

รายการอ้างอิง

- Bird, R.B. , Stewart, W.E. , and Lightfoot, E. N. Transport phenomena. Singapore: John Wiley & Sons, 1960.
- Chaddock, J.B. Free convection heat transfer from vertical rectangular fin arrays. ASHRAE Journal 12 (August 1970) : 53-60.
- Elenbass, W. Heat dissipations of parallel plates by free convection . Physica 9 (1942) : 1-28.
- Incropera, F.P. and Dewitt, D.P. Fundamentals of heat and mass Transfer. 3rd ed. Singapore: John Wiley & Sons, 1990.
- Leung, C.W. , and Probert, S.D. Steady-state heat transfer from vertical fins protruding upwards from horizontal bases. Journal of the Institute of Energy (June 1989a) : 94-101.
- Leung, C.W. , and Probert, S.D. Heat-Exchanger Performance : Effect of Orientation. Applied Energy, 33 (1989b) : 235-252.
- Patankar, S.V. Numerical heat transfer and fluid flow. New York : Hemisphere, 1980.
- Schlichting, H. Boundary Layer Theory. 6th ed. New York : McGraw-Hill, 1968.
- Starner, K. E. , and McManus, H. N. An experimental investigation of free convection fin arrays. Trans. ASME . J. Heat Transfer, 85 (1963) : 273-278.
- Welling, J.R. , and Wooldridge, C.B. Free-convection heat-transfer coefficients for rectangular vertical fins. Trans. ASME . J. Heat Transfer, 87 (1965) : 439-444.
- Zahed, S. Where do you put the cooling vents?. Machine Design (April 1994).



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

PROGRAM VERTICAL_FIN

```

C   XMAX = MAXIMUM BOUNDARY IN X-AXIS
C   YMAX = MAXIMUM BOUNDARY IN Y-AXIS
C   ZMAX = MAXIMUM BOUNDARY IN Z-AXIS
C   IMAX = INDEX NUMBER CORRESPONDED TO XMAX
C   JMAX = INDEX NUMBER CORRESPONDED TO YMAX
C   KMAX = INDEX NUMBER CORRESPONDED TO ZMAX
C   YLIM = LIMIT LEVEL FOR FINE GRID
C   JLIM = INDEX NUMBER CORRESPONDED TO YLIM
C   IMIN = INDEX NUMBER CORRESPONDED TO XMIN
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c   PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=150, XMAX=1.,
c +   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=100)
C   IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+   XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+   XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+   XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+   V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+   W(IMIN:IMAX,0:JMAX,-1:KMAX),
+   P(IMIN:IMAX,0:JMAX,0:KMAX),
+   PC(IMIN:IMAX,0:JMAX,0:KMAX),
+   TF(IMIN:0,0:JTIP,0:KMAX),
+   TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
INTEGER CH
REAL L,NOF,MAXST,MAXERR
CHARACTER*1 VME
CHARACTER*8 TIME
COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW
COMMON/VARIABLE/U,V,W,P,PC,TA,TF
COMMON/MS/IMS,JMS,KMS

```

```
COMMON/MAXPOINT/MME,IME,JME,KME
```

```
C
```

```
WRITE(*,*) 'ENTER YOUR CHOICE 1=NEW , 2=CONTINUE'
```

```
READ(*,*) IQ
```

```
IF (IQ.EQ.2) THEN
```

```
  OPEN(4,FILE='UO',STATUS='OLD')
```

```
  OPEN(5,FILE='VO',STATUS='OLD')
```

```
  OPEN(6,FILE='WO',STATUS='OLD')
```

```
  OPEN(7,FILE='ATO',STATUS='OLD')
```

```
  OPEN(8,FILE='FTO',STATUS='OLD')
```

```
  OPEN(9,FILE='DAT',STATUS='OLD')
```

```
  OPEN(10,FILE='PO',STATUS='OLD')
```

```
  OPEN(11,FILE='PCO',STATUS='OLD')
```

```
C ***** READ DATA *****
```

```
  DO 1 K=-1,KMAX+1
```

```
  DO 1 J=0,JMAX
```

```
    READ(4,*) (U(I,J,K),I=IMIN-1,IMAX)
```

```
1  CONTINUE
```

```
  DO 2 K=-1,KMAX+1
```

```
  DO 2 J=-1,JMAX
```

```
    READ(5,*) (V(I,J,K),I=IMIN,IMAX)
```

```
2  CONTINUE
```

```
  DO 3 K=-1,KMAX
```

```
  DO 3 J=0,JMAX
```

```
    READ(6,*) (W(I,J,K),I=IMIN,IMAX)
```

```
3  CONTINUE
```

```
  DO 4 K=-1,KMAX+1
```

```
  DO 4 J=0,JMAX
```

```
    READ(7,*) (TA(I,J,K),I=IMIN,IMAX)
```

```
4  CONTINUE
```

```
  DO 5 K=0,KMAX
```

```
  DO 5 J=0,JTIP
```

```
    READ(8,*) (TF(I,J,K),I=IMIN,0)
```

```

5  CONTINUE
   DO 6 K=0,KMAX
   DO 6 J=0,JMAX
     READ(10,*) (P(I,J,K),I=IMIN,IMAX)
     READ(11,*) (PC(I,J,K),I=IMIN,IMAX)
6  CONTINUE
   READ(9,*) MIT
   READ(9,*) S
   READ(9,*) MAXIT
C
  CLOSE(4)
  CLOSE(5)
  CLOSE(6)
  CLOSE(7)
  CLOSE(8)
  CLOSE(9)
  CLOSE(10)
  CLOSE(11)
ENDIF
OPEN(4,FILE='U')
OPEN(5,FILE='V')
OPEN(6,FILE='W')
OPEN(7,FILE='AT')
OPEN(8,FILE='FT')
OPEN(9,FILE='RES')
OPEN(10,FILE='P')
OPEN(11,FILE='PC')
OPEN(12,FILE='MP')
WRITE(*,*) 'PROGRAM START'
CALL DATA_INPUT (TH,B,L,S,TB,TS,NOF,NOI,MIT,MAXIT,IQ)
CALL PROPERTY_CAL (TB,TS,L,B,S,Gr,Pr)
CALL COOR_SPEC (TH, S)
XX = S/L*Gr*Pr

```

```

WRITE(*,*) 'S/L.Gr.Pr = ',XX
IF (IQ.EQ.1) THEN
  CALL ASSIGN_VALUE ()
  MIT = 0
ENDIF
WRITE(9,7)
7 FORMAT(T5,'MIT.',T10,'MAX.-VEL.ERR',T25,'MAX.-ST.',T40,
+ 'SUM.ERR.-PC',T55,'SUM.ERR.-AT',T70,'SUM.ERR.-FT',T85,
+ 'HEAT-TRANSFER')
WRITE(12,8)
8 FORMAT(T1,'MIT.',T10,'VME',T18,'IME',T26,'JME',T34,'KME',
+ T50,'IMS',T58,'JMS',T66,'KMS')
10 MIT=MIT+1
C
WRITE(*,*) ' '
C WRITE(9,*) ' '
WRITE(*,*) 'MAIN ITERATION NO. ',MIT
IF (MOD(MIT,5).EQ.0) THEN
  WRITE(4,*) 'MAIN ITERATION NO. ',MIT
  WRITE(5,*) 'MAIN ITERATION NO. ',MIT
  WRITE(6,*) 'MAIN ITERATION NO. ',MIT
  WRITE(7,*) 'MAIN ITERATION NO. ',MIT
  WRITE(8,*) 'MAIN ITERATION NO. ',MIT
  WRITE(10,*) 'MAIN ITERATION NO. ',MIT
  WRITE(11,*) 'MAIN ITERATION NO. ',MIT
ENDIF
C WRITE(9,*) 'MAIN ITERATION NO. ',MIT
WRITE(*,*) 'PRESENT TIME : ',TIME()
C WRITE(9,*) 'PRESENT TIME : ',TIME()
c
CALL FLOW_CAL (S, B, L, Gr, MAXIT, MAXERR, MAXST, ERRP)
c
IF (MOD(MIT,5).EQ.0) THEN

```

```

DO 20 J=JMAX,0,-1
  WRITE(4,25) (U(I,J,5),I=IMIN-1,IMAX)
  WRITE(6,30) (W(I,J,5),I=IMIN,IMAX)
20 CONTINUE
  DO 22 J=JMAX,-1,-1
    WRITE(5,30) (V(I,J,5),I=IMIN,IMAX)
22 CONTINUE
  ENDIF
25 FORMAT (17F12.4)
30 FORMAT (16F12.4)
  IF (MOD(MIT,5).EQ.0) THEN
    DO 35 J=JMAX,0,-1
      WRITE(10,40) (P(I,J,5),I=IMIN,IMAX)
      WRITE(11,40) (PC(I,J,5),I=IMIN,IMAX)
35 CONTINUE
    ENDIF
40 FORMAT (16F14.8)
C
  CALL AIRTEMP_CAL (S, B, L, Gr, Pr, MAXIT, ERRT)
C
  IF (MOD(MIT,5).EQ.0) THEN
    DO 50 J=JMAX,0,-1
      WRITE(7,55) (TA(I,J,5),I=IMIN,IMAX)
50 CONTINUE
    ENDIF
55 FORMAT (16F11.5)
C
  CALL FIN_TEMP_CAL (B, S, MAXIT, ERRF)
C
  IF (MOD(MIT,5).EQ.0) THEN
    DO 60 J=JTIP,0,-1
      WRITE(8,65) (TF(I,J,5),I=IMIN,0)
60 CONTINUE

```

```

ENDIF
65 FORMAT (6F11.5)
C
CALL HEAT_CAL (TB,TS,B,L,S,Q,NOF)
C
WRITE(9,70) MIT,MAXERR,MAXST,ERRP,ERRT,ERRF,Q
70 FORMAT(T5,I3,T10,E10.4,T25,E10.4,T40,E10.4,T55,F10.2,T70,E10.4,
+ T85,F8.2)
WRITE(12,72) MIT,VME,IME,JME,KME,IMS,JMS,KMS
72 FORMAT(T1,I3,T10,A1,T18,I3,T26,I3,T34,I3,T50,I3,T58,I3,T66,I3)
c WRITE(*,*) 'ENTER CHOICE : 1 = GO ON , 0 = STOP '
c READ(*,*) CH
c IF (CH.EQ.0) GO TO 85
IF (MIT-NOI) 10,85,85
85 WRITE(*,*) 'ENTER CHOICE : 1 = GO ON , 0 = STOP '
READ(*,*) CH
IF (CH.EQ.0) GO TO 90
IF (CH.EQ.1) THEN
WRITE(*,*) 'NO OF NEXT ITERATION : '
READ(*,*) NNOI
NOI = NOI + NNOI
GO TO 10
ENDIF
90 WRITE(*,*) ' KEEP DATA FOR CONTINUE CAL.: 1=KEEP , 2=NOT KEEP'
READ(*,*) IKE
IF (IKE.EQ.1) THEN
CLOSE(4)
CLOSE(5)
CLOSE(6)
CLOSE(7)
CLOSE(8)
CLOSE(9)
CLOSE(10)

```



```

CLOSE(11)
OPEN(4,FILE='UO')
OPEN(5,FILE='VO')
OPEN(6,FILE='WO')
OPEN(7,FILE='ATO')
OPEN(8,FILE='FTO')
OPEN(9,FILE='DAT')
OPEN(10,FILE='PO')
OPEN(11,FILE='PCO')
C ***** WRITE DATA *****
DO 91 K=-1,KMAX+1
DO 91 J=0,JMAX
WRITE(4,25) (U(I,J,K),I=IMIN-1,IMAX)
91 CONTINUE
DO 92 K=-1,KMAX+1
DO 92 J=-1,JMAX
WRITE(5,30) (V(I,J,K),I=IMIN,IMAX)
92 CONTINUE
DO 93 K=-1,KMAX
DO 93 J=0,JMAX
WRITE(6,30) (W(I,J,K),I=IMIN,IMAX)
93 CONTINUE
DO 94 K=-1,KMAX+1
DO 94 J=0,JMAX
WRITE(7,55) (TA(I,J,K),I=IMIN,IMAX)
94 CONTINUE
DO 95 K=0,KMAX
DO 95 J=0,JTIP
WRITE(8,65) (TF(I,J,K),I=IMIN,0)
95 CONTINUE
DO 96 K=0,KMAX
DO 96 J=0,JMAX
WRITE(10,30) (P(I,J,K),I=IMIN,IMAX)

```



```

98 CONTINUE
   DO 99 K=1,25,2
     WRITE(6,*) 'K=',K
   DO 99 J=0,JMAX
     WRITE(6,30) (W(I,J,K),I=IMIN,IMAX)
99 CONTINUE
   DO 100 K=1,25,2
     WRITE(7,*) 'K=',K
   DO 100 J=0,JMAX
     WRITE(7,55) (TA(I,J,K),I=IMIN,IMAX)
100 CONTINUE
   DO 101 K=5,25,10
     WRITE(8,*) 'K=',K
   DO 101 J=0,JTIP
     WRITE(8,65) (TF(I,J,K),I=IMIN,0)
101 CONTINUE
   DO 102 K=5,25,10
     WRITE(9,*) 'K=',K
   DO 102 J=0,JMAX
     WRITE(9,30) (P(I,J,K),I=IMIN,IMAX)
102 CONTINUE
   ENDIF
   WRITE(*,*) ' PROGRAM STOP'
   STOP
   END
C
C-----
C
SUBROUTINE DATA_INPUT (TH,B,L,S,TB,TS,NOF,NOI,MIT,MAXIT,IQ)
C IMPLICIT REAL*8 (A-H,O-Z)
REAL TH,B,L,S,TB,TS,NOF
TH = 3.
B = 60.

```

```

L = 250.
IF (IQ.EQ.2) THEN
  WRITE(*,*) 'FIN SEPERATION = ',S,'mm.'
  WRITE(*,*) 'LAST MIT NO. = ',MIT
  WRITE(*,*) 'MAX. ITER. FOR FLOW CAL. = ',MAXIT
ENDIF
IF (IQ.EQ.1) THEN
  WRITE(*,*) 'ENTER FIN SEPARATION (mm.): '
  READ(*,*) S
ENDIF
TB = 95.
TS = 15.
C  NOI = 1
C  WRITE(*,*) '  ENTER FOLLOWING DATA '
C  WRITE(*,*) '  FIN THICKNESS (mm.): '
C  READ(*,*) TH
C  WRITE(*,*) '  FIN HEIGTH (mm.): '
C  READ(*,*) B
C  WRITE(*,*) '  FIN LENGTH (mm.): '
C  READ(*,*) L
C  WRITE(*,*) '  FIN SEPARATION (mm.): '
C  READ(*,*) S
C  WRITE(*,*) '  BASE TEMPERATURE (CELCIUS)'
C  READ(*,*) TB
C  WRITE(*,*) '  SURROUNDING TEMPERATURE (CELCIUS)'
C  READ(*,*) TS
  WRITE(*,*) '  NO. OF ITERATIONS '
  READ(*,*) NOI
  WRITE(*,*) '  NO. OF MAX ITERATION FOR FLOW CAL. '
  READ(*,*) MAXIT
  NOF = 2.*(190./(S+TH))
  RETURN
END

```

C

C

C

SUBROUTINE PROPERTY_CAL (TB,TS,L,B,S,Gr,Pr)

C IMPLICIT REAL*8 (A-H,O-Z)

REAL MUE,KA,KF,NUE,L

COMMON/CONDUCT/KA,KF

COMMON/C/CX,CY,CZ

COMMON/CT/CTX,CTY,CTZ

IF ((TB.GE.20.) .AND. (TB.LE.100.)) THEN

 KF = 164. + (TS - 20.) / 80. * (182. - 164.)

ELSEIF ((TB.GT.100.) .AND. (TB.LE.200.)) THEN

 KF = 182. + (TS - 100.) / 100. * (194. - 182.)

ENDIF

TAV = ((TB+273.)+(TS+273.))/2.

IF ((TAV.GE.250.) .AND. (TAV.LT.300.)) THEN

 FAC = (TAV-250.) / 50.

 RO = 1.3947 + FAC * (1.1614 - 1.3947)

 MUE = (159.6 + FAC * (184.6 - 159.6)) * 1E-7

 KA = (22.3 + FAC * (26.3 - 22.3)) * 1E-3

 CP = (1.006 + FAC * (1.007 - 1.006)) * 1E3

ELSEIF ((TAV.GE.300.) .AND. (TAV.LT.350.)) THEN

 FAC = (TAV-300.) / 50.

 RO = 1.1614 + FAC * (0.9950 - 1.1614)

 MUE = (184.6 + FAC * (208.2 - 184.6)) * 1E-7

 KA = (26.3 + FAC * (30.0 - 26.3)) * 1E-3

 CP = (1.007 + FAC * (1.009 - 1.007)) * 1E3

ELSEIF ((TAV.GE.350.) .AND. (TAV.LE.400.)) THEN

 FAC = (TAV-350.) / 50.

 RO = 0.9950 + FAC * (0.8711 - 0.9950)

 MUE = (208.2 + FAC * (230.1 - 208.2)) * 1E-7

 KA = (30.0 + FAC * (33.8 - 30.0)) * 1E-3

 CP = (1.009 + FAC * (1.014 - 1.009)) * 1E3

```

ELSE
  PAUSE 'AVERAGE AIR TEMP. OUT OF RANGE (250 TO 400 K)'
ENDIF

NUE = MUE/RO
BETA = 1./TAV
Pr = MUE*CP/KA
Gr = (9.81*BETA*(TB-TS)*(S*1.E-3)**3)/NUE**2
C1 = L/(B*Gr)
CX = 4.*C1
CY = ((S/B)**2)*C1
CZ = ((S/L)**2)*C1
CT1 = L/(B*Gr*Pr)
CTX = 4.*CT1
CTY = ((S/B)**2)*CT1
CTZ = ((S/L)**2)*CT1
WRITE(*,*) 'SUBROUTINE PROPERTY_CAL PASSED'
RETURN
END

C
C-----
C
SUBROUTINE COOR_SPEC (TH, S)
C  IMPLICIT REAL*8 (A-H,O-Z)
  PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+    IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
  PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=150, XMAX=1.,
+    IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=100)
  PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
+    IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
  DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+    XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+    XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+    XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)

```

COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW

C

C *** MAIN GRID COORDINATE SPECIFICATION ***

DXA = XMAX/IMAX

XMIN = -TH/S

DXF = XMIN/IMIN

DYF = YLIM/JLIM

DYC = (YMAX - YLIM)/(JMAX - JLIM)

DZ = ZMAX/KMAX

XM(IMIN-1) = XMIN - DXF

DO 10 I=IMIN,0

XM(I) = XM(I-1) + DXF

10 CONTINUE

DO 20 I=1,IMAX+1

XM(I) = XM(I-1) + DXA

20 CONTINUE

YM(-1) = -DYF

DO 30 J=0,JLIM

YM(J) = YM(J-1) + DYF

30 CONTINUE

DO 40 J=JLIM+1,JMAX+1

YM(J) = YM(J-1) + DYC

40 CONTINUE

ZM(-1) = -DZ

DO 50 K=0,KMAX+1

ZM(K) = ZM(K-1) + DZ

50 CONTINUE

C

C *** X-VELOCITY GRID COORDINATE SPECIFICATION ***

C

DO 55 I=IMIN-1,-1

XU(I) = XM(I) + (DXF/2.)

55 CONTINUE

```
DO 60 I=0,IMAX
XU(I) = XM(I) + (DXA/2.)
60 CONTINUE
DO 61 J=0,JMAX
YU(J) = YM(J)
61 CONTINUE
DO 62 K=-1,KMAX+1
ZU(K) = ZM(K)
62 CONTINUE
C
C *** Y-VELOCITY GRID COORDINATE SPECIFICATION ***
C
DO 63 I=IMIN,IMAX
XV(I) = XM(I)
63 CONTINUE
DO 64 J=-1,JLIM-1
YV(J) = YM(J) + (DYF/2.)
64 CONTINUE
DO 65 J=JLIM,JMAX
YV(J) = YM(J) + (DYC/2.)
65 CONTINUE
DO 66 K=-1,KMAX+1
ZV(K) = ZM(K)
66 CONTINUE
C
C *** Z-VELOCITY GRID COORDINATE SPECIFICATION ***
C
DO 67 I=IMIN,IMAX
XW(I) = XM(I)
67 CONTINUE
DO 68 J=0,JMAX
YW(J) = YM(J)
68 CONTINUE
```



```

DO 69 K=-1,KMAX
  ZW(K) = ZM(K) + (DZ/2.)
69 CONTINUE
C
  WRITE(*,*) 'SUBROUTINE COORDINATE PASSED'
c  PAUSE 'SHOW COORDINATE AT BOUNDARY'
  WRITE(*,*) 'XM(',IMIN-1,') = ',XM(IMIN-1)
  WRITE(*,*) 'XM(',IMAX+1,') = ',XM(IMAX+1)
  WRITE(*,*) 'YM(-1) = ',YM(-1)
  WRITE(*,*) 'YM(',JMAX+1,') = ',YM(JMAX+1)
  WRITE(*,*) 'ZM(-1) = ',ZM(-1)
  WRITE(*,*) 'ZM(',KMAX+1,') = ',ZM(KMAX+1)
  WRITE(*,*) 'XU(',IMIN-1,') = ',XU(IMIN-1)
  WRITE(*,*) 'XU(',IMAX,') = ',XU(IMAX)
  WRITE(*,*) 'YU(0) = ',YU(0)
  WRITE(*,*) 'YU(',JMAX,') = ',YU(JMAX)
  WRITE(*,*) 'ZU(-1) = ',ZU(-1)
  WRITE(*,*) 'ZU(',KMAX+1,') = ',ZU(KMAX+1)
c  PAUSE ''
  DO 100 I=IMIN,IMAX
    WRITE(*,*) 'XV(',I,') = ',XV(I)
100 CONTINUE
C  PAUSE ''
  WRITE(*,*) 'XV(',IMIN+1,') = ',XV(IMIN+1)
  WRITE(*,*) 'XV(',IMAX,') = ',XV(IMAX)
  WRITE(*,*) 'YV(-1) = ',YV(-1)
  WRITE(*,*) 'YV(',JMAX,') = ',YV(JMAX)
  WRITE(*,*) 'ZV(-1) = ',ZV(-1)
  WRITE(*,*) 'ZV(',KMAX+1,') = ',ZV(KMAX+1)
  WRITE(*,*) 'XW(',IMIN,') = ',XW(IMIN)
  WRITE(*,*) 'XW(',IMAX,') = ',XW(IMAX)
  WRITE(*,*) 'YW(0) = ',YW(0)
  WRITE(*,*) 'YW(',JMAX,') = ',YW(JMAX)

```

```

WRITE(*,*) 'ZW(-1) = ',ZW(-1)
WRITE(*,*) 'ZW(1,KMAX,') = ',ZW(KMAX)

C PAUSE ''
RETURN
END

C
C-----
C
SUBROUTINE ASSIGN_VALUE ()
C IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+ IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+ V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+ W(IMIN:IMAX,0:JMAX,-1:KMAX),
+ P(IMIN:IMAX,0:JMAX,0:KMAX),
+ PC(IMIN:IMAX,0:JMAX,0:KMAX),
+ TF(IMIN:0,0:JTIP,0:KMAX),
+ TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
COMMON/VARIABLE/U,V,W,P,PC,TA,TF
C
C *** SET VELOCITY AT ALL POINT EQUAL TO ZERO ***
C
DO 10 I=IMIN-1,IMAX
DO 10 J=0,JMAX
DO 10 K=-1,KMAX+1
U(I,J,K) = 0.
10 CONTINUE
DO 15 I=IMIN,IMAX

```

```

DO 15 J=-1,JMAX
DO 15 K=-1,KMAX+1
V(I,J,K) = 0.
15 CONTINUE
DO 20 I=IMIN,IMAX
DO 20 J=0,JMAX
DO 20 K=-1,KMAX
W(I,J,K) = 0.
20 CONTINUE
C
C *** SET PRESSURE + PRESS. CORREC. AT ALL POINT EQUAL TO ZERO ***
C
DO 25 I=IMIN,IMAX
DO 25 J=0,JMAX
DO 25 K=0,KMAX
P(I,J,K) = 0.
PC(I,J,K) = 0.
25 CONTINUE
C
C *** SET FIN'S TEMPERATURE AND AIR TEMP. ***
C *** SET TEMP. AT FIN'S BASE ***
C
DO 30 I=1,IMAX
DO 30 K=0,KMAX
TA(I,0,K) = 1.
30 CONTINUE
C
C *** SET AIR TEMP. ABOVE FIN'S SURFACE ***
C
DO 35 I=IMIN,IMAX
DO 35 J=1,JMAX
DO 35 K=-1,KMAX+1
TA(I,J,K) = 0.

```

35 CONTINUE

C

C *** SET INSIDE FIN'S TEMP. + AIR TEMP. AT FIN'S SURFACE ***

C

DO 40 I=IMIN,0

DO 40 J=0,JTIP

DO 40 K=0,KMAX

TF(I,J,K) = 1.

TA(I,J,K) = 1.

40 CONTINUE

C

WRITE(*,*) 'SUBROUTINE ASSIGN_VALUE PASSED'

RETURN

END

C

C _____

C

SUBROUTINE CHECKDIM (A,B,C,D,E,F,G,H,O,I,J,K)

C IMPLICIT REAL*8 (A-H,O-Z)

IF ((A.LT.0.) .OR. (B.LT.0.)) THEN

WRITE(*,*) '!!! DX LESS THAN ZERO '

WRITE(*,*) 'I,J,K = ',I,J,K

WRITE(*,*) 'DXE = ',A

WRITE(*,*) 'DXW = ',B

PAUSE ''

ELSEIF ((C.LT.0.) .OR. (D.LT.0.)) THEN

WRITE(*,*) '!!! DY LESS THAN ZERO '

WRITE(*,*) 'I,J,K = ',I,J,K

PAUSE ''

ELSEIF ((E.LT.0.) .OR. (F.LT.0.)) THEN

WRITE(*,*) '!!! DZ LESS THAN ZERO '

WRITE(*,*) 'I,J,K = ',I,J,K

PAUSE ''

```

ELSEIF ((A.EQ.0.) .OR. (B.EQ.0.) .OR. (C.EQ.0.) .OR. (D.EQ.0.)
+ .OR. (E.EQ.0.) .OR. (F.EQ.0.)) THEN
    WRITE(*,*) '!!! SOME D EQUAL TO ZERO '
    WRITE(*,*) 'I,J,K = ',I,J,K
ENDIF
IF ((G.LT.0.) .OR. (H.LT.0.) .OR. (O.LT.0.)) THEN
    PAUSE '!!! SOME DELX,DELY,DELZ LESS THAN ZERO '
C ELSEIF ((G.EQ.0.) .OR. (H.EQ.0.) .OR. (I.EQ.0.)) THEN
C WRITE(*,*) '!!! SOME DELX,DELY,DELZ EQUAL TO ZERO '
ENDIF
RETURN
END
C
C-----
C
SUBROUTINE CHECKCOEF (A,B,C,D,E,F,G)
C IMPLICIT REAL*8 (A-H,O-Z)
C REAL*8 A,B,C,D,E,F,G
IF ((A.LT.0.) .OR. (B.LT.0.) .OR. (C.LT.0.) .OR. (D.LT.0.)
+ .OR. (E.LT.0.) .OR. (F.LT.0.) .OR. (G.LT.0.)) THEN
    PAUSE '!!! SOME COEFFICIENTS LESS THAN ZERO'
ELSEIF (A.EQ.0.) THEN
    WRITE(*,*) '!!! ALL COEFFICIENTS EQUAL TO ZERO'
ENDIF
RETURN
END
C สถาบันวิทยบริการ
C-----
C จุฬาลงกรณ์มหาวิทยาลัย
C
FUNCTION COEF (D,PEC,F)
C IMPLICIT REAL*8 (A-H,O-Z)
A = (1-0.1*ABS(PEC))**5
IF (A.LE.0.) THEN

```

```

      A = 0.
    ENDIF
    B = F
    IF (B.LE.0.) THEN
      B = 0.
    ENDIF
    COEF = D*A + B
    RETURN
  END
C
C-----
C
      SUBROUTINE COEF_CAL (NO, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+       AREA, VOL, S, B, L, Gr)
C  IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+       IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c  PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
c +       IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c  PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
c +       IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
      DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+       XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+       XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+       XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
      DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+       V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+       W(IMIN:IMAX,0:JMAX,-1:KMAX),
+       P(IMIN:IMAX,0:JMAX,0:KMAX),
+       PC(IMIN:IMAX,0:JMAX,0:KMAX),
+       TF(IMIN:0,0:JTIP,0:KMAX),
+       TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
      COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW

```

COMMON/VARIABLE/U,V,W,P,PC,TA,TF

COMMON/C/CX,CY,CZ

REAL L

IF (NO.EQ.1) THEN

DXE = XU(I+1)-XU(I)

DXW = XU(I)-XU(I-1)

DYN = YU(J+1)-YU(J)

DYS = YU(J)-YU(J-1)

DZT = ZU(K+1)-ZU(K)

DZB = ZU(K)-ZU(K-1)

DELX = XM(I+1) - XM(I)

DELY = (YM(J+1) - YM(J-1))/2.

DELZ = (ZM(K+1) - ZM(K-1))/2.

FE = (U(I+1,J,K)+U(I,J,K))/2.*DELY*DELZ

FW = (U(I-1,J,K)+U(I,J,K))/2.*DELY*DELZ

FN = (V(I+1,J,K)+V(I,J,K))/2.*DELZ*DELX

FS = (V(I+1,J-1,K)+V(I,J-1,K))/2.*DELZ*DELX

FT = (W(I+1,J,K)+W(I,J,K))/2.*DELX*DELY

FB = (W(I+1,J,K-1)+W(I,J,K-1))/2.*DELX*DELY

AREA = DELY*DELZ

IF ((I.EQ.0) .AND. (J.LE.JTIP)) THEN

FW = 0.

ENDIF

IF (I.EQ.IMIN) THEN

FW = 0.

ENDIF

IF (I.EQ.IMAX-1) THEN

FE = 0.

ENDIF

ELSEIF (NO.EQ.2) THEN

DXE = XV(I+1)-XV(I)

DXW = XV(I)-XV(I-1)

DYN = YV(J+1)-YV(J)

```

DYS = YV(J)-YV(J-1)
DZT = ZV(K+1)-ZV(K)
DZB = ZV(K)-ZV(K-1)
DELX = (XM(I+1) - XM(I-1))/2.
DELY = YM(J+1) - YM(J)
DELZ = (ZM(K+1) - ZM(K-1))/2.

FE = (U(I,J,K)+U(I,J+1,K))/2.*DELY*DELZ
FW = (U(I-1,J,K)+U(I-1,J+1,K))/2.*DELY*DELZ
FN = (V(I,J+1,K)+V(I,J,K))/2.*DELZ*DELX
FS = (V(I,J-1,K)+V(I,J,K))/2.*DELZ*DELX
FT = (W(I,J,K)+W(I,J+1,K))/2.*DELX*DELY
FB = (W(I,J,K-1)+W(I,J+1,K-1))/2.*DELX*DELY

AREA = DELZ*DELX

IF ((J.EQ.JTIP) .AND. (I.LE.0)) THEN
  FS = 0.
ENDIF

IF ((J.EQ.0) .AND. (I.GT.0)) THEN
  FS = 0.
ENDIF

ELSEIF (NO.EQ.3) THEN
DXE = XW(I+1)-XW(I)
DXW = XW(I)-XW(I-1)
DYN = YW(J+1)-YW(J)
DYS = YW(J)-YW(J-1)
DZT = ZW(K+1)-ZW(K)
DZB = ZW(K)-ZW(K-1)
DELX = (XM(I+1) - XM(I-1))/2.
DELY = (YM(J+1) - YM(J-1))/2.
DELZ = ZM(K+1) - ZM(K)

FE = (U(I,J,K)+U(I,J,K+1))/2.*DELY*DELZ
FW = (U(I-1,J,K)+U(I-1,J,K+1))/2.*DELY*DELZ
FN = (V(I,J,K)+V(I,J,K+1))/2.*DELZ*DELX
FS = (V(I,J-1,K)+V(I,J-1,K+1))/2.*DELZ*DELX

```



```

      FT = (W(I,J,K+1)+W(I,J,K))/2.*DELX*DELY
      FB = (W(I,J,K-1)+W(I,J,K))/2.*DELX*DELY
      AREA = DELX*DELY
    ENDIF
    VOL = DELX*DELY*DELZ
    CALL CHECKDIM (DXE,DXW,DYN,DYS,DZT,DZB,DELX,DELY,DELZ,I,J,K)
C   PAUSE 'DIMENSIONS CAL. PASSED'
    DE = CX*DELY*DELZ/DXE
    DW = CX*DELY*DELZ/DXW
    DN = CY*DELZ*DELX/DYN
    DS = CY*DELZ*DELX/DYS
    DT = CZ*DELX*DELY/DZT
    DB = CZ*DELX*DELY/DZB
    PE = FE/DE
    PW = FW/DW
    PN = FN/DN
    PS = FS/DS
    PT = FT/DT
    PB = FB/DB
    AE = COEF (DE,PE,-FE)
    AW = COEF (DW,PW,FW)
    AN = COEF (DN,PN,-FN)
    AS = COEF (DS,PS,FS)
    AT = COEF (DT,PT,-FT)
    AB = COEF (DB,PB,FB)
    AP = AE + AW + AN + AS + AT + AB
    CALL CHECKCOEF (AP,AE,AW,AN,AS,AT,AB)
    IF (AP.EQ.0.) THEN
    IF (NO.EQ.1) THEN
    WRITE(*,*) 'ERROR ON U_CAL.'
    ELSEIF (NO.EQ.2) THEN
    WRITE(*,*) 'ERROR ON V_CAL.'
    ELSEIF (NO.EQ.3) THEN

```

```

WRITE(*,*) 'ERROR ON W_CAL.'
ENDIF
WRITE(*,*) 'I,J,K = ',I,J,K
WRITE(*,*) 'FE , DE = ',FE,' ',DE
WRITE(*,*) 'FW , DW = ',FW,' ',DW
WRITE(*,*) 'FN , DN = ',FN,' ',DN
WRITE(*,*) 'FS , DS = ',FS,' ',DS
WRITE(*,*) 'FT , DT = ',FT,' ',DT
WRITE(*,*) 'FB , DB = ',FB,' ',DB
WRITE(*,*) 'AE = ',AE
WRITE(*,*) 'AW = ',AW
WRITE(*,*) 'AN = ',AN
WRITE(*,*) 'AS = ',AS
WRITE(*,*) 'AT = ',AT
WRITE(*,*) 'AB = ',AB
WRITE(*,*) 'AP = ',AP
PAUSE 'PRESS ANY KEY'
WRITE(*,*) 'U = ',U(I-1,J,K),U(I,J,K),U(I+1,J,K)
WRITE(*,*) 'V = ',V(I,J-1,K),V(I,J,K),V(I,J+1,K)
WRITE(*,*) 'W = ',W(I,J,K-1),W(I,J,K),W(I,J,K+1)
ENDIF
RETURN
END

```

C

C

C

```

SUBROUTINE FLOW_CAL (S, B, L, Gr, MAXIT, MAXERR, MAXST, ERRP)

```

C IMPLICIT REAL*8 (A-H,O-Z)

```

REAL MAXST,ERRMAX,ERR,L,MAXERR

```

```

CHARACTER*1 VME

```

```

PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,

```

```

+   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)

```

C PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,

```

C +   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
C   PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
C +   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
      DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+     XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+     XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+     XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
      DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+     V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+     W(IMIN:IMAX,0:JMAX,-1:KMAX),
+     UNEW(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+     VNEW(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+     WNEW(IMIN:IMAX,0:JMAX,-1:KMAX),
+     P(IMIN:IMAX,0:JMAX,0:KMAX),
+     PC(IMIN:IMAX,0:JMAX,0:KMAX),
+     TF(IMIN:0,0:JTIP,0:KMAX),
+     TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
      COMMON/COORDINATE/XM, YM, ZM, XU, YU, ZU, XV, YV, ZV, XW, YW, ZW
      COMMON/VARIABLE/U, V, W, P, PC, TA, TF
      COMMON/NEW/UNEW, VNEW, WNEW
      COMMON/MAXPOINT/ME, IME, JME, KME
C
      DO 1 I=IMIN-1,IMAX
      DO 1 J=0,JMAX
      DO 1 K=-1,KMAX+1
      UNEW(I,J,K) = U(I,J,K)
1  CONTINUE
      DO 2 I=IMIN,IMAX
      DO 2 J=-1,JMAX
      DO 2 K=-1,KMAX+1
      VNEW(I,J,K) = V(I,J,K)
2  CONTINUE
      DO 3 I=IMIN,IMAX

```

```

DO 3 J=0,JMAX
DO 3 K=-1,KMAX
WNEW(I,J,K) = W(I,J,K)
3 CONTINUE
ERRM = 0.
C
UR = 0.2
ITMAX = MAXIT
ITER = 0
ERRMAX = 0.0000005
5 ITER = ITER + 1
MAXERR = 0.
C
C *** X-VELOCITY (U) CALCULATION ***
C
DO 10 I=0,IMAX-1
DO 10 J=1,JMAX-1
DO 10 K=0,KMAX
CALL COEF_CAL (1, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
PRV = U(I,J,K)
UNEW(I,J,K) = UNEW(I,J,K) + UR*((1./AP)*(AE*UNEW(I+1,J,K) +
+ AW*UNEW(I-1,J,K) + AN*UNEW(I,J+1,K) +
+ AS*UNEW(I,J-1,K) + AT*UNEW(I,J,K+1) +
+ AB*UNEW(I,J,K-1) + (P(I,J,K) - P(I+1,J,K))
+ *AREA*(2.*L/S)**2) - UNEW(I,J,K))
ERR = ABS(UNEW(I,J,K) - PRV)
IF (ERR.GT.MAXERR) THEN
MAXERR = ERR
VME = 'U'
IME = I
JME = J
KME = K

```

```

ENDIF
10 CONTINUE
DO 15 I=IMIN,-1
DO 15 J=JTIP+1,JMAX-1
DO 15 K=0,KMAX
CALL COEF_CAL (1, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
PRV = U(I,J,K)
UNEW(I,J,K) = UNEW(I,J,K) + UR*((1./AP)*(AE*UNEW(I+1,J,K) +
+ AW*UNEW(I-1,J,K) + AN*UNEW(I,J+1,K) +
+ AS*UNEW(I,J-1,K) + AT*UNEW(I,J,K+1) +
+ AB*UNEW(I,J,K-1) + (P(I,J,K) - P(I+1,J,K))
+ *AREA*(2.*L/S)**2) - UNEW(I,J,K))
ERR = ABS(UNEW(I,J,K) - PRV)
IF (ERR.GT.MAXERR) THEN
MAXERR = ERR
VME = 'U'
IME = I
JME = J
KME = K
ENDIF
15 CONTINUE
C PAUSE 'X-VELOCITY CAL. PASSED'
C
C *** Y-VELOCITY (V) CALCULATION ***
C
DO 20 I=1,IMAX-1
DO 20 J=0,JMAX-1
DO 20 K=0,KMAX
CALL COEF_CAL (2, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
PRV = V(I,J,K)
VNEW(I,J,K) = VNEW(I,J,K) + UR*((1./AP)*(AE*VNEW(I+1,J,K) +

```

```

+      AW*VNEW(I-1,J,K) + AN*VNEW(I,J+1,K) +
+      AS*VNEW(I,J-1,K) + AT*VNEW(I,J,K+1) +
+      AB*VNEW(I,J,K-1) + (P(I,J,K) - P(I,J+1,K))
+      *AREA*(L/B)**2) - VNEW(I,J,K)
ERR = ABS(VNEW(I,J,K) - PRV)
IF (ERR.GT.MAXERR) THEN
  MAXERR = ERR
  VME = V
  IME = I
  JME = J
  KME = K
ENDIF
20 CONTINUE
  DO 25 I=IMIN+1,0
    DO 20 J=JMIN,JMAX-1
      DO 20 K=KMIN,KMAX
        CALL COEF_CAL (2, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+      AREA, VOL, S, B, L, Gr)
        PRV = V(I,J,K)
        VNEW(I,J,K) = VNEW(I,J,K) + UR*((1./AP)*(AE*VNEW(I+1,J,K) +
+      AW*VNEW(I-1,J,K) + AN*VNEW(I,J+1,K) +
+      AS*VNEW(I,J-1,K) + AT*VNEW(I,J,K+1) +
+      AB*VNEW(I,J,K-1) + (P(I,J,K) - P(I,J+1,K))
+      *AREA*(L/B)**2) - VNEW(I,J,K))
ERR = ABS(VNEW(I,J,K) - PRV)
IF (ERR.GT.MAXERR) THEN
  MAXERR = ERR
  VME = V
  IME = I
  JME = J
  KME = K
ENDIF
25 CONTINUE

```

```

C   PAUSE 'Y-VELOCITY CAL. PASSED'
C
C   *** Z-VELOCITY (W) CALCULATION ***
C
DO 30 I=1,IMAX-1
DO 30 J=1,JMAX-1
DO 30 K=0,KMAX-1
CALL COEF_CAL (3, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+   AREA, VOL, S, B, L, Gr)
PRV = W(I,J,K)
WNEW(I,J,K) = WNEW(I,J,K) + UR*((1./AP)*(AE*WNEW(I+1,J,K) +
+   AW*WNEW(I-1,J,K) + AN*WNEW(I,J+1,K) +
+   AS*WNEW(I,J-1,K) + AT*WNEW(I,J,K+1) +
-   AB*WNEW(I,J,K-1) + (P(I,J,K) - P(I,J,K+1))
+   *AREA + (L/(B*Gr))*(S/B)*
-   TA(I,J,K)+TA(I,J,K+1))/2.*VOL) - WNEW(I,J,K)
ERR = ABS(WNEW(I,J,K) - PRV)
IF (ERR.GT.MAXERR) THEN
MAXERR = ERR
VME = 'W'
IME = I
JME = J
KME = K
ENDIF
30 CONTINUE
DO 35 I=IMIN+1,0
DO 35 J=JTIP+1,JMAX-1
DO 35 K=0,KMAX-1
CALL COEF_CAL (3, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+   AREA, VOL, S, B, L, Gr)
PRV = W(I,J,K)
WNEW(I,J,K) = WNEW(I,J,K) + UR*((1./AP)*(AE*WNEW(I+1,J,K) +
+   AW*WNEW(I-1,J,K) + AN*WNEW(I,J+1,K) +

```

```

+      AS*WNEW(I,J-1,K) + AT*WNEW(I,J,K+1) +
+      AB*WNEW(I,J,K-1) + (P(I,J,K) - P(I,J,K+1))
+      *AREA + (L(B*Gr))*(S/B)*
+      (TA(I,J,K)+TA(I,J,K+1))/2.*VOL) - WNEW(I,J,K)
ERR = ABS(WNEW(I,J,K) - PRV)
IF (ERR.GT.MAXERR) THEN
  MAXERR = ERR
  VME = 'W'
  IME = I
  JME = J
  KME = K
ENDIF
35 CONTINUE
C  PAUSE 'Z-VELOCITY CAL. PASSED'
C  *** CAL. VELOCITY AT BOUNDARY ***
CALL VELO_AT_BOUND (1)
C
LL = MOD(ITER,5)
IF ((LL.EQ.0).OR.(MAXERR.LE.ERRMAX).OR.(ITER.EQ.ITMAX)) THEN
  DO 36 I=IMIN-1,IMAX
  DO 36 J=0,JMAX
  DO 36 K=-1,KMAX+1
    U(I,J,K) = UNEW(I,J,K)
36 CONTINUE
  DO 37 I=IMIN,IMAX
  DO 37 J=-1,JMAX
  DO 37 K=-1,KMAX+1
    V(I,J,K) = VNEW(I,J,K)
37 CONTINUE
  DO 38 I=IMIN,IMAX
  DO 38 J=0,JMAX
  DO 38 K=-1,KMAX
    W(I,J,K) = WNEW(I,J,K)

```



```

38 CONTINUE
   ENDIF
   IF (MAXERR.LE.ERRMAX) GO TO 39
   IF (ITER - ITMAX) 5,39,39
39 WRITE(*,*) 'VELOCITY CAL. PASSED'
   WRITE(*,*) 'NO OF ITER. = ',ITER
   WRITE(*,*) 'MAX. ERROR = ',MAXERR
C
C WRITE(9,*) 'VELOCITY CAL. PASSED'
C WRITE(9,*) 'NO OF ITER. = ',ITER
C WRITE(9,*) 'MAX. ERROR = ',MAXERR
C
C *** PRESSURE CORRECTION (PC) CALCULATION ***
   CALL PRESS_CORR_CAL (MAXST, ERRP, S, B, L, Gr, MAXIT)
C
C *** NEW PRESSURE ***
   DO 70 I=1,IMAX-1
   DO 70 J=1,JTIP
   DO 70 K=1,KMAX-1
70 P(I,J,K) = P(I,J,K) + 0.2*PC(I,J,K)
   DO 75 I=IMIN+1,IMAX-1
   DO 75 J=JTIP+1,JMAX-1
   DO 75 K=1,KMAX-1
75 P(I,J,K) = P(I,J,K) + 0.2*PC(I,J,K)
C *** NEW VELOCITY ***
   DO 80 I=0,IMAX-1
   DO 80 J=1,JMAX-1
   DO 80 K=0,KMAX
   CALL COEF_CAL (1, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+     AREA, VOL, S, B, L, Gr)
80 U(I,J,K) = U(I,J,K) + (AREA*(2.*L/S)**2/AP
+     (PC(I,J,K) - PC(I+1,J,K)))
   DO 85 I=IMIN,-1

```

```

DO 85 J=JTIP+1,JMAX-1
DO 85 K=0,KMAX
CALL COEF_CAL (1, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
85 U(I,J,K) = U(I,J,K) + (AREA*(2.*L/S)**2/AP*
+ (PC(I,J,K) - PC(I+1,J,K))
DO 90 I=1,IMAX-1
DO 90 J=0,JMAX-1
DO 90 K=0,KMAX
CALL COEF_CAL (2, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
90 V(I,J,K) = V(I,J,K) + (AREA*(L/B)**2/AP*
+ (PC(I,J,K) - PC(I,J+1,K))
DO 95 I=IMIN+1,0
DO 95 J=JTIP,JMAX-1
DO 95 K=0,KMAX
CALL COEF_CAL (2, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
95 V(I,J,K) = V(I,J,K) + (AREA*(L/B)**2/AP*
+ (PC(I,J,K) - PC(I,J+1,K))
DO 100 I=1,IMAX-1
DO 100 J=1,JMAX-1
DO 100 K=0,KMAX-1
CALL COEF_CAL (3, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
100 W(I,J,K) = W(I,J,K) + (AREA/AP)*(PC(I,J,K) - PC(I,J,K+1))
DO 105 I=IMIN+1,0
DO 105 J=JTIP+1,JMAX-1
DO 105 K=0,KMAX-1
CALL COEF_CAL (3, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
105 W(I,J,K) = W(I,J,K) + (AREA/AP)*(PC(I,J,K) - PC(I,J,K+1))

```

C

```

C *** CAL. NEW VELOCITY AT BOUNDARY AND INFLOW,OUTFLOW ***
CALL VELO_AT_BOUND (2)
WRITE(*,*) 'FLOW_CAL PASSED'
c WRITE(9,*) 'FLOW_CAL PASSED'
RETURN
END

C
C-----
C
SUBROUTINE PRESS_CORR_CAL (MAXST, ERRP, S, B, L, Gr, MAXIT)
C IMPLICIT REAL*8 (A-H,O-Z)
REAL L,MAXST,ERRP,ERRMAX
COMMON/MS/MS,JMS,KMS
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+ IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+ XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+ XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+ XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+ V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+ W(IMIN:IMAX,0:JMAX,-1:KMAX),
+ P(IMIN:IMAX,0:JMAX,0:KMAX),
+ PC(IMIN:IMAX,0:JMAX,0:KMAX),
+ TF(IMIN:0,0:JTIP,0:KMAX),
+ TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW
COMMON/VARIABLE/U,V,W,P,PC,TA,TF

```

C

```

ITMAX = MAXIT
ITER = 0
ERRMAX = 1.0E-12
10 ITER = ITER + 1
ERRP = 0.
MAXST = 0.
DO 15 I=1,IMAX-1
DO 15 J=1,JTIP
DO 15 K=1,KMAX-1
DELX = (XM(I+1) - XM(I-1))/2.
DELY = (YM(J+1) - YM(J-1))/2.
DELZ = (ZM(K+1) - ZM(K-1))/2.
ST = (U(I-1,J,K) - U(I,J,K))*DELY*DELZ +
+ (V(I,J-1,K) - V(I,J,K))*DELZ*DELX +
+ (W(I,J,K-1) - W(I,J,K))*DELX*DELY
C
IF (ABS(ST).GT.MAXST) THEN
MAXST = ABS(ST)
IMS = I
JMS = J
KMS = K
ENDIF
CALL COEF_CAL ( 1, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
AEP = AREA*(2.*L/S)**2/AP*DELY*DELZ
CALL COEF_CAL ( 1, I-1, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
AWP = AREA*(2.*L/S)**2/AP*DELY*DELZ
CALL COEF_CAL ( 2, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)
ANP = AREA*(L/B)**2/AP*DELZ*DELX
CALL COEF_CAL ( 2, I, J-1, K, AP, AE, AW, AN, AS, AT, AB,
+ AREA, VOL, S, B, L, Gr)

```

```

ASP = AREA*(L/B)**2/AP*DELZ*DELX
CALL COEF_CAL ( 3, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+   AREA, VOL, S, B, L, Gr)
ATP = AREA/AP*DELX*DELY
CALL COEF_CAL ( 3, I, J, K-1, AP, AE, AW, AN, AS, AT, AB,
+   AREA, VOL, S, B, L, Gr)
ABP = AREA/AP*DELX*DELY
APP = AEP + AWP + ANP + ASP + ATP + ABP
PCPRV = PC(I,J,K)
PC(I,J,K) = (1./APP)*(AEP*PC(I+1,J,K) + AWP*PC(I-1,J,K) +
+   ANP*PC(I,J+1,K) + ASP*PC(I,J-1,K) +
+   ATP*PC(I,J,K+1) + ABP*PC(I,J,K-1) +
+   ST)
ERRP = ERRP + ABS(PCPRV - PC(I,J,K))
15 CONTINUE
DO 16 I=IMIN+1,IMAX-1
DO 16 J=JTIP+1,JMAX-1
DO 16 K=1,KMAX-1
DELX = (XM(I+1) - XM(I-1))/2.
DELY = (YM(J+1) - YM(J-1))/2.
DELZ = (ZM(K+1) - ZM(K-1))/2.
ST = (U(I-1,J,K) - U(I,J,K))*DELY*DELZ +
+   (V(I,J-1,K) - V(I,J,K))*DELZ*DELX +
+   (W(I,J,K-1) - W(I,J,K))*DELX*DELY
c
IF (ABS(ST).GT.MAXST) THEN
MAXST = ABS(ST)
IMS = I
JMS = J
KMS = K
ENDIF
CALL COEF_CAL ( 1, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+   AREA, VOL, S, B, L, Gr)

```

```

AEP = AREA*(2.*L/S)**2/AP*DELY*DELZ
CALL COEF_CAL ( 1, I-1, J, K, AP, AE, AW, AN, AS, AT, AB,
+           AREA, VOL, S, B, L, Gr)
AWP = AREA*(2.*L/S)**2/AP*DELY*DELZ
CALL COEF_CAL ( 2, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+           AREA, VOL, S, B, L, Gr)
ANP = AREA*(L/B)**2/AP*DELZ*DELX
CALL COEF_CAL ( 2, I, J-1, K, AP, AE, AW, AN, AS, AT, AB,
+           AREA, VOL, S, B, L, Gr)
ASP = AREA*(L/B)**2/AP*DELZ*DELX
CALL COEF_CAL ( 3, I, J, K, AP, AE, AW, AN, AS, AT, AB,
+           AREA, VOL, S, B, L, Gr)
ATP = AREA/AP*DELX*DELY
CALL COEF_CAL ( 3, I, J, K-1, AP, AE, AW, AN, AS, AT, AB,
+           AREA, VOL, S, B, L, Gr)
ABP = AREA/AP*DELX*DELY
APP = AEP + AWP + ANP + ASP + ATP + ABP
PCPRV = PC(I,J,K)
PC(I,J,K) = (1./APP)*(AEP*PC(I+1,J,K) + AWP*PC(I-1,J,K) +
+           ANP*PC(I,J+1,K) + ASP*PC(I,J-1,K) +
+           ATP*PC(I,J,K+1) + ABP*PC(I,J,K-1) +
+           ST)
C   IF (PC(I,J,K).NE.0.) THEN
C   WRITE(*,*) 'I,J,K = ',I,J,K
C   WRITE(*,*) ' '
C   PAUSE ' 2NEW PC NE. 0 '
C   ENDIF
ERRP = ERRP + ABS(PCPRV - PC(I,J,K))
16 CONTINUE
IF (ERRP.LE.ERRMAX) GO TO 20
IF (ITER-ITMAX) 10,20,20
20 WRITE(*,*) '***PRESS. CORRECT. ITERATION SUCCESS ****'
WRITE(*,*) 'NO. OF ITERATION OF PRESS. CORRECTION = ',ITER

```

```
WRITE(*,*) 'SUM OF ERROR = ',ERRP
```

```
WRITE(*,*) 'MAX SOURCE TERM = ',MAXST
```

```
C
```

```
C WRITE(0,*) '***PRESS. CORRECT. ITERATION SUCCESS ***'
```

```
C WRITE(0,*) 'NO. OF ITERATION OF PRESS. CORRECTION = ',ITER
```

```
C WRITE(0,*) 'SUM OF ERROR = ',ERROR
```

```
C WRITE(0,*) 'MAX SOURCE TERM = ',MAXST
```

```
RETURN
```

```
END
```

```
C
```

```
C-----
```

```
C
```

```
  SUBROUTINE VELO_AT_BOUND (IN)
```

```
C  IMPLICIT REAL*8 (A-H,O-Z)
```

```
  PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
```

```
+   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
```

```
C  PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
```

```
C +   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
```

```
C  PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
```

```
C +   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
```

```
  DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
```

```
+   V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
```

```
+   W(IMIN:IMAX,0:JMAX,-1:KMAX),
```

```
+   UNEW(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
```

```
+   VNEW(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
```

```
+   WNEW(IMIN:IMAX,0:JMAX,-1:KMAX),
```

```
+   P(IMIN:IMAX,0:JMAX,0:KMAX),
```

```
+   PC(IMIN:IMAX,0:JMAX,0:KMAX),
```

```
+   TF(IMIN:0,0:JTIP,0:KMAX),
```

```
+   TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
```

```
  INTEGER IN
```

```
  COMMON/VARIABLE/U,V,W,P,PC,TA,TF
```

```
  COMMON/NEW/UNEW,VNEW,WNEW
```

```

C
  IF (IN.EQ.1) THEN
C
c   *** U(IMIN-1) = U(IMAX) = 0 ***
C
c   DO 10 J=1,JMAX-1
c   DO 10 K=0,KMAX
c 10 UNEW(IMAX,J,K) = UNEW(IMAX-1,J,K)
c   DO 15 J=JTIP+1,JMAX-1
c   DO 15 K=0,KMAX
c 15 UNEW(IMIN-1,J,K) = UNEW(IMIN,J,K)
C
  DO 20 I=1,IMAX
  DO 20 J=1,JMAX-1
 20 UNEW(I,J,KMAX+1) = UNEW(I,J,KMAX)
  DO 25 I=IMIN-1,0
  DO 25 J=JTIP+1,JMAX-1
 25 UNEW(I,J,KMAX+1) = UNEW(I,J,KMAX)
  DO 30 J=0,JMAX-1
  DO 30 K=0,KMAX
 30 VNEW(IMAX,J,K) = (4.*VNEW(IMAX-1,J,K) - VNEW(IMAX-2,J,K))/3.
  DO 35 J=JTIP,JMAX-1
  DO 35 K=0,KMAX
 35 VNEW(IMIN,J,K) = (4.*VNEW(IMIN+1,J,K) - VNEW(IMIN+2,J,K))/3.
  DO 40 I=1,IMAX
  DO 40 J=0,JMAX-1
 40 VNEW(I,J,KMAX+1) = VNEW(I,J,KMAX)
  DO 45 I=IMIN,0
  DO 45 J=JTIP,JMAX-1
 45 VNEW(I,J,KMAX+1) = VNEW(I,J,KMAX)
  DO 50 J=1,JMAX-1
  DO 50 K=0,KMAX-1
 50 WNEW(IMAX,J,K) = (4.*WNEW(IMAX-1,J,K) - WNEW(IMAX-2,J,K))/3.

```



```

DO 55 J=JTIP+1,JMAX-1
DO 55 K=0,KMAX-1
55 WNEW(IMIN,J,K) = (4.*WNEW(IMIN+1,J,K) - WNEW(IMIN+2,J,K))/3.
DO 60 I=1,IMAX
DO 60 J=1,JMAX-1
WNEW(I,J,KMAX) = WNEW(I,J,KMAX-1)
WNEW(I,J,-1) = WNEW(I,J,0)
60 CONTINUE
DO 65 I=IMIN,0
DO 65 J=JTIP+1,JMAX-1
WNEW(I,J,KMAX) = WNEW(I,J,KMAX-1)
WNEW(I,J,-1) = WNEW(I,J,0)
65 CONTINUE
C
ELSEIF (IN.EQ.2) THEN
C
C *** U(IMIN-1) = U(IMAX) = 0 ***
C
C DO 70 J=1,JMAX-1
C DO 70 K=0,KMAX
C 70 U(IMAX,J,K) = U(IMAX-1,J,K)
C DO 75 J=JTIP+1,JMAX-1
C DO 75 K=0,KMAX
C 75 U(IMIN-1,J,K) = U(IMIN,J,K)
C
C
DO 80 I=1,IMAX
DO 80 J=1,JMAX-1
80 U(I,J,KMAX+1) = U(I,J,KMAX)
DO 85 I=IMIN-1,0
DO 85 J=JTIP+1,JMAX-1
85 U(I,J,KMAX+1) = U(I,J,KMAX)
DO 90 J=0,JMAX-1
DO 90 K=0,KMAX

```

```

90 V(IMAX,J,K) = (4.*V(IMAX-1,J,K) - V(IMAX-2,J,K))/3.
   DO 95 J=JTIP,JMAX-1
   DO 95 K=0,KMAX
95 V(IMIN,J,K) = (4.*V(IMIN+1,J,K) - V(IMIN+2,J,K))/3.
   DO 100 I=1,IMAX
   DO 100 J=0,JMAX-1
100 V(I,J,KMAX+1) = V(I,J,KMAX)
   DO 105 I=IMIN,0
   DO 105 J=JTIP,JMAX-1
105 V(I,J,KMAX+1) = V(I,J,KMAX)
   DO 110 J=1,JMAX-1
   DO 110 K=0,KMAX-1
110 W(IMAX,J,K) = (4.*W(IMAX-1,J,K) - W(IMAX-2,J,K))/3.
   DO 115 J=JTIP+1,JMAX-1
   DO 115 K=0,KMAX-1
115 W(IMIN,J,K) = (4.*W(IMIN+1,J,K) - W(IMIN+2,J,K))/3.
   DO 120 I=1,IMAX
   DO 120 J=1,JMAX-1
   W(I,J,KMAX) = W(I,J,KMAX-1)
   W(I,J,-1) = W(I,J,0)
120 CONTINUE
   DO 125 I=IMIN,0
   DO 125 J=JTIP+1,JMAX-1
   W(I,J,KMAX) = W(I,J,KMAX-1)
   W(I,J,-1) = W(I,J,0)
125 CONTINUE
   ENDIF
   RETURN
   END

```

C

C

C

SUBROUTINE AIRTEMP_CAL (S, B, L, Gr, Pr, MAXIT, ERRT)

C IMPLICIT REAL*8 (A-H,O-Z)

REAL L,ERR,ERRMAX

PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,

+ IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)

C PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,

C + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)

C PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,

C + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)

DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),

+ XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),

+ XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),

+ XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)

DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),

+ V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),

+ W(IMIN:IMAX,0:JMAX,-1:KMAX),

+ P(IMIN:IMAX,0:JMAX,0:KMAX),

+ PC(IMIN:IMAX,0:JMAX,0:KMAX),

+ TF(IMIN:0,0:JTIP,0:KMAX),

+ TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)

COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW

COMMON/VARIABLE/U,V,W,P,PC,TA,TF

COMMON/CT/CTX,CTY,CTZ

C

URT = 0.7

ITER = 0

ITMAX = MAXIT

ERRMAX = 0.025

5 ITER = ITER + 1

ERRT = 0.

DO 10 I=1,IMAX-1

DO 10 J=1,JTIP

DO 10 K=0,KMAX

DXE = XM(I+1) - XM(I)

$$DXW = XM(I) - XM(I-1)$$

$$DYN = YM(J+1) - YM(J)$$

$$DYS = YM(J) - YM(J-1)$$

$$DZT = ZM(K+1) - ZM(K)$$

$$DZB = ZM(K) - ZM(K-1)$$

$$DELX = (XM(I+1) - XM(I-1))/2.$$

$$DELY = (YM(J+1) - YM(J-1))/2.$$

$$DELZ = (ZM(K+1) - ZM(K-1))/2.$$

$$FE = U(I,J,K)*DELY*DELZ$$

$$FW = U(I-1,J,K)*DELY*DELZ$$

$$FN = V(I,J,K)*DELX*DELZ$$

$$FS = V(I,J-1,K)*DELX*DELZ$$

$$FT = W(I,J,K)*DELX*DELY$$

$$FB = W(I,J,K-1)*DELX*DELY$$

$$DE = CTX*DELY*DELZ/DXE$$

$$DW = CTX*DELY*DELZ/DXW$$

$$DN = CTY*DELZ*DELX/DYN$$

$$DS = CTY*DELZ*DELX/DYS$$

$$DT = CTZ*DELX*DELY/DZT$$

$$DB = CTZ*DELX*DELY/DZB$$

$$PE = FE/DE$$

$$PW = FW/DW$$

$$PN = FN/DN$$

$$PS = FS/DS$$

$$PT = FT/DT$$

$$PB = FB/DB$$

$$AET = \text{COEF}(DE, PE, -FE)$$

$$AWT = \text{COEF}(DW, PW, FW)$$

$$ANT = \text{COEF}(DN, PN, -FN)$$

$$AST = \text{COEF}(DS, PS, FS)$$

$$ATT = \text{COEF}(DT, PT, -FT)$$

$$ABT = \text{COEF}(DB, PB, FB)$$

$$APT = AET + AWT + ANT + AST + ATT + ABT$$

```

CALL CHECKCOEF(APT,AET,AWT,ANT,AST,ATT,ABT)
PRV = TA(I,J,K)
TA(I,J,K) = TA(I,J,K) + URT*((1./APT)*(AET*TA(I+1,J,K)
+
      + AWT*TA(I-1,J,K) + ANT*TA(I,J+1,K)
+
      + AST*TA(I,J-1,K) + ATT*TA(I,J,K+1)
+
      + ABT*TA(I,J,K-1)) - TA(I,J,K))
IF (K.EQ.KMAX) THEN
TA(I,J,KMAX+1) = TA(I,J,KMAX)
ENDIF
ERRT = ERRT + ABS(TA(I,J,K) - PRV)
10 CONTINUE
DO 20 I=IMIN+1,IMAX-1
DO 20 J=JTIP+1,JMAX-1
DO 20 K=0,KMAX
DXE = XM(I+1) - XM(I)
DXW = XM(I) - XM(I-1)
DYN = YM(J+1) - YM(J)
DYS = YM(J) - YM(J-1)
DZT = ZM(K+1) - ZM(K)
DZB = ZM(K) - ZM(K-1)
DELX = (XM(I+1) - XM(I-1))/2.
DELY = (YM(J+1) - YM(J-1))/2.
DELZ = (ZM(K+1) - ZM(K-1))/2.
FE = U(I,J,K)*DELY*DELZ
FW = U(I-1,J,K)*DELY*DELZ
FN = V(I,J,K)*DELX*DELZ
FS = V(I,J-1,K)*DELX*DELZ
FT = W(I,J,K)*DELX*DELY
FB = W(I,J,K-1)*DELX*DELY
DE = CTX*DELY*DELZ/DXE
DW = CTX*DELY*DELZ/DXW
DN = CTY*DELZ*DELX/DYN
DS = CTY*DELZ*DELX/DYS

```

```

DT = CTZ*DELX*DELY/DZT
DB = CTZ*DELX*DELY/DZB
PE = FE/DE
PW = FW/DW
PN = FN/DN
PS = FS/DS
PT = FT/DT
PB = FB/DB
AET = COEF (DE,PE,-FE)
AWT = COEF (DW,PW,FW)
ANT = COEF (DN,PN,-FN)
AST = COEF (DS,PS,FS)
ATT = COEF (DT,PT,-FT)
ABT = COEF (DB,PB,FB)
APT = AET + AWT + ANT + AST + ATT + ABT
CALL CHECKCOEF(APT,AET,AWT,ANT,AST,ATT,ABT)
PRV = TA(I,J,K)
TA(I,J,K) = TA(I,J,K) + URT*((1./APT)*(AET*TA(I+1,J,K)
+          + AWT*TA(I-1,J,K) + ANT*TA(I,J+1,K)
+          + AST*TA(I,J-1,K) + ATT*TA(I,J,K+1)
+          + ABT*TA(I,J,K-1)) - TA(I,J,K))
IF (K.EQ.KMAX) THEN
TA(I,J,KMAX+1) = TA(I,J,KMAX)
ENDIF
ERRT = ERRT + ABS(TA(I,J,K) - PRV)
20 CONTINUE
C *** CAL. AIR TEMP. AT BOUNDARY ***
CALL AIRTEMP_AT_BOUND ()
CALL INTERFACE_TEMP_CAL (B,S)
IF (ERRT.LE.ERRMAX) GO TO 45
IF (ITER-ITMAX) 5,45,45
45 WRITE(*,*) 'AIR TEMP CAL. PASSED'
WRITE(*,*) 'NO. OF ITER. = ',ITER

```

```

WRITE(*,*) 'SUM OF ERROR = ',ERRT
C
C WRITE(9,*) 'AIR TEMP CAL. PASSED'
C WRITE(9,*) 'NO. OF ITER. = ',ITER
C WRITE(9,*) 'SUM OF ERROR = ',ERR
RETURN
END
C
C .....
C
SUBROUTINE AIRTEMP_AT_BOUND ()
C IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+ IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+ V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+ W(IMIN:IMAX,0:JMAX,-1:KMAX),
+ P(IMIN:IMAX,0:JMAX,0:KMAX),
+ PC(IMIN:IMAX,0:JMAX,0:KMAX),
+ TF(IMIN:0,0:JTIP,0:KMAX),
+ TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
COMMON/VARIABLE/U,V,W,P,PC,TA,TF
C *** TEMP. AT RIGHT SIDE ***
DO 10 J=1,JMAX-1
DO 10 K=0,KMAX
10 TA(IMAX,J,K) = (4.*TA(IMAX-1,J,K) - TA(IMAX-2,J,K))/3.
C *** TEMP. AT LEFT SIDE ***
DO 20 J=JTIP+1,JMAX-1
DO 20 K=0,KMAX

```

```

20 TA(IMIN,J,K) = (4.*TA(IMIN+1,J,K) - TA(IMIN+2,J,K))/3.
C *** TEMP. AT OUTFLOW ***
DO 30 I=1,IMAX
DO 30 J=1,JMAX-1
30 TA(I,J,KMAX+1) = TA(I,J,KMAX)
DO 40 I=IMIN,0
DO 40 J=JTIP+1,JMAX-1
40 TA(I,J,KMAX+1) = TA(I,J,KMAX)
RETURN
END
C
C .....
C
SUBROUTINE INTERFACE_TEMP_CAL (B,S)
C IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+ IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
c PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
c + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+ XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+ XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+ XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+ V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+ W(IMIN:IMAX,0:JMAX,-1:KMAX),
+ P(IMIN:IMAX,0:JMAX,0:KMAX),
+ PC(IMIN:IMAX,0:JMAX,0:KMAX),
+ TF(IMIN:0,0:JTIP,0:KMAX),
+ TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW

```


COMMON/VARIABLE/U,V,W,P,PC,TA,TF

COMMON/CONDUCT/KA,KF

REAL KA,KF

C *** TEMP. AT FIN'S FACE ***

DELXA = XM(1) - XM(0)

DELXF = XM(0) - XM(-1)

C1 = 1./(2.*DELXF)

C2 = KA/(KF*2.*DELXA)

DO 10 J=1,JTIP-1

DO 10 K=0,KMAX

TA(0,J,K) = (4.*C1*TF(-1,J,K) - C1*TF(-2,J,K) + 4.*C2*TA(1,J,K)

+ - C2*TA(2,J,K))/(3.*C1 + 3.*C2)

TF(0,J,K) = TA(0,J,K)

10 CONTINUE

C *** TEMP. AT FIN'S TIP ***

DELYA = YM(JTIP+1) - YM(JTIP)

DELYF = YM(JTIP) - YM(JTIP-1)

D1 = 1./(2.*DELYF)

D2 = KA/(KF*2.*DELYA)

DO 20 I=IMIN+1,-1

DO 20 K=0,KMAX

TA(I,JTIP,K) = (4.*D1*TF(I,JTIP-1,K) - D1*TF(I,JTIP-2,K) +

+ 4.*D2*TA(I,JTIP+1,K) - D2*TA(I,JTIP+2,K))/

+ (3.*D1 + 3.*D2)

TF(I,JTIP,K) = TA(I,JTIP,K)

20 CONTINUE

DO 30 K=0,KMAX

J = JTIP

TA(IMIN,J,K) = (4.*TA(IMIN+1,J,K) - TA(IMIN+2,J,K))/3.

TF(IMIN,J,K) = TA(IMIN,J,K)

30 CONTINUE

C *** TEMP. AT THE CORNER OF FIN'S TIP ***

E1 = KF*B*DELYF/((S/2.)*DELXF)

```

E2 = KF*(S/2.)*DELXF/(B*DELYF)
E3 = KA*B*DELYF/((S/2.)*DELXA)
E4 = KA*(S/2.)*DELXF/(B*DELYA)
DO 40 K=0,KMAX
TA(0,JTIP,K) = (4.*E1*TF(-1,JTIP,K) - E1*TF(-2,JTIP,K) +
+ 4.*E2*TF(0,JTIP-1,K) - E2*TF(0,JTIP-2,K) +
+ 4.*E3*TA(1,JTIP,K) - E3*TA(2,JTIP,K) +
+ 4.*E4*TA(0,JTIP+1,K) - E4*TA(0,JTIP+2,K))/
+ (3.*(E1 + E2 + E3 + E4))
TF(0,JTIP,K) = TA(0,JTIP,K)
40 CONTINUE
RETURN
END
C
C *****
C
SUBROUTINE FIN_TEMP_CAL (B,S,MAXIT,ERRF)
C IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+ IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
C PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
C + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
C PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
C + IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+ XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+ XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+ XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+ V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+ W(IMIN:IMAX,0:JMAX,-1:KMAX),
+ P(IMIN:IMAX, 0:JMAX,0:KMAX),
+ PC(IMIN:IMAX,0:JMAX,0:KMAX),

```

```

+   TF(IMIN:0,0:JTIP,0:KMAX),
+   TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
DIMENSION TFOLD(IMIN:0,0:JTIP,0:KMAX)
COMMON/COORDINATE/XM,YM,ZM,XU,YU,ZU,XV,YV,ZV,XW,YW,ZW
COMMON/VARIABLE/U,V,W,P,PC,TA,TF
COMMON/CONDUCT/KA,KF
REAL KA,KF,ER
C   *** START ITERATION ***
ITER = 0
ITERMAX = MAXIT
5  ITER = ITER + 1
   ERRF = 0.
C   *** TEMP. INSIDE FIN ***
DO 10 I=IMIN+1,-1
DO 10 J=1,JTIP-1
DO 10 K=0,KMAX
DXE = XM(I+1) - XM(I)
DXW = XM(I) - XM(I-1)
DYN = YM(J+1) - YM(J)
DYS = YM(J) - YM(J-1)
DELX = (DXE + DXW)/2.
DELY = (DYN + DYS)/2.
AE = 4.*DELY/DXE
AW = 4.*DELY/DXW
AN = ((S/B)**2)*DELX/DYN
AS = ((S/B)**2)*DELX/DYS
AP = AE + AW + AN + AS
TFPRV = TF(I,J,K)
TF(I,J,K) = (1./AP)*(AE*TF(I+1,J,K) + AW*TF(I-1,J,K) +
+   AN*TF(I,J+1,K) + AS*TF(I,J-1,K))
ERRF = ERRF + ABS(TF(I,J,K) - TFPRV)
10 CONTINUE
C   *** TEMP. ON THE LEFT SIDE (DT/DX=0) ***

```

```

DO 11 J=1,JTIP-1
DO 11 K=0,KMAX
11 TF(IMIN,J,K) = (4.*TF(IMIN+1,J,K) - TF(IMIN+2,J,K))/3.
C *** TEMP. AT INTERFACE ***
DO 15 J=1,JTIP
DO 15 K=0,KMAX
15 TFOLD(0,J,K) = TF(0,J,K)
DO 16 I=IMIN,-1
DO 16 K=0,KMAX
16 TFOLD(I,JTIP,K) = TF(I,JTIP,K)
CALL INTERFACE_TEMP_CAL (B,S)
DO 17 J=1,JTIP
DO 17 K=0,KMAX
17 ERRF = ERRF + ABS(TF(0,J,K) - TFOLD(0,J,K))
DO 18 I=IMIN,-1
DO 18 K=0,KMAX
18 ERRF = ERRF + ABS(TF(I,JTIP,K) - TFOLD(I,JTIP,K))
C WRITE(*,*) 'ITERATION NO.',ITER
C WRITE(*,*) 'SUM OF ERROR = ',ER
IF (ERRF.LE.0.00005) GO TO 30
IF (ITER-ITERMAX) 5,30,30
30 WRITE (*,*) 'FIN TEMP. CAL PASSED'
WRITE (*,*) 'NO. OF ITER. = ',ITER
WRITE (*,*) 'SUM OF ERROR = ',ERRF
C PAUSE ''
C WRITE (9,*) 'FIN TEMP. CAL PASSED'
C WRITE (9,*) 'NO. OF ITER. = ',ITER
C WRITE (9,*) 'SUM OF ERROR = ',ER
RETURN
END
C
C .....
C

```

```

SUBROUTINE HEAT_CAL (TB,TS,B,L,S,Q,NOF)
C  IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (YMAX=3., JMAX=100, YLIM=1.5, JLIM=75, XMAX=1.,
+   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
C  PARAMETER (YMAX=3., JMAX=200, YLIM=1.5, JLIM=75, XMAX=1.,
+   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
C  PARAMETER (YMAX=5., JMAX=125, YLIM=1.5, JLIM=75, XMAX=1.,
+   IMAX=10, IMIN=-5, ZMAX=1., KMAX=25, JTIP=50)
DIMENSION XM(IMIN-1:IMAX+1), YM(-1:JMAX+1), ZM(-1:KMAX+1),
+   XU(IMIN-1:IMAX), YU(0:JMAX), ZU(-1:KMAX+1),
+   XV(IMIN:IMAX), YV(-1:JMAX), ZV(-1:KMAX+1),
+   XW(IMIN:IMAX), YW(0:JMAX), ZW(-1:KMAX)
DIMENSION U(IMIN-1:IMAX,0:JMAX,-1:KMAX+1),
+   V(IMIN:IMAX,-1:JMAX,-1:KMAX+1),
+   W(IMIN:IMAX,0:JMAX,-1:KMAX),
+   P(IMIN:IMAX, 0:JMAX,0:KMAX),
+   PC(IMIN:IMAX,0:JMAX,0:KMAX),
+   TF(IMIN:0,0:JTIP,0:KMAX),
+   TA(IMIN:IMAX,0:JMAX,-1:KMAX+1)
COMMON/COORDINATE/XM, YM, ZM, XU, YU, ZU, XV, YV, ZV, XW, YW, ZW
COMMON/VARIABLE/U, V, W, P, PC, TA, TF
COMMON/CONDUCT/KA, KF
REAL L, KA, KF, NOF
Q = 0.
C  *** HEAT TRANS. FROM FIN BASE ***
DY = YM(1) - YM(0)
DO 10 I=0,IMAX
DO 10 K=0,KMAX
IF (I.EQ.0) THEN
DELX = (XM(1) - XM(0))/2.
ELSEIF (I.EQ.IMAX) THEN
DELX = (XM(IMAX) - XM(IMAX-1))/2.
ELSE

```

```

DELX = (XM(I+1) - XM(I-1))/2.
ENDIF
IF (K.EQ.0) THEN
DELZ = (ZM(1) - ZM(0))/2.
ELSEIF (K.EQ.KMAX) THEN
DELZ = (ZM(KMAX) - ZM(KMAX-1))/2.
ELSE
DELZ = (ZM(K+1) - ZM(K-1))/2.
ENDIF
AREA = DELX*DELZ*(S/2.)*L*1.E-6
Q = Q + (-KA)*AREA*(TB-TS)/((S/2.)*1.E-3)*
+ (-3.*TA(I,0,K)+4.*TA(I,1,K)-TA(I,2,K))/2.*DY)
10 CONTINUE
C *** HEAT TRAN. AT FIN'S FACE ***
DX = XM(1) - XM(0)
DO 20 J=0, JTIP
DO 20 K=0, KMAX
IF (J.EQ.0) THEN
DELY = (YM(1) - YM(0))/2.
ELSEIF (J.EQ. JTIP) THEN
DELY = (YM(JTIP) - YM(JTIP-1))/2.
ELSE
DELY = (YM(J+1) - YM(J-1))/2.
ENDIF
IF (K.EQ.0) THEN
DELZ = (ZM(1) - ZM(0))/2.
ELSEIF (K.EQ.KMAX) THEN
DELZ = (ZM(KMAX) - ZM(KMAX-1))/2.
ELSE
DELZ = (ZM(K+1) - ZM(K-1))/2.
ENDIF
AREA = DELY*DELZ*B*L*1.E-6
Q = Q + (-KA)*AREA*(TB-TS)/((S/2.)*1.E-3)*

```

```

+ (-3.*TA(0,J,K)+4.*TA(1,J,K)-TA(2,J,K))/(2.*DX)
20 CONTINUE
C *** HEAT TRAN. AT FIN'S TIP ***
DY = YM(JTIP+1) - YM(JTIP)
DO 30 I=IMIN,0
DO 30 K=0,KMAX
IF (I.EQ.IMIN) THEN
DELX = (XM(IMIN+1) - XM(IMIN))/2.
ELSEIF (I.EQ.0) THEN
DELX = (XM(0) - XM(-1))/2.
ELSE
DELX = (XM(I+1) - XM(I-1))/2.
ENDIF
IF (K.EQ.0) THEN
DELZ = (ZM(1) - ZM(0))/2.
ELSEIF (K.EQ.KMAX) THEN
DELZ = (ZM(KMAX) - ZM(KMAX-1))/2.
ELSE
DELZ = (ZM(K+1) - ZM(K-1))/2.
ENDIF
AREA = DELX*DELZ*(S/2.)*(L)*1.E-6
Q = Q + (-KA)*AREA*(TB-TS)/(B*1.E-3)*
+ (-3.*TA(I,JTIP,K)+4.*TA(I,JTIP+1,K)-TA(I,JTIP+2,K))/
+ (2.*DY)
30 CONTINUE
Q = NOF*Q
WRITE(*,*) 'HEAT CAL PASSED'
WRITE(*,*) 'TOTAL HEAT TRANSFER = ',Q
C PAUSE 'PRESS ENTER'
C WRITE(9,*) 'HEAT CAL PASSED'
C WRITE(9,*) 'TOTAL HEAT TRANSFER = ',Q
RETURN
END

```

ประวัติผู้เขียน

นาย อัญชลพร ขรจันทร์ เกิดเมื่อวันที่ 28 มีนาคม พ.ศ. 2514 ที่อำเภอเมือง จังหวัดระนอง สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ในปีการศึกษา 2534 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อพ.ศ. 2536 ปัจจุบันเป็นพนักงานบริษัท ไทยซีอาร์ที่ อำเภอศรีราชา จังหวัดชลบุรี



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย