รายการอ้างอิง

1. Rastogi, A. K. and Rodi, W. Predictions of heat and mass transfer in open channels. Journal of the Hydraulics Division 104(HY3) (1978): 397-420.

2. Vreugdenhill, C. B. and Wijbenga, J. H. A. Computation of flow patterns in rivers. Journal of the Hydraulics Division 108(HY11) (1982): 1296-1310.

3. Demuran, A. O. and Rodi, W. Calculation of flow and pollutant dispersion in meandering channels. Journal of Fluid Mechanics 172 (1986): 63-92.

4. Molls, T. and Chaudhry, M. H. Depth-averaged open-channel flow model. Journal of Hydraulic Engineering 121(6) (1995): 453-465.

5. Borthwick, A. G. L. and Akponasa, G. A. Reservoir flow prediction by contravariant shallow water equations. Journal of Hydraulic Engineering 123(5) (1997): 432-439.

6. Zhou, J. G. and Goodwill, I. M. A finite volume method for steady state 2D shallow water flows. International Journal of Numerical Methods for Heat and Fluid Flow 7(1) (1997): 4-23.

7. Yu, L. and Righetto, A. M. Depth-averaged turbulence $\tilde{k}-\tilde{w}$ model and applications. Advances in Engineering Software 32 (2001): 375-394.

8. Zienkiewicz, O. C. and Heinrich, J. C. A unified treatment of steady-state shallow water and two-dimensional Navier-Stokes equations–finite element approach. Computer Methods in Applied Mechanics and Engineering 17/18 (1979): 673-698.

9. Cochet, J. F., Dhatt, D., Hubert, G. and Touzot, G. River and estuary flows by a new penalty finite element. In T. Kawai (ed.), Finite Element Flow Analysis, pp. 563-570. Tokyo: University of Tokyo Press, 1982.

10. Leclerc, M., Bellemare, J., Dumas, G. and Dhatt, G. A finite element model of estuarian and river flows with moving boundaries. Advances in Water Resources 13(4) (1990): 158-168.

11. Shrestha, P. L. An integrated model suite for sediment pollutant transport in shallow lakes. Advances in Engineering Software 27 (1996): 201-212.

12. Tabuenca, P., Vila, J., Cardona, J. and Samartin, A. Finite element simulation of dispersion in the bay of Santander. Advances in Engineering Software 28 (1997): 313-332.

13. Peraire, J., Vahjdati, M., Morgan, K. and Zienkiewicz, O. C. Adaptive remeshing for compressible flow computation. Journal of Computational Physics 72 (1987): 449-466.

14. Dechaumphai, P. and Janphaisaeng, P. Adaptive finite element technique for high-speed compressible flows. Thammasat International Journal of Science and Technology. 3(1) (1988).

15. Huebner, K. H., Thornton, E. A. and Byrom, T. G. The Finite Element Method for Engineers. 3 rd ed. New York: John Wiley & Sons, 1995.

16. Dechaumphai, P. Transient finite element thermal analysis using adaptive mesh movement. Journal of Energy, Heat and Mass Transfer 19 (1997): 153-158.

17. สุพัฒนพงศ์ สิกขาบัณฑิต. เทคนิคการปรับขนาดไฟไนต์เอลิเมนต์เพื่อการวิเคราะห์การไหลแบบหนืด. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2541.

18. ปราโมทย์ เดชะอำไพ. ระเบียบวิธีไฟไนต์เอลิเมนต์เพื่อการคำนวณพลศาสตร์ของไหล. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2545.

19. Zienkiewicz, O. C. and Taylor, R. L. The Finite Element Method. Vol. 3: Fluid Dynamics. 5 th ed. Oxford: Butterworth-Heinemann, 2000.

20. ปราโมทย์ เดชะอำไพ. ไฟไนต์เอลิเมนต์ในงานวิศวกรรม. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2542.

21. Ramaswamy, B. and Jue, T. C. A segregated finite element formulation of Navier-Stokes equations under laminar conditions. Finite Elements in Analysis and Design 9 (1991): 257-270.

22. Guillou, S. and Nguyen, K. D. An improved technique for solving two-dimensional shallow water problems. International Journal for numerical methods in fluid 29 (1999): 465-483.

23. Mase, G. E. and Mase, G. T. Continuum Mechanics for Engineers. Boca Raton: CRC Press, 1992.

24. เจ้าท่า, กรม. แผนที่ความลึกของแม่น้ำเจ้าพระยาจังหวัดสมุทรปราการ. กรุงเทพมหานคร: กรมเจ้าท่า, 2539.

ภาคผนวก

ภาคผนวก ก

รายละเอียดของโปรแกรม SWFLOW

โปรแกรมคอมพิวเตอร์ SWFLOW ที่ประดิษฐ์ขึ้นดังที่ได้กล่าวไว้ในบทที่ 4 มีรายละเอียดดังนี้

```
C
C     PROGRAM SWFLOW
C
C     A FINITE ELEMENT COMPUTER PROGRAM FOR SOLVING
C     SHALLOW WATER FLOW EQUATIONS
C
C                    MR.SOMBOON OTARAWANNA
C                    MECHANICAL ENGINEERING DEPARTMANT
C                    FACULTY OF ENGINEERING
C                    CHULALONGKORN UNIVERSITY
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 WATER LEVEL NODES IN THE MODEL
C         MXELE  = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C
      PARAMETER (MXPOIV=441, MXPOIP=121, MXELE=200, MXFLUX=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), SH(MXPOIV)
     *           ,CC(MXPOIV)
      DIMENSION  SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ)
      DIMENSION  SOL(MXNEQ), DSOL(MXNEQ)
      CHARACTER*20  NAME1, NAME2, NAME3, NAME4
C
      INTEGER  INTMAT(MXELE,6), INTFLUX(MXFLUX,3)
      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   NFLUX   NITER    TOL')
      READ(7,*)    NPOIV, NPOIP, NELEM, NFLUX, NITER, TOL
      WRITE(9,105) NPOIV, NPOIP, NELEM, NFLUX, 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(NFLUX.GT.MXFLUX)  WRITE(6,140) NFLUX
  140 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXFLUX TO',I5)
      IF(NFLUX.GT.MXFLUX)  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), SH(I), CC(I)
      WRITE(9,152) I, IBCU(I), IBCV(I), IBCP(I),
     *(COORD(I,K), K=1,2), UVEL(I), VVEL(I), PRES(I), SH(I), CC(I)
  152 FORMAT(I6, 3I4, 7E12.4)
      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 FLUX BOUNDARY (FLOW INLET) INFORMATION:
C
```

```
      READ(7,1)  TEXT
      WRITE(9,168)  NFLUX
  168 FORMAT(' OUTFLOW INFORMATION (ELE NO., 3 NODE NO.): [',
     *       I4, ']')
      DO 170  IB=1,NFLUX
      READ(7,*)     (INTFLUX(IB,J), J=1,3)
      WRITE(9,172) (INTFLUX(IB,J), J=1,3)
  172 FORMAT(4I8)
  170 CONTINUE
C
      WRITE(6,200)  NPOIV, NPOIP, NELEM, NFLUX, NITER, TOL
  200 FORMAT('     THE FINITE ELEMENT MODEL CONSISTS OF:', /,
     *       '          NUMBER OF VELOCITY NODES       =', I6, /,
     *       '          NUMBER OF WATER LEVEL NODES    =', I6, /,
     *       '          NUMBER OF ELEMENTS             =', I6, /,
     *       '          NUMBER OF INFLOW 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',
     *       I6, ':')
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,   NFLUX,   NEQ,  DEN, GRA,
     *          VIS,  COORD, INTMAT, INTMATF,  SYSK, SYSR,
     *          SOL, MXPOIV,  MXELE,  MXFLUX, MXNEQ, SH, CC)
C
      IF (NFLUX.NE.0) THEN
      CALL VELTRACT(SYSR, COORD, INTFLUX, UVEL, MXPOIV,
     *     MXNEQ, MXFLUX, SOL, SH, SYSK)
      ENDIF
```

```
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
C
  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
      DO 201  I=1,3
      DO 201  J=1,3
      II = INTMAT(IE,I)
      JJ = INTMAT(IE,J)
      K  = I + 12
      L  = J + 12
      KK = 2*NPOIV + II
      LL = 2*NPOIV + JJ
      SYSK(KK,LL) = SYSK(KK,LL) + AKELE(K,L)
  201 CONTINUE
C
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 aSCALE(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 aSCALE(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,   NFLUX,   NEQ,  DEN,
GRA,
     *               VIS,  COORD, INTMAT, INTMATF,  SYSK, SYSR,
     *               SOL, MXPOIV,  MXELE,  MXFLUX, MXNEQ, SH, CC)
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), SHELE(3), SH(MXPOIV)
      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),
     *AI(3,6), ZX(3,6), ZY(3,6), Q(3,3), HMX(3), HMY(3)
      DIMENSION  AKELE(15,15), RELE(15), FX(6), FY(6), FI(3)
      DIMENSION  CC(MXPOIV), EXL(6,6), CAB(6,6), CCELE(3)
C
      INTEGER  INTMAT(MXELE,6), INTFLUX(MXFLUX,3)
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 [I]
C
      AFAC  = AREA/60.
      AFAC2 = 2.*AFAC
      AFAC6 = 6.*AFAC
      AI(1,1)= AFAC6
      AI(1,2)= AFAC2
      AI(1,3)= AFAC2
      AI(1,4)= AFAC
      AI(1,5)= AFAC2
      AI(1,6)= AFAC2
      AI(2,1)= AFAC2
      AI(2,2)= AFAC6
      AI(2,3)= AFAC2
      AI(2,4)= AFAC2
      AI(2,5)= AFAC
      AI(2,6)= AFAC2
      AI(3,1)= AFAC2
      AI(3,2)= AFAC2
      AI(3,3)= AFAC6
      AI(3,4)= AFAC2
      AI(3,5)= AFAC2
      AI(3,6)= AFAC
C
C     SET UP [HMX]
C
      HMX(1) = B1
      HMX(2) = B2
      HMX(3) = B3
C
C     SET UP [HMY]
C
      HMY(1) = C1
      HMY(2) = C2
      HMY(3) = C3
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)
      SHELE(1)=SH(II)
      SHELE(2)=SH(JJ)
      SHELE(3)=SH(KK)
      CCELE(1)=CC(II)
      CCELE(2)=CC(JJ)
      CCELE(3)=CC(KK)
C
C     SETUP U, V, P, h BAR OF ELEMENT
C
      UELEB = (UELE(1)+UELE(2)+UELE(3))/3.
      VELEB = (VELE(1)+VELE(2)+VELE(3))/3.
      PELEB = (PELE(1)+PELE(2)+PELE(3))/3.
      SHELEB = (SHELE(1)+SHELE(2)+SHELE(3))/3.
      CCELEB = (CCELE(1)+CCELE(2)+CCELE(3))/3.
C
C     COMPUTE  TERMS ASSOCIATE WITH BOTTOM FRICTION
C
      EXL(1,1) = 2./720.*AREA*24.
      EXL(1,2) = 2./720.*AREA*4.
      EXL(1,3) = 2./720.*AREA*4
      EXL(1,4) = 2./720.*AREA*2.
      EXL(1,5) = 2./720.*AREA*6.
      EXL(1,6) = 2./720.*AREA*6.
      EXL(2,1) = 2./720.*AREA*4.
      EXL(2,2) = 2./720.*AREA*24.
      EXL(2,3) = 2./720.*AREA*4.
      EXL(2,4) = 2./720.*AREA*6.
      EXL(2,5) = 2./720.*AREA*2.
      EXL(2,6) = 2./720.*AREA*6.
      EXL(3,1) = 2./720.*AREA*4.
      EXL(3,2) = 2./720.*AREA*4.
```

```
      EXL(3,3) = 2./720.*AREA*24.
      EXL(3,4) = 2./720.*AREA*6.
      EXL(3,5) = 2./720.*AREA*6.
      EXL(3,6) = 2./720.*AREA*2.
      EXL(4,1) = 2./720.*AREA*2.
      EXL(4,2) = 2./720.*AREA*6.
      EXL(4,3) = 2./720.*AREA*6.
      EXL(4,4) = 2./720.*AREA*4.
      EXL(4,5) = 2./720.*AREA*2.
      EXL(4,6) = 2./720.*AREA*2.
      EXL(5,1) = 2./720.*AREA*6.
      EXL(5,2) = 2./720.*AREA*2.
      EXL(5,3) = 2./720.*AREA*6.
      EXL(5,4) = 2./720.*AREA*2.
      EXL(5,5) = 2./720.*AREA*4.
      EXL(5,6) = 2./720.*AREA*2.
      EXL(6,1) = 2./720.*AREA*6.
      EXL(6,2) = 2./720.*AREA*6.
      EXL(6,3) = 2./720.*AREA*2.
      EXL(6,4) = 2./720.*AREA*2.
      EXL(6,5) = 2./720.*AREA*2.
      EXL(6,6) = 2./720.*AREA*4.
      DO 901  IA=1,6
      DO 901  IB=1,6
      TRAN = 0.
      DO 902  I=1,6
      DO 902  J=1,6
      TRAN = TRAN+(GRA*SQRT(UELEB**2.+VELEB**2.)/CCELEB**2.)
     */(PELEB+SHELEB)*A(IA,I)*A(IB,J)*EXL(I,J)
  902 CONTINUE
      CAB(IA,IB) = TRAN
  901 CONTINUE
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
      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*GRA
      HY(IA,IB) = CY*GRA
  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
      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
     *        + A(IA,I)*A(IB,J)*A(K,L)*B(L,M)*F(I,J,M)*UELE(K)
      CAGBXUG = CAGBXUG
     *        + A(IA,I)*A(K,J)*A(IB,L)*B(L,M)*F(I,J,M)*UELE(K)
      CAGBYVG = CAGBYVG
     *        + A(IA,I)*A(K,J)*A(IB,L)*C(L,M)*F(I,J,M)*VELE(K)
      CABGYVG = CABGYVG
     *        + A(IA,I)*A(IB,J)*A(K,L)*C(L,M)*F(I,J,M)*VELE(K)
      CABGXVG = CABGXVG
     *        + A(IA,I)*A(IB,J)*A(K,L)*B(L,M)*F(I,J,M)*VELE(K)
      CABGYUG = CABGYUG
     *        + 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)
     *                  + CAB(I,J)
      GYY(I,J) = ABGYVG(I,J) + AGBYVG(I,J) + AGBXUG(I,J) + SYY(I,J)
     *                  + CAB(I,J)
      ALX(I,J) = ABGXVG(I,J) + SXY(I,J)
```

```
      ALY(I,J) = ABGYUG(I,J) + SYX(I,J)
  220 CONTINUE
C
C     SET UP [ZX], [ZY] AND [Q]
C
      DO 600 IA=1,3
      DO 600 IB=1,6
      AJMBGX=0.
      AJMBGY=0.
      DO 610 I=1,6
      DO 610 J=1,3
      AJMBGX = AJMBGX+A(IB,I)*HMX(IA)*AI(J,I)*(SHELE(J)+PELE(J))
      AJMBGY = AJMBGY+A(IB,I)*HMY(IA)*AI(J,I)*(SHELE(J)+PELE(J))
  610 CONTINUE
      ZX(IA,IB) = AJMBGX
      ZY(IA,IB) = AJMBGY
  600 CONTINUE
      DO 700 IA=1,3
      DO 700 IB=1,3
      BJMBGX=0.
      BJMBGY=0.
      DO 710 I=1,6
      DO 710 J=1,6
      BJMBGX = BJMBGX+A(I,J)*HMX(IB)*AI(IA,J)*UELE(I)
      BJMBGY = BJMBGY+A(I,J)*HMY(IB)*AI(IA,J)*VELE(I)
  710 CONTINUE
      Q(IA,IB) = BJMBGX+BJMBGY
  700 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  ) = ZX(I,J)
      AKELE(I+12,J+6) = ZY(I,J)
  270 CONTINUE
      DO 280  I=1,3
      DO 280  J=1,3
      AKELE(I+12,J+12) = Q(I,J)
  280 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.
      TERM6 = 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)
      TERM6 = TERM6 + CAB(I,J)*UELE(J)
  310 CONTINUE
      DO 320  J=1,3
      TERM3 = TERM3 + HXT(I,J)*PELE(J)
  320 CONTINUE
      FX(I) = TERM1 + TERM2 + TERM3 + TERM4 + TERM5+TERM6
  300 CONTINUE
C
      DO 350  I=1,6
      TERM1 = 0.
      TERM2 = 0.
      TERM3 = 0.
      TERM4 = 0.
      TERM5 = 0.
      TERM6 = 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)
      TERM6 = TERM6 + CAB(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+TERM6
  350 CONTINUE
C
      DO 400  I=1,3
      TERM1 = 0.
      TERM2 = 0.
      DO 410  J=1,6
      TERM1 = TERM1 + ZX(I,J)*UELE(J)
      TERM2 = TERM2 + ZY(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
      SUBROUTINE VELTRACT(SYSR, COORD, INTFLUX, UVEL, MXPOIV,
     *     MXNEQ, MXFLUX, SOL, SH, SYSK)
C
      IMPLICIT REAL*8(A-H, O-Z)
C
      DIMENSION SYSR(MXNEQ), SYSK(MXNEQ,MXNEQ)
      DIMENSION COORD(MXPOIV,2), INTFLUX(MXFLUX,3)
      DIMENSION UVEL(MXPOIV), SOL(MXNEQ), SH(MXPOIV), R(3,3)
C
      DO 10 I=1,MXFLUX
C
C     FIND BOUNDARY LOCAL COORDINATES:
C
      JJ=INTFLUX(I,2)
      KK=INTFLUX(I,3)
      X1=COORD(JJ,1)
      X2=COORD(KK,1)
      Y1=COORD(JJ,2)
      Y2=COORD(KK,2)
C
C     CALCULATE LENGTH OF BOUNDARY:
C
      DX=X2-X1
      DY=Y2-Y1
      DL=SQRT(DX*DX+DY*DY)
C
C     CALCULATE VELOCITY TRACTION THAT FLOW INTO DOMAIN:
C
      R(2,2)=DL*(UVEL(JJ)/4.+UVEL(KK)/12.)
      R(2,3)=DL*(UVEL(JJ)/12.+UVEL(KK)/12.)
      R(3,2)=DL*(UVEL(JJ)/12.+UVEL(KK)/12.)
      R(3,3)=DL*(UVEL(JJ)/12.+UVEL(KK)/4.)
C
C     ASSEMBLING VELOCITY TRACTION INTO SYSTEM LOAD VECTOR
C
      SYSR(2*MXPOIV+JJ)=SYSR(2*MXPOIV+JJ)-R(2,2)*
     *(SOL(2*MXPOIV+JJ)+SH(JJ))-R(2,3)*
     *(SOL(2*MXPOIV+KK)+SH(KK))
      SYSR(2*MXPOIV+kk)=SYSR(2*MXPOIV+KK)-R(3,2)*
     *(SOL(2*MXPOIV+JJ)+SH(JJ))-R(3,3)*
     *(SOL(2*MXPOIV+KK)+SH(KK))
C
C     ASSEMBLING STIFFNESS MATRIX ASSCIATED TO VOLUME FLOWRATE
C
      SYSK(2*MXPOIV+JJ,2*MXPOIV+JJ)=SYSK(2*MXPOIV+JJ,2*MXPOIV+JJ)
     *+R(2,2)
      SYSK(2*MXPOIV+JJ,2*MXPOIV+KK)=SYSK(2*MXPOIV+JJ,2*MXPOIV+KK)
     *+R(2,3)
      SYSK(2*MXPOIV+KK,2*MXPOIV+JJ)=SYSK(2*MXPOIV+KK,2*MXPOIV+JJ)
     *+R(3,2)
      SYSK(2*MXPOIV+KK,2*MXPOIV+KK)=SYSK(2*MXPOIV+KK,2*MXPOIV+KK)
```

```
     *+R(3,3)
C
   10 CONTINUE
C
      RETURN
      END
```

ภาคผนวก ข

รายละเอียดของโปรแกรม SWCONC

โปรแกรมคอมพิวเตอร์ SWCONC ที่ประดิษฐ์ขึ้นดังที่ได้กล่าวไว้ในบทที่ 4 มีรายละเอียดดังนี้

```
C
C     PROGRAM SWCONC
C
C     A FINITE ELEMENT COMPUTER PROGRAM FOR SOLVING
C     2D DEPTH-AVARAGED TRANSIENT
C     POLLUTANT TRANSPORT EQUATION
C
C     USING LUMPED CAPACITANCE MATRIX [C]
C
C                    MR.SOMBOON OTARAWANNA
C                    MECHANICAL ENGINEERING DEPARTMANT
C                    FACULTY OF ENGINEERING
C                    CHULALONGKORN UNIVERSITY
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         MXPOIT = MAXIMUM NUMBER OF NODES IN THE MODEL
C         MXELE  = MAXIMUM NUMBER OF ELEMENTS IN THE MODEL
C
      PARAMETER (MXPOIT=909,MXELE=1594, MXFLUX=1)
      PARAMETER (MXNEQ=MXPOIT)
C
      IMPLICIT  REAL*8 (A-H,O-Z)
      DIMENSION  COORD(MXPOIT,2), TEXT(20)
      DIMENSION  UVEL(MXPOIT), VVEL(MXPOIT), TEMP(MXPOIT)
      DIMENSION  FLUX(MXPOIT)
      DIMENSION  SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ)
      DIMENSION  SYSC(MXNEQ,MXNEQ), SYSAKCV(MXNEQ,MXNEQ)
      DIMENSION  SH(MXPOIT), ZETA(MXPOIT)
      CHARACTER*20  NAME1, NAME2, NAME3, NAME4
C
      INTEGER  INTMAT(MXELE,3), INTFLUX(MXFLUX,3)
      INTEGER  IBCT(MXPOIT)
C
   11 WRITE(6,21)
   21 FORMAT(/,' PLEASE ENTER NSHOW:', /)
      READ(*,*) NSHOW
   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,*)   NPOIT, NELEM, NFLUX
C
      IF(NPOIT.GT.MXPOIT)  WRITE(6,110) NPOIT
  110 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXPOIT TO',I5)
      IF(NPOIT.GT.MXPOIT)  STOP
      IF(NFLUX.GT.MXFLUX)  WRITE(6,120) NFLUX
  120 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXFLUX TO',I5)
      IF(NFLUX.GT.MXFLUX)  STOP
      IF(NELEM.GT.MXELE)  WRITE(6,130) NELEM
  130 FORMAT(/,' PLEASE INCREASE THE PARAMETER MXELE TO',I5)
      IF(NELEM.GT.MXELE)  STOP
C
C     READ FLUID PROPERTIES:
C
      READ(7,1)  TEXT
      READ(7,*)  DEN, TCON, CV, DT, NDT
C
C     READ NODAL COORDINATES, BOUNDARY CONDITIONS, THEIR VALUES:
C
      READ(7,1)  TEXT
C
      DO 150  IP=1,NPOIT
      READ(7,*)     I, IBCT(I),
     *     (COORD(I,K), K=1,2), UVEL(I), VVEL(I), TEMP(I), FLUX(I),
     *      SH(I), ZETA(I)
C
      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
C
      DO 160  IE=1,NELEM
      READ(7,*)     I, (INTMAT(I,J), J=1,3)
C
      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 SPECIFIED FLUX BC:
C
      READ(7,1) TEXT
      DO 170 IB=1, MXFLUX
      READ(7,*) (INTFLUX(IB,J), J=1,3)
  170 CONTINUE
C
C     START THE COMPUTATION:
C
      WRITE(6,200)  NPOIT, NELEM, NFLUX
  200 FORMAT('     THE FINITE ELEMENT MODEL CONSISTS OF:', /,
     *       '          NUMBER OF NODES           =', I6, /,
     *       '          NUMBER OF ELEMENTS        =', I6, /,
```

```
     *       '    NUMBER OF HEAT FLUX BOUNDARY   =', I6 )
C
      NEQ = NPOIT
C-------------------------
C     ENTER TIME LOOP
C-------------------------
      ISHOW = 1
      DO 500 NT=1,NDT
      TIME = NT*DT
      WRITE(6,203) NT, TIME
  203 FORMAT('TIME STEP NO.',I6,4X,'TIME=',F12.2,/)
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
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(NPOIT,  NELEM,   NEQ,  DEN, TCON, CV, UVEL, VVEL,
     *          COORD, INTMAT, SYSC, SYSAKCV,
     *          TEMP, MXPOIT, MXELE, MXNEQ, DT, SH, ZETA )
C
      CALL RECUR(NEQ, SYSK, SYSC, DT, SYSR, SYSAKCV,
     *                            TEMP, 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(NPOIT,  NEQ,   IBCT,  TEMP,
     *             SYSK,  SYSR, MXPOIT, MXNEQ      )
      CALL HEATFLUX(SYSR, COORD, INTFLUX, FLUX, MXPOIT,
     *          MXNEQ, MXFLUX, DEN, CV   )
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 )')
C
      CALL GAUSS(NEQ, SYSK, SYSR, TEMP, MXNEQ)
C
C     PRINT OUT SOLUTIONS:
C
      IF (NT.NE.1) GO TO 202
  710 WRITE(6,720)
  720 FORMAT(' PLEASE ENTER THE OUTPUT FILE NAME:')
```

```
      READ(5, '(A)', ERR=710)  NAME2
      OPEN(UNIT=8, FILE=NAME2, STATUS='NEW', ERR=710)
  202 CONTINUE
C
      IF ((MOD(NT,NSHOW).EQ.0).OR.(NT.EQ.1)) THEN
C
      WRITE(8,201) NT, TIME
  201 FORMAT('TIME STEP NO.',I6,4X,'TIME=',F12.2)
C
      WRITE(8,730)  NPOIT
  730 FORMAT('  NODAL TEMPERATURE SOLUTIONS [', I5,']:',
     *       /, 2X,  'NODE', 6X, 'TEMPERATURE', 6X,'U-VELOCITY',
     *        6X, 'V-VELOCITY',  8X, 'PRESSURE', /)
C
      DO 750  IP=1,NPOIT
      IEQT = IP
      WRITE(8,760)  IP, TEMP(IEQT)
  760 FORMAT(I6, E16.6)
  750 CONTINUE
      ENDIF
C-------------------------
C     EXIT TIME LOOP
C-------------------------
  500 CONTINUE
C
      STOP
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE APPLYBC(NPOIT,  NEQ,   IBCT,    TEMP,
     *                   SYSK,  SYSR, MXPOIT, 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),  TEMP(MXPOIT)
C
      INTEGER  IBCT(MXPOIT)
C
C     APPLY BOUNDARY CONDITIONS:
C
      IEQ1 = 1
      IEQ2 = NPOIT
      DO 100  IEQ=IEQ1,IEQ2
      IEQT = IEQ
      IF(IBCT(IEQT).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)*TEMP(IEQ)
      SYSK(IR,IEQ) = 0.
  110 CONTINUE
C
      DO 120  IC=1,NEQ
      SYSK(IEQ,IC) = 0.
  120 CONTINUE
      SYSK(IEQ,IEQ) = 1.
```

```
      SYSR(IEQ) = TEMP(IEQ)
C
  100 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE ASSMBLE(   IE, INTMAT, AKELE,  RELE,  SYSK, SYSR,
     *                   NPOIT,    NEQ, NELEM, MXNEQ, MXELE       )
C
C     ASSEMBLE ELEMENT EQUATIONS INTO SYSTEM EQUATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  AKELE(3,3), RELE(3,3)
      DIMENSION  SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ,MXNEQ)
C
      INTEGER  INTMAT(MXELE,3)
C
C     ASSEMBLING SYSTEM STIFFNESS MATRIX:
C
      DO 100  I=1,3
      DO 100  J=1,3
      II = INTMAT(IE,I)
      JJ = INTMAT(IE,J)
      SYSK(II,JJ) = SYSK(II,JJ) + AKELE(I,J)
  100 CONTINUE
C
C     ASSEMBLING SYSTEM LOAD VECTOR:
C
      DO 101  I=1,3
      DO 101  J=1,3
      II = INTMAT(IE,I)
      JJ = INTMAT(IE,J)
      SYSR(II,JJ) = SYSR(II,JJ) + RELE(I,J)
  101 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(NPOIT,  NELEM,  NEQ,  DEN, TCON, CV, UVEL,
     *           VVEL, COORD, INTMAT,  SYSC, SYSAKCV,
     *           TEMP, MXPOIT,  MXELE,  MXNEQ, DT, SH, ZETA )
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(MXPOIT,2), UVEL(MXPOIT), VVEL(MXPOIT)
      DIMENSION  SYSC(MXNEQ,MXNEQ), SYSAKCV(MXNEQ,MXNEQ), TEMP(MXNEQ)
      DIMENSION  C(3,3), AKCV(3,3)
      DIMENSION  SH(MXPOIT), ZETA(MXPOIT), SHELE(3), ZETAELE(3)
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
      SHELE(1) = SH(II)
      SHELE(2) = SH(JJ)
      SHELE(3) = SH(KK)
      ZETAELE(1) = ZETA(II)
      ZETAELE(2) = ZETA(JJ)
      ZETAELE(3) = ZETA(KK)
      UAV = (UVEL(II)+UVEL(JJ)+UVEL(KK))/3.
      VAV = (VVEL(II)+VVEL(JJ)+VVEL(KK))/3.
      SHAV = (SH(II)+SH(JJ)+SH(KK))/3.
      ZETAAV = (ZETA(II)+ZETA(JJ)+ZETA(KK))/3.
C
      C(1,1) = DEN*CV*AREA/3.*1.
      C(1,2) = 0.
      C(1,3) = 0.
      C(2,1) = 0.
      C(2,2) = DEN*CV*AREA/3.*1.
      C(2,3) = 0.
      C(3,1) = 0.
      C(3,2) = 0.
      C(3,3) = DEN*CV*AREA/3.*1.
C
      AKCV(1,1)= TCON/4./AREA*(B1*B1+C1*C1)
     *+DEN*CV/6.*(UAV*B1+VAV*C1)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B1*(ZETAELE(1)+SHELE(1))+B1*B2*(ZETAELE(2)+SHELE(2))
     *+B1*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C1*(ZETAELE(1)+SHELE(1))+C1*C2*(ZETAELE(2)+SHELE(2))
     *+C1*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(1,2)= TCON/4./AREA*(B1*B2+C1*C2)
     *+DEN*CV/6.*(UAV*B2+VAV*C2)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B2*(ZETAELE(1)+SHELE(1))+B2*B2*(ZETAELE(2)+SHELE(2))
     *+B2*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C2*(ZETAELE(1)+SHELE(1))+C2*C2*(ZETAELE(2)+SHELE(2))
     *+C2*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(1,3)= TCON/4./AREA*(B1*B3+C1*C3)
     *+DEN*CV/6.*(UAV*B3+VAV*C3)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B3*(ZETAELE(1)+SHELE(1))+B2*B3*(ZETAELE(2)+SHELE(2))
     *+B3*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C3*(ZETAELE(1)+SHELE(1))+C2*C3*(ZETAELE(2)+SHELE(2))
     *+C3*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(2,1)= TCON/4./AREA*(B1*B2+C1*C2)
     *+DEN*CV/6.*(UAV*B1+VAV*C1)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B1*(ZETAELE(1)+SHELE(1))+B1*B2*(ZETAELE(2)+SHELE(2))
     *+B1*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C1*(ZETAELE(1)+SHELE(1))+C1*C2*(ZETAELE(2)+SHELE(2))
```

```
     *+C1*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(2,2)= TCON/4./AREA*(B2*B2+C2*C2)
     *+DEN*CV/6.*(UAV*B2+VAV*C2)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B2*(ZETAELE(1)+SHELE(1))+B2*B2*(ZETAELE(2)+SHELE(2))
     *+B2*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C2*(ZETAELE(1)+SHELE(1))+C2*C2*(ZETAELE(2)+SHELE(2))
     *+C2*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(2,3)= TCON/4./AREA*(B2*B3+C2*C3)
     *+DEN*CV/6.*(UAV*B3+VAV*C3)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B3*(ZETAELE(1)+SHELE(1))+B2*B3*(ZETAELE(2)+SHELE(2))
     *+B3*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C3*(ZETAELE(1)+SHELE(1))+C2*C3*(ZETAELE(2)+SHELE(2))
     *+C3*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(3,1)= TCON/4./AREA*(B1*B3+C1*C3)
     *+DEN*CV/6.*(UAV*B1+VAV*C1)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B1*(ZETAELE(1)+SHELE(1))+B1*B2*(ZETAELE(2)+SHELE(2))
     *+B1*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C1*(ZETAELE(1)+SHELE(1))+C1*C2*(ZETAELE(2)+SHELE(2))
     *+C1*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(3,2)= TCON/4./AREA*(B2*B3+C2*C3)
     *+DEN*CV/6.*(UAV*B2+VAV*C2)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B2*(ZETAELE(1)+SHELE(1))+B2*B2*(ZETAELE(2)+SHELE(2))
     *+B2*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C2*(ZETAELE(1)+SHELE(1))+C2*C2*(ZETAELE(2)+SHELE(2))
     *+C2*C3*(ZETAELE(3)+SHELE(3)))
      AKCV(3,3)= TCON/4./AREA*(B3*B3+C3*C3)
     *+DEN*CV/6.*(UAV*B3+VAV*C3)-TCON/12./AREA/(ZETAAV+SHAV)*
     *(B1*B3*(ZETAELE(1)+SHELE(1))+B2*B3*(ZETAELE(2)+SHELE(2))
     *+B3*B3*(ZETAELE(3)+SHELE(3))
     *+C1*C3*(ZETAELE(1)+SHELE(1))+C2*C3*(ZETAELE(2)+SHELE(2))
     *+C3*C3*(ZETAELE(3)+SHELE(3)))
C
C     ASSEMBLE THESE ELEMENT MATRICES TO FORM SYSTEM EQUATIONS:
C
      CALL ASSMBLE(   IE, INTMAT, C,  AKCV,  SYSC, SYSAKCV,
     *             NPOIT,    NEQ, NELEM, MXNEQ, MXELE      )
C
  500 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE HEATFLUX( SYSR,COORD,INTFLUX, FLUX, MXPOIT, MXNEQ,
     *                     MXFLUX,   DEN,     CV )
C
      IMPLICIT  REAL*8 (A-H,O-Z)
C
      DIMENSION  SYSR(MXPOIT),FLUX(MXPOIT)
      DIMENSION  COORD(MXPOIT,2),INTFLUX(MXFLUX,3)
C
      DO 10 I=1,MXFLUX
C
C     FIND BOUNDARY LOCAL COORDINATES:
C
      JJ = INTFLUX(I,2)
      KK = INTFLUX(I,3)
      X1 = COORD(JJ,1)
      X2 = COORD(KK,1)
      Y1 = COORD(JJ,2)
```

```
      Y2 = COORD(KK,2)
C
C     CALCULATE LENGTH OF BOUNDARY:
C
      DX = X2 - X1
      DY = Y2 - Y1
      DL = SQRT(DX*DX + DY*DY)
C
C     ASSEMBLING HEAT FLUX INTO SYSTEM LOAD VECTOR:
C
      SYSR(JJ)=SYSR(JJ)+FLUX(JJ)*DL/2.
      SYSR(KK)=SYSR(KK)+FLUX(KK)*DL/2.
C
   10 CONTINUE
C
      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 RECUR(NEQ, SYSK, SYSC, DT, SYSR, SYSAKCV,
     *                         TEMP, MXNEQ)
C
C     APPLY RECURRENCE RELATIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SYSK(MXNEQ,MXNEQ), SYSR(MXNEQ),  SYSC(MXNEQ,MXNEQ),
     *          RR(MXNEQ,MXNEQ), SYSAKCV(MXNEQ,MXNEQ), TEMP(MXNEQ)
      DO 1 I=1,NEQ
      DO 1 J=1,NEQ
      SYSK(I,J)=0.
    1 CONTINUE
      DO 2 I=1,NEQ
      DO 2 J=1,NEQ
      SYSK(I,J)=SYSC(I,J)/DT
    2 CONTINUE
      DO 3 I=1,NEQ
      SYSR(I)=0.
    3 CONTINUE
      DO 4 I=1,NEQ
      DO 4 J=1,NEQ
```

```
      RR(I,J)=SYSC(I,J)/DT-SYSAKCV(I,J)
    4 CONTINUE
      CALL MULMAT(RR, TEMP, SYSR, NEQ, NEQ, 1)
      RETURN
      END
C
C-----------------------------------------------------------------------
C
```

ภาคผนวก ค

การเปรียบเทียบระหว่างสมการของการไหลในสองมิติเฉลี่ยตลอดความลึก กับสมการของการไหลในสองมิติ

หากละทิ้งพจน์ที่เกี่ยวข้องกับความเสียดทานที่พื้นน้ำ สมการที่ครอบคลุมการไหลในน้ำตื้นซึ่งพิจารณาเป็นปัญหาในสองมิติเฉลี่ยตลอดความลึกที่สภาวะอยู่ตัว (2.34a-c) ในบทที่ 2 จะสามารถเขียนได้เป็น

$$\frac{\partial(Hu)}{\partial x}+\frac{\partial(Hv)}{\partial y}=0 \qquad (ค.1)$$

$$\rho\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\rho g\frac{\partial \zeta}{\partial x}+\frac{\partial}{\partial x}\left(2\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \qquad (ค.2)$$

$$\rho\left(u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right)=-\rho g\frac{\partial \zeta}{\partial y}+\frac{\partial}{\partial x}\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(2\mu\frac{\partial v}{\partial y}\right) \qquad (ค.3)$$

ซึ่งจะสังเกตได้ว่าสมการ (ค.1)-(ค.3) มีลักษณะคล้ายกับสมการของการไหลแบบหนืดแต่ไม่อัดตัวในสองมิติที่สภาวะอยู่ตัว ซึ่งคือ

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \qquad (ค.4)$$

$$\rho\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(2\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \qquad (ค.5)$$

$$\rho\left(u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}\right)=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(2\mu\frac{\partial v}{\partial y}\right) \qquad (ค.6)$$

จากความคล้ายคลึงกันของสมการของการไหลในสองมิติเฉลี่ยตลอดความลึก (ค.1)-(ค.3) กับสมการในสองมิติแบบปกติ (ค.4)-(ค.6) หากกำหนดให้ค่าความหนาแน่น ρ และค่าความเร่งเนื่องจากแรงโน้มถ่วง g เท่ากับ 1 พร้อมทั้งกำหนดค่าความลึกเฉลี่ยของการไหล h ให้มีค่าคงที่และมากกว่าค่าระดับผิวน้ำ ζ มากๆ ซึ่งมีผลทำให้ค่าความลึกรวม H มีค่าคงที่ จะทำให้สมการ (ค.1)-(ค.3) มี

ลักษณะตรงกับสมการ (ค.4)-(ค.6) ทุกประการ โดยตัวแปรระดับผิวน้ำ ζ ในสมการ (ค.1)-(ค.3) สามารถเปรียบเทียบได้กับความดัน p ในสมการ (ค.4)-(ค.6) ดังนั้นจึงสามารถนำโปรแกรมคอมพิวเตอร์ SWFLOW ที่ใช้สำหรับวิเคราะห์ปัญหาการไหลในน้ำตื้นที่สภาวะอยู่ตัวมาใช้ในการวิเคราะห์ปัญหาการไหลแบบหนืดแต่ไม่อัดตัวในสองมิติที่สภาวะอยู่ตัวได้

ประวัติผู้เขียนวิทยานิพนธ์

นายสมบูรณ์ โอตรวรรณะ เกิดเมื่อวันที่ 20 เดือนพฤศจิกายน พุทธศักราช 2522 จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิตจากภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2543