รายการอ้างอิง

ภาษาไทย

ปราโมทย์ เดชะอำไพ. ไฟไนต์เอลิเมนต์ในงานวิศวกรรม. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2537.

ปราโมทย์ เดชะอำไพ. ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม. กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ภาษาอังกฤษ

Boley, B. A., and Weiner, J. H. Theory of Thermal Stresses. New York : John Wiley & Sons, 1960.

Chandrupatla, T. R., and Belegundu, A. D. Introduction to Finite Elements in Engineering. New Jersey : Prentice-Hall, 1991.

Church, E. F. Steam Turbines. 3rd Ed. New York : McGraw-Hill, 1950.

Cohen, H., Rogers G. F. C., and Saravanamuttoo H. I. H. Gas Turbine Theory. 3rd Ed. New York : John Wiley & Sons, 1987.

Cook, R. D., Malkus, D. S., and Plesha, M. E. Concepts and Applications of Finite Element Analysis. 3rd Ed. New York : John Wiley & Sons, 1989.

Hildeband, F. B. Advanced Calculus for Applications. 2nd Ed. New Jersey : Prentice-Hall, 1976.

Hoffman, O., and Sachs G. Introduction to Theory of Plasticity for Engineer. New York : McGraw-Hill, 1953.

Huebner, K. H., and Thornton, E. A. The Finite Element Method for Engineers. 2nd Ed. New York : John Wiley & Sons, 1982.

Incropera, F. P., and Dewitt, D. P. Introduction to Heat Transfer, 2nd Ed. New York : John Wiley & Sons, 1985.

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

Logan, D. L. A First Course in the Finite Element Method. Boston : PWS Publishing Company, 1993.

Nyhoff, L., and Leestma, S. FORTRAN77 and Numerical Methods for Engineers and Scientists. New Jersey : Prentice-Hall, 1995.

Timoshenko, S. P., and Goodier, J. N. Theory of Elasticity, 3rd Ed. Singapore : McGraw-Hill International Edition, , 1970.

Ugural, A. C., and Fenster, S. K.. Advanced Strength and Applied Elasticity. New York : Elsevier, 1977.

ภาคผนวก

ภาคผนวก ก

รายละเอียดของโปรแกรม DISEXACT

<u>รายละเอียดของโปรแกรม</u> DISEXACT

โปรแกรม DISEXACT จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
***********************************************************************
c A FINITE ELEMENT PROGRAM
c FOR SOLVING 1-D ANNULAR DISK WITH RADIAL TEMPERATURE VARIATION
c FOR  T(r) = a+b*ln(r) WITH EXACT U(r) INTERPOLATION FUNCTION
c
c       MXELE = maximum number of elements in model (can be adjusted)
c       MXPOI = maximum number of nodes in model (can be adjusted)
c
***********************************************************************
      PARAMETER (MXELE=100,MXPOI=101)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(MXPOI),STR(MXPOI),UR(MXPOI),T(MXPOI)
      DIMENSION SYSK(MXPOI,MXPOI),SYSA(MXPOI),BNEW(MXPOI),R(MXPOI)
      CHARACTER*20 NAME1,NAME2
      INTEGER IBC(MXPOI)
c
   10 WRITE(6,20)
   20 FORMAT(/,'**FINITE ELEMENT FOR SOLVING ANNULAR DISK WITH RADIAL
     *TEMP VARIATION PROBLEM**',/,'  WITH T(r) = a+b*ln(r) ',//,
     *'  PLEASE INPUT THE FILE NAME OF INPUT DATA:',/)
      READ(5, '(A)',ERR=10) NAME1
      OPEN(UNIT=7,FILE=NAME1,STATUS='OLD',ERR=10)
c
c read title of computation:
      READ(7,*) NLINES
      DO 100 ILINES = 1,NLINES
         READ(7,1) TEXT
    1 FORMAT(20A4)
  100 CONTINUE
c
c ---------------------- read input data ----------------------------
      READ(7,1) TEXT
      READ(7,*) NE,NP,NFIX
      IF(NE.GT.MXELE) WRITE(6,110)NE
  110 FORMAT(/,'PLEASE INCREASE MXELE TO',I5)
      IF(NE.GT.MXELE) STOP
      IF(NP.GT.MXPOI) WRITE(6,120)NP
  120 FORMAT(/,'PLEASE INCREASE MXPOI TO',I5)
      IF(NP.GT.MXPOI) STOP
      READ(7,1) TEXT
      READ(7,*) AL,E,PR,TREF,PI,PO,TI,TO
c read the boundary condition (if IBC=0 U(I) not )
c                             (if IBC=1 U(I) fix )
c set matric to zero
      DO 130 I=1,NP
         STR(I)=0
        U(I)=0
        IBC(I)=0
  130 CONTINUE
c
c
      READ(7,1) TEXT
      DO 131 IB = 1,NP
         READ(7,*) IP,R(IB)
```

```
       IF(IP.NE.IB) WRITE(6,132) IB
 132   FORMAT(/,'**ERROR**  NODE',I5,'IN DATA FILE WAS MISSING')
      STR(1)   = PI
      STR(NP)  = PO
 131 CONTINUE
     DO 135 I=1,NP
        T(I)=LOG(R(I)/R(NP))/LOG(R(1)/R(NP))*TI+LOG(R(1)/R(I))/
    *LOG(R(1)/R(NP))*TO
 135 CONTINUE
     IF(NFIX.EQ.0) GO TO 133
     READ(7,1) TEXT
     DO 134 IP=1,NFIX
        READ(7,*) IFIX
        IBC(IFIX)=1
 134 CONTINUE
c
c ---------------------------- processing ----------------------------
c
 133 WRITE(6,200) NP,NE
 200 FORMAT(/,'**THE MODEL CONSISTS OF',I5,' NODES',I5,' ELEMENTS**')
c
c establish all element matrics and assemble them to form system equation
c
     WRITE(6,210)
 210 FORMAT(/,'**ESTABLISHING ELEMENT MATRICS AND'
    *,/,'  ASSEMBLE THEM TO SYSTEM EQUATION**')
     CALL ELE(NE,NP,MXPOI,PR,AL,STR,SYSK,SYSA,T,R,TREF,E)
c
c apply boundary condition and set them to be new matric to solve
c
     WRITE(6,220)
 220 FORMAT(/,'**APPLYING BOUNDARY CONDITION OF NODAL ',/,
    *' AND FORMING SET OF NEW MATRICS TO BE SOLVED**')
     CALL APPLYBC(IBC,NP,MXPOI,U,SYSA,SYSK,BNEW)
c
c solve with gauss
c
     WRITE(6,230)
 230 FORMAT(/,'** SOLVING A SET OF SIMULTANEOUS EQUATIONS **')
     WRITE(6,231) NP
 231 FORMAT(5X,'TOTAL OF',I5,' EQUATIONS TO BE SOLVED')
     CALL GAUSS(NP,SYSK,BNEW,UR,MXPOI)
c
c
c --------------------- print out nodal solution ---------------------
c
 300 WRITE(6,310)
 310 FORMAT(/,' PLEASE ENTER FILE NAME FOR DISP. SOLUTION:'
    *,/)
     READ(5 ,'(A)',ERR=300) NAME2
     OPEN(UNIT=8,FILE=NAME2,STATUS='NEW',ERR=300)
     WRITE(8,320) NP
 320 FORMAT('NODAL SOLUTION[',I5,']',//,2X,'NODE',10X,'U',/)
     DO 330 IP=1,NP
        WRITE(8,340) IP,UR(IP)
 340    FORMAT(I6,2E14.6)
 330 CONTINUE
     STOP
     END
************************************************************************
c
      SUBROUTINE ELE(NE,NP,MXPOI,PR,AL,STR,SYSK,SYSA,T,R,TREF,E)
c
c establish all element matrics and assemble them to system equation
c
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AK(2,2), AE(2), STR(MXPOI), T(MXPOI),R(MXPOI)
      DIMENSION SYSK(MXPOI,MXPOI),SYSA(MXPOI)
c
```

```
c set value in matric sys to zero
      DO 110 IR = 1,NP
         DO 120 IC = 1,NP
            SYSK(IR,IC) = 0
            SYSA(IR) = 0
  120 CONTINUE
  110 CONTINUE
c
c loop over the number of elements
      DO 100 IE = 1,NE
         R1 = R(IE)
         R2 = R(IE+1)
         T1 = T(IE)
         T2 = T(IE+1)
          C = (T1-T2)/log(R1/R2)
c
c find element matric [K]e
c
         SP = 1/(1-PR**2)
         SR = (R2**2-R1**2)
c
         AK(1,1) = SP*(2*R1/SR+(1-PR)/R1)
         AK(1,2) = SP*(-2*R2/SR)
         AK(2,1) = SP*(-2*R1/SR)
         AK(2,2) = SP*(2*R2/SR-(1-PR)/R2)
c
         I1=IE
         I2=IE+1
         AE(1) = AL*C/(2*(1-PR)*SR)*(2*R2**2*log(R1/R2)+SR)-STR(I1)/E
     *-AL/(1-PR)*(T1-TREF)
         AE(2) = -AL*C/(2*(1-PR)*SR)*(2*R1**2*log(R1/R2)+SR)+STR(I2)/E
     *+AL/(1-PR)*(T2-TREF)
c
c assemble these element equations
c
         CALL ASSMBLE(IE,AK,AE,MXPOI,SYSK,SYSA)
c
  100 CONTINUE
      RETURN
      END
**********************************************************************
      SUBROUTINE ASSMBLE(IE,AK,AE,MXPOI,SYSK,SYSA)
c
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AK(2,2), AE(2)
      DIMENSION SYSK(MXPOI,MXPOI) ,SYSA(MXPOI)
c
      NNODE = 2
      DO 1000 IR = 1,NNODE
         DO 2000 IC =1,NNODE
            IROW = IE+(IR-1)
            ICOL = IE+(IC-1)
            SYSK(IROW,ICOL) = SYSK(IROW,ICOL)+AK(IR,IC)
 2000    CONTINUE
         SYSA(IROW) = SYSA(IROW)+AE(IR)
 1000 CONTINUE
      RETURN
      END
**********************************************************************
      SUBROUTINE APPLYBC(IBC,NP,MXPOI,U,SYSA,SYSK,BNEW)
c
c try to set new matric to be solved
c
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(MXPOI),BNEW(MXPOI)
      DIMENSION SYSK(MXPOI,MXPOI) ,SYSA(MXPOI)
      INTEGER IBC(MXPOI)
c
c change only IBC = 1 (known value of u)
```

```
c
      DO 100 IR = 1,NP
         IF(IBC(IR).EQ.0) GO TO 100
         DO 200 IC =1,NP
            SYSK(IR,IC) = 0
  200    CONTINUE
         BNEW(IR) = U(IR)
         SYSK(IR,IR) = 1
  100 CONTINUE
c
c for only IBC = 0 (known value of stress)
c
      DO 300 IR = 1,NP
         IF(IBC(IR).EQ.1) GO TO 300
         BNEW(IR) = SYSA(IR)
         DO 400 IC = 1,NP
            IF(IBC(IC).EQ.0) GO TO 400
            BNEW(IR) = BNEW(IR)-SYSK(IR,IC)*U(IC)
            SYSK(IR,IC) = 0
  400    CONTINUE
  300 CONTINUE
      RETURN
      END
************************************************************************
C
      SUBROUTINE GAUSS(N, A, B, X, MXPOI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI), X(MXPOI)
C
C     PERFORM SCALING:
C
      CALL SCALE(N, A, B, MXPOI)
C
C     FORWARD ELIMINATION:
C
C     PERFORM ACCORDING TO ORDER OF 'PRIME' FROM 1 TO N-1:
C
      DO 100  IP=1,N-1
C
C     PERFORM PARTIAL PIVOTING:
C
      CALL PIVOT(N, A, B, MXPOI, IP)
C
C     LOOP OVER EACH EQUATION STARTING FROM THE ONE THAT CORRESPONDS
C     WITH THE ORDER OF 'PRIME' PLUS ONE:
C
      DO 200  IE=IP+1,N
      RATIO = A(IE,IP)/A(IP,IP)
C
C     COMPUTE NEW COEFFICIENTS OF THE EQUATION CONSIDERED:
C
      DO 300  IC=IP+1,N
      A(IE,IC) = A(IE,IC) - RATIO*A(IP,IC)
  300 CONTINUE
      B(IE) = B(IE) - RATIO*B(IP)
  200 CONTINUE
C
C     SET COEFFICIENTS ON LOWER LEFT PORTION TO ZERO:
C
      DO 400  IE=IP+1,N
      A(IE,IP) = 0.
  400 CONTINUE
  100 CONTINUE
C
C     BACK SUBSTITUTION:
C
C     COMPUTE SOLUTION OF THE LAST EQUATION:
C
      X(N) = B(N)/A(N,N)
```

```
C
C     THEN COMPUTE SOLUTIONS FROM EQUATION N-1 TO 1:
C
      DO 500  IE=N-1,1,-1
      SUM = 0.
      DO 600  IC=IE+1,N
      SUM = SUM + A(IE,IC)*X(IC)
  600 CONTINUE
      X(IE) = (B(IE) - SUM)/A(IE,IE)
  500 CONTINUE
      RETURN
      END
C
C--------------------------------------------------------------
C
      SUBROUTINE PIVOT(N, A, B, MXPOI, IP)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
C     PERFORM PARTIAL PIVOTING:
C
      JP = IP
      BIG = ABS(A(IP,IP))
      DO 10  I=IP+1,N
      AMAX = ABS(A(I,IP))
      IF(AMAX.GT.BIG)  THEN
         BIG = AMAX
         JP  = I
      ENDIF
   10 CONTINUE
         IF(JP.NE.IP)  THEN
      DO 20  J=IP,N
      DUMY    = A(JP,J)
      A(JP,J) = A(IP,J)
      A(IP,J) = DUMY
   20 CONTINUE
      DUMY  = B(JP)
      B(JP) = B(IP)
      B(IP) = DUMY
         ENDIF
      RETURN
      END
C
C--------------------------------------------------------------
C
      SUBROUTINE SCALE(N, A, B, MXPOI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
C     PERFORM SCALING:
C
      DO 10  IE=1,N
      BIG = ABS(A(IE,1))
      DO 20  IC=2,N
      AMAX = ABS(A(IE,IC))
      IF(AMAX.GT.BIG)  BIG = AMAX
   20 CONTINUE
      DO 30  IC=1,N
      A(IE,IC) = A(IE,IC)/BIG
   30 CONTINUE
      B(IE) = B(IE)/BIG
   10 CONTINUE
      RETURN
      END
C
C--------------------------------------------------------------
*****************************************************************
```

รายละเอียดของโปรแกรม DISLNEAR

รายละเอียดของโปรแกรม DISLNEAR

โปรแกรม DISLNEAR จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่างๆทั้งหมดดังนี้

```
***********************************************************************
c A FINITE ELEMENT PROGRAM
c FOR SOLVING 1-D ANNULAR DISK WITH RADIAL TEMPERATURE VARIATION
c FOR  OVERALL T(r) = a+b*ln(r)
c WITH LINEAR U(r) AND T(r) INTERPOLATION FUNCTION IN EACH ELEMENT
c
c        MXELE = maximum number of elements in model (can be adjusted)
c        MXPOI = maximum number of nodes in model (can be adjusted)
c
***********************************************************************
      PARAMETER (MXELE=100,MXPOI=101)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(MXPOI),STR(MXPOI),UR(MXPOI),T(MXPOI)
      DIMENSION SYSK(MXPOI,MXPOI),SYSA(MXPOI),BNEW(MXPOI),R(MXPOI)
      CHARACTER*20 NAME1,NAME2
      INTEGER IBC(MXPOI)
c
   10 WRITE(6,20)
   20 FORMAT(/,'**FINITE ELEMENT FOR SOLVING ANNULAR DISK WITH RADIAL
     *TEMP VARIATION PROBLEM**',/,'  WITH LINEAR TEMP AND DISP INTERPOLA
     *TION ',//,'  PLEASE INPUT THE FILE NAME OF INPUT DATA:',/)
      READ(5, '(A)',ERR=10) NAME1
      OPEN(UNIT=7,FILE=NAME1,STATUS='OLD',ERR=10)
c
c read title of computation:
      READ(7,*) NLINES
      DO 100 ILINES = 1,NLINES
         READ(7,1) TEXT
    1 FORMAT(20A4)
  100 CONTINUE
c
c ---------------------- read input data ----------------------------
      READ(7,1) TEXT
      READ(7,*) NE,NP,NFIX
      IF(NE.GT.MXELE) WRITE(6,110)NE
  110 FORMAT(/,'PLEASE INCREASE MXELE TO',I5)
      IF(NE.GT.MXELE) STOP
      IF(NP.GT.MXPOI) WRITE(6,120)NP
  120 FORMAT(/,'PLEASE INCREASE MXPOI TO',I5)
      IF(NP.GT.MXPOI) STOP
      READ(7,1) TEXT
      READ(7,*) AL,E,PR,TREF,PI,PO,TI,TO
c read the boundary condition (if IBC=0 U(I) not )
c                             (if IBC=1 U(I) fix )
c set matric to zero
      DO 130 I=1,NP
         STR(I)=0
        U(I)=0
        IBC(I)=0
  130 CONTINUE
c
c
      READ(7,1) TEXT
      DO 131 IB = 1,NP
         READ(7,*) IP,R(IB)
         IF(IP.NE.IB) WRITE(6,132) IB
```

```
 132     FORMAT(/,'**ERROR**  NODE',I5,'IN DATA FILE WAS MISSING')
         STR(1)   = PI
         STR(NP)  = PO
 131 CONTINUE
     DO 135 I=1,NP
        T(I)=LOG(R(I)/R(NP))/LOG(R(1)/R(NP))*TI+LOG(R(1)/R(I))/
    *LOG(R(1)/R(NP))*TO
 135 CONTINUE
     IF(NFIX.EQ.0) GO TO 133
     READ(7,1) TEXT
     DO 134 IF=1,NFIX
        READ(7,*) IFIX
        IBC(IFIX)=1
 134 CONTINUE
c
c --------------------------- processing ---------------------------
c
 133 WRITE(6,200) NP,NE
 200 FORMAT(/,'**THE MODEL CONSISTS OF',I5,' NODES',I5,' ELEMENTS**')
c
c establish all element matrics and assemble them to form system equation
c
     WRITE(6,210)
 210 FORMAT(/,'**ESTABLISHING ELEMENT MATRICS AND'
    *,/,'  ASSEMBLE THEM TO SYSTEM EQUATION**')
     CALL ELE(NE,NP,MXPOI,PR,AL,STR,SYSK,SYSA,T,R,TREF,E)
c
c apply boundary condition and set them to be new matric to solve
c
     WRITE(6,220)
 220 FORMAT(/,'**APPLYING BOUNDARY CONDITION OF NODAL ',/,
    *' AND FORMING SET OF NEW MATRICS TO BE SOLVED**')
     CALL APPLYBC(IBC,NP,MXPOI,U,SYSA,SYSK,BNEW)
c
c solve with gauss
c
     WRITE(6,230)
 230 FORMAT(/,'** SOLVING A SET OF SIMULTANEOUS EQUATIONS **')
     WRITE(6,231) NP
 231 FORMAT(5X,'TOTAL OF',I5,' EQUATIONS TO BE SOLVED')
     CALL GAUSS(NP,SYSK,BNEW,UR,MXPOI)
c
c
c ---------------------- print out nodal solution ---------------------
c
 300 WRITE(6,310)
 310 FORMAT(/,' PLEASE ENTER FILE NAME FOR DISP. SOLUTION:'
    *,/)
     READ(5 ,'(A)',ERR=300) NAME2
     OPEN(UNIT=8,FILE=NAME2,STATUS='NEW',ERR=300)
     WRITE(8,320) NP
 320 FORMAT('NODAL SOLUTION[',I5,']',//,2X,'NODE',10X,'U',/)
     DO 330 IP=1,NP
        WRITE(8,340) IP,UR(IP)
 340    FORMAT(I6,2E14.6)
 330 CONTINUE
     STOP
     END
***********************************************************************
c
     SUBROUTINE ELE(NE,NP,MXPOI,PR,AL,STR,SYSK,SYSA,T,R,TREF,E)
c
c establish all element matrics and assemble them to system equation
c
     IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION AK(2,2), AE(2), STR(MXPOI), T(MXPOI),R(MXPOI)
     DIMENSION SYSK(MXPOI,MXPOI),SYSA(MXPOI)
c
c set value in matric sys to zero
```

```
      DO 110 IR = 1,NP
         DO 120 IC = 1,NP
            SYSK(IR,IC) = 0
            SYSA(IR) = 0
  120 CONTINUE
  110 CONTINUE
c
c loop over the number of elements
      DO 100 IE = 1,NE
         R1 = R(IE)
         R2 = R(IE+1)
         T1 = T(IE)
         T2 = T(IE+1)
c
c find element matric [K]e
c
         SP = 1./(1-PR**2)
         SR = (R2-R1)**2
c
         AK(1,1) = SP*((2.*(R2-R1)-R2*LOG(R2/R1))/SR+(1-PR)/R1)
         AK(1,2) = SP*((2.*(R1-R2)-R1*LOG(R1/R2))/SR)
         AK(2,1) = SP*((2.*(R1-R2)+R2*LOG(R2/R1))/SR)
         AK(2,2) = SP*((2.*(R2-R1)+R1*LOG(R1/R2))/SR-(1-PR)/R2)
c
         I1=IE
         I2=IE+1
         AE(1) = -AL*((T1-TREF)+(T2-TREF))/(2.*(1-PR))-STR(I1)/E
         AE(2) = AL*((T1-TREF)+(T2-TREF))/(2.*(1-PR))+STR(I2)/E
c
c assemble these element equations
c
         CALL ASSEMBLE(IE,AK,AE,MXPOI,SYSK,SYSA)
c
  100 CONTINUE
      RETURN
      END
*************************************************************************
      SUBROUTINE ASSEMBLE(IE,AK,AE,MXPOI,SYSK,SYSA)
c
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AK(2,2), AE(2)
      DIMENSION SYSK(MXPOI,MXPOI) ,SYSA(MXPOI)
c
      NNODE = 2
      DO 1000 IR = 1,NNODE
         DO 2000 IC =1,NNODE
            IROW = IE+(IR-1)
            ICOL = IE+(IC-1)
            SYSK(IROW,ICOL) = SYSK(IROW,ICOL)+AK(IR,IC)
 2000    CONTINUE
         SYSA(IROW) = SYSA(IROW)+AE(IR)
 1000 CONTINUE
      RETURN
      END
*************************************************************************
      SUBROUTINE APPLYBC(IBC,NP,MXPOI,U,SYSA,SYSK,BNEW)
c
c try to set new matric to be solved
c
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION U(MXPOI),BNEW(MXPOI)
      DIMENSION SYSK(MXPOI,MXPOI) ,SYSA(MXPOI)
      INTEGER IBC(MXPOI)
c
c change only IBC = 1 (known value of u)
c
      DO 100 IR = 1,NP
         IF(IBC(IR).EQ.0) GO TO 100
         DO 200 IC =1,NP
```

```
            SYSK(IR,IC) = 0
  200    CONTINUE
         BNEW(IR) = U(IR)
         SYSK(IR,IR) = 1
  100 CONTINUE
c
c for only IBC = 0 (known value of stress)
c
      DO 300 IR = 1,NP
         IF(IBC(IR).EQ.1) GO TO 300
         BNEW(IR) = SYSA(IR)
         DO 400 IC = 1,NP
            IF(IBC(IC).EQ.0) GO TO 400
            BNEW(IR) = BNEW(IR)-SYSK(IR,IC)*U(IC)
            SYSK(IR,IC) = 0
  400    CONTINUE
  300 CONTINUE
      RETURN
      END
************************************************************************
C
      SUBROUTINE GAUSS(N, A, B, X, MXPOI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI), X(MXPOI)
C
C     PERFORM SCALING:
C
      CALL SCALE(N, A, B, MXPOI)
C
C     FORWARD ELIMINATION:
C
C     PERFORM ACCORDING TO ORDER OF 'PRIME' FROM 1 TO N-1:
C
      DO 100  IP=1,N-1
C
C     PERFORM PARTIAL PIVOTING:
C
      CALL PIVOT(N, A, B, MXPOI, IP)
C
C     LOOP OVER EACH EQUATION STARTING FROM THE ONE THAT CORRESPONDS
C     WITH THE ORDER OF 'PRIME' PLUS ONE:
C
      DO 200  IE=IP+1,N
      RATIO = A(IE,IP)/A(IP,IP)
C
C     COMPUTE NEW COEFFICIENTS OF THE EQUATION CONSIDERED:
C
      DO 300  IC=IP+1,N
      A(IE,IC) = A(IE,IC) - RATIO*A(IP,IC)
  300 CONTINUE
      B(IE) = B(IE) - RATIO*B(IP)
  200 CONTINUE
C
C     SET COEFFICIENTS ON LOWER LEFT PORTION TO ZERO:
C
      DO 400  IE=IP+1,N
      A(IE,IP) = 0.
  400 CONTINUE
  100 CONTINUE
C
C     BACK SUBSTITUTION:
C
C     COMPUTE SOLUTION OF THE LAST EQUATION:
C
      X(N) = B(N)/A(N,N)
C
C     THEN COMPUTE SOLUTIONS FROM EQUATION N-1 TO 1:
C
      DO 500  IE=N-1,1,-1
```

```
      SUM = 0.
      DO 600  IC=IE+1,N
      SUM = SUM + A(IE,IC)*X(IC)
  600 CONTINUE
      X(IE) = (B(IE) - SUM)/A(IE,IE)
  500 CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------
C
      SUBROUTINE PIVOT(N, A, B, MXPOI, IP)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
C     PERFORM PARTIAL PIVOTING:
C
      JP = IP
      BIG = ABS(A(IP,IP))
      DO 10  I=IP+1,N
      AMAX = ABS(A(I,IP))
      IF(AMAX.GT.BIG)  THEN
         BIG = AMAX
         JP  = I
      ENDIF
   10 CONTINUE
         IF(JP.NE.IP)  THEN
      DO 20  J=IP,N
      DUMY    = A(JP,J)
      A(JP,J) = A(IP,J)
      A(IP,J) = DUMY
   20 CONTINUE
      DUMY  = B(JP)
      B(JP) = B(IP)
      B(IP) = DUMY
         ENDIF
      RETURN
      END
C
C-------------------------------------------------------------
C
      SUBROUTINE SCALE(N, A, B, MXPOI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXPOI,MXPOI), B(MXPOI)
C
C     PERFORM SCALING:
C
      DO 10  IE=1,N
      BIG = ABS(A(IE,1))
      DO 20  IC=2,N
      AMAX = ABS(A(IE,IC))
      IF(AMAX.GT.BIG)  BIG = AMAX
   20 CONTINUE
      DO 30  IC=1,N
      A(IE,IC) = A(IE,IC)/BIG
   30 CONTINUE
      B(IE) = B(IE)/BIG
   10 CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------
*****************************************************************
```

ภาคผนวก ข

รายละเอียดของโปรแกรม AXISSYM

รายละเอียดของโปรแกรม AXISSYM

โปรแกรม AXISSYM จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่างๆทั้งหมดดังนี้

```
**********************************************************************
*     PROGRAM  AXISSYM                                               *
*                                                                    *
*     A FINITE ELEMENT MECHANICAL/THERMAL STRESS ANALYSIS PROGRAM    *
*     FOR AXISYMETRIC PROBLEM (TWO-DIMENSIONAL) OPERATIONING UNDER   *
*     PRESSURE, CENTRIFUGAL FORCE, BODY FORCE AND TEMPERATURE.       *
*                                     MISS JUKSANEE VIRULSRI         *
*                                     FACULTY OF ENGINEERING         *
*                                     CHULALONGKORN UNIVERSITY       *
**********************************************************************
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        MXHBW = MAXIMUM NUMBER OF HALF-BANDWIDTH
C
C     COORDINATE IS R-Z-ZETA
C
      PARAMETER (MXPOI=781, MXELE=1400, MXHBW=1556 )
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  COORD(MXPOI,2), TEMP(MXPOI)
      REAL  PR1(MXELE), PZ1(MXELE)
      REAL  PR2(MXELE), PZ2(MXELE)
      REAL  SYSK(MXPOI*2,MXHBW), SYSF(MXPOI*2)
      REAL  SRR(MXPOI),SZZ(MXPOI),SOO(MXPOI),SRZ(MXPOI),ONE(MXPOI)
C
      CHARACTER*20  NAME1, NAME2,name4
      CHARACTER*4   TEXT
      CHARACTER*8   TEXT1
C
      INTEGER INTMAT(MXELE,3), IBC(MXPOI,2)
      INTEGER IEDGEP1(MXELE,2), IEDGEP2(MXELE,2), S(MXELE)
C
   10 WRITE(6,15)
   15 FORMAT(/,'**FINITE ELEMENT FOR SOLVING AXISYMETRIC PROBLEM**',/,
     *'  WITH PRESSURE TEMPERATURE AND CONSTANT ANGULAR VELOCITY',/,
     *'  PLEASE INPUT THE FILE NAME',/)
          READ(5, '(A)', ERR=10)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
C     READ TITLE OF COMPUTATION:
C
      READ(7,*)  NLINES
      DO 100  ILINE=1,NLINES
      READ(7,1)  TEXT
    1 FORMAT(A4)
  100 CONTINUE
C
C     READ INPUT DATA:
C
      READ(7,1)  TEXT
```

```
      READ(7,*)  NPOIN, NELEM, NFORCE ,NSGROUP
C
      IF(NPOIN.GT.MXPOI)  WRITE(6,110)  NPOIN
  110 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
      IF(NPOIN.GT.MXPOI)  STOP
      IF(NELEM.GT.MXELE)  WRITE(6,120)  NELEM
  120 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
C
      IF(NELEM.GT.MXELE)  STOP
      READ(7,1)  TEXT
      READ(7,*)  ELAS, PR, DENS, ALPHA, TREF, ANGVEL, GRAV
      READ(7,1)  TEXT
      DO 130  IP=1,NPOIN
      READ(7,*)  I, (COORD(I,K), K=1,2)
      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
C
      READ(7,1)  TEXT
      DO 131 IP = 1,NPOIN
         READ(7,*) I, TEMP(I)
         IF(I.NE.IP)  WRITE(6,136)  IP
  136    FORMAT(/, ' NODAL TEMP. NO.', I5, ' IN DATA FILE IS MISSING')
         IF(I.NE.IP)  STOP
  131 CONTINUE
C
      DO 132  IP = 1,NPOIN
         DO 133 IC = 1,2
            IBC(IP,IC) = 0.
  133 CONTINUE
  132 CONTINUE
C
      READ(7,1) TEXT
  134 READ(7,*) TEXT1, IPP, ISPC
      IF(TEXT1.EQ.'ENDCONST') GO TO 137
C
C FIX FOR COORDINATE IN PLANE X-Z
C
      IF(ISPC.EQ.1) IBC(IPP,1) = 1
      IF(ISPC.EQ.3) IBC(IPP,2) = 1
C
      IF(ISPC.EQ.13) THEN
          IBC(IPP,1) = 1
          IBC(IPP,2) = 1
       ELSE
       ENDIF
      IF(TEXT1.NE.'ENDCONST') GO TO 134
C
  137 READ(7,1) TEXT
C
      DO 140  IE=1,NELEM
      READ(7,*)  I, (INTMAT(I,J), J=1,3)
      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     NFORCE MEAN NUMBER OF NODES THAT SUPPORT THE EXTERNAL FORCE
C     NEPRESS MEAN NUMBER OF ELEMENT THAT FACE TO PRESSURE
C
      DO 200 ISF=1,NELEM
         S(ISF) = 0
  200 CONTINUE
C
      DO 205 IEP = 1,NELEM
         PR1(IEP) = 0.
         PR2(IEP) = 0.
         PZ1(IEP) = 0.
```

```
          PZ2(IEP) = 0.
          DO 206 J = 1,2
             IEDGEP1(IEP,J) = 0.
             IEDGEP2(IEP,J) = 0.
  206     CONTINUE
  205 CONTINUE
      IF(NSGROUP.EQ.0) GO TO 240
C
      DO 210 IG = 1,NSGROUP
         READ(7,1) TEXT
         READ(7,1) TEXT
  220    READ(7,*) TEXT1, IEP, PRR, PZZ, N1, N2
            IF(TEXT1.EQ.'ENDGROUP') GO TO 210
            IF(S(IEP).EQ.0) THEN
               PR1(IEP) = PRR
               PZ1(IEP) = PZZ
               IEDGEP1(IEP,1) = N1
               IEDGEP1(IEP,2) = N2
            ELSE
            ENDIF
            IF(S(IEP).EQ.1) THEN
               PR2(IEP) = PRR
               PZ2(IEP) = PZZ
               IEDGEP2(IEP,1) = N1
               IEDGEP2(IEP,2) = N2
            ELSE
            ENDIF
            S(IEP) = S(IEP)+1
            IF(TEXT1.NE.'ENDGROUP') GO TO 220
  210 CONTINUE
C
C
  240 NDF  = 2
      NDOF = 6
      NEQ  = NPOIN*NDF
      DO 300  I=1,NEQ
      SYSF(I) = 0.
  300 CONTINUE
      IF (NFORCE.EQ.0) GO TO 415
      READ(7,1)  TEXT
      DO 310  II=1,NFORCE
      READ(7,*)  N, FR, FZ
      write(10,*)'frfz=',n,fr,fz
      IEQ = (N-1)*NDF
      SYSF(IEQ+1) = FR
      write(10,*) fr
      SYSF(IEQ+2) = FZ
      write(10,*) fz
  310 CONTINUE
C
C     COMPUTE HALF-BANDWIDTH:
C
  415 NHBW = 0
      DO 400  IE=1,NELEM
      MIN = 100000
      MAX = 0
      DO 410  IN=1,3
      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
C
      NHBW = NHBW*NDF
      IF(NHBW.GT.MXHBW)  WRITE(6,420)  NHBW
  420 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.
  430 CONTINUE
      WRITE(6,435)  NPOIN, NELEM
  435 FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I5,
     *         ' NODES AND', I5,' ELEMENTS ***')
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(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *         ' ASSEMBLING ELEMENT EQUATIONS ***'      )
      CALL AXIS(NELEM, INTMAT, IEDGEP1, IEDGEP2, COORD, ELAS, PR,ALPHA,
     *ANGVEL, GRAV, DENS, PR1, PZ1, PR2, PZ2, S, TREF, TEMP, SYSK, SYSF,
     *MXPOI, MXELE, MXHBW, NPOIN)
C
      WRITE(6,450)
  450 FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS ***')
      CALL APPLYBC(NHBW, NPOIN, IBC, SYSK, SYSF, MXPOI, MXHBW)
C
      WRITE(6,460)
  460 FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS',
     *         ' FOR DISPLACEMENT SOLUTIONS ***'              )
      WRITE(6,465)  NEQ, NHBW
  465 FORMAT(5X,'( TOTAL OF', I5,' EQUATIONS WITH HALF-BANDWIDTH OF',
     *       I4, ' )' ,/ )
C
      CALL SOLVE(NEQ, NHBW, SYSK, SYSF, MXPOI, MXHBW)
C
C     PRINT OUT NODAL DISPLACEMENT SOLUTIONS:
C
  470 WRITE(6,480)
  480 FORMAT(/, ' PLEASE ENTER FILE NAME FOR DISPLACEMENT',
     *          ' AND STRESS SOLUTIONS:',/                 )
      READ(5, '(A)', ERR=470)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=470)
C
      WRITE(8,485)  NPOIN, NELEM,NHBW
  485 FORMAT('THE FINITE ELEMENT MODEL ',/,
     * 'CONSISTS OF', I5,' NODES AND', I5,' ELEMENTS ',/,
     * 'AND HALF BAND WIDTH OF', I5 )
      WRITE(8,490) NPOIN
  490 FORMAT('  NODAL DISPLACEMENT SOLUTIONS [', I5,']:',
     *       /,2X, 'NODE', 12X, 'U', 14X, 'W')
      DO 500  IP=1,NPOIN
         I1 = (IP-1)*NDF+1
         I2 = (IP-1)*NDF+2
         WRITE(8,510)  IP, SYSF(I1), SYSF(I2)
  510    FORMAT(I6, 5X, 2E16.7)
  500 CONTINUE
C
C     COMPUTE NODAL STRESSES:
C
      DO 700  I=1,NPOIN
      SRR(I) = 0.
      SZZ(I) = 0.
      SOO(I) = 0.
      SRZ(I) = 0.
      ONE(I) = 0.
  700 CONTINUE
C
C
C
      CALL STRESS(NPOIN, NELEM, INTMAT, COORD, SYSF, ELAS,  PR,
     *ALPHA,  TREF,   TEMP,  MXPOI, MXELE, SRR, SZZ, SOO, SRZ,
```

```
     *ONE)
C
      STOP
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NHBW, NPOIN, IBC, SYSK, SYSF, MXPOI, MXHBW)
C
C     APPLY DISPLACEMENT BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C           0 = FREE TO MOVE
C           1 = FIXED
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  SYSK(MXPOI*2,MXHBW), SYSF(MXPOI*2)
C
      INTEGER  IBC(MXPOI,2)
C
      NDF = 2
      DO 100  IN=1,NPOIN
      DO 200  ID=1,NDF
      IF(IBC(IN,ID).NE.1)  GO TO 200
C
      IEQ = (IN-1)*NDF + ID
      SYSF(IEQ)    = 0.
C
      SYSK(IEQ,1) = 1.
      DO 300  I=2,NHBW
      SYSK(IEQ,I) = 0.
  300 CONTINUE
C
      IF(IEQ.EQ.1)  GO TO 450
      DO 400  N=1,IEQ-1
      IROW = IEQ - N
      ICOL =    N + 1
      IF(ICOL.GT.NHBW)  GO TO 450
      SYSK(IROW,ICOL) = 0.
  400 CONTINUE
  450 CONTINUE
C
  200 CONTINUE
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE ASSMBLE(   IE, INTMAT,  SGBL, FGBL, SYSK, SYSF,
     *                   MXPOI,  MXELE, MXHBW                  )
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  SGBL(6,6), FGBL(6)
      REAL  SYSK(MXPOI*2,MXHBW), SYSF(MXPOI*2)
C
      INTEGER  INTMAT(MXELE,3)
C
      NNODE = 3
      NDF   = 2
C
      DO 100  NR=1,NNODE
      NODR = INTMAT(IE,NR)
      DO 100  MR=1,NDF
C
C     DENOTE:  NSR = ROW POSITION IN THE SYSTEM  EQS.
C              NER = ROW POSITION IN THE ELEMENT EQS.
C
```

```
      NSR = (NODR-1)*NDF + MR
      NER = (NR   -1)*NDF + MR
      SYSF(NSR) = SYSF(NSR) + FGBL(NER)
C
      DO 200  NC=1,NNODE
      NODC = INTMAT(IE,NC)
      DO 200  MC=1,NDF
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)*NDF + MC - NSR + 1
      NEC = (NC   -1)*NDF + 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 AXIS(NELEM, INTMAT, IEDGEP1, IEDGEP2, COORD, ELAS, PR,
     *ALPHA, ANGVEL, GRAV, DENS, PR1, PZ1, PR2, PZ2, S, TREF, TEMP,SYSK,
     *SYSF, MXPOI, MXELE, MXHBW, NPOIN)
C
C     COMPUTE ELEMENT MATRICES AND ASSEMBLE THEM FOR SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  COORD(MXPOI,2), TEMP(MXPOI)
      REAL  SYSK(MXPOI*2,MXHBW), SYSF(MXPOI*2)
      REAL  STRA(6,6), FT(6), C(4,4), B(4,6), BT(6,4)
      REAL  FEB(6), FES(6), FET(6)
      REAL  DUMA(4,6), DUMB(4), AL(4)
      REAL  PR1(MXELE), PZ1(MXELE)
      REAL  PR2(MXELE), PZ2(MXELE)
C
      INTEGER INTMAT(MXELE,3),IEDGEP1(MXELE,2),IEDGEP2(MXELE,2),S(MXELE)
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)
C
      RG1 = COORD(II,1)
      RG2 = COORD(JJ,1)
      RG3 = COORD(KK,1)
      ZG1 = COORD(II,2)
      ZG2 = COORD(JJ,2)
      ZG3 = COORD(KK,2)
C
      AVGR = (RG1 + RG2 + RG3)/3.
      AVGZ = (ZG1 + ZG2 + ZG3)/3.
C
      A1 = RG2*ZG3-RG3*ZG2
      A2 = RG3*ZG1-RG1*ZG3
      A3 = RG1*ZG2-RG2*ZG1
C
      B1 = ZG2-ZG3
      B2 = ZG3-ZG1
      B3 = ZG1-ZG2
```

```
C
      C1 = RG3-RG2
      C2 = RG1-RG3
      C3 = RG2-RG1
C
C
      AREA=1/2.*(A1 + A2 + A3)
C
C
      IF(AREA.LE.0.)  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(AREA.LE.0.)  STOP
C
C
      DO 110  I=1,4
      DO 110  J=1,6
      B(I,J) = 0.
  110 CONTINUE
C
      B(1,1) = B1
      B(1,3) = B2
      B(1,5) = B3
      B(2,2) = C1
      B(2,4) = C2
      B(2,6) = C3
      B(3,1) = A1/AVGR + B1 + C1*AVGZ/AVGR
      B(3,3) = A2/AVGR + B2 + C2*AVGZ/AVGR
      B(3,5) = A3/AVGR + B3 + C3*AVGZ/AVGR
      B(4,1) = C1
      B(4,2) = B1
      B(4,3) = C2
      B(4,4) = B2
      B(4,5) = C3
      B(4,6) = B3
C
      DO 120  I=1,4
      DO 130  J=1,6
      B(I,J) = B(I,J)/(2.*AREA)
      BT(J,I) = B(I,J)
  130 CONTINUE
  120 CONTINUE
C
C     ELASTICITY MATRIX:
C
      DO 140 I = 1,4
      DO 150 J = 1,4
      C(I,J) = 0.
  150 CONTINUE
  140 CONTINUE
      FAC = ELAS/((1.+PR)*(1.-2.*PR))
      C(1,1) = FAC*(1-PR)
      C(1,2) = FAC*PR
      C(1,3) = FAC*PR
      C(2,1) = C(1,2)
      C(2,2) = C(1,1)
      C(2,3) = FAC*PR
      C(3,1) = C(1,3)
      C(3,2) = C(2,3)
      C(3,3) = C(1,1)
      C(4,4) = FAC*(1.-2.*PR)/2.
C
C     ELEMENT STIFFNESS MATRIX:
C
      DO 100  I=1,4
      DO 100  J=1,6
      DUMA(I,J) = 0.
```

```
      DO 200  K=1,4
      DUMA(I,J) = DUMA(I,J) + C(I,K)*B(K,J)
  200 CONTINUE
  100 CONTINUE
C
      DO 300  I=1,6
      DO 300  J=1,6
      STRA(I,J) = 0.
      DO 400  K=1,4
      STRA(I,J) = STRA(I,J) + BT(I,K)*DUMA(K,J)
  400 CONTINUE
  300 CONTINUE
C
      DO 500  I=1,6
      DO 500  J=1,6
      STRA(I,J) = 2.*3.141592654*AVGR*AREA*STRA(I,J)
  500 CONTINUE
C
C     ELEMENT NODAL FORCE DUE TO THERMAL EXPANSION:
C
      AL(1) = ALPHA
      AL(2) = ALPHA
      AL(3) = ALPHA
      AL(4) = 0.
C
      DO 600  I=1,4
      DUMB(I) = 0.
      DO 700  J=1,4
      DUMB(I) = DUMB(I) + C(I,J)*AL(J)
  700 CONTINUE
  600 CONTINUE
C
      DO 800  I=1,6
      FET(I) = 0.
      DO 900  J=1,4
      FET(I) = FET(I) + BT(I,J)*DUMB(J)
  900 CONTINUE
  800 CONTINUE

C
C     AVERAGE ELEMENT TEMPERATURE:
C
      TAVG = (TEMP(II) + TEMP(JJ) + TEMP(KK))/3.
C
      FAC = 2*3.141592654*AVGR*AREA*(TAVG - TREF)
      DO 1000  I=1,6
      FET(I) = FET(I)*FAC
 1000 CONTINUE
C
C     ELEMENT BODY FORCE DUE TO CENTRIFUGAL FORCE
C
      FACB  = 2*3.141592654*AVGR*AREA/3.
      FACBR = FACB*(DENS*ANGVEL*ANGVEL*AVGR)
      FACBZ = FACB*(-DENS*GRAV)
C
      FEB(1) = FACBR
      FEB(2) = FACBZ
      FEB(3) = FACBR
      FEB(4) = FACBZ
      FEB(5) = FACBR
      FEB(6) = FACBZ
C
C     ELEMENT NADAL FORCE DUE TO SURFACE FORCE BY UNIFORM PRESSURE
C
      DO 1200 IF = 1,6
      FES(IF) = 0.
 1200 CONTINUE
      IF(S(IE).EQ.0.) GO TO 1310
C
```

```
C
      IIS = S(IE)
      DO 1210 IL = 1,IIS
       IF(S(IE).EQ.1)  THEN
         IS = IEDGEP1(IE,1)
         JS = IEDGEP1(IE,2)
         PRR = PR1(IE)
         PZZ = PZ1(IE)
       ELSE
         IS = IEDGEP2(IE,1)
         JS = IEDGEP2(IE,2)
         PRR = PR2(IE)
         PZZ = PZ2(IE)
       ENDIF
        RI = COORD(IS,1)
        RJ = COORD(JS,1)
        ZI = COORD(IS,2)
        ZJ = COORD(JS,2)
        AVGRS = (RI + RJ)/2.
        DIFFZ = ABS(ZI - ZJ)
        FACS = 2.*3.141592654*AVGRS*DIFFZ/2.
C
        IF(IS.EQ.II) then
              IS = 1
        else
              IF(IS.EQ.JJ) then
            IS = 2
          else
                IF(IS.EQ.KK) then
              IS = 3
            else
            endif
          endif
        endif
        IF(JS.EQ.II) then
               JS = 1
        else
              IF(JS.EQ.JJ) then
             JS = 2
          else
                IF(JS.EQ.KK) then
               JS = 3
            else
            endif
          endif
        endif
C
      NNODE = 3.
      NDF = 2.
      DO 1300 I = 1,NNODE
      IF((I.NE.IS).AND.(I.NE.JS)) GO TO 1300
         IRR = (I-1)*NDF + 1.
         IRZ = (I-1)*NDF + 2.
         FES(IRR) = FACS*PRR + FES(IRR)
         FES(IRZ) = FACS*PZZ + FES(IRZ)
 1300 CONTINUE
 1210 CONTINUE

C
C     FIND OF TOTAL FORCE OF ELEMENT
 1310 DO 1400 ISUM = 1,6
      FT(ISUM) = FEB(ISUM) + FES(ISUM) + FET(ISUM)
 1400 CONTINUE

C
C
C     ASSEMBLE THESE ELEMENT EQUATIONS INTO THE SYSTEM EQUATIONS:
C
      CALL ASSMBLE(   IE, INTMAT,  STRA, FT, SYSK, SYSF,
```

```
     *              MXPOI,  MXELE, MXHBW                    )
C
 5000 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE SOLVE(NROW, NHBW, GSTIF, XL, MXPOI, MXHBW)
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 REAL*8(A-H,O-Z)
C
      REAL  GSTIF(MXPOI*2,MXHBW), XL(MXPOI*2)
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.E-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) GO TO 30
      PIVOT2=GSTIF(I,MM+1)*PIVOT1
      XL(II)=XL(II)-XL(I)*PIVOT2
      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=1
      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 STRESS(NPOIN, NELEM, INTMAT, COORD, DISP, ELAS,  PR,
     *      ALPHA, TREF, TEMP, MXPOI, MXELE, SRR, SZZ, SOO, SRZ, ONE)
C
C     COMPUTE NODAL STRESS COMPONENTS FOR TETRAHEDRAL ELEMENTS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  COORD(MXPOI,2), TEMP(MXPOI), DISP(MXPOI*2),ONE(MXPOI)
      REAL  SRR(MXPOI), SZZ(MXPOI), SOO(MXPOI), SRZ(MXPOI)
C
      REAL  C(4,4), B(4,6), EPS(4), UG(4), WG(4)
C
      INTEGER  INTMAT(MXELE,3)
C
C     PRINT OUT ELEMENT STRESSES:
      WRITE(8,11) NELEM
  11 FORMAT( ' ELEMENTAL STRESS SOLUTIONS [', I5,']:',/,
     *        2X, 'ELEM', 11X, 'SRR', 11X, 'SZZ',
     *        11X, 'SOO', 11X, 'SRZ' )
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)
C
      RG1 = COORD(II,1)
      RG2 = COORD(JJ,1)
      RG3 = COORD(KK,1)
      ZG1 = COORD(II,2)
      ZG2 = COORD(JJ,2)
      ZG3 = COORD(KK,2)
C
C
      AVGR = (RG1 + RG2 + RG3)/3.
      AVGZ = (ZG1 + ZG2 + ZG3)/3.
C
      A1 = RG2*ZG3-RG3*ZG2
      A2 = RG3*ZG1-RG1*ZG3
      A3 = RG1*ZG2-RG2*ZG1
C
      B1 = ZG2-ZG3
      B2 = ZG3-ZG1
      B3 = ZG1-ZG2
C
      C1 = RG3-RG2
```

```
      C2 = RG1-RG3
      C3 = RG2-RG1
C
C
      AREA=1/2.*(A1 + A2 + A3)
C
C
      IF(AREA.LE.0.)  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(AREA.LE.0.)  STOP
C
      DO 110  I=1,4
      DO 110  J=1,6
      B(I,J) = 0.
  110 CONTINUE
C
C
      B(1,1) = B1
      B(1,3) = B2
      B(1,5) = B3
      B(2,2) = C1
      B(2,4) = C2
      B(2,6) = C3
      B(3,1) = A1/AVGR + B1 + C1*AVGZ/AVGR
      B(3,3) = A2/AVGR + B2 + C2*AVGZ/AVGR
      B(3,5) = A3/AVGR + B3 + C3*AVGZ/AVGR
      B(4,1) = C1
      B(4,2) = B1
      B(4,3) = C2
      B(4,4) = B2
      B(4,5) = C3
      B(4,6) = B3
C
      DO 120  I=1,4
      DO 130  J=1,6
      B(I,J) = B(I,J)/(2.*AREA)
  130 CONTINUE
  120 CONTINUE
C
C     ELASTICITY MATRIX:
C
      DO 140 I = 1,4
      DO 150 J = 1,4
      C(I,J) = 0.
  150 CONTINUE
  140 CONTINUE
C
      FAC = ELAS/((1.+PR)*(1.-2.*PR))
      C(1,1) = FAC*(1-PR)
      C(1,2) = FAC*PR
      C(1,3) = FAC*PR
      C(2,1) = C(1,2)
      C(2,2) = C(1,1)
      C(2,3) = FAC*PR
      C(3,1) = C(1,3)
      C(3,2) = C(2,3)
      C(3,3) = C(1,1)
      C(4,4) = FAC*(1.-2.*PR)/2.
C
C     GATHER ELEMENT NODAL DISPLACEMENTS:
C
      NDOF = 2
      DO 200  J1=1,3
      I1  = INTMAT(IE,J1)
      IEQ = (I1-1)*NDOF + 1
      UG(J1) = DISP(IEQ)
```

```
      WG(J1) = DISP(IEQ+1)
  200 CONTINUE
C
C     COMPUTE THE TOTAL STRAINS:
C
      DO 220  I=1,4
      EPS(I) = 0.
      DO 230  J=1,3
      J1 = (J-1)*NDOF + 1
      J2 = J1 + 1
      EPS(I) = EPS(I) + B(I,J1)*UG(J) + B(I,J2)*WG(J)
  230 CONTINUE
  220 CONTINUE
C
C     COMPUTE THERMAL STRAINS USING AVERAGE ELEMENT NODAL TEMPERATURES:
C
      TAVG = (TEMP(II) + TEMP(JJ) + TEMP(KK))/3.
C
C     COMPUTE THE NET STRAINS:
C
      EPS(1) = EPS(1) - ALPHA*(TAVG - TREF)
      EPS(2) = EPS(2) - ALPHA*(TAVG - TREF)
      EPS(3) = EPS(3) - ALPHA*(TAVG - TREF)
C
C     COMPUTE THE ELEMENT STRESSES:
C
      SRRE = C(1,1)*EPS(1) + C(1,2)*EPS(2) + C(1,3)*EPS(3)
      SZZE = C(2,1)*EPS(1) + C(2,2)*EPS(2) + C(2,3)*EPS(3)
      SOOE = C(3,1)*EPS(1) + C(3,2)*EPS(2) + C(3,3)*EPS(3)
      SRZE = C(4,4)*EPS(4)
C
C     COMPUTE NODAL STRESSES FROM ELEMENT STRESSES:
C
      SRR(II) = SRR(II) + SRRE
      SRR(JJ) = SRR(JJ) + SRRE
      SRR(KK) = SRR(KK) + SRRE
      SZZ(II) = SZZ(II) + SZZE
      SZZ(JJ) = SZZ(JJ) + SZZE
      SZZ(KK) = SZZ(KK) + SZZE
      SOO(II) = SOO(II) + SOOE
      SOO(JJ) = SOO(JJ) + SOOE
      SOO(KK) = SOO(KK) + SOOE
      SRZ(II) = SRZ(II) + SRZE
      SRZ(JJ) = SRZ(JJ) + SRZE
      SRZ(KK) = SRZ(KK) + SRZE
C
      ONE(II) = ONE(II) + 1.
      ONE(JJ) = ONE(JJ) + 1.
      ONE(KK) = ONE(KK) + 1.
C
C     PRINT OUT ELEMENTAL STRESS SOLUTION
      WRITE(8,12) IE,SRRE,SZZE,SOOE,SRZE
   12 FORMAT(I6,4E14.6)
C
 1000 CONTINUE
C
C     PRINT OUT THESE NODAL STRESSES:
C
      DO 1100  I=1,NPOIN
      IF(ONE(I).EQ.0.)  WRITE(6,1200)  I
 1200 FORMAT(' *** WARNING ***  NO STRESS CONTRIBUTION AT NODE', I5)
      IF(ONE(I).EQ.0.)  ONE(I) = 1.
      SRR(I) = SRR(I)/ONE(I)
      SZZ(I) = SZZ(I)/ONE(I)
      SOO(I) = SOO(I)/ONE(I)
      SRZ(I) = SRZ(I)/ONE(I)
 1100 CONTINUE
C
      WRITE(8,1300)  NPOIN
```

```
 1300 FORMAT( '  NODAL STRESS SOLUTIONS [', I5,']:', /,
     *         2X, 'NODE', 11X, 'SRR', 11X, 'SZZ',
     *         11X, 'SOO', 11X, 'SRZ' )
      DO 1400  I=1,NPOIN
      WRITE(8,1500)  I, SRR(I), SZZ(I), SOO(I), SRZ(I)
 1500 FORMAT(I6, 4E14.6)
 1400 CONTINUE
C
C     PRINT OUT ELEMENT NODAL CONNECTION
      WRITE(8,*) 'ELEMENT NODAL CONNECTION'
      WRITE(8,13)
   13 FORMAT(3X,'ELE',8X,'I',3X,'J',5X,'K')
      DO 1600 IE = 1,NELEM
      WRITE(8,14) IE,INTMAT(IE,1),INTMAT(IE,2),INTMAT(IE,3)
   14 FORMAT(I5,5X,3I5)
 1600 CONTINUE
C
      RETURN
      END
************************************************************************
```

รายละเอียดของโปรแกรม TRANAXIS

รายละเอียดของโปรแกรม TRANAXIS

โปรแกรม TRANAXIS จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่าง ๆ ทั้งหมดดังนี้

```
***********************************************************************
*          TRANSLATE OUTPUT FILE OF AXISSYM TO NASTRAN FORMAT         *
*                 AND CONCULATE VONMISES  STRESS                      *
*                                     MISS JUKSANEE VIRULSRI          *
*                                     FACULTY OF ENGINEERING          *
*                                     CHULALONGKORN UNIVERSITY        *
***********************************************************************
      PARAMETER (MXPOI=444, MXELE=1282)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  SRREE(MXELE), SZZEE(MXELE), SOOEE(MXELE), SRZEE(MXELE)
      REAL  VONE(MXELE), SRRE, SZZE, SOOE, SRZE, SVONE

C
      CHARACTER*20  NAME1, NAME2
C
   10 WRITE(6,15)
   15 FORMAT(/,'TRANSLATE OUTPUT FILE OF AXIS PROGRAM TO NASTRAN FORMAT'
     *,/,'  PLEASE INPUT THE OUTPUT FILE TO TRANSLATE:',/)
      READ(5, '(A)', ERR=10)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
      READ(7,1)  TEXT
    1 FORMAT(20A4)
      READ(7,200)  NPOIN, NELEM
  200 FORMAT(T12,I5,T27,I5)
C
      IF(NPOIN.GT.MXPOI)  WRITE(6,20)  NPOIN
   20 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
      IF(NPOIN.GT.MXPOI)  STOP
      IF(NELEM.GT.MXELE)  WRITE(6,30)  NELEM
   30 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)  STOP
C
      READ(7,1) TEXT
      READ(7,1) TEXT
      READ(7,1) TEXT
C
C     PRINT DISPLACEMENT
C
   40 WRITE(6,45)
   45 FORMAT(/, ' PLEASE ENTER FILE NAME FOR DISPLACEMENT'
     *          ' AND STRESS SOLUTIONS OF NASTRAN FORMAT:',/  )
      READ(5, '(A)', ERR=40)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=40)
      WRITE(8,50)
   50 FORMAT(T1,'1 MSC/NASTRAN PAGE')
      WRITE(8,60) 0
   60 FORMAT(T1,I1)
      WRITE(8,*) 'D I S P L A C E M E N T'
      WRITE(8,*) 'POINT ID. TYPE          T1             T2              T3
     *          R1              R2             R3'
C
      DO 65 I = 1,NPOIN
      READ(7,*) IP, U, W
      V = 0.
      R1 = 0.
      R2 = 0.
      R3 = 0.
      IF(I.NE.IP)  WRITE(6,75)  I
```

```
   75 FORMAT(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP)  STOP
      WRITE(8,70)  IP, U, V, W, R1, R2, R3
   70 FORMAT(t7,I8,t21,'G',t27,E13.6,F6.1,t57,e13.6,F6.1,t85,F6.1,t100,
     *F6.1)
   65 CONTINUE
      WRITE(8,60) 1
C
C
      DO 66 I = 1,NELEM
         SRREE(I) = 0.
         SZZEE(I) = 0.
         SOOEE(I) = 0.
         SRZEE(I) = 0.
         VONE(I) = 0.
   66 CONTINUE
C
      READ(7,1) TEXT
      READ(7,1) TEXT
      DO 300 IE = 1,NELEM
      READ(7,*) II,SRREE(II),SZZEE(II),SOOEE(II),SRZEE(II)
      IF(II.NE.IE)  WRITE(6,80)  IE
   80 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(II.NE.IE)  STOP
      SRRE = SRREE(II)
      SZZE = SZZEE(II)
      SOOE = SOOEE(II)
      SRZE = SRZEE(II)
C
C     SOLVE FOR VON MISES STRESS OF EACH ELEMENT
      CALL VONM(SRRE,SZZE,SOOE,SRZE,SVONE)
      VONE(II) = SVONE
  300 CONTINUE
C
C     PRINT  STRESS DISTRIBUTION
C
      WRITE(8,60) 0
      WRITE(8,90)
   90 FORMAT(t39,'S T R E S S E S   I N    T R I A X 6   E L E M E N T S
     * ',/)
      WRITE(8,1010)
 1010 FORMAT(T7,'ID',T24,'RADIAL',T38,'AZIMUTHAL',T52,'AXIAL',T66,'SHEAR
     *',T108,'VON MISES')
      DO 600 IE = 1,NELEM
      WRITE(8,640) IE,0,SRREE(IE),SOOEE(IE),SZZEE(IE),SRZEE(IE),0,0
     * ,VONE(IE)
  640 FORMAT(T3,I8,T19,I1,T21,4(1x,E13.6),T90,I1,T104,I1,T106,E13.6)
  600 CONTINUE
      WRITE(8,60) 1
C
      STOP
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE VONM(SRRE,SZZE,SOOE,SRZE,SVONE)
C
      REAL SRRE, SZZE, SOOE, SRZE, SPE1, SPE2, SPE3, SVONE
C
      SPE1 = (SRRE+SZZE)/2.+SQRT(((SRRE-SZZE)/2.)**2+SRZ**2)
      SPE2 = (SRRE+SZZE)/2.-SQRT(((SRRE-SZZE)/2.)**2+SRZ**2)
      SPE3 = SOOE
C
C  FIND VON MISES STRESS
C
      SVONE = 1./SQRT(2.)*SQRT((SPE1-SPE2)**2+(SEP2-SPE3)**2+
     *(SPE3-SPE1)**2)
C
      RETURN
      END
************************************************************************
```

ภาคผนวก ค

รายละเอียดของโปรแกรม SOLID3D

รายละเอียดของโปรแกรม SOLID3D

โปรแกรม SOLID3D จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่างๆทั้งหมดดังนี้

```
*************************************************************************
*     PROGRAM  STRES3D                                                  *
*                                                                       *
*     A FINITE ELEMENT MECHANICAL/THERMAL STRESS ANALYSIS PROGRAM       *
*     FOR SOLID PROBLEM(THREE-DIMENSIONAL) OPERATIONING UNDER           *
*     STEAM PRESSURE, CENTRIFUGAL FORCE AND HIGH TEMPERATURE.           *
*                                     MISS JUKSANEE VIRULSRI            *
*                                     FACULTY OF ENGINEERING            *
*                                     CHULALONGKORN UNIVERSITY          *
*************************************************************************
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        MXHBW = MAXIMUM NUMBER OF HALF-BANDWIDTH
C
      PARAMETER (MXPOI=1134, MXELE=4000, MXHBW=300)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  COORD(MXPOI,3), TEMP(MXPOI)
      REAL  P1(MXELE,2), P2(MXELE,2), P3(MXELE,2)
      REAL  SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
      REAL  SXX(MXPOI),SYY(MXPOI),SZZ(MXPOI),SXY(MXPOI)
      REAL  SXZ(MXPOI),SYZ(MXPOI),ONE(MXPOI)
C
      CHARACTER*20  NAME1, NAME2, TEXT1
      CHARACTER*4   TEXT
C
      INTEGER INTMAT(MXELE,4), IBC(MXPOI,3), S(MXELE)
      INTEGER INTMATP1(MXELE,3),INTMATP2(MXELE,3),INTMATP3(MXELE,3)
C
   10 WRITE(6,15)
   15 FORMAT(/,'**FINITE ELEMENT FOR SOLVING DISPLACEMENT AND STRESS OF
     *TURBINE BLADE PROBLEM**',/,'  WITH CONSTANT PRESSURE AND CONSTANT
     *ANGULAR VELOCITY',//,
     *'  PLEASE INPUT THE FILE NAME OF INPUT DATA:',/)
      READ(5, '(A)', ERR=10)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
C     READ TITLE OF COMPUTATION:
C
      READ(7,*)  NLINES
      DO 100  ILINE=1,NLINES
      READ(7,1)  TEXT
    1 FORMAT(A4)
  100 CONTINUE
C
C     READ INPUT DATA:
C
      READ(7,1)  TEXT
      READ(7,*)  NPOIN, NELEM, NFORCE , NSGROP
C
```

```
      IF(NPOIN.GT.MXPOI)  WRITE(6,110)  NPOIN
  110 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
      IF(NPOIN.GT.MXPOI)  STOP
      IF(NELEM.GT.MXELE)  WRITE(6,120)  NELEM
  120 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)  STOP
      READ(7,1)  TEXT
      READ(7,*)  ELAS, PR, DENS, ALPHA, TREF, ANGVEL
      READ(7,1)  TEXT
      DO 130  IP=1,NPOIN
      READ(7,*)  I, (COORD(I,K), K=1,3)
c    2 FORMAT(T9,I8,T25,F8.0,T33,F8.0,T41,F8.0)
      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 131 IP = 1,NPOIN
         READ(7,*) I, TEMP(I)
c    3    FORMAT(T17,I8,T25,F8.0)
         IF(I.NE.IP)  WRITE(6,136)  IP
  136    FORMAT(/, ' NODAL TEMP. NO.', I5, ' IN DATA FILE IS MISSING')
         IF(I.NE.IP)  STOP
  131 CONTINUE
C
      DO 132  IP = 1,NPOIN
         DO 133 IC = 1,3
            IBC(IP,IC) = 0
  133 CONTINUE
  132 CONTINUE
      READ(7,1) TEXT
  134 READ(7,*) TEXT1, IPP, ISPC
c  141 FORMAT(T1,A8,T17,I8,T25,I8)
      IF(TEXT1.EQ.'ENDCONST') GO TO 137
      DO 138 I = 1,3
          IF(ISPC.EQ.I) IBC(IPP,I) = 1
  138 CONTINUE
       IF(ISPC.EQ.12) THEN
          IBC(IPP,1) = 1
          IBC(IPP,2) = 1
       ELSE
       ENDIF
       IF(ISPC.EQ.13) THEN
          IBC(IPP,1) = 1
          IBC(IPP,3) = 1
       ELSE
       ENDIF
       IF(ISPC.EQ.123) THEN
          IBC(IPP,1) = 1
          IBC(IPP,2) = 1
          IBC(IPP,3) = 1
       ELSE
       ENDIF
       IF(ISPC.EQ.23) THEN
          IBC(IPP,2) = 1
          IBC(IPP,3) = 1
       ELSE
       ENDIF
      IF(TEXT1.NE.'ENDCONST') GO TO 134
C
  137 READ(7,1) TEXT
      DO 140  IE=1,NELEM
      READ(7,*)  I, (INTMAT(I,J), J=1,4)
c    4 FORMAT(T9,I8,T25,I8,T33,I8,T41,I8,T49,I8)
      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     NFORCE MEAN NUMBER OF NODES THAT SUPPORT THE EXTERNAL FORCE
C     NSGROP MEAN GROUP NUMBER OF ELEMENT THAT FACE TO PRESSURE
C     NESURF MEAN NUMBER OF ELEMENT THAT FACE TO PRESSURE IN EACH NSGROP
C
      DO 200 ISF=1,NELEM
         S(ISF) = 0
  200 CONTINUE
C
C     CONSIDER INTMATP OF THE ELEMENT THAT FACE TO PRESSURE
C     FROM EACH NSGROP
      IF(NSGROP.EQ.0) GO TO 240
      DO 205 IEP = 1,NELEM
         DO 207 JJ = 1,2
            P1(IEP,JJ) = 0.
            P2(IEP,JJ) = 0.
            P3(IEP,JJ) = 0.
  207    CONTINUE
         DO 206 J = 1,3
            INTMATP1(IEP,J) = 0.
            INTMATP2(IEP,J) = 0.
            INTMATP3(IEP,J) = 0.
  206    CONTINUE
  205 CONTINUE
C
      DO 210 IG = 1,NSGROP
         READ(7,1) TEXT
  220       IIC = 1
            READ(7,*) TEXT1, IEP, P, INP
c    5        FORMAT(T1,A8,T17,I8,T25,F8.0,T65,I8)
            IF(TEXT1.EQ.'ENDGROUP') GO TO 210
            IF(S(IEP).EQ.0) THEN
               P1(IEP,1) = P
               P1(IEP,2) = INP
            ENDIF
            IF(S(IEP).EQ.1) THEN
               P2(IEP,1) = P
               P2(IEP,2) = INP
            ENDIF
            IF(S(IEP).EQ.2) THEN
               P3(IEP,1) = P
               P3(IEP,2) = INP
            ENDIF
            DO 230 IC = 1,4
               IF(INP.EQ.INTMAT(IEP,IC)) GO TO 230
               IF(S(IEP).EQ.0) INTMATP1(IEP,IIC) = INTMAT(IEP,IC)
               IF(S(IEP).EQ.1) INTMATP2(IEP,IIC) = INTMAT(IEP,IC)
               IF(S(IEP).EQ.2) INTMATP3(IEP,IIC) = INTMAT(IEP,IC)
               IIC = IIC+1
  230 CONTINUE
            S(IEP) = S(IEP)+1
            IF(TEXT1.NE.'ENDGROUP') GO TO 220
  210 CONTINUE
C
  240 NDF  = 3
      NDOF = 12
      NEQ  = NPOIN*NDF
      DO 300  I=1,NEQ
      SYSF(I) = 0.
  300 CONTINUE
      IF (NFORCE.EQ.0) GO TO 415
      READ(7,1)  TEXT
      DO 310  II=1,NFORCE
      READ(7,*)  N, FX, FY ,FZ
c    6 FORMAT(T17,I8,T41,F8.0,T49,F8.0,T57,F8.0)
      IEQ = (N-1)*NDF
      SYSF(IEQ+1) = FX
      SYSF(IEQ+2) = FY
      SYSF(IEQ+3) = FZ
  310 CONTINUE
```

```
C
C     COMPUTE HALF-BANDWIDTH:
C
  415 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
C
      NDIF = MAX - MIN + 1
      IF(NDIF.GT.NHBW)  NHBW = NDIF
  400 CONTINUE
C
      NHBW = NHBW*NDF
      IF(NHBW.GT.MXHBW)  WRITE(6,420)  NHBW
  420 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.
  430 CONTINUE
C
     WRITE(6,435)  NPOIN, NELEM
  435 FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I5,
     *          ' NODES AND', I5,' ELEMENTS ***')
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(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *          ' ASSEMBLING ELEMENT EQUATIONS ***'      )
      CALL TETRA(NELEM, INTMAT, INTMATP1, INTMATP2, INTMATP3,
     *COORD, ELAS, PR, ALPHA, ANGVEL, DENS, P1, P2, P3, S, TREF,
     *TEMP, SYSK, SYSF, MXPOI, MXELE, MXHBW)
C
      WRITE(6,450)
  450 FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS ***')
      CALL APPLYBC(NHBW, NPOIN, IBC, SYSK, SYSF, MXPOI, MXHBW)
C
      WRITE(6,460)
  460 FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS',
     *          ' FOR DISPLACEMENT SOLUTIONS ***'             )
      MEM = 1288+144*MXPOI+104*MXELE+24*MXPOI*MXHBW
      WRITE(6,465)  NEQ, NHBW, MEM
  465 FORMAT(5X,'( TOTAL OF', I5,' EQUATIONS WITH HALF-BANDWIDTH OF',
     *        I4, ' )' ,/,' REQUIRED MEMORY ABOUT ',12I,/ )
C
      CALL SOLVE(NEQ, NHBW, SYSK, SYSF, MXPOI, MXHBW)
C
C     PRINT OUT NODAL DISPLACEMENT SOLUTIONS:
C
C
  470 WRITE(6,480)
  480 FORMAT(/, ' PLEASE ENTER FILE NAME FOR DISPLACEMENT'
     *           ' AND STRESS SOLUTIONS:',/                )
      READ(5, '(A)', ERR=470)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=470)
      WRITE(8,485)  NAME1,NPOIN, NELEM,NHBW
  485 FORMAT(,'THE FINITE ELEMENT MODEL OF INPUT FILE "   ',A20,'"',/,
     * 'CONSISTS OF', I5,' NODES AND', I5,' ELEMENTS ',/,
     * 'AND HALF BAND WIDTH OF', I5 ,/,' MEMORY REQUIRED ',I12,' bytes')
      WRITE(8,490) NPOIN
```

```
 490 FORMAT('  NODAL DISPLACEMENT SOLUTIONS [', I5,']:',
    *        /,2X, 'NODE', 12X, 'U', 14X, 'V', 14X, 'W',
    *        12X,'R1',12X,'R2',12X,'R3')
     DO 500  IP=1,NPOIN
         R = 0.
        I1 = (IP-1)*NDF+1
        I2 = (IP-1)*NDF+2
        I3 = (IP-1)*NDF+3
        WRITE(8,510)  IP, SYSF(I1), SYSF(I2), SYSF(I3), R, R, R
 510    FORMAT(I6, 5X, 6E14.6)
 500 CONTINUE
C
C
C     COMPUTE NODAL STRESSES:
C
      DO 700  I=1,NPOIN
      SXX(I) = 0.
      SYY(I) = 0.
      SZZ(I) = 0.
      SXY(I) = 0.
      SYZ(I) = 0.
      SXZ(I) = 0.
      ONE(I) = 0.
 700  CONTINUE
C
C
C     PRINT OUT CHECK FILE :
C
      CALL STRESS(NPOIN, NELEM, INTMAT, COORD, SYSF, ELAS,  PR,
     *ALPHA,  TREF,   TEMP,  MXPOI, MXELE, SXX, SYY, SZZ, SXY, SXZ,
     *SYZ, ONE)
C
      STOP
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NHBW, NPOIN, IBC, SYSK, SYSF, MXPOI, MXHBW)
C
C     APPLY DISPLACEMENT BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C           0 = FREE TO MOVE
C           1 = FIXED
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
C
      INTEGER  IBC(MXPOI,3)
C
      NDF = 3
      DO 100  IN=1,NPOIN
      DO 200  ID=1,NDF
      IF(IBC(IN,ID).NE.1)  GO TO 200
C
      IEQ = (IN-1)*NDF + ID
      SYSF(IEQ)   = 0.
C
      SYSK(IEQ,1) = 1.
      DO 300  I=2,NHBW
      SYSK(IEQ,I) = 0.
 300  CONTINUE
C
      IF(IEQ.EQ.1)  GO TO 450
      DO 400  N=1,IEQ-1
      IROW = IEQ - N
      ICOL =    N + 1
      IF(ICOL.GT.NHBW)  GO TO 450
      SYSK(IROW,ICOL) = 0.
 400  CONTINUE
 450  CONTINUE
```

```
C
  200 CONTINUE
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ASSMBLE(   IE, INTMAT,  SGBL, FGBL, SYSK, SYSF,
     *                      MXPOI,  MXELE, MXHBW                  )
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  SGBL(12,12), FGBL(12)
      REAL  SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
C
      INTEGER  INTMAT(MXELE,4)
C
      NNODE = 4
      NDF   = 3
C
      DO 100  NR=1,NNODE
      NODR = INTMAT(IE,NR)
      DO 100  MR=1,NDF
C
C     DENOTE:  NSR = ROW POSITION IN THE SYSTEM  EQS.
C              NER = ROW POSITION IN THE ELEMENT EQS.
C
      NSR = (NODR-1)*NDF + MR
      NER = (NR   -1)*NDF + MR
      SYSF(NSR)  = SYSF(NSR)  + FGBL(NER)
C
      DO 200  NC=1,NNODE
      NODC = INTMAT(IE,NC)
      DO 200  MC=1,NDF
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)*NDF + MC - NSR + 1
      NEC = (NC   -1)*NDF + MC
      IF(NSC.GT.0)
     &     SYSK(NSR,NSC)  = SYSK(NSR,NSC)  + SGBL(NER,NEC)
  200 CONTINUE
  100 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE TETRA(NELEM, INTMAT, INTMATP1, INTMATP2, INTMATP3,
     *COORD, ELAS, PR, ALPHA, ANGVEL, DENS, P1, P2, P3, S, TREF,
     *TEMP, SYSK, SYSF, MXPOI, MXELE, MXHBW)
C
C     COMPUTE ELEMENT MATRICES AND ASSEMBLE THEM FOR SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  COORD(MXPOI,3), TEMP(MXPOI)
      REAL  SYSK(MXPOI*3,MXHBW), SYSF(MXPOI*3)
      REAL  STRA(12,12), FT(12), C(6,6), B(6,12), BT(12,6)
      REAL  FEB(12), FES(12), FET(12)
      REAL  DUMA(6,12), DUMB(6), AL(6)
      REAL  P1(MXELE,2), P2(MXELE,2), P3(MXELE,2)
c       CHARACTER*20 NAME3
C
      INTEGER  INTMAT(MXELE,4) , S(MXELE)
      INTEGER  INTMATP1(MXELE,3),INTMATP2(MXELE,3),INTMATP3(MXELE,3)
```

```
      REAL CX,CY,CZ,FACC,P,NX,NY,NZ
C
C
c 4070 WRITE(6,4800)
c 4800 FORMAT(/, ' PLEASE ENTER FILE NAME FOR CHECKING',/)
c       READ(5, '(A)', ERR=4070)  NAME3
c       OPEN(UNIT=9, FILE=NAME3, STATUS='NEW', ERR=4070)
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)
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)
      ZG1 = COORD(II,3)
      ZG2 = COORD(JJ,3)
      ZG3 = COORD(KK,3)
      ZG4 = COORD(LL,3)
C
      VOL=1/6.*(((XG2-XG3)*YG4+(XG4-XG2)*YG3+(XG3-XG4)*YG2)*ZG1+
     *((XG1-XG4)*YG3+(XG3-XG1)*YG4+(XG4-XG3)*YG1)*ZG2+
     *((XG2-XG4)*YG1+(XG1-XG2)*YG4+(XG4-XG1)*YG2)*ZG3+
     *((XG1-XG3)*YG2+(XG3-XG2)*YG1+(XG2-XG1)*YG3)*ZG4)
C
      IF(VOL.LE.0.)  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(VOL.LE.0.)  STOP
C
      A1 = XG2*YG3*ZG4-XG2*ZG3*YG4-XG3*YG2*ZG4+XG3*ZG2*YG4+
     *XG4*YG2*ZG3-XG4*ZG2*YG3
      A2 =-XG1*YG3*ZG4+XG1*ZG3*YG4+XG3*YG1*ZG4-XG3*ZG1*YG4-
     *XG4*YG1*ZG3+XG4*ZG1*YG3
      A3 = XG1*YG2*ZG4-XG1*ZG2*YG4-XG2*YG1*ZG4+XG2*ZG1*YG4+
     *XG4*YG1*ZG2-XG4*ZG1*YG2
      A4 =-XG1*YG2*ZG3+XG1*ZG2*YG3+XG2*YG1*ZG3-XG2*ZG1*YG3-
     *XG3*YG1*ZG2+XG3*ZG1*YG2
      B1 =-YG3*ZG4+ZG3*YG4+YG2*ZG4-ZG2*YG4-YG2*ZG3+ZG2*YG3
      B2 = YG3*ZG4-ZG3*YG4-YG1*ZG4+ZG1*YG4+YG1*ZG3-ZG1*YG3
      B3 =-YG2*ZG4+ZG2*YG4+YG1*ZG4-ZG1*YG4-YG1*ZG2+ZG1*YG2
      B4 = YG2*ZG3-ZG2*YG3-YG1*ZG3+ZG1*YG3+YG1*ZG2-ZG1*YG2
      C1 =-XG2*ZG4+XG2*ZG3+XG3*ZG4-XG3*ZG2-XG4*ZG3+XG4*ZG2
      C2 = XG1*ZG4-XG1*ZG3-XG3*ZG4+XG3*ZG1+XG4*ZG3-XG4*ZG1
      C3 =-XG1*ZG4+XG1*ZG2+XG2*ZG4-XG2*ZG1-XG4*ZG2+XG4*ZG1
      C4 = XG1*ZG3-XG1*ZG2-XG2*ZG3+XG2*ZG1+XG3*ZG2-XG3*ZG1
      D1 =-XG2*YG3+XG2*YG4+XG3*YG2-XG3*YG4-XG4*YG2+XG4*YG3
      D2 = XG1*YG3-XG1*YG4-XG3*YG1+XG3*YG4+XG4*YG1-XG4*YG3
      D3 =-XG1*YG2+XG1*YG4+XG2*YG1-XG2*YG4-XG4*YG1+XG4*YG2
      D4 = XG1*YG2-XG1*YG3-XG2*YG1+XG2*YG3+XG3*YG1-XG3*YG2
C
      DO 110  I=1,6
      DO 110  J=1,12
      B(I,J) = 0.
  110 CONTINUE
```

```
C
      B(1,1)  = B1
      B(1,4)  = B2
      B(1,7)  = B3
      B(1,10) = B4
      B(2,2)  = C1
      B(2,5)  = C2
      B(2,8)  = C3
      B(2,11) = C4
      B(3,3)  = D1
      B(3,6)  = D2
      B(3,9)  = D3
      B(3,12) = D4
      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
      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
      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.*VOL)
      BT(J,I) = B(I,J)
  130 CONTINUE
  120 CONTINUE
C
C     ELASTICITY MATRIX:
C
      DO 140 I = 1,6
      DO 150 J = 1,6
      C(I,J)  = 0.
  150 CONTINUE
  140 CONTINUE
      FAC = ELAS/((1.+PR)*(1.-2.*PR))
      C(1,1)  = FAC*(1-PR)
      C(1,2)  = FAC*PR
      C(1,3)  = FAC*PR
      C(2,1)  = C(1,2)
      C(2,2)  = C(1,1)
      C(2,3)  = FAC*PR
      C(3,1)  = C(1,3)
      C(3,2)  = C(2,3)
      C(3,3)  = C(1,1)
      C(4,4)  = FAC*(1.-2.*PR)/2.
      C(5,5)  = C(4,4)
      C(6,6)  = C(4,4)
C
C     ELEMENT STIFFNESS MATRIX:
C
      DO 100  I=1,6
```

```
      DO 100  J=1,12
      DUMA(I,J) = 0.
      DO 200  K=1,6
      DUMA(I,J) = DUMA(I,J) + C(I,K)*B(K,J)
  200 CONTINUE
  100 CONTINUE
C
      DO 300  I=1,12
      DO 300  J=1,12
      STRA(I,J) = 0.
      DO 400  K=1,6
      STRA(I,J) = STRA(I,J) + BT(I,K)*DUMA(K,J)
  400 CONTINUE
  300 CONTINUE
C
      DO 500  I=1,12
      DO 500  J=1,12
      STRA(I,J) = STRA(I,J)*VOL
  500 CONTINUE
C
C     ELEMENT NODAL FORCE DUE TO THERMAL EXPANSION:
C
      AL(1) = ALPHA
      AL(2) = ALPHA
      AL(3) = ALPHA
      AL(4) = 0.
      AL(5) = 0.
      AL(6) = 0.
      DO 600  I=1,6
      DUMB(I) = 0.
      DO 700  J=1,6
      DUMB(I) = DUMB(I) + C(I,J)*AL(J)
  700 CONTINUE
  600 CONTINUE
C
      DO 800  I=1,12
      FET(I) = 0.
      DO 900  J=1,6
      FET(I) = FET(I) + BT(I,J)*DUMB(J)
  900 CONTINUE
  800 CONTINUE
C
C     AVERAGE ELEMENT TEMPERATURE:
C
      TAVG = (TEMP(II) + TEMP(JJ) + TEMP(KK) + TEMP(LL))/4.
      FAC = (TAVG - TREF)*VOL
      DO 1000  I=1,12
      FET(I) = FET(I)*FAC
 1000 CONTINUE
C
C     ELEMENT BODY FORCE DUE TO CENTRIFUGAL FORCE
C     (NEGLECT GRAVITATIONAL FORCE):
C
      AVX =(XG1 + XG2 + XG3 + XG4)/4.
      AVY =(YG1 + YG2 + YG3 + YG4)/4.
C
      FACB1 = DENS*ANGVEL*ANGVEL*AVX*VOL/4.
      FACB2 = DENS*ANGVEL*ANGVEL*AVY*VOL/4.
C
      FEB(1) = FACB1
      FEB(2) = FACB2
      FEB(3) = 0.
      FEB(4) = FACB1
      FEB(5) = FACB2
      FEB(6) = 0.
      FEB(7) = FACB1
      FEB(8) = FACB2
      FEB(9) = 0.
      FEB(10)= FACB1
```

```
      FEB(11)= FACB2
      FEB(12)= 0.
C
C     ELEMENT NADAL FORCE DUE TO SURFACE FORCE BY UNIFORM PRESSURE
C
      DO 1200 IF = 1,12
      FES(IF) = 0.
 1200 CONTINUE
      IF(S(IE).EQ.0.) GO TO 1310
C
C     FIND AREA OF THE SURFACE THAT FACE TO PRESSURE
C     FROM AREA = SQRT(S(S-A)*(S-B)*(S-C))
C
      IIS = S(IE)
      DO 1210 ISE = 1,IIS
         IF(ISE.EQ.1) THEN
            IS = INTMATP1(IE,1)
            JS = INTMATP1(IE,2)
            KS = INTMATP1(IE,3)
           INP = P1(IE,2)
             P = P1(IE,1)
         ELSE
         ENDIF
         IF(ISE.EQ.2) THEN
            IS = INTMATP2(IE,1)
            JS = INTMATP2(IE,2)
            KS = INTMATP2(IE,3)
           INP = P2(IE,2)
             P = P2(IE,1)
         ELSE
         ENDIF
         IF(ISE.EQ.3) THEN
            IS = INTMATP3(IE,1)
            JS = INTMATP3(IE,2)
            KS = INTMATP3(IE,3)
           INP = P3(IE,2)
             P = P3(IE,1)
         ELSE
         ENDIF
C
      XS1 = COORD(IS,1)
      XS2 = COORD(JS,1)
      XS3 = COORD(KS,1)
      YS1 = COORD(IS,2)
      YS2 = COORD(JS,2)
      YS3 = COORD(KS,2)
      ZS1 = COORD(IS,3)
      ZS2 = COORD(JS,3)
      ZS3 = COORD(KS,3)
      XSN = COORD(INP,1)
      YSN = COORD(INP,2)
      ZSN = COORD(INP,3)
C
C CHECKING FOR NORMAL VECTOR OF PLANE THAT POINT FROM PLANE TO INP
C
      VOLP=1/6.*(((XS2-XS3)*YSN+(XSN-XS2)*YS3+(XS3-XSN)*YS2)*ZS1+
     *((XS1-XSN)*YS3+(XS3-XS1)*YSN+(XSN-XS3)*YS1)*ZS2+
     *((XS2-XSN)*YS1+(XS1-XS2)*YSN+(XSN-XS1)*YS2)*ZS3+
     *((XS1-XS3)*YS2+(XS3-XS2)*YS1+(XS2-XS1)*YS3)*ZSN)
      IF(VOLP.LE.0) THEN
         JSN = KS
         KSN = JS
         XS2 = COORD(JSN,1)
         XS3 = COORD(KSN,1)
         YS2 = COORD(JSN,2)
         YS3 = COORD(KSN,2)
         ZS2 = COORD(JSN,3)
         ZS3 = COORD(KSN,3)
      ENDIF
```

```
C
C     FIND NORMAL VECTOR OF SURFACE THAT FACE TO PRESSURE
C
      CX = (ZS2-ZS3)*YS1 + (ZS3-ZS1)*YS2 + (ZS1-ZS2)*YS3
      CY = (XS2-XS3)*ZS1 + (XS3-XS1)*ZS2 + (XS1-XS2)*ZS3
      CZ = (YS2-YS3)*XS1 + (YS3-YS1)*XS2 + (YS1-YS2)*XS3
      FACC = SQRT(CX*CX+CY*CY+CZ*CZ)
C
C     FIND  PX, PY, PZ
      PX = P*CX/FACC
      PY = P*CY/FACC
      PZ = P*CZ/FACC
C
      AA = XS2-XS1
      AB = YS2-YS1
      AC = ZS2-ZS1
      BA = XS3-XS1
      BB = YS3-YS1
      BC = ZS3-ZS1
      CA = XS3-XS2
      CB = YS3-YS2
      CC = ZS3-ZS2
C
      SA = SQRT(AA*AA+AB*AB+AC*AC)
      SB = SQRT(BA*BA+BB*BB+BC*BC)
      SC = SQRT(CA*CA+CB*CB+CC*CC)
      SL = (SA + SB + SC)/2.
      DELA = SL - SA
      DELB = SL - SB
      DELC = SL - SC
      SAREA = SQRT(SL*DELA*DELB*DELC)
C     CHECK
      IF(IS.EQ.II) IS = 1
      IF(IS.EQ.JJ) IS = 2
      IF(IS.EQ.KK) IS = 3
      IF(IS.EQ.LL) IS = 4
      IF(JS.EQ.II) JS = 1
      IF(JS.EQ.JJ) JS = 2
      IF(JS.EQ.KK) JS = 3
      IF(JS.EQ.LL) JS = 4
      IF(KS.EQ.II) KS = 1
      IF(KS.EQ.JJ) KS = 2
      IF(KS.EQ.KK) KS = 3
      IF(KS.EQ.LL) KS = 4
C
      NNODE = 4
      NDF = 3
      DO 1300 I = 1,NNODE
      IF((I.NE.IS).AND.(I.NE.JS).AND.(I.NE.KS)) GO TO 1300
C
      IRX = (I-1)*NDF + 1
      IRY = (I-1)*NDF + 2
      IRZ = (I-1)*NDF + 3
      FES(IRX) = PX*SAREA/3. + FES(IRX)
      FES(IRY) = PY*SAREA/3. + FES(IRY)
      FES(IRZ) = PZ*SAREA/3. + FES(IRZ)
 1300 CONTINUE
 1210 CONTINUE
C
C     FIND OF TOTAL FORCE OF ELEMENT
 1310 DO 1400 ISUM = 1,12
      FT(ISUM) = FEB(ISUM) + FES(ISUM) + FET(ISUM)
 1400 CONTINUE
C
C
C     ASSEMBLE THESE ELEMENT EQUATIONS INTO THE SYSTEM EQUATIONS:
C
      CALL ASSMBLE(   IE, INTMAT,  STRA, FT, SYSK, SYSF,
```

```
     *              MXPOI,  MXELE, MXHBW                )
C
 5000 CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------------------
C
      SUBROUTINE SOLVE(NROW, NHBW, GSTIF, XL, MXPOI, MXHBW)
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 REAL*8(A-H,O-Z)
C
      REAL  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.E-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) GO TO 30
      PIVOT2=GSTIF(I,MM+1)*PIVOT1
      XL(II)=XL(II)-XL(I)*PIVOT2
      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=1
      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 STRESS(NPOIN, NELEM, INTMAT, COORD, DISP, ELAS,  PR,
     *ALPHA,  TREF,   TEMP,  MXPOI, MXELE, SXX, SYY, SZZ, SXY, SXZ, SYZ
     *, ONE)
C
C     COMPUTE NODAL STRESS COMPONENTS FOR TETRAHEDRAL ELEMENTS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  COORD(MXPOI,3), TEMP(MXPOI), DISP(MXPOI*3),ONE(MXPOI)
      REAL  SXX(MXPOI),SYY(MXPOI),SZZ(MXPOI),SXY(MXPOI)
      REAL  SYZ(MXPOI),SXZ(MXPOI)
C
      REAL  C(6,6), B(6,12), EPS(6), UG(6), VG(6), WG(6)
C
      INTEGER  INTMAT(MXELE,4)
C
C     PRINT OUT ELEMENT STRESSES:
      WRITE(8,11) NELEM
   11 FORMAT( ' ELEMENTAL STRESS SOLUTIONS [', I5,']:',/,
     *        2X, 'ELEM', 11X, 'SXX', 11X, 'SYY', 11X, 'SZZ',
     *        11X, 'SXY', 11X, 'SXZ', 11X, 'SYZ' )
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)
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)
      ZG1 = COORD(II,3)
      ZG2 = COORD(JJ,3)
      ZG3 = COORD(KK,3)
      ZG4 = COORD(LL,3)
      VOL=1./6.*(((XG2-XG3)*YG4+(XG4-XG2)*YG3+(XG3-XG4)*YG2)*ZG1+
     *((XG1-XG4)*YG3+(XG3-XG1)*YG4+(XG4-XG3)*YG1)*ZG2+
     *((XG2-XG4)*YG1+(XG1-XG2)*YG4+(XG4-XG1)*YG2)*ZG3+
     *((XG1-XG3)*YG2+(XG3-XG2)*YG1+(XG2-XG1)*YG3)*ZG4)
C
```

```
      B1 =-YG3*ZG4+ZG3*YG4+YG2*ZG4-ZG2*YG4-YG2*ZG3+ZG2*YG3
      B2 = YG3*ZG4-ZG3*YG4-YG1*ZG4+ZG1*YG4+YG1*ZG3-ZG1*YG3
      B3 =-YG2*ZG4+ZG2*YG4+YG1*ZG4-ZG1*YG4-YG1*ZG2+ZG1*YG2
      B4 = YG2*ZG3-ZG2*YG3-YG1*ZG3+ZG1*YG3+YG1*ZG2-ZG1*YG2
      C1 =-XG2*ZG4+XG2*ZG3+XG3*ZG4-XG3*ZG2-XG4*ZG3+XG4*ZG2
      C2 = XG1*ZG4-XG1*ZG3-XG3*ZG4+XG3*ZG1+XG4*ZG3-XG4*ZG1
      C3 =-XG1*ZG4+XG1*ZG2+XG2*ZG4-XG2*ZG1-XG4*ZG2+XG4*ZG1
      C4 = XG1*ZG3-XG1*ZG2-XG2*ZG3+XG2*ZG1+XG3*ZG2-XG3*ZG1
      D1 =-XG2*YG3+XG2*YG4+XG3*YG2-XG3*YG4-XG4*YG2+XG4*YG3
      D2 = XG1*YG3-XG1*YG4-XG3*YG1+XG3*YG4+XG4*YG1-XG4*YG3
      D3 =-XG1*YG2+XG1*YG4+XG2*YG1-XG2*YG4-XG4*YG1+XG4*YG2
      D4 = XG1*YG2-XG1*YG3-XG2*YG1+XG2*YG3+XG3*YG1-XG3*YG2
C
      DO 110  I=1,6
      DO 110  J=1,12
      B(I,J) = 0.
  110 CONTINUE
C
      B(1,1)  = B1
      B(1,4)  = B2
      B(1,7)  = B3
      B(1,10)= B4
      B(2,2)  = C1
      B(2,5)  = C2
      B(2,8)  = C3
      B(2,11)= C4
      B(3,3)  = D1
      B(3,6)  = D2
      B(3,9)  = D3
      B(3,12)= D4
      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
      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
      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.*VOL)
  130 CONTINUE
  120 CONTINUE
C
C     ELASTICITY MATRIX:
C
      DO 140 I = 1,6
      DO 150 J = 1,6
      C(I,J) = 0.
  150 CONTINUE
  140 CONTINUE
      FAC = ELAS/((1.+PR)*(1.-2.*PR))
```

```
      C(1,1) = FAC*(1-PR)
      C(1,2) = FAC*PR
      C(1,3) = FAC*PR
      C(2,1) = C(1,2)
      C(2,2) = C(1,1)
      C(2,3) = FAC*PR
      C(3,1) = C(1,3)
      C(3,2) = C(2,3)
      C(3,3) = C(1,1)
      C(4,4) = FAC*(1.-2.*PR)/2.
      C(5,5) = C(4,4)
      C(6,6) = C(4,4)
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.
      DO 230  J=1,4
      J1 = (J-1)*3 + 1
      J2 = J1 + 1
      J3 = J1 + 2
      EPS(I) = EPS(I) + B(I,J1)*UG(J) + B(I,J2)*VG(J) + B(I,J3)*WG(J)
  230 CONTINUE
  220 CONTINUE
C
C     COMPUTE THERMAL STRAINS USING AVERAGE ELEMENT NODAL TEMPERATURES:
C
      TAVG = (TEMP(II) + TEMP(JJ) + TEMP(KK) + TEMP(LL))/4.
C
C     COMPUTE THE NET STRAINS:
C
      EPS(1) = EPS(1) - ALPHA*(TAVG - TREF)
      EPS(2) = EPS(2) - ALPHA*(TAVG - TREF)
      EPS(3) = EPS(3) - ALPHA*(TAVG - TREF)
C
C     COMPUTE THE ELEMENT STRESSES:
      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
      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
C
      ONE(II) = ONE(II) + 1.
      ONE(JJ) = ONE(JJ) + 1.
      ONE(KK) = ONE(KK) + 1.
      ONE(LL) = ONE(LL) + 1.
C
C     PRINT OUT ELEMENTAL STRESS SOLUTION
      WRITE(8,12) IE,SXXE,SYYE,SZZE,SXYE,SXZE,SYZE
   12 FORMAT(I6,6E14.6)
C
 1000 CONTINUE
C
C     PRINT OUT THESE NODAL STRESSES:
C
      DO 1100  I=1,NPOIN
      IF(ONE(I).EQ.0.)  WRITE(6,1200)  I
 1200 FORMAT(' *** WARNING ***  NO STRESS CONTRIBUTION AT NODE', I5)
      IF(ONE(I).EQ.0.)  ONE(I) = 1.
      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(8,1300)  NPOIN
 1300 FORMAT( '  NODAL STRESS SOLUTIONS [', I5,']:', /,
     *       2X, 'NODE', 11X, 'SXX', 11X, 'SYY', 11X, 'SZZ',
     *       11X, 'SXY', 11X, 'SXZ', 11X, 'SYZ' )
      DO 1400  I=1,NPOIN
      WRITE(8,1500)  I, SXX(I), SYY(I), SZZ(I), SXY(I), SXZ(I), SYZ(I)
 1500 FORMAT(I6, 6E14.6)
 1400 CONTINUE
C
C     PRINT OUT ELEMENT NODAL CONNECTION
      WRITE(8,*) 'ELEMENT NODAL CONNECTION'
      WRITE(8,13)
   13 FORMAT(3X,'ELE',8X,'I',3X,'J',5X,'K',4X,'L')
      DO 1600 IE = 1,NELEM
      WRITE(8,14) IE,INTMAT(IE,1),INTMAT(IE,2),INTMAT(IE,3),INTMAT(IE,4)
   14 FORMAT(I5,5X,4I5)
 1600 CONTINUE
C
      RETURN
      END
**************************************************************************
```

รายละเอียดของโปรแกรม TRAN3D

รายละเอียดของโปรแกรม TRAN3D

โปรแกรม TRAN3D จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วยโปรแกรมย่อยต่างๆทั้งหมดดังนี้

```
*********************************************************************
*          TRANSLATE OUTPUT FILE OF SOLID3D TO NASTRAN FORMAT       *
*                   AND CONCULATE VONMISES  STRESS                  *
*                                    MISS JUKSANEE VIRULSRI         *
*                                    FACULTY OF ENGINEERING         *
*                                    CHULALONGKORN UNIVERSITY       *
*********************************************************************
      PARAMETER (MXPOI=1666, MXELE=6760)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL  SXXEE(MXELE), SYYEE(MXELE), SZZEE(MXELE)
      REAL  SXYEE(MXELE), SYZEE(MXELE), SXZEE(MXELE)
C
      CHARACTER*20  NAME1, NAME2
C
      INTEGER INTMAT(MXELE,4)
      REAL    PSE1(MXELE),PSE2(MXELE),PSE3(MXELE),MPE(MXELE)
      REAL    LE1(MXELE),LE2(MXELE),LE3(MXELE),ME1(MXELE),ME2(MXELE)
      REAL    ME3(MXELE),NE1(MXELE),NE2(MXELE),NE3(MXELE),VONE(MXELE)
      REAL SXXE,SYYE,SZZE,SXYE,SXZE,SYZE,SPE1,SPE2,SPE3,DLE1,DLE2,DLE3
      REAL DME1,DME2,DME3,DNE1,DNE2,DNE3,MPRE,SVONE

C
   10 WRITE(6,15)
   15 FORMAT(/,'  TRANSLATE OUTPUT FILE OF STRES3D TO NASTRAN FORMAT'
     *,/,'  PLEASE INPUT THE OUTPUT FILE TO TRANSLATE:',/)
      READ(5, '(A)', ERR=10)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=10)
C
      READ(7,1)  TEXT
C
    1 FORMAT(20A4)
      READ(7,200)  NPOIN, NELEM
  200 FORMAT(T12,I5,T27,I5)
C
      IF(NPOIN.GT.MXPOI)  WRITE(6,20)  NPOIN
   20 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOI TO ', I5)
      IF(NPOIN.GT.MXPOI)  STOP
      IF(NELEM.GT.MXELE)  WRITE(6,30)  NELEM
   30 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)  STOP
C
      READ(7,1) TEXT
      READ(7,1) TEXT
      READ(7,1) TEXT
      READ(7,1) TEXT
C
C     PRINT DISPLACEMENT
C
   40 WRITE(6,45)
   45 FORMAT(/, ' PLEASE ENTER FILE NAME FOR DISPLACEMENT'
     *           ' AND STRESS SOLUTIONS OF NASTRAN FORMAT:',/  )
      READ(5, '(A)', ERR=40)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=40)
      WRITE(8,50)
   50 FORMAT(T1,'1 MSC/NASTRAN PAGE')
      WRITE(8,60) 0
   60 FORMAT(T1,I1)
```

```
      WRITE(8,*) 'D I S P L A C E M E N T'
      WRITE(8,*) 'POINT ID. TYPE          T1              T2               T3
     *            R1              R2              R3'
C
      DO 65 I = 1,NPOIN
      READ(7,*) IP, U, V, W, R1, R2, R3
      IF(I.NE.IP)  WRITE(6,75)  I
   75 FORMAT(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP)  STOP
      WRITE(8,70)  IP, U, V, W, R1, R2, R3
   70 FORMAT(I6, 5X, 'G', 5X, 6E14.6)
   65 CONTINUE
      WRITE(8,60) 1
C
      READ(7,1) TEXT
      READ(7,1) TEXT
      DO 300 IE = 1,NELEM
      READ(7,*) II,SXXEE(II),SYYEE(II),SZZEE(II),SXYEE(II),SXZEE(II)
     *          ,SYZEE(IE)
      IF(II.NE.IE)  WRITE(6,80)  IE
   80 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(II.NE.IE)  STOP
      SXXE = SXXEE(II)
      SYYE = SYYEE(II)
      SZZE = SZZEE(II)
      SXYE = SXYEE(II)
      SXZE = SXZEE(II)
      SYZE = SYZEE(II)
C
C     SOLVE FOR MEAN PRESSURE AND VON MISES STRESS OF EACH ELEMENT
      CALL VONM(SXXE,SYYE,SZZE,SXYE,SXZE,SYZE,SPE1,SPE2,SPE3,DLE1,DLE2,
     *          DLE3,DME1,DME2,DME3,DNE1,DNE2,DNE3,MPRE,SVONE          )
      PSE1(IE) = SPE1
      PSE2(IE) = SPE2
      PSE3(IE) = SPE3
      LE1(IE)  = DLE1
      LE2(IE)  = DLE2
      LE3(IE)  = DLE3
      ME1(IE)  = DME1
      ME2(IE)  = DME2
      ME3(IE)  = DME3
      NE1(IE)  = DNE1
      NE2(IE)  = DNE2
      NE3(IE)  = DNE3
      MPE(IE)  = MPRE
      VONE(IE) = SVONE
  300 CONTINUE
C
      READ(7,1) TEXT
      READ(7,1) TEXT
      DO 400 IP = 1,NPOIN
         READ(7,1) TEXT
  400 CONTINUE
C
      READ(7,1) TEXT
      READ(7,1) TEXT
      DO 500  IE=1,NELEM
      READ(7,*)  I, (INTMAT(I,J), J=1,4)
      IF(I.NE.IE)  WRITE(6,80)  IE
      IF(I.NE.IE)  STOP
  500 CONTINUE
C
C     PRINT  STRESS DISTRIBUTION
C
      WRITE(8,60) 0
      WRITE(8,90)
   90 FORMAT('                    S T R E S S E S   I N   T E T R A H E D
     *R O N   S O L I D   E L E M E N T S   ( C T E T R A )')
      WRITE(8,1000) 0
```

```
1000 FORMAT(T1,I1,T18,'CORNER         ------CENTER AND CORNER POINT STRES
    *SES---------           DIR.  COSINES        MEAN')
     WRITE(8,1010)
1010 FORMAT('  ELEMENT-ID    GRID-ID        NORMAL                SHEAR
    *              PRINCIPAL      -A-   -B-   -C-      PRESSURE        VON MIS
    *ES ')
     DO 600 IE = 1,NELEM
     WRITE(8,630) 0, IE
 630 FORMAT(T1,I1,T4,I8,T23,'0GRID CS  4 GP')
     WRITE(8,640) 0,SXXEE(IE),SXYEE(IE),PSE1(IE),LE1(IE),LE2(IE),
    * LE3(IE),MPE(IE),VONE(IE)
 640 FORMAT(T1,I1,T18,'CENTER  X',T29,E13.6,T44,'XY',T48,E13.6,T64,'A',
    *T67,E13.6,T82,'LX',T84,F5.2,T89,F5.2,T94,F5.2,T101,E13.6,T117,
    *E13.6)
     WRITE(8,650) SYYEE(IE),SYZEE(IE),PSE2(IE),ME1(IE),ME2(IE),ME3(IE)
 650 FORMAT(T26,'Y',T29,E13.6,T44,'YZ',T48,E13.6,T64,'B',T67,E13.6,T82,
    *'LY',T84,F5.2,T89,F5.2,T94,F5.2)
     WRITE(8,660) SZZEE(IE),SXZEE(IE),PSE3(IE),NE1(IE),NE2(IE),NE3(IE)
 660 FORMAT(T26,'Z',T29,E13.6,T44,'ZX',T48,E13.6,T64,'C',T67,E13.6,T82,
    *'LZ',T84,F5.2,T89,F5.2,T94,F5.2)
       DO 620 IPE =1,4
          II = INTMAT(IE,IPE)
     WRITE(8,670) 0,II,SXXEE(IE),SXYEE(IE),PSE1(IE),LE1(IE),LE2(IE),
    *LE3(IE),MPE(IE),VONE(IE)
 670 FORMAT(T1,I1,T16,I8,T26,'X',T29,E13.6,T44,'XY',T48,E13.6,T64,'A',
    *T67,E13.6,T82,'LX',T84,F5.2,T89,F5.2,T94,F5.2,T101,E13.6,T117,
    *E13.6)
     WRITE(8,650) SYYEE(IE),SYZEE(IE),PSE2(IE),ME1(IE),ME2(IE),ME3(IE)
     WRITE(8,660) SZZEE(IE),SXZEE(IE),PSE3(IE),NE1(IE),NE2(IE),NE3(IE)
 620 CONTINUE
 600 CONTINUE
     WRITE(8,60) 1
C
      STOP
      END
C-----------------------------------------------------------------------
C
      SUBROUTINE VONM(SSXX,SSYY,SSZZ,SSXY,SSXZ,SSYZ,SP1,SP2,SP3,DL1,DL2,
     *                DL3,DM1,DM2,DM3,DN1,DN2,DN3,MPR,SVONM              )
C
      REAL SSXX,SSYY,SSZZ,SSXY,SSXZ,SSYZ,SP1,SP2,SP3,DL1,DL2,DL3
      REAL DM1,DM2,DM3,DN1,DN2,DN3,MPR,SVONM,ANG
      SI1 = SSXX + SSYY + SSZZ
      SI2 = SSXX*SSYY+SSXX*SSZZ+SSYY*SSZZ-SSXY*SSXY-SSYZ*SSYZ-SSXZ*SSXZ
      SI3 = SSXX*SSYY*SSZZ+2.*SSXY*SSYZ*SSXZ-SSYY*SSXZ*SSXZ
     *      -SSXX*SSYZ*SSYZ-SSZZ*SSXY*SSXY
C
C  DEFINE
C
      a = SI1*SI1/3. - SI2
      b = -2.*(SI1/3.)**3 + (SI1*SI2)/3. - SI3
      c = 2.*SQRT(a/3.)
      ang = 1/3.*ACOS(-3.*b/(a*c))
c      test = abs(-3.*b/(a*c))
c      if(test.gt.1.) write(8,*) 'error with angle value'
C
C  THE PRINCIPLE STRESSES ARE GIVEN BY
C
      SP1 = SI1/3. + c*COS(ang)
      SP2 = SI1/3. + c*COS(ang+2.094395102)
      SP3 = SI1/3. + c*COS(ang+4.188790205)
C
C
C  FIND THE PRINCIPLE DIRECTION
      A1=-((SSZZ-SP1)-(SSYZ*SSYZ)/(SSYY-SP1))/
     *    (SSXZ-(SSXY*SSYZ)/(SSYY-SP1))
      B1= (SSYZ-(SSXY*SSXZ)/(SSXX-SP1))/
     *    ((SSYY-SP1)-(SSXY*SSXY)/(SSXX-SP1))
      DN1 = SQRT(1./(A1*A1+B1*B1+1.))
```

```
      DL1 = A1*DN1
      DM1 = B1*DN1
      A2=-((SSZZ-SP2)-(SSYZ*SSYZ)/(SSYY-SP2))/
     *     (SSXZ-(SSXY*SSYZ)/(SSYY-SP2))
      B2= (SSYZ-(SSXY*SSXZ)/(SSXX-SP2))/
     *     ((SSYY-SP2)-(SSXY*SSXY)/(SSXX-SP2))
      DN2 = SQRT(1./(A2*A2+B2*B2+1.))
      DL2 = A2*DN2
      DM2 = B2*DN2
      A3=-((SSZZ-SP3)-(SSYZ*SSYZ)/(SSYY-SP3))/
     *     (SSXZ-(SSXY*SSYZ)/(SSYY-SP3))
      B3= (SSYZ-(SSXY*SSXZ)/(SSXX-SP3))/
     *     ((SSYY-SP3)-(SSXY*SSXY)/(SSXX-SP3))
      DN3 = SQRT(1./(A3*A3+B3*B3+1.))
      DL3 = A3*DN3
      DM3 = B3*DN3
C  FIND VON MISES STRESS
  200 SVONM = 1/SQRT(2.)*SQRT((SP1-SP2)**2+(SP2-SP3)**2+(SP3-SP1)**2)
      MPR  = -SI1/3.
C
      RETURN
      END
************************************************************************
```

ประวัติผู้วิจัย

นางสาวจักษณี วิรุพห์ศรี เกิดเมื่อวันที่ 28 เดือนมิถุนายน พุทธศักราช 2517 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล ภาควิชาวิชาวิศวกรรมเครื่อง-กล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2537 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2538