Appendix A
Computer Program Listing

```
      PROGRAM PP9F
C PROGRAM TO CALCULATE TWO-PHASE EJECTOR FLOW BY
C SPALDING'S IPSA METHOD
C INCOMPRESSIBLE, PARABOLIC FLOW
C MIXING-LENGTH TURBULENCE MODEL WITH WALL FUNCTION
C SOLVES MOMENTUM, CONTINUITY FOR BOTH PHASES
C DOES NOT SOLVE ENERGY EQUATION
C GUESSED DROPLET DIAMETER
C CONTAINS BODY FITTED GRID
C SOLVES FOR FLOW IN MIXING SECTION UP TO POINT OF RECIRCULATION.
C
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION R(2000),U(2,2000),V(2,2000),UU(2,2000),VU(2,2000)
     2,UG(2,2000)
     2,VG(2,2000),VISC(2,2000),RHO(2,2000),RHOU(2,2000),ALPHA(2,2000)
     2,ALPHAU(2,2000),VISCU(2,2000)
     2,RINT1S(2000),RINT1N(2000),RINT2S(2000),RINT2N(2000)
     2,RINT4S(2000),RINT4N(2000),RINT3M(2000),RINTE(2000),RINTW(2000)
     2,RINT5(2000),DELPR(2000),RINT6S(2000),RINT6N(2000),C1IFD(2000)
     2,RDRAG(2000),GAM(2,2000),ALPHANEW(2,2000)
     2,U3(2,85,85),VISCL(2,2000),VISCLU(2,2000),VISCT(2,2000)
     2,VISCTU(2,2000)
     2,V3(2,85,85),DELPR3(85,85),P3(2000),ALPHA3(2,85,85)
     2,RHO3(2,85,85),VISC3(2,85,85),RINTE3(85,85),RINT53(85,85)
     2,RINT4N3(85,85),RINT6N3(85,85),RINT3M3(85,85),C1IFD3(85,85)
     2,C1IFDX(2000),C1IFDR(2000),UTURBOLD(2,2000),VISCTURBOLD(2,2000)
     2,UTEMP(2,2000),VTEMP(2,2000)
     2,ALTEST(2,2000),ALTEST2(2,2000)
C
      DOUBLE PRECISION LINNERRCO,LOUTERRCO,JETTHICK,LMIX(2000)
     2,JETTHICKOLD,KAPPA
C
      OPEN(13,FILE='OUT1.PRN')
      OPEN(14,FILE='OUT2.PRN')
      OPEN(15,FILE='OUT3.PRN')
      OPEN(16,FILE='OUT4.PRN')
      OPEN(17,FILE='OUT5.PRN')
      OPEN(18,FILE='OUT6.PRN')
C
      IGOWALL=0
C     INPUT AND SETUP ROUTINES
      CALL USERINPUT(RE,XMAX,NCVX,NCVR,NCVRMOT,URATIO,RRATIO
     2,SLIPMN,SLIPSN,RHOVAPOR,VISCVAPOR,ALPHAVAPORMN,ALPHAVAPORSN
     2,IREGIME,IPHCONT,IPHDISC,RHOLIQ,VISCLIQ)
C
      CALL GRID(NCVX,NCVR,NCVRMOT,XMAX,DELX,DELR,M,N,NMOT,R)
C
      CALL FLUIDPROP(RHOVAPOR,VISCVAPOR,ALPHAVAPORMN,ALPHAVAPORSN
```

```
     2,RHO,VISCL,ALPHA,RHOLIQ,VISCLIQ,N,NMOT)
C
      CALL INIT(N,NMOT,R,RRATIO,URATIO,SLIPMN,SLIPSN
     2,RHOVAPOR,ALPHAVAPORMN
     2,VISCVAPOR,ALPHAVAPORSN
     2,UU,VU,RHOU,ALPHAU,VISCLU,IMERGE,RHOLIQ,VISCLIQ)
C
      CALL GUESSVEL(UG,VG,UU,VU,N)
C
      CALL PINIT(DELPCOEFF,PW)
C
      CALL GLOBX1INIT(DMOMIN,DMOMOUT,PFIN,PFOUT,FVISCTOP,PFTOP)
C SET SOME MOMENTUM CHECK TERMS TO ZERO
      TOTFTAU2=0.0D0
      TOTFTAU1=0.0D0
      TOTPF2IN=0.0D0
      TOTPF1IN=0.0D0
      TOTPF2OUT=0.0D0
      TOTPF1OUT=0.0D0
      TOTFINT1=0.0D0
      TOTFINT2=0.0D0
C
C
      X=DELX
      I=2
      IPH=2
      CALL GEOMCOEFF(N,R,DELR,X,DELX,RINT1S,RINT1N,RINT2S,RINT2N
     2,RINT4S,RINT4N,RINT6S,RINT6N,RINT3M,RINTE,RINTW,RINT5,RDXM,RTXE
     2,IGOWALL,RDXW,RDXE)
C
      CALL CALCFLOWIN(IPHCONT,N,RHOU,ALPHAU,UU,RINTW,FLOWIN,UAVE,UAVE2
     2,TOTFLOW1,TOTFLOW2)
C
      CALL GLOBCONT(I,M,N,ALPHA,ALPHAU,RHO,RHOU,U,UU,RINTW,RINTE
     2,FLOWINTOT)
C
      CALL PARTICLESIZE(N,RDRAG)
C
C INITIAL VALUES OF KAPPA AND B - TO BE USED IN FINDNEARWALLPROFILE
      KAPPA=.6D0
      B=5.0D0
C
C     FIND A TAUWALL TO USE INITIALLY FROM EQN. 6-55 IN WHITE
      TAUWALL655=RE**(.75D0)*(.0396D0*RHO(IPH,N-1)**.75D0
     2*UAVE2**(7.0D0/4.0D0)*VISCL(IPH,N-1)**.25D0
     2*(2.0D0*RDXW)**(-.25D0))
      TAUWALL=TAUWALL655
C
C CALL TO SELECT A POINT NEAR THE WALL WITHIN RANGE YPLUS=30 TO 200
      RLOW=90.0D0
      RHIGH=140.0D0
      CALL FINDJNEARWALL(IPH,N,RE,RDXW,R,RHO,TAUWALL,RTMINUSRPLUS
     2,RNEARWALL,RLOW,RHIGH,JNEARWALL,VISCL)
C
C CALL TO FIND THE VELOCITY PROFILE NEAR THE WALL BASED ON
C THE WALL FUNCTION
```

```
      CALL FINDNEARWALLPROFILE(IPH,N,JNEARWALL,KAPPA,B,RE,RDXW
     2,TAUWALL,RHO,R,VISCL,U,UG)
C
C ESTABLISH SOME INITIAL VALUES OF JETTHICKOLD AND CORETHICKOLD
      JJETTHICKOLD=NMOT
      JETTHICKOLD=(2.0D0*DBLE(JJETTHICKOLD)-3.0D0)/(2.0D0*DBLE(NCVR))
      JCORETHICKOLD=NMOT-1
      CORETHICKOLD=(2.0D0*DBLE(JCORETHICKOLD)-3.0D0)/(2.0D0*DBLE(NCVR))
      JWALLTHICKOLD=N-1
      WALLTHICKOLD=(2.0D0*DBLE(JWALLTHICKOLD)-3.0D0)/(2.0D0*DBLE(NCVR))
C
2     KACCESS=0
      JWALLTHICKPREV=0
      JJETTHICKPREV=0
      JCORETHICKPREV=0
3     IPH=2
      DIRE=1.0D0
      IPHOTHER=1
      KTAUINTERP=1
      KNEW=0
      DO 950 J=2,N-1
       UTURBOLD(IPH,J)=0.0D0
       VISCTURBOLD(IPH,J)=0.0D0
950   CONTINUE
5     CONTINUE
C SEPARATE OUT THICK AND TURBVISC OPERATIONS
C
C FINDS THICKNESS OF THE JET, WALL, AND CORE TO BE USED IN TURBULENCE
C MODEL
      CALL THICK2(N,I,IPH,R,U,DELR,WALLTHICK,JETTHICK,CORETHICK
     2,JWALLTHICK,JJETTHICK,JCORETHICK,IMERGE
     2,CORETHICKOLD,JCORETHICKOLD,WALLTHICKOLD,JWALLTHICKOLD
     2,JETTHICKOLD,JJETTHICKOLD,KACCESS
     2,JCORETHICKPREV,JWALLTHICKPREV,JJETTHICKPREV)
C
C FINDS TURBULENT VISCOSITIES BY MIXING LENGTH METHOD
10    CALL TURBVISC(I,N,IPH,DELX,X,R,RDXM,DELR
     2,RE,UG,UU,DELPX,RHO,RHOU,VISCL
     2,VISCLU,VISCT,VISCTU,VISC,VISCU,RCO,LINNERRCO,LOUTERRCO,WALLTHICK
     2,JETTHICK,CORETHICK,IMERGE,LMIX,RCOD,JRCOD
     2,CORETHICKOLD,JCORETHICKOLD,WALLTHICKOLD,JWALLTHICKOLD
     2,JETTHICKOLD,JJETTHICKOLD,JWALLTHICK,JJETTHICK,JCORETHICK,KACCESS)
C
11    CONTINUE
C
C     FINDS INTERFACIAL DRAG FORCE
      CALL INTDRAG(I,N,IREGIME,IPHCONT,IPHDISC,UG,UU,VG,VU,RHO,VISCL
     2,ALPHA,ALPHAU,ALPHAVAPORMN,RDRAG,RE,C1IFD)
      DO 12 J=2,N-1
       ALTEST(IPHDISC,J)=ALPHA(IPHDISC,J)
12    CONTINUE
C
C FINDS INTERFACIAL MASS TRANSFER (DISABLED IN THIS VERSION)
      CALL INTMASS(N,GAM)
      KDELPITER=1
      DELPCOEFF=0.0D0
```

```
C BEGIN ITERATION LOOP TO FIND AXIAL PRESSURE GRADIENT, DELPX

20    DELPX=DELPCOEFF*RHO(IPH,1)*UU(IPH,1)**2
      PE=PW+DELPX
C     SOLVE CONTINUOUS PHASE X-MOMENTUM EQUATION FOR U
C     ITERATE ON DELPX UNTIL GLOBAL MASS IS SATISFIED
      CALL X1MOM(ERRALPHA,I,M,IPH,IPHOTHER,N,X,R,DELX,JNEARWALL
     2,DELR,RHO,ALPHA,VISC
     2,UU,UG,VU,VG,U,V,PE,PW,RE,ALPHAU,RHOU,C1IFD,GAM
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT4S,RINT4N,RINT3M,RINTE,RINTW)
C
      CALL FLOWERROR(IPH,N,RHO,ALPHA,U,FLOWIN,RINTE,FLOWERR)
C      IF(ABS(FLOWERR).LT.1.0D-14)GO TO 30
C      IF(ABS(FLOWERR).LT.1.0D-13)GO TO 30
      IF(ABS(FLOWERR).LT.1.0D-10)GO TO 30
      CALL DPITER(DIRE,KDELPITER,DELPCOEFF,FLOWERR)
      GO TO 20
30    CONTINUE
C     SOLVE THE CONTINUITY EQUATION FOR CONT. PHASE FOR V
C     ITERATE UNTIL U,UG AND V,VG CONVERGE
      CALL CONT1(IPH,N,X,R,DELX,DELR,U,UU,V,RHO,RHOU,ALPHA,ALPHAU
     2,GAM,VU
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT3M,RINTE,RINTW)
      CALL CONERROR(N,IPH,U,UG,V,VG,CONERR)
C      IF(ABS(CONERR).LT.1.0D-13)GO TO 40
C      IF(ABS(CONERR).LT.1.0D-10)GO TO 40
      IF(ABS(CONERR).LT.1.0D-8)GO TO 40
      CALL VELUPDATE(IPH,I,U,V,UG,VG,N,JNEARWALL)
C
      GO TO 10
C
40    CONTINUE
70    FORMAT(1X,2I5,5E14.6)
C
C CHECK GLOBAL CONSERVATION OF X-MOMENTUM AND UPDATE TAU
      CALL GLOBX1MOM1ERROR(IPH,IPHOTHER,N,JNEARWALL,U,UG,UU,RHO
     2,RHOU,ALPHA,ALPHAU
     2,RINTW,RINTE,RINT3M,PE,PW,TAUWALL,RINT1N,RINT2N,RE,GLOBX1MOM1ERR
     2,DMOMIN1,PFIN1,DMOMOUT1,DMOMOUT1NEAR,PFOUT1,FTAUWALL1,PFTOP1
     2,C1IFD,RINT4N)
C
      RTMINUSRPLUSWRITE=RE**.5D0*RDXW*(1.0D0-R(JNEARWALL))
     2*(RHO(IPH,N-1)*TAUWALL)**.5D0/VISCL(IPH,N-1)
C
C      IF(ABS(GLOBX1MOM1ERR).LE.1.0D-13)GO TO 28
C      IF(ABS(GLOBX1MOM1ERR).LE.1.0D-10)GO TO 28
      IF(ABS(GLOBX1MOM1ERR).LE.1.0D-8)GO TO 28
C
      KNEW=KNEW+1
      IF(KNEW.GT.100)THEN
        KTAUINTERP=1
        KNEW=0
      ENDIF
C FIND NEW TAUWALL (WALL SHEAR STRESS)
      CALL FINDNEWTAUWALL(IPH,N,JNEARWALL,KTAUINTERP
     2,GLOBX1MOM1ERR,TAUWALL)
```

```
C     IF RTMINUSRPLUS IS WITHIN RANGE 30 TO 200 DON'T CHANGE JNEARWALL
C     IE KEEP USING THE SAME JNEARWALL AS LONG AS IT STAYS WITHIN RANGE
      RTMINUSRPLUSCHECK=RE**.5D0*RDXW*(1.0D0-R(JNEARWALL))
     2*(RHO(IPH,N-1)*TAUWALL)**.5D0/VISCL(IPH,N-1)
      IF(RTMINUSRPLUSCHECK.GE.30.AND.RTMINUSRPLUSCHECK.LE.200)GO TO 310
C
      RLOW=50.0D0
      RHIGH=200.0D0
      CALL FINDJNEARWALL(IPH,N,RE,RDXW,R,RHO,TAUWALL,RTMINUSRPLUS
     2,RNEARWALL,RLOW,RHIGH,JNEARWALL,VISCL)
C
310   CONTINUE
C
C CALL TO FIND THE VELOCITY PROFILE NEAR THE WALL BASED
C ON THE WALL FUNCTION
      CALL FINDNEARWALLPROFILE(IPH,N,JNEARWALL,KAPPA,B,RE,RDXW
     2,TAUWALL,RHO,R,VISCL,U,UG)
C
      GO TO 10
C
28    CONTINUE
      KNEW=0
C SAVE VALUES OF UG(IPHOTHER,J), VG(IPHOTHER,J) FOR FUTURE ITERATION
      DO 910 J=1,N
       UTEMP(1,J)=UG(IPHOTHER,J)
       VTEMP(1,J)=VG(IPHOTHER,J)
910   CONTINUE
C REMEMBER CERTAIN VALUES FOR CONTINUOUS PHASE THAT WILL BE OVERWRITTEN
C BY DISCONTINUOUS PHASE
      TAUWALLC=TAUWALL
      JNEARWALLC=JNEARWALL
      RNEARWALLC=RNEARWALL
C     GO TO 75 FOR JUST SINGLE PHASE CALCULATION
C      GO TO 75
C
C
C     FIND THE PRESSURE GRADIENT IN THE R-DIR. FROM R1-MOMENTUM
      CALL DELPRCALC(I,N,R,RINT2N,RINT2S,RINT1N,RINT1S,RINT4N,RINT4S
     2,RINT6S,RINT6N,RINT3M,RINT5,RINTE,RINTW,DELX,DELR,GAM,C1IFD
     2,RE,U,UU,VG,VU,RHO,RHOU,ALPHA,ALPHAU,VISC,VISCU,DELPR
     2,IPH,IPHOTHER,RDXW,RDXE)
C     TWO-PHASE CALCULATION
      IPH=1
      IPHOTHER=2
      KTAUINTERP=1
C
C TAUWALL FOR DISCONTINUOUS PHASE / BOUNDARY VELOCITY FOR DISC. PHASE
      TAUWALLD=TAUWALLC*VISCL(1,N-1)/VISCL(2,N-1)
      DO 71 J=JNEARWALL,N
       UG(IPH,J)=U(IPHOTHER,J)
       U(IPH,J)=U(IPHOTHER,J)
71    CONTINUE
C
C TURBULENT VISCOSITY FOR DISCONTINUOUS PHASE
72    CONTINUE
      DO 73 J=2,N-1
```

```
       UN=.5D0*(UG(IPH,J+1)+UG(IPH,J))
       US=.5D0*(UG(IPH,J)+UG(IPH,J-1))
       IF(J.EQ.N-1)UN=0.0D0
       IF(J.EQ.2)US=UG(IPH,2)
       DUDR=(UN-US)/DELR
       VISCT(IPH,J)=RE*RHO(IPH,J)*LMIX(J)**2*DABS(DUDR)/RDXM
       VISCT(IPH,J)=VISCT(IPHOTHER,J)*RHO(IPH,J)/RHO(IPHOTHER,J)
       VISC(IPH,J)=VISCT(IPH,J)+VISCL(IPH,J)
73    CONTINUE
C     SOLVE FOR V2 FROM R2MOM - DISCONTINUOUS PHASE
1000  CONTINUE
      CALL R2MOM(I,IPH,IPHOTHER,N,DELX,DELR,RHO,ALPHA,VISC,UU,UG,VU,VG
     2,V,RE,ALPHAU,RHOU,DELPR,RINT5,RINT6N,RINT6S,VISCU,GAM,C1IFD
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT4S,RINT4N,RINT3M,RINTE,RINTW
     2,RDXW,RDXE)
C     SOLVE FOR U2 FROM X2MOM - DISCONTINUOUS PHASE
      CALL X2MOM(I,M,IPH,IPHOTHER,N,X,R,DELX,JNEARWALL,DELR,RHO
     2,ALPHA,VISC,UU,UG,VU,VG
     2,U,V,PE,PW,RE,ALPHAU,RHOU,C1IFD,GAM,TAUWALLD
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT4S,RINT4N,RINT3M,RINTE,RINTW)
C     UPDATE GUESS VELOCITIES - COMPARE U(IPH,J) WITH UG(IPH,J)
C                                       V(IPH,J) WITH VG(IPH,J)
      CALL ERRORVEL2(I,N,IPH,U,UG,V,VG,ERRVEL2,ALPHA)
      IF(ERRVEL2.LE.1.0D-8)GO TO 74
      CALL UPDATEVEL2(I,IPH,N,U,UG,V,VG)
      GO TO 72
74    CONTINUE
C
C NEW ITERATION LOOP FOR SECOND PHASE VELOCITY USED IN
C FIRST PHASE ROUTINE
C COMPARE VALUES OF U CALC. HERE WITH UTEMP(IPHOTHER,J)
      UTEMPADD=0.0D0
      UADD=0.0D0
      VTEMPADD=0.0D0
      VADD=0.0D0
      DO 920 J=2,N-1
       UTEMPADD=UTEMP(1,J)+UTEMPADD
       UADD=U(IPH,J)+UADD
       VTEMPADD=VTEMP(1,J)+VTEMPADD
       VADD=V(IPH,J)+VADD
920   CONTINUE
      UDIFF=UTEMPADD-UADD
      VDIFF=VTEMPADD-VADD
      WRITE(*,*)'UDIFF,VDIFF = ',UDIFF,VDIFF
C  REMOVE/REPLACE FOR UDIFF ITERATION
C      IF(ABS(UDIFF).LE.1.0D-13)GO TO 930
C      IF(ABS(UDIFF).LE.1.0D-10)GO TO 930
      IF(ABS(UDIFF).LE.1.0D-8)GO TO 930
      IPH=2
      IPHOTHER=1
      GO TO 11
C
C
C
930   CONTINUE
      WRITE(*,931)JWALLTHICK,JJETTHICK,JCORETHICK
```

```
931   FORMAT(1X,3I7)
C RETURN TO THICK ROUTINE TO OBTAIN CONVERGENCE
      IF(JWALLTHICKPREV.EQ.JWALLTHICK.AND.JCORETHICKPREV.EQ.JCORETHICK
     2.AND.JJETTHICKPREV.EQ.JJETTHICK)GO TO 940
      IPH=2
      IPHOTHER=1
      JWALLTHICKPREV=JWALLTHICK
      JCORETHICKPREV=JCORETHICK
      JJETTHICKPREV=JJETTHICK
      GO TO 5
940   CONTINUE
C     SOLVE FOR VOID FRACTION (ALPHA2) FROM DISC.
C     PHASE CONTINUITY (CONT2)
      CALL CONT2(I,IPH,N,X,R,DELX,DELR,U,UU,V,RHO,RHOU,ALPHANEW
     2,ALPHA,ALPHAU,GAM,RINT1S,RINT1N,RINT2S,RINT2N,RINT3M,RINTE
     2,RINTW)
      DO 941 J=2,N-1
       ALTEST2(IPHDISC,J)=ALPHANEW(IPHDISC,J)
941   CONTINUE
C     CHECK FOR ALPHA (VOID FRACTION) CONVERGENCE
      CALL ERRORALPHA(N,IPH,IPHOTHER,ALPHA,ALPHANEW,ERRALPHA)
      WRITE(*,932)I,X*5.0D0/16.0D0,ERRALPHA,UVDIFF,PE*1264.45D0,TAUWALL
932   FORMAT(1X,I4,5E14.7)
C      IF(ERRALPHA.LE.1.0D-12)GO TO 75
C      IF(ERRALPHA.LE.1.0D-8)GO TO 75
      IF(ERRALPHA.LE.1.0D-6)GO TO 75
      CALL UPDATEALPHA(N,I,IPH,IPHOTHER,ALPHANEW,ALPHA,ERRALPHA)
      IPH=2
      IPHOTHER=1
      GO TO 5
75    CONTINUE
      WRITE(*,*)'RTMINUSRPLUS = ',RTMINUSRPLUSCHECK,I
      WRITE(*,*)'TAUWALL = ',TAUWALL
C CHECK GLOBAL MOMENTUM BALANCE FOR SECOND PHASE
      CALL GLOBX1MOM1ERROR(IPH,IPHOTHER,N,JNEARWALL,U,UG,UU,RHO
     2,RHOU,ALPHA
     2,ALPHAU,RINTW,RINTE,RINT3M,PE,PW,TAUWALLD,RINT1N,RINT2N
     2,RE,GLOBX2MOM2ERR
     2,DMOMIN2,PFIN2,DMOMOUT2,DMOMOUT2NEAR,PFOUT2,FTAUWALL2
     2,PFTOP2,C1IFD,RINT4N)
      WRITE(*,*)'GLOBX2MOM2ERR = ',GLOBX2MOM2ERR
C
      TOTFLOW=0.0D0
      TOTAREA=0.0D0
      DO 80 J=2,N-1
       RNORTH=R(J)+1.0D0/(DBLE(N-2)*2.0D0)
       RSOUTH=R(J)-1.0D0/(DBLE(N-2)*2.0D0)
       AREA=.5D0*(RNORTH**2-RSOUTH**2)
       FLOW=AREA*U(1,J)
       TOTFLOW=TOTFLOW+FLOW
       TOTAREA=TOTAREA+AREA
80    CONTINUE
      UAVE=TOTFLOW/TOTAREA
C
C WRITE OUTPUT
      IF(I.EQ.2)THEN
```

```
        DO 83 J=1,N
        DMDOTVAP=RINTE(J)*RHOU(2,J)*ALPHAU(2,J)*UU(2,J)
        DMDOTLIQ=RINTE(J)*RHOU(1,J)*ALPHAU(1,J)*UU(1,J)
        IF((DMDOTLIQ+DMDOTVAP).EQ.0.0D0)THEN
         QUALITY=0.0D0
        ELSE
         QUALITY=DMDOTVAP/(DMDOTVAP+DMDOTLIQ)
        ENDIF
C
        WRITE(14,70)I-1,J,R(J),X-DELX,UU(1,J),UU(2,J),RTMINUSRPLUSCHECK
        WRITE(15,70)I-1,J,WALLTHICK,JETTHICK,CORETHICK,ALPHAU(2,J)
      2,QUALITY
        WRITE(16,70)I-1,J,VISCU(1,J),0.0D0,VISCU(2,J),VU(1,J),VU(2,J)
        WRITE(17,70)I,J,0.0D0,0.0D0,0.0D0,0.0D0,0.0D0
83      CONTINUE
        WRITE(18,*)X-DELX,PW
       ENDIF
C
C FINDS QUALITY IN EACH CONTROL VOLUME (FOR TESTING PURPOSES)
C AND WRITES OUTPUT TO FILE; ICCONST CAN BE 1,2,4,8 ETC.
C DEPENDING ON FINENESS
       ICCONST=1
       DO 82 ICOARSE=2,(M-2)/ICCONST+1
        IFINE=ICCONST*(ICOARSE-1)+1
        IF(I.EQ.IFINE)THEN

         DO 81 J=1,N
          DMDOTVAP=RINTE(J)*RHO(2,J)*ALPHA(2,J)*U(2,J)
          DMDOTLIQ=RINTE(J)*RHO(1,J)*ALPHA(1,J)*U(1,J)
          IF((DMDOTLIQ+DMDOTVAP).EQ.0.0D0)THEN
           QUALITY=0.0D0
          ELSE
           QUALITY=DMDOTVAP/(DMDOTVAP+DMDOTLIQ)
          ENDIF
          WRITE(14,70)I,J,R(J),X,U(1,J),U(2,J),RTMINUSRPLUSCHECK
          WRITE(15,70)I,J,WALLTHICK,JETTHICK,CORETHICK,ALPHA(2,J),QUALITY
          WRITE(16,70)I,J,VISC(1,J),DELPR(J),VISC(2,J),V(1,J),V(2,J)
          WRITE(17,70)I,J,PE,C1IFD(J),LMIX(J),TAUWALL,RNEARWALL
81       CONTINUE
         WRITE(18,*)X,PE

        ENDIF

82     CONTINUE
C REMEMBER SOME VALUES FOR GLOBAL X-MOM. CHECK.
       CALL GLOBXREM(N,TAUWALLC,TAUWALLD,RINT4N,ALPHA,RE
      2,PW,PE,RINTE,RINTW,RINT3M,C1IFD,U,TOTFTAU2,TOTFTAU1,TOTPF2IN
      2,TOTPF2OUT,TOTPF1IN,TOTPF1OUT,TOTFINT1,TOTFINT2)
C FIND THE TOTAL INLET MOMENTUM FLUX IF I=2
       IF(I.EQ.2)THEN
        TOTMOMIN2=0.0D0
        TOTMOMIN1=0.0D0
        DO 3000 J=2,N-1
          TOTMOMIN2=TOTMOMIN2+RHOU(2,J)*RINTW(J)*ALPHAU(2,J)*UU(2,J)**2
          TOTMOMIN1=TOTMOMIN1+RHOU(1,J)*RINTW(J)*ALPHAU(1,J)*UU(1,J)**2
3000    CONTINUE
```

```
      ENDIF
C
      CALL VELUPDATE2(U,UU,UG,V,VU,VG,RHO,RHOU,VISC,VISCU,VISCL,VISCLU
     2,VISCT,VISCTU,ALPHA,ALPHAU
     2,N,PW,PE)
C UPDATE THICKNESSES
      CORETHICKOLD=CORETHICK
      JCORETHICKOLD=JCORETHICK
      WALLTHICKOLD=WALLTHICK
      JWALLTHICKOLD=JWALLTHICK
      JETTHICKOLD=JETTHICK
      JJETTHICKOLD=JJETTHICK
C
C IF AT END OF DUCT STOP PROGRAM
      IF(I.EQ.M-1)GO TO 50
C     TAKE STEP DOWNSTREAM AND CONTINUE CALCULATION
      X=X+DELX
      I=I+1
      CALL GEOMCOEFF(N,R,DELR,X,DELX,RINT1S,RINT1N,RINT2S,RINT2N
     2,RINT4S,RINT4N,RINT6S,RINT6N,RINT3M,RINTE,RINTW,RINT5,RDXM,RTXE
     2,IGOWALL,RDXW,RDXE)
      GO TO 2
50    CONTINUE
C FIND THE TOTAL OUTLET MOMENTUM FLUX
      TOTMOMOUT1=0.0D0
      TOTMOMOUT2=0.0D0
      DO 2000 J=2,N-1
       TOTMOMOUT1=TOTMOMOUT1+RHO(1,J)*ALPHA(1,J)*RINTE(J)*U(1,J)**2
       TOTMOMOUT2=TOTMOMOUT2+RHO(2,J)*ALPHA(2,J)*RINTE(J)*U(2,J)**2
2000  CONTINUE
C FIND THE GLOBAL MOMENTUM ERROR
      TOTERRMOM1=((TOTMOMOUT1+TOTPF1OUT-TOTPF1IN+TOTFTAU1-TOTFINT1)
     2-TOTMOMIN1)/TOTMOMIN1
      TOTERRMOM2=((TOTMOMOUT2+TOTPF2OUT-TOTPF2IN+TOTFTAU2-TOTFINT2)
     2-TOTMOMIN2)/TOTMOMIN2
      WRITE(*,*)'TOTMOMIN2 = ',TOTMOMIN2
      WRITE(*,*)'TOTMOMOUT2 = ',TOTMOMOUT2
      WRITE(*,*)'TOTPF2IN = ',TOTPF2IN
      WRITE(*,*)'TOTPF2OUT = ',TOTPF2OUT
      WRITE(*,*)'TOTFTAU2 = ',TOTFTAU2
      WRITE(*,*)'TOTFINT2 = ',TOTFINT2
      WRITE(*,*)'TOTERRMOM2 = ',TOTERRMOM2
      WRITE(*,*)
C
      WRITE(*,*)'TOTMOMIN1 = ',TOTMOMIN1
      WRITE(*,*)'TOTMOMOUT1 = ',TOTMOMOUT1
      WRITE(*,*)'TOTPF1IN = ',TOTPF1IN
      WRITE(*,*)'TOTPF1OUT = ',TOTPF1OUT
      WRITE(*,*)'TOTFTAU1 = ',TOTFTAU1
      WRITE(*,*)'TOTFINT1 = ',TOTFINT1
      WRITE(*,*)'TOTERRMOM1 = ',TOTERRMOM1
C
      CALL GLOBCONT(I,M,N,ALPHA,ALPHAU,RHO,RHOU,U,UU,RINTW,RINTE
     2,FLOWINTOT)
C
      CALL CALCFLOWOUT(I,M,N,ALPHA,RHO,U,RINTE,FLOWOUT1,FLOWOUT2)
```

```
      ERRFLOW1=(FLOWOUT1-TOTFLOW1)/TOTFLOW1
      ERRFLOW2=(FLOWOUT2-TOTFLOW2)/TOTFLOW2
      WRITE(*,*)'FLOWIN1,FLOWOUT1 = ',TOTFLOW1,FLOWOUT1
      WRITE(*,*)'ERRFLOW1 = ',ERRFLOW1
      WRITE(*,*)'FLOWIN2,FLOWOUT2 = ',TOTFLOW2,FLOWOUT2
      WRITE(*,*)'ERRFLOW2 = ',ERRFLOW2
C
      STOP
      END
C
C
C
C REMEMBERS SOME VALUES FOR GLOBAL X-MOM. CHECK
      SUBROUTINE GLOBXREM(N,TAUWALLC,TAUWALLD,RINT4N,ALPHA,RE
     2,PW,PE,RINTE,RINTW,RINT3M,C1IFD,U,TOTFTAU2,TOTFTAU1,TOTPF2IN
     2,TOTPF2OUT,TOTPF1IN,TOTPF1OUT,TOTFINT1,TOTFINT2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION RINT4N(2000),ALPHA(2,2000),RINTE(2000),RINTW(2000)
     2,RINT3M(2000),C1IFD(2000),U(2,2000)
C     WALL SHEAR
      TOTFTAU2=TOTFTAU2+TAUWALLC*RINT4N(N-1)*ALPHA(2,N-1)/RE
      TOTFTAU1=TOTFTAU1+TAUWALLD*RINT4N(N-1)*ALPHA(1,N-1)/RE
      DO 100 J=2,N-1
C      PRESSURE FORCES
       TOTPF2IN=TOTPF2IN+PW*ALPHA(2,J)*RINTW(J)
       TOTPF2OUT=TOTPF2OUT+PE*ALPHA(2,J)*RINTE(J)
       TOTPF1IN=TOTPF1IN+PW*ALPHA(1,J)*RINTW(J)
       TOTPF1OUT=TOTPF1OUT+PE*ALPHA(1,J)*RINTE(J)
C      INTERFACIAL FORCES
       TOTFINT1=TOTFINT1+RINT3M(J)*C1IFD(J)*(U(2,J)-U(1,J))
       TOTFINT2=TOTFINT2+RINT3M(J)*C1IFD(J)*(U(1,J)-U(2,J))
100   CONTINUE
      RETURN
      END
C
C
C
C FINDS NEW WALL SHEAR STRESS (TAUWALL) FOR WALL FUNCTION APPROX.
      SUBROUTINE FINDNEWTAUWALL(IPH,N,JNEARWALL,KTAUINTERP
     2,GLOBX1MOM1ERR,TAUWALL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     FIND NEW TAUWALL BY LINEAR INTERPOLATION
      TAUINTERPFACTOR=.1D0
      IF(KTAUINTERP.EQ.1)THEN
       TAUWALL1ST=TAUWALL
       GLOBX1MOM11ST=GLOBX1MOM1ERR
       KTAUINTERP=KTAUINTERP+1
       TAUWALL=TAUWALL*(1.0D0+TAUINTERPFACTOR)
       GO TO 21
      ENDIF
      IF(KTAUINTERP.EQ.2)THEN
       TAUWALL2ND=TAUWALL
       GLOBX1MOM12ND=GLOBX1MOM1ERR
       KTAUINTERP=KTAUINTERP+1
       TAUWALL=((TAUWALL2ND-TAUWALL1ST)/(GLOBX1MOM12ND-GLOBX1MOM11ST))
     2         *(-GLOBX1MOM12ND)+TAUWALL2ND
```

```
        GO TO 21
       ENDIF
       IF(KTAUINTERP.GE.3)THEN
        TAUWALL3RD=TAUWALL
        GLOBX1MOM13RD=GLOBX1MOM1ERR
        IF(GLOBX1MOM13RD*GLOBX1MOM12ND.LT.0.0D0)THEN
         TAUWALL1ST=TAUWALL2ND
         GLOBX1MOM11ST=GLOBX1MOM12ND
         TAUWALL2ND=TAUWALL3RD
         GLOBX1MOM12ND=GLOBX1MOM13RD
         TAUWALL=((TAUWALL2ND-TAUWALL1ST)/(GLOBX1MOM12ND-GLOBX1MOM11ST))
      2          *(-GLOBX1MOM12ND)+TAUWALL2ND
         GO TO 21
        ELSEIF(GLOBX1MOM13RD*GLOBX1MOM11ST.LT.0.0D0)THEN
         TAUWALL2ND=TAUWALL3RD
         GLOBX1MOM12ND=GLOBX1MOM13RD
         TAUWALL=((TAUWALL2ND-TAUWALL1ST)/(GLOBX1MOM12ND-GLOBX1MOM11ST))
      2          *(-GLOBX1MOM12ND)+TAUWALL2ND
         GO TO 21
        ELSE
         TAUWALL1ST=TAUWALL2ND
         GLOBX1MOM11ST=GLOBX1MOM12ND
         TAUWALL2ND=TAUWALL3RD
         GLOBX1MOM12ND=GLOBX1MOM13RD
         TAUWALL=((TAUWALL2ND-TAUWALL1ST)/(GLOBX1MOM12ND-GLOBX1MOM11ST))
      2          *(-GLOBX1MOM12ND)+TAUWALL2ND
         GO TO 21
        ENDIF
       ENDIF
       WRITE(*,*)'SOMETHING WENT WRONG IN TAUWALL ITERATION'
       STOP
21     CONTINUE
C      ENSURE THAT TAUWALL NEVER FALLS BELOW ZERO
       IF(TAUWALL.LE.0.0D0)TAUWALL=1.0D0
       RETURN
       END
C
C
C
C CHECKS GLOBAL CONSERVATION OF X-MOM. FOR CONT. PHASE TO DETERMINE IF
C WALL SHEAR (TAUWALL) WAS GUESSED CORRECTLY).
       SUBROUTINE GLOBX1MOM1ERROR(IPH,IPHOTHER,N,JNEARWALL,U,UG,UU,RHO
      2,RHOU,ALPHA
      2,ALPHAU,RINTW,RINTE,RINT3M,PE,PW,TAUWALL,RINT1N,RINT2N
      2,RE,GLOBX1MOM1ERR
      2,DMOMIN1,PFIN1,DMOMOUT1,DMOMOUT1NEAR,PFOUT1,FTAUWALL1
      2,PFTOP1,C1IFD,RINT4N)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION R(2000),U(2,2000),UU(2,2000),RHO(2,2000),RHOU(2,2000)
      2,ALPHA(2,2000),ALPHAU(2,2000),RINTW(2000),RINTE(2000),RINT1N(2000)
      2,RINT2N(2000),RINT3M(2000),C1IFD(2000),UG(2,2000),RINT4N(2000)
       DMOMIN1=0.0D0
       DMOMOUT1=0.0D0
       PFIN1=0.0D0
       PFOUT1=0.0D0
       PFTOP1=0.0D0
```

```
      FCHECK1=0.0D0
      FINT1=0.0D0
      DO 26 J=2,N-1
       DMOMIN1=DMOMIN1+RHOU(IPH,J)*ALPHAU(IPH,J)*RINTW(J)*UU(IPH,J)**2
       PFIN1=PFIN1+PW*ALPHA(IPH,J)*RINTW(J)
       DMOMOUT1=DMOMOUT1+RHO(IPH,J)*ALPHA(IPH,J)*RINTE(J)*U(IPH,J)**2
       FCHECK1=FCHECK1+RHO(IPH,J)*ALPHA(IPH,J)*RINTE(J)*U(IPH,J)
       PFOUT1=PFOUT1+PE*ALPHA(IPH,J)*RINTE(J)
       FTAUWALL1=TAUWALL*RINT1N(N-1)*ALPHA(IPH,N-1)/RE
       FTAUWALLCHECK=TAUWALL*RINT4N(N-1)*ALPHA(IPH,N-1)/RE
       FINT1=FINT1+RINT3M(J)*C1IFD(J)*(UG(IPHOTHER,J)-U(IPH,J))
26    CONTINUE
      DMOMOUT1NEAR=0.0D0
      DO 500 J=JNEARWALL,N-1
       DMOMOUT1NEAR=DMOMOUT1NEAR+RHO(IPH,J)*ALPHA(IPH,J)*RINTE(J)
     2              *U(IPH,J)**2
500   CONTINUE
      PFTOP1=.5D0*ALPHA(IPH,N-1)*(PE+PW)*RINT2N(N-1)
      GLOBX1MOM1ERR=DMOMOUT1-DMOMIN1+PFOUT1-PFIN1-PFTOP1+FTAUWALL1-FINT1
      RETURN
      END
C
C
C
C FINDS A POINT WITHIN WALL FUNCTION REGION WHICH CAN BE USED AS A
C BOUNDARY
C SUBROUTINE TO FIND JNEARWALL WITHIN THE RANGE YPLUS=30-200
C DOESN'T NECESSARILY HAVE TO BE R(N-1)
C THE SEARCH RANGE MAY BE TIGHTER THAN 30-200 TO GET A MIDDLE POINT
      SUBROUTINE FINDJNEARWALL(IPH,N,RE,RDXW,R,RHO,TAUWALL,RTMINUSRPLUS
     2,RNEARWALL,RLOW,RHIGH,JNEARWALL,VISCL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RHO(2,2000),VISCL(2,2000)
5     CONTINUE
      DO 10 J=N-1,2,-1
      RTMINUSRPLUS=RE**.5D0*RDXW*(1.0D0-R(J))
     2*(RHO(IPH,N-1)*TAUWALL)**.5D0/VISCL(IPH,J)
      IF(RTMINUSRPLUS.GE.RLOW.AND.RTMINUSRPLUS.LE.RHIGH)THEN
       RNEARWALL=R(J)
       JNEARWALL=J
       GO TO 20
      ENDIF
10    CONTINUE
      IF(RLOW.EQ.30.0D0.AND.RHIGH.EQ.200.0D0)THEN
      RNEARWALL=R(N-1)
      JNEARWALL=N-1
      RTMINUSRPLUS=RE**.5D0*RDXW*(1.0D0-R(N-1))
     2*(RHO(IPH,N-1)*TAUWALL)**.5D0/VISCL(IPH,N-1)
      GO TO 20
      ENDIF
C
      RLOW=RLOW-10.0D0
      RHIGH=RHIGH+10.0D0
      IF(RLOW.LE.30.0D0)RLOW=30.0D0
      IF(RHIGH.GE.200.0D0)RHIGH=200.0D0
      GO TO 5
```

```
20    CONTINUE
      RETURN
      END
C
C
C FINDS THE VELOCITY AT THE NEAR WALL POINT (JNEARWALL) FROM THE
C WALL FUNCTION.  FINDS THE VELOCITY PROFILE BETWEEN THIS POINT AND
C THE WALL ASSUMING A 1/7 POWER LAW PROFILE
      SUBROUTINE FINDNEARWALLPROFILE(IPH,N,JNEARWALL,KAPPA,B,RE,RDXW
     2,TAUWALL,RHO,R,VISCL,U,UG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION KAPPA
      DIMENSION R(2000),UG(2,2000),RHO(2,2000),U(2,2000),VISCL(2,2000)
      IF(TAUWALL.LE.0.0D0)THEN
       GO TO 400
      ENDIF
      UG(IPH,JNEARWALL)=((TAUWALL/RHO(IPH,JNEARWALL)/RE)**.5/KAPPA)
     2*LOG(RE**.5*RDXW*(TAUWALL/RHO(IPH,JNEARWALL))**.5
     2*(1.0D0-R(JNEARWALL))*RHO(IPH,JNEARWALL)/VISCL(IPH,JNEARWALL))
     2+B*(TAUWALL/RHO(IPH,JNEARWALL)/RE)**.5
      U(IPH,JNEARWALL)=UG(IPH,JNEARWALL)
C     GET REST OF VELOCITY POINTS ABOVE JNEARWALL
C     BUT BELOW RTMINUSRPLUS=30
      IF(JNEARWALL.EQ.(N-1))GO TO 400
C     GET REST OF VELOCITY POINTS ABOVE RTMINUSRPLUS=30 (TO WALL)
C     ASSUMING A 1/7 POWER LAW PROFILE
      DO 420 J=JNEARWALL+1,N-1
       UG(IPH,J)=(((1.0D0-R(J))/(1.0D0-R(JNEARWALL)))**(1.0D0/7.0D0))
     2          *U(IPH,JNEARWALL)
       U(IPH,J)=UG(IPH,J)
420   CONTINUE
C
400   CONTINUE
C
      RETURN
      END
C
C
C
C
C CALCULATES TURBULENT CONTRIBUTION TO VISCOSITY FOR JET FLOW
C AND DOWNSTREAM.  USES SIMPLEST POSSIBLE MIXING LENGTH MODEL
      SUBROUTINE TURBVISC(I,N,IPH,DELX,X,R,RDXM,DELR
     2,RE,U,UU,DELPX,RHO,RHOU,VISCL
     2,VISCLU,VISCT,VISCTU,VISC,VISCU,RCO,LINNERRCO,LOUTERRCO,WALLTHICK
     2,JETTHICK,CORETHICK,IMERGE,LMIX,RCOD,JRCOD
     2,CORETHICKOLD,JCORETHICKOLD,WALLTHICKOLD,JWALLTHICKOLD
     2,JETTHICKOLD,JJETTHICKOLD,JWALLTHICK,JJETTHICK,JCORETHICK,KACCESS)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DOUBLE PRECISION LMIX(2000),KAPPA,LINNERRCO,LOUTERRCO
     2,JETTHICK,KAPPA2,KAPPANOT,LMIXMERGE(2000),JETTHICKOLD
C
      DIMENSION U(2,2000),UU(2,2000),RHO(2,2000),R(2000),VISCT(2,2000)
     2,VISCL(2,2000),VISCLU(2,2000),VISC(2,2000),VISCU(2,2000)
```

```
     2,VISCTU(2,2000)
     2,RHOU(2,2000)
C SOME CONSTANTS FOR TURBULENCE MODELS
      IF(I.LE.IMERGE)THEN
       KAPPA=.6D0
C       KAPPA=0.01D0
       A=26.0D0
C       KAPPANOT=.08D0
       KAPPANOT=.12D0
      ELSE
       KAPPA=.6D0
C       KAPPA=0.01D0
       A=26.0D0
C       KAPPANOT=.08D0
       KAPPANOT=.12D0
      ENDIF
C      COUTERWALL=.09D0
      COUTERWALL=.12D0
      CALL FINDRCO(IPH,N,KAPPA,WALLTHICK,R,RCO,RCOD,JRCOD)
C REGIME 1
C      IF(I.LE.IMERGE)THEN
C TURBULENT VISCOSITY FOR CORE REGION
      DO 50 J=2,JCORETHICK
       VISCT(IPH,J)=0.0D0
       VISC(IPH,J)=VISCL(IPH,J)+VISCT(IPH,J)
50    CONTINUE
C TURBULENT VISCOSITY FOR JET REGION
      DO 200 J=JCORETHICK+1,JJETTHICK
C      DO 200 J=2,JJETTHICK
       UN=.5D0*(U(IPH,J+1)+U(IPH,J))
       US=.5D0*(U(IPH,J)+U(IPH,J-1))
C       WRITE(*,*)U(IPH,N),U(IPH,1),U(IPH,2)
       IF(J.EQ.N-1)UN=U(IPH,N)
       IF(J.EQ.2)US=U(IPH,1)
       DUDR=(UN-US)/DELR
       LMIX(J)=KAPPANOT*(JETTHICK-CORETHICK)
       VISCT(IPH,J)=RE*RHO(IPH,J)*LMIX(J)**2*DABS(DUDR)/RDXM
C SET VISCT TO ZERO FOR TESTING PURPOSES
C       VISCT(IPH,J)=0.0D0
C
       VISC(IPH,J)=VISCL(IPH,J)+VISCT(IPH,J)
200   CONTINUE
C
C TURBULENT VISCOSITY FOR WALL REGION (START AT JJETTHICK+1
C TO INCORPORATE INVISCID REGION INTO WALL REGION)
C
      DO 100 J=JJETTHICK+1,N-1
       IF(I.EQ.2)THEN
        VISCTU(IPH,J)=0.0D0
        VISCU(IPH,J)=VISCTU(IPH,J)+VISCLU(IPH,J)
       ENDIF
       UN=.5D0*(U(IPH,J+1)+U(IPH,J))
       US=.5D0*(U(IPH,J)+U(IPH,J-1))
       IF(J.EQ.N-1)UN=U(IPH,N)
       IF(J.EQ.2)US=U(IPH,1)
       DUDR=(UN-US)/DELR
```

```
       DUDRWALL=(U(IPH,N)-U(IPH,N-1))/(.5D0*DELR)
C FIND MIXING LENGTHS FOR INNER AND OUTER LAYERS AND INVISCID CORE
C      INNER LAYER
       IF(R(J).GT.RCO)THEN
        C1=DABS((-RE*RHO(IPH,J)*RDXM*DUDRWALL/(VISCL(IPH,N-1))))**.5D0

        C2=-(1.0D0-R(J))/A
        C3=1.0D0-DEXP(C2*C1)
        LMIX(J)=KAPPA*(1.0D0-R(J))*RDXM*C3
C      OUTER LAYER
       ELSE
        LMIX(J)=COUTERWALL*(1.0D0-WALLTHICK)*RDXM
       ENDIF
C
       VISCT(IPH,J)=RE*RHO(IPH,J)*LMIX(J)**2*DABS(DUDR)/RDXM
C
       VISC(IPH,J)=VISCL(IPH,J)+VISCT(IPH,J)
100   CONTINUE
C LINEAR FIT FOR JET TO CORE LAYER BEFORE MERGE
C   TAKING 10% OF JET LAYER AND MAKING IT LINEARLY FIT WITH CORE
      IF(I.LT.IMERGE)THEN
       LLAYER=.1D0*JJETTHICK
       LLAYER=JJETTHICK-LLAYER
       DO 900 J=LLAYER,JJETTHICK
        UN=.5D0*(U(IPH,J+1)+U(IPH,J))
        US=.5D0*(U(IPH,J)+U(IPH,J-1))
        IF(J.EQ.N-1)UN=U(IPH,N)
        IF(J.EQ.2)US=U(IPH,1)
        DUDR=(UN-US)/DELR
        SL=(LMIX(LLAYER)-LMIX(JJETTHICK+1))
     2     /(R(LLAYER)-R(JJETTHICK+1))
        YI=LMIX(LLAYER)-SL*R(LLAYER)
        LMIX(J)=SL*R(J)+YI
        VISCT(IPH,J)=RE*RHO(IPH,J)*LMIX(J)**2*DABS(DUDR)/RDXM

        VISC(IPH,J)=VISCL(IPH,J)+VISCT(IPH,J)
900    CONTINUE
      ENDIF
C LINEAR FIT FOR OUTER WALL LAYER AFTER MERGE
      IF(I.GE.IMERGE)THEN
       DO 800 J=JJETTHICK+1,JRCOD-1
          UN=.5D0*(U(IPH,J+1)+U(IPH,J))
          US=.5D0*(U(IPH,J)+U(IPH,J-1))
          IF(J.EQ.N-1)UN=U(IPH,N)
          IF(J.EQ.2)US=U(IPH,1)
          DUDR=(UN-US)/DELR
          SL=(LMIX(JRCOD)-LMIX(JJETTHICK))
     2       /(R(JRCOD)-R(JJETTHICK))
          YI=LMIX(JJETTHICK)-SL*R(JJETTHICK)
          LMIX(J)=SL*R(J)+YI
          VISCT(IPH,J)=RE*RHO(IPH,J)*LMIX(J)**2*DABS(DUDR)/RDXM
C
          VISC(IPH,J)=VISCL(IPH,J)+VISCT(IPH,J)
800    CONTINUE
      ENDIF
C REMEMBER VALUES OF LMIX(J) AT I=IMERGE
```

```
      IF(I.EQ.IMERGE)THEN
       DO 300 J=2,N-1
        LMIXMERGE(J)=LMIX(J)
300    CONTINUE
      ENDIF
C
C SET THE VALUE OF THE TURBULENT VISCOSITY NEXT TO THE BOUNDARY TO ZERO
C THIS IS DONE TO SIMULATE THE FACT THAT VERY CLOSE TO THE WALL, THE
C LAMINAR VISCOSITY IS DOMINANT OVER THE TURBULENT VISCOSITY.  SINCE
C IT IS THIS VISCOSITY THAT IS IMPORTANT IN CALCULATING WALL SHEAR, AND
C HENCE PRESSURE RISE, ITS TURBULENT VALUE MUST BE SET TO ZERO.
C      VISCT(IPH,N-1)=0.0D0
      VISC(IPH,N-1)=VISCL(IPH,N-1)+VISCT(IPH,N-1)
C
C VALUES OF TURBULENT VISCOSITY AND TOTAL VISCOSITY AT THE BOUNDARIES
      VISCT(IPH,1)=VISCT(IPH,2)
      VISCT(IPH,N)=VISCT(IPH,N-1)
      VISC(IPH,1)=VISC(IPH,2)
      VISC(IPH,N)=VISC(IPH,N-1)
C SET TURBULENT VISCOSITY TO ZERO FOR TESTING PURPOSES
C      DO 1000 J=2,N-1
C       VISCT(IPH,J)=0.0D0
C       VISC(IPH,J)=VISCT(IPH,J)+VISCL(IPH,J)
C1000  CONTINUE
      RETURN
      END
C
C
C
C FINDS CROSSOVER POINT BETWEEN OUTER AND WALL LAYER
      SUBROUTINE FINDRCO(IPH,N,KAPPA,WALLTHICK,R,RCO,RCOD,JRCOD)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION KAPPA
      DIMENSION R(2000)
      IOVER=1
      RCO=1.0D0-(.09D0/KAPPA)*(1.0D0-WALLTHICK)
      DO 100 J=1,N
       IF(R(J).GE.RCO)THEN
         IF(IOVER.EQ.1)THEN
           RCOD=R(J)
           JRCOD=J
           IOVER=2
         ENDIF
       ENDIF
100   CONTINUE
      END
C
C
C
C FINDS THICKNESSES OF JET, WALL, AND CORE LAYERS TO BE USED IN
C MIXING LENGTH TURBULENCE MODEL.
      SUBROUTINE THICK2(N,I,IPH,R,U,DELR,WALLTHICK,JETTHICK,CORETHICK
     2,JWALLTHICK,JJETTHICK,JCORETHICK,IMERGE
     2,CORETHICKOLD,JCORETHICKOLD,WALLTHICKOLD,JWALLTHICKOLD
     2,JETTHICKOLD,JJETTHICKOLD,KACCESS
     2,JCORETHICKPREV,JWALLTHICKPREV,JJETTHICKPREV)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION JETTHICK,JETTHICKOLD
      DIMENSION U(2,2000),R(2000)
      EPS=.1D0
C
      IF(KACCESS.EQ.0)THEN
       KACCESS=KACCESS+1
       JJETTHICK=JJETTHICKOLD
       JETTHICK=JETTHICKOLD
       JCORETHICK=JCORETHICKOLD
       CORETHICK=CORETHICKOLD
       JWALLTHICK=JWALLTHICKOLD
       WALLTHICK=WALLTHICKOLD
       RETURN
      ENDIF
C
      KACCESS=KACCESS+1
C FIND INVISCID CORE THICKNESS
      DO 100 J=2,JCORETHICKOLD
       DELU=DABS((U(IPH,J+1)-U(IPH,J))/(R(J+1)-R(J)))
       IF(DELU.GT.0.1D0)THEN
        CORETHICK=R(J)
        JCORETHICK=J
        GO TO 200
       ENDIF
100   CONTINUE
C
200   CONTINUE
C FIX CORE THICKNESS AT J=2 FROM BEGINNING
      JCORETHICK=2
      CORETHICK=R(JCORETHICK)
C IF THE CURRENT POSITION IS AFTER THE JET AND WALL LAYERS HAVE MERGED,
C USE THE OLD VALUES OF JETTHICK AND WALLTHICK;
C IE JETTHICK AND WALLTHICK DON'T MOVE AFTER THIS POINT.
      IF(I.GT.IMERGE)THEN
       WALLTHICK=WALLTHICKOLD
       JWALLTHICK=JWALLTHICKOLD
       JETTHICK=JETTHICKOLD
       JJETTHICK=JJETTHICKOLD
       GO TO 600
      ENDIF
C
C FIND WALLTHICKNESS
      DO 300 J=JWALLTHICKOLD,JJETTHICKOLD,-1
       DELU=DABS((U(IPH,J)-U(IPH,J-1))/(R(J)-R(J-1)))

       IF(DELU.LT.EPS)THEN
        WALLTHICK=R(J)
        JWALLTHICK=J
        GO TO 400
       ENDIF
300   CONTINUE
C
C IF ROUTINE NEVER FINDS THE LOW SLOPE WE KNOW THE LAYERS HAVE MERGED
      WALLTHICK=WALLTHICKOLD
      JWALLTHICK=JWALLTHICKOLD
```

```
C
C THE NEXT TWO STATEMENTS FORCE THE JET THICKNESS
C TO BE ADJACENT TO THE WALL THICKNESS
      JETTHICK=WALLTHICK-DELR
      JJETTHICK=JWALLTHICK-1
      IMERGE=I
      GO TO 600
C
400   CONTINUE
C FIND JETTHICKNESS
      DO 500 J=JJETTHICKOLD,JWALLTHICK
       DELU=DABS((U(IPH,J+1)-U(IPH,J))/(R(J+1)-R(J)))
       IF(DELU.LT.EPS)THEN
        JETTHICK=R(J)
        JJETTHICK=J
        GO TO 600
       ENDIF
500   CONTINUE
C
600   CONTINUE
C PREVENT THICKNESSES FROM GOING DOWN
      IF(KACCESS.GT.1)THEN
       IF(JCORETHICK.GT.JCORETHICKPREV)JCORETHICK=JCORETHICKPREV
       IF(JJETTHICK.LT.JJETTHICKPREV)JJETTHICK=JJETTHICKPREV
       IF(JWALLTHICK.GT.JWALLTHICKPREV)JWALLTHICK=JWALLTHICKPREV
       CORETHICK=R(JCORETHICK)
       JETTHICK=R(JJETTHICK)
       WALLTHICK=R(JWALLTHICK)
      ENDIF
C
C IF JETTHICK TURNS OUT GREATER THAN WALLTHICK, MAKE IT LOWER ADJACENT
      IF(JJETTHICK.GE.JWALLTHICK)THEN
       JJETTHICK=JWALLTHICK-1
       JETTHICK=R(JJETTHICK)
      ENDIF
C
      RETURN
      END
C
C
C
C
C CALCULATES THE ERROR IN THE GLOBAL MASS BALANCE
      SUBROUTINE GLOBCONT(I,M,N,ALPHA,ALPHAU,RHO,RHOU,U,UU,RINTW,RINTE
     2,FLOWINTOT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ALPHA(2,2000),ALPHAU(2,2000),RHO(2,2000),RHOU(2,2000)
     2,U(2,2000),UU(2,2000),RINTW(2000),RINTE(2000)
      IF(I.EQ.2)THEN
       FLOWIN1=0.0D0
       FLOWIN2=0.0D0
       DO 100 J=2,N-1
        FLOWIN1=FLOWIN1+ALPHAU(1,J)*RHOU(1,J)*UU(1,J)*RINTW(J)
        FLOWIN2=FLOWIN2+ALPHAU(2,J)*RHOU(2,J)*UU(2,J)*RINTW(J)
100    CONTINUE
       FLOWINTOT=FLOWIN1+FLOWIN2
```

```
      ELSEIF(I.EQ.M-1)THEN
       FLOWOUT1=0.0D0
       FLOWOUT2=0.0D0
       DO 200 J=2,N-1
        FLOWOUT1=FLOWOUT1+ALPHA(1,J)*RHO(1,J)*U(1,J)*RINTE(J)
        FLOWOUT2=FLOWOUT2+ALPHA(2,J)*RHO(2,J)*U(2,J)*RINTE(J)
200    CONTINUE
       FLOWOUTTOT=FLOWOUT1+FLOWOUT2
       FLOWERROR=DABS((FLOWOUTTOT-FLOWINTOT)/FLOWINTOT)
      ENDIF
      RETURN
      END
C
C
C
C FINDS FLOWRATE EXITING DOMAIN
      SUBROUTINE CALCFLOWOUT(I,M,N,ALPHA,RHO,U,RINTE,FLOWOUT1,FLOWOUT2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ALPHA(2,2000),RHO(2,2000),U(2,2000),RINTE(2000)
      FLOWOUT1=0.0D0
      FLOWOUT2=0.0D0
      DO 100 J=2,N-1
       FLOWOUT1=FLOWOUT1+RINTE(J)*RHO(1,J)*ALPHA(1,J)*U(1,J)
       FLOWOUT2=FLOWOUT2+RINTE(J)*RHO(2,J)*ALPHA(2,J)*U(2,J)
100   CONTINUE
      RETURN
      END
C
C
C
C CALCULATES THE ERROR BETWEEN UG AND U, AND VG AND V
C FINDS A TOTAL ERROR WHICH INCLUDES ALL U'S AND V'S
      SUBROUTINE ERRORVEL2(I,N,IPH,U,UG,V,VG,ERRVEL2,ALPHA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(2,2000),UG(2,2000),V(2,2000),VG(2,2000)
     2,ALPHA(2,2000)
      ERRORTOT=0.0D0
      DO 100 J=2,N-1
       ERRORV=DABS((V(IPH,J)-VG(IPH,J))/V(IPH,J))
50     ERRORU=DABS((U(IPH,J)-UG(IPH,J))/U(IPH,J))
       ERRORTOT=ERRORTOT+ERRORU+ERRORV
100   CONTINUE
      ERRVEL2=ERRORTOT
      RETURN
      END
C
C
C UPDATES THE GUESS VELOCITY TO BE USED IN THE NEXT ITERATION
      SUBROUTINE UPDATEVEL2(I,IPH,N,U,UG,V,VG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(2,2000),UG(2,2000),V(2,2000),VG(2,2000)
C     INTERPOLATION FACTOR
      FI=.7D0
C      IF(I.EQ.10)FI=.1D0
C
      DO 100 J=2,N-1
```

```
         VG(IPH,J)=FI*V(IPH,J)+(1.0D0-FI)*VG(IPH,J)
         UG(IPH,J)=FI*U(IPH,J)+(1.0D0-FI)*UG(IPH,J)
100   CONTINUE
      RETURN
      END
C
C
C FINDS A NEW ALPHA (VOID FRACTION) TO USE FOR THE NEXT ITERATION
      SUBROUTINE UPDATEALPHA(N,I,IPH,IPHOTHER,ALPHANEW,ALPHA,ERRALPHA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ALPHA(2,2000),ALPHANEW(2,2000)
C     INTERPOLATION FACTOR
      FI=.5D0
C      FI=.99D0
      DO 100 J=2,N-1
       ALPHA(IPH,J)=FI*ALPHA(IPH,J)+(1.0D0-FI)*ALPHANEW(IPH,J)
       ALPHA(IPHOTHER,J)=1.0D0-ALPHA(IPH,J)
100   CONTINUE
      ALPHA(IPH,N)=ALPHA(IPH,N-1)
      ALPHA(IPHOTHER,N)=1.0D0-ALPHA(IPH,N)
      ALPHA(IPH,1)=ALPHA(IPH,2)
      ALPHA(IPHOTHER,1)=1.0D0-ALPHA(IPH,1)
      RETURN
      END
C
C
C CALCULATES THE ERROR IN ALPHA (VOID FRACTION); IE GUESSED VS.
C CALCULATED VOID FRACTION.
      SUBROUTINE ERRORALPHA(N,IPH,IPHOTHER,ALPHA,ALPHANEW,ERRALPHA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION ALPHA(2,2000),ALPHANEW(2,2000)
      ERRORTOT=0.0D0
      DO 100 J=2,N-1
       ALPHANEW(IPHOTHER,J)=1.0D0-ALPHANEW(IPH,J)
       ERROR=DABS(ALPHANEW(IPHOTHER,J)-ALPHA(IPHOTHER,J))
       ERRORTOT=ERRORTOT+ERROR
100   CONTINUE
      ERRALPHA=ERRORTOT/DBLE(N-2)
      RETURN
      END
C
C
C CALCULATES THE RADIUS OF THE BUBBLE OR DROPLET
      SUBROUTINE PARTICLESIZE(N,RDRAG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION RDRAG(2000)
      DO 100 J=2,N-1
C      RDRAG(J)=.000635D0
      RDRAG(J)=.002D0
C      RDRAG(J)=.002D0
100   CONTINUE
      WRITE(13,*)'RDRAG = .002'
      CLOSE(13)
      RETURN
      END
C
```

```
C
C
C DECIDES WHICH FLOW REGIME WE ARE IN AND SOME PARAMETERS BASED ON IT
C IREGIME=1 FOR BUBBLES IN LIQUID       NOTE:  IPH=1 FOR LIQUID
C IREGIME=2 FOR DROPLETS IN VAPOR              IPH=2 FOR VAPOR
      SUBROUTINE REGIME(IREGIME,IPHCONT,IPHDISC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IF(IREGIME.EQ.1)THEN
       IPHCONT=1
       IPHDISC=2
      ELSEIF(IREGIME.EQ.2)THEN
       IPHCONT=2
       IPHDISC=1
      ENDIF
      RETURN
      END
C
C
C
C CALCULATES A COEFFICIENT WHICH IS USED TO CALCULATE THE
C INTERFACIAL DRAG FORCE
C SEE ISHII, P. 112 ("TWO-FLUID MODEL AND HYDRODYNAMIC
C CONSTITUTIVE EQUATIONS",
C NUCLEAR ENGINEERING AND DESIGN 82 (1984), PP. 107-126)
      SUBROUTINE INTDRAG(I,N,IREGIME,IPHCONT,IPHDISC,UG,UU,VG,VU,RHO
     2,VISCL
     2,ALPHA,ALPHAU,ALPHAVAPORMN,RDRAG,RE,C1IFD)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION UG(2,2000),UU(2,2000),VG(2,2000),VU(2,2000),RHO(2,2000)
     2,VISCL(2,2000),ALPHA(2,2000),RDRAG(2000),C1IFD(2000)
     2,ALPHAU(2,2000)
C
      DO 100 J=2,N-1
C
C      AVERAGE VELOCITIES AT THE MIDDLE OF THE CV FOR EACH PHASE
       U1=.5D0*(UG(1,J)+UU(1,J))
       U2=.5D0*(UG(2,J)+UU(2,J))
       V1=VG(1,J)
       V2=.5D0*(VG(2,J)+VG(2,J-1))
C
C      MAGNITUDE OF THE RELATIVE VELOCITY VECTOR BETWEEN THE PHASES

       VREL=DSQRT((U1-U2)**2+(V1-V2)**2)
C
       IF(IREGIME.EQ.1)VISCM=VISCL(IPHCONT,J)/(1.0D0-ALPHA(IPHDISC,J))
       IF(IREGIME.EQ.2)VISCM=VISCL(IPHCONT,J)/(1.0D0-ALPHA(IPHDISC,J))
     2 **2.5D0
      ALPHATERM=1.0D0-ALPHAVAPORMN
C      IF(ALPHATERM.LT.(1.0D0-ALPHAVAPORMN))ALPHATERM=1.0D0-ALPHAVAPORMN
C      IF(ALPHATERM.LT.1.0D-4)ALPHATERM=1.0D0-ALPHAVAPORMN
      C1=2.0D0*RE*RDRAG(J)*RHO(IPHCONT,J)/VISCM
C      C1=2.0D0*RE*RDRAGTEST*RHO(IPHCONT,J)/VISCM
C      C2=(3.0D0/8.0D0)*ALPHA(IPHDISC,J)*RHO(IPHCONT,J)/RDRAG(J)
C      C2=(3.0D0/8.0D0)*ALPHA(IPHDISC,J)*RHO(IPHCONT,J)/RDRAGTEST
```

```
C        C2=(3.0D0/8.0D0)*.023D0*RHO(IPHCONT,J)/RDRAG(J)
      C2=(3.0D0/8.0D0)*ALPHATERM*RHO(IPHCONT,J)/RDRAG(J)
      C1IFD(J)=24.0D0*C2/C1+(2.4*C2/C1**.25D0)*VREL**.75D0
50    CONTINUE
100   CONTINUE
C
      RETURN
      END
C
C
C CALCULATES THE INTERFACIAL MASS TRANSFER
C (GAM) DUE TO EVAPORATION OR CONDENSATION
      SUBROUTINE INTMASS(N,GAM)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION GAM(2,2000)
      DO 100 J=2,N-1
      GAM(1,J)=0.0D0
      GAM(2,J)=-GAM(1,J)
100   CONTINUE
      RETURN
      END
C
C
C     SOLVES 2ND PHASE CONTINUITY EQUATION FOR ALPHA
C     THE PRIME NOTATION REFERS TO THE FACT THAT THE "FLOWS"
C     DO NOT CONTAIN ALPHA.
      SUBROUTINE CONT2(I,IPH,N,X,R,DELX,DELR,U,UU,V,RHO,RHOU,ALPHANEW
     2,ALPHA,ALPHAU,GAM,RINT1S,RINT1N,RINT2S,RINT2N,RINT3M,RINTE
     2,RINTW)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RHO(2,2000),RHOU(2,2000),ALPHANEW(2,2000)
     2,ALPHAU(2,2000),UU(2,2000),VU(2,2000),U(2,2000),V(2,2000)
     2,GAM(2,2000)
     2,RINT1S(2000),RINT1N(2000),RINT2S(2000),RINT2N(2000)
     2,RINT3M(2000),RINTE(2000),RINTW(2000),ALPHA(2,2000)
     2,A(2000),B(2000),C(2000),D(2000),SOL(2000)
      XE=X
      XW=X-DELX
      DO 100 J=2,N-1
C     EAST,WEST FLOWS
      FEPRIME=RINTE(J)*RHO(IPH,J)*U(IPH,J)
      FWPRIME=RINTW(J)*RHOU(IPH,J)*UU(IPH,J)
C     NORTH FLOW
      RHON=.5D0*(RHO(IPH,J)+RHO(IPH,J+1))
      UN=.25D0*(U(IPH,J)+U(IPH,J+1)+UU(IPH,J)+UU(IPH,J+1))
      VN=.5D0*(V(IPH,J)+V(IPH,J+1))
      IF(J.EQ.N-1)THEN
       UN=0.0D0
       VN=0.0D0
      ENDIF
      FNUPRIME=-RHON*UN*RINT2N(J)
      FNVPRIME=RHON*VN*RINT1N(J)
      FNPRIME=FNUPRIME+FNVPRIME
C     SOUTH FLOW
      RHOS=.5D0*(RHO(IPH,J)+RHO(IPH,J-1))
      US=.25D0*(U(IPH,J)+U(IPH,J-1)+UU(IPH,J)+UU(IPH,J-1))
```

```
      VS=.5D0*(V(IPH,J)+V(IPH,J-1))
      IF(J.EQ.2)THEN
       VS=0.0D0
       US=U(IPH,1)
      ENDIF
      FSUPRIME=RHOS*US*RINT2S(J)
      FSVPRIME=-RHOS*VS*RINT1S(J)
      FSPRIME=FSUPRIME+FSVPRIME
C     SOURCE TERMS - WILL NEED TO BE MODIFIED
      SCGAMC2=RINT3M(J)*GAM(IPH,J)
C     COEFFICIENTS IN TRIDIAGONAL FORM
      A(J)=-DMAX1(-FSPRIME,0.0D0)
      B(J)=FEPRIME+DMAX1(FNPRIME,0.0D0)+DMAX1(FSPRIME,0.0D0)
      C(J)=-DMAX1(-FNPRIME,0.0D0)
      D(J)=ALPHAU(IPH,J)*FWPRIME+SCGAMC2
100   CONTINUE
C     BOUNDARY CONDITIONS FOR COEFFICIENTS (MAY BE UNNECESSARY)
      A(2)=0.0D0
      C(N-1)=0.0D0
C     SOLVE THE SYSTEM
      IF=2
      L=N-1
      CALL TRIDAG(IF,L,A,B,C,D,SOL)
      DO 200 J=2,N-1
       ALPHANEW(IPH,J)=SOL(J)
       IF(ALPHANEW(IPH,J).GT.1.0D0)ALPHANEW(IPH,J)=1.0D0
       IF(ALPHANEW(IPH,J).LT.0.0D0)ALPHANEW(IPH,J)=0.0D0
200   CONTINUE
C     IMPOSE BOUNDARY CONDITIONS
      ALPHANEW(IPH,N)=ALPHANEW(IPH,N-1)
      ALPHANEW(IPH,1)=ALPHANEW(IPH,2)
300   FORMAT(1X,5E13.5,2I4)
      RETURN
      END
C
C
C SOLVES FOR V2 FROM THE R2-MOM. EQN.
      SUBROUTINE R2MOM(I,IPH,IPHOTHER,N,DELX,DELR
     2,RHO,ALPHA,VISC,UU,UG,VU,VG
     2,V,RE,ALPHAU,RHOU,DELPR,RINT5,RINT6N,RINT6S,VISCU,GAM,C1IFD
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT4S,RINT4N,RINT3M,RINTE,RINTW
     2,RDXW,RDXE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RHO(2,2000),RHOU(2,2000),ALPHA(2,2000)
     2,ALPHAU(2,2000)
     2,VISC(2,2000),UU(2,2000)
     2,UG(2,2000),VU(2,2000),VG(2,2000),U(2,2000),V(2,2000),A(2000)
     2,B(2000)
     2,C(2000),D(2000),SOL(2000),GAM(2,2000),C1IFD(2000)
     2,RINT1S(2000),RINT1N(2000),RINT2S(2000),RINT2N(2000)
     2,RINT4S(2000),RINT4N(2000),RINT3M(2000),RINTE(2000),RINTW(2000)
     2,DELPR(2000),RINT5(2000),RINT6N(2000),RINT6S(2000),VISCU(2,2000)

      DO 100 J=2,N-1
C     EAST, WEST FLOWS
      FE=RINTE(J)*ALPHA(IPH,J)*RHO(IPH,J)*UG(IPH,J)
```

```
      FW=RINTW(J)*ALPHAU(IPH,J)*RHOU(IPH,J)*UU(IPH,J)
C     NORTH FLOW
      ALPHAN=2.0D0*ALPHA(IPH,J)*ALPHA(IPH,J+1)
     2/(ALPHA(IPH,J)+ALPHA(IPH,J+1))
      RHON=.5D0*(RHO(IPH,J)+RHO(IPH,J+1))
      UN=.25D0*(UG(IPH,J)+UG(IPH,J+1)+UU(IPH,J)+UU(IPH,J+1))
      VN=.5D0*(VG(IPH,J)+VG(IPH,J+1))
      FNUPRIME=-RHON*UN*RINT2N(J)
      FNVPRIME=RHON*VN*RINT1N(J)
      FNPRIME=FNUPRIME+FNVPRIME
      FN=ALPHA(IPH,J)*DMAX1(FNPRIME,0.0D0)
     2  -ALPHA(IPH,J+1)*DMAX1(-FNPRIME,0.0D0)
      IF(J.EQ.N-1)FN=0.0D0
C     SOUTH FLOW
      ALPHAS=2.0D0*ALPHA(IPH,J)*ALPHA(IPH,J-1)
     2/(ALPHA(IPH,J)+ALPHA(IPH,J-1))
      RHOS=.5D0*(RHO(IPH,J)+RHO(IPH,J-1))
      US=.25D0*(UG(IPH,J)+UG(IPH,J-1)+UU(IPH,J)+UU(IPH,J-1))
      VS=.5D0*(VG(IPH,J)+VG(IPH,J-1))
      FSUPRIME=RHOS*US*RINT2S(J)
      FSVPRIME=-RHOS*VS*RINT1S(J)
      FSPRIME=FSUPRIME+FSVPRIME
      FS=ALPHA(IPH,J)*DMAX1(FSPRIME,0.0D0)
     2  -ALPHA(IPH,J-1)*DMAX1(-FSPRIME,0.0D0)
      IF(J.EQ.2)FS=0.0D0
C     PRESSURE COEFFICIENTS
      CP1R2=RINT5(J)*ALPHA(IPH,J)*DELX*DELPR(J)
C     SOME AVERAGE VISCOSITIES
      VISCN=.5D0*(VISC(IPH,J)+VISC(IPH,J+1))
      VISCS=.5D0*(VISC(IPH,J)+VISC(IPH,J-1))
C     VISCOUS COEFFICIENTS
      CV1R2=2.0D0*ALPHAN*VISCN*RINT4N(J)/(RE*DELR)
      CV2R2=2.0D0*ALPHAS*VISCS*RINT4S(J)/(RE*DELR)
      IF(J.EQ.2)CV2R2=0.0D0
      IF(J.EQ.N-1)CV1R2=2.0D0*CV1R2
C     REGULAR SOURCE TERMS
      SV1R2=2.0D0*VISC(IPH,J)*ALPHA(IPH,J)*RINT3M(J)/(RE*RINT5(J)**2)
C       FIND DUDR TERMS (EAST,WEST) AND SC1R2
      DUDRE=(UG(IPH,J+1)-UG(IPH,J-1))/(2.0D0*DELR)
      IF(J.EQ.N-1)DUDRE=(UG(IPH,J+1)-.5D0*(UG(IPH,J)+UG(IPH,J-1)))/DELR
      IF(J.EQ.2)DUDRE=(.5D0*(UG(IPH,J+1)+UG(IPH,J))-UG(IPH,J-1))/DELR
      DUDRW=(UU(IPH,J+1)-UU(IPH,J-1))/(2.0D0*DELR)
      IF(J.EQ.N-1)DUDRW=(UU(IPH,J+1)-.5D0*(UU(IPH,J)+UU(IPH,J-1)))/DELR
      IF(J.EQ.2)DUDRW=(.5D0*(UU(IPH,J+1)+UU(IPH,J))-UU(IPH,J-1))/DELR
      SC1R2=ALPHA(IPH,J)*VISC(IPH,J)*DUDRE*RINTE(J)/RDXE/RE
     2     -ALPHAU(IPH,J)*VISCU(IPH,J)*DUDRW*RINTW(J)/RDXW/RE
C       FIND DUDR TERMS (NORTH,SOUTH) AND SC2R2
      DUDRN=.5D0*((UG(IPH,J+1)+UU(IPH,J+1))-.5D0*(UG(IPH,J)+UU(IPH,J)))
     2     /DELR
      IF(J.EQ.N-1)DUDRN=2.0D0*DUDRN
      DUDRS=.5D0*((UG(IPH,J)+UU(IPH,J))-.5D0*(UG(IPH,J-1)+UU(IPH,J-1)))
     2     /DELR
      IF(J.EQ.2)DUDRS=0.0D0
      SC2R2=-ALPHAN*VISCN*DUDRN*RINT6N(J)/RE
     2      +ALPHAS*VISCS*DUDRS*RINT6S(J)/RE
C     INTERFACIAL SOURCE TERMS
```

```
      SCIFDR2=RINT3M(J)*C1IFD(J)*(.5D0*(V(IPHOTHER,J)+V(IPHOTHER,J-1)))
      SVIFDR2=RINT3M(J)*C1IFD(J)
      SCGAMR2=DMAX1(GAM(IPH,J),0.0D0)*RINT3M(J)*(.5D0*(V(IPHOTHER,J)
     2+V(IPHOTHER,J-1)))
      SVGAMR2=DMAX1(-GAM(IPH,J),0.0D0)*RINT3M(J)
C     INTERFACIAL SOURCE TERMS FROM OTHER EQUATIONS
      SCGAMC2=GAM(IPH,J)*RINT3M(J)
C     COEFFICIENTS OF THE TRIDIAGONAL MATRIX
      A(J)=-DMAX1(-FS,0.0D0)-CV2R2
      B(J)=FW+DMAX1(-FN,0.0D0)+DMAX1(-FS,0.0D0)+CV1R2+CV2R2+SCGAMC2
     2    +SV1R2+SVIFDR2+SVGAMR2
      C(J)=-DMAX1(-FN,0.0D0)-CV1R2
      D(J)=FW*VU(IPH,J)-CP1R2+SC1R2+SC2R2+SCIFDR2+SCGAMR2
C      WRITE(*,101)J,A(J),B(J),C(J),D(J),CV1R2
100   CONTINUE
101   FORMAT(1X,I4,5E13.6)
C     BOUNDARY CONDITIONS FOR COEFFICIENTS (MAY BE UNNECESSARY)
      A(2)=0.0D0
      C(N-1)=0.0D0
C     SOLVE THE SYSTEM
      IF=2
      L=N-1
      CALL TRIDAG(IF,L,A,B,C,D,SOL)
      DO 200 J=2,N-1
       V(IPH,J)=SOL(J)
C       IF(ALPHA(IPH,J).LE.1.0D-6)V(IPH,J)=V(IPHOTHER,J)
200   CONTINUE
C     IMPOSE BOUNDARY CONDITIONS
      V(IPH,N)=0.0D0
      V(IPH,1)=0.0D0
      RETURN
      END
C
C
C SOLVES FOR THE R-DIRECTION PRESSURE GRADIENT FROM THE R1-MOM. EQN.
      SUBROUTINE DELPRCALC(I,N,R,RINT2N,RINT2S,RINT1N,RINT1S,RINT4N
     2,RINT4S
     2,RINT6S,RINT6N,RINT3M,RINT5,RINTE,RINTW,DELX,DELR,GAM,C1IFD
     2,RE,U,UU,V,VU,RHO,RHOU,ALPHA,ALPHAU,VISC,VISCU,DELPR,IPH,IPHOTHER
     2,RDXW,RDXE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RINT2N(2000),RINT2S(2000),RINT1N(2000)
     2,RINT1S(2000)
     2,RINT4N(2000),RINT4S(2000),RINT3M(2000),RINT5(2000),U(2,2000)
     2,UU(2,2000)
     2,V(2,2000),VU(2,2000),RHO(2,2000),RHOU(2,2000),ALPHA(2,2000)
     2,ALPHAU(2,2000),VISC(2,2000),VISCU(2,2000),DELPR(2000)
     2,RINTE(2000),RINTW(2000),RINT6S(2000),RINT6N(2000)
     2,GAM(2,2000),C1IFD(2000)
      DO 100 J=2,N-1
C     EAST,WEST FLOWS
      FE=RINTE(J)*RHO(IPH,J)*ALPHA(IPH,J)*U(IPH,J)
      FW=RINTW(J)*RHOU(IPH,J)*ALPHAU(IPH,J)*UU(IPH,J)
C     NORTH FLOW
      ALPHAN=.5D0*(ALPHA(IPH,J)+ALPHA(IPH,J+1))
      RHON=.5D0*(RHO(IPH,J)+RHO(IPH,J+1))
```

```
      UN=.25D0*(U(IPH,J)+U(IPH,J+1)+UU(IPH,J)+UU(IPH,J+1))
      FNU=-ALPHAN*RHON*UN*RINT2N(J)
      FNV=ALPHAN*RHON*V(IPH,J)*RINT1N(J)
      FN=FNU+FNV
      IF(J.EQ.N-1)FN=0.0D0
C     SOUTH FLOW
      ALPHAS=.5D0*(ALPHA(IPH,J)+ALPHA(IPH,J-1))
      RHOS=.5D0*(RHO(IPH,J)+RHO(IPH,J-1))
      US=.25D0*(U(IPH,J)+U(IPH,J-1)+UU(IPH,J)+UU(IPH,J-1))
      FSU=ALPHAS*RHOS*US*RINT2S(J)
      FSV=-ALPHAS*RHOS*V(IPH,J-1)*RINT1S(J)
      FS=FSU+FSV
      IF(J.EQ.2)FS=0.0D0
C     SOME AVERAGE VELOCITIES
      VM=.5D0*(V(IPH,J)+V(IPH,J-1))
      VMU=.5D0*(VU(IPH,J)+VU(IPH,J-1))
      VE=VM
      VW=VMU
C     SOME AVERAGE VISCOSITIES
      VISCN=.5D0*(VISC(IPH,J)+VISC(IPH,J+1))
      VISCS=.5D0*(VISC(IPH,J)+VISC(IPH,J-1))
C     FIND DVDR AND ASSOCIATED COEFFICIENTS
      DVDRN=(V(IPH,J+1)-V(IPH,J-1))/(2.0D0*DELR)
      IF(J.EQ.N-1)DVDRN=(V(IPH,J)-V(IPH,J-1))/DELR
      DVDRS=(V(IPH,J)-V(IPH,J-2))/(2.0D0*DELR)
      IF(J.EQ.2)DVDRS=(V(IPH,J)-V(IPH,J-1))/DELR
      C1R1=2.0D0*ALPHAN*VISCN*DVDRN*RINT4N(J)/RE
      C2R1=2.0D0*ALPHAS*VISCS*DVDRS*RINT4S(J)/RE
      C3R1=2.0D0*VISC(IPH,J)*ALPHA(IPH,J)*VM*RINT3M(J)/(RE*RINT5(J)**2)
C     FIND DUDRE,DUDRW, AND ASSOCIATED COEFFICIENTS
      DUDRE=(U(IPH,J+1)-U(IPH,J-1))/(2.0D0*DELR)
      IF(J.EQ.N-1)DUDRE=(U(IPH,J+1)-.5D0*(U(IPH,J)+U(IPH,J-1)))/DELR
      IF(J.EQ.2)DUDRE=(.5D0*(U(IPH,J+1)+U(IPH,J))-U(IPH,J-1))/DELR
      DUDRW=(UU(IPH,J+1)-UU(IPH,J-1))/(2.0D0*DELR)
      IF(J.EQ.N-1)DUDRW=(UU(IPH,J+1)-.5D0*(UU(IPH,J)+UU(IPH,J-1)))/DELR
      IF(J.EQ.2)DUDRW=(.5D0*(UU(IPH,J+1)+UU(IPH,J))-UU(IPH,J-1))/DELR
      C4R1=ALPHA(IPH,J)*VISC(IPH,J)*DUDRE*RINTE(J)/RDXE/RE
      C5R1=ALPHAU(IPH,J)*VISCU(IPH,J)*DUDRW*RINTW(J)/RDXW/RE
C     FIND DUDRS,DUDRN, AND ASSOCIATED COEFFICIENTS
      DUDRN=.5D0*(U(IPH,J+1)+UU(IPH,J+1)-U(IPH,J)-UU(IPH,J))/DELR
      IF(J.EQ.N-1)DUDRN=(U(IPH,J+1)+UU(IPH,J+1)-U(IPH,J)-UU(IPH,J))/DELR
      DUDRS=.5*(U(IPH,J)+UU(IPH,J)-U(IPH,J-1)-UU(IPH,J-1))/DELR
      IF(J.EQ.2)DUDRS=0.0D0
      C6R1=ALPHAN*VISCN*DUDRN*RINT6N(J)/RE
      C7R1=ALPHAS*VISCS*DUDRS*RINT6S(J)/RE
C     FIND OTHER COEFFICIENTS FROM INTERFACIAL TERMS
C      TEMP. VALUES OF IFDR, GAMMA, IPHOTHER!
      C8R1=RINT3M(J)*C1IFD(J)*(V(IPHOTHER,J)-.5D0*(V(IPH,J)+V(IPH,J-1)))
      C9R1=DMAX1(GAM(IPH,J),0.0D0)*V(IPHOTHER,J)*RINT3M(J)
      C10R1=DMAX1(-GAM(IPH,J),0.0D0)*VM*RINT3M(J)
C     SOLVE FOR DELPR(J)
      DELPR(J)=(FE*VE-FW*VW+FN*V(IPH,J)+FS*V(IPH,J-1)
     2        -C1R1+C2R1+C3R1-C4R1+C5R1+C6R1-C7R1-C8R1-C9R1+C10R1)
     2        /(-RINT5(J)*ALPHA(IPH,J)*DELX)
100   CONTINUE
200   FORMAT(1X,4E14.7,2I3)
```

```
      RETURN
      END
C
C
C FINDS AREAS FOR BODY FITTED GRID
      SUBROUTINE GEOMCOEFF(N,R,DELR,X,DELX,RINT1S,RINT1N,RINT2S,RINT2N
     2,RINT4S,RINT4N,RINT6S,RINT6N,RINT3M,RINTE,RINTW,RINT5,RDXM,RTXE
     2,IGOWALL,RDXW,RDXE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RINT1S(2000),RINT1N(2000),RINT2S(2000)
     2,RINT2N(2000)
     2,RINT4S(2000),RINT4N(2000),RINT3M(2000),RINTE(2000),RINTW(2000)
     2,RINT5(2000),RINT6S(2000),RINT6N(2000)
      XE=X
      XW=X-DELX
      XM=.5D0*(XE+XW)
      CALL WALL(IGOWALL,XW,RTXW,RBXW,RDXW,A1,B1,C1,D1,E1,F1)
      CALL WALL(IGOWALL,XE,RTXE,RBXE,RDXE,A1,B1,C1,D1,E1,F1)
      CALL WALL(IGOWALL,XM,RTXM,RBXM,RDXM,A1,B1,C1,D1,E1,F1)
      DO 100 J=2,N-1
       RN=R(J)+DELR/2.0D0
       RS=R(J)-DELR/2.0D0
       ANS=.5D0*(RN**2-RS**2)
       RNORS=RS
       CALL RINT(XE,XW,RS,RN,RNORS,ANS,A1,B1,C1,D1,E1,F1,RINT1
     2,RINT2,RINT3,RINT4,RINT6)
       RINT1S(J)=RINT1
       RINT2S(J)=RINT2
       RINT3M(J)=RINT3
       RINT4S(J)=RINT4
       RINT6S(J)=RINT6
       RNORS=RN
       CALL RINT(XE,XW,RS,RN,RNORS,ANS,A1,B1,C1,D1,E1,F1,RINT1
     2,RINT2,RINT3,RINT4,RINT6)
       RINT1N(J)=RINT1
       RINT2N(J)=RINT2
       RINT3M(J)=RINT3
       RINT4N(J)=RINT4
       RINT6N(J)=RINT6
       RINTE(J)=ANS*RDXE**2+RBXE*RDXE*(RN-RS)
       RINTW(J)=ANS*RDXW**2+RBXW*RDXW*(RN-RS)
       RINT5(J)=R(J)*RDXM+RBXM
100   CONTINUE
      RETURN
      END
C
C
C INITIALIZES VARIABLES FOR GLOBAL MOMENTUM CHECK LATER ON
      SUBROUTINE GLOBX1INIT(DMOMIN,DMOMOUT,PFIN,PFOUT,FVISCTOP,PFTOP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DMOMIN=0.0D0
      DMOMOUT=0.0D0
      PFIN=0.0D0
      PFOUT=0.0D0
      FVISCTOP=0.0D0
      PFTOP=0.0D0
```

```
      RETURN
      END
C
C
C
C
C     THIS SUBROUTINE SOLVES THE MOMENTUM EQUATION FOR THE SECOND PHASE
      SUBROUTINE X2MOM(I,M,IPH,IPHOTHER,N,X,R,DELX,JNEARWALL
     2,DELR,RHO,ALPHA,VISC,UU,UG,VU,VG
     2,U,V,PE,PW,RE,ALPHAU,RHOU,C1IFD,GAM,TAUWALLD
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT4S,RINT4N,RINT3M,RINTE,RINTW)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RHO(2,2000),RHOU(2,2000),ALPHA(2,2000)
     2,ALPHAU(2,2000)
     2,VISC(2,2000),UU(2,2000)
     2,UG(2,2000),VU(2,2000),VG(2,2000),U(2,2000),V(2,2000),A(2000)
     2,B(2000)
     2,C(2000),D(2000),SOL(2000),GAM(2,2000),C1IFD(2000)
     2,RINT1S(2000),RINT1N(2000),RINT2S(2000),RINT2N(2000)
     2,RINT4S(2000),RINT4N(2000),RINT3M(2000),RINTE(2000),RINTW(2000)
      XE=X
      XW=X-DELX
      DO 100 J=2,JNEARWALL-1
C     EAST,WEST FLOWS
      FE=RINTE(J)*RHO(IPH,J)*ALPHA(IPH,J)*UG(IPH,J)
      FW=RINTW(J)*RHOU(IPH,J)*ALPHAU(IPH,J)*UU(IPH,J)
C     NORTH FLOW
      ALPHAN=2.0D0*ALPHA(IPH,J)*ALPHA(IPH,J+1)
     2/(ALPHA(IPH,J)+ALPHA(IPH,J+1))
C
      RHON=.5D0*(RHO(IPH,J)+RHO(IPH,J+1))
      UN=.25D0*(UG(IPH,J)+UG(IPH,J+1)+UU(IPH,J)+UU(IPH,J+1))
      VN=.5D0*(VG(IPH,J)+VG(IPH,J+1))
      FNUPRIME=-RHON*UN*RINT2N(J)
      FNVPRIME=RHON*VN*RINT1N(J)
      FNPRIME=FNUPRIME+FNVPRIME
      FN=ALPHA(IPH,J)*DMAX1(FNPRIME,0.0D0)
     2  -ALPHA(IPH,J+1)*DMAX1(-FNPRIME,0.0D0)
      IF(J.EQ.N-1)FN=0.0D0
C     SOUTH FLOW
      ALPHAS=2.0D0*ALPHA(IPH,J)*ALPHA(IPH,J-1)
     2/(ALPHA(IPH,J)+ALPHA(IPH,J-1))
C
      RHOS=.5D0*(RHO(IPH,J)+RHO(IPH,J-1))
      US=.25D0*(UG(IPH,J)+UG(IPH,J-1)+UU(IPH,J)+UU(IPH,J-1))
      VS=.5D0*(VG(IPH,J)+VG(IPH,J-1))
      FSUPRIME=RHOS*US*RINT2S(J)
      FSVPRIME=-RHOS*VS*RINT1S(J)
      FSPRIME=FSUPRIME+FSVPRIME
      FS=ALPHA(IPH,J)*DMAX1(FSPRIME,0.0D0)
     2  -ALPHA(IPH,J-1)*DMAX1(-FSPRIME,0.0D0)
      IF(J.EQ.2)FS=0.0D0
C     PRESSURE TERM COEFFICIENTS
      CXP1=ALPHA(IPH,J)*(RINTW(J)+.5D0*(RINT2N(J)-RINT2S(J)))
      CXP2=ALPHA(IPH,J)*(RINTE(J)+.5D0*(RINT2S(J)-RINT2N(J)))
C     VISCOUS TERM COEFFIENTS
```

```
      VISCN=.5D0*(VISC(IPH,J)+VISC(IPH,J+1))
      VISCS=.5D0*(VISC(IPH,J)+VISC(IPH,J-1))
      CXV1=ALPHAN*VISCN*RINT4N(J)/(RE*(R(J+1)-R(J)))
C REPLACE/REMOVE FOR WALL FUNCTION
      IF(J.EQ.N-1)CXV1=0.0D0
      CXV2=ALPHAS*VISCS*RINT4S(J)/(RE*(R(J)-R(J-1)))
C     SOURCE TERMS
      SCIFDX=RINT3M(J)*C1IFD(J)*U(IPHOTHER,J)
      SVIFDX=RINT3M(J)*C1IFD(J)
      SCGAMC=GAM(IPH,J)*RINT3M(J)
      SVGAMX=RINT3M(J)*DMAX1(-GAM(IPH,J),0.0D0)
      SCGAMX=RINT3M(J)*DMAX1(-GAM(IPH,J),0.0D0)*U(IPHOTHER,J)
C     COEFFICIENTS IN PATANKAR'S FORM
      AS=DMAX1(-FS,0.0D0)+CXV2
      AN=DMAX1(-FN,0.0D0)+CXV1
      AW=FW
      AP=AS+AN+AW+SVIFDX+SVGAMX+SCGAMC
C     COEFFICIENTS IN TRIDIAGONAL FORM
      A(J)=-AS
      B(J)=AP
      C(J)=-AN
      D(J)=AW*UU(IPH,J)+PW*CXP1-PE*CXP2+SCIFDX+SCGAMX
100   CONTINUE
101   FORMAT(1X,I3,4E13.6)
C     BOUNDARY CONDITIONS FOR COEFFICIENTS
      A(2)=0.0D0
      C(N-1)=0.0D0
      D(JNEARWALL-1)=D(JNEARWALL-1)-C(JNEARWALL-1)*U(IPH,JNEARWALL)
      C(JNEARWALL-1)=0.0D0
C REPLACE/REMOVE FOR WALL FUNCTION
C      D(N-1)=D(N-1)-ALPHA(IPH,N-1)*TAUWALLD*RINT4N(N-1)/RE
C     SOLVE THE SYSTEM
      IF=2
      L=JNEARWALL-1
      CALL TRIDAG(IF,L,A,B,C,D,SOL)
      DO 200 J=2,JNEARWALL-1
       U(IPH,J)=SOL(J)
200   CONTINUE
C     IMPOSE BOUNDARY CONDITIONS - FIRST ORDER FIT AT BOTTOM
      U(IPH,N)=0.0D0
      U(IPH,1)=U(IPH,2)
C
      RETURN
      END
C
C
C     THIS SUBROUTINE SOLVES THE MOMENTUM EQUATION FOR THE FIRST PHASE
      SUBROUTINE X1MOM(ERRALPHA,I,M,IPH,IPHOTHER,N,X,R,DELX,JNEARWALL
     2,DELR
     2,RHO,ALPHA,VISC
     2,UU,UG,VU,VG
     2,U,V,PE,PW,RE,ALPHAU,RHOU,C1IFD,GAM
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT4S,RINT4N,RINT3M,RINTE,RINTW)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RHO(2,2000),RHOU(2,2000),ALPHA(2,2000)
     2,ALPHAU(2,2000)
```

```
     2,VISC(2,2000),UU(2,2000),C1IFD(2000)
     2,UG(2,2000),VU(2,2000),VG(2,2000),U(2,2000),V(2,2000),A(2000)
     2,B(2000)
     2,C(2000),D(2000),SOL(2000),GAM(2,2000)
     2,RINT1S(2000),RINT1N(2000),RINT2S(2000),RINT2N(2000)
     2,RINT4S(2000),RINT4N(2000),RINT3M(2000),RINTE(2000),RINTW(2000)
      XE=X
      XW=X-DELX
      DO 100 J=2,JNEARWALL-1
C     EAST,WEST FLOWS
      FE=RINTE(J)*RHO(IPH,J)*ALPHA(IPH,J)*UG(IPH,J)
      FW=RINTW(J)*RHOU(IPH,J)*ALPHAU(IPH,J)*UU(IPH,J)
C     NORTH FLOW
      ALPHAN=.5D0*(ALPHA(IPH,J)+ALPHA(IPH,J+1))
      RHON=.5D0*(RHO(IPH,J)+RHO(IPH,J+1))
      UN=.25D0*(UG(IPH,J)+UG(IPH,J+1)+UU(IPH,J)+UU(IPH,J+1))
      FNU=-ALPHAN*RHON*UN*RINT2N(J)
      FNV=ALPHAN*RHON*VG(IPH,J)*RINT1N(J)
      FN=FNU+FNV
      IF(J.EQ.N-1)FN=0.0D0
C     SOUTH FLOW
      ALPHAS=.5D0*(ALPHA(IPH,J)+ALPHA(IPH,J-1))
      RHOS=.5D0*(RHO(IPH,J)+RHO(IPH,J-1))
      US=.25D0*(UG(IPH,J)+UG(IPH,J-1)+UU(IPH,J)+UU(IPH,J-1))
      FSU=ALPHAS*RHOS*US*RINT2S(J)
      FSV=-ALPHAS*RHOS*VG(IPH,J-1)*RINT1S(J)
      FS=FSU+FSV
      IF(J.EQ.2)FS=0.0D0
C     PRESSURE TERM COEFFICIENTS
      CXP1=ALPHA(IPH,J)*(RINTW(J)+.5D0*(RINT2N(J)-RINT2S(J)))
      CXP2=ALPHA(IPH,J)*(RINTE(J)+.5D0*(RINT2S(J)-RINT2N(J)))
C     VISCOUS TERM COEFFIENTS
      VISCN=.5D0*(VISC(IPH,J)+VISC(IPH,J+1))
      VISCS=.5D0*(VISC(IPH,J)+VISC(IPH,J-1))
      CXV1=ALPHAN*VISCN*RINT4N(J)/(RE*(R(J+1)-R(J)))
      CXV2=ALPHAS*VISCS*RINT4S(J)/(RE*(R(J)-R(J-1)))
C     SOURCE TERMS - WILL NEED TO BE MODIFIED
      SCIFDX=RINT3M(J)*C1IFD(J)*UG(IPHOTHER,J)
      SVIFDX=RINT3M(J)*C1IFD(J)
      SCGAMC=GAM(IPH,J)*RINT3M(J)
      SVGAMX=RINT3M(J)*DMAX1(-GAM(IPH,J),0.0D0)
      SCGAMX=RINT3M(J)*DMAX1(-GAM(IPH,J),0.0D0)*UG(IPHOTHER,J)
C     COEFFICIENTS IN PATANKAR'S FORM
      AS=DMAX1(-FS,0.0D0)+CXV2
      AN=DMAX1(-FN,0.0D0)+CXV1
      AW=FW
      AP=AS+AN+AW+SVIFDX+SVGAMX+SCGAMC
C     COEFFICIENTS IN TRIDIAGONAL FORM
      A(J)=-AS
      B(J)=AP
      C(J)=-AN
      D(J)=AW*UU(IPH,J)+PW*CXP1-PE*CXP2+SCIFDX+SCGAMX
99    FORMAT(1X,I3,5E13.6)
C
100   CONTINUE
C     BOUNDARY CONDITIONS FOR COEFFICIENTS
```

```
      A(2)=0.0D0
C      C(N-1)=0.0D0
      D(JNEARWALL-1)=D(JNEARWALL-1)-C(JNEARWALL-1)*U(IPH,JNEARWALL)
      C(JNEARWALL-1)=0.0D0
C     SOLVE THE SYSTEM
      IF=2
      L=JNEARWALL-1
      CALL TRIDAG(IF,L,A,B,C,D,SOL)
      DO 200 J=2,JNEARWALL-1
       U(IPH,J)=SOL(J)
200   CONTINUE
C     IMPOSE BOUNDARY CONDITIONS - FIRST ORDER FIT ON BOTTOM
      U(IPH,N)=0.0D0
      U(IPH,1)=U(IPH,2)
C
      RETURN
      END
C
C
C
C
C CALCULATES V1 FROM 1ST PHASE (CONTINUOUS) CONTINUITY EQN.
      SUBROUTINE CONT1(IPH,N,X,R,DELX,DELR,U,UU,V,RHO,RHOU,ALPHA,ALPHAU
     2,GAM,VU
     2,RINT1S,RINT1N,RINT2S,RINT2N,RINT3M,RINTE,RINTW)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000),RHO(2,2000),RHOU(2,2000),ALPHA(2,2000)
     2,ALPHAU(2,2000)
     2,UU(2,2000),VU(2,2000),U(2,2000),V(2,2000),GAM(2,2000)
     2,RINT1S(2000),RINT1N(2000),RINT2S(2000),RINT2N(2000)
     2,RINT3M(2000),RINTE(2000),RINTW(2000)
      XE=X
      XW=X-DELX
      DO 100 J=2,N-1
C     EAST,WEST FLOWS
      FE=RINTE(J)*RHO(IPH,J)*ALPHA(IPH,J)*U(IPH,J)
      FW=RINTW(J)*RHOU(IPH,J)*ALPHAU(IPH,J)*UU(IPH,J)
C     NORTH FLOW
      ALPHAN=.5D0*(ALPHA(IPH,J)+ALPHA(IPH,J+1))
      RHON=.5D0*(RHO(IPH,J)+RHO(IPH,J+1))
      UN=.25D0*(U(IPH,J)+U(IPH,J+1)+UU(IPH,J)+UU(IPH,J+1))
      FNU=-ALPHAN*RHON*UN*RINT2N(J)
      IF(J.EQ.N-1)FNU=0.0D0
C     SOUTH FLOW
      ALPHAS=.5D0*(ALPHA(IPH,J)+ALPHA(IPH,J-1))
      RHOS=.5D0*(RHO(IPH,J)+RHO(IPH,J-1))
      US=.25D0*(U(IPH,J)+U(IPH,J-1)+UU(IPH,J)+UU(IPH,J-1))
      FSU=ALPHAS*RHOS*US*RINT2S(J)
      FSV=-ALPHAS*RHOS*V(IPH,J-1)*RINT1S(J)
      FS=FSU+FSV
      IF(J.EQ.2)FS=0.0D0
C     SOURCE TERM
      SCGAMC=GAM(IPH,J)*RINT3M(J)
C     SOLVE FOR V
      V(IPH,J)=(FW-FE-FS-FNU+SCGAMC)/(ALPHAN*RHON*RINT1N(J))
100   CONTINUE
```

```
C     IMPOSE BOUNDARY CONDITIONS
      V(IPH,N-1)=0.0D0
      V(IPH,N)=0.0D0
      V(IPH,1)=0.0D0
      RETURN
      END
C
C
C
C
C  FINDS ERROR BETWEEN GUESSED AND CALCULATED VELOCITIES FOR CONTINUOUS
C  PHASE.
      SUBROUTINE CONERROR(N,IPH,U,UG,V,VG,CONERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(2,2000),UG(2,2000),V(2,2000),VG(2,2000)
      CONERR=0.0D0
      DO 100 J=2,N-2
       CONERR=CONERR+ABS((U(IPH,J)-UG(IPH,J)))
     2              +ABS((V(IPH,J)-VG(IPH,J)))
100   CONTINUE
      J=N-1
      CONERR=CONERR+ABS((U(IPH,J)-UG(IPH,J)))
      CONERR=CONERR/U(IPH,1)
      RETURN
      END
C
C
C
C UPDATES VELOCITIES FOR NEW ITERATION WITHIN CONTINUOUS PHASE
      SUBROUTINE VELUPDATE(IPH,I,U,V,UG,VG,N,JNEARWALL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(2,2000),V(2,2000),UG(2,2000),VG(2,2000)
      FI=.8D0
      DO 100 J=2,JNEARWALL-1
        UG(IPH,J)=U(IPH,J)*FI+UG(IPH,J)*(1.0D0-FI)
        VG(IPH,J)=V(IPH,J)*FI+VG(IPH,J)*(1.0D0-FI)
100   CONTINUE
      DO 200 J=JNEARWALL,N
       UG(IPH,J)=U(IPH,J)
       VG(IPH,J)=V(IPH,J)
200   CONTINUE
      UG(IPH,1)=U(IPH,1)
      VG(IPH,1)=V(IPH,1)
      RETURN
      END
C
C
C
C UPDATES VELOCITIES FOR BOTH PHASES
      SUBROUTINE VELUPDATE2(U,UU,UG,V,VU,VG,RHO,RHOU
     2,VISC,VISCU,VISCL,VISCLU,VISCT,VISCTU,ALPHA
     2,ALPHAU,N,PW,PE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(2,2000),UU(2,2000),UG(2,2000),V(2,2000),VU(2,2000)
     2,VG(2,2000),ALPHA(2,2000),ALPHAU(2,2000),RHO(2,2000),RHOU(2,2000)
     2,VISC(2,2000),VISCU(2,2000),VISCL(2,2000),VISCLU(2,2000)
```

```
     2,VISCT(2,2000),VISCTU(2,2000)
      DO 200 IPH=1,2
      DO 100 J=1,N
       UG(IPH,J)=U(IPH,J)
       UU(IPH,J)=U(IPH,J)
       VG(IPH,J)=V(IPH,J)
       VU(IPH,J)=V(IPH,J)
       ALPHAU(IPH,J)=ALPHA(IPH,J)
       RHOU(IPH,J)=RHO(IPH,J)
       VISCU(IPH,J)=VISC(IPH,J)
       VISCLU(IPH,J)=VISCL(IPH,J)
       VISCTU(IPH,J)=VISCT(IPH,J)
100   CONTINUE
200   CONTINUE
      PW=PE
      RETURN
      END
C
C
      SUBROUTINE USERINPUT(RE,XMAX,NCVX,NCVR,NCVRMOT,URATIO,RRATIO
     2,SLIPMN,SLIPSN,RHOVAPOR,VISCVAPOR,ALPHAVAPORMN,ALPHAVAPORSN
     2,IREGIME,IPHCONT,IPHDISC,RHOLIQ,VISCLIQ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IREGIME=2
      CALL REGIME(IREGIME,IPHCONT,IPHDISC)
C TWO-PHASE CONDITIONS FROM HAKAN'S DATA
C ASSUMING ULMN=275 FT/S, AND UVMN=100 FT/S FROM GREG'S PREVIOUS RUNS
      RE=3490648.0D0
      XMAX=20.0D0
      NCVX=40
      NCVR=100
      RRATIO=.72D0
      SLIPMN=0.39D0
      SLIPSN=1.0D0
      RHOVAPOR=.019D0
      RHOLIQ=1.0D0
      VISCVAPOR=.049D0
      VISCLIQ=1.0D0
      QUALITY=.26202
      ALPHAVAPORMN=QUALITY/(QUALITY+SLIPMN*(1.0D0-QUALITY)*RHOVAPOR)
      ALPHAVAPORSN=0.9999999999999D0
      URATIO=.519D0
      WRITE(*,*)'ALPHAVAPORMN = ',ALPHAVAPORMN
      WRITE(*,*)'ALPHAVAPORSN = ',ALPHAVAPORSN
      WRITE(*,*)'ALPHALIQUIDMN = ',1.0D0-ALPHAVAPORMN
      WRITE(*,*)'ALPHALIQUIDSN = ',1.0D0-ALPHAVAPORSN
      NCVRMOT=DINT(RRATIO*NCVR)
C ENTER USERINPUT CONDITIONS TO FILE
      WRITE(13,*)'NCVX = ',NCVX
      WRITE(13,*)'NCVR = ',NCVR
      WRITE(13,*)'NCVRMOT = ',NCVRMOT
      WRITE(13,*)'XMAX = ',XMAX
      WRITE(13,*)'RRATIO = ',RRATIO
      WRITE(13,*)'RE = ',RE
      WRITE(13,*)'SLIPMN = ',SLIPMN
      WRITE(13,*)'SLIPSN = ',SLIPSN
```

```
      WRITE(13,*)'URATIO = ',URATIO
      WRITE(13,*)'RHOVAPOR = ',RHOVAPOR
      WRITE(13,*)'VISCVAPOR = ',VISCVAPOR
      WRITE(13,*)'RHOLIQ = ',RHOLIQ
      WRITE(13,*)'VISCLIQ = ',VISCLIQ
      WRITE(13,*)'QUALITY (MN OUTLET) = ',QUALITY
      WRITE(13,*)'ALPHAVAPORMN = ',ALPHAVAPORMN
      WRITE(13,*)'ALPHAVAPORSN = ',ALPHAVAPORSN
      WRITE(13,*)'IREGIME = ',IREGIME
      RETURN
      END
C
C
C     GRID -- GEOMETRIC PARAMETERS OF THE GRID
      SUBROUTINE GRID(NCVX,NCVR,NCVRMOT,XMAX,DELX,DELR,M,N,NMOT,R)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(2000)
      DELR=1.0D0/DBLE(NCVR)
      DELX=XMAX/DBLE(NCVX)
      N=NCVR+2
      NMOT=NCVRMOT+2
      M=NCVX+2
C ESTABLISH THE VALUES OF R FOR EACH CELL
      R(1)=0.0D0
      DO 10 J=2,N-1
       R(J)=(2.0D0*DBLE(J)-3.0D0)/(2.0D0*DBLE(NCVR))
10    CONTINUE
      R(N)=1.0D0
      RETURN
      END
C
C
C FLUID PROPERTIES (ASSUMED CONSTANT)
      SUBROUTINE FLUIDPROP(RHOVAPOR,VISCVAPOR,ALPHAVAPORMN,ALPHAVAPORSN
     2,RHO,VISCL,ALPHA,RHOLIQ,VISCLIQ,N,NMOT)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION RHO(2,2000),VISCL(2,2000),ALPHA(2,2000)
      DO 100 J=1,N
       RHO(1,J)=RHOLIQ
       RHO(2,J)=RHOVAPOR
       VISCL(1,J)=VISCLIQ
       VISCL(2,J)=VISCVAPOR
100   CONTINUE
      DO 200 J=1,NMOT-1
       ALPHA(1,J)=1.0D0-ALPHAVAPORMN
       ALPHA(2,J)=ALPHAVAPORMN
200   CONTINUE
      DO 300 J=NMOT,N
       ALPHA(1,J)=1.0D0-ALPHAVAPORSN
       ALPHA(2,J)=ALPHAVAPORSN
300   CONTINUE
      RETURN
      END
C
C
C
```

```
C INITIAL VALUES AT INLET FOR SEVERAL PARAMETERS
      SUBROUTINE INIT(N,NMOT,R,RRATIO,URATIO,SLIPMN,SLIPSN,RHOVAPOR
     2,ALPHAVAPORMN,VISCVAPOR,ALPHAVAPORSN
     2,UU,VU,RHOU,ALPHAU,VISCLU,IMERGE,RHOLIQ,VISCLIQ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION UU(2,2000),VU(2,2000),RHOU(2,2000),ALPHAU(2,2000)
     2,VISCLU(2,2000),GAM(2,2000),R(2000)
C CONDITIONS IN MOTIVE PORTION
      DO 200 J=2,NMOT-1
       UU(1,J)=1.0D0
       UU(2,J)=1.0D0*SLIPMN
C INITIAL RADIAL VELOCITY DISTRIBUTION
       VU(1,J)=0.0D0
       VU(2,J)=0.0D0
       VU(1,NMOT-1)=0.0D0
       VU(2,NMOT-1)=0.0D0
       VU(1,NMOT-2)=0.0D0
       VU(2,NMOT-2)=0.0D0
       VU(1,NMOT-3)=0.0D0
       VU(2,NMOT-3)=0.0D0
       VU(1,2)=0.0D0
       VU(2,2)=0.0D0
       VU(1,3)=0.0D0
       VU(2,3)=0.0D0
       VU(3,3)=0.0D0
C
       RHOU(1,J)=RHOLIQ
       RHOU(2,J)=RHOVAPOR
       ALPHAU(1,J)=1.0D0-ALPHAVAPORMN
       ALPHAU(2,J)=ALPHAVAPORMN
       VISCLU(1,J)=VISCLIQ
       VISCLU(2,J)=VISCVAPOR
       GAM(1,J)=0.0D0
       GAM(2,J)=GAM(1,J)
200   CONTINUE
C CONDITIONS IN SUCTION PORTION
      DO 100 J=NMOT,N-1
       UU(1,J)=URATIO
       UU(2,J)=URATIO*SLIPSN
       VU(1,J)=0.0D0
       VU(2,J)=0.0D0
       RHOU(1,J)=RHOLIQ
       RHOU(2,J)=RHOVAPOR
       ALPHAU(1,J)=1.0D0-ALPHAVAPORSN
       ALPHAU(2,J)=ALPHAVAPORSN
       VISCLU(1,J)=VISCLIQ
       VISCLU(2,J)=VISCVAPOR
       GAM(1,J)=0.0D0
       GAM(2,J)=-GAM(1,J)
100   CONTINUE
C BOUNDARY CONDITIONS VALID FOR BOTH PORTIONS
      UU(1,1)=1.0D0
      UU(2,1)=SLIPMN
      UU(1,N)=0.0D0
      UU(2,N)=0.0D0
      VU(1,1)=0.0D0
```

```
      VU(2,1)=0.0D0
      VU(1,N)=0.0D0
      VU(2,N)=0.0D0
      VU(1,N-1)=0.0D0
      RHOU(1,1)=RHOLIQ
      RHOU(2,1)=RHOVAPOR
      RHOU(1,N)=RHOLIQ
      RHOU(2,N)=RHOVAPOR
      ALPHAU(1,1)=1.0D0-ALPHAVAPORMN
      ALPHAU(2,1)=ALPHAVAPORMN
      ALPHAU(1,N)=1.0D0-ALPHAVAPORSN
      ALPHAU(2,N)=ALPHAVAPORSN
      VISCLU(1,1)=VISCLIQ
      VISCLU(2,1)=VISCVAPOR
      VISCLU(1,N)=VISCLIQ
      VISCLU(2,N)=VISCVAPOR
C INITIALIZE VALUE OF IMERGE, THE I POSITION FOR WHEN THE
C JET AND WALL BOUNDARY LAYERS HAVE MERGED
      IMERGE=100000000
C
      RETURN
      END
C
C
C CALCULATES THE INLET FLOWRATE FOR THE CURRENT PHASE
      SUBROUTINE CALCFLOWIN(IPH,N,RHOU,ALPHAU,UU,RINTW,FLOWIN,UAVE
     2,UAVE2,TOTFLOW1,TOTFLOW2)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION RHOU(2,2000),ALPHAU(2,2000),UU(2,2000),R(2000)
     2,RINTW(2000)
      DO 100 J=2,N-1
       FW=RINTW(J)*RHOU(IPH,J)*ALPHAU(IPH,J)*UU(IPH,J)
       FLOWIN=FLOWIN+FW
100   CONTINUE
C CALCULATE THE AVERAGE VELOCITY BASED ON INLET FLOWRATE

       TOTFLOW=0.0D0
       TOTAREA=0.0D0
       TOTFLOW1=0.0D0
       TOTFLOW2=0.0D0
       TOTAREA2=0.0D0
       DO 300 J=2,N-1
        TOTFLOW=TOTFLOW+RINTW(J)*(RHOU(1,J)*ALPHAU(1,J)*UU(1,J)
     2         +RHOU(2,J)*ALPHAU(2,J)*UU(2,J))
        TOTFLOW1=TOTFLOW1+RINTW(J)*RHOU(1,J)*ALPHAU(1,J)*UU(1,J)
        TOTFLOW2=TOTFLOW2+RINTW(J)*RHOU(2,J)*ALPHAU(2,J)*UU(2,J)
        TOTAREA2=TOTAREA2+RINTW(J)*ALPHAU(2,J)
        TOTAREA=TOTAREA+RINTW(J)
300    CONTINUE
C    ASSUME VAPOR DENSITY HOLDS
      UAVE=TOTFLOW/(TOTAREA*RHOU(2,N-1))
      UAVE2=TOTFLOW2/(TOTAREA2*RHOU(2,N-1))
      WRITE(*,*)'TOTFLOW1 = ',TOTFLOW1
      WRITE(*,*)'TOTFLOW2 = ',TOTFLOW2
      WRITE(*,*)'TOTFLOW = ',TOTFLOW
      RETURN
```

```
      END
C
C
C INITIAL GUESS VELOCITIES
      SUBROUTINE GUESSVEL(UG,VG,UU,VU,N)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION UG(2,2000),VG(2,2000),UU(2,2000),VU(2,2000)
      DO 100 J=2,N-1
       UG(1,J)=UU(1,J)
       UG(2,J)=UU(2,J)
       VG(1,J)=VU(1,J)
       VG(2,J)=VU(2,J)
100   CONTINUE
      UG(1,1)=UU(1,1)
      UG(2,1)=UU(2,1)
      UG(1,N)=UU(1,N)
      UG(2,N)=UU(2,N)
      VG(1,1)=VU(1,1)
      VG(2,1)=VU(2,1)
      VG(1,N)=VU(1,N)
      VG(2,N)=VU(2,N)
      RETURN
      END
C
C
C INITIALIZES INLET PRESSURE AND COEFFICIENT FOR PRESSURE GRADIENT
      SUBROUTINE PINIT(DELPCOEFF,PW)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DELPCOEFF=0.0D0
      PW=0.0D0
      RETURN
      END
C
C
C FINDS ERROR IN CONTINUOUS PHASE FLOWRATE BASED ON GUESSED
C PRESSURE GRADIENT
      SUBROUTINE FLOWERROR(IPH,N,RHO,ALPHA,U,FLOWIN,RINTE,FLOWERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION U(2,2000),RHO(2,2000),ALPHA(2,2000),RINTE(2000)
      FLOWOUT=0.0D0
      DO 100 J=2,N-1
       FE=RINTE(J)*RHO(IPH,J)*ALPHA(IPH,J)*U(IPH,J)
       FLOWOUT=FLOWOUT+FE
100   CONTINUE
      FLOWERR=(FLOWOUT-FLOWIN)/FLOWIN
      RETURN
      END
C
C
C GIVES THE COEFFICIENTS FOR THE EQUATION OF THE LINE
C FOR THE TOP AND BOTTOM BOUNDARY OF THE DOMAIN.
      SUBROUTINE WALL(IGOWALL,X,RT,RB,RD,A,B,C,D,E,F)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IGOWALL=IGOWALL+1
C     SIMPLE STRAIGHT DUCT
      A=0.0D0
```

```
      B=1.0D0
      C=0.0D0
      D=0.0D0
      IF(IGOWALL.EQ.1)THEN
       WRITE(13,*)'WALL GEOMETRY'
       WRITE(13,*)'A=0.0'
       WRITE(13,*)'B=1.0'
       WRITE(13,*)'C=0.0'
       WRITE(13,*)'D=0.0'
      ENDIF
      E=A-C
      F=B-D
      RT=A*X+B
      RB=C*X+D
      RD=E*X+F
      RETURN
      END
C
C
C
C FINDS AREAS FOR BODY FITTED CELLS BASED ON SHAPE OF TOP AND BOTTOM
C BOUNDARY OF DOMAIN.
      SUBROUTINE RINT(XE,XW,RS,RN,RNORS,ANS,A,B,C,D,E,F,RINT1,RINT2
     2,RINT3,RINT4,RINT6)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      RNS=RN-RS
      RINT1=.5D0*(XE**2-XW**2)*(RNORS*E+C)+(XE-XW)*(RNORS*F+D)
      RINT2=(RNORS*E+C)*(.5D0*(XE**2-XW**2)*(RNORS*E+C)+(XE-XW)
     2*(RNORS*F+D))
      RINT3=(1.0D0/3.0D0)*(ANS*E**2+C*E*RNS)*(XE**3-XW**3)
     2+.5D0*(2.0D0*ANS*F*E+D*E*RNS+F*C*RNS)*(XE**2-XW**2)
     2+(ANS*F**2+D*F*RNS)*(XE-XW)
      IF(E.NE.0.0D0)THEN
       RINT4=(RNORS+C/E)*(XE-XW)+(D/E-F*C/E**2)*DLOG((E*XE+F)/(E*XW+F))
       RINT6=(RNORS**2*E+RNORS*C)*(XE-XW)+(RNORS*E+C)*(C*(XE-XW)/E
     2        +(D/E-C*F/E**2)*DLOG((E*XE+F)/(E*XW+F)))
      ELSE
       RINT4=(C/(2.0D0*F))*(XE**2-XW**2)+(RNORS+D/F)*(XE-XW)
       RINT6=(RNORS**2*E+RNORS*C+D*(RNORS*E+C)/F)*(XE-XW)
     2        +C*(RNORS*E+C)*(XE**2-XW**2)/(2.0D0*F)
      ENDIF
      RETURN
      END
C
C
C ROUTINE TO QUICKLY FIND THE NEXT GUESS AXIAL PRESSURE GRADIENT.
      SUBROUTINE DPITER(DIRE,KDELPITER,DELPCOEFF,FLOWERR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C IF BOUNDED THEN JUMPS IMMEDIATELY TO BOUNDED PART OF ROUTINE
      IF(KDELPITER.GE.3)GO TO 10
C ESTABLISHES FIRST TWO VALUES OF DELPCOEFF
      IF(KDELPITER.EQ.1)THEN
        DELPCOEFF1=DELPCOEFF
        FLOWERR1=FLOWERR
        DIR=-1.0D0*DIRE
        DELPCOEFF=DELPCOEFF1+DIR*0.1D0
```

```
        KDELPITER=KDELPITER+1
        RETURN
      ELSEIF(KDELPITER.EQ.2)THEN
        DELPCOEFF2=DELPCOEFF
        FLOWERR2=FLOWERR
      ENDIF
C DECIDES IF VALUES 1 AND 2 ARE BOUNDED
      IF(FLOWERR1*FLOWERR2.LT.0.0D0)GO TO 10
C IF VALUES 1 AND 2 ARE UNBOUNDED
      IF(ABS(FLOWERR2).LE.ABS(FLOWERR1))THEN
        DELPCOEFF1=DELPCOEFF2
        FLOWERR1=FLOWERR2
        DELPCOEFF=DELPCOEFF2+DIR*0.1D0
        RETURN
      ELSEIF(ABS(FLOWERR1).LT.ABS(FLOWERR2))THEN
        DIR=1.0D0*DIRE
        DELPCOEFF=DELPCOEFF1+DIR*0.1D0
        RETURN
      ENDIF
C IF VALUES 1 AND 2 ARE BOUNDED
10    KDELPITER=KDELPITER+1
      IF(KDELPITER.EQ.3)THEN
        DELPCOEFF=-FLOWERR2*((DELPCOEFF1-DELPCOEFF2)
     2/(FLOWERR1-FLOWERR2))+DELPCOEFF2
        RETURN
      ELSE
        FLOWERR3=FLOWERR
        DELPCOEFF3=DELPCOEFF
        IF(FLOWERR3*FLOWERR2.LT.0.0D0)THEN
          DELPCOEFF1=DELPCOEFF2
          FLOWERR1=FLOWERR2
          DELPCOEFF2=DELPCOEFF3
          FLOWERR2=FLOWERR3
          DELPCOEFF=-FLOWERR2*((DELPCOEFF1-DELPCOEFF2)
     2/(FLOWERR1-FLOWERR2))+DELPCOEFF2
          RETURN
        ELSE
          DELPCOEFF2=DELPCOEFF3
          FLOWERR2=FLOWERR3
          DELPCOEFF=-FLOWERR2*((DELPCOEFF1-DELPCOEFF2)
     2/(FLOWERR1-FLOWERR2))+DELPCOEFF2
          RETURN
        ENDIF
      ENDIF
      END
C
C
C THOMAS ALGORITHM TO SOLVE A TRIDIAGONAL MATRIX OF LINEAR EQUATIONS.
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
C TRIDIAGONAL MATRIX SOLVER
C  A,B,C,D = COEFFICIENTS IN THE TRIDIAGONAL SYSTEM
C  V       = SOLUTION VECTOR
C
C  A(I)*V(I-1)+B(I)*V(I)+C(I)*V(I+1)=D(I)
C
C  (NOTE THAT PATANKAR'S NOMENCLATURE CORRESPONDS TO
```

```
C   AP=B(I), AW=-A(I), AE=C(I) )
C
C  IF,L    = BEGINING (FIRST) AND ENDING (LAST) INDEXES FOR V
C   SOLUTION (USUALLY IF=1 OR 2 AND L=LMAX OR LMAX-1
C             DEPENDING ON BOUNDARY CONDITIONS)
C
C REMOVE THE COMMENT C IN FRONT OF THE FOLLOWING LINE TO USE
C DOUBLE PRECISION
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(1),B(1),C(1),D(1),V(1),BETA(2000),GAMMA(2000)
C
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
C
      DO 1 I=IFP1,L
        BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
5       FORMAT(1X,I2,1X,4F8.3)
1       GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      V(L)=GAMMA(L)
      LAST=L-IF
C
      DO 2 K=1,LAST
        I=L-K
2       V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END
```