1,4
        IF((I.EQ.2).OR.(J.EQ.2)) THEN
              AKH(I,J)=0.0D+00
        ELSE
              IF(I.EQ.J) AKH(I,J) = 2.0D+00*FAC
              IF(I.NE.J) AKH(I,J) = FAC
        END IF
  292   CONTINUE
        FAC = H*AREA4*TI/3.0D+00
        DO 295  I=1,4
        IF(I.EQ.2) THEN
              QH(I)=0.0D+00
        ELSE
              QH(I) = FAC
        END IF
  295   CONTINUE
        DO 298  I=1,4
        QE(I) = QE(I) + QH(I)
        DO 298  J=1,4
        AKE(I,J) = AKE(I,J) + AKH(I,J)
  298   CONTINUE
        END IF
C
      RETURN
      END
C
C-----------------------------
C
      SUBROUTINE EXHTE( IE, MXELE, QS, NGFACE, AREA1,
     *                  AREA2, AREA3, AREA4,    QE )
C
C     AREA1 IS THE AREA OF I,J,K NODES
C     AREA2 IS THE AREA OF I,,J,I NODES
C     AREA3 IS THE AREA OF I,K,J NODES
C     AREA4 IS THE AREA OF I,I,K NODES
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 QSS(4), QE(4)
C
      INTEGER*2  NGFACE(MXELE,4)
C
      IF(NGFACE(IE,1).EQ.1) THEN
        FAC = QS*AREA1/3.0D+00
        DO 410  I=1,4
        IF (I.EQ.4) THEN
              QSS(I)=0.0D+00
        ELSE
              QSS(I) = FAC
        END IF
  410   CONTINUE
        DO 420  I=1,4
        QE(I) = QE(I) + QSS(I)
  420   CONTINUE
C
      END IF
C
      IF(NGFACE(IE,2).EQ.1) THEN
        FAC = QS*AREA2/3.0D+00
        DO 430  I=1,4
        IF (I.EQ.3) THEN
              QSS(I)=0.0D+00
        ELSE
              QSS(I) = FAC
        END IF
  430   CONTINUE
        DO 435  I=1,4
        QE(I) = QE(I) + QSS(I)
  435   CONTINUE
        ENDIF
C
      IF(NGFACE(IE,3).EQ.1) THEN
        FAC = QS*AREA3/3.0D+00
        DO 440  I=1,4
        IF (I.EQ.1) THEN
              QSS(I)=0.0D+00
        ELSE
              QSS(I) = FAC
        END IF
  440   CONTINUE
        DO 445  I=1,4
        QE(I) = QE(I) + QSS(I)
  445   CONTINUE
        ENDIF
C
      IF(NGFACE(IE,4).EQ.1) THEN
        FAC = QS*AREA4/3.0D+00
        DO 450  I=1,4
        IF (I.EQ.2) THEN
              QSS(I)=0.0D+00
        ELSE
              QSS(I) = FAC
        END IF
  450   CONTINUE
        DO 455  I=1,4
        QE(I) = QE(I) + QSS(I)
  455   CONTINUE
        END IF
        RETURN
        END
C
C-----------------------------
C
      SUBROUTINE TRANTE( DEN, SH, TS, VOLUME, INTMAT, MXELE, MXPOI,
     *                   AKE, QE, TEMP, IE,      AKP, QP)
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 TEMP(MXPOI)
      REAL*8 AKE(4,4), QE(4), AKM(4,4), AKP(4), QP(4)
      INTEGER*2 INTMAT(MXELE,4)
C
      FAC=DEN*SH*VOLUME/4.0D+00
      DO 650 I=1,4
      DO 650 J=1,4
      IF(I.EQ.J) AKM(I,J) = FAC
      IF(I.NE.J) AKM(I,J) = 0.0D+00
  650 CONTINUE
      DO 750 I=1,4
      QP(I) = 0.0D+00
      AKP(I) = 0.0D+00
  750 CONTINUE
      DO 820 I=1,4
      AQE = 0.0D+00
      DO 800 J=1,4
      AKP(I) = AKP(I) + AKM(I,J)
```

```
      AQE = AQE + (AKM(I,J) - TS*AKE(I,J))*TEMP(INTMAT(IE,J))
  800 CONTINUE
      QP(I) = QP(I) + TS*QE(I) + AQE
  820 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE PRTEMP(MXPOI, NPOIN, SYSQ)
      IMPLICIT double precision (A-H,O-Z)
      REAL*8  SYSQ(MXPOI)
C
C     PRINT PLOT FILE FOR I-DEAS
C
      TM = 0.0D+00
      WRITE(9,'(A)') '    -1'
      WRITE(9,'(A)') '  2414'
      WRITE(9,'(A)') '         2'
      WRITE(9,'(A)') ' B.C. 1,LOAD 2,TEMPERATURE'
      WRITE(9,'(A)') '         1'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)')  '         1        1        4        2
     *2        6'
      WRITE(9,'(A)')  '       -10        0        1        1
     *1         0        0         0'
      WRITE(9,'(A)')  '         2         0'
      WRITE(9,2000) TM, TM, TM, TM, TM, TM
      WRITE(9,2000) TM, TM, TM, TM, TM, TM
 2000 FORMAT(6E13.5)
      CONV = 9806570.182D+00
      DO 2100 L=1,NPOIN
      TEMP = SYSQ(L)*CONV
      WRITE(9,2050) L
 2050 FORMAT(6X,I4)
      WRITE(9,2000) TEMP,TM,TM,TM,TM,TM
 2100 CONTINUE
      WRITE(9,'(A)') '    -1'
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE PRNEXT( DEN,  SH,  H, Q,  QS, TI, ST,  FT, TS, MXPOI,
     *                   NPOIN, MXELE, NELEM, IBC, COORD, SYSQ, INTMAT,
     *                   LTYPE, NGFACE, NFACE, NAME1, TK )
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 COORD(MXPOI,3), SYSQ(MXPOI)
      INTEGER*2  INTMAT(MXELE,4), LTYPE(MXELE,3), NFACE(MXELE,4),
     *           NGFACE(MXELE,4), IBC(MXPOI)
      CHARACTER*20 NAME1
C
C     WRITE TITLE OF COMPUTATION:
C
      IL = 1
      WRITE(10,*) IL
      WRITE(10,50) NAME1
   50 FORMAT(' THIS DATA FILE CAME FROM : ',A20)
C
C     READ INPUT DATA:
C
      WRITE(10,'(A)')'   NPOIN   NELEM'
      WRITE(10,100) NPOIN, NELEM
  100 FORMAT(T5,I4,TR4,I4)
      WRITE(10,'(A)')'    DEN       SH       TK       H        Q
     *     QS       TI  '
      WRITE(10,200) DEN, SH, TK, H, Q, QS, TI
  200 FORMAT(E10.3,1X,5E10.3,1X,F8.3,1X,F8.3)
      WRITE(10,'(A)')'   ST     FT        TS'
      ST = FT
      FT = FT*2
      WRITE(10,500) ST, FT, TS
  500 FORMAT(T2,E10.3,T9,E10.3,T16,E10.4)
      WRITE(10,550) NPOIN
  550 FORMAT(' NODAL BOUNDARYCONDITIONS AND COORDINATES I',I6,'I')
      DO 700 I=1,NPOIN
      WRITE(10,600) I, IBC(I), (COORD(I,J), J=1,3), SYSQ(I)
  600 FORMAT(I4,TR4,I2,2X,F10.5,1X,F10.5,1X,F10.5,6X,F8.4)
  700 CONTINUE
      WRITE(10,750) NELEM
  750 FORMAT(' ELEMENT NODAL CONNECTIONS AND HEAT MODES I',I6,'I')
      DO 900 I=1,NELEM
      WRITE(10,800) I, (INTMAT(I,J), J=1,4), (LTYPE(I,J), J=1,3),
     *           (NGFACE(I,J), J=1,4), (NFACE(I,J), J=1,4)
  800 FORMAT(I4,TR3,I4,1X,I4,1X,I4,1X,I4,1X,I4,TR3,4I2,TR3,4I2,
     *     TR3,4I2)
  900 CONTINUE
C
      RETURN
      END
C
C          *****     INPUT DATA DETAILS     *****
C
C          NPOIN  is the number of total node
C          NELEM  is the number of total element
C          DEN    is the density of material
C          SH     is the specific heat of material
C          TK     is the thermal conductivity of material
C          H      is the convection coefficient of material
C          Q      is the internal heat generation
C          QS     is the external heat generation
C          TI     is the ambient temperature
C          ST     is the started time
C          FT     is the final time
C          TS     is the time step
C          IBC    is the boundary condition in each node
C          COORD  is the corrdinate in each node
C          TEMP   is the initial temperature in each node
C          INTMAT is the conectivity node in each element
C          LTYPE  is type of heat load
C          NFACE  is the convection surface
C          NGFACE is the surface of external heat generation
C
C
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

รายละเอียดของโปรแกรม STRES3D

รายละเอียดของโปรแกรม STRES3D

โปรแกรม STRES3D จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
C     PROGRAM STRES3D.FOR
C                                  UPDATE : 28/10/37
C     A FINITE ELEMENT MECHANICAL/THERMAL STRESS ANALYSIS PROGRAM
C     FOR THREE-DIMENSIONAL STRUCTURES.
C
C     THE VALUES DECLARED IN THE PARAMETER STATEMENT BELOW SHOULD
C     BE ASSIGNED ACCORDING TO THE SIZE OF THE PROBLEMS
C
C       NPOIN = NUMBER OF NODES IN THE MODEL
C       NELEM = NUMBER OF ELEMENTS IN THE MODEL
C       NHBW  = NUMBER OF HALF-BANDWIDTH
C
      PARAMETER (MXPOI=100, MXELE=300, MXHBW=130)
C
      IMPLICIT double precision (A-H,O-Z)
      DIMENSION TEXT(20)
      REAL*8 COORD(MXPOI,3), TEMP(MXPOI)
      REAL*8 SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
      REAL*8 SIX(MXPOI), SYY(MXPOI), SZZ(MXPOI), SXY(MXPOI)
      REAL*8 SYZ(MXPOI), SXZ(MXPOI), ONE(MXPOI), FT(MXELE,4)
      INTEGER*2 LTYPE(MXELE,4), INTMAT(MXELE,4), IBC(MXPOI,3)
      CHARACTER*20  NAME1, NAME2, NAME3
C
      NDF = 3
      NDOF = 12
      NNODE  = 4
C
C     READ INPUT FILENAME
C
   10 WRITE(6,15)
   15 FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
      READ(*, '(A)', ERR=10)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
C     READ OUTPUT FILENAME
C
   50 WRITE(6,60)
   60 FORMAT(/, ' PLEASE ENTER FILE NAME FOR DISPLACEMENT'
     *       ' AND STRESS SOLUTIONS:'              )
      READ(*, '(A)', ERR=50)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=50)
C
C     READ OUTPUT-PLOT FILE FOR I-DEAS
C
   60 WRITE(6,*)' CREAT THE PLOT FILE NAME ( YES:1 , NO:0 ) : '
      READ(5,*) IPT
      IF(IPT.EQ.0) GO TO 90
      IF(IPT.NE.1) GO TO 60
   70 WRITE(6,85)
   85 FORMAT(/, ' PLEASE INPUT THE PLOT FILE NAME :: ')
      READ(*, '(A)',ERR=70) NAME3
      OPEN(UNIT=9, FILE=NAME3//'.UNV',STATUS='NEW', ERR=70)
C
C     READ TITLE OF COMPUTATION:
C
   90 READ(7,*)  NLINES
      DO 100 ILINE=1,NLINES
      READ(7,1)  TEXT
    1 FORMAT(20A4)
  100 CONTINUE
C
C     READ INPUT DATA:
C
      WRITE(6,*)
      WRITE(6,'(A)') ' ****  BEGIN READING INPUT DATA ****'
      READ(7,1)  TEXT
      READ(7,*)  NPOIN, NELEM, NFORCE, ITYPE
      WRITE(6,105)  NPOIN
  105 FORMAT(/,15,'the total number of NODES      =',I4)
      WRITE(6,108)  NELEM
  108 FORMAT(/,15,'the total number of ELEMENTS  =',I4)
C
      READ(7,1)  TEXT
      READ(7,*)  ELAS, PR, ALPHA, TREF
      READ(7,1)  TEXT
      DO 130  IP=1,NPOIN
      READ(7,*)  I, (IBC(I,J), J=1,3), (COORD(I,K), K=1,3), TEMP(I)
      IF(I.NE.IP)  WRITE(6,135)  IP
  135 FORMAT(/,' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP)  STOP
  130 CONTINUE
      READ(7,1)  TEXT
      DO 140  IE=1,NELEM
      IF(ITYPE.GT.1) READ(7,*)  I, (INTMAT(I,J), J=1,4),
     *                (LTYPE(I,K), K=1,4), (FT(I,L), L=1,4)
      IF(ITYPE.LE.1) READ(7,*)  I, (INTMAT(I,J), J=1,4)
      IF(I.NE.IE)  WRITE(6,150)  IE
  150 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IE)  STOP
  140 CONTINUE
C
C     PRINT PROBLEM STATEMENT
C
      WRITE(8,'(A)') ' **** MATERIAL PROPERTIES ****'
      WRITE(8,200) ELAS
  200 FORMAT(10X,' YOUNG MODULUS  '       = ',E10.3)
      WRITE(8,210) PR
  210 FORMAT(10X,' POSSION RATIO          = ',F10.3)
      WRITE(8,220) ALPHA
  220 FORMAT(10X,' THERMAL EXPANSION      = ',E10.3)
      WRITE(8,230) TREF
  230 FORMAT(10X,' REFERENCE TEMPERATURE = ',F10.3,/)
      WRITE(8,'(A)') ' **** FINITE ELEMENT MODEL DATA ****'
      WRITE(8,240) NPOIN
  240 FORMAT(5X,' NUMBER OF NODE    = ',I6)
      WRITE(8,250) NELEM
  250 FORMAT(5X,' NUMBER OF ELEMENT = ',I6,/)
C
C     COMPUTE NODAL FORCES
C
      NEQ  = NPOIN*NDF
      DO 300  I=1,NEQ
      SYSF(I) = 0.0D+00
  300 CONTINUE
      READ(7,1)  TEXT
      DO 310  II=1,NFORCE
      READ(7,*)  N, FX, FY, FZ
      IEQ = (N-1)*NDF
      SYSF(IEQ+1) = FX
      SYSF(IEQ+2) = FY
      SYSF(IEQ+3) = FZ
  310 CONTINUE
C
C     COMPUTE HALF-BANDWIDTH:
C
      NHBW = 0
      DO 400  IE=1,NELEM
      MIN = 100000
      MAX = 0
      DO 410  IN=1,4
      II = INTMAT(IE,IN)
      IF(II.GT.MAX)  MAX = II
      IF(II.LT.MIN)  MIN = II
  410 CONTINUE
      NDIF = MAX - MIN + 1
      IF(NDIF.GT.NHBW)  NHBW = NDIF
  400 CONTINUE
      NHBW = NHBW*NDF
      RMEMORY = (118+24*NHBW)*NPOIN + 48*NELEM + 5842
      WRITE(6,420) RMEMORY
  420 FORMAT(/15,'required memory              =',E12.4,'  Bytes')
      IF(NHBW.GT.MXHBW)  WRITE(6,425)  NHBW
  425 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXHBW TO ', I5)
      IF(NHBW.GT.MXHBW)  STOP
C
      DO 430  I=1,NEQ
      DO 430  J=1,NHBW
      SYSK(I,J) = 0.0D+00
  430 CONTINUE
C
```

```
C     LOOP OVER ALL ELEMENTS TO COMPUTE ELEMENT MATRICES AND ASSEMBLE
C     THEM FOR SYSTEM MATRICES IN THE FORM NEEDED FOR MINIMUM MEMORY
C     REQUIREMENT:
C
      WRITE(6,440)
  440 FORMAT(/5,'+++ BEGIN ESTABLISHING ELEMENT MATRICES AND',
     *       ' ASSEMBLING ELEMENT EQUATIONS +++'    )
C
      CALL CSTFR(NELEM, INTMAT, COORD, ELAS, PR, ALPHA, TREF,
     * TEMP, MXPOI, MXELE, MXHBW, LTYPE, ITYPE, FT, SYSK, SYSF)
C
      WRITE(6,445)
  445 FORMAT(/5,'+++ END ASSEMBLING ELEMENT EQUATIONS +++')
      WRITE(6,1A1)' **** END READING INPUT DATA  ****'
C
C     END LOOP ALL ELEMENTS
C
C     APPLYING BOUNDARY CONDITIONS
C
      WRITE(6,450)
  450 FORMAT(/' **** BEGIN APPLYING BOUNDARY CONDITIONS ****')
      CALL APPLYBC(IBC, NPOIN, NHBW, MXPOI,MXHBW,  SYSK, SYSF)
      WRITE(6,1A1)' **** END APPLYING BOUNDARY CONDITIONS  ****'
C
C     END APPLYING BOUNDARY CONDITIONS
C
C     SOLVING DISPLACEMENT SOLUTIONS
C
      WRITE(6,460)
  460 FORMAT(/' **** BEGIN SOLVING DISPLACEMENT SOLUTIONS ****')
      WRITE(6,470) ' **** SOLVER DATA ****'
      WRITE(6,470) NEQ
      WRITE(6,470) NEQ
  470 FORMAT(5X,' total number of Equations          =',I5)
      WRITE(6,480) NHBW
      WRITE(6,480) NHBW
      WRITE(6,*)
  480 FORMAT(5X,' required number of Half-bandwidth  =',I5)
      CALL SOLVEQ(NEQ, NHBW, SYSK, MXPOI, MXHBW,  SYSF )
      WRITE(6,1A1)' **** END SOLVING DISPLACEMENT SOLUTIONS  ****'
C
C     END SOLVING DISPLACEMENT SOLUTIONS
C
C     PRINT DISPLACEMENTS
C
      CALL PRDISP(IPT, NDF, NPOIN, MXPOI, SYSF)
C
C     COMPUTE NODAL STRESSES:
C
      WRITE(6,530)
  530 FORMAT(/' **** BEGIN SOLVING STRESS SOLUTIONS **** ')
      CALL STRESS(NPOIN, NELEM, INTMAT, COORD, SYSF, ELAS,  PR,
     *       ALPHA,  TREF, TEMP, ONE, ITYPE, IPT, MXPOI,
     *       MXELE,       SXX, SYY, SZZ, SXY, SYZ, SXZ )
      WRITE(6,1A1)' **** END SOLVING STRESS SOLUTIONS  ****'
      WRITE(6,*)
C
      STOP
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE APPLYBC(IBC, NPOIN, NHBW, MXPOI, MXHBW,  SYSK, SYSF)
C
C     APPLY DISPLACEMENT BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C          0 = FREE TO MOVE
C          1 = FIXED
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
C
      INTEGER*2 IBC(MXPOI,3)
C
      DO 100 ID=1,NPOIN
      DO 200 IB=1,3
      IF(IBC(ID,IB).NE.1) GO TO 200
C
      IEQ = (ID-1)*3 + IB
      SYSF(IEQ)  = 0.0D+00
C
      SYSK(IEQ,1) = 1.0D+00
      DO 300 J=2,NHBW
      SYSK(IEQ,J) = 0.0D+00
  300 CONTINUE
C
      IF(IEQ.EQ.1) GO TO 450
      DO 400 I=1,IEQ-1
      IROW = IEQ - I
      ICOL = I - 1
      IF(ICOL.LT.NHBW) GO TO 450
      SYSK(IROW,ICOL) = 0.0D+00
  400 CONTINUE
  450 CONTINUE
C
  200 CONTINUE
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE ASSMBL(  IE, INTMAT, SGBL, FGBL, MXPOI,  MXELE,
     *                    MXHBW,   SYSK, SYSF )
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 SGBL(12,12), FGBL(12)
      REAL*8 SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
C
      INTEGER*2 INTMAT(MXELE,4)
C
      DO 100 NR=1,4
      NODR = INTMAT(IE,NR)
      DO 100 MR=1,3
C
C     DENOTE:  NSR = ROW POSITION IN THE SYSTEM  EQS.
C              NER = ROW POSITION IN THE ELEMENT EQS.
C
      NSR = (NODR-1)*3 + MR
      NER = (NR  -1)*3 + MR
      SYSF(NSR) = SYSF(NSR) + FGBL(NER)
C
      DO 200 NC=1,4
      NODC = INTMAT(IE,NC)
      DO 200 MC=1,3
C
C     DENOTE:  NSC = COLUMN POSITION IN THE SYSTEM  EQS.
C                    (AFTER ROTATION - READY FOR BANDED SOLVER)
C              NEC = COLUMN POSITION IN THE ELEMENT EQS.
C
      NSC = (NODC-1)*3 + MC - NSR + 1
      NEC = (NC  -1)*3 + MC
      IF(NSC.GT.0)
     &   SYSK(NSR,NSC) = SYSK(NSR,NSC) + SGBL(NER,NEC)
  200 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE CSTFR(NELEM, INTMAT, COORD, ELAS, PR, ALPHA, TREF,
     *  TEMP, MXPOI, MXELE, MXHBW, LTYPE, ITYPE, FT,  SYSK, SYSF )
C
C     COMPUTE ELEMENT MATRICES AND ASSEMBLE THEM FOR SYSTEM EQUATIONS
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 COORD(MXPOI,3), TEMP(MXPOI)
      REAL*8  SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3), FT(MXELE,4)
      REAL*8 SCST(12,12), FCST(12), CI(6,6), BI(6,12), BT(12,6)
      REAL*8 DUMA(6,12), DUMB(6), AL(6)
      INTEGER*2 LTYPE(MXELE,4),INTMAT(MXELE,4)
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 5000  IE=1,NELEM
C
C     FIND ELEMENT LOCAL COORDINATES:
C
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
      LL = INTMAT(IE,4)
      CALL LOCAL(MXPOI, II, JJ, KK, LL, ELAS, PR, COORD, B, C,
     *           VOLUME)
C
      DO 60 I=1,6
      DO 50 J=1,12
      BT(J,I) = B(I,J)
   50 CONTINUE
   60 CONTINUE
C
C     ELEMENT STIFFNESS MATRIX:
C
      DO 100  I=1,6
      DO 100  J=1,12
      DUMA(I,J) = 0.0D+00
      DO 200  K=1,6
      DUMA(I,J) = DUMA(I,J) + CI(I,K)*B(K,J)
  200 CONTINUE
  100 CONTINUE
C
      DO 300  I=1,12
      DO 300  J=1,12
      SCST(I,J) = 0.0D+00
      DO 400  K=1,6
      SCST(I,J) = SCST(I,J) + BT(I,K)*DUMA(K,J)
  400 CONTINUE
  300 CONTINUE
C
      DO 500  I=1,12
      DO 500  J=1,12
      SCST(I,J) = SCST(I,J)*VOLUME
  500 CONTINUE
C
C     ELEMENT NODAL FORCE DUE TO IN-PLANE THERMAL EXPANSION
C
      DO 550 I=1,12
      FCST(I) = 0.0D+00
  550 CONTINUE
C
C     DEFINE THE TYPE OF FORCE WITH :
C       ITYPE = 0 : ONLY NODAL FORCE
C               1 : NODAL FORCE & NODAL TEMPERATURE
C               2 : NODAL FORCE & SURFACE TRACTION
C               3 : NODAL FORCE, NODAL TENPERATURE & SURFACE TRACTION
C
      IF((ITYPE.EQ.1).OR.(ITYPE.EQ.3)) THEN
         AL(1) = ALPHA
         AL(2) = ALPHA
         AL(3) = ALPHA
         AL(4) = 0.0D+00
         AL(5) = 0.0D+00
         AL(6) = 0.0D+00
         DO 600  I=1,6
         DUMB(I) = 0.0D+00
```

```
      DO 700 J=1,6
      DUMB(I) = DUMB(I) + CB(J,I)*ALU(J)
  700 CONTINUE
  600 CONTINUE
C
      DO 800 I=1,12
      DO 900 J=1,6
      FCST(I) = FCST(I) + BT(J,I)*DUMB(J)
  900 CONTINUE
  800 CONTINUE
C
C     AVERAGE ELEMENT TEMPERATURE:
C
      TAVG = (TEMP(II) + TEMP(JJ) + TEMP(KK) + TEMP(LL))/4.0D+00
C
      FAC = (TAVG - TREF)*VOLUME
      DO 1000 I=1,12
      FCST(I) = FCST(I)*FAC
 1000 CONTINUE
    END IF
C
C     FIND THE SURFACE TRACTION
C
    IF(ITYPE .GT. 1) CALL FORCE( MXPOI, MXELE, IE, II, JJ, KK, LL,
   *                     COORD, LTYPE, FT,    SYSF )
C
C     ASSEMBLE THESE ELEMENT EQUATIONS INTO THE SYSTEM EQUATIONS.
C
    CALL ASSMBLE( IE, INTMAT,  SCST, FCST, MXPOI,  MXELE, MXHBW,
   *              SYSK, SYSF )
C
 5000 CONTINUE
C
    RETURN
    END
C
C-----------------------------------------------------------------------
C
    SUBROUTINE SOLVE(NROW, NHBW, GSTIF, MXPOI, MXHBW,   XL)
C
C     SOLVE A SET OF SIMULTANEOUS EQUATIONS USING GAUSS ELIMINATION.
C     THIS SOLVER ROUTINE CAN BE DESCRIBED BY USING AN EXAMPLE OF A
C     SET OF FOUR SIMULTANEOUS EQUATIONS (AFTER APPLYING BOUNDARY
C     CONDITIONS) AS SHOWN BELOW:
C
C        [ A11  A12  A13   0 ]  [ X1 ]      [ F1 ]
C        [                   ]  [    ]      [    ]
C        [ A12  A22  A23  A24 ]  [ X2 ]      [ F2 ]
C        [                   ]  [    ]  =   [    ]
C        [ A13  A23  A33  A34 ]  [ X3 ]      [ F3 ]
C        [                   ]  [    ]      [    ]
C        [  0   A24  A34  A44 ]  [ X4 ]      [ F4 ]
C
C     WHERE THE VARIABLE XL IS THE LOAD VECTOR ON RHS OF THE EQUATIONS.
C     THE GLOBAL STIFFNESS MATRIX ABOVE IS STORED IN THE VARIABLE
C     GSTIF IN THE FORMAT SHOWN BELOW. (HERE  NROW = 4  AND  NHBW = 3)
C
C                     [ A11  A12  A13 ]
C                     [               ]
C                     [ A22  A23  A24 ]
C        [ GSTIF ]  =  [               ]
C                     [ A33  A34   0  ]
C                     [               ]
C                     [ A44   0   0   ]
C
C     AND THE OUTPUT SOLUTIONS WILL BE STORED IN THE VARIABLE XL.
C
    IMPLICIT double precision (A-H,O-Z)
C
    REAL*8 GSTIF(MXPOI*3,MXHBW), XL(MXPOI*3)
C
    NR=NROW
    NC=NHBW
C
C     DIAGONALIZATION THE MATRIX.
C
    DO 10 I=1,NR
    PIVOT1=GSTIF(I,1)
    IF(ABS(PIVOT1).LT.10.0D-10) THEN
      WRITE(6,1025) I, PIVOT1
 1025   FORMAT( ' EQ. NO.', I5, ' HAS NEARLY ZERO PIVOT OF', E14.6,
   *     ' ** STOP **', /,
   *     ' *** CHECK NODE AND ELEMENT NUMBERING IN F.E. MODEL ***')
      STOP
    ENDIF
C
    XL(I)=XL(I)/PIVOT1
    DO 20 J=1,NC
   20 GSTIF(I,J)=GSTIF(I,J)/PIVOT1
    MM=0
    DO 30 II=I+1,NR
    MM=MM+1
    IF(MM+1.GT.NC) GOTO 30
    PIVOT2=GSTIF(II,MM+1)*PIVOT1
    XL(II)=XL(II)-XL(I)*PIVOT2
    IF(GSTIF(II,1).EQ.1.0D+00) THEN
      NB = 1
      DO 25 K=2,NHBW
      IF(GSTIF(II,K).EQ.0.0D+00) NB=NB+1
      IF(NB.EQ.NHBW) GO TO 30
      IF(GSTIF(II,K).NE.0.0D+00) GO TO 50
   25   CONTINUE
    END IF
   50 DO 40 JJ=1,NC
      JJJ=JJ+MM
      IF(JJJ.LE.NC)
     & GSTIF(II,JJ)=GSTIF(II,JJ) -
     &              GSTIF(I,JJJ)*PIVOT2
   40 CONTINUE
   30 CONTINUE
   10 CONTINUE
C
C     BACK SUBSTITUTION.
C
    DO 70 I=NR-1,1,-1
    II=I
    DO 80 J=I+1,NR
    II=II+1
    IF(II.LE.NHBW) XL(I)=XL(I)-GSTIF(I,II)*XL(J)
   80 CONTINUE
   70 CONTINUE
C
    RETURN
    END
C
C-----------------------------------------------------------------------
C
    SUBROUTINE STRSTR(NPOIN, NELEM, INTMAT, COORD, DISP, ELAS,  PR,
   *                  ALPHA, TREF, TEMP, ONE, ITYPE, IPT, MXPOI,
   *                  MXELE,       SXX, SYY, SZZ, SXY, SYZ, SXZ )
C
C     COMPUTE NODAL STRESS COMPONENTS FOR CST ELEMENTS
C
    IMPLICIT double precision (A-H,O-Z)
    REAL*8 COORD(MXPOI,3), TEMP(MXPOI)
    REAL*8  DISP(MXPOI*3), ONE(MXPOI)
    REAL*8 SXX(MXPOI), SYY(MXPOI), SZZ(MXPOI), SXY(MXPOI),
   *                   SYZ(MXPOI), SXZ(MXPOI)
    REAL*8 C(6,6), B(6,12), UG(4), VG(4), WG(4), EPS(6)
C
    INTEGER*2  INTMAT(MXELE,4)
C
    DO 10  I=1,NPOIN
    SXX(I) = 0.0D+00
    SYY(I) = 0.0D+00
    SZZ(I) = 0.0D+00
    SXY(I) = 0.0D+00
    SYZ(I) = 0.0D+00
    SXZ(I) = 0.0D+00
    ONE(I) = 0.0D+00
   10 CONTINUE
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
    DO 1000  IE=1,NELEM
C
C     FIND ELEMENT LOCAL COORDINATES:
C
    II = INTMAT(IE,1)
    JJ = INTMAT(IE,2)
    KK = INTMAT(IE,3)
    LL = INTMAT(IE,4)
    CALL LOCAL(MXPOI, II, JJ, KK, LL, ELAS, PR, COORD,    B, C,
   *           VOLUME)
C
C     GATHER ELEMENT NODAL DISPLACEMENTS:
C
    DO 200  J1=1,4
    I1  = INTMAT(IE,J1)
    IEQ = (I1-1)*3 + 1
    UG(J1) = DISP(IEQ  )
    VG(J1) = DISP(IEQ+1)
    WG(J1) = DISP(IEQ+2)
  200 CONTINUE
C
C     COMPUTE THE TOTAL STRAINS:
C
    DO 220  I=1,6
    EPS(I) = 0.0D+00
    DO 230  J=1,4
    J1 = (J-1)*3 + 1
    EPS(I) = EPS(I) + B(I,J1)*UG(J) + B(I,J1+1)*VG(J) +
   1                  B(I,J1+2)*WG(J)
  230 CONTINUE
  220 CONTINUE
C
C     COMPUTE THERMAL STRAINS USING AVERAGE ELEMENT NODAL TEMPERATURES:
C
    IF((ITYPE.EQ.1).OR.(ITYPE.EQ.3)) THEN
      TAVG = (TEMP(II) + TEMP(JJ) + TEMP(KK) + TEMP(LL))/4.0D+00
      FAC = TAVG - TREF
C
C     COMPUTE THE NET STRAINS:
C
      FAC = ALPHA*FAC
      EPS(1) = EPS(1) - FAC
      EPS(2) = EPS(2) - FAC
      EPS(3) = EPS(3) - FAC
      EPS(4) = EPS(4) - 0.0D+00
      EPS(5) = EPS(5) - 0.0D+00
      EPS(6) = EPS(6) - 0.0D+00
    END IF
C
C     COMPUTE THE ELEMENT STRESSES:
C
    SXXE = C(1,1)*EPS(1) + C(1,2)*EPS(2) + C(1,3)*EPS(3)
    SYYE = C(2,1)*EPS(1) + C(2,2)*EPS(2) + C(2,3)*EPS(3)
    SZZE = C(3,1)*EPS(1) + C(3,2)*EPS(2) + C(3,3)*EPS(3)
    SXYE = C(4,4)*EPS(4)
    SYZE = C(5,5)*EPS(5)
    SXZE = C(6,6)*EPS(6)
C
C     COMPUTE NODAL STRESSES FROM ELEMENT STRESSES.
C
    SXX(II) = SXX(II) + SXXE
    SXX(JJ) = SXX(JJ) + SXXE
    SXX(KK) = SXX(KK) + SXXE
    SXX(LL) = SXX(LL) + SXXE
    SYY(II) = SYY(II) + SYYE
    SYY(JJ) = SYY(JJ) + SYYE
```

```
      SYY(KK) = SYY(KK) + SYYE
      SYY(LL) = SYY(LL) + SYYE
      SZZ(II) = SZZ(II) + SZZE
      SZZ(JJ) = SZZ(JJ) + SZZE
      SZZ(KK) = SZZ(KK) + SZZE
      SZZ(LL) = SZZ(LL) + SZZE
C
      SXY(II) = SXY(II) + SXYE
      SXY(JJ) = SXY(JJ) + SXYE
      SXY(KK) = SXY(KK) + SXYE
      SXY(LL) = SXY(LL) + SXYE
      SYZ(II) = SYZ(II) + SYZE
      SYZ(JJ) = SYZ(JJ) + SYZE
      SYZ(KK) = SYZ(KK) + SYZE
      SYZ(LL) = SYZ(LL) + SYZE
      SXZ(II) = SXZ(II) + SXZE
      SXZ(JJ) = SXZ(JJ) + SXZE
      SXZ(KK) = SXZ(KK) + SXZE
      SXZ(LL) = SXZ(LL) + SXZE
      ONE(II) = ONE(II) + 1.0D+00
      ONE(JJ) = ONE(JJ) + 1.0D+00
      ONE(KK) = ONE(KK) + 1.0D+00
      ONE(LL) = ONE(LL) + 1.0D+00
C
 1000 CONTINUE
C
C     PRINT OUT THESE NODAL STRESSES:
C
      DO 1100 I=1,NPOIN
      IF(ONE(I).EQ.0.0D+00) WRITE(6,1200) I
 1200 FORMAT(' *** WARNING ***  NO STRESS CONTRIBUTION AT NODE', I5)
      IF(ONE(I).EQ.0.0D+00) ONE(I) = 1.0D+00
      SXX(I) = SXX(I)/ONE(I)
      SYY(I) = SYY(I)/ONE(I)
      SZZ(I) = SZZ(I)/ONE(I)
      SXY(I) = SXY(I)/ONE(I)
      SYZ(I) = SYZ(I)/ONE(I)
      SXZ(I) = SXZ(I)/ONE(I)
 1100 CONTINUE
C
      WRITE(6,1300) NPOIN
 1300 FORMAT(/,' NODAL STRESS SOLUTIONS I', I5,')', //,
     *       2X, 'NODE', 4X, 'SXX', 17X, 'SYY', 17X, 'SZZ', /,
     *       20X,'SXY', 17X, 'SYZ', 17X, 'SXZ', 8X, 'SVON',/)
      DO 1400 I=1,NPOIN
      SVONA = 0.5*(SXX(I) - SYY(I))*(SXX(I) - SYY(I))
      SVONB = 0.5*(SYY(I) - SZZ(I))*(SYY(I) - SZZ(I))
      SVONC = 0.5*(SZZ(I) - SXX(I))*(SZZ(I) - SXX(I))
      SVOND = 3.*(SXY(I)*SXY(I)+SYZ(I)*SYZ(I)+SXZ(I)*SXZ(I))
      SVON  = SQRT(SVONA+SVONB+SVONC+SVOND)
      IF (I.EQ.1) THEN
        SXMAX  = SXX(I)
        SYMAX  = SYY(I)
        SZMAX  = SZZ(I)
        SXYMAX = SXY(I)
        SYZMAX = SYZ(I)
        SXZMAX = SXZ(I)
        SVNMAX = SVON
        SXMIN  = SXX(I)
        SYMIN  = SYY(I)
        SZMIN  = SZZ(I)
        SXYMIN = SXY(I)
        SYZMIN = SYZ(I)
        SXZMIN = SXZ(I)
        SVNMIN = SVON
        IXMAX  = 1
        IXYMAX = 1
        IYMAX  = 1
        IYZMAX = 1
        IZMAX  = 1
        IXZMAX = 1
        IVNMAX = 1
        IXMIN  = 1
        IXYMIN = 1
        IYMIN  = 1
        IYZMIN = 1
        IZMIN  = 1
        IXZMIN = 1
        IVNMIN = 1
      ELSE
        IF(SXX(I).GT.SXMAX) IXMAX  = I
        IF(SXX(I).GT.SXMAX) SXMAX  = SXX(I)
        IF(SXX(I).LT.SXMIN) IXMIN  = I
        IF(SXX(I).LT.SXMIN) SXMIN  = SXX(I)
        IF(SYY(I).GT.SYMAX) IYMAX  = I
        IF(SYY(I).GT.SYMAX) SYMAX  = SYY(I)
        IF(SYY(I).LT.SYMIN) IYMIN  = I
        IF(SYY(I).LT.SYMIN) SYMIN  = SYY(I)
        IF(SZZ(I).GT.SZMAX) IZMAX  = I
        IF(SZZ(I).GT.SZMAX) SZMAX  = SZZ(I)
        IF(SZZ(I).LT.SZMIN) IZMIN  = I
        IF(SZZ(I).LT.SZMIN) SZMIN  = SZZ(I)
        IF(SXY(I).GT.SXYMAX) IXYMAX = I
        IF(SXY(I).GT.SXYMAX) SXYMAX = SXY(I)
        IF(SXY(I).LT.SXYMIN) IXYMIN = I
        IF(SXY(I).LT.SXYMIN) SXYMIN = SXY(I)
        IF(SYZ(I).GT.SYZMAX) IYZMAX = I
        IF(SYZ(I).GT.SYZMAX) SYZMAX = SYZ(I)
        IF(SYZ(I).LT.SYZMIN) IYZMIN = I
        IF(SYZ(I).LT.SYZMIN) SYZMIN = SYZ(I)
        IF(SXZ(I).GT.SXZMAX) IXZMAX = I
        IF(SXZ(I).GT.SXZMAX) SXZMAX = SXZ(I)
        IF(SXZ(I).LT.SXZMIN) IXZMIN = I
        IF(SXZ(I).LT.SXZMIN) SXZMIN = SXZ(I)
        IF(SVON.GT.SVNMAX) IVNMAX = I
        IF(SVON.GT.SVNMAX) SVNMAX = SVON
        IF(SVON.LT.SVNMIN) IVNMIN = I
        IF(SVON.LT.SVNMIN) SVNMIN = SVON
      END IF
C
        WRITE(6,1500) I, SXX(I), SYY(I), SZZ(I)
 1500 FORMAT(I4, 1X, E12.5, 8X, E12.5, 8X, E12.5)
        WRITE(6,1600) SXY(I), SYZ(I), SXZ(I), SVON
 1600 FORMAT(15X, E12.5, 8X, E12.5, 8X, E12.5, E12.5)
 1400 CONTINUE
        WRITE(6,*)
        WRITE(6,*)
        WRITE(2,'(A)') ' MAXIMUM STRESSES '
        WRITE(6,*)
        WRITE(6,1700) IXMAX,IXYMAX,IYMAX,IYZMAX,IZMAX,IXZMAX,IVNMAX
        WRITE(6,*)
        WRITE(6,1800) SXMAX,SYMAX,SZMAX
        WRITE(6,1900) SXYMAX,SYZMAX,SXZMAX,SVNMAX
 1700 FORMAT(6X,I4,6X,I4,6X,I4,6X,I4,6X,I4,6X,I4,6X,I4,6X,I4)
 1800 FORMAT(5X, E12.5, 8X, E12.5, 8X, E12.5)
 1900 FORMAT(15X, E12.5, 8X, E12.5, 8X, E12.5, E12.5)
        WRITE(6,*)
        WRITE(6,*)
        WRITE(6,'(A)') ' MINIMUM STRESSES '
        WRITE(6,*)
        WRITE(6,1700) IXMIN,IXYMIN,IYMIN,IYZMIN,IZMIN,IXZMIN,IVNMIN
        WRITE(6,*)
        WRITE(6,1800) SXMIN,SYMIN,SZMIN
        WRITE(6,1900) SXYMIN,SYZMIN,SXZMIN,SVNMIN
C
C     PRINT PLOT FILE FOR I-DEAS
C
      IF(IPT.NE.1) RETURN
      TM = 0.0D+00
      WRITE(9,'(A)') '    -1'
      WRITE(9,'(A)') '  2414'
      WRITE(9,'(A)') '         2'
      WRITE(9,'(A)') ' B.C. 1,LOAD 1,STRESS_2'
      WRITE(9,'(A)') '         1'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') 'NONE'
      WRITE(9,'(A)') '         1         1         4         2
     *2         6'
      WRITE(9,'(A)') '       -10         0         1         1
     *1         0         0         0'
      WRITE(9,'(A)') '         2         0'
      WRITE(9,2000) TM, TM, TM, TM, TM, TM
      WRITE(9,2000) TM, TM, TM, TM, TM, TM
 2000 FORMAT(6E13.5)
      CONV = 1.4503D-04
      DO 2100 I=1,NPOIN
      XX = SXX(I)/CONV
      XY = SXY(I)/CONV
      YY = SYY(I)/CONV
      XZ = SXZ(I)/CONV
      YZ = SYZ(I)/CONV
      ZZ = SZZ(I)/CONV
      WRITE(9,2050) I
 2050 FORMAT(6X,I4)
      WRITE(9,2000) XX,XY,YY,XZ,YZ,ZZ
 2100 CONTINUE
      WRITE(9,'(A)') '    -1'
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE LOCAL(MXPOI, II, JJ, KK, LL, ELAS, PR, COORD,   B, C,
     *                 VOLUME)
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 COORD(MXPOI,3), C(6,6), B(6,12)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      XG4 = COORD(LL,1)
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      YG4 = COORD(LL,2)
C
      ZG1 = COORD(II,3)
      ZG2 = COORD(JJ,3)
      ZG3 = COORD(KK,3)
      ZG4 = COORD(LL,3)
C
      DETA11 = XG2*(YG3*ZG4-YG4*ZG3) + XG3*(YG4*ZG2-YG2*ZG4)
     1       + XG4*(YG2*ZG3-YG3*ZG2)
      DETA21 = XG1*(YG4*ZG3-YG3*ZG4) + XG3*(YG1*ZG4-YG4*ZG1)
     1       + XG4*(YG3*ZG1-YG1*ZG3)
      DETA31 = XG1*(YG2*ZG4-YG4*ZG2) + XG2*(YG4*ZG1-YG1*ZG4)
     1       + XG4*(YG1*ZG2-YG2*ZG1)
      DETA41 = XG1*(YG3*ZG2-YG2*ZG3) + XG2*(YG1*ZG3-YG3*ZG1)
     1       + XG3*(YG2*ZG1-YG1*ZG2)
      VOLUME = (DETA11 + DETA21 +DETA31 + DETA41)/6.0D+00
C
      B1 = YG2*(ZG4-ZG3)+YG3*(ZG2-ZG4)+YG4*(ZG3-ZG2)
      B2 = YG1*(ZG3-ZG4)+YG3*(ZG4-ZG1)+YG4*(ZG1-ZG3)
      B3 = YG1*(ZG4-ZG2)+YG2*(ZG1-ZG4)+YG4*(ZG2-ZG1)
      B4 = YG1*(ZG2-ZG3)+YG2*(ZG3-ZG1)+YG3*(ZG1-ZG2)
      C1 = XG2*(ZG3-ZG4)+XG3*(ZG4-ZG2)+XG4*(ZG2-ZG3)
      C2 = XG1*(ZG4-ZG3)+XG3*(ZG1-ZG4)+XG4*(ZG3-ZG1)
      C3 = XG1*(ZG2-ZG4)+XG2*(ZG4-ZG1)+XG4*(ZG1-ZG2)
      C4 = XG1*(ZG3-ZG2)+XG2*(ZG1-ZG3)+XG3*(ZG2-ZG1)
      D1 = XG2*(YG4-YG3)+XG3*(YG2-YG4)+XG4*(YG3-YG2)
      D2 = XG1*(YG3-YG4)+XG3*(YG4-YG1)+XG4*(YG1-YG3)
      D3 = XG1*(YG4-YG2)+XG2*(YG1-YG4)+XG4*(YG2-YG1)
      D4 = XG1*(YG2-YG3)+XG2*(YG3-YG1)+XG3*(YG1-YG2)
C
      DO 110 I=1,6
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      DO 110  J=1,12
      B(I,J) = 0.0D+00
  110 CONTINUE
C
      B(1,1)  = B1
      B(1,4)  = B2
      B(1,7)  = B3
      B(1,10) = B4
C
      B(2,2)  = C1
      B(2,5)  = C2
      B(2,8)  = C3
      B(2,11) = C4
C
      B(3,3)  = D1
      B(3,6)  = D2
      B(3,9)  = D3
      B(3,12) = D4
C
      B(4,1)  = C1
      B(4,2)  = B1
      B(4,4)  = C2
      B(4,5)  = B2
      B(4,7)  = C3
      B(4,8)  = B3
      B(4,10) = C4
      B(4,11) = B4
C
      B(5,2)  = D1
      B(5,3)  = C1
      B(5,5)  = D2
      B(5,6)  = C2
      B(5,8)  = D3
      B(5,9)  = C3
      B(5,11) = D4
      B(5,12) = C4
C
      B(6,1)  = D1
      B(6,3)  = B1
      B(6,4)  = D2
      B(6,6)  = B2
      B(6,7)  = D3
      B(6,9)  = B3
      B(6,10) = D4
      B(6,12) = B4
C
      DO 120  I=1,6
      DO 130  J=1,12
      B(I,J) = B(I,J)/(6.0D+00*VOLUME)
  130 CONTINUE
  120 CONTINUE
C
C     ELASTICITY MATRIX:
C
      DO 50 I=1,6
      DO 50 J=1,6
      C(I,J) = 0.0D+00
   50 CONTINUE
      FAC = ELAS/((1+PR)*(1.0D+00-2.0D+00*PR))
      C(1,1) = FAC*(1.0D+00-PR)
      C(1,2) = FAC*PR
      C(1,3) = C(1,2)
      C(2,1) = C(1,2)
      C(2,2) = C(1,1)
      C(2,3) = C(2,1)
      C(3,1) = C(1,3)
      C(3,2) = C(1,3)
      C(3,3) = C(1,1)
      C(4,4) = FAC*(1.0D+00-2.0D+00*PR)/2.0D+00
      C(5,5) = C(4,4)
      C(6,6) = C(4,4)
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE FORCE(MXPOI, MXELE, IE, II, JJ, KK, LL, COORD,
     *                 LTYPE, FT,  SYSF)
C
C     COMPUTE THE SURFACE TRACTION
C
      IMPLICIT double precision (A-H,O-Z)
      REAL*8 COORD(MXPOI,3), SYSF(MXPOI*3), FT(MXELE,4)
      INTEGER*2 LTYPE(MXELE,4)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      XG4 = COORD(LL,1)
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      YG4 = COORD(LL,2)
C
      ZG1 = COORD(II,3)
      ZG2 = COORD(JJ,3)
      ZG3 = COORD(KK,3)
      ZG4 = COORD(LL,3)
C
      AS = SQRT((XG2-XG1)*(XG2-XG1)+(YG2-YG1)*(YG2-YG1)
     1          +(ZG2-ZG1)*(ZG2-ZG1))
      BS = SQRT((XG3-XG1)*(XG3-XG1)+(YG3-YG1)*(YG3-YG1)
     1          +(ZG3-ZG1)*(ZG3-ZG1))
      CS = SQRT((XG3-XG2)*(XG3-XG2)+(YG3-YG2)*(YG3-YG2)
     1          +(ZG3-ZG2)*(ZG3-ZG2))
      DS = SQRT((XG4-XG3)*(XG4-XG3)+(YG4-YG3)*(YG4-YG3)
     1          +(ZG4-ZG3)*(ZG4-ZG3))
      ES = SQRT((XG4-XG1)*(XG4-XG1)+(YG4-YG1)*(YG4-YG1)
     1          +(ZG4-ZG1)*(ZG4-ZG1))
      FS = SQRT((XG4-XG2)*(XG4-XG2)+(YG4-YG2)*(YG4-YG2)
     1          +(ZG4-ZG2)*(ZG4-ZG2))
C
C        THIS FACE IS FACE1 , I-J-K (LOCAL COORDINATE)
C
      IF(LTYPE(IE,1).EQ.1) THEN
        S = 0.5*(AS+BS+CS)
        AREA1 = SQRT(S*(S-AS)*(S-BS)*(S-CS))
        FAC = AREA1*FT(IE,1)/3
        DETA = XG1*(YG3*ZG2 - YG2*ZG3)
        DETB = XG2*(YG1*ZG3 - YG3*ZG1)
        DETC = XG3*(YG2*ZG1 - YG1*ZG2)
        DET1 = DETA + DETB + DETC
        A1 = (YG1*(ZG3-ZG2)+YG2*(ZG1-ZG3)+YG3*(ZG2-ZG1))/DET1
        B1 = (XG1*(ZG2-ZG3)+XG2*(ZG3-ZG1)+XG3*(ZG1-ZG2))/DET1
        C1 = (XG1*(YG3-YG2)+XG2*(YG1-YG3)+XG3*(YG2-YG1))/DET1
        R1 = SQRT(A1*A1 + B1*B1 + C1*C1)
C
        IN = (II-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A1/R1
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B1/R1
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C1/R1
C
        IN = (JJ-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A1/R1
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B1/R1
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C1/R1
C
        IN = (KK-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A1/R1
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B1/R1
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C1/R1
      END IF
C
C        THIS FACE IS FACE2 , J-K-L (LOCAL COORDINATE)
C
      IF(LTYPE(IE,2).EQ.1) THEN
        S = 0.5*(AS+ES+FS)
        AREA2 = SQRT(S*(S-AS)*(S-ES)*(S-FS))
        FAC = AREA2*FT(IE,2)/3
        DETA = XG3*(YG4*ZG2 - YG2*ZG4)
        DETB = XG2*(YG3*ZG4 - YG4*ZG3)
        DETC = XG4*(YG2*ZG3 - YG3*ZG2)
        DET2 = DETA + DETB + DETC
        A2 = (YG3*(ZG4-ZG2)+YG2*(ZG3-ZG4)+YG4*(ZG2-ZG3))/DET2
        B2 = (XG3*(ZG2-ZG4)+XG2*(ZG4-ZG3)+XG4*(ZG3-ZG2))/DET2
        C2 = (XG3*(YG4-YG2)+XG2*(YG3-YG4)+XG4*(YG2-YG3))/DET2
        R2 = SQRT(A2*A2 + B2*B2 + C2*C2)
C
        IN = (JJ-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A2/R2
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B2/R2
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C2/R2
C
        IN = (KK-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A2/R2
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B2/R2
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C2/R2
C
        IN = (LL-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A2/R2
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B2/R2
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C2/R2
      END IF
C
C        THIS FACE IS FACE3 , I-J-L (LOCAL COORDINATE)
C
      IF(LTYPE(IE,3).EQ.1) THEN
        S = 0.5*(CS+DS+FS)
        AREA3 = SQRT(S*(S-CS)*(S-DS)*(S-FS))
        FAC   = AREA3*FT(IE,3)/3
        DETA = XG4*(YG1*ZG2 - YG2*ZG1)
        DETB = XG2*(YG4*ZG1 - YG1*ZG4)
        DETC = XG1*(YG2*ZG4 - YG4*ZG2)
        DET3 = DETA + DETB + DETC
        A3 = (YG4*(ZG1-ZG2)+YG2*(ZG4-ZG1)+YG1*(ZG2-ZG4))/DET3
        B3 = (XG4*(ZG2-ZG1)+XG2*(ZG1-ZG4)+XG1*(ZG4-ZG2))/DET3
        C3 = (XG4*(YG1-YG2)+XG2*(YG4-YG1)+XG1*(YG2-YG4))/DET3
        R3 = SQRT(A3*A3 + B3*B3 + C3*C3)
C
        IN = (II-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A3/R3
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B3/R3
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C3/R3
C
        IN = (JJ-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A3/R3
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B3/R3
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C3/R3
C
        IN = (LL-1)*3
        SYSF(IN+1) = SYSF(IN+1) +  FAC*A3/R3
        SYSF(IN+2) = SYSF(IN+2) +  FAC*B3/R3
        SYSF(IN+3) = SYSF(IN+3) +  FAC*C3/R3
      END IF
C
C        THIS FACE IS FACE4 , I-K-L (LOCAL COORDINATE)
C
      IF(LTYPE(IE,4).EQ.1) THEN
        S = 0.5*(BS+DS+ES)
        AREA4 = SQRT(S*(S-BS)*(S-DS)*(S-ES))
        FAC = AREA4*FT(IE,4)/3
        DETA = XG1*(YG4*ZG3 - YG3*ZG4)
        DETB = XG3*(YG1*ZG4 - YG4*ZG1)
        DETC = XG4*(YG3*ZG1 - YG1*ZG3)
        DET4 = DETA + DETB + DETC
        A4 = (YG1*(ZG4-ZG3)+YG3*(ZG1-ZG4)+YG4*(ZG3-ZG1))/DET4
        B4 = (XG1*(ZG3-ZG4)+XG3*(ZG4-ZG1)+XG4*(ZG1-ZG3))/DET4
        C4 = (XG1*(YG4-YG3)+XG3*(YG1-YG4)+XG4*(YG3-YG1))/DET4
        R4 = SQRT(A4*A4 + B4*B4 + C4*C4)
C
```

```
      IN = (II-1)*3
      SYSF(IN+1) = SYSF(IN+1) +  FAC*A4/R4
      SYSF(IN+2) = SYSF(IN+2) +  FAC*B4/R4
      SYSF(IN+3) = SYSF(IN+3) +  FAC*C4/R4
C
      IN = (LL-1)*3
      SYSF(IN+1) = SYSF(IN+1) +  FAC*A4/R4
      SYSF(IN+2) = SYSF(IN+2) +  FAC*B4/R4
      SYSF(IN+3) = SYSF(IN+3) +  FAC*C4/R4
C
      IN = (KK-1)*3
      SYSF(IN+1) = SYSF(IN+1) +  FAC*A4/R4
      SYSF(IN+2) = SYSF(IN+2) +  FAC*B4/R4
      SYSF(IN+3) = SYSF(IN+3) +  FAC*C4/R4
     END IF
C
     RETURN
     END
C
C-----------------------------
C
     SUBROUTINE PRDIS(IPT, NDF, NPOIN, MXPOI, SYSF)
     IMPLICIT double precision (A-H,O-Z)
     REAL*8  SYSF(MXPOI*3)
C
     TM = 0.0D+00
     UMAX = SYSF(1)
     UMIN = SYSF(1)
     VMAX = SYSF(2)
     VMIN = SYSF(2)
     WMAX = SYSF(3)
     WMIN = SYSF(3)
     IUMAX = 1
     IVMAX = 1
     IWMAX = 1
     IUMIN = 1
     IVMIN = 1
     IWMIN = 1
     DO 490 K=2,NPOIN
     K1 = (K-1)*NDF + 1
     K2 = K1 + 1
     K3 = K2 + 1
     IF(SYSF(K1).GT.UMAX) IUMAX  =  K
     IF(SYSF(K1).GT.UMAX) UMAX   =  SYSF(K1)
     IF(SYSF(K1).LT.UMIN) IUMIN  =  K
     IF(SYSF(K1).LT.UMIN) UMIN   =  SYSF(K1)
     IF(SYSF(K2).GT.VMAX) IVMAX  =  K
     IF(SYSF(K2).GT.VMAX) VMAX   =  SYSF(K2)
     IF(SYSF(K2).LT.VMIN) IVMIN  =  K
     IF(SYSF(K2).LT.VMIN) VMIN   =  SYSF(K2)
     IF(SYSF(K3).GT.WMAX) IWMAX  =  K
     IF(SYSF(K3).GT.WMAX) WMAX   =  SYSF(K3)
     IF(SYSF(K3).LT.WMIN) IWMIN  =  K
     IF(SYSF(K3).LT.WMIN) WMIN   =  SYSF(K3)
490 CONTINUE
     WRITE(6,495)  NPOIN
495 FORMAT(' NODAL DISPLACEMENT SOLUTIONS !', I5,'!',
    *      //, 2X, 'NODE', 13X, 'U', 13X, 'V', 13X, 'W',/)
     I1 = 1
     DO 500  IP=1,NPOIN
     I3 = IP*NDF
     WRITE(6,510)  IP, (SYSF(I), I=I1,I3)
510 FORMAT(I6, 3E14.6)
     I1 = I3 + 1
500 CONTINUE
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,'(A)') ' MAXIMUM DISPLACEMENTS '
     WRITE(6,*)
     WRITE(6,700) IUMAX,IVMAX,IWMAX
     WRITE(6,800) UMAX,VMAX,WMAX
700 FORMAT(12X,I4,10X,I4,10X,I4)
800 FORMAT(5X, 3E14.6)
     WRITE(6,*)
     WRITE(6,*)
     WRITE(6,'(A)') ' MINIMUM DISPLACEMENTS '
     WRITE(6,*)
     WRITE(6,700) IUMIN,IVMIN,IWMIN
     WRITE(6,800) UMIN,VMIN,WMIN
C
C    PRINT PLOT FILE FOR I-DEAS
C
     IF(IPT.NE.1) RETURN
     REWIND(9)
     WRITE(9,'(A)') '    -1'
     WRITE(9,'(A)') '  2414'
     WRITE(9,'(A)') '         1'
     WRITE(9,'(A)') ' B.C. 1,LOAD 1,DISPLACEMENT_1'
     WRITE(9,'(A)') '         1'
     WRITE(9,'(A)') 'NONE'
     WRITE(9,'(A)') 'NONE'
     WRITE(9,'(A)') 'NONE'
     WRITE(9,'(A)') 'NONE'
     WRITE(9,'(A)') 'NONE'
     WRITE(9,'(A)') '         1         1         3         8
    *2         6'
     WRITE(9,'(A)') '       -10         0         1         1
    *1         0         0         0'
     WRITE(9,'(A)') '         2         0'
     WRITE(9,850) TM, TM, TM, TM, TM, TM
     WRITE(9,850) TM, TM, TM, TM, TM, TM
850 FORMAT(6E13.5)
     CONV = 2.540D-02
     J1 = 1
     DO 880 J=1,NPOIN
     J2 = J1 + 1
     J3 = J2 + 1
     UP = SYSF(J1)*CONV
     VP = SYSF(J2)*CONV
     WP = SYSF(J3)*CONV
     WRITE(9,840) J
840 FORMAT(6X,I4)
     WRITE(9,850) UP,VP,WP,TM,TM,TM
     J1 = J*NDF + 1
880 CONTINUE
     WRITE(9,'(A)') '    -1'
     RETURN
     END
□
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

รายละเอียดของโปรแกรม CNFH

รายละเอียดของโปรแกรม CNFH

โปรแกรม CNFH จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
C       CNFH.FOR                    Update : 9/1/95
C
C       THIS PROGRAM WILL TRANSFER FORMAT DATA FILE FROM
C       COSMIC-NASTRAN (I-DEAS) TO HEAT3D.FOR ON UNIX
C
      PARAMETER (MXPOI=200,MXELE=400)
C
      IMPLICIT double precision (A-H,O-Z)
      DIMENSION TEXT(20)
      CHARACTER*20 NAME1, NAME2
C
      INTEGER INTMAT(MXELE,4),NFACE(MXELE,4),NRFACE(MXELE,4),
     *       IBC(MXPOI),LTYPE(MXELE,4),NGFACE(MXELE,4)
      REAL COORD(MXPOI,3),TEMP(MXPOI),Q(MXPOI),H(MXELE,4),
     *    R(MXELE,4)
C
      REWIND(7)
   10 WRITE(6,20)
   20 FORMAT(/,' PLEASE ENTER THE INPUT FILE NAME:')
      READ(5, '(A)', ERR=10) NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
      M = 0
      CALL FIND(NPOIN,NELEM,IS,IQS,IH,IR,IQ,M,ICONV)
      REWIND(7)
C
      M = M - NPOIN - 1
      IF(M.LE.0) THEN
        WRITE(6,'(A)')  Your label node data has some mistake'
        STOP
      ENDIF
      DO 50 I=1,M
      READ(7,1) TEXT
    1 FORMAT(20A4)
   50 CONTINUE
C
C     READ COORDINATE OF NODES
C
      DO 70 I=1,NPOIN
      READ(7,100) N, IZ, (COORD(N,J), J=1,3), IZ
      IF(I.NE.N) STOP
   70 CONTINUE
  100 FORMAT(I13,I4,T23,I2,3F8.5,T55,I2)
C
C     READ CONNECTIVITY OF ELEMENT
C
      DO 150 I=1,NELEM
      DO 130 J =1,4
      LTYPE(I,J) = 0
  130 CONTINUE
      READ(7,200) N, IZ, (INTMAT(N,J), J=1,4)
      IF(I.NE.N) STOP
  150 CONTINUE
  200 FORMAT(I13,I4,T23,I2,T29,I4,T37,I4,T45,I4,T53,I4)
C
C     READ DATA OF SPECIFIED TEMPERATURE
C
      IF(IS.EQ.0) GO TO 380
      READ(7,250) TK
  250 FORMAT(T18,F10.4)
      DO 350 I=1,IS
      READ(7,300) ISI, TEMP(ISI)
      IBC(ISI) = 1
  300 FORMAT(T21,I4,T33,F8.4)
  350 CONTINUE
  380 IF(IS.EQ.0) THEN
        READ(7,250) TK
        READ(7,1) TEXT
      END IF
      IF(IS.NE.0) READ(7,1) TEXT
C
C     *** READ DATA OF EXTERNAL HEAT GENERATION ***
C
      CALL EXTH(MXELE,NELEM,IQS,INTMAT,NGFACE,LTYPE,QS)
C
C     *** READ DATA OF CONVECTION HEAT ***
C
      CALL CONV(MXELE,NELEM,I...,INTMAT,NFACE,HH,H,LTYPE,TI,ICONV)
C
C     READ DATA OF RADIATION HEAT
C
      CALL RAD(MXELE,NELEM,IR,INTMAT,NRFACE,RR,R,LTYPE)
C
C     READ DATA OF INTERNAL HEAT
C
      QT = 0.0D+00
      IF(IQ.EQ.0) GO TO 800
      DO 500 I=1,IQ
      READ(7,400) QT, IQI
  400 FORMAT(T17,F8.5,T74,I4)
      Q(IQI) = QT
      LTYPE(IQI,1) = 1
  500 CONTINUE
C
  800 CONTINUE
C
C     WRITE OUTPUT FILE
C
  900 WRITE(6,920)
  920 FORMAT(/,' PLEASE ENTER THE OUTPUT FILE NAME:')
      READ(5, '(A)', ERR=10) NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=900)
C
      IDIM = 3
      DEN = 1.0D+00
      SH = 1.0D+00
      ST = 0
      FT = 1
      TS = .5d+00
      LINE = 1
      STEF = 5.67D-08
C
      WRITE(8,*) LINE
      WRITE(8,930) NAME1
  930 FORMAT(' DATA CAME FROM I-DEAS IN COSMIC-NASTRAN FORMAT FILE :',
     *     A20)
      WRITE(6,'(A)')  NPOIN   NELEM'
      WRITE(6,950)  NPOIN, NELEM
  950 FORMAT(T5,I4,TR4,I4)
      WRITE(6,'(A)')   DEN       SH       TK       H       Q
     *    QS       TI'
      WRITE(6,960)DEN,SH,TK,HH,QT,QS,TI
  960 FORMAT(E10.3,1X,5E10.3,1X,F8.3)
      WRITE(6,'(A)')
     * '  EMISSIVITY   STEF-BOLTZMAN     ABSORBTION   INCIDENT-HEAT '
      WRITE(8,990) RR,STEF,DEN,DEN
  990 FORMAT(2X,E10.3,4X,E10.3,8X,E10.3,4X,E10.3)
      WRITE(6,'(A)')   ST     FT     TS'
      WRITE(6,1000)ST,FT,TS
 1000 FORMAT(T2,F6.3,T9,F6.3,T17,F6.3)
      WRITE(8,1100) NPOIN
 1100 FORMAT(' NODAL BOUNDARYCONDITIONS AND COORDINATES [',I6,'] :')
      DO 1300 I=1,NPOIN
      WRITE(8,1200) I, IBC(I), (COORD(I,J), J=1,3), TEMP(I)
 1200 FORMAT(I4,TR4,I2,2X,F10.5,1X,F10.5,1X,F10.5,6X,F8.4)
 1300 CONTINUE
      WRITE(8,1350) NELEM
 1350 FORMAT(' ELEMENT NODAL CONNECTIONS AND HEAT MODES [',I6,'] :')
      DO 1500 I=1,NELEM
      WRITE(8,1400) I, (INTMAT(I,J), J=1,4), (LTYPE(I,J), J=1,4),
     *            (NGFACE(I,J), J=1,4), (NFACE(I,J), J=1,4),
     *            (NRFACE(I,J), J=1,4)
 1400 FORMAT(I4,TR3,I4,1X,I4,1X,I4,1X,I4,TR3,4I2,TR3,4I2,
     *     TR3,4I2,TR3,4I2 )
 1500 CONTINUE
C
      STOP
      END
C
C ---------------------
```

```
C
      SUBROUTINE FINDKNP,NE,IS,IQS,IH,IR,IQ,M,ICONV)
C
      IMPLICIT double precision (A-H,O-Z)
      CHARACTER*7 TEXT,TEXT1*4
C
      I = 0
      IS = 0
      IH = 0
      IR = 0
      IQ = 0
      ICONV = 0
C
   50 READ(7,'(A)') TEXT
      IF(I.EQ.0) M = M + 1
C
C     FIND NUMBER OF NODES
C
      IF((TEXT.EQ.'CTETRA').AND.(I.EQ.0)) THEN
        BACKSPACE(7)
        BACKSPACE(7)
        I = 1
        READ(7,100) NP
  100   FORMAT(T13,I4)
      END IF
C
C     FIND NUMBER OF ELEMENT
C
      IF((TEXT.NE.'CTETRA').AND.(I.EQ.1)) THEN
        BACKSPACE(7)
        BACKSPACE(7)
        I = 2
        READ(7,100) NE
        READ(7,'(A)') TEXT
        READ(7,'(A)') TEXT
C                       Check QVOL only
        READ(7,'(A)') TEXT
        IF(TEXT.EQ.'QVOL') THEN
          I = 6
          ICON = 0
        END IF
        BACKSPACE(7)
        BACKSPACE(7)
        READ(7,'(A)') TEXT
C
      END IF
C
C     FIND NUMBER OF SPECIFIED TEMPERATURE
C
      IF((TEXT.EQ.'SPC').AND.(I.EQ.2)) THEN
  150   IS = IS + 1
        READ(7,'(A)') TEXT
        IF(TEXT.EQ.'SPC') GO TO 150
        I = 3
C                       Check QVOL only
        READ(7,'(A)') TEXT
        IF(TEXT.EQ.'QVOL') THEN
          I = 6
          BACKSPACE(7)
          GO TO 850
        END IF
        BACKSPACE(7)
        BACKSPACE(7)
        READ(7,'(A)') TEXT
C
      ELSE
        IF((TEXT.EQ.'TEMPD').AND.(I.EQ.2)) I = 3
      END IF
C
C     FIND NUMBER OF EXTERNAL HEAT SOURCE
C
      IF(I.EQ.3) THEN
        IQS = -1
  250   READ(7,'(A)') TEXT
        IQS = IQS + 1
        IF(TEXT.EQ.'CHBDY') GO TO 250
        IF(TEXT.EQ.'QVOL') THEN
          I = 6
          BACKSPACE(7)
          GO TO 850
        END IF
        BACKSPACE(7)
        I = 4
      END IF
C
C     FIND NUMBER OF CONVECTION HEAT SURFACE
C
      IF(I.EQ.4) THEN
        ICON = 1
        ICONV = 0
        NCON = 7
        NCON1 = 4
        READ(7,'(A)') TEXT
        IF(TEXT.EQ.'PARAM') RETURN
        READ(7,'(A)') TEXT1
  300   IF(ICON.EQ.1) GO TO 350
        READ(7,'(A)') TEXT
        READ(7,'(A)') TEXT1
  350   IF(TEXT1.NE.'+HA') GO TO 400
        ICON = ICON + 1
        READ(7,'(A)') TEXT
        READ(7,'(A)') TEXT
        READ(7,'(A)') TEXT
        READ(7,'(A)') TEXT
        IF((TEXT.NE.'CHBDY').AND.(TEXT.NE.'SPC')) ICONV = 1
        IF(TEXT.EQ.'CHBDY') THEN
          BACKSPACE(7)
          TEXT = 'SPC'
          ICONV = 1
        END IF
        IF(TEXT.EQ.'SPC') GO TO 300
        DO 380 J=1,5
  380   BACKSPACE(7)
        GO TO 510
  400   IF((TEXT1.EQ.'PHBD').AND.(ICON.EQ.1)) THEN
          I = 5
          GO TO 650
        END IF
        NCON = NCON - ICONV
        DO 500 J=1,NCON
  500   BACKSPACE(7)
  510   I = 5
        READ(7,520) IH
  520   FORMAT(T5,I4)
        NCON1 = NCON1 - ICONV
        DO 600 J=1,NCON1
  600   READ(7,'(A)') TEXT
        IF(TEXT.EQ.'QVOL') THEN
          I = 6
          BACKSPACE(7)
        END IF
      END IF
C
C     FIND NUMBER OF RADIATION HEAT SURFACE
C
  650 IF(I.EQ.5) THEN
        IF(ICON.EQ.1) THEN
          BACKSPACE(7)
          BACKSPACE(7)
        END IF
  700   READ(7,'(A)') TEXT
        IF(TEXT.NE.'CHBDY') GO TO 800
        READ(7,'(A)') TEXT
        IR = IR + 1
        IF(TEXT.EQ.'PHBDY') GO TO 700
  800   I = 6
        IF(TEXT.EQ.'PARAM') RETURN
        BACKSPACE(7)
      END IF
C
C     FIND NUMBER OF INTERNAL HEAT GENERATION
C
  850 IF(I.EQ.6) THEN
        IQ = -1
  900   READ(7,'(A)') TEXT
        IQ = IQ + 1
        IF(TEXT.EQ.'QVOL') GO TO 900
        I = 7
        IF(TEXT.EQ.'PARAM') RETURN
      END IF
C
      IF(TEXT.NE.'ENDDATA') GO TO 50
      RETURN
      END
C
C     ------------------------
C
      SUBROUTINE EXTH(MXELE,NELEM,IQS,INTMAT,NGFACE,LTYPE,QS)
C
      IMPLICIT double precision (A-H,O-Z)
C
      INTEGER INTMAT(MXELE,4),NGFACE(MXELE,4),IEI(3),
     *        LTYPE(MXELE,4)
C
      QS = 0.0D+00
C
C     SET NGFACE TO BE ZERO
C
      DO 100 IE=1,NELEM
      DO 100 J=1,4
      NGFACE(IE,J) = 0
  100 CONTINUE
      IF(IQS.EQ.0) RETURN
C
      DO 700 II=1,IQS
      IEK = 0
      READ(7,400) QS,(IEI(J), J=1,3)
  400 FORMAT(T25,F10.4,T45,I4,TR4,I4,TR4,I4)
      DO 650 I=1,NELEM
      IEL = 0
      DO 520 J=1,4
      DO 520 K=1,3
      IF(INTMAT(I,J).EQ.IEI(K)) THEN
        IEL = IEL + 1
      END IF
  520 CONTINUE
      IF(IEL.EQ.3) THEN
        LTYPE(I,J) = 1
        DO 550 K=1,3
        DO 550 J=1,4
        IF(INTMAT(I,J).EQ.IEI(K)) IEK = IEK + J
  550   CONTINUE
        IF(IEK.EQ.6) THEN
          NGFACE(I,1) = 1
          NI = 1
        END IF
        IF(IEK.EQ.9) THEN
          NGFACE(I,2) = 1
          NI = 2
        END IF
        IF(IEK.EQ.7) THEN
          NGFACE(I,3) = 1
          NI = 3
        END IF
        IF(IEK.EQ.8) THEN
          NGFACE(I,4) = 1
          NI = 4
        END IF
      END IF
      N = I
```

```
      650 CONTINUE
      700 CONTINUE
       1 FORMAT(20A4)
         RETURN
         END
C
C  ---------------------
C
      SUBROUTINE CONV(MXELE,NELEM,IH,INTMAT,NFACE,HH,H,LTYPE,TI,ICONV)
C
      IMPLICIT double precision (A-H,O-Z)
C
      INTEGER INTMAT(MXELE,4),NFACE(MXELE,4),IHI(3),
     *        LTYPE(MXELE,4)
      REAL    HH(MXELE,4)
C
      TI = 0.0D+00
C
C     SET NFACE TO BE ZERO
C
      DO 100 IE=1,NELEM
      DO 100 J=1,4
      NFACE(IE,J) = 0
  100 CONTINUE
      IF(IH.EQ.0) RETURN
C
      DO 700 II=1,IH
      IHK = 0
      READ(7,400) (IHI(J), J=1,3)
  400 FORMAT(T37,I4,T74,I4,T84,I4)
      DO 650 I=1,NELEM
      IHL = 0
      DO 520 J=1,4
      DO 520 K=1,3
      IF(INTMAT(I,J).EQ.IHI(K)) THEN
        IHL = IHL + 1
      END IF
  520 CONTINUE
      IF(IHL.EQ.3) THEN
        LTYPE(I,J) = 1
        DO 550 K=1,3
        DO 550 J=1,4
        IF(INTMAT(I,J).EQ.IHI(K)) IHK = IHK + J
  550   CONTINUE
        IF(IHK.EQ.6) THEN
          NFACE(I,1) = 1
          NI = 1
        END IF
        IF(IHK.EQ.9) THEN
          NFACE(I,2) = 1
          NI = 2
        END IF
        IF(IHK.EQ.7) THEN
          NFACE(I,3) = 1
          NI = 3
        END IF
        IF(IHK.EQ.8) THEN
          NFACE(I,4) = 1
          NI = 4
        END IF
        N = 1
      END IF
  650 CONTINUE
      READ(7,*) TEXT
      READ(7,*) TEXT
      READ(7,680) H(N,NI)
  680 FORMAT(T17,E10.3)
      READ(7,*) TEXT
      IF(I.EQ.1) THEN
        IF(ICONV.EQ.0) READ(7,690) TI
  690   FORMAT(32,F10.4)
      ELSE
        IF(ICONV.EQ.0) READ(7,*) TEXT
      END IF
      HH = H(N,NI)
  700 CONTINUE
    1 FORMAT(20A4)
      RETURN
      END
C
C  ---------------------
C
      SUBROUTINE RAD(MXELE,NELEM,IR,INTMAT,NRFACE,RR,R,LTYPE)
C
      IMPLICIT double precision (A-H,O-Z)
C
      INTEGER INTMAT(MXELE,4),NRFACE(MXELE,4),IRI(3),
     *        LTYPE(MXELE,4)
      REAL    R(MXELE,4)
C
C     SET NFACE TO BE ZERO
C
      DO 100 IE=1,NELEM
      DO 100 J=1,4
      NRFACE(IE,J) = 0
  100 CONTINUE
      IF(IR.EQ.0) RETURN
C
      DO 700 II=1,IR
      READ(7,400) (IRI(J), J=1,3)
  400 FORMAT(T37,I4,T74,I4,T84,I4)
      IRK = 0
      DO 650 I=1,NELEM
      IRL = 0
      DO 520 J=1,4
      DO 520 K=1,3
      IF(INTMAT(I,J).EQ.IRI(K)) THEN
        IRL = IRL + 1
      END IF
  520 CONTINUE
      IF(IRL.EQ.3) THEN
        LTYPE(I,4) = 1
        DO 550 K=1,3
        DO 550 J=1,4
        IF(INTMAT(I,J).EQ.IRI(K)) IRK = IRK + J
  550   CONTINUE
        IF(IRK.EQ.6) THEN
          NRFACE(I,1) = 1
          NI = 1
        END IF
        IF(IRK.EQ.9) THEN
          NRFACE(I,2) = 1
          NI = 2
        END IF
        IF(IRK.EQ.7) THEN
          NRFACE(I,3) = 1
          NI = 3
        END IF
        IF(IRK.EQ.8) THEN
          NRFACE(I,4) = 1
          NI = 4
        END IF
        N = 1
      END IF
  650 CONTINUE
      READ(7,680) R(N,NI)
  680 FORMAT(T31,E10.4)
      RR = R(N,NI)
  700 CONTINUE
      RETURN
      END
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

รายละเอียดของโปรแกรม CNFS

รายละเอียดของโปรแกรม CNFS

โปรแกรม CNFS จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
C              CNFS.FOR
C                                  Update : 30/07/94
C       THIS PROGRAM WILL TRANSFER FORMAT DATA FILE FROM
C       COSMIC-NASTRAN (I-DEAS) TO STRESS3D.FOR
C
      PARAMETER (MXPOI=100,MXELE=300)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  TEXT(20)
      CHARACTER NAME1*20,NAME2*20
C
      INTEGER INTMAT(MXELE,4),IBC(MXPOI,3),MF(MXPOI),NFACE(MXELE,4),
     *        NTI(MXPOI)
      REAL COORD(MXPOI,3),TEMP(MXPOI),FORCE(MXPOI,3),TR(MXELE,4)
C
      REWIND(7)
   10 WRITE(6,20)
   20 FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
      READ(6, '(A)', ERR=10) NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
C     FIND THE NUMBER OF NODE,ELEMENT AND BOUNDARY CONDITION
C
C         IB  is the number of displacement condition, SPC1
C         IBR is the number of displacement condition, SPC
C         IS  is the number of nodal temperature
C         IBF is the number of nodal force
C         NT  is the number of surface traction
C
      M = 0
      CALL FIND( NPOIN, NELEM, M, IB, IBR, IS, IBF, NT )
      REWIND(7)
C
      M = M - NPOIN - 1
      DO 50 I=1,M
      READ(7,1) TEXT
    1 FORMAT(20A4)
   50 CONTINUE
C
C     READ COORDINATE OF NODES
C
      DO 70 I=1,NPOIN
      READ(7,100) N, IZ, (COORD(N,J), J=1,3), IZ
      IF(I.NE.N) STOP
   70 CONTINUE
  100 FORMAT(T11,I6,T23,I2,3F8.5,T55,I2)
C
C     READ CONNECTIVITY OF ELEMENT
C
      READ(7,200) N, IZ, (INTMAT(N,J), J=1,4)
      IF(I.NE.N) STOP
  150 CONTINUE
  200 FORMAT(T11,I6,T23,I2,T29,I4,T37,I4,T45,I4,T53,I4)
C
C                    READ BOUNDARY CONDITION
C
C     NB,NBR i. the type of displacement condition
C           123 : fixed XYZ
C           12  : fixed XY
C           13  : fixed XZ
C           23  : fixed YZ
C     ISB  is the start number of SPC1
C     IFB  is the final number of SPC1
C     ISBR is the number of SPC
C
      DO 220 I=1,NPOIN
      IBC(I,1) = 0
      IBC(I,2) = 0
      IBC(I,3) = 0
  220 CONTINUE
      READ(7,230) ELAS,PR,ALPHA
  230 FORMAT(T17,F8.2,T34,F8.6,T49,F8.2)
      READ(7,*)
      IF(IB.GT.0) THEN
        DO 300 I=1,IB
          READ(7,250) NB,ISB,IFB
  250     FORMAT(T21,I4,4X,I4,12X,I4)
          DO 275 J=ISB,IFB
          CALL BOUD(MXPOI, J, NB, IBC)
  275     CONTINUE
  300   CONTINUE
      END IF
      IF(IBR.GT.0) THEN
        DO 400 I=1,IBR
        READ(7,350) ISBR,NBR
  350   FORMAT(T21,I4,T29,I4)
        CALL BOUD(MXPOI, ISBR, NBR, IBC)
  400   CONTINUE
      END IF
C
C     READ THE NODAL TEMPERATURE
C
      IF(IS.GT.0) THEN
        DO 430 J=1,NPOIN
        TEMP(J) = 0.
  430   CONTINUE
        DO 500 I=1,IS
        READ(7,450) IS1,S1, IS2,S2, IS3,S3
  450   FORMAT(T21,I4,F10.3,2X,I4,F10.3,2X,I4,F10.3)
        IF (IS1.GT.0) TEMP(IS1) = S1
        IF (IS2.GT.0) TEMP(IS2) = S2
        IF (IS3.GT.0) TEMP(IS3) = S3
  500   CONTINUE
      END IF
      READ(7,550) TREF
  550 FORMAT(T17,F12.4)
C
C     READ THE NODAL FORCE
C
      IF(IBF.GT.0) THEN
        DO 580 J=1,NPOIN
        MF(J) = 0
        FORCE(J,1) = 0.
        FORCE(J,2) = 0.
        FORCE(J,3) = 0.
  580   CONTINUE
        DO 650 I=1,IBF
        READ(7,600) NF,FX,FY,FZ
  600   FORMAT(T21,I4,T41,3F8.3)
        MF(NF) = 1
        FORCE(NF,1) = FX
        FORCE(NF,2) = FY
        FORCE(NF,3) = FZ
  650   CONTINUE
      END IF
C
C       FIND THE SURFACE TRACTION
C
C       TR(I) is the value of traction
C       NFACE(I,J) is the face side of traction
C
C       SET NFACE TO BE ZERO
C
      DO 700 IE=1,NELEM
      DO 700 J=1,4
      TR(I,J) = 0.
      NFACE(IE,J) = 0
  700 CONTINUE
C
      IF(NT.GT.0) THEN
        DO 800 II=1,NT
        NTK = 0
        READ(7,710) TRAC, (NTI(J), J=1,3)
  710   FORMAT(T17,F10.3,T29,I4,T84,I4,T84,I4)
        DO 750 I=1,NELEM
        NTL = 0
        DO 730 J=1,4
        DO 730 K=1,3
        IF(INTMAT(I,J).EQ.NTI(K)) THEN
          NTL = NTL + 1
```

```
      END IF
  730 CONTINUE
      IF(NTK.EQ.3) THEN
        DO 740 K=1,3
        DO 740 J=1,4
        IF(NTMAT(I,J).EQ.NTK(K)) NTK = NTK + J
  740   CONTINUE
        IF(NTK.EQ.6) THEN
          NFACE(I,1) = 1
          NI = 1
        END IF
        IF(NTK.EQ.5) THEN
          NFACE(I,2) = 1
          NI = 2
        END IF
        IF(NTK.EQ.7) THEN
          NFACE(I,3) = 1
          NI =3
        END IF
        IF(NTK.EQ.8) THEN
          NFACE(I,4) = 1
          NI = 4
        END IF
        write(5,*) TRAC
        TR(I,NI) = TRAC
      END IF
  750 CONTINUE
  800 CONTINUE
    END IF
C
C     ITYPE CODES ARE :
C           0 : ONLY NODAL FORCE
C           1 : NODAL FORCE & NODAL TEMPERATURE
C           2 : NODAL FORCE & SURFACE TRACTION
C           3 : NODAL TEMPERATURE & SURFACE TRACTION
C
    IF((IS.GT.0).AND.(NT.GT.0)) ITYPE = 3
    IF((IS.EQ.0).AND.(NT.GT.0)) ITYPE = 2
    IF((IS.GT.0).AND.(NT.EQ.0)) ITYPE = 1
    IF((IS.EQ.0).AND.(NT.EQ.0)) ITYPE = 0
C
C                                   PRINT OUTPUT FILE
C
  810 WRITE(5,820)
  820 FORMAT(/,' PLEASE ENTER THE OUTPUT FILE NAME:')
    READ(5,'(A)', ERR=810) NAME2
    OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=810)
C
    IDIM = 3
    LINE = 1
    I1 = 0
    IF(IBF.EQ.0) I1 = 1
    IF(IBF.EQ.0) IBF = 1
    WRITE(8,*) LINE
    WRITE(8,'(A)') ' DATA FROM I-DEAS IN COSMIC-NASTRAN FORMAT'
    WRITE(8,'(A)')  NPOIN  NELEM  NFORCE  ITYPE'
    WRITE(8,950) NPOIN, NELEM, IBF, ITYPE
  950 FORMAT(4I4,I6,3I2,I6,3I5,I4,3I4)
    WRITE(8,'(A)') '   ELAS      PR       ALPHA       TREF'
    WRITE(8,980) ELAS, PR, ALPHA, TREF
  980 FORMAT(E12.5,2X,F5.3,2X,E12.4,2X,F10.3)
    WRITE(8,'(A)') 'NODAL BOUNDARYCONDITIONS AND COORDINATES :'
    DO 1300 I=1,NPOIN
    WRITE(8,1200) I, (IBC(I,J),J=1,3), (COORD(I,K), K=1,3), TEMP(I)
 1200 FORMAT(I6,3I2,3X,F10.5,1X,F10.5,1X,F10.5,6X,F10.4)
 1300 CONTINUE
    WRITE(8,*)' ELEMENT NODAL CONNECTIONS AND FORCE MODES :'
    DO 1500 I=1,NELEM
    WRITE(8,1400) I,  (NTMAT(I,K), K=1,4), (NFACE(I,J), J=1,4),
   *            (TR(I,L), L=1,4)
 1400 FORMAT(I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,I6,1X,4I2,1X,E10.3,1X,E9.3,
   *       1X,E9.3,1X,E9.3)
 1500 CONTINUE
    WRITE(8,*)'   NODAL FORCES : '
    IF(I1.EQ.0) THEN
      DO 1700 I=1,NPOIN
      IF(IMF(I).GT.0) WRITE(8,1600) I, FORCE(I,1), FORCE(I,2),
   *                        FORCE(I,3)
 1600   FORMAT(I6,4X,F12.4,1X,F12.4,1X,F12.4)
 1700   CONTINUE
    ELSE
      WRITE(8,*) ' 1    0.    0.    0.'
    END IF
    STOP
    END
C
C ---------------------
C
    SUBROUTINE BOUD( MXPOI, N, NB, IBC)
C
    INTEGER IBC(MXPOI,3)
C
    IF(NB.EQ.1) IBC(N,1) = 1
    IF(NB.EQ.2) IBC(N,2) = 1
    IF(NB.EQ.3) IBC(N,3) = 1
    IF(NB.EQ.12) THEN
      IBC(N,1) = 1
      IBC(N,2) = 1
    END IF
    IF(NB.EQ.13) THEN
      IBC(N,1) = 1
      IBC(N,3) = 1
    END IF
    IF(NB.EQ.23) THEN
      IBC(N,2) = 1
      IBC(N,3) = 1
    END IF
    IF(NB.EQ.123) THEN
      IBC(N,1) = 1
      IBC(N,2) = 1
      IBC(N,3) = 1
    END IF
    RETURN
    END
C
C ---------------------
C
    SUBROUTINE FIND( NP, NE, M, IB, IBR, IS, IBF, NT)
C
    CHARACTER*7 TEXT
C
    I = 0
    IS = 0
    IB = 0
    IBR = 0
    IBF = 0
    NT = 0
C
  50 READ(7,'(A)') TEXT
    IF(I.EQ.0) M = M + 1
C
C   FIND NUMBER OF NODES
C
    IF((TEXT.EQ.'CTETRA').AND.(I.EQ.0)) THEN
      BACKSPACE(7)
      BACKSPACE(7)
      I = 1
      READ(7,100) NP
  100   FORMAT(T11,I6)
    END IF
C
C   FIND NUMBER OF ELEMENTS
C
    IF((TEXT.NE.'CTETRA').AND.(I.EQ.1)) THEN
      BACKSPACE(7)
      BACKSPACE(7)
      I = 2
      READ(7,100) NE
      READ(7,'(A)') TEXT
      READ(7,'(A)') TEXT
C                   Check NODAL TEMPERATURE,NODAL FORCE
      READ(7,'(A)') TEXT
      IF(TEXT.EQ.'TEMP') THEN
        I = 3
      ENDIF
      IF(TEXT.EQ.'TEMPD') I = 4
      BACKSPACE(7)
      READ(7,'(A)') TEXT
C
    END IF
C
C                   FIND NUMBER OF RESTRAIN
C
    IF((TEXT.EQ.'SPC1').OR.(TEXT.EQ.'SPC')) THEN
      IF((TEXT.EQ.'SPC1').AND.(I.EQ.2)) THEN
  140   IB = IB + 1
        READ(7,'(A)') TEXT
        IF(TEXT.EQ.'SPC1') GO TO 140
        IF(TEXT.EQ.'TEMP') I = 3
        IF(TEXT.EQ.'SPC') BACKSPACE(7)
      ELSE
        IF(I.EQ.2) THEN
  150     READ(7,'(A)') TEXT
          IBR = IBR + 1
          IF(TEXT.EQ.'SPC') GO TO 150
          IF(TEXT.EQ.'TEMP') I = 3
        END IF
      END IF
      IF(TEXT.EQ.'TEMPD') I = 4
    END IF
C
C                   FIND NUMBER OF SPECIFICED HEAT
C
    IF(I.EQ.3) THEN
  250   READ(7,'(A)') TEXT
      IS = IS + 1
      IF(TEXT.EQ.'TEMP') GO TO 250
      I = 4
    END IF
C
C                   FIND NUMBER OF NODAL FORCE
C
  300 IF(I.EQ.4) THEN
      READ(7,'(A)') TEXT
      IF(TEXT.EQ.'PLOAD') GO TO 500
  400   IF(TEXT.EQ.'PARAM') RETURN
      READ(7,'(A)') TEXT
      IBF = IBF + 1
      IF(TEXT.EQ.'FORCE') GO TO 400
  500   I = 5
    END IF
C
C     FIND THE SURFACE TRACTION
C
    IF(I.EQ.5) THEN
      IF(TEXT.EQ.'PARAM') RETURN
      IF(TEXT.EQ.'ENDDATA') RETURN
  600   READ(7,'(A)') TEXT
      NT = NT + 1
      IF(TEXT.EQ.'PLOAD') GO TO 600
    END IF
C
    IF(TEXT.NE.'ENDDATA') GO TO 50
    RETURN
    END
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้วิจัย

วิทยานิพนธ์ฉบับนี้จัดทำโดย นาย วิโรจน์ แซ่ลิ่ม เกิดเมื่อวันที่ 10 เดือนเมษายน พุทธศักราช 2511 ที่ อำเภอหาดใหญ่ จังหวัดสงขลา สำเร็จการศึกษา ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จากมหาวิทยาลัยเชียงใหม่ เมื่อปีการศึกษา 2532 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2535

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย