รายการอ้างอิง

ภาษาไทย

ปราโมทย์ เดชะอำไพ, ไฟไนต์เอลิเมนต์ในงานวิศวกรรม, กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2537.

ปราโมทย์ เดชะอำไพ, ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม, กรุงเทพมหานคร : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ภาษาอังกฤษ

B-N. Jiang, "A Least-Squares Finite Element Method for Incompressible Navier-Stokes Problems," *Int. J. Numer. Methods Fluids*, Vol. 14, 1992, pp. 843-895.

Cook, R. D., Malkus, D. S. and Plesha, M. E., Concepts and Applications of Finite Element Analysis, 3d ed., Wiley, New York, 1989.

J. G. Rice and R. J. Schnipke, "A Monotone Streamline Upwind Finite Element Method for Convection-Dominated Flows," *Comput. methods in Appl. Mech. Eng.*, Vol. 58, 1985, pp. 313-327.

Jin, J., The Finite Element Method in Electromagnetics, John Wiley & Sons, New York, 1993.

J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and O. C. Zienkiewicz, "Finite Element Euler Computations in Three Dimensions," *Int. J. Num. Methd Eng.*, Vol. 26, 1988, pp. 2135-2159.

J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher, and C. Taylor (eds.), *Finite Element Methods in Flow Problems*, Proceedings of the International Symposium on Finite Element Methods in Flow Problems held at University College of Swansea, Wales, January 1974, University of Alabama Press, Huntsville, 1974.

Kawahara, M., Yoshimura, N., Nakagawa, K. and Ohsaka, H., "Steady and Unsteady Finite Element Analysis of Incompressible Viscous Fluid," *Int. J. Num. Mthd. Eng.*, Vol. 10, 1976, pp. 437-456.

Lam, C., Applied Numerical Methods for Partial Differential Equations, Prentice Hall, New York, 1994.

O. Hassan, K. Morgan, and J. Peraire, "An Implicit Finite Element Method for High Speed Flows," *Int. J. Num, Mthd Eng.*, Vol. 32, 1991, pp. 183-205.

Ramaswamy, B. and Jue, T. C., "A Segregated Finite Element Formulation of Navier-Stokes Equations under Laminar Conditions," *Finite Element Analysis and Design*, Vol. 9, 199, pp. 257-270.

R. H. Gallagher, J. T. Oden, C. Taylor, and O. C. Zienkiewicz (eds.), *Finite Element in Fluids*, Vols. 1 and 2, Proceeding of the Symposium on Finite Element Methods in Flow Problems held at the University College of Swansea, Wales, 1974, Wiley, New York, 1975.

T. J. R. Hughes, W. K. Liu, and A. Brooks, "Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation," *J. Comput. Phys.*, Vol. 30, No. 1, January 1979, pp. 1-60.

Yamada, Y., Ito, K., Yokouchi, Y., Yamano, T. and Ohtsubo, T., "Finite Element analysis of Steady Fluid and Metal Flow, Finite Element in Fluid, Edited by Gallagher, R. H., et. al., John Wiley and Sons, New York, 1975, pp. 73-94.

Zienkiewicz, O. C. and Taylor, R. L., The Finite Element Method, Fourth Ed., McGraw-Hill International, 1991.

ภาคผนวก

ภาคผนวก ก

รายละเอียดของโปรแกรม STOKES

<u>รายละเอียดของโปรแกรม STOKES</u>

โปรแกรม STOKES จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วย โปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
C       PROGRAM STOKES
C
C       A FINITE ELEMENT COMPUTER PROGRAM FOR SOLVING THE STOKES
C       EQUATIONS OF VISCOUS INCOMPRESSIBLE FLOW WITHOUT INERTIA
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=35, MXPOIP=12, MXELE=12, 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), SOL(MXNEQ)
        CHARACTER*20  NAME1, NAME2
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')
        READ(7,*)     NPOIV, NPOIP, NELEM, NFREE
        WRITE(9,105) NPOIV, NPOIP, NELEM, NFREE
    105 FORMAT(4I8)
```

```
      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')
      READ(7,*)  DEN, VIS
      WRITE(9,135)  DEN, VIS
  135 FORMAT(2E12.4)
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
      NEQ = 2*NPOIV + NPOIP
      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(6,220)  NPOIV, NPOIP, NELEM, NFREE
  220 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,  )
C
C     ESTABLISH ALL ELEMENT MATRICES AND ASSEMBLE THEM TO FORM
C     FORM UP SYSTEM EQUATIONS
C
      WRITE(6,230)
  230 FORMAT(/,' *** ESTABLISHING ELEMENT MATRICES AND',
     *        ' ASSEMBLING ELEMENT EQUATIONS ***'      )
      CALL TRI(NPOIV, NPOIP,  NELEM,  NFREE,     NEQ,
     *              DEN,    VIS,  COORD, INTMAT, INTMATF,
     *              SYSK, SYSR, MXPOIV,  MXELE,  MXFREE, MXNEQ)
C
C     APPLY BOUNDARY CONDITIONS OF NODAL VELOCITIES AND PRESSURE
C
      WRITE(6,240)
  240 FORMAT(/,' *** APPLYING BOUNDARY CONDITIONS OF NODAL',
     *         ' VELOCITIES AND PRESSURES ***'              )
      CALL APPLYBC(NPOIV, NPOIP,  NEQ, IBCU, IBCV,   IBCP,
     *                SYSK,  SYSR, UVEL, VVEL, PRES, MXPOIV,
     *              MXPOIP, MXNEQ                          )
C
C     SOLVE A SET OF SIMULTANEOUS SYSTEM EQUATIONS FOR SOLUTIONS
C
      WRITE(6,250)
  250 FORMAT(/,' *** SOLVING A SET OF SIMULTANEOUS EQS. FOR',
     *         ' VELOCITY AND PRESSURE SOLUTIONS ***'        )
      WRITE(6,260)  NEQ
  260 FORMAT(5X,'( TOTAL OF', I5,' EQUATIONS TO BE SOLVED )')
      CALL GAUSS(NEQ, SYSK, SYSR, SOL, MXNEQ)
C
C     PRINT OUT SOLUTIONS OF NODAL VELOCITIES AND PRESSURES:
C
  270 WRITE(6,280)
  280 FORMAT(/, ' PLEASE ENTER FILE NAME FOR VELOCITY & PRESSURE',
     *          ' SOLUTIONS:', /                                 )
      READ(5, '(A)', ERR=270)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=270)
      WRITE(8,290)  NPOIV
  290 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 295  IEQ=1,NEQ
      IF(SOL(IEQ).LT.ABS(ROFF))  SOL(IEQ) = 0.
  295 CONTINUE
```

```
C
      DO 300  IP=1,NPOIP
      IEQU = IP
      IEQV = NPOIV + IP
      IEQP = 2*NPOIV + IP
      WRITE(8,310)  IP, SOL(IEQU), SOL(IEQV), SOL(IEQP)
  310 FORMAT(I6, 3E16.6)
  300 CONTINUE
      DO 320  IP=NPOIP+1,NPOIV
      IEQU = IP
      IEQV = NPOIV + IP
      WRITE(8,330)  IP, SOL(IEQU), SOL(IEQV)
  330 FORMAT(I6, 2E16.6)
  320 CONTINUE
C
      STOP
      END
C
C-------------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NPOIV, NPOIP,  NEQ, IBCU, IBCV,    IBCP,
     *                     SYSK,  SYSR, UVEL, VVEL, PRES, MXPOIV,
     *                   MXPOIP, MXNEQ                            )
C
C     APPLY BOUNDARY CONDITIONS FOR NODAL VELOCITIES AND PRESSURES
C     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(MXNEQ,MXNEQ), SYSR(MXNEQ)
      DIMENSION  UVEL(MXPOIV), VVEL(MXPOIV), PRES(MXPOIV)
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
      SYSR(IR) = SYSR(IR) - SYSK(IR,IEQ)*UVEL(IEQU)
      SYSK(IR,IEQ) = 0.
  110 CONTINUE
C
      DO 120  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  120 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = UVEL(IEQU)
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
      SYSR(IR) = SYSR(IR) - SYSK(IR,IEQ)*VVEL(IEQV)
      SYSK(IR,IEQ) = 0.
  210 CONTINUE
C
      DO 220  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  220 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = VVEL(IEQV)
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
      SYSR(IR) = SYSR(IR) - SYSK(IR,IEQ)*PRES(IEQP)
      SYSK(IR,IEQ) = 0.
  310 CONTINUE
C
      DO 320  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  320 CONTINUE
      SYSK(IEQ,IEQ) = 1.
      SYSR(IEQ) = PRES(IEQP)
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 OS 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
```

```
      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 MULMAT(A, B, C, I, J, K)
C
C     PERFORM MATRIX MULTIPLICATION: [C(I,K)] = [A(I,J)] [B(J,K)]
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  A(I,J), B(J,K), C(I,K)
C
      DO 10  IR=1,I
      DO 10  IC=1,K
      C(IR,IC) = 0.
      DO 20  IS=1,J
      C(IR,IC) = C(IR,IC) + A(IR,IS)*B(IS,IC)
   20 CONTINUE
   10 CONTINUE
C
      RETURN
      END
C
C---------------------------------------------------------------------
C
      SUBROUTINE TRI(NPOIV, NPOIP,  NELEM,  NFREE,     NEQ,
     *                  DEN,   VIS,  COORD, INTMAT, INTMATF,
     *                  SYSK, SYSR, 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), SYSR(MXNEQ)
      DIMENSION  A(6,6), B(3,6), C(3,6), G(3,3)
      DIMENSION  AT(6,6), BT(6,3), CT(6,3)
      DIMENSION  P66(6,6), P36(3,6), Q36(3,6), P63(6,3)
      DIMENSION  AK11(6,6), AK22(6,6), AK12(6,6), AK21(6,6)
```

```
      DIMENSION  AL1(6,3), AL2(6,3), AL1T(3,6), AL2T(3,6)
      DIMENSION  AKELE(15,15), RELE(15)
C
      INTEGER  INTMAT(MXELE,6), INTMATF(MXFREE,4)
C
C     SET UP [A] MATRIX:
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(4,2) = -1.
      A(4,3) = -1.
      A(5,1) = -1.
      A(5,3) = -1.
      A(6,1) = -1.
      A(6,2) = -1.
C
C     ALSO COMPUTE [A] TRANSPOSE:
C
      DO 20  I=1,6
      DO 20  J=1,6
      AT(J,I) = A(I,J)
   20 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
      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:
```

```
C
      DO 30  I=1,3
      DO 30  J=1,6
      B(I,J) = 0.
      C(I,J) = 0.
   30 CONTINUE
      B(1,1) = 2.*B1
      B(1,5) = B3
      B(1,6) = B2
      B(2,2) = 2.*B2
      B(2,4) = B3
      B(2,6) = B1
      B(3,3) = 2.*B3
      B(3,4) = B2
      B(3,5) = B1
      C(1,1) = 2.*C1
      C(1,5) = C3
      C(1,6) = C2
      C(2,2) = 2.*C2
      C(2,4) = C3
      C(2,6) = C1
      C(3,3) = 2.*C3
      C(3,4) = C2
      C(3,5) = C1
C
C     COMPUTE [B] AND [C] TRANSPOSE:
C
      DO 40  I=1,3
      DO 40  J=1,6
      BT(J,I) = B(I,J)
      CT(J,I) = C(I,J)
   40 CONTINUE
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     COMPUTE [K11] MATRIX (WITHOUT VIS):
C
      CALL MULMAT( B,    A,  P36, 3, 6, 6)
      CALL MULMAT( G, P36,  Q36, 3, 3, 6)
      CALL MULMAT(BT, Q36,  P66, 6, 3, 6)
      CALL MULMAT(AT, P66, AK11, 6, 6, 6)
C
C     COMPUTE [K22] MATRIX (WITHOUT VIS):
C
      CALL MULMAT( C,    A,  P36, 3, 6, 6)
      CALL MULMAT( G, P36,  Q36, 3, 3, 6)
      CALL MULMAT(CT, Q36,  P66, 6, 3, 6)
      CALL MULMAT(AT, P66, AK22, 6, 6, 6)
C
C     COMPUTE [K12] MATRIX (WITHOUT VIS):
C
      CALL MULMAT( B,    A,  P36, 3, 6, 6)
```

```
      CALL MULMAT( G, P36,  Q36, 3, 3, 6)
      CALL MULMAT(CT, Q36,  P66, 6, 3, 6)
      CALL MULMAT(AT, P66, AK12, 6, 6, 6)
C
C     COMPUTE ACTUAL [K11], [K22], [K12], AND [K21]:
C
      DO 50  I=1,6
      DO 50  J=1,6
      AK11(I,J) = VIS*AK11(I,J)
      AK22(I,J) = VIS*AK22(I,J)
      AK12(I,J) = VIS*AK12(I,J)
      AK21(J,I) = AK12(I,J)
   50 CONTINUE
C
C     COMPUTE [L1] AND [L2] MATRICES:
C
      CALL MULMAT(BT,   G, P63, 6, 3, 3)
      CALL MULMAT(AT, P63, AL1, 6, 6, 3)
      CALL MULMAT(CT,   G, P63, 6, 3, 3)
      CALL MULMAT(AT, P63, AL2, 6, 6, 3)
C
      DO 60  I=1,6
      DO 60  J=1,3
      AL1(I,J)  = -AL1(I,J)
      AL2(I,J)  = -AL2(I,J)
      AL1T(J,I) =  AL1(I,J)
      AL2T(J,I) =  AL2(I,J)
   60 CONTINUE
C
C     FORM UP ELEMENT STIFFNESS MATRIX AND LOAD VECTOR:
C
      DO 100  I=1,15
      RELE(I) = 0.
      DO 100  J=1,15
      AKELE(I,J) = 0.
  100 CONTINUE
C
      DO 110  I=1,6
      DO 120  J=1,6
      AKELE(I  ,J  ) = 2.*AK11(I,J) + AK22(I,J)
      AKELE(I+6,J+6) = AK11(I,J) + 2.*AK22(I,J)
      AKELE(I  ,J+6) = AK12(I,J)
      AKELE(I+6,J  ) = AK21(I,J)
  120 CONTINUE
      DO 130  J=1,3
      AKELE(I  ,J+12) = AL1(I,J)
      AKELE(I+6,J+12) = AL2(I,J)
  130 CONTINUE
  110 CONTINUE
      DO 140  I=1,3
      DO 140  J=1,6
      AKELE(I+12,J  ) = AL1T(I,J)
      AKELE(I+12,J+6) = AL2T(I,J)
  140 CONTINUE
CC
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
```

ภาคผนวก ข

รายละเอียดของโปรแกรม NAVIER

<u>รายละเอียดของโปรแกรม NAVIER</u>

โปรแกรม STOKES จะมีรายละเอียดเริ่มจากโปรแกรมหลักและตามด้วย โปรแกรมย่อยต่าง ๆทั้งหมดดังนี้

```
C      PROGRAM NAVIER
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=25, MXPOIP=9, MXELE=8, 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')
      READ(7,*)  DEN, VIS
      WRITE(9,135)  DEN, VIS
  135 FORMAT(2E12.4)
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',
     *         I3, ':')
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,  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.', I5, ' 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 OS 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,  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
      HY(IA,IB) = CY
  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 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
```

ประวัติผู้วิจัย

นายจิตดิน ตรีพุทธรัตน์ เกิดเมื่อวันที่ 24 เดือนพฤษภาคม พุทธศักราช 2511 ที่ อำเภอโพธาราม จังหวัดราชบุรี สำเร็จการศึกษาปริญญาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมเครื่องกล ภาควิชาวิศกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จากมหาวิทยาลัยรังสิต เมื่อปีการศึกษา 2534 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัยเมื่อปีการศึกษา 2536