รายการอ้างอิง

1. ปราโมทย์ เดชะอำไพ. ระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อการคำนวณพลศาสตร์ของไหล. พิมพ์ครั้งที่ 1. กรุงเทพฯ : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2545.

2. จิตติน ตรีพุทธรัตน์. การศึกษาการไหลผ่านวัตถุด้วยวิธีการไฟไนต์เอลิเมนต์. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2539.

3. Fox, R. W. and McDonald, A. T. Introduction to Fluid Mechanics. Fourth Edition. New York : John Wiley & Sons, Inc., 1994.

4. White, F. M. Viscous Fluid Mechanics. Second Edition. McGraw-Hill Series in Mechanical Engineering. Singapore : McGraw-Hill, 1991.

5. Versteeg, H. K. and Malalasekera, W. An Introduction to Computational Fluid Dynamics: The Finite Volume Method. London : Longman Scientific & Technical, 1995.

6. Munson, B. R., Young, D. F. and Okiishi, T. H. Fundamentals of Fluid Mechanics. Third Edition. New York : John Wiley & Sons, Inc., 1998.

7. De Vries, G. and Norrie, D. H. The applications of the finite element technique to potential flow problems. Trans. ASME, Series E: Journal of Applied Mechanics 38 (1971) : 798-802.

8. ปราโมทย์ เดชะอำไพ. ไฟไนต์เอลิเมนต์ในงานวิศวกรรม. พิมพ์ครั้งที่ 2. กรุงเทพฯ : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2542.

9. Huebner, K. H. and Thornton, E. A. The Finite Element Method for Engineers. Third Edition. New York : John Wiley & Sons, Inc., 1995.

10. Zienkiewicz, O. C. and Taylor, R. L. The Finite Element Method. Fourth Edition. New York : McGraw-Hill, 1991.

11. Yamada, Y., Ito, K., Yokouchi, Y., Tamano, T. and Ohtsubo, T. Finite element analysis of steady fluid and metal flow. Finite Elements in Fluids (1975) : 73-94.

12. Hood, P. and Taylor, C. Navier-Stokes equations using mixed interpolation. Finite Element Method in Flow Problems (1974) : 121-132.

13. Streeter, V. L. and Wylie, B. E. Fluid Mechanics. McGraw-Hill international, Singapore : McGraw-Hill, 1987.

14. Schlichting, H. Boundary Layer Theory. Seventh Edition. New York : McGraw-Hill, 1978.

15. Ramaswamy, B. and Jue, T. C. A segregated finite element formulation for Navier-Stokes equations under laminar conditions. Finite Element Analysis and Design 9 (1989) : 257-270.

16. Burggraf, O. R. Analytical and numerical studies of the structure of steady separated flows. Journal of Fluid Mechanics 24 (1966) : 113-151.

17. Dechaumphai, P. Adaptive finite element technique for heat transfer problems. Energy, Heat and Mass Transfer 17 (1995) : 87-94.

18. Dechaumphai, P. and Janphaisaeng, P. Adaptive finite element technique for high-speed compressible flows. Thammasat International Journal of Science and Technology Vol. 3. No. 1, 1998.

19. สุพัฒนพงศ์ สิขาบัณฑิต. เทคนิคการปรับขนาดไฟไนต์เอลิเมนต์เพื่อการวิเคราะห์การไหลแบบหนืด. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2541.

ภาคผนวก

ภาคผนวก ก.

รายละเอียดโปรแกรม POTEN2D

รายละเอียดโปรแกรม POTEN2D จะมีรายละเอียดเริ่มจากโปรแกรมหลัก และตามด้วยโปรแกรมย่อยต่าง ๆ ดังนี้

```
C     PROGRAM POTEN2D
C
C     A FINITE ELEMENT COMPUTER PROGRAM FOR SOLVING PARTIAL
C     DIFFERENTIAL EQUATION IN THE FORM OF POISSON'S EQUATION
C     FOR TWO-DIMENSIONAL STEADY-STATE INVISCID INCOMPRESSIBLE FLOW.
C
C     THE VALUES DECLARED IN THE PARAMETER STATEMENT BELOW SHOULD
C     BE ADJUSTED ACCORDING TO THE SIZE OF THE PROBLEMS AND TYPES
C     OF COMPUTERS:
C         MXPOI = MAXIMUM NUMBER OF NODES IN THE MODEL
C         MXELE = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C
      PARAMETER (MXPOI=640, MXELE=1136, MXREFP=1)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  TEXT(20)
      DIMENSION  COORD(MXPOI,2), POTEN(MXPOI), VEL(MXPOI), P(MXPOI)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSR(MXPOI), U(MXPOI), V(MXPOI)
      DIMENSION  UE(MXELE), VE(MXELE)
      CHARACTER*20  NAME1, NAME2
C
      INTEGER  INTMAT(MXELE,3), IBC(MXPOI)
C
C     SELECT TYPE ANALYSIS OF POTENTIAL FLOW:
C
   10 WRITE(6,20)
   20 FORMAT(/, '  *** POTENTIAL FLOW ANALYSIS TYPE IN 2D ***',/,
     *          '      1.- STREAM FUNCTION   ',/,
     *          '      2.- POTENTIAL FUNCTION',//,
     *          '          OPTION ?: ',$)
      READ(*,*)  ICASE
C
   30 WRITE(6,40)
   40 FORMAT(/, ' PLEASE ENTER THE INPUT FILE NAME:')
      READ(5, '(A)', ERR=30)  NAME1
      OPEN(UNIT=7, FILE=NAME1, STATUS='OLD', ERR=30)
C
C     READ TITLE OF COMPUTATION:
C
      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, NREFP
      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(6,115)  NELEM
  115 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)  STOP
      IF(NREFP.GT.MXREFP)  WRITE(6,120)
  120 FORMAT(/,' PLEASE CHECK THE PARAMETER NREFP ')
      IF(NREFP.GT.MXREFP)  STOP
C
C     READ FLUID PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  DEN, GRA
C
C     READ NODAL COORDINATES, BOUNDARY CONDITIONS, THEIR VALUES:
C     REQUIREMENT:  MAIN NODES MUST BE NUMBERED FIRST
C
      READ(7,1)  TEXT
      DO 130  IP=1,NPOIN
      READ(7,*)  I, IBC(I), (COORD(I,K), K=1,2), POTEN(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
      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
      WRITE(6,160)
  160 FORMAT(/,' THE F.E. MODEL INCLUDES THE FOLLOWING',
     *         ' POTENTIAL FLOW')
      READ(7,1)  TEXT
      DO 170  IP=1,NREFP
      READ(7,*)  IREFP
  170 CONTINUE
C
      NEQ  = NPOIN
      DO 180  I=1,NEQ
      SYSR(I) = 0.
  180 CONTINUE
      DO 190  I=1,NEQ
      DO 190  J=1,NEQ
      SYSK(I,J) = 0.
  190 CONTINUE
C
      WRITE(6,200)  NPOIN, NELEM
  200 FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I5,
     *         ' NODES AND', I5,' ELEMENTS ***')
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     FORM UP SYSTEM EQUATIONS
C
      WRITE(6,210)
  210 FORMAT(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *         ' ASSEMBLING ELEMENT EQUATIONS ***'     )
      CALL TRI(NELEM, INTMAT, COORD, SYSK, SYSR, MXPOI, MXELE)
C
      WRITE(6,220)
  220 FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS OF NODAL',
     *         ' POTENTIAL FLOW ***'                          )
      CALL APPLYBC(NPOIN, IBC, POTEN, SYSK, SYSR, MXPOI)
C
      WRITE(6,230)
  230 FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS',
```

```
     *         ' FOR POTENTIAL FLOW SOLUTIONS ***'              )
      WRITE(6,240)  NEQ
  240 FORMAT(5X,'( TOTAL OF', I5,' EQUATIONS TO BE SOLVED )')
      CALL GAUSS(NEQ, SYSK, SYSR, POTEN, MXPOI)
      CALL CVEL(NPOIN, NELEM, INTMAT, COORD, ICASE,
     *          MXPOI, MXELE, POTEN, U, V, UE, VE)
      CALL PRESS(NPOIN, COORD, IREFP, MXPOI, DEN, GRA,
     *           U, V, VEL, P)
C
C     PRINT OUT NODAL POTENTIAL FLOW SOLUTIONS:
C
  250 WRITE(6,260)
  260 FORMAT(/, ' PLEASE ENTER FILE NAME FOR POTENTIAL FLOW'
     *          ' SOLUTIONS:'                              )
      READ(5, '(A)', ERR=250)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=250)
      WRITE(8,270)  NPOIN
  270 FORMAT('  NODAL POTENTIAL FLOW SOLUTIONS [',I5,']:',
     *       //, 2X, 'NODE', 3X, 'POTENTIAL FLOW', //      )
      DO 280  IP=1,NPOIN
      WRITE(8,290)  IP, POTEN(IP), U(IP), V(IP), VEL(IP), P(IP)
  290 FORMAT(I6, E15.6, E15.6, E15.6, E15.6, E15.6)
  280 CONTINUE
C
C     PRINT OUT VELOCITY OF ELEMENT SOLUTIONS:
C
      WRITE(8,300)  NELEM
  300 FORMAT(//, '  VELOCITY OF ELEMENT SOLUTIONS [',I5,']:',
     *       //, 2X, 'ELEMENT', 11X, 'CONNECTION', 16X, 'U', 16X, 'V',
     *    //)
      DO 310  IE=1,NELEM
      WRITE(8,320)  IE, (INTMAT(IE,J), J=1,3), UE(IE), VE(IE)
  320 FORMAT(I6, I10, I10, I10, E16.6, E16.6)
  310 CONTINUE
C
      STOP
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NPOIN, IBC, POTEN, SYSK, SYSR, MXPOI)
C
C     APPLY POTENTIAL FLOW BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C           0 = FREE TO CHANGE (TO BE COMPUTED)
C           1 = FIXED AS SPECIFIED
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSR(MXPOI), POTEN(MXPOI)
C
      INTEGER  IBC(MXPOI)
C
      DO 100  IEQ=1,NPOIN
      IF(IBC(IEQ).EQ.0)  GO TO 100
C
      DO 200  IR=1,NPOIN
      IF(IR.EQ.IEQ)  GO TO 200
      SYSR(IR) = SYSR(IR) - SYSK(IR,IEQ)*POTEN(IEQ)
      SYSK(IR,IEQ) = 0.
  200 CONTINUE
C
      DO 300  IC=1,NPOIN
      SYSK(IEQ,IC) = 0.
  300 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = POTEN(IEQ)
C
```

```
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ASSMBLE(IE, INTMAT, AKC, SYSK, SYSR, MXPOI, MXELE)
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  AKC(3,3)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSR(MXPOI)
C
      INTEGER  INTMAT(MXELE,3)
C
      NNODE = 3
C
      DO 100  IR=1,NNODE
      DO 200  IC=1,NNODE
      IROW = INTMAT(IE,IR)
      ICOL = INTMAT(IE,IC)
      SYSK(IROW,ICOL) = SYSK(IROW,ICOL) + AKC(IR,IC)
  200 CONTINUE
      SYSR(IROW) = SYSR(IROW)
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
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-----------------------------------------------------------------------
C
      SUBROUTINE TRI(NELEM, INTMAT, COORD, SYSK, SYSR, MXPOI, MXELE)
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     UP SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), SYSK(MXPOI,MXPOI), SYSR(MXPOI)
      DIMENSION  AKC(3,3), B(2,3), BT(3,2)
C
      INTEGER  INTMAT(MXELE,3)
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  IE=1,NELEM
C
C     FIND ELEMENT LOCAL COORDINATES:
C
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      AREA= 0.5*(XG2*(YG3-YG1) + XG1*(YG2-YG3) + XG3*(YG1-YG2))
      IF(AREA.LE.0.)  WRITE(6,5)  IE
    5 FORMAT(/,'  !!! ERROR !!!  ELEMENT NO.', I5,
     *         ' HAS NEGATIVE OR ZERO AREA ', /,
     *         '  --- CHECK F.E. MODEL FOR NODAL COORDINATES',
     *         ' AND ELEMENT NODAL CONNECTIONS ---'          )
      IF(AREA.LE.0.)  STOP
C
      B1 = YG2 - YG3
      B2 = YG3 - YG1
      B3 = YG1 - YG2
      C1 = XG3 - XG2
      C2 = XG1 - XG3
      C3 = XG2 - XG1
C
      DO 10  I=1,2
      DO 10  J=1,3
      B(I,J) = 0.
   10 CONTINUE
C
      B(1,1) = B1
      B(1,2) = B2
      B(1,3) = B3
      B(2,1) = C1
      B(2,2) = C2
      B(2,3) = C3
C
```

```
      DO 20  I=1,2
      DO 30  J=1,3
      B(I,J)  = B(I,J)/(2.*AREA)
      BT(J,I) = B(I,J)
   30 CONTINUE
   20 CONTINUE
C
C     ELEMENT [K] MATRIX:
C
      DO 100  I=1,3
      DO 100  J=1,3
      AKC(I,J) = 0.
      DO 110  K=1,2
      AKC(I,J) = AKC(I,J) + BT(I,K)*B(K,J)
  110 CONTINUE
      AKC(I,J) = AREA*AKC(I,J)
  100 CONTINUE
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
      CALL ASSMBLE(IE, INTMAT, AKC, SYSK, SYSR, MXPOI, MXELE)
C
  500 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CVEL(NPOIN, NELEM, INTMAT, COORD, ICASE,
     *                MXPOI, MXELE, POTEN, U, V, UE, VE)
C
C     CALCULATE VELOCITY OF ALL ELEMENT FORM NODAL SOLUTIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), POTEN(MXPOI), UE(MXELE), VE(MXELE)
      DIMENSION  U(MXPOI), V(MXPOI), ONE(MXPOI)
C
      INTEGER  INTMAT(MXELE,3)
C
      DO 100 I=1,NELEM
      UE(I) = 0.
      VE(I) = 0.
  100 CONTINUE
      DO 150 I=1,NPOIN
      U(I)   = 0.
      V(I)   = 0.
      ONE(I) = 0.
  150 CONTINUE
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  IE=1,NELEM
C
C     FIND ELEMENT LOCAL COORDINATES:
C
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
```

```
      AREA= 0.5*(XG2*(YG3-YG1) + XG1*(YG2-YG3) + XG3*(YG1-YG2))
      IF(AREA.LE.0.)  WRITE(6,5)  IE
    5 FORMAT(/,'  !!! ERROR !!!  ELEMENT NO.', I5,
     *         ' HAS NEGATIVE OR ZERO AREA ', /,
     *         '  --- CHECK F.E. MODEL FOR NODAL COORDINATES',
     *         ' AND ELEMENT NODAL CONNECTIONS ---'          )
      IF(AREA.LE.0.)  STOP
C
      B1 = YG2 - YG3
      B2 = YG3 - YG1
      B3 = YG1 - YG2
      C1 = XG3 - XG2
      C2 = XG1 - XG3
      C3 = XG2 - XG1
C
      POTEN1 = POTEN(II)
      POTEN2 = POTEN(JJ)
      POTEN3 = POTEN(KK)
C
C     COMPUTE THE ELEMENT VELOCITY FOR STREAM FUNCTION:
C
      IF(ICASE.EQ.2) GOTO 300
      UE(IE) = ((C1*POTEN1)+(C2*POTEN2)+(C3*POTEN3))/(2.*AREA)
      VE(IE) = ((-B1*POTEN1)+(-B2*POTEN2)+(-B3*POTEN3))/(2.*AREA)
  300 CONTINUE
C
C     COMPUTE THE ELEMENT VELOCITY FOR POTENTIAL FUNCTION:
C
      IF(ICASE.EQ.1) GOTO 400
      UE(IE) = ((-B1*POTEN1)+(-B2*POTEN2)+(-B3*POTEN3))/(2.*AREA)
      VE(IE) = ((-C1*POTEN1)+(-C2*POTEN2)+(-C3*POTEN3))/(2.*AREA)
  400 CONTINUE

C
C     COMPUTE NODAL VELOCITIES FROM ELEMENT VELOCITY:
C
      U(II) = U(II) + UE(IE)
      U(JJ) = U(JJ) + UE(IE)
      U(KK) = U(KK) + UE(IE)
      V(II) = V(II) + VE(IE)
      V(JJ) = V(JJ) + VE(IE)
      V(KK) = V(KK) + VE(IE)
      ONE(II) = ONE(II) + 1.
      ONE(JJ) = ONE(JJ) + 1.
      ONE(KK) = ONE(KK) + 1.
C
  500 CONTINUE
C
      DO 600  I=1,NPOIN
      IF(ONE(I).EQ.0.)  WRITE(6,650)  I
  650 FORMAT(' *** WARNING ***  NO VELOCITY CONTRIBUTION AT NODE', I5)
      IF(ONE(I).EQ.0.)  ONE(I) = 1.
      U(I) = U(I)/ONE(I)
      V(I) = V(I)/ONE(I)
  600 CONTINUE
C
C     ROUND-OFF SOLUTION VALUES FOR OUTPUT:
C
      ROFF = 1.E-6
      DO 700  IEQ=1,NPOIN
      VPOTEN = POTEN(IEQ)
      VALUEU  =  U(IEQ)
      VALUEV  =  V(IEQ)
      IF(ABS(VPOTEN).LT.ROFF) POTEN(IEQ) = 0.
      IF(ABS(VALUEU).LT.ROFF)     U(IEQ) = 0.
      IF(ABS(VALUEV).LT.ROFF)     V(IEQ) = 0.
```

```
  700 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PRESS(NPOIN, COORD, IREFP, MXPOI, DEN, GRA,
     *                 U, V, VEL, P)
C
C     CALCULATE PRESSURE OF ALL NODAL SOLUTIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,2), P(MXPOI)
      DIMENSION  U(MXPOI), V(MXPOI), VEL(MXPOI)
C
      DO 100 I=1,NPOIN
      P(I)   = 0.
  100 CONTINUE
C
      VREF = SQRT((U(IREFP)*U(IREFP))+(V(IREFP)*V(IREFP)))
      YREF = (COORD(IREFP,2))
C
C     COMPUTE THE NODAL PRESSURE:
C
      CONST = ((VREF*VREF)/(2.*GRA))+YREF
      DO 200 I=1,NPOIN
      H = COORD(I,2)
      VEL(I) = SQRT((U(I)*U(I))+(V(I)*V(I)))
      P(I) = DEN*GRA*(CONST-(((VEL(I)*VEL(I))/(2.*GRA))+H))
  200 CONTINUE
C
C     ROUND-OFF SOLUTION VALUES FOR OUTPUT:
C
      ROFF = 1.E-6
      DO 300  IEQ=1,NPOIN
      VALUEP   = P(IEQ)
      VALUEVEL = VEL(IEQ)
      IF(ABS(VALUEP).LT.ROFF)      P(IEQ) = 0.
      IF(ABS(VALUEVEL).LT.ROFF)  VEL(IEQ) = 0.
  300 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
```

ภาคผนวก ข.

รายละเอียดโปรแกรม POTEN3D

รายละเอียดโปรแกรม POTEN3D จะมีรายละเอียดเริ่มจากโปรแกรมหลัก และตามด้วยโปรแกรมย่อยต่าง ๆ ดังนี้

```
C     PROGRAM POTEN3D
C
C     A FINITE ELEMENT COMPUTER PROGRAM FOR SOLVING PARTIAL
C     DIFFERENTIAL EQUATION IN THE FORM OF POISSON'S EQUATION
C     FOR THREE-DIMENSIONAL STEADY-STATE INVISCID INCOMPRESSIBLE FLOW.
C
C     THE VALUES DECLARED IN THE PARAMETER STATEMENT BELOW SHOULD
C     BE ADJUSTED ACCORDING TO THE SIZE OF THE PROBLEMS AND TYPES
C     OF COMPUTERS:
C         MXPOI = MAXIMUM NUMBER OF NODES IN THE MODEL
C         MXELE = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C
      PARAMETER (MXPOI=1490, MXELE=5673, MXREFP=1)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  TEXT(20)
      DIMENSION  COORD(MXPOI,3), POTEN(MXPOI), VEL(MXPOI)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSR(MXPOI)
      DIMENSION  U(MXPOI), V(MXPOI), W(MXPOI), P(MXPOI)
      DIMENSION  UE(MXELE), VE(MXELE), WE(MXELE)
      CHARACTER*20  NAME1, NAME2
C
      INTEGER  INTMAT(MXELE,4), IBC(MXPOI)
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
C     READ TITLE OF COMPUTATION:
C
      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, NREFP
      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(6,115)  NELEM
  115 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO ', I5)
      IF(NELEM.GT.MXELE)  STOP
      IF(NREFP.GT.MXREFP)  WRITE(6,120)
  120 FORMAT(/,' PLEASE CHECK THE PARAMETER NREFP ')
      IF(NREFP.GT.MXREFP)  STOP
C
```

```
C     READ FLUID PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  DEN, GRA
C
C     READ NODAL COORDINATES, BOUNDARY CONDITIONS, THEIR VALUES:
C     REQUIREMENT:  MAIN NODES MUST BE NUMBERED FIRST
C
      READ(7,1)  TEXT
      DO 130  IP=1,NPOIN
      READ(7,*)  I, IBC(I), (COORD(I,K), K=1,3), POTEN(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
      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
      WRITE(6,160)
  160 FORMAT(/,' THE F.E. MODEL INCLUDES THE FOLLOWING',
     *          ' POTENTIAL FLOW')
      READ(7,1)  TEXT
      DO 170  IP=1,NREFP
      READ(7,*)  IREFP
  170 CONTINUE
C
      NEQ  = NPOIN
      DO 180  I=1,NEQ
      SYSR(I) = 0.
  180 CONTINUE
      DO 190  I=1,NEQ
      DO 190  J=1,NEQ
      SYSK(I,J) = 0.
  190 CONTINUE
C
      WRITE(6,200)  NPOIN, NELEM
  200 FORMAT(/,' *** THE FINITE ELEMENT MODEL CONSISTS OF', I5,
     *          ' NODES AND', I5,' ELEMENTS ***')
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     FORM UP SYSTEM EQUATIONS
C
      WRITE(6,210)
  210 FORMAT(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *          ' ASSEMBLING ELEMENT EQUATIONS ***'     )
      CALL TET(NELEM, INTMAT, COORD, SYSK, SYSR, MXPOI, MXELE)
C
      WRITE(6,220)
  220 FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS OF NODAL',
     *          ' POTENTIAL FLOW ***'                          )
      CALL APPLYBC(NPOIN, IBC, POTEN, SYSK, SYSR, MXPOI)
C
      WRITE(6,230)
  230 FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQUATIONS',
     *          ' FOR POTENTIAL FLOW SOLUTIONS ***'              )
      WRITE(6,240)  NEQ
  240 FORMAT(5X,'( TOTAL OF', I5,' EQUATIONS TO BE SOLVED )')
      CALL GAUSS(NEQ, SYSK, SYSR, POTEN, MXPOI)
      CALL CVEL(NPOIN, NELEM, INTMAT, COORD,
     *          MXPOI, MXELE, POTEN, U, V, W, UE, VE, WE)
      CALL PRESS(NPOIN, COORD, IREFP, MXPOI, DEN, GRA,
     *           U, V, W, VEL, P)
```

```
C
C     PRINT OUT NODAL POTENTIAL FLOW SOLUTIONS:
C
  250 WRITE(6,260)
  260 FORMAT(/, ' PLEASE ENTER FILE NAME FOR POTENTIAL FLOW'
     *          ' SOLUTIONS:'                              )
      READ(5, '(A)', ERR=250)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=250)
      WRITE(8,270)  NPOIN
  270 FORMAT('  NODAL POTENTIAL FLOW SOLUTIONS [',I5,']:',
     *       //, 2X, 'NODE', 3X, 'POTENTIAL FLOW', //      )
      DO 280  IP=1,NPOIN
      WRITE(8,290)  IP, POTEN(IP), U(IP), V(IP), W(IP), VEL(IP), P(IP)
  290 FORMAT(I6, E15.6, E15.6, E15.6, E15.6, E15.6, E15.6)
  280 CONTINUE
C
C     PRINT OUT VELOCITY OF ELEMENT SOLUTIONS:
C
      WRITE(8,300)  NELEM
  300 FORMAT(//, '  VELOCITY OF ELEMENT SOLUTIONS [',I5,']:',
     *       //, 2X, 'ELEMENT', 16X, 'CONNECTION', 21X, 'U', 15X, 'V',
     *    15X, 'W', //
                          )
      DO 310  IE=1,NELEM
      WRITE(8,320)  IE, (INTMAT(IE,J), J=1,4), UE(IE), VE(IE), WE(IE)
  320 FORMAT(I6, I10, I10, I10, I10, E16.6, E16.6, E16.6)
  310 CONTINUE
C
      STOP
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NPOIN, IBC, POTEN, SYSK, SYSR, MXPOI)
C
C     APPLY POTENTIAL FLOW BOUNDARY CONDITIONS WITH CONDITION CODES OF:
C           0 = FREE TO CHANGE (TO BE COMPUTED)
C           1 = FIXED AS SPECIFIED
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSR(MXPOI), POTEN(MXPOI)
C
      INTEGER  IBC(MXPOI)
C
      DO 100  IEQ=1,NPOIN
      IF(IBC(IEQ).EQ.0)  GO TO 100
C
      DO 200  IR=1,NPOIN
      IF(IR.EQ.IEQ)  GO TO 200
      SYSR(IR) = SYSR(IR) - SYSK(IR,IEQ)*POTEN(IEQ)
      SYSK(IR,IEQ) = 0.
  200 CONTINUE
C
      DO 300  IC=1,NPOIN
      SYSK(IEQ,IC) = 0.
  300 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = POTEN(IEQ)
C
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
```

```
      SUBROUTINE ASSMBLE(IE, INTMAT, AKC, SYSK, SYSR, MXPOI, MXELE)
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  AKC(4,4)
      DIMENSION  SYSK(MXPOI,MXPOI), SYSR(MXPOI)
C
      INTEGER  INTMAT(MXELE,4)
C
      NNODE = 4
C
      DO 100  IR=1,NNODE
      DO 200  IC=1,NNODE
      IROW = INTMAT(IE,IR)
      ICOL = INTMAT(IE,IC)
      SYSK(IROW,ICOL) = SYSK(IROW,ICOL) + AKC(IR,IC)
  200 CONTINUE
      SYSR(IROW) = SYSR(IROW)
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
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-----------------------------------------------------------------------
C
      SUBROUTINE TET(NELEM, INTMAT, COORD, SYSK, SYSR, MXPOI, MXELE)
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     UP SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,3), SYSK(MXPOI,MXPOI), SYSR(MXPOI)
      DIMENSION  AKC(4,4), B(3,4), BT(4,3)
C
      INTEGER  INTMAT(MXELE,4)
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  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 = ((XG2*YG3*ZG4)-(XG2*ZG3*YG4)-(XG3*YG2*ZG4)+(XG3*ZG2*YG4)
     *      +(XG4*YG2*ZG3)-(XG4*ZG2*YG3)-(XG1*YG3*ZG4)+(XG1*ZG3*YG4)
     *      +(XG3*YG1*ZG4)-(XG3*ZG1*YG4)-(XG4*YG1*ZG3)+(XG4*ZG1*YG3)
     *      +(XG1*YG2*ZG4)-(XG1*ZG2*YG4)-(XG2*YG1*ZG4)+(XG2*ZG1*YG4)
     *      +(XG4*YG1*ZG2)-(XG4*ZG1*YG2)-(XG1*YG2*ZG3)+(XG1*ZG2*YG3)
     *      +(XG2*YG1*ZG3)-(XG2*ZG1*YG3)-(XG3*YG1*ZG2)+(XG3*ZG1*YG2))
     *      /6.
      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 =  (XG3*ZG4)-(ZG3*XG4)
     *      -(XG2*ZG4)+(ZG2*XG4)
     *      +(XG2*ZG3)-(ZG2*XG3)
      C2 = -(XG3*ZG4)+(ZG3*XG4)
     *      +(XG1*ZG4)-(ZG1*XG4)
     *      -(XG1*ZG3)+(ZG1*XG3)
      C3 =  (XG2*ZG4)-(ZG2*XG4)
     *      -(XG1*ZG4)+(ZG1*XG4)
     *      +(XG1*ZG2)-(ZG1*XG2)
      C4 = -(XG2*ZG3)+(ZG2*XG3)
     *      +(XG1*ZG3)-(ZG1*XG3)
     *      -(XG1*ZG2)+(ZG1*XG2)
      D1 = -(XG3*YG4)+(YG3*XG4)
     *      +(XG2*YG4)-(YG2*XG4)
     *      -(XG2*YG3)+(YG2*XG3)
      D2 =  (XG3*YG4)-(YG3*XG4)
     *      -(XG1*YG4)+(YG1*XG4)
     *      +(XG1*YG3)-(YG1*XG3)
      D3 = -(XG2*YG4)+(YG2*XG4)
     *      +(XG1*YG4)-(YG1*XG4)
     *      -(XG1*YG2)+(YG1*XG2)
      D4 =  (XG2*YG3)-(YG2*XG3)
     *      -(XG1*YG3)+(YG1*XG3)
     *      +(XG1*YG2)-(YG1*XG2)
C
      DO 10  I=1,3
      DO 10  J=1,4
      B(I,J) = 0.
   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.*VOL)
      BT(J,I) = B(I,J)
   30 CONTINUE
   20 CONTINUE
C
C     ELEMENT [K] MATRIX:
C
      DO 100  I=1,4
      DO 100  J=1,4
      AKC(I,J) = 0.
      DO 110  K=1,3
      AKC(I,J) = AKC(I,J) + BT(I,K)*B(K,J)
  110 CONTINUE
      AKC(I,J) = VOL*AKC(I,J)
```

```
  100 CONTINUE
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
      CALL ASSMBLE(IE, INTMAT, AKC, SYSK, SYSR, MXPOI, MXELE)
C
  500 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CVEL(NPOIN, NELEM, INTMAT, COORD,
     *                MXPOI, MXELE, POTEN, U, V, W, UE, VE, WE)
C
C     CALCULATE VELOCITY OF ALL ELEMENT FORM NODAL SOLUTIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,3), POTEN(MXPOI), ONE(MXPOI)
      DIMENSION  UE(MXELE), VE(MXELE), WE(MXELE)
      DIMENSION  U(MXPOI), V(MXPOI), W(MXPOI)
C
      INTEGER  INTMAT(MXELE,4)
C
      DO 100 I=1,NELEM
      UE(I) = 0.
      VE(I) = 0.
      WE(I) = 0.
  100 CONTINUE
      DO 150 I=1,NPOIN
      U(I)   = 0.
      V(I)   = 0.
      W(I)   = 0.
      ONE(I) = 0.
  150 CONTINUE
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  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 = ((XG2*YG3*ZG4)-(XG2*ZG3*YG4)-(XG3*YG2*ZG4)+(XG3*ZG2*YG4)
     *      +(XG4*YG2*ZG3)-(XG4*ZG2*YG3)-(XG1*YG3*ZG4)+(XG1*ZG3*YG4)
     *      +(XG3*YG1*ZG4)-(XG3*ZG1*YG4)-(XG4*YG1*ZG3)+(XG4*ZG1*YG3)
     *      +(XG1*YG2*ZG4)-(XG1*ZG2*YG4)-(XG2*YG1*ZG4)+(XG2*ZG1*YG4)
     *      +(XG4*YG1*ZG2)-(XG4*ZG1*YG2)-(XG1*YG2*ZG3)+(XG1*ZG2*YG3)
     *      +(XG2*YG1*ZG3)-(XG2*ZG1*YG3)-(XG3*YG1*ZG2)+(XG3*ZG1*YG2))
```

```
     *      /6.
      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
      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 =  (XG3*ZG4)-(ZG3*XG4)
     *     -(XG2*ZG4)+(ZG2*XG4)
     *     +(XG2*ZG3)-(ZG2*XG3)
      C2 = -(XG3*ZG4)+(ZG3*XG4)
     *     +(XG1*ZG4)-(ZG1*XG4)
     *     -(XG1*ZG3)+(ZG1*XG3)
      C3 =  (XG2*ZG4)-(ZG2*XG4)
     *     -(XG1*ZG4)+(ZG1*XG4)
     *     +(XG1*ZG2)-(ZG1*XG2)
      C4 = -(XG2*ZG3)+(ZG2*XG3)
     *     +(XG1*ZG3)-(ZG1*XG3)
     *     -(XG1*ZG2)+(ZG1*XG2)
      D1 = -(XG3*YG4)+(YG3*XG4)
     *     +(XG2*YG4)-(YG2*XG4)
     *     -(XG2*YG3)+(YG2*XG3)
      D2 =  (XG3*YG4)-(YG3*XG4)
     *     -(XG1*YG4)+(YG1*XG4)
     *     +(XG1*YG3)-(YG1*XG3)
      D3 = -(XG2*YG4)+(YG2*XG4)
     *     +(XG1*YG4)-(YG1*XG4)
     *     -(XG1*YG2)+(YG1*XG2)
      D4 =  (XG2*YG3)-(YG2*XG3)
     *     -(XG1*YG3)+(YG1*XG3)
     *     +(XG1*YG2)-(YG1*XG2)
C
      POTEN1 = POTEN(II)
      POTEN2 = POTEN(JJ)
      POTEN3 = POTEN(KK)
      POTEN4 = POTEN(LL)
C
C     COMPUTE THE ELEMENT VELOCITY:
C
      UE(IE) = ((B1*POTEN1)+(B2*POTEN2)+(B3*POTEN3)+(B4*POTEN4))
     *         /(-6.*VOL)
      VE(IE) = ((C1*POTEN1)+(C2*POTEN2)+(C3*POTEN3)+(C4*POTEN4))
     *         /(-6.*VOL)
      WE(IE) = ((D1*POTEN1)+(D2*POTEN2)+(D3*POTEN3)+(D4*POTEN4))
     *         /(-6.*VOL)
C
C     COMPUTE NODAL VELOCITIES FROM ELEMENT VELOCITY:
C
      U(II) = U(II) + UE(IE)
      U(JJ) = U(JJ) + UE(IE)
      U(KK) = U(KK) + UE(IE)
      U(LL) = U(LL) + UE(IE)
      V(II) = V(II) + VE(IE)
```

```
      V(JJ) = V(JJ) + VE(IE)
      V(KK) = V(KK) + VE(IE)
      V(LL) = V(LL) + VE(IE)
      W(II) = W(II) + WE(IE)
      W(JJ) = W(JJ) + WE(IE)
      W(KK) = W(KK) + WE(IE)
      W(LL) = W(LL) + WE(IE)
      ONE(II) = ONE(II) + 1.
      ONE(JJ) = ONE(JJ) + 1.
      ONE(KK) = ONE(KK) + 1.
      ONE(LL) = ONE(LL) + 1.
C
  500 CONTINUE
C
      DO 600  I=1,NPOIN
      IF(ONE(I).EQ.0.)  WRITE(6,650)  I
  650 FORMAT(' *** WARNING ***  NO VELOCITY CONTRIBUTION AT NODE', I5)
      IF(ONE(I).EQ.0.)  ONE(I) = 1.
      U(I) = U(I)/ONE(I)
      V(I) = V(I)/ONE(I)
      W(I) = W(I)/ONE(I)
  600 CONTINUE
C
C     ROUND-OFF SOLUTION VALUES FOR OUTPUT:
C
      ROFF = 1.E-6
      DO 700  IEQ=1,NPOIN
      VPOTEN = POTEN(IEQ)
      VALUEU = U(IEQ)
      VALUEV = V(IEQ)
      VALUEW = W(IEQ)
      IF(ABS(VPOTEN).LT.ROFF) POTEN(IEQ) = 0.
      IF(ABS(VALUEU).LT.ROFF)     U(IEQ) = 0.
      IF(ABS(VALUEV).LT.ROFF)     V(IEQ) = 0.
      IF(ABS(VALUEW).LT.ROFF)     W(IEQ) = 0.
  700 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE PRESS(NPOIN, COORD, IREFP, MXPOI, DEN, GRA,
     *                 U, V, W, VEL, P)
C
C     CALCULATE PRESSURE OF ALL NODAL SOLUTIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOI,3), P(MXPOI)
      DIMENSION  U(MXPOI), V(MXPOI),  W(MXPOI), VEL(MXPOI)
C
      DO 100 I=1,NPOIN
      P(I)    = 0.
  100 CONTINUE
C
      VREF = SQRT((U(IREFP)*U(IREFP))+(V(IREFP)*V(IREFP))
     *           +(W(IREFP)*W(IREFP)))
      YREF = (COORD(IREFP,2))
C
C     COMPUTE THE NODAL PRESSURE:
C
      CONST = ((VREF*VREF)/(2.*GRA))+YREF
      DO 200 I=1,NPOIN
      H = COORD(I,2)
      VEL(I) = SQRT((U(I)*U(I))+(V(I)*V(I))+(W(I)*W(I)))
      P(I) = DEN*GRA*(CONST-(((VEL(I)*VEL(I))/(2.*GRA))+H))
```

```
  200 CONTINUE
C
C     ROUND-OFF SOLUTION VALUES FOR OUTPUT:
C
      ROFF = 1.E-6
      DO 300  IEQ=1,NPOIN
      VALUEP    = P(IEQ)
      VALUEVEL = VEL(IEQ)
      IF(ABS(VALUEP).LT.ROFF)       P(IEQ) = 0.
      IF(ABS(VALUEVEL).LT.ROFF)  VEL(IEQ) = 0.
  300 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
```

ภาคผนวก ค.

รายละเอียดโปรแกรม NAVIER2D

รายละเอียดโปรแกรม NAVIER2D จะมีรายละเอียดเริ่มจากโปรแกรมหลัก และตามด้วยโปรแกรมย่อยต่าง ๆ ดังนี้

```
C     PROGRAM NAVIER2D
C
C     A FINITE ELEMENT COMPUTER PROGRAM FOR SOLVING NAVIER-STOKES
C     EQUATION FOR TWO-DIMENTIONAL VISCOUS INCOMPRESSIBLE FLOWS.
C
C     THE VALUES DECLARED IN THE PARAMETER STATEMENT BELOW SHOULD
C     BE ADJUSTED ACCORDING TO THE SIZE OF THE PROBLEMS AND TYPES
C     OF COMPUTERS:
C         MXPOIV = MAXIMUM NUMBER OF VELOCITY NODES IN THE MODEL
C         MXPOIP = MAXIMUM NUMBER OF PRESSURE NODES IN THE MODEL
C         MXELE  = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C
      PARAMETER (MXPOIV=3000, MXPOIP=1000, MXELE=2000, MXFREE=1)
      PARAMETER (MXNEQ=2*MXPOIV+MXPOIP)
C
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOIV,2), TEXT(20)
      DIMENSION  UVEL(MXPOIV), VVEL(MXPOIV), PRES(MXPOIV)
      DIMENSION  SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ)
      DIMENSION  SOL(MXNEQ), DSOL(MXNEQ)
      CHARACTER*20  NAME1, NAME2, NAME3, NAME4
C
      INTEGER  INTMAT(MXELE,6), INTMATF(MXFREE,4)
      INTEGER  IBCU(MXPOIV), IBCV(MXPOIV), IBCP(MXPOIV)
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)
      OPEN(UNIT=9, FILE='CHECK.OUT', STATUS='NEW')
C
C     READ TITLE OF COMPUTATION:
C
      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
      WRITE(9,104)
  104 FORMAT('   NPOIV   NPOIP   NELEM   NFREE   NITER    TOL')
      READ(7,*)    NPOIV, NPOIP, NELEM, NFREE, NITER, TOL
      WRITE(9,105) NPOIV, NPOIP, NELEM, NFREE, NITER, TOL
  105 FORMAT(5I8, F8.2)
      IF(NPOIV.GT.MXPOIV)  WRITE(6,110) NPOIV
  110 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOIV TO',I5)
      IF(NPOIV.GT.MXPOIV)  STOP
      IF(NPOIP.GT.MXPOIP)  WRITE(6,120) NPOIP
```

```
  120 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOIP TO',I5)
      IF(NPOIP.GT.MXPOIP)  STOP
      IF(NELEM.GT.MXELE)  WRITE(6,130) NELEM
  130 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO',I5)
      IF(NELEM.GT.MXELE)  STOP
      IF(NFREE.GT.MXFREE)  WRITE(6,140) NFREE
  140 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXFREE TO',I5)
      IF(NFREE.GT.MXFREE)  STOP
C
C     READ FLUID PROPERTIES:
C
      READ(7,1)  TEXT
      WRITE(9,134)
  134 FORMAT('     DENSITY   VISCOSITY   GRAVITY')
      READ(7,*)  DEN, VIS, GRA
      WRITE(9,135)  DEN, VIS, GRA
  135 FORMAT(3E12.4)
C
C     READ NODAL COORDINATES, BOUNDARY CONDITIONS, THEIR VALUES:
C     REQUIREMENT:  MAIN NODES MUST BE NUMBERED FIRST
C
      READ(7,1)  TEXT
      WRITE(9,138)  NPOIV
  138 FORMAT(' NODAL INFORMATION (NODE NO., U-V-P BC, X-Y COORD,',
     *       ' U-V-P VALUES): [', I4, ']'                          )
      DO 150  IP=1,NPOIV
      READ(7,*)    I, IBCU(I), IBCV(I), IBCP(I),
     *             (COORD(I,K), K=1,2), UVEL(I), VVEL(I), PRES(I)
      WRITE(9,152) I, IBCU(I), IBCV(I), IBCP(I),
     *             (COORD(I,K), K=1,2), UVEL(I), VVEL(I), PRES(I)
  152 FORMAT(I6, 3I4, 5E12.4)
      IF(I.NE.IP)  WRITE(6,155) IP
  155 FORMAT(/, ' NODE NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IP) STOP
  150 CONTINUE
C
C     READ ELEMENT NODAL CONNECTIONS:
C
      READ(7,1)  TEXT
      WRITE(9,157) NELEM
  157 FORMAT(' ELEMENT NODAL CONNECTIONS: [', I4, ']')
      DO 160  IE=1,NELEM
      READ(7,*)    I, (INTMAT(I,J), J=1,6)
      WRITE(9,162) I, (INTMAT(I,J), J=1,6)
  162 FORMAT(7I8)
      IF(I.NE.IE)  WRITE(6,165) IE
  165 FORMAT(/, ' ELEMENT NO.', I5, ' IN DATA FILE IS MISSING')
      IF(I.NE.IE) STOP
  160 CONTINUE
C
C     READ FREE BOUNDARY (FLOW EXIT) INFORMATION:
C
      READ(7,1)  TEXT
      WRITE(9,168)  NFREE
  168 FORMAT(' OUTFLOW INFORMATION (ELE NO., 3 NODE NO.): [',
     *       I4, ']')
      DO 170  IB=1,NFREE
      READ(7,*)    (INTMATF(IB,J), J=1,4)
      WRITE(9,172) (INTMATF(IB,J), J=1,4)
  172 FORMAT(4I8)
  170 CONTINUE
C
      WRITE(6,200)  NPOIV, NPOIP, NELEM, NFREE, NITER, TOL
  200 FORMAT('    THE FINITE ELEMENT MODEL CONSISTS OF:', /,
     *       '         NUMBER OF VELOCITY NODES       =', I6, /,
     *       '         NUMBER OF PRESSURE NODES       =', I6, /,
```

```
     *       '          NUMBER OF ELEMENTS              =', I6, /,
     *       '          NUMBER OF OUTFLOW BOUNDARY      =', I6, /,
     *       '     WITH NUMBER OF ITERATIONS REQUIRED   =', I6, /,
     *       '     OR SPECIFIED STOPPING TOLERANCE      =', F6.2 )
C
      DO 400  I=1,NPOIV
      SOL(I       ) = UVEL(I)
      SOL(I+NPOIV) = VVEL(I)
  400 CONTINUE
      DO 410  I=1,NPOIP
      SOL(I+NPOIV+NPOIV) = PRES(I)
  410 CONTINUE
C
      NEQ = 2*NPOIV + NPOIP
C
C     ENTER ITERATION LOOP:
C
      DO 500  ITER=1,NITER
C
C     RESET THE SYSTEM EQUATIONS
C
      DO 510  I=1,NEQ
      SYSR(I) = 0.
  510 CONTINUE
      DO 520  I=1,NEQ
      DO 520  J=1,NEQ
      SYSK(I,J) = 0.
  520 CONTINUE
C
      WRITE(6,530)  ITER
  530 FORMAT(/, 3X, ' * PERFORMING COMPUTATION AT ITERATION NUMBER',
     *       I16, ':')
C
C     ESTABLISH ELEMENT MATRICES AND ASSEMBLE ELEMENT EQUATIONS
C
      WRITE(6,540)
  540 FORMAT(8X, ' ESTABLISHING ELEMENT MATRICES AND',
     *           ' ASSEMBLING ELEMENT EQS.'     )
C
      CALL TRI(NPOIV,  NPOIP,  NELEM,   NFREE,   NEQ,  DEN,
     *          VIS, GRA, COORD, INTMAT, INTMATF,  SYSK, SYSR,
     *          SOL, MXPOIV,  MXELE,  MXFREE, MXNEQ      )
C
C     APPLY BOUNDARY CONDITIONS OF NODAL INCREMENTS
C
      WRITE(6,550)
  550 FORMAT(8X, ' APPLYING BOUNDARY CONDITIONS OF NODAL',
     *           ' INCREMENTS'                          )
C
      CALL APPLYBC(NPOIV, NPOIP,    NEQ,   IBCU,  IBCV, IBCP,
     *            SYSK,  SYSR, MXPOIV, MXPOIP, MXNEQ    )
C
C     SOLVE A SET OF SIMULTANEOUS EQUATIONS FOR NODAL INCREMENTS:
C
      WRITE(6,560)
  560 FORMAT(8X, ' SOLVING SET OF SIMULTANEOUS EQS. FOR',
     *           ' NODAL INCREMENTS'                    )
      WRITE(6,570)  NEQ
  570 FORMAT(8X, ' ( TOTAL OF', I5,' EQUATIONS TO BE SOLVED )')
      CALL GAUSS(NEQ, SYSK, SYSR, DSOL, MXNEQ)
C
C     CHECK FOR CONVERGENCE:
C
      UP   = 0.
      DOWN = 0.
      DO 580  I=1,NEQ
```

```
      ERROR = DSOL(I)
      UP    = UP + ABS(ERROR)
      VALUE = SOL(I)
      DOWN  = DOWN + ABS(VALUE)
  580 CONTINUE
      RATIO = UP*100./DOWN
      WRITE(6,585)  RATIO
  585 FORMAT(6X, 'CURRENT SOLUTION HAS GLOBAL ERROR OF',
     *       F8.2, ' %'                                    )
      WRITE(9,587)  ITER, RATIO
  587 FORMAT(6X, 'ITERATION NO.', I16, ' HAS GLOBAL ERROR OF',
     *       F8.2, ' %'                                     )
      IF(RATIO.GT.TOL)  GO TO 600
C
C     SOLUTION CONVERGED WITHIN THE SPECIFIED TOLERANCE
C
      WRITE(6,590)
  590 FORMAT(/, 3X, ' *** SOLUTION CONVERGED WITHIN SPECIFIED',
     *       ' TOLERANCE ***', //                           )
      GO TO 700
  600 CONTINUE
C
C     UPDATE NODAL SOLUTIONS:
C
      DO 610  I=1,NEQ
      SOL(I) = SOL(I) + DSOL(I)
  610 CONTINUE
  500 CONTINUE
C
C     SOLUTION NOT CONVERGED WITHIN THE SPECIFIED TOLERANCE
C
      WRITE(6,620)
  620 FORMAT(/, 3X, ' ??? SOLUTION NOT CONVERGED WITHIN',
     *       ' SPECIFIED TOLERANCE ???', //                )
C
  700 CONTINUE
C
C     PRINT OUT SOLUTIONS OF NODAL VELOCITIES AND PRESSURES:
C
  710 WRITE(6,720)
  720 FORMAT(' PLEASE ENTER FILE NAME FOR VELOCITY & PRESSURE',
     *       ' SOLUTIONS:', /                              )
      READ(5, '(A)', ERR=710)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=710)
      WRITE(8,730)  NPOIV
  730 FORMAT('  NODAL VELOCITY AND PRESSURE SOLUTIONS [', I5,']:',
     *       //, 2X, 'NODE', 6X, 'U-VELOCITY', 6X, 'V-VELOCITY',
     *       8X, 'PRESSURE', /                                  )
C
C     ROUND-OFF SOLUTION VALUES FOR NEAT OUTPUT:
C
      ROFF = 1.E-6
      DO 740  IEQ=1,NEQ
      VALUE = SOL(IEQ)
      IF(ABS(VALUE).LT.ROFF)  SOL(IEQ) = 0.
  740 CONTINUE
C
      DO 750  IP=1,NPOIP
      IEQU = IP
      IEQV = NPOIV + IP
      IEQP = 2*NPOIV + IP
      WRITE(8,760)  IP, SOL(IEQU), SOL(IEQV), SOL(IEQP)
  760 FORMAT(I6, 3E16.6)
  750 CONTINUE
      DO 770  IP=NPOIP+1,NPOIV
      IEQU = IP
```

```
      IEQV = NPOIV + IP
      WRITE(8,780)  IP, SOL(IEQU), SOL(IEQV)
  780 FORMAT(I6, 2E16.6)
  770 CONTINUE
C
C     CREATE DATA FILE FOR GRAPHIC DISPLAY (FEPLOT):
C
  800 WRITE(6,810)
  810 FORMAT(' PLEASE ENTER FILE NAME FOR U-V-P DISPLAY:', /)
      READ(5,'(A)', ERR=800)  NAME3
      OPEN(UNIT=10, FILE=NAME3, STATUS='NEW', ERR=800)
      NVAR = 3
      WRITE(10,820)  NPOIP, NELEM, NVAR
  820 FORMAT('   NPOIP   NELEM    NVAR', /, 3I8)
      WRITE(10,830)  NPOIP
  830 FORMAT(' NODAL COORDINATES & U-V-P SOLUTIONS [', I5, ']:')
      DO 840  I=1,NPOIP
      IEQU = I
      IEQV = NPOIV + I
      IEQP = 2*NPOIV + I
      WRITE(10,850)  I, (COORD(I,J), J=1,2), SOL(IEQU), SOL(IEQV),
     *               SOL(IEQP)
  850 FORMAT(I8, 5E13.5)
  840 CONTINUE
      WRITE(10,860)  NELEM
  860 FORMAT(' ELEMENT NODAL CONNECTIONS [', I5, ']:')
      DO 870  IE=1,NELEM
      WRITE(10,880)  IE, (INTMAT(IE,J), J=1,3)
  880 FORMAT(4I8)
  870 CONTINUE
C
  900 WRITE(6,910)
  910 FORMAT(' PLEASE ENTER FILE NAME FOR U-V DISPLAY:', /)
      READ(5,'(A)', ERR=900)  NAME4
      OPEN(UNIT=11, FILE=NAME4, STATUS='NEW', ERR=900)
      NVAR = 2
      NELEM4 = 4*NELEM
      WRITE(11,920)  NPOIV, NELEM4, NVAR
  920 FORMAT('   NPOIV   NELEM    NVAR', /, 3I8)
      WRITE(11,930)  NPOIV
  930 FORMAT(' NODAL COORDINATES & U-V SOLUTIONS [', I5, ']:')
      DO 940  I=1,NPOIV
      IEQU = I
      IEQV = NPOIV + I
      WRITE(11,950)  I, (COORD(I,J), J=1,2), SOL(IEQU), SOL(IEQV)
  950 FORMAT(I8, 4E13.5)
  940 CONTINUE
      WRITE(11,960)  NELEM4
  960 FORMAT(' ELEMENT NODAL CONNECTIONS [', I5, ']:')
      ICE = 1
      DO 970  IE=1,NELEM
      II = INTMAT(IE,1)
      JJ = INTMAT(IE,2)
      KK = INTMAT(IE,3)
      LL = INTMAT(IE,4)
      MM = INTMAT(IE,5)
      NN = INTMAT(IE,6)
      WRITE(11,980)  ICE, II, NN, MM
      ICE = ICE + 1
      WRITE(11,980)  ICE, JJ, LL, NN
      ICE = ICE + 1
      WRITE(11,980)  ICE, KK, MM, LL
      ICE = ICE + 1
      WRITE(11,980)  ICE, LL, MM, NN
      ICE = ICE + 1
  980 FORMAT(4I8)
```

```
  970 CONTINUE
C
      STOP
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NPOIV, NPOIP,    NEQ,   IBCU,  IBCV, IBCP,
     *                   SYSK,  SYSR, MXPOIV, MXPOIP, MXNEQ      )
C
C     APPLY BOUNDARY CONDITIONS BEFORE SOLVING FOR NODAL INCREMENTS
C     WITH CONDITION CODES OF:
C           0 = FREE TO CHANGE (INCREMENTS COMPUTED)
C           1 = FIXED AS SPECIFIED (INCREMENTS FIXED AS ZERO)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ)
C
      INTEGER  IBCU(MXPOIV), IBCV(MXPOIV), IBCP(MXPOIV)
C
C     APPLY BOUNDARY CONDITIONS FOR NODAL U-VELOCITIES:
C
      IEQ1 = 1
      IEQ2 = NPOIV
      DO 100  IEQ=IEQ1,IEQ2
      IEQU = IEQ
      IF(IBCU(IEQU).EQ.0)  GO TO 100
C
      DO 110  IR=1,NEQ
      IF(IR.EQ.IEQ)  GO TO 110
      SYSK(IR,IEQ) = 0.
  110 CONTINUE
C
      DO 120  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  120 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = 0.
C
  100 CONTINUE
C
C     APPLY BOUNDARY CONDITIONS FOR NODAL V-VELOCITIES:
C
      IEQ1 = NPOIV + 1
      IEQ2 = 2*NPOIV
      DO 200  IEQ=IEQ1,IEQ2
      IEQV = IEQ - NPOIV
      IF(IBCV(IEQV).EQ.0)  GO TO 200
C
      DO 210  IR=1,NEQ
      IF(IR.EQ.IEQ)  GO TO 210
      SYSK(IR,IEQ) = 0.
  210 CONTINUE
C
      DO 220  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  220 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = 0.
C
  200 CONTINUE
C
C     APPLY BOUNDARY CONDITIONS FOR NODAL PRESSURES:
C
      IEQ1 = 2*NPOIV + 1
      IEQ2 = NEQ
```

```
      DO 300  IEQ=IEQ1,IEQ2
      IEQP = IEQ - 2*NPOIV
      IF(IBCP(IEQP).EQ.0)  GO TO 300
C
      DO 310  IR=1,NEQ
      IF(IR.EQ.IEQ)  GO TO 310
      SYSK(IR,IEQ) = 0.
  310 CONTINUE
C
      DO 320  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  320 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = 0.
C
  300 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ASSMBLE(   IE, INTMAT, AKELE,  RELE,  SYSK, SYSR,
     *                   NPOIV,    NEQ, NELEM, MXNEQ, MXELE      )
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  AKELE(15,15), RELE(15)
      DIMENSION  SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ)
C
      INTEGER  INTMAT(MXELE,6)
C
C     ASSEMBLING SYSTEM STIFFNESS MATRIX
C
C     CONTRIBUTION OF COEFFICIENTS ASSOCIATED WITH U & V VELOCITIES:
C
      DO 100  I=1,6
      DO 100  J=1,6
      II = INTMAT(IE,I)
      JJ = INTMAT(IE,J)
      K  = I + 6
      L  = J + 6
      KK = NPOIV + II
      LL = NPOIV + JJ
      SYSK(II,JJ) = SYSK(II,JJ) + AKELE(I,J)
      SYSK(II,LL) = SYSK(II,LL) + AKELE(I,L)
      SYSK(KK,JJ) = SYSK(KK,JJ) + AKELE(K,J)
      SYSK(KK,LL) = SYSK(KK,LL) + AKELE(K,L)
  100 CONTINUE
C
C     CONTRIBUTION OF COEFFICIENTS ASSOCIATED WITH PRESSURE:
C
      DO 200  I=1,6
      DO 200  J=1,3
      II = INTMAT(IE,I)
      JJ = INTMAT(IE,J)
      K  = I + 6
      L  = J + 12
      KK = NPOIV + II
      LL = 2*NPOIV + JJ
      SYSK(II,LL) = SYSK(II,LL) + AKELE(I,L)
      SYSK(KK,LL) = SYSK(KK,LL) + AKELE(K,L)
      SYSK(LL,II) = SYSK(LL,II) + AKELE(L,I)
      SYSK(LL,KK) = SYSK(LL,KK) + AKELE(L,K)
  200 CONTINUE
```

```
C
C     ASSEMBLING SYSTEM LOAD VECTOR
C
C     CONTRIBUTION OF VALUES ASSOCIATED WITH U & V VELOCITIES:
C
      DO 300  I=1,6
      II = INTMAT(IE,I)
      K  = I + 6
      KK = NPOIV + II
      SYSR(II) = SYSR(II) + RELE(I)
      SYSR(KK) = SYSR(KK) + RELE(K)
  300 CONTINUE
C
C     CONTRIBUTION OF VALUES ASSOCIATED WITH PRESSURE:
C
      DO 400  I=1,3
      II = INTMAT(IE,I)
      K  = I + 12
      KK = 2*NPOIV + II
      SYSR(KK) = SYSR(KK) + RELE(K)
  400 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE GAUSS(N, A, B, X, MXNEQ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXNEQ,MXNEQ), B(MXNEQ), X(MXNEQ)
C
C     PERFORM SCALING:
C
      CALL SCALE(N, A, B, MXNEQ)
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, MXNEQ, 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, MXNEQ, IP)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXNEQ,MXNEQ), B(MXNEQ)
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, MXNEQ)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(MXNEQ,MXNEQ), B(MXNEQ)
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-----------------------------------------------------------------------
C
      SUBROUTINE TRI(NPOIV,  NPOIP,  NELEM,   NFREE,   NEQ,  DEN,
     *               VIS, GRA, COORD, INTMAT, INTMATF,  SYSK, SYSR,
     *               SOL, MXPOIV,  MXELE,  MXFREE, MXNEQ      )
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     UP SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOIV,2), SYSK(MXNEQ,MXNEQ)
      DIMENSION  SYSR(MXNEQ), SOL(MXNEQ)
      DIMENSION  A(6,6), B(6,3), C(6,3), G(3,3), F(6,6,3)
      DIMENSION  UELE(6), VELE(6), PELE(3)
      DIMENSION  SXX(6,6), SXY(6,6), SYX(6,6), SYY(6,6)
      DIMENSION  HX(3,6), HY(3,6), HXT(6,3), HYT(6,3)
      DIMENSION  ABGXUG(6,6), AGBXUG(6,6), AGBYVG(6,6)
      DIMENSION  ABGYVG(6,6), ABGXVG(6,6), ABGYUG(6,6)
      DIMENSION  GXX(6,6), GYY(6,6), ALX(6,6), ALY(6,6)
      DIMENSION  AKELE(15,15), RELE(15), FX(6), FY(6), FI(3)
C
      INTEGER  INTMAT(MXELE,6), INTMATF(MXFREE,4)
C
C     SET UP [A] MATRIX BASED ON TENSOR NOTATIONS:
C
      DO 10  I=1,6
      DO 10  J=1,6
      A(I,J) = 0.
   10 CONTINUE
      A(1,1) =  1.
      A(2,2) =  1.
      A(3,3) =  1.
      A(4,4) =  4.
      A(5,5) =  4.
      A(6,6) =  4.
      A(1,5) = -1.
      A(1,6) = -1.
      A(2,4) = -1.
      A(2,6) = -1.
      A(3,4) = -1.
      A(3,5) = -1.
C
C     COMPUTE KINEMATIC VISCOSITY:
C
      ANEW = VIS/DEN
C
C     LOOP OVER THE NUMBER OF ELEMENTS:
C
      DO 500  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)
      MM = INTMAT(IE,5)
      NN = INTMAT(IE,6)
C
      XG1 = COORD(II,1)
      XG2 = COORD(JJ,1)
      XG3 = COORD(KK,1)
```

```
      YG1 = COORD(II,2)
      YG2 = COORD(JJ,2)
      YG3 = COORD(KK,2)
      AREA= 0.5*(XG2*(YG3-YG1) + XG1*(YG2-YG3) + XG3*(YG1-YG2))
      IF(AREA.LE.0.)  WRITE(6,5)  IE
    5 FORMAT(/,'  !!! ERROR !!!  ELEMENT NO.', I5,
     *         ' HAS NEGATIVE OR ZERO AREA ', /,
     *         '  --- CHECK F.E. MODEL FOR NODAL COORDINATES',
     *         ' AND ELEMENT NODAL CONNECTIONS ---'          )
      IF(AREA.LE.0.)  STOP
C
      AREA2 = 2.*AREA
      B1 = (YG2 - YG3)/AREA2
      B2 = (YG3 - YG1)/AREA2
      B3 = (YG1 - YG2)/AREA2
      C1 = (XG3 - XG2)/AREA2
      C2 = (XG1 - XG3)/AREA2
      C3 = (XG2 - XG1)/AREA2
C
C     SET UP [B] AND [C] MATRICES BASED ON TENSOR NOTATIONS:
C
      DO 30  I=1,6
      DO 30  J=1,3
      B(I,J) = 0.
      C(I,J) = 0.
   30 CONTINUE
      B(1,1) = 2.*B1
      B(2,2) = 2.*B2
      B(3,3) = 2.*B3
      B(4,2) = B3
      B(4,3) = B2
      B(5,1) = B3
      B(5,3) = B1
      B(6,1) = B2
      B(6,2) = B1
      C(1,1) = 2.*C1
      C(2,2) = 2.*C2
      C(3,3) = 2.*C3
      C(4,2) = C3
      C(4,3) = C2
      C(5,1) = C3
      C(5,3) = C1
      C(6,1) = C2
      C(6,2) = C1
C
C     SET UP [G] MATRIX:
C
      FAC  = AREA/12.
      FAC2 = 2.*FAC
      G(1,1) = FAC2
      G(2,2) = FAC2
      G(3,3) = FAC2
      G(1,2) = FAC
      G(1,3) = FAC
      G(2,1) = FAC
      G(2,3) = FAC
      G(3,1) = FAC
      G(3,2) = FAC
C
C     SET UP [F] MATRIX BASED ON TENSOR NOTATIONS:
C
      FACTOR = 2.*AREA/5040.
      F4   = FACTOR*4.
      F6   = FACTOR*6.
      F12  = FACTOR*12.
      F24  = FACTOR*24.
```

```
      F120 = FACTOR*120.
C
      F(1,1,1) = F120
      F(1,2,1) = F12
      F(1,3,1) = F12
      F(1,4,1) = F6
      F(1,5,1) = F24
      F(1,6,1) = F24
      F(2,2,1) = F24
      F(2,3,1) = F4
      F(2,4,1) = F6
      F(2,5,1) = F4
      F(2,6,1) = F12
      F(3,3,1) = F24
      F(3,4,1) = F6
      F(3,5,1) = F12
      F(3,6,1) = F4
      F(4,4,1) = F4
      F(4,5,1) = F4
      F(4,6,1) = F4
      F(5,5,1) = F12
      F(5,6,1) = F6
      F(6,6,1) = F12
      DO 40  I=1,6
      DO 40  J=I,6
      F(J,I,1) = F(I,J,1)
   40 CONTINUE
C
      F(1,1,2) = F24
      F(1,2,2) = F12
      F(1,3,2) = F4
      F(1,4,2) = F4
      F(1,5,2) = F6
      F(1,6,2) = F12
      F(2,2,2) = F120
      F(2,3,2) = F12
      F(2,4,2) = F24
      F(2,5,2) = F6
      F(2,6,2) = F24
      F(3,3,2) = F24
      F(3,4,2) = F12
      F(3,5,2) = F6
      F(3,6,2) = F4
      F(4,4,2) = F12
      F(4,5,2) = F4
      F(4,6,2) = F6
      F(5,5,2) = F4
      F(5,6,2) = F4
      F(6,6,2) = F12
      DO 50  I=1,6
      DO 50  J=I,6
      F(J,I,2) = F(I,J,2)
   50 CONTINUE
C
      F(1,1,3) = F24
      F(1,2,3) = F4
      F(1,3,3) = F12
      F(1,4,3) = F4
      F(1,5,3) = F12
      F(1,6,3) = F6
      F(2,2,3) = F24
      F(2,3,3) = F12
      F(2,4,3) = F12
      F(2,5,3) = F4
      F(2,6,3) = F6
      F(3,3,3) = F120
```

```
      F(3,4,3) = F24
      F(3,5,3) = F24
      F(3,6,3) = F6
      F(4,4,3) = F12
      F(4,5,3) = F6
      F(4,6,3) = F4
      F(5,5,3) = F12
      F(5,6,3) = F4
      F(6,6,3) = F4
      DO 60  I=1,6
      DO 60  J=I,6
      F(J,I,3) = F(I,J,3)
   60 CONTINUE
C
C     EXTRACT ELEMENT NODAL  U, V, P:
C
      UELE(1) = SOL(II)
      UELE(2) = SOL(JJ)
      UELE(3) = SOL(KK)
      UELE(4) = SOL(LL)
      UELE(5) = SOL(MM)
      UELE(6) = SOL(NN)
      VELE(1) = SOL(II+NPOIV)
      VELE(2) = SOL(JJ+NPOIV)
      VELE(3) = SOL(KK+NPOIV)
      VELE(4) = SOL(LL+NPOIV)
      VELE(5) = SOL(MM+NPOIV)
      VELE(6) = SOL(NN+NPOIV)
      PELE(1) = SOL(II+NPOIV+NPOIV)
      PELE(2) = SOL(JJ+NPOIV+NPOIV)
      PELE(3) = SOL(KK+NPOIV+NPOIV)
C
C     COMPUTE  [SXX], [SXY], [SYX], [SYY]  MATRICES:
C
      DO 100  IA=1,6
      DO 100  IB=1,6
      CXX = 0.
      CYY = 0.
      CXY = 0.
      CYX = 0.
      DO 110  I=1,6
      DO 110  J=1,3
      DO 110  K=1,3
      DO 110  L=1,6
      CXX = CXX + A(IA,I)*B(I,J)*A(IB,L)*B(L,K)*G(J,K)
      CYY = CYY + A(IA,I)*C(I,J)*A(IB,L)*C(L,K)*G(J,K)
      CXY = CXY + A(IA,I)*C(I,J)*A(IB,L)*B(L,K)*G(J,K)
      CYX = CYX + A(IA,I)*B(I,J)*A(IB,L)*C(L,K)*G(J,K)
  110 CONTINUE
      SXX(IA,IB) = 2.*ANEW*CXX + ANEW*CYY
      SXY(IA,IB) = ANEW*CXY
      SYX(IA,IB) = ANEW*CYX
      SYY(IA,IB) = ANEW*CXX + 2.*ANEW*CYY
  100 CONTINUE
C
C     COMPUTE  [HX] AND [HY]  MATRICES:
C     (SAME AS MATRICES ON THE LOWER LEFT OF LINEAR EQS.)
C
      DO 150  IA=1,3
      DO 150  IB=1,6
      CX = 0.
      CY = 0.
      DO 160  I=1,6
      DO 160  J=1,3
      CX = CX + A(IB,I)*B(I,J)*G(J,IA)
      CY = CY + A(IB,I)*C(I,J)*G(J,IA)
```

```
  160 CONTINUE
      HX(IA,IB) = CX/DEN
      HY(IA,IB) = CY/DEN
  150 CONTINUE
C
C     THEN THE CORRESPONDING TWO MATRICES ON THE UPPER RIGHT ARE:
C
      DO 170  IA=1,3
      DO 170  IB=1,6
      HXT(IB,IA) = -HX(IA,IB)
      HYT(IB,IA) = -HY(IA,IB)
  170 CONTINUE
C
C     COMPUTE ALL MATRICES ASSOCIATED WITH THE INERTIA TERMS:
C     (SEE DERIVATION IN NOTE FOR BETTER UNDERSTANDING)
C
      DO 200  IA=1,6
      DO 200  IB=1,6
      CABGXUG = 0.
      CAGBXUG = 0.
      CAGBYVG = 0.
      CABGYVG = 0.
      CABGXVG = 0.
      CABGYUG = 0.
      DO 210  I=1,6
      DO 210  J=1,6
      DO 210  K=1,6
      DO 210  L=1,6
      DO 210  M=1,3
      CABGXUG = CABGXUG
     1        + A(IA,I)*A(IB,J)*A(K,L)*B(L,M)*F(I,J,M)*UELE(K)
      CAGBXUG = CAGBXUG
     1        + A(IA,I)*A(K,J)*A(IB,L)*B(L,M)*F(I,J,M)*UELE(K)
      CAGBYVG = CAGBYVG
     1        + A(IA,I)*A(K,J)*A(IB,L)*C(L,M)*F(I,J,M)*VELE(K)
      CABGYVG = CABGYVG
     1        + A(IA,I)*A(IB,J)*A(K,L)*C(L,M)*F(I,J,M)*VELE(K)
      CABGXVG = CABGXVG
     1        + A(IA,I)*A(IB,J)*A(K,L)*B(L,M)*F(I,J,M)*VELE(K)
      CABGYUG = CABGYUG
     1        + A(IA,I)*A(IB,J)*A(K,L)*C(L,M)*F(I,J,M)*UELE(K)
  210 CONTINUE
      ABGXUG(IA,IB) = CABGXUG
      AGBXUG(IA,IB) = CAGBXUG
      AGBYVG(IA,IB) = CAGBYVG
      ABGYVG(IA,IB) = CABGYVG
      ABGXVG(IA,IB) = CABGXVG
      ABGYUG(IA,IB) = CABGYUG
  200 CONTINUE
C
      DO 220  I=1,6
      DO 220  J=1,6
      GXX(I,J) = ABGXUG(I,J) + AGBXUG(I,J) + AGBYVG(I,J) + SXX(I,J)
      GYY(I,J) = ABGYVG(I,J) + AGBYVG(I,J) + AGBXUG(I,J) + SYY(I,J)
      ALX(I,J) = ABGXVG(I,J) + SXY(I,J)
      ALY(I,J) = ABGYUG(I,J) + SYX(I,J)
  220 CONTINUE
C
C     THEN THE MATRIX (15X15) ON LHS OF THE ELEMENT EQS. IS:
C
      DO 230  I=1,15
      DO 230  J=1,15
      AKELE(I,J) = 0.
  230 CONTINUE
C
      DO 240  I=1,6
```

```
      DO 250  J=1,6
      AKELE(I  ,J  ) = GXX(I,J)
      AKELE(I+6,J+6) = GYY(I,J)
      AKELE(I  ,J+6) = ALY(I,J)
      AKELE(I+6,J  ) = ALX(I,J)
  250 CONTINUE
      DO 260  J=1,3
      AKELE(I  ,J+12) = HXT(I,J)
      AKELE(I+6,J+12) = HYT(I,J)
  260 CONTINUE
  240 CONTINUE
      DO 270  I=1,3
      DO 270  J=1,6
      AKELE(I+12,J  ) = HX(I,J)
      AKELE(I+12,J+6) = HY(I,J)
  270 CONTINUE
C
C     BEGIN COMPUTING THE RESIDUALS ON RHS OF ELEMENT EQS.:
C
      DO 300  I=1,6
      TERM1 = 0.
      TERM2 = 0.
      TERM3 = 0.
      TERM4 = 0.
      TERM5 = 0.
      DO 310  J=1,6
      TERM1 = TERM1 + ABGXUG(I,J)*UELE(J)
      TERM2 = TERM2 + ABGYUG(I,J)*VELE(J)
      TERM4 = TERM4 + SXX(I,J)*UELE(J)
      TERM5 = TERM5 + SXY(I,J)*VELE(J)
  310 CONTINUE
      DO 320  J=1,3
      TERM3 = TERM3 + HXT(I,J)*PELE(J)
  320 CONTINUE
      FX(I) = TERM1 + TERM2 + TERM3 + TERM4 + TERM5
  300 CONTINUE
C
      DO 350  I=1,6
      TERM1 = 0.
      TERM2 = 0.
      TERM3 = 0.
      TERM4 = 0.
      TERM5 = 0.
      DO 360  J=1,6
      TERM1 = TERM1 + ABGXVG(I,J)*UELE(J)
      TERM2 = TERM2 + ABGYVG(I,J)*VELE(J)
      TERM4 = TERM4 + SYX(I,J)*UELE(J)
      TERM5 = TERM5 + SYY(I,J)*VELE(J)
  360 CONTINUE
      DO 370  J=1,3
      TERM3 = TERM3 + HYT(I,J)*PELE(J)
  370 CONTINUE
      FY(I) = TERM1 + TERM2 + TERM3 + TERM4 + TERM5
  350 CONTINUE
C
      DO 380  I=4,6
      FY(I) = FY(I) + GRA*AREA/3.
  380 CONTINUE
C
      DO 400  I=1,3
      TERM1 = 0.
      TERM2 = 0.
      DO 410  J=1,6
      TERM1 = TERM1 + HX(I,J)*UELE(J)
      TERM2 = TERM2 + HY(I,J)*VELE(J)
  410 CONTINUE
```

```
      FI(I) = TERM1 + TERM2
  400 CONTINUE
C
C     THUS THE RESIDUAL VECTOR ON RHS OF ELEMENT EQS. IS:
C
      DO 420  I=1,6
      RELE(I  ) = -FX(I)
      RELE(I+6) = -FY(I)
  420 CONTINUE
      DO 430  I=1,3
      RELE(I+12) = -FI(I)
  430 CONTINUE
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
      CALL ASSMBLE(   IE, INTMAT, AKELE,  RELE,  SYSK, SYSR,
     *             NPOIV,    NEQ, NELEM, MXNEQ, MXELE      )
C
  500 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
```

ประวัติผู้เขียนวิทยานิพนธ์

นายเสฏฐวรรธ สุจริตภวัตสกุล เกิดเมื่อวันที่ 16 เดือนกันยายน พุทธศักราช 2518 ที่จังหวัดอุดรธานี สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จากมหาวิทยาลัยขอนแก่น เมื่อปีการศึกษา 2539 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542