```
C
C     THE DIMENSIONS ARE SET BY THE FOLLOWING PARAMETER STATEMENT.:
C     NG  = TOTAL NUMBER OF POINTS USED ON THE STREAM SURFACE.
C     NST = NUMBER OF STAGES.
C     NSC = NUMBER OF BLADE SECTIONS TO BE GENERATED.
      PARAMETER(NG=99, NST=20, NSC= 11)
C
      DIMENSION
     &   HO(NG),V(NG),S(NG),P(NG),T(NG),G(NG),VS(NG),RHO(NG),
     &   WET(NG),PSI(NG),PHI(NG),RHOMID(NG),RHOEXIT(NG),HEXIT(NG),
     &   DHO(NG),U(NG),VXOUT(NG),SMID(NG),SEXIT(NG),VMRAT(NG),
     &   VXMID(NG),HOEXIT(NG),HOMID(NG),HMID(NG),PO(NG),PMID(NG),
     &   PEXIT(NG),ETA(NG),ASPN(NG),ASPR(NG),SPAN(NG),
     &   RMEANALL(NST,NG),XMEANALL(NST,NG),VMLOCALL(NST,NG),
     &   RHUBALL(NST,NG),RTIPALL(NST,NG),NLE1_ALL(NST),NTE1_ALL(NST),
     &   XHUBALL(NST,NG),XTIPALL(NST,NG),NLE2_ALL(NST),NTE2_ALL(NST),
     &   NSS_STG(NST),NLE1_STG(NST),NTE1_STG(NST),
     &   NLE2_STG(NST),NTE2_STG(NST),ALPHA_IN(NG),ALPHA_OUT(NG),
     &   ROWGAP(NST),STAGEGAP(NST),DEVN1(NG),DEVN2(NG),AINC1(NG),
     &   AINC2(NG)
C
      COMMON /SET7/ HOIN,SI,RGAS,CPGAS,POIN,TOIN,GAMM
C
      DIMENSION RHUB(NG),RTIP(NG),XHUB(NG),XTIP(NG),DHOIS(NG),VX(NG),
     &          REACN(NG),RDES(NG),VIN(NG),AXCHRD1(NG),AXCHRD2(NG),
     &          XSURFHUB(NG),XSURFTIP(NG),RSURFHUB(NG),RSURFTIP(NG),
     &          SDISTHUB(NG),SDISTTIP(NG)
C
      DIMENSION  NBLADE(NG),XSECT(NG),RSECT(NG),
     &           PSTATIN(NG),PSTATOUT(NG),PROTOUT(NG),
     &           HINLET(NG),SINLET(NG),PINLET(NG),VM(NG),
     &           XMEAN(NG),RMEAN(NG),VMER(NG),SDIST(NG),PITCH_ANGL(NG),
     &           FBLOCK_LE(NST),FBLOCK_TE(NST),FBLOCK(NG)
C
      DIMENSION BIN_ROW1(NG),BOUT_ROW1(NG), BIN_ROW2(NG),BOUT_ROW2(NG),
     &          QLE_ROW1(NG),QTE_ROW1(NG),QLE_ROW2(NG),QTE_ROW2(NG)
C
      DIMENSION   RHOINLET(NG),HOINLET(NG),VXIN(NG),
     &            TIN(NG),TMID(NG),TEXIT(NG),POREL(NG),POABS(NG),
     &            VRELIN(NG),VABSIN(NG),VRELMID(NG),VABSMID(NG),
     &            VRELEX(NG),VABSEX(NG),RHUBIN(NG),RHUBMID(NG),
     &            RHUBEXIT(NG),RTIPIN(NG),RTIPMID(NG),RTIPEXIT(NG),
     &            TKMAX_S(NST,NSC),XTKMAX_S(NST,NSC),TINLET(NG),
     &            TKMAX_R(NST,NSC),XTKMAX_R(NST,NSC)
C
      DIMENSION   VABS(NG),VREL(NG),HOLOC(NG),SLOC(NG),PLOC(NG),
     &            TLOC(NG),RHOLOC(NG),PHI_LOC(NG),VM_LOC(NG),U_LOC(NG),
     &            VTLOC(NG)
C
      REAL        MACH_REL(NG),MACH_ABS(NG)
C
      CHARACTER*10  IFSAME_RAD, IF_RDES, IFHUB,
     &              IFSAME_ADM, IFSAME_FLO, IFSAME_ANG, RADTYPE
      CHARACTER*1   INTYPE, ASP_TYP, ROWTYP, TURBO_TYP,ANSTK,ANSFLO,
     &              IFSAME_ALL,ANSSS,ANS,ANSANGL, MIXTYP,ANSOUT,ANSIN,
     &              IFOUT(NG),IF_ROT
      CHARACTER*3   FLO_TYP
      CHARACTER*72  DUMMY_LINE
C
      OPEN(UNIT=10, FILE= 'meangen.out')
      OPEN(UNIT=5,  FILE= '/dev/tty')
C
      PI     = 3.14159
      DEGRAD = PI/180.
      RADDEG = 180./PI
      DEG    = PI/180.
C
      WRITE(6,*)'*************************************************
     &************************************************************'
      WRITE(6,*)'*************************************************
     &************************************************************'
      WRITE(6,*)
      WRITE(6,*)'                WELCOME TO MEANGEN '
```

```
      WRITE(6,*)
      WRITE(6,*)'THIS IS AN INTERACTIVE PROGRAM FOR THE ONE-DIMENSIONAL'
      WRITE(6,*)'              DESIGN OF AXIAL TURBOMACHINES.'
      WRITE(6,*)
      WRITE(6,*)'ANSWER THE QUESTIONS AS THEY APPEAR ON THE SCREEN '
      WRITE(6,*)'AND THE PROGRAM WILL WRITE A DATA SET FOR THE'
      WRITE(6,*)'BLADE GEOMETRY PROGRAM "STAGEN" WHICH IN TURN WILL'
      WRITE(6,*)'GENERATE A 3D DATASET FOR "MULTALL-OPEN".'
      WRITE(6,*)
      WRITE(6,*)'*********************************************************
     &*****************************************************************'
      WRITE(6,*)'*********************************************************
     &*****************************************************************'
C
      WRITE(6,*) ' INPUT FROM SCREEN OR FILE ? '
      WRITE(6,*) ' ANSWER "S"  or  "F" .'
      READ(5,*)    ANSIN
      IF(ANSIN.EQ.'F'.OR.ANSIN.EQ.'f') THEN
           ANSIN = 'F'
           CLOSE(5)
           OPEN(UNIT=5, FILE='meangen.in' )
      END IF
C
      WRITE(6,*)'*********************************************************'
      WRITE(6,*)'*********************************************************'
      WRITE(6,*)
      WRITE(6,*) '      IS THIS A COMPRESSOR OR A TURBINE ?'
      WRITE(6,*) '      ANSWER "C"  or  "T" . '
           READ(5,*)            TURBO_TYP
           IF(TURBO_TYP.EQ.'t') TURBO_TYP= 'T'
           IF(TURBO_TYP.EQ.'c') TURBO_TYP= 'C'
           WRITE(10,101)        TURBO_TYP
      FORMAT(A1,T25,' TURBO_TYP,"C" FOR A COMPRESSOR,"T" FOR A TURBINE')
      WRITE(6,*)'*********************************************************'
      WRITE(6,*)'*********************************************************'
C
      WRITE(6,*)
C
      WRITE(6,*)    ' DO YOU WANT TO DESIGN AN AXIAL FLOW MACHINE WITH A
     &CONSTANT RADIUS AT A FIXED SPANWISE POSITION ON EACH STAGE ?'
      WRITE(6,*)    ' AND WITH REPEATING FLOW CONDITIONS.'
      WRITE(6,*)    ' OR A MIXED FLOW MACHINE WITH SIGNIFICANT CHANGES IN
     & RADIUS THROUGH A STAGE ?'
      WRITE(6,*)    ' ANSWER "AXI" or "MIX"  '
      READ(5,*)     FLO_TYP
      IF(FLO_TYP.EQ.'axi') FLO_TYP = 'AXI'
      IF(FLO_TYP.EQ.'mix') FLO_TYP = 'MIX'
      WRITE(10,5)  FLO_TYP
      FORMAT(A3,T25, ' FLO_TYP FOR AXIAL OR MIXED FLOW MACHINE ')
C
      WRITE(6,*)
      WRITE(6,*)
     &  'THE BLADE ROTATION MUST BE IN THE POSITIVE THETA DIRECTION.'
      WRITE(6,*)
      WRITE(6,*)
     &  'ALL FLOW ANGLES ARE POSITIVE IF THE ASSOCIATED FLOW VECTOR HAS
     & A POSITIVE COMPONENT IN THE DIRECTION OF ROTATION.'
      ROTN = 1.0
      WRITE(6,*)'*********************************************************'
      WRITE(6,*)'*********************************************************'
      WRITE(6,*)

C*******************************************************************************
C*******************************************************************************
C    SET DEFAULTS
C*******************************************************************************
C*******************************************************************************
C
      IF(TURBO_TYP.EQ.'T') THEN   !   DEFAULTS FOR TURBINES.
           TKLE     = 0.04 ! LEADING EDGE THICKNESS/AXIAL CHORD.
           TKTE     = 0.04 ! TRAILING EDGE THICKNESS/AXIAL CHORD.
           TKMAXS   = 0.30 ! STATOR MAXIMUM THICKNESS/AXIAL CHORD.
           TKMAXR   = 0.25 ! ROTOR MAXIMUM THICKNESS/AXIAL CHORD.
```

```
147             XTKMAXS  = 0.45 ! FRACTION OF AXIAL CHORD AT MAXIMUM THICKNESS FOR STATOR
148             XTKMAXR  = 0.40 ! FRACTION OF AXIAL CHORD AT MAXIMUM THICKNESS FOR ROTOR
149             XMODLE   = 0.02 ! FRACTION OF AXIAL CHORD OVER WHICH THE LE IS MODIFIED.
150             XMODTE   = 0.01 ! FRACTION OF AXIAL CHORD OVER WHICH THE TE IS MODIFIED.
151             TK_TYP   = 2.0  ! FORM OF BLADE THICKNESS DISTRIBUTION.
152             ZWEIFEL  = 0.85 ! ZWEIFEL COEFFICIENT FOR TURBINES
153             EXPO     = 1.0  ! EXPONENT FOR TRANSFORMING THE AXIAL POSITION. IT IS USED
                TO
154   C                          VARY THE CAMBER LINE SHAPE. INCREASING EXPO MOVES THE
      BLADE LOADING UPSTREAM.
155             QLE_ROW1(1) = 92.0  ! LEADING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 1.
156             QTE_ROW1(1) = 88.0  ! TRAILING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 1.
157             QLE_ROW2(1) = 88.0  ! LEADING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 2.
158             QTE_ROW2(1) = 92.0  ! TRAILING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 2.
159         END IF
160   C
161         IF(TURBO_TYP.EQ.'C') THEN   !  DEFAULTS FOR COMPRESSORS.
162             TKLE     = 0.02  ! LEADING EDGE THICKNESS/AXIAL CHORD.
163             TKTE     = 0.01  ! TRAILING EDGE THICKNESS/AXIAL CHORD.
164             TKMAXS   = 0.10  ! STATOR MAXIMUM THICKNESS/AXIAL CHORD.
165             TKMAXR   = 0.075 ! ROTOR MAXIMUM THICKNESS/AXIAL CHORD.
166             XTKMAXS  = 0.45  ! FRACTION OF AXIAL CHORD AT MAXIMUM THICKNESS FOR STATOR
167             XTKMAXR  = 0.40  ! FRACTION OF AXIAL CHORD AT MAXIMUM THICKNESS FOR ROTOR
168             XMODLE   = 0.02  ! FRACTION OF AXIAL CHORD OVER WHICH THE LE IS MODIFIED.
169             XMODTE   = 0.01  ! FRACTION OF AXIAL CHORD OVER WHICH THE TE IS MODIFIED.
170             TK_TYP   = 2.0   ! DETERMINES THE SHAPE OF THE BLADE THICKNESS
                DISTRIBUTION.TYPICALLY = 2,
171   C                          LARGER VALUES GIVE MORE UNIFORM THICKNESS.
172             ZWEIFEL  = 0.5   ! ZWEIFEL COEFFICIENT FOR COMPRESSORS.
173             D_FAC    = 0.35  ! DIFFUSION FACTOR FOR COMPRESSORS. THIS IS NOT NOW USED.
174             EXPO     = 1.35  ! EXPONENT FOR TRANSFORMING THE AXIAL POSITION. IT IS
                USED TO
175   C                          VARY THE CAMBER LINE SHAPE. INCREASING EXPO MOVES THE
      BLADE LOADING UPSTREAM.
176             QLE_ROW1(1) = 88.0  ! LEADING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 1.
177             QTE_ROW1(1) = 92.0  ! TRAILING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 1.
178             QLE_ROW2(1) = 92.0  ! LEADING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 2.
179             QTE_ROW2(1) = 88.0  ! TRAILING EDGE ANGLE TO AXIAL DIRECTION IN MERIDINAL
                VIEW ROW 2.
180         END IF
181   C
182         IPROPS      = 1       ! USE PERFECT GAS PROPERTIES.
183         RGAS        = 287.5   ! GAS CONSTANT, VALUE FOR AIR.
184         GAMM        = 1.40    ! GAS SPECIFIC HEAT RATIO, VALUE FOR AIR.
185         AXCHRD1(1)  = 0.05    ! AXIAL CHORD OF ROW 1, METRES.
186         AXCHRD2(1)  = 0.04    ! AXIAL CHORD OF ROW 2, METRES.
187         ROWGAP(1)   = 0.25    ! GAP BETWEEN BLADE ROWS AS A FRACTION OF THE AXIAL CHORD.
188         STAGEGAP(1)= 0.5      ! GAP BETWEEN STAGES AS A FRACTION OF THE AXIAL CHORD.
189         DEVN_1      = 5.0     ! DEVIATION ANGLE FROM ROW 1, DEGREES.
190         DEVN_2      = 5.0     ! DEVIATION ANGLE FRON ROW 2, DEGREES.
191         AINC_1      = -2.0    ! INCIDENCE ANGLE ON ROW 1, DEGREES.
192         AINC_2      = -2.0    ! INCIDENCE ANGLE ON ROW 2, DEGREES.
193         ETA(1)      = 0.9     ! ISENTROPIC EFFICIENCY.
194         NSMOOTH     = 5       ! NUMBER OF SMOOTHINGS OF THE STREAM SURFACE COORDINATES.
195         SFAC        = 0.1     ! SMOOTHING FACTOR FOR THE STREAM SURFACE SMOOTHING.
196         FBLOCK_LE(1) = 0.0    ! BLOCKAGE FACTOR AT FIRST LEADING EDGE.
197         FBLOCK_TE(1) = 0.0    ! BLOCKAGE FACTOR AT SECOND BLADE TRAILING EDGE.
198   C
199         NOSECT  = 3    ! NUMBER OS STREAM SURFACES TO BE GENERATED.
200         IM      = 37   ! NUMBER OF GRID POINTS IN THE PITCHWISE DIRECTION.
201         KM      = 37   ! NUMBER OF GRID POINTS IN THE SPANWISE DIRECTION.
202         NINTUP  = 20   ! NUMBER OF MERIDIONAL GRID POINTS UPSTREAM OF THE LEADING EDGE.
203         NINTON  = 70   ! NUMBER OF MERIDIONAL GRID POINTS ON THE BLADE.
204         NINTDWN = 15   ! NUMBER OF MERIDIONAL GRID POINTS BEHIND THE TRAILING EDGE.
205         NADDUP  = 5    ! EXTRA MERIDIONAL GRID POINTS UPSTREAM OF ROW 1.
206         NADDWN  = 5    ! EXTRA MERIIONAL GRID POINTS DOWNSTREAM OF THE LAST ROW.
```

```
        FPRAT   = 1.25 ! GRID EXPANSION RATIO IN THE PITCHWISE DIRECTION.
        FPMAX   = 20.0 ! MAXIMUM GRID EXPANSION IN THE PITCHWISE DIRECTION.
        FRRAT   = 1.25 ! GRID EXPANSION RATIO IN THE SPANWISE DIRECTION.
        FRMAX   = 20.0 ! MAXIMUM GRID EXPANSION IN THE SPANWISE DIRECTION.
C
C*******************************************************************************
C*******************************************************************************
C   END OF SETTING DEFAULTS.
C*******************************************************************************
C*******************************************************************************
C
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
     & 'INPUT THE GAS CONSTANT IN J/KG K, AND GAS SPECIFIC HEAT RATIO.'
      WRITE(6,*)
     &'THE DEFAULT VALUES ARE THOSE FOR AIR, RGAS = 287.15, GAMMA=1.4.'
      WRITE(6,*) ' TYPE "A" TO ACCEPT THESE,  OR TYPE IN NEW VALUES.'
           READ(5,*,ERR=1111)  RGAS, GAMM
           CONTINUE
           WRITE(10,102) RGAS,GAMM
      FORMAT(2F10.3,T25,' GAS PROPERTOES, RGAS, GAMMA ')
           CPGAS  = RGAS*GAMM/(GAMM-1.)
           FGA    = GAMM/(GAMM - 1.0)
C
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
      WRITE(6,*)
     & 'INPUT THE INLET STAGNATION PRESSURE IN BAR AND INLET TEMPERATURE
     & IN DEG K'
           READ(5,*)    POIN, TOIN
           WRITE(10,103)POIN,TOIN
           FORMAT(2F10.3, T25, ' POIN,  TOIN ')
           HOIN    = CPGAS*TOIN
           PSTAGIN = POIN*1.0E05
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
C
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE NUMBER OF STAGES IN THE MACHINE.'
           READ(5,*)     NSTAGES
           WRITE(10,104) NSTAGES
      FORMAT(I5,T25,  ' NUMBER OF STAGES IN THE MACHINE ')
           WRITE(6,*) ' NUMBER OF STAGES = ', NSTAGES
           NROWS = 2*NSTAGES
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
      WRITE(6,*)    ' BASE THE DESIGN ON THE HUB, MEAN OR TIP RADIUS ? '
      WRITE(6,*)    ' INPUT  "H" ,  "M"   or  "T" '
           READ(5,*)  IFHUB
           IF(IFHUB.EQ.'h') IFHUB = 'H'
           IF(IFHUB.EQ.'m') IFHUB = 'M'
           IF(IFHUB.EQ.'t') IFHUB = 'T'
           WRITE(10,106)IFHUB
      FORMAT(A1,T25, ' CHOICE OF DESIGN POINT RADIUS, HUB, MID or TIP')
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
C
C*******************************************************************
C*******************************************************************
C     CALL PROPS TO GET THE INLET STAGNATION CONDITIONS
C
           HO(1)  = HOIN
           PO(1)  = POIN
           S(1)   = 0.0
           SI     = 0.0
           VIN(1) = 10.0
```

```
C
      CALL PROPS(1,1,HO(1),S(1),PO(1),T(1),RHO(1),WET(1),
     &           VIN(1),G(1),VS(1),1,IPROPS,IWET)
C
C
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
      WRITE(6,*) ' INLET STAGNATION PRESSURE    = ', PO(1), ' BAR.'
      WRITE(6,*) ' INLET STAGNATION TEMPERATURE = ', T(1) , ' K. '
      WRITE(6,*) ' INLET STAGNATION DENSITY     = ', RHO(1) , 'Kg/M3.'
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
C
C
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
      WRITE(6,*)' INPUT THE ROTATION SPEED IN RPM, IT MUST BE POSITIVE.'
           READ(5,*)      RPM
           WRITE(6,*) '  RPM = ', RPM
           WRITE(10,107) RPM
      FORMAT(F12.3,T25, ' ROTATION SPEED, RPM ')
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
C
C
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE REQUIRED INLET MASS FLOW RATE IN kg/sec.'
           READ(5,*)       FLOWIN
           WRITE(10,108)   FLOWIN
      FORMAT(F12.3,T25,' MASS FLOW RATE, FLOWIN. ')
           WRITE(6,*)   ' INLET FLOW = ', FLOWIN
           FLOW = FLOWIN
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
C
      OMEGA       = RPM*PI/30.
      DHO(1)      = 0.0
      DHOTOTAL    = 0.0
C
C*************************************************************************
C*************************************************************************
C   START THE LOOP OVER NSTG STAGES. RETURN TO  "1100"  AFTER EVERY STAGE.
C*************************************************************************
C*************************************************************************
C*************************************************************************
      NSTG = 0
      CONTINUE
C
C*************************************************************************
C*************************************************************************
      NSTG = NSTG + 1
C
      NROW = 2*NSTG -1
C
      WRITE(6,*)
      WRITE(6,*)'*******************************************************
     &*************************************************************'
      WRITE(6,*)'*******************************************************
     &*************************************************************'
      WRITE(6,*) '              STARTING STAGE NUMBER ', NSTG
      WRITE(6,*)'*******************************************************
     &*************************************************************'
      WRITE(6,*)'*******************************************************
     &*************************************************************'
      WRITE(6,*)
```

```
C
C
      IF(NSTG.GT.1) THEN
C
      IFSAME_ALL = 'N'
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      WRITE(6,*) '  ARE THE ANGLES, MASS FLOW, DESIGN RADIUS,'
      WRITE(6,*) '  EFFICIENCY, ETC, FOR THIS STAGE  "ALL" THE SAME AS'
      WRITE(6,*) '  FOR THE LAST STAGE ?'
      WRITE(6,*)
      IF(FLO_TYP.EQ.'AXI') THEN
      WRITE(6,*) '  ANSWER  "Y" or  "N", OR ANSWER  "C"  TO CHANGE FROM'
      WRITE(6,*) '  "FLO_TYP" = "AXI" TO "FLO_TYP" = "MIX".'
      ELSE
      WRITE(6,*) '  ANSWER  "Y" or  "N", OR ANSWER  "C"  TO CHANGE FROM'
      WRITE(6,*) '  "FLO_TYP" = "MIX" TO "FLO_TYP" = "AXI".'
      END IF
C
           READ(5,*)         IFSAME_ALL
           WRITE(6,*)      ' IFSAME_ALL= ', IFSAME_ALL
           WRITE(10,109)     IFSAME_ALL
      IF(IFSAME_ALL.EQ.'y') IFSAME_ALL = 'Y'
      IF(IFSAME_ALL.EQ.'n') IFSAME_ALL = 'N'
      IF(IFSAME_ALL.EQ.'c') IFSAME_ALL = 'C'
C
      IF(IFSAME_ALL.EQ.'C'.AND.(FLO_TYP.EQ.'AXI'))THEN
           FLO_TYP = 'MIX'
           GO TO 2000
      END IF
      IF(IFSAME_ALL.EQ.'C'.AND.(FLO_TYP.EQ.'MIX'))THEN
           FLO_TYP = 'AXI'
           MIXTYP  = 'N'
           GO TO 500
      END IF
C
      FORMAT(A1,T25,' IFSAME_ALL, SET = "Y" TO REPEAT THE LAST STAGE INP
     &UT TYPE AND VELOCITY TRIANGLES, SET = "C" TO CHANGE INPUT TYPE.')
C
      IF(FLO_TYP.EQ.'AXI'.AND.(IFSAME_ALL.EQ.'Y')) GO TO 600
      IF(FLO_TYP.EQ.'MIX'.AND.(IFSAME_ALL.EQ.'Y')) GO TO 700
C
C   END OF NSTAGE GT 1 LOOP
      END IF
C

C    RE ENTER HERE IF CHANGING THE ANGLES FOR THIS STAGE.
C
      CONTINUE
C
C*****************************************************************************
C*****************************************************************************
C
      IF(FLO_TYP.EQ.'MIX') GO TO 2000
C
C*****************************************************************************
C*****************************************************************************
C
C    SET THE VELOCITY TRIANGLES FOR THE STAGE FOR FLO_TYP = "AXI".
C
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      WRITE(6,*)   'YOU MAY SPECIFY THE STAGE VELOCITY TRIANGLES IN ONE
     &OF 3 WAYS '
      WRITE(6,*)
      WRITE(6,*) ' METHOD "A"-SPECIFY THE REACTION, FLOW COEFFICIENT AND
     & STAGE LOADING COEFFICIENT'
      WRITE(6,*)
```

```
      WRITE(6,*) ' METHOD "B"- SPECIFY THE FLOW COEFFICIENT, THE STATOR
     &EXIT ANGLE AND THE ROTOR EXIT ANGLE'
      WRITE(6,*)
      WRITE(6,*) ' METHOD "C"- SPECIFY THE FLOW COEFFICIENT, THE ROTOR
     &INLET ANGLE AND THE ROTOR EXIT ANGLE'
      WRITE(6,*)
      WRITE(6,*) ' METHOD "D" SPECIFY THE STAGE REACTION,THE FIRST BLADE
     &ROW INLET ANGLE AND THE FIRST BLADE ROW EXIT ANGLE'
      WRITE(6,*)
C
      WRITE(6,*)
      WRITE(6,*) '                    CHOOSE YOUT INPUT METHOD.'
      WRITE(6,*) '                TYPE IN  "A",  "B", "C"  or "D" '
C
           READ(5,*) INTYPE
           IF(INTYPE.EQ.'a') INTYPE ='A'
           IF(INTYPE.EQ.'b') INTYPE ='B'
           IF(INTYPE.EQ.'c') INTYPE ='C'
           IF(INTYPE.EQ.'d') INTYPE ='D'
C
           WRITE(10,110) INTYPE
      FORMAT(A1,T25,' INTYPE, TO CHOOSE THE METHOD OF DEFINING THE VELOC
     &ITY TRIANGLES')
C
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
C
C
*****************************************************************************************
C
*****************************************************************************************
C
      IF(INTYPE.EQ.'A') THEN
      WRITE(6,*) ' INTYPE = "A" CHOSEN.'
C
      IF(TURBO_TYP.EQ.'T')  WRITE(6,*) '  THIS IS A TURBINE.'
      IF(TURBO_TYP.EQ.'C')  WRITE(6,*) '  THIS IS A COMPRESSOR.'
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE REACTION, FLOW COEFFICIENT AND STAGE LOADIN
     &G COEFFICIENT. '
           READ(5,*)     REACN(NSTG), PHI(NSTG), PSI(NSTG)
           WRITE(6,*)    REACN(NSTG), PHI(NSTG), PSI(NSTG)
           WRITE(10,111)REACN(NSTG), PHI(NSTG), PSI(NSTG)
      FORMAT(3F7.3,T25,' REACTION, FLOW COEFF., LOADING COEFF.')
C
      IF(TURBO_TYP.EQ.'T') THEN
      BIN_ROW1(NSTG)  =
     &                ATAN( (1.- REACN(NSTG) - 0.5*PSI(NSTG))/PHI(NSTG))
      BOUT_ROW2(NSTG) =
     &                ATAN( TAN(BIN_ROW1(NSTG))  - 1.0/PHI(NSTG) )
      BOUT_ROW1(NSTG) =
     &                ATAN( TAN(BIN_ROW1(NSTG))  +  PSI(NSTG)/PHI(NSTG))
      BIN_ROW2(NSTG)  =
     &                ATAN( TAN(BOUT_ROW1(NSTG))  - 1.0/PHI(NSTG) )
      WRITE(6,*)
      WRITE(6,*) ' CALCULATED FLOW ANGLES FOR THIS STAGE '
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      END IF
C
      IF(TURBO_TYP.EQ.'C') THEN
      BIN_ROW2(NSTG)  =
     &                ATAN( (1. -REACN(NSTG) + 0.5*PSI(NSTG))/PHI(NSTG))
      BOUT_ROW1(NSTG) =
     &                ATAN( TAN(BIN_ROW2(NSTG))  - 1.0/PHI(NSTG) )
      BIN_ROW1(NSTG)  =
     &                ATAN( TAN(BOUT_ROW1(NSTG)) - PSI(NSTG)/PHI(NSTG))
      BOUT_ROW2(NSTG) =
     &                ATAN( TAN(BIN_ROW1(NSTG))  + 1.0/PHI(NSTG))
      WRITE(6,*)
      WRITE(6,*) ' CALCULATED FLOW ANGLES FOR THIS STAGE '
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
```

```
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      END IF
C   END OF INTYPE = "A" .
      END IF
C
C
*********************************************************************************
C
*********************************************************************************
C
      IF(INTYPE.EQ.'B') THEN
      WRITE(6,*) ' INTYPE = "B" CHOSEN. '
C
      IF(TURBO_TYP.EQ.'T')  WRITE(6,*) '  THIS IS A TURBINE.'
      IF(TURBO_TYP.EQ.'C')  WRITE(6,*) '  THIS IS A COMPRESSOR.'
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE FLOW COEFFICIENT, STATOR EXIT ANGLE AND ROT
     &OR EXIT ANGLE'
           READ(5,*)      PHI(NSTG), B2NOZ, B2ROT
           WRITE(10,112) PHI(NSTG), B2NOZ, B2ROT
      FORMAT(3F12.3, T25, ' FLOW COEFF, STATOR ANGLES ')
C
      IF(TURBO_TYP.EQ.'T') THEN
      BOUT_ROW1(NSTG)  = B2NOZ*DEGRAD
      BOUT_ROW2(NSTG)  = B2ROT*DEGRAD
      BIN_ROW2(NSTG)   = ATAN( TAN(BOUT_ROW1(NSTG)) - 1.0/PHI(NSTG) )
      BIN_ROW1(NSTG)   = ATAN( TAN(BOUT_ROW2(NSTG)) + 1.0/PHI(NSTG) )
      REACN(NSTG)    =
     &    -0.5*PHI(NSTG)*(TAN(BIN_ROW2(NSTG)) + TAN(BOUT_ROW2(NSTG)))
      PSI(NSTG)      =
     &          2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW1(NSTG)))
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
      END IF
C
      IF(TURBO_TYP.EQ.'C') THEN
      BOUT_ROW2(NSTG) = B2NOZ*DEGRAD
      BOUT_ROW1(NSTG) = B2ROT*DEGRAD
      BIN_ROW1(NSTG)  = ATAN(  TAN(BOUT_ROW2(NSTG))  - 1.0/PHI(NSTG))
      BIN_ROW2(NSTG)  = ATAN(  TAN(BOUT_ROW1(NSTG))  + 1.0/PHI(NSTG))
      REACN(NSTG)     =
     &    -0.5*PHI(NSTG)*(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
      PSI(NSTG)       =
     &         -2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW2(NSTG)))
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
      END IF
C
C     END OF INTYPE =  "B" .
      END IF
C
C
*********************************************************************************
C
*********************************************************************************
C
      IF(INTYPE.EQ.'C') THEN
      WRITE(6,*) ' INPUT TYPE "C" CHOSEN.'
C
      IF(TURBO_TYP.EQ.'T')  WRITE(6,*) '  THIS IS A TURBINE.'
      IF(TURBO_TYP.EQ.'C')  WRITE(6,*) '  THIS IS A COMPRESSOR.'
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE ROTOR INLET ANGLE, THE ROTOR EXIT ANGLE AND
     & THE FLOW COEFFICIENT.'
           READ(5,*)      B1ROT, B2ROT, PHI(NSTG)
           WRITE(10,113) B1ROT, B2ROT, PHI(NSTG)
      FORMAT(3F12.3,T25,' ROTOR ANGLES, FLOW COEFF.')
C
      IF(TURBO_TYP.EQ.'T') THEN
      BIN_ROW2(NSTG)  = B1ROT*DEGRAD
      BOUT_ROW2(NSTG) = B2ROT*DEGRAD
```

```
      BIN_ROW1(NSTG)  = ATAN(  TAN(BOUT_ROW2(NSTG)) + 1.0/PHI(NSTG))
      BOUT_ROW1(NSTG) = ATAN(  TAN(BIN_ROW2(NSTG))  + 1.0/PHI(NSTG))
      REACN(NSTG)     =
     &   -0.5*PHI(NSTG)*(TAN(BIN_ROW2(NSTG)) + TAN(BOUT_ROW2(NSTG)))
      PSI(NSTG)       =
     &         2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW1(NSTG)))
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
      END IF
C
      IF(TURBO_TYP.EQ.'C') THEN
      BIN_ROW1(NSTG)  = B1ROT*DEGRAD
      BOUT_ROW1(NSTG) = B2ROT*DEGRAD
      BIN_ROW2(NSTG)  = ATAN(  TAN(BOUT_ROW1(NSTG))  +  1.0/PHI(NSTG))
      BOUT_ROW2(NSTG) = ATAN(  TAN(BIN_ROW1(NSTG))   +  1.0/PHI(NSTG))
      REACN(NSTG)     =
     &     -0.5*PHI(NSTG)*(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
      PSI(NSTG)       =
     &          -2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW2(NSTG)))
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
      END IF
C   END OF INTYPE = "C" OPTION.
      END IF
C
C
C*****************************************************************************
C
C*****************************************************************************
C
      IF(INTYPE.EQ.'D') THEN
      WRITE(6,*) 'INPUT TYPE "D" CHOSEN.'
C
      IF(TURBO_TYP.EQ.'T')  WRITE(6,*) '  THIS IS A TURBINE.'
      IF(TURBO_TYP.EQ.'C')  WRITE(6,*) '  THIS IS A COMPRESSOR.'
      WRITE(6,*)
      WRITE(6,*)' INPUT THE FIRST BLADE ROW INLET ANGLE, THE FIRST BLADE
     & ROW EXIT ANGLE AND THE STAGE REACTION.'
           READ(5,*) BIN_ROW1(NSTG), BOUT_ROW1(NSTG), REACN(NSTG)
           WRITE(10,114) BIN_ROW1(NSTG), BOUT_ROW1(NSTG), REACN(NSTG)
      FORMAT(3F12.3,T25,' FIRST ROW ANGLES, REACTION')
           BIN_ROW1(NSTG)  = BIN_ROW1(NSTG)*DEGRAD
           BOUT_ROW1(NSTG) = BOUT_ROW1(NSTG)*DEGRAD
C
      IF(TURBO_TYP.EQ.'T') THEN
      PHI(NSTG) =
     &  2*(1.0-REACN(NSTG))/(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
      BIN_ROW2(NSTG)  = ATAN(  TAN(BOUT_ROW1(NSTG))  -  1.0/PHI(NSTG))
      BOUT_ROW2(NSTG) = ATAN(  TAN(BIN_ROW1(NSTG))   -  1.0/PHI(NSTG))
      PSI(NSTG)       =
                2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW1(NSTG)))
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' PHI= ', PHI(NSTG), ' LOADING = ', PSI(NSTG)
      END IF
C
      IF(TURBO_TYP.EQ.'C') THEN
      PHI(NSTG) =
     &   -2*REACN(NSTG)/(TAN(BIN_ROW1(NSTG))   + TAN(BOUT_ROW1(NSTG)))
      BIN_ROW2(NSTG)  = ATAN(  TAN(BOUT_ROW1(NSTG))  +  1.0/PHI(NSTG))
      BOUT_ROW2(NSTG) = ATAN(  TAN(BIN_ROW1(NSTG))   +  1.0/PHI(NSTG))
      PSI(NSTG)       =
     &          -2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW2(NSTG)))
      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
     &           BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' PHI= ', PHI(NSTG), ' LOADING = ', PSI(NSTG)
      END IF
C  END OF INTYPE = "D" OPTION.
      END IF
C
C*****************************************************************************
```

```
C*************************************************************************
C
C    NEXT SET THE ROTATIONAL SPEED AND DESIGN RADIUS. FOR FLO_TYP = "AXI" .
C
C*************************************************************************
C*************************************************************************
C
      WRITE(6,*)
      WRITE(6,*) ' YOU MAY SET THE DESIGN RADIUS IN ONE OF 2 WAYS'
      WRITE(6,*)
      WRITE(6,*) ' METHOD "A".  INPUT THE DESIGN RADIUS DIRECTLY.'
      WRITE(6,*) ' METHOD "B".  INPUT STAGE ENTHALPY CHANGE.'
      WRITE(6,*)
      WRITE(6,*) ' CHOOSE METHOD "A" OR "B".'
      WRITE(6,*) ' TYPE IN "A" or "B". '
      WRITE(6,*)
           READ(5,*)  RADTYPE
           IF(RADTYPE.EQ.'a') RADTYPE = 'A'
           IF(RADTYPE.EQ.'b') RADTYPE = 'B'
           WRITE(6,*) ' RADTYPE = ', RADTYPE
           WRITE(10,115) RADTYPE
      FORMAT(A1,T25,' RADTYPE, TO CHOOSE THE DESIGN POINT RADIUS')
C
      IF(RADTYPE.EQ.'A') THEN
      WRITE(6,*)     ' INPUT THE DESIGN RADIUS IN METRES.'
           READ(5,*)  RDES(NSTG)
           WRITE(6,*) ' DESIGN POINT RADIUS =', RDES(NSTG)
           WRITE(10,116)  RDES(NSTG)
      FORMAT(F12.3,T25, ' THE DESIGN POINT RADIUS ')
           U(NSTG)   = RDES(NSTG)*OMEGA
           DHO(NSTG) = PSI(NSTG)*U(NSTG)*U(NSTG)
      END IF
C
      IF(RADTYPE.EQ.'B') THEN
      WRITE(6,*) ' INPUT THE STAGE ACTUAL ENTHALPY CHANGE IN KJ/Kg. '
      WRITE(6,*) ' THE ENTHALPY CHANGE IS ALWAYS TREATED AS POSITIVE.'
            READ(5,*)     DHO(NSTG)
            WRITE(6,*)    DHO(NSTG)
            WRITE(10,117) DHO(NSTG)
      FORMAT(F12.3,T25, '  STAGE ENTHALPY CHANGE, KJ/Kg')
            DHO(NSTG)  = 1000.0*DHO(NSTG)
            U(NSTG)    = SQRT(DHO(NSTG)/PSI(NSTG))
            RDES(NSTG) = U(NSTG)/OMEGA
      END IF
C
      IF(TURBO_TYP.EQ.'T') DHO(NSTG) = -DHO(NSTG)
C
C*************************************************************************
C*************************************************************************
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      WRITE(6,*) ' STAGE NUMBER. ', NSTG
      WRITE(6,*) ' INPUT THE FIRST AND SECOND BLADE ROW AXIAL CHORDS IN
     & METRES '
      WRITE(6,*) ' THE CURRENT VALUES ARE ', AXCHRD1(NSTG),AXCHRD2(NSTG)
      WRITE(6,*) ' Press  A  to accept these, or type in new values.'
           READ(5,*,ERR=20,END=20)  AXCHRD1(NSTG),AXCHRD2(NSTG)
           WRITE(6,*) ' STATOR AND ROTOR AXIAL CHORDS ARE ',
     &                  AXCHRD1(NSTG),AXCHRD2(NSTG)
           WRITE(10,120) AXCHRD1(NSTG),AXCHRD2(NSTG)
      FORMAT(2F12.3,T25, ' BLADE AXIAL CHORDS IN METRES.')
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
C
C*************************************************************************
C*************************************************************************
C
      WRITE(6,*) ' INPUT THE INTER ROW GAP AND INTER STAGE GAP AS FRACTI
     &ONS OF THE FIRST ROW AXIAL CHORD '
      WRITE(6,*) ' THE CURRENT VALUES ARE ', ROWGAP(NSTG),STAGEGAP(NSTG)
```

```
      WRITE(6,*) ' Press  A  to accept these, or type in new values.'
      READ(5,*,ERR=218,END=218) ROWGAP(NSTG), STAGEGAP(NSTG)
      CONTINUE
      WRITE(6,*) ' ROWGAP =', ROWGAP(NSTG),'STAGEGAP = ', STAGEGAP(NSTG)
      WRITE(10,119) ROWGAP(NSTG),STAGEGAP(NSTG)
      FORMAT(2F12.3,T25,' ROW GAP  AND STAGE GAP ')
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
C
C*******************************************************************************
C*******************************************************************************
C   JUMP TO HERE IF IFSAME_ALL = "Y" .
C   SET THE AXIAL POSITIONS OF THE BLADES FOR IN_TYPE = "AXI" .
C
      CONTINUE
C
      IF(NSTG.EQ.1) THEN
           XMEAN(1) = -1.0*AXCHRD1(NSTG)
           XMEAN(2) = -0.5*AXCHRD1(NSTG)
           XMEAN(3) = 0.0
           XMEAN(4) = XMEAN(3) + AXCHRD1(NSTG)
           XMEAN(5) = XMEAN(4) + 0.5*ROWGAP(NSTG)*AXCHRD1(NSTG)
           XMEAN(6) = XMEAN(4) +     ROWGAP(NSTG)*AXCHRD1(NSTG)
           XMEAN(7) = XMEAN(6) + AXCHRD2(NSTG)
           XMEAN(8) = XMEAN(7) + 0.5*STAGEGAP(NSTG)*AXCHRD2(NSTG)
           XMEAN(9) = XMEAN(7) +     STAGEGAP(NSTG)*AXCHRD2(NSTG)
      IF(NSTG.EQ.NSTAGES) XMEAN(8) = XMEAN(7) + 0.5*AXCHRD2(NSTG)
      IF(NSTG.EQ.NSTAGES) XMEAN(9) = XMEAN(7) +     AXCHRD2(NSTG)
      ELSE
C
C    SET THE LOCATION OF THE FIRST POINT IF THE INPUT TYPE HAS BEEN CHANGED FROM "MIX"
 TO "AXI"
      XMEAN1 = XMEAN(7)
      IF(IFSAME_ALL.EQ.'C')  XMEAN1 = XMEANTE
C
           XMEAN(1) = XMEAN1   +  0.333*STAGEGAP(NSTG)*AXCHRD1(NSTG)
           XMEAN(2) = XMEAN1   +  0.667*STAGEGAP(NSTG)*AXCHRD1(NSTG)
           XMEAN(3) = XMEAN1   +        STAGEGAP(NSTG)*AXCHRD1(NSTG)
           XMEAN(4) = XMEAN(3) + AXCHRD2(NSTG)
           XMEAN(5) = XMEAN(4) + 0.5*ROWGAP(NSTG)*AXCHRD2(NSTG)
           XMEAN(6) = XMEAN(4) +     ROWGAP(NSTG)*AXCHRD2(NSTG)
           XMEAN(7) = XMEAN(6) + AXCHRD2(NSTG)
           XMEAN(8) = XMEAN(7) + 0.5*STAGEGAP(NSTG)*AXCHRD2(NSTG)
           XMEAN(9) = XMEAN(8) +     STAGEGAP(NSTG)*AXCHRD2(NSTG)
      IF(NSTG.EQ.NSTAGES) XMEAN(8) = XMEAN(7) + 0.5*AXCHRD2(NSTG)
      IF(NSTG.EQ.NSTAGES) XMEAN(9) = XMEAN(7) + AXCHRD2(NSTG)
C
      END IF
C
      DO N=1,9
            RMEAN(N) = RDES(NSTG)
            VMRAT(N) = 1.0
      END DO
C
      NSS  = 9
      NLE1 = 3
      NTE1 = 4
      NLE2 = 6
      NTE2 = 7
      PHI_REF = PHI(NSTG)
C
      WRITE(6,*)
      WRITE(6,*) ' MEAN LINE COORDINATES, STAGE NUMBER',NSTG
      WRITE(6,*) ' XMEAN     ',(XMEAN(N),N=1,NSS)
      WRITE(6,*) ' RMEAN     ',(RMEAN(N),N=1,NSS)
      WRITE(6,*) ' VMRAT     ',(VMRAT(N),N=1,NSS)
      WRITE(6,*)
      WRITE(6,*) 'NLE1,NTE1,NLE2,NTE2', NLE1,NTE1,NLE2,NTE2
      WRITE(6,*)

C
C*******************************************************************************
```

```
C*************************************************************************
C     GO TO 3000 IF FLO_TYP = "AXI"
C
      GO TO 3000
C
C*************************************************************************
C*************************************************************************
C     RE ENTER HERE IF FLO_TYP = "MIX"
C
      CONTINUE
C
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************
C     NOW INPUT DATA FOR  FLO_TYP = "MIX" .
C
C*************************************************************************
C*************************************************************************
      WRITE(6,*)' FOR FLO_TYP= "MIX" YOU HAVE A CHOICE OF TWO INPUT METH
     &ODS'
      WRITE(6,*)
      WRITE(6,*)' EITHER INPUT ALL 4 BLADE ANGLES.'
      WRITE(6,*)
      WRITE(6,*)'OR'
      WRITE(6,*)
      WRITE(6,*)'INPUT THE ABSOLUTE FLOW ANGLES AT STAGE INLET AND EXIT'
      WRITE(6,*)'AND THE FLOW COEFFICIENT AND STAGE LOADING COEFFICIENT'
      WRITE(6,*)
      WRITE(6,*)' INPUT  "A"  FOR THE FIRST METHOD,  "B" FOR THE SECOND'
      READ(5,*)     MIXTYP
      WRITE(10,410) MIXTYP
      FORMAT(A1,T25,' MIXTYP = INPUT TYPE FOR FLO_TYP = "MIX" .')
      IF(MIXTYP.EQ.'a') MIXTYP = 'A'
      IF(MIXTYP.EQ.'b') MIXTYP = 'B'
C
C*************************************************************************
C*************************************************************************
C
      IF(MIXTYP.EQ.'A') THEN
C
      WRITE(6,*) ' INPUT THE STATOR INLET AND EXIT FLOW ANGLES, IN DEG.'
      READ(5,*)     STATOR_IN, STATOR_OUT
      WRITE(10,411) STATOR_IN, STATOR_OUT
      FORMAT(2F10.3,T25,' ANGLES, STATOR_IN, STATOR_OUT')
C
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE ROTOR INLET AND EXIT RELATIVE FLOW ANGLES,
     & IN DEGREES.'
      READ(5,*)     ROTOR_IN, ROTOR_OUT
      WRITE(10,412) ROTOR_IN, ROTOR_OUT
      FORMAT(2F10.3,T25,'ANGLES, ROTOR_IN, ROTOR_OUT')
C
      IF(TURBO_TYP.EQ.'C') THEN
           BIN_ROW1(NSTG)  = ROTOR_IN*DEGRAD
           BOUT_ROW1(NSTG) = ROTOR_OUT*DEGRAD
           BIN_ROW2(NSTG)  = STATOR_IN*DEGRAD
           BOUT_ROW2(NSTG) = STATOR_OUT*DEGRAD
           PHI_STG1  = 1.0/(TAN(BIN_ROW2(NSTG))
     &               - TAN(BOUT_ROW1(NSTG)))
      ELSE
           BIN_ROW1(NSTG)  = STATOR_IN*DEGRAD
           BOUT_ROW1(NSTG) = STATOR_OUT*DEGRAD
           BIN_ROW2(NSTG)  = ROTOR_IN*DEGRAD
           BOUT_ROW2(NSTG) = ROTOR_OUT*DEGRAD
           PHI_STG1  = 1.0/(TAN(BOUT_ROW1(NSTG))
     &               - TAN(BIN_ROW2(NSTG)))
      END IF
C
C   END OF MIXTYP = "A"  LOOP.
      END IF
C*************************************************************************
C*************************************************************************
```

```
C*************************************************************************
C*************************************************************************

      IF(MIXTYP.EQ.'B') THEN
C
      IF(NSTG.EQ.1) THEN
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE FLOW COEFFICIENT AT THE FIRST ROTOR LEADING
     & EDGE.'
      READ(5,*)     PHI_REF
      WRITE(10,82) PHI_REF
      FORMAT(F10.4,T25,
     &           ' FLOW COEFFICIENT AT THE FIRST ROTOR LEADING EDGE.')
      WRITE(6,*) ' FLOW COEFFICIENT = ', PHI_REF
      WRITE(6,*)
      END IF
C
C*************************************************************************
C*************************************************************************
C
      WRITE(6,*) ' INPUT THE INLET AND EXIT ABSOLUTE FLOW ANGLES FOR THE
     & WHOLE STAGE.'
      READ(5,*)        ALPHA_IN(NSTG), ALPHA_OUT(NSTG)
      WRITE(10,81)     ALPHA_IN(NSTG), ALPHA_OUT(NSTG)
      FORMAT(2F10.3,T25,' STAGE INLET AND OUTLET ABSOLUTE FLOW ANGLES.')
      WRITE(6,*) ' STAGE INLET & EXIT ANGLES',
     &                  ALPHA_IN(NSTG), ALPHA_OUT(NSTG)
      ALPHA_IN(NSTG)  = ALPHA_IN(NSTG)*DEGRAD
      ALPHA_OUT(NSTG) = ALPHA_OUT(NSTG)*DEGRAD
C
C*************************************************************************
C*************************************************************************
C
      WRITE(6,*)
      WRITE(6,*)' INPUT THE STAGE LOADING COEFFICIENT BASED ON THE BLADE
     & SPEED AT THE ROTOR LEADING EDGE '
      READ(5,*)    PSI(NSTG)
      WRITE(10,83) PSI(NSTG)
      FORMAT(F10.4,T25, ' STAGE LOADING COEFFICIENT AT THE ROTOR LEADING
     & EDGE.')
      WRITE(6,*) ' STAGE LOADING COEFFICIENT = ', PSI(NSTG)
C
C  END OF MIXTYP = "B" LOOP.
      END IF
C
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************
C    RE ENTER HERE  IF FLO_TYP = "MIX"  AND THE ANGLES, ETC,
C    WERE THE SAME AS FOR THE LAST STAGE.
C
      CONTINUE
C*************************************************************************
C*************************************************************************
C   NOW INPUT THE STREAM SURFACE COORDINATES. FOR FLO_TYP = "MIX" .
C
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)' THE STREAM SURFACE COORDINATES AND MERIDIONAL VELOCITY
     & RATIOS MUST NOW BE INPUT '
C
      WRITE(6,*)
      WRITE(6,*)     ' THE NEW VALUES MUST FORM A SMOOTH CONTINUATION OF
     & THE LAST STREAM SURFACE '
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
C
      WRITE(6,*)
      WRITE(6,*)'INPUT THE NUMBER OF POINTS ON THE MEAN STREAM SURFACE'
      IF(NSTG.GT.1) THEN
```

```
927         WRITE(6,*)'LAST NSS   = ',    NSS
928         WRITE(6,*)'PRESS "A" TO ACCEPT THE PREVIOUS VALUE OR TYPE IN A NEW
929        & VALUE'
930         END IF
931         READ(5,*,ERR=134) NSS
932         CONTINUE
933         WRITE(6,*) ' NSS = ', NSS
934   C
935         WRITE(6,*)
936         WRITE(6,*) 'INPUT ',NSS, ' AXIAL COORDINATES OF THE MEAN SS '
937         IF(NSTG.GT.1) THEN
938         WRITE(6,*) 'LAST XMEAN = ',    (XMEAN(NP),NP=1,NSS )
939         WRITE(6,*) 'PRESS "A" TO ACCEPT THE PREVIOUS VALUES OR TYPE IN NEW
940        & VALUES'
941         END IF
942         IF(ANSIN.EQ.'F') READ(5,*) DUMMY_LINE
943         READ(5,*,ERR=135) (XMEAN(NP),NP=1,NSS)
944         CONTINUE
945         WRITE(6,*) ' XMEAN=',(XMEAN(NP),NP=1,NSS )
946   C
947         WRITE(6,*)
948         WRITE(6,*)' INPUT ',NSS, ' RADIAL COORDINATES OF THE MEAN SS '
949         IF(NSTG.GT.1) THEN
950         WRITE(6,*)' LAST RMEAN = ',    (RMEAN(NP),NP=1,NSS )
951         WRITE(6,*)' PRESS "A" TO ACCEPT THE PREVIOUS VALUES OR TYPE IN NEW
952        & VALUES.'
953         END IF
954         IF(ANSIN.EQ.'F') READ(5,*) DUMMY_LINE
955         READ(5,*,ERR=136) (RMEAN(NP),NP=1,NSS )
956         CONTINUE
957         WRITE(6,*) ' RMEAN=',(RMEAN(NP),NP=1,NSS )
958   C
959         WRITE(6,*)
960         WRITE(6,*) ' INPUT ',NSS, ' MERIDIONAL VELOCITY RATIOS ON THE MEAN
961        & STREAM SURFACE.'
962         WRITE(6,*)' THESE ARE RELATIVE TO THE VALUE AT THE LEADING EDGE OF
963        & THE FIRST ROTOR.'
964         IF(NSTG.GT.1) THEN
965         WRITE(6,*) 'LAST VMRAT = ',    (VMRAT(NP),NP=1,NSS )
966         WRITE(6,*) 'PRESS "A" TO ACCEPT THE PREVIOUS VALUES OR TYPE IN NEW
967        & VALUES'
968         END IF
969         IF(ANSIN.EQ.'F') READ(5,*) DUMMY_LINE
970         READ(5,*,ERR=137) (VMRAT(NP),NP=1,NSS )
971         CONTINUE
972         WRITE(6,*) ' VMRAT=', (VMRAT(NP),NP=1,NSS )
973   C
974         WRITE(6,*)
975         WRITE(6,*)    ' INPUT THE POINT NUMBERS OF THE LEADING AND TRAILING
976        & EDGES, 4 POINTS IN TOTAL.'
977         IF(NSTG.GT.1) THEN
978         WRITE(6,*) ' THE PREVIOUS VALUES WERE ',NLE1,NTE1,NLE2,NTE2
979         WRITE(6,*) ' PRESS "A" TO ACCEPT THESE OR TYPE IN NEW VALUES'
980         END IF
981         READ(5,*,ERR = 139)      NLE1,NTE1,NLE2,NTE2
982         CONTINUE
983         WRITE(6,*)   ' NLE1,NTE1,NLE2,NTE2 ', NLE1,NTE1,NLE2,NTE2
984         WRITE(6,*)
985   C
986         WRITE(6,*)
987         WRITE(6,*)'  NEW XMEAN = ',    (XMEAN(NP),NP=1,NSS )
988         WRITE(6,*) ' NEW RMEAN = ',    (RMEAN(NP),NP=1,NSS )
989         WRITE(6,*) ' NEW VMRAT = ',    (VMRAT(NP),NP=1,NSS )
990         WRITE(6,*) ' NEW NLE1,ETC',     NLE1,NTE1,NLE2,NTE2
991   C
992   C     SET THE LAST POINT COORDINATES FOR USE IF CHANGING FROM "MIX" TO "AXI"
993         XMEANTE = XMEAN(NTE2)
994         WRITE(6,*) ' SETTING XMEANTE = ', XMEANTE
995   C
996   C
997         ANSSS = 'N'
998         WRITE(6,*) ' DO YOU WANT TO CHANGE THE NEW STREAM SURFACE COORDINA
999        &TES ? '
```

```
      WRITE(6,*)  ' ANSWER  "Y"  or "N" '
      READ(5,*)     ANSSS
      WRITE(6,*)  ' ANSSS =', ANSSS
C
      IF(ANSSS.EQ.'Y'.OR.ANSSS.EQ.'y') GO TO 700
C
      WRITE(10,128) NSS
      WRITE(10,*)'THE FOLLOWING LINE OF DATA CONTAINS THE STREAM SURFACE
     & AXIAL COORDINATES.'
      WRITE(10,129) (XMEAN(NP),NP=1,NSS )
      WRITE(10,*)'THE FOLLOWING LINE OF DATA CONTAINS THE STREAM SURFACE
     & RADIAL COORDINATES.'
      WRITE(10,129) (RMEAN(NP),NP=1,NSS )
      WRITE(10,*)'THE FOLLOWING LINE OF DATA CONTAINS THE MERIDIONAL VEL
     &OCITY RATIOS.'
      WRITE(10,129) (VMRAT(NP),NP=1,NSS )
      WRITE(10,130) NLE1,NTE1,NLE2,NTE2
      WRITE(10,138) ANSSS
      FORMAT(A1,T25, ' DO YOU WANT TO CHANGE THE STREAM SURFACE COORDINA
     &TES ?')
  128 FORMAT(I5,T25, ' NUMBER OF POINTS ON THE STREAM SURFACE.')
      FORMAT(8F10.4)
      FORMAT(4I5,T25,' LEADING AND TRAILING EDGE POINTS ON THE MEAN STRE
     &AM SURFACE.')
C
C*************************************************************************
C*************************************************************************
C*************************************************************************
C*************************************************************************
C    RE ENTER HERE IF FLO_TYP = "AXI"
C
      CONTINUE
C
C*************************************************************************
C*************************************************************************
C     FOR BOTH VALUES  OF "FLO_TYP"
C     CALCULATE THE STREAM SURFACE DISTANCES.
C
      SDIST(1) = 0.0
      DO NP = 2,NSS
      XDIF  = XMEAN(NP) - XMEAN(NP-1)
      RDIF  = RMEAN(NP) - RMEAN(NP-1)
      SDIF  = SQRT(XDIF*XDIF + RDIF*RDIF)
      IF(SDIF.LT.1.001*ABS(RDIF)) SDIF = 1.001*ABS(RDIF)
      SDIST(NP) = SDIST(NP-1) + SDIF
      END DO
C
C*************************************************************************
C*************************************************************************
C
      WRITE(6,*) 'INPUT THE BLOCKAGE FACTOR AT THE LEADING EDGES OF THE
     &FIRST BLADE ROW AND AT THE TRAILING EDGE OF THE'
      WRITE(6,*) 'SECOND BLADE ROW.'
      WRITE(6,*) 'THIS IS THE SUM OF THE DISPLACEMENT THICKNESSES OF THE
     & HUB AND CASING BOUNDARY LAYERS DIVIDED BY THE BLADE SPAN. '
      WRITE(6,*) 'THE CURRENT VALUES ARE ', FBLOCK_LE(NSTG),
     &            FBLOCK_TE(NSTG)
      WRITE(6,*) 'Press  "A"  to accept these, or type in new values.'
      READ(5,*, ERR= 219,END=219) FBLOCK_LE(NSTG), FBLOCK_TE(NSTG)
      CONTINUE
      WRITE(6,*)  ' FBLOCK_LE = ', FBLOCK_LE(NSTG), ' FBLOCK_TE = ',
     &              FBLOCK_TE(NSTG)
      WRITE(10,220) FBLOCK_LE(NSTG), FBLOCK_TE(NSTG)
      FORMAT(2F10.5,T25,' BLOCKAGE FACTORS, FBLOCK_LE,  FBLOCK_TE ')
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
C
C*************************************************************************
C*************************************************************************
C
C  SET THE BLOCKAGE FACTORS
C  MAKE THE BLOCKAGE VARY LINEARLY WITH STREAM SURFACE  DISTANCE.
```

```
      DBLOCK_DS = (FBLOCK_TE(NSTG)-FBLOCK_LE(NSTG))
     &           /(SDIST(NTE2)- SDIST(NLE1))
      DO N = 1,NSS
            FBLOCK(N)= FBLOCK_LE(NSTG)+DBLOCK_DS*(SDIST(N)-SDIST(NLE1))
      END DO
C
      WRITE(6,*)
      WRITE(6,*) ' BLOCKAGE FACTOR THROUGH THE STAGE '
      WRITE(6,*)  (FBLOCK(N),N=1,NSS)
      WRITE(6,*)
C
C    CALCULATE THE MERIDIONAL PITCH ANGLES.
      DO NP = 2,NSS-1
      PITCH_ANGL(NP) = ASIN( (RMEAN(NP+1) - RMEAN(NP-1))
     &               /(SDIST(NP+1)-SDIST(NP-1)) )
      END DO
      PITCH_ANGL(1)   = ASIN( (RMEAN(2)-RMEAN(1))/(SDIST(2)-SDIST(1)) )
      PITCH_ANGL(NSS) = ASIN( (RMEAN(NSS)-RMEAN(NSS-1))
     &                  /(SDIST(NSS)-SDIST(NSS-1)) )
C
C
      WRITE(6,*) 'STREAM SURFACE COORDINATES AND SLOPE BEFORE SMOOTHING'
      WRITE(6,*) '      XMEAN       RMEAN       SDIST   PITCH ANGLE '
      DO NP=1,NSS
      WRITE(6,131) XMEAN(NP),RMEAN(NP),SDIST(NP),PITCH_ANGL(NP)*RADDEG
      END DO
      WRITE(6,*)
C
C    SMOOTH THE MEAN STREAM SURFACE COORDINATES AND PITCH ANGLE.
C    CHANGED TO USE SMOOTH2  29/09/2017
      CALL SMOOTH2(1,NSS,NSMOOTH,SFAC,XMEAN,RMEAN)
C      CALL SMOOTH(1, NSS, 4, 0.25, SDIST, XMEAN)
C      CALL SMOOTH(1, NSS, 4, 0.25, SDIST, RMEAN)
      CALL SMOOTH(1, NSS, 4, 0.25, SDIST, PITCH_ANGL)
C
      WRITE(6,*)
      WRITE(6,*) ' STREAM SURFACE COORDINATES AND SLOPE AFTER SMOOTHING'
      WRITE(6,*) '      XMEAN       RMEAN       SDIST   PITCH ANGLE '
      DO NP=1,NSS
      WRITE(6,131) XMEAN(NP),RMEAN(NP),SDIST(NP),PITCH_ANGL(NP)*RADDEG
      FORMAT(4F12.4)
      END DO
C
C*************************************************************************************
C*************************************************************************************
C*************************************************************************************
C*************************************************************************************
C  SET THE REFERENCE MERIDIONAL VELOCITY
C
      IF(MIXTYP.EQ.'A') THEN
C
      IF(NSTG.EQ.1.AND.TURBO_TYP.EQ.'C')THEN
      PHI_REF = PHI_STG1*VMRAT(NLE1)/VMRAT(NLE2)*RMEAN(NLE2)/RMEAN(NLE1)
      ELSE
      PHI_REF = PHI_STG1
      END IF
C
      END IF
C*************************************************************************************
C*************************************************************************************
C  SET THE REFERENCE MERIDIONAL VELOCITY.
      IF(NSTG.EQ.1.AND.TURBO_TYP.EQ.'C')
     &           VM_REF = PHI_REF*RMEAN(NLE1)*OMEGA
      IF(NSTG.EQ.1.AND.TURBO_TYP.EQ.'T')
     &           VM_REF = PHI_REF*RMEAN(NLE2)*OMEGA
C*************************************************************************************
C*************************************************************************************
C    SET THE LOCAL MERIDIONAL VELOCITY AND BLADE SPEED
C
      WRITE(6,*)
      WRITE(6,*) ' NS     U_LOCAL     VM_LOCAL    PHI_LOCAL'
      DO  NS = 1,NSS
      VM_LOC(NS)  = VM_REF*VMRAT(NS)
```

```
        U_LOC(NS)    = 0.0
        IF(TURBO_TYP.EQ.'C'.AND.NS.GE.NLE1.AND.NS.LE.NTE1)
       &     U_LOC(NS)= RMEAN(NS)*OMEGA
        IF(TURBO_TYP.EQ.'T'.AND.NS.GE.NLE2.AND.NS.LE.NTE2)
       &     U_LOC(NS) = RMEAN(NS)*OMEGA
        PHI_LOC(NS)    = VM_LOC(NS)/(RMEAN(NS)*OMEGA)
        WRITE(6,132) NS, U_LOC(NS),VM_LOC(NS),PHI_LOC(NS)
        FORMAT(I5,3F12.4)
        END DO
        WRITE(6,*)
C
C    SET THE ABSOLUTE TANGENTIAL VELOCITIES. THIS IS NOT CORRECT FOR MIXTYP = "B".
        IF(MIXTYP.EQ.'B') GO TO 133
        WRITE(6,*) ' SETTING TANGENTIAL VELS, STAGE', NSTG
             VT_LE1 = VM_LOC(NLE1)*TAN(BIN_ROW1(NSTG))  + U_LOC(NLE1)
             VT_TE1 = VM_LOC(NTE1)*TAN(BOUT_ROW1(NSTG)) + U_LOC(NTE1)
             VT_LE2 = VM_LOC(NLE2)*TAN(BIN_ROW2(NSTG))  + U_LOC(NLE2)
             VT_TE2 = VM_LOC(NTE2)*TAN(BOUT_ROW2(NSTG)) + U_LOC(NTE2)
             VT_IN  = VT_LE1*RMEAN(NLE1)/RMEAN(1)
             VT_OUT = VT_TE2*RMEAN(NTE2)/RMEAN(NSS)
        CONTINUE
C   SET THE ABSOLUTE TANGENTIAL VELOCITIES FOR MIXTYP = "B" .
        IF(MIXTYP.EQ.'B') THEN
             VT_LE1 = VM_LOC(NLE1)*TAN(ALPHA_IN(NSTG))
             VT_IN  = VT_LE1*RMEAN(NLE1)/RMEAN(1)
             VT_TE2 = VM_LOC(NTE2)*TAN(ALPHA_OUT(NSTG))
             VT_OUT = VT_TE2*RMEAN(NTE2)/RMEAN(NSS)
        END IF
C
C*******************************************************************************
C*******************************************************************************
        IF(MIXTYP.EQ.'A') THEN
C
        IF(TURBO_TYP.EQ.'C') THEN
        ALPHA_IN(NSTG)  = ATAN(1/PHI_LOC(NLE1) + TAN(BIN_ROW1(NSTG)))
        ALPHA_OUT(NSTG) = BOUT_ROW2(NSTG)
        PSI(NSTG) = (VT_TE1 - VT_LE1)/U_LOC(NLE1)
        ELSE
        ALPHA_IN(NSTG)  = BIN_ROW1(NSTG)
        ALPHA_OUT(NSTG) = ATAN(1./PHI_LOC(NTE2) + TAN(BOUT_ROW2(NSTG)))
        PSI(NSTG)       = (VT_LE2 - VT_TE2)/U_LOC(NLE2)
        END IF
C
C    END OF MIXTYP = "A" LOOP.
        END IF
C
C*******************************************************************************
C*******************************************************************************
C
        IF(FLO_TYP.EQ.'AXI') GO TO 5500
C
C*******************************************************************************
C*******************************************************************************
C    CALCULATE THE BLADE ANGLES FOR FLO_TYP = "MIX"
C
        IF(TURBO_TYP.EQ.'C')THEN
           RDES(NSTG)       = RMEAN(NLE1)
           U(NSTG)          = U_LOC(NLE1)
           VMER(NSTG)       = VM_LOC(NLE1)
           DHO(NSTG)        = PSI(NSTG)*U_LOC(NLE1)*U_LOC(NLE1)
           DRVT             = DHO(NSTG)/OMEGA
           VT_TE1           = (RMEAN(NLE1)*VT_LE1 + DRVT)/RMEAN(NTE1)
           VT_LE2           = VT_TE1*RMEAN(NTE1)/RMEAN(NLE2)
           IF(MIXTYP.EQ.'B') THEN
               BOUT_ROW1(NSTG) = ATAN((VT_TE1 - U_LOC(NTE1))/VM_LOC(NTE1))
               BIN_ROW2(NSTG)  = ATAN(VT_LE2/VM_LOC(NLE2))
               BIN_ROW1(NSTG)  = ATAN((VT_LE1 - U_LOC(NLE1))/VM_LOC(NLE1))
               BOUT_ROW2(NSTG) = ATAN(VT_TE2/VM_LOC(NTE2))
           END IF
        END IF
C
C
        IF(TURBO_TYP.EQ.'T') THEN
```

```
            RDES(NSTG)       = RMEAN(NLE2)
            U(NSTG)          = U_LOC(NLE2)
            VMER(NSTG)       = VM_LOC(NLE2)
            DHO(NSTG)        = -PSI(NSTG)*U_LOC(NLE2)*U_LOC(NLE2)
            DRVT             = DHO(NSTG)/OMEGA
            VT_LE2           = (VT_TE2*RMEAN(NTE2) - DRVT)/RMEAN(NLE2)
            VT_TE1           = VT_LE2*RMEAN(NLE2)/RMEAN(NTE1)
            IF(MIXTYP.EQ.'B') THEN
                BOUT_ROW1(NSTG) = ATAN(VT_TE1/VM_LOC(NTE1))
                BIN_ROW2(NSTG)  = ATAN((VT_LE2 - U_LOC(NLE2))/VM_LOC(NLE2))
                BIN_ROW1(NSTG)  = ATAN(VT_LE1/VM_LOC(NLE1))
                BOUT_ROW2(NSTG) = ATAN((VT_TE2 - U_LOC(NTE2))/VM_LOC(NTE2))
            END IF
         END IF
C
C*************************************************************************************
C*************************************************************************************
C    RE ENTER HERE IF FLO_TYPE = "AXI"
C
         CONTINUE
C
C*************************************************************************************
C*************************************************************************************
C
C  NEXT SECTION FOR BOTH TYPES OF INPUT I.E. BOTH "FLO_TYP" = "MIX" and = "AXI"
C
C   STORE THE STREAM SURFACE COORDINATES AND VELOCITY RATIO.
C
         NSS_STG(NSTG)    = NSS
         NLE1_STG(NSTG)   = NLE1
         NTE1_STG(NSTG)   = NTE1
         NLE2_STG(NSTG)   = NLE2
         NTE2_STG(NSTG)   = NTE2
C
         DO NS =1,NSS
         XMEANALL(NSTG,NS)   = XMEAN(NS)
         RMEANALL(NSTG,NS)   = RMEAN(NS)
         VMLOCALL(NSTG,NS)   = VM_LOC(NS)
         END DO
C
C*********************************************************************
C*********************************************************************
C
C  CALL PROPS FOR STEAM CONDITIONS AT STAGE INLET '
C
         CALL PROPS(1,1,HO(NSTG),S(NSTG),PO(NSTG),T(NSTG),RHO(NSTG),
        &           WET(NSTG), VIN(1),G(NSTG),VS(NSTG),1,IPROPS,IWET)
C
C    WRITE OUTPUT TO SCREEN
C
         WRITE(6,*) ' STAGE INLET STAGNATION ENTHALPY = ', HO(NSTG)
         WRITE(6,*) ' STAGE INLET ENTROPY             = ', S(NSTG)
         WRITE(6,*) ' STAGE INLET PRESSURE            = ', PO(NSTG),' BAR.'
         WRITE(6,*) ' STAGE INLET TEMPERATURE         = ', T(NSTG) , ' K. '
         WRITE(6,*) ' STAGE INLET DENSITY             = ', RHO(NSTG),
        &           ' Kg/M3.'
         WRITE(6,*)
         WRITE(6,*) ' ROTATIONAL SPEED = ', RPM
         WRITE(6,*) ' BLADE SPEED      = ', U(NSTG),' Metres/sec.'
         WRITE(6,*) ' DESIGN RADIUS    = ', RDES(NSTG),' Metres.'
         WRITE(6,*) ' STAGE STAGNATION ENTHALPY CHANGE   = ',
        &             DHO(NSTG)/1000., ' KJ/Kg.'
         WRITE(6,*)
         ROIN = RHO(1)
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
C
C   NOW  INPUT THE DETAILS OF THE STAGE
C
         IF(IFSAME_ALL.EQ.'Y') GO TO 666
```

```
C
      WRITE(6,*)
      WRITE(6,*) ' STARTING INPUT FOR STAGE NUMBER ', NSTG
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      CONTINUE
      WRITE(6,*) ' STAGE NUMBER. ', NSTG
      WRITE(6,*) ' INPUT A GUESS OF THE STAGE EFFICIENCY'
      WRITE(6,*) ' THE CURRENT VALUE IS ', ETA(NSTG)
      WRITE(6,*) ' Press  A  to accept this, or type in a new value.'
            READ(5,*,ERR=13,END=13) ETA(NSTG)
      CONTINUE
      WRITE(6,*) ' THE GUESSED STAGE EFFICIENCY IS ', ETA(NSTG)
      IF(ETA(NSTG).GT.1.0) THEN
      WRITE(6,*) ' VALUE MUST BE A DECIMAL NOT A PERCENTAGE, e.g. 0.8.'
            GO TO 1113
      END IF
            WRITE(10,118) ETA(NSTG)
      FORMAT(F12.3,T25, ' GUESS OF THE STAGE ISENTROPIC EFFICIENCY')
C
      IF(TURBO_TYP.EQ.'T') DHOIS(NSTG)  =   DHO(NSTG)/ETA(NSTG)
      IF(TURBO_TYP.EQ.'C') DHOIS(NSTG)  =   DHO(NSTG)*ETA(NSTG)
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
C
C**********************************************************************
C**********************************************************************
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      WRITE(6,*) ' INPUT A GUESS OF THE DEVIATION ANGLES FOR THE FIRST'
      WRITE(6,* )' AND SECOND BLADE ROWS '
      WRITE(6,*) ' THIS IS THE DIFFERENCE BETWEEN THE FLOW ANGLE AND'
      WRITE(6,*) ' THE METAL ANGLE AT THE TRAILING EDGE'
      WRITE(6,*) ' THE DEVIATION ANGLES INPUT SHOULD ALWAYS BE POSITIVE'
      WRITE(6,*) ' THE CURRENT VALUES ARE', DEVN_1,DEVN_2,'DEGREES'
      WRITE(6,*) ' Press  A  to accept these, or type in new values'
           READ(5,*,ERR=21,END=21)  DEVN_1, DEVN_2
      CONTINUE
      WRITE(6,*) ' THE DEVIATION ANGLES ARE ', DEVN_1, DEVN_2
C
           WRITE(10,121) DEVN_1, DEVN_2
      FORMAT(2F8.3,T25,' ESTIMATE OF THE FIRST AND SECOND ROW DEVIATION
     &ANGLES')
      DEVN1(NSTG) = DEVN_1
      DEVN2(NSTG) = DEVN_2
      WRITE(6,*)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
C
C**********************************************************************
C**********************************************************************
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE REQUIRED INCIDENCE ANGLES FOR THE FIRST'
      WRITE(6,* )' AND SECOND BLADE ROWS '
      WRITE(6,*) ' THIS IS THE DIFFERENCE BETWEEN THE FLOW ANGLE AND'
      WRITE(6,*) ' THE METAL ANGLE AT THE LEADING EDGE'
      WRITE(6,*) ' IT CAN BE EITHER POSITIVE OR NEGATIVE'
      WRITE(6,*) ' THE CURRENT VALUES ARE',AINC_1, AINC_2,'DEGREES'
      WRITE(6,*) ' Press  A  to accept these, or type in new values'
           READ(5,*,ERR=22,END=22)  AINC_1, AINC_2
      CONTINUE
      WRITE(6,*) ' THE INCIDENCE ANGLES ARE ', AINC_1, AINC_2
C
```

```
            WRITE(10,222) AINC_1, AINC_2
        FORMAT(2F8.3,T25,' FIRST AND SECOND ROW INCIDENCE ANGLES')
        AINC1(NSTG) = AINC_1
        AINC2(NSTG) = AINC_2
        WRITE(6,*)
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)
C
C*********************************************************************
C*********************************************************************
C
        IF(IFSAME_ALL.EQ.'Y') GO TO 774
C
C*************************************************************************************
C*************************************************************************************
C    NEXT FOR BOTH VALUES OF  "FLO_TYP",  CHOOSE THE BLADE TWIST. THE TWIST CAN BE ANY
C    MULTIPLE OF THAT REQUIRED BY A FREE-VORTEX DESIGN.
C
        WRITE(6,*)
        WRITE(6,*)' INPUT "FRAC_TWIST", THE FRACTION OF FREE-VORTEX TWIST
       & THAT YOU WANT TO USE ON THIS STAGE.'
        WRITE(6,*)' FRAC_TWIST = 1.0  GIVES FULL FREE-VORTEX TWIST. '
        WRITE(6,*)' FRAC_TWIST = 0.0  GIVES NO TWIST, SO THE BLADE ANGLES
       &ARE THE SAME AT ALL SPANWISE POSITIONS.'
        WRITE(6,*) 'VALUES OF FRAC_TWIST GREATER THAN 1.0 OR LESS THAN 0.0
       & CAN ALSO BE USED.'
        IF(NSTG.EQ.1)  FRAC_TWIST = 1.0
        WRITE(6,*) ' THE CURRENT VALUE IS ', FRAC_TWIST
        WRITE(6,*) ' Press  A  to accept this, or type in a new value.'
        READ(5,*,ERR=140) FRAC_TWIST
        CONTINUE
C
        WRITE(10,141) FRAC_TWIST
        FORMAT(F10.5,T25, ' BLADE TWIST OPTION, FRAC_TWIST')
C
C*************************************************************************************
C*************************************************************************************
C    NEXT INPUT THE OPTION TO ROTATE THE BLADE SECTIONS. EACH SECTION GENERATED
C    CAN BE ROTATED BY A DIFFERNT ANGLE.
C
        IF_ROT = 'N'
        WRITE(6,*)' DO YOU WISH TO ROTATE THE SECTIONS GENERATED BY ANGLES
       & TO BE INPUT LATER ? '
        WRITE(6,*) ' THE ROTATION CAN BE DIFFERENT FOR EACH BLADE SECTION'
        WRITE(6,*) ' INPUT  Y  or N  '
        READ(5,*,ERR=143) IF_ROT
        CONTINUE
        IF(IF_ROT.EQ.'y') IF_ROT = 'Y'
        WRITE(10,144) IF_ROT
        FORMAT(A1,T25,' BLADE ROTATION OPTION , Y or N' )
C
C*********************************************************************
C*********************************************************************
C
        CONTINUE
C
C*************************************************************************************
C*************************************************************************************
C
        WRITE(6,*)
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)
        WRITE(6,*)
        WRITE(6,*) ' INPUT THE QO LINE ANGLES, MEASURED FROM THE AXIAL DIR
       &ECTION AT THE LE AND TE OF THE FIRST BLADE ROW .'
        WRITE(6,*) ' THE CURRENT VALUES ARE',QLE_ROW1(NSTG),QTE_ROW1(NSTG)
        WRITE(6,*) ' Press  A  to accept these, or type in new values.'
        READ(5,*,ERR=775,END=775) QLE_ROW1(NSTG), QTE_ROW1(NSTG)
        CONTINUE
        WRITE(6,*)' QLE_ROW1, QTE_ROW1 =', QLE_ROW1(NSTG), QTE_ROW1(NSTG)
        WRITE(10,776)  QLE_ROW1(NSTG), QTE_ROW1(NSTG)
```

```
      FORMAT(2F8.3,T25,' QO ANGLES AT LE  AND TE OF ROW 1 ')
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
C
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE QO LINE ANGLES, MEASURED FROM THE AXIAL DIR
     &ECTION AT THE LE AND TE OF THE SECOND BLADE ROW .'
      WRITE(6,*) ' THE CURRENT VALUES ARE',QLE_ROW2(NSTG),QTE_ROW2(NSTG)
      WRITE(6,*) ' Press  A  to accept these, or type in new values.'
      READ(5,*,ERR=777,END=777) QLE_ROW2(NSTG), QTE_ROW2(NSTG)
      CONTINUE
      WRITE(6,*)' QLE_ROW2, QTE_ROW2 =', QLE_ROW2(NSTG), QTE_ROW2(NSTG)
      WRITE(10,779)  QLE_ROW2(NSTG), QTE_ROW2(NSTG)
      FORMAT(2F8.3,T25,' QO ANGLES AT LE  AND TE OF ROW 2 ')
      WRITE(6,*)
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)'*******************************************************'
      WRITE(6,*)
C
C*******************************************************************************
C*******************************************************************************
C   RETURN TO HERE IF  IFSAME_ALL = "Y" SO NOT CHANGING THE FLOW ANGLES, ETC.
C
      CONTINUE
C
C*******************************************************************************
C*******************************************************************************
C
            VX(NSTG)    = VM_LOC(NLE1)
            IF(TURBO_TYP.EQ.'C') PHI(NSTG) = VM_LOC(NLE1)/U_LOC(NLE1)
            IF(TURBO_TYP.EQ.'T') PHI(NSTG) = VM_LOC(NLE2)/U_LOC(NLE2)
C
      DHOTOTAL = DHOTOTAL + DHOIS(NSTG)
C
C*******************************************************************************
C*******************************************************************************
C*******************************************************************************
C
      TEXITT = (HO(NSTG) + DHO(NSTG))/CPGAS
      IF(TURBO_TYP.EQ.'T') SEXIT(NSTG) = S(NSTG) -
     &                     (1. - ETA(NSTG))*DHOIS(NSTG)/TEXITT
      IF(TURBO_TYP.EQ.'C') SEXIT(NSTG) = S(NSTG) +
     &                     (1. - ETA(NSTG))*DHO(NSTG)/TEXITT
C
C     CALCULATE PROPERTIES AT STAGE EXIT
C
      DO 155 NS = NTE2,NSS
              VTLOC(NS) = VT_TE2*RMEAN(NTE2)/RMEAN(NS)
              WTLOC     = VTLOC(NS) - U_LOC(NS)
              VMLOC     = VM_LOC(NS)
              VABS(NS)  = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
              VREL(NS)  = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
              HOLOC(NS) = HO(NSTG) + DHO(NSTG)
              SLOC(NS)  = SEXIT(NSTG)
      CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),
     &              RHOLOC(NS),WET(NS),VABS(NS),G(NS),
     &              VS(NS),1,IPROPS,IWET)
              HSTATIC   = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
              HOREL     = HSTATIC   + 0.5*VREL(NS)*VREL(NS)
              POREL(NS) = PLOC(NS)*(HOREL/HSTATIC)**FGA
              POABS(NS) = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
              MACH_REL(NS) = VREL(NS)/VS(NS)
              MACH_ABS(NS) = VABS(NS)/VS(NS)
C
      IF(NS.EQ.NTE2) THEN
              VABSEX(NSTG)  = VABS(NS)
              VRELEX(NSTG)  = VREL(NS)
```

```
                HOEXIT(NSTG)   = HOLOC(NS)
                HEXIT(NSTG)    = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
                RHOEXIT(NSTG)  = RHOLOC(NS)
                PEXIT(NSTG)    = PLOC(NS)
                TEXIT(NSTG)    = TLOC(NS)
                TO_EXIT        = HOLOC(NS)/CPGAS
                PO_EXIT        = PEXIT(NSTG)
       &                       *(HOEXIT(NSTG)/HEXIT(NSTG))**FGA
                VXOUT(NSTG)    = VMLOC
         END IF
C
         CONTINUE
C
C    CALCULATE PROPERTIES AT STAGE INLET
C
         DO 153 NS = 1,NLE1
                VTLOC(NS)   = VT_IN*RMEAN(1)/RMEAN(NS)
                WTLOC       = VTLOC(NS) - U_LOC(NS)
                VMLOC       = VM_LOC(NS)
                VABS(NS)    = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
                VREL(NS)    = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
                HOLOC(NS)   = HO(NSTG)
                SLOC(NS)    = S(NSTG)
         CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
       &           WET(NS),VABS(NS),G(NS),VS(NS),1,IPROPS,IWET)
                HSTATIC    = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
                HOREL      = HSTATIC    + 0.5*VREL(NS)*VREL(NS)
                POREL(NS)  = PLOC(NS)*(HOREL/HSTATIC)**FGA
                POABS(NS)  = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
                MACH_REL(NS) = VREL(NS)/VS(NS)
                MACH_ABS(NS) = VABS(NS)/VS(NS)
C
         IF(NS.EQ.NLE1) THEN
                VABSIN(NSTG)    = VABS(NS)
                VRELIN(NSTG)    = VREL(NS)
                HOINLET(NSTG)   = HO(NSTG)
                HINLET(NSTG)    = HO(NSTG) - 0.5*VABS(NS)*VABS(NS)
                RHOINLET(NSTG)  = RHOLOC(NS)
                PINLET(NSTG)    = PLOC(NS)
                TINLET(NSTG)    = TLOC(NS)
                SINLET(NSTG)    = SLOC(NS)
                VXIN(NSTG)      = VMLOC
                FGA             = G(NS)/(G(NS) - 1.0)
                PO_INLET        = PINLET(NSTG)
       &                        *(HOINLET(NSTG)/HINLET(NSTG))**FGA
         END IF
                IF(NS.EQ.1.AND.NSTG.EQ.1) VM_INLET = VMLOC
         CONTINUE
C
C      CALCULATE PROPERTIES WITHIN THE FIRST BLADE ROW
C
         DO 154 NS = NLE1+1,NTE1
                FRAC   = (SDIST(NS)- SDIST(NLE1))/(SDIST(NTE1)-SDIST(NLE1))
                VTLOC(NS) = VT_LE1 + FRAC*(VT_TE1 - VT_LE1)
                WTLOC      = VTLOC(NS) - U_LOC(NS)
                VMLOC      = VM_LOC(NS)
                VABS(NS)   = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
                VREL(NS)   = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
                HOLOC(NS)  = HO(NSTG) + U_LOC(NS)*(VTLOC(NS) - VT_LE1)
                SLOC(NS)   = SINLET(NSTG) + 0.5*FRAC*(SEXIT(NSTG)- S(NSTG))
         CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
       &           WET(NS),VABS(NS),G(NS),VS(NS),1,IPROPS,IWET)
                HSTATIC    = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
                HOREL      = HSTATIC    + 0.5*VREL(NS)*VREL(NS)
                POREL(NS)  = PLOC(NS)*(HOREL/HSTATIC)**FGA
                POABS(NS)  = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
                MACH_REL(NS) = VREL(NS)/VS(NS)
                MACH_ABS(NS) = VABS(NS)/VS(NS)
C
         IF(NS.EQ.NTE1) THEN
                VABSMID(NSTG)   = VABS(NS)
                VRELMID(NSTG)   = VREL(NS)
                HOMID(NSTG)     = HOLOC(NS)
```

```
                HMID(NSTG)      = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
                RHOMID(NSTG)    = RHOLOC(NS)
                PMID(NSTG)      = PLOC(NS)
                TMID(NSTG)      = TLOC(NS)
                SMID(NSTG)      = SLOC(NS)
                TO_MID          = HOMID(NSTG)/CPGAS
        END IF
        CONTINUE
C
C     CALCULATE PROPERTIES IN THE GAP BETWEEN ROWS
C
        DO 157 NS = NTE1+1,NLE2
                VTLOC(NS) = VT_TE1*RMEAN(NTE1)/RMEAN(NS)
                WTLOC     = VTLOC(NS) - U_LOC(NS)
                VMLOC     = VM_LOC(NS)
                VABS(NS)  = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
                VREL(NS)  = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
                HOLOC(NS) = HOMID(NSTG)
                SLOC(NS)  = SMID(NSTG)
        CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
       &           WET(NS),VABS(NS),G(NS),VS(NS),1,IPROPS,IWET)
                HSTATIC   = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
                HOREL     = HSTATIC   + 0.5*VREL(NS)*VREL(NS)
                POREL(NS) = PLOC(NS)*(HOREL/HSTATIC)**FGA
                POABS(NS) = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
        CONTINUE
C
C     CALCULATE PROPERTIES WITHIN THE SECOND BLADE ROW
C
        DO 156 NS = NLE2,NTE2
                FRAC  = (SDIST(NS)- SDIST(NLE2))/(SDIST(NTE2)-SDIST(NLE2))
                VTLOC(NS)     = VT_LE2 + FRAC*(VT_TE2 - VT_LE2)
                WTLOC     = VTLOC(NS) - U_LOC(NS)
                VMLOC     = VM_LOC(NS)
                VABS(NS)  = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
                VREL(NS)  = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
                HOLOC(NS) = HOMID(NSTG) + U_LOC(NS)*(VTLOC(NS) - VT_LE2)
                SLOC(NS)  = SMID(NSTG)  + 0.5*FRAC*(SEXIT(NSTG) - S(NSTG))
        CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
       &           WET(NS),VABS(NS),G(NS),VS(NS),1,IPROPS,IWET)
                HSTATIC   = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
                HOREL     = HSTATIC   + 0.5*VREL(NS)*VREL(NS)
                POREL(NS) = PLOC(NS)*(HOREL/HSTATIC)**FGA
                POABS(NS) = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
                MACH_REL(NS) = VREL(NS)/VS(NS)
                MACH_ABS(NS) = VABS(NS)/VS(NS)
        CONTINUE
C
C   CALCULATE THE STAGE REACTION.
C
        IF(TURBO_TYP.EQ.'T')REACN(NSTG) =
       &     (TLOC(NLE2) - TLOC(NTE2))/(TLOC(NLE1)-TLOC(NTE2))
        IF(TURBO_TYP.EQ.'C')REACN(NSTG) =
       &     (TLOC(NTE1) - TLOC(NLE1))/(TLOC(NTE2)-TLOC(NLE1))
C
C   OUTPUT TO SCREEN
C
        WRITE(6,*) ' LOCAL FLOW PROPERTIES THROUGH THE STAGE'
        WRITE(6,*) '       VM          P           T           RHO
       &  HO           S'
        DO NS = 1,NSS
        WRITE(6,158) VM_LOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),HOLOC(NS),
       &             SLOC(NS)
        END DO
        FORMAT(6F12.4)
C
C*******************************************************************************
C*******************************************************************************
C
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)
        WRITE(6,*) '  STAGE NUMBER ', NSTG
```

```
      WRITE(6,*)
      WRITE(6,*) '  MASS FLOW RATE                          = ', FLOW
      WRITE(6,*) '  ROTATIONAL SPEED, RPM                   = ', RPM
      WRITE(6,*) '  THE FLOW COEFFICIENT, Vx/U              = ', PHI(NSTG)
      WRITE(6,*) '  THE STAGE LOADING COEFFICIENT DH/U**2 = ', PSI(NSTG)
      WRITE(6,*) '  THE STAGE REACTION BASED ON ENTHALPY  = ',
     &           REACN(NSTG)
      WRITE(6,*) '  THE AXIAL VELOCITY    =                 ', VX(NSTG)
      WRITE(6,*) '  DESIGN POINT ROTATIONAL SPEED  =        ', U(NSTG)
      WRITE(6,*) '  DESIGN POINT  RADIUS           =        ',
     &           RDES(NSTG)
      WRITE(6,*) '  STAGE ISENTROPIC ENTHALPY CHANGE   =    ',
     &           DHOIS(NSTG)/1000.
      WRITE(6,*) '  STAGE ACTUAL ENTHALPY CHANGE     =       ',
     &           DHO(NSTG)/1000.
      WRITE(6,*)
      WRITE(6,*)'*************************************************'
C
C*********************************************************************
C*********************************************************************
C   OUTPUT TO SCREEN
C
      WRITE(6,*) '      XHUB        XTIP        RHUB        RTIP
     & QLEN      PITCH       QANGL       DENSITY'
C
      DO 95 NS = 1,NSS
C
C     SET THE QO LINE ANGLES
C
      IF(NS.LE.NLE1) THEN
          DENS   = RHOLOC(NS)
          FRAC   = (SDIST(NS)- SDIST(1))/(SDIST(NLE1)-SDIST(1))
          QANGL  = FRAC*QLE_ROW1(NSTG)*DEGRAD
     &           + (1.-FRAC)*(0.5*PI+PITCH_ANGL(1))
      END IF
      IF(NS.GE.NLE1.AND.NS.LE.NTE1) THEN
          FRAC   = (SDIST(NS)- SDIST(NLE1))/(SDIST(NTE1)-SDIST(NLE1))
          DENS   = RHOLOC(NS)
          QANGL  = DEGRAD*(FRAC*QTE_ROW1(NSTG)+(1.-FRAC)*QLE_ROW1(NSTG))
      END IF
      IF(NS.GT.NTE1.AND.NS.LE.NLE2) THEN
          FRAC   = (SDIST(NS)- SDIST(NTE1))/(SDIST(NLE2)-SDIST(NTE1))
          DENS   = RHOLOC(NS)
          QANGL  = DEGRAD*(FRAC*QLE_ROW2(NSTG)+(1.-FRAC)*QTE_ROW1(NSTG))
      END IF
      IF(NS.GT.NLE2.AND.NS.LE.NTE2) THEN
          FRAC   = (SDIST(NS)- SDIST(NLE2))/(SDIST(NTE2)-SDIST(NLE2))
          DENS   = RHOLOC(NS)
          QANGL  = DEGRAD*(FRAC*QTE_ROW2(NSTG)+(1.-FRAC)*QLE_ROW2(NSTG))
      END IF
      IF(NS.GT.NTE2) THEN
          FRAC   = (SDIST(NS)- SDIST(NTE2))/(SDIST(NSS)-SDIST(NTE2))
          DENS   = RHOLOC(NS)
          QANGL  = FRAC*(0.5*PI + PITCH_ANGL(NSS))
     &           + (1.-FRAC)*QTE_ROW2(NSTG)*DEGRAD
      END IF
C
C     SET THE HUB AND TIP COORDINATES
C
      DENSMID = DENS
      NLOOP   = 0
C*********************************************************************
C   START THE ITERATION TO ESTIMATE THE MEAN DENSITY.
C
      CONTINUE
C
      NLOOP = NLOOP + 1
C
      VOLFLO = FLOW/DENSMID

      QLEN1  = VOLFLO/(VM_LOC(NS)*SIN(QANGL- PITCH_ANGL(NS)))
     &         /(2*PI*RMEAN(NS))
C    INCREASE QLEN TO ALLOW FOR BLOCKAGE.
```

```
      QLEN   = QLEN1/(1.0 - FBLOCK(NS))
C
      AAXIAL = VOLFLO*SIN(QANGL)
     &       /(PI*VM_LOC(NS)*SIN(QANGL-PITCH_ANGL(NS)))
C   INCREASE THE ANNULUS AREA TO ALLOW FOR BLOCKAGE
      AAXIAL = AAXIAL/(1.0 - FBLOCK(NS))
C
      IF(IFHUB.EQ.'M') THEN
           XHUB(NS) = XMEAN(NS) - 0.5*QLEN*COS(QANGL)
           XTIP(NS) = XMEAN(NS) + 0.5*QLEN*COS(QANGL)
           RHUB(NS) = RMEAN(NS) - 0.5*QLEN*SIN(QANGL)
           RTIP(NS) = RMEAN(NS) + 0.5*QLEN*SIN(QANGL)
           GO TO 161
      END IF
C
      IF(IFHUB.EQ.'H') THEN
           RHUB(NS) = RMEAN(NS)
           XHUB(NS) = XMEAN(NS)
           RTIP(NS) = SQRT(RHUB(NS)*RHUB(NS)  + AAXIAL)
           XTIP(NS) = XHUB(NS) + (RTIP(NS) - RHUB(NS))/TAN(QANGL)
      IF(QANGL.GT.0.99*PI.OR.QANGL.LT.0.01*PI) THEN
           RTIP(NS) = RHUB(NS) + QLEN*SIN(QANGL)
           XTIP(NS) = XHUB(NS) + QLEN*COS(QANGL)
      END IF
      END IF
C
      IF(IFHUB.EQ.'T') THEN
           RTIP(NS) = RMEAN(NS)
           XTIP(NS) = XMEAN(NS)
           RHUB(NS) = SQRT(RTIP(NS)*RTIP(NS)  - AAXIAL)
           XHUB(NS) = XTIP(NS) - (RTIP(NS) - RHUB(NS))/TAN(QANGL)
      IF(QANGL.GT.0.99*PI.OR.QANGL.LT.0.01*PI) THEN
           RHUB(NS) = RTIP(NS) - QLEN*SIN(QANGL)
           XHUB(NS) = XTIP(NS) - QLEN*COS(QANGL)
      END IF
      END IF
C*************************************************************************
C     MAKE AN ESTIMATE OF THE MEAN DENSITY
C     AND USE IT TO CALCULATE THE VOLUME FLOW. ITERATE 5 TIMES .
C
      IF(IFHUB.EQ.'H') THEN
      REQL    = SQRT((RTIP(NS)*RTIP(NS) + RHUB(NS)*RHUB(NS))/2.0)
      RAVG    = 0.5*(RHUB(NS) + REQL)
      VTAVG   = 0.5*(VTLOC(NS) + VTLOC(NS)*RHUB(NS)/REQL)
      DENSAVG = 0.5*(DENS + DENSMID)
      DPDR    = DENSAVG*VTAVG*VTAVG/RAVG
      DENSMID = DENS + DPDR*(REQL-RHUB(NS))/(GAMM*RGAS*TLOC(NS))
      END IF
C
      IF(IFHUB.EQ.'T') THEN
      REQL    = SQRT((RTIP(NS)*RTIP(NS) + RHUB(NS)*RHUB(NS))/2.0)
      RAVG    = 0.5*(RTIP(NS) + REQL)
      VTAVG   = 0.5*(VTLOC(NS) + VTLOC(NS)*RTIP(NS)/REQL)
      DENSAVG = 0.5*(DENS + DENSMID)
      DPDR    = DENSAVG*VTAVG*VTAVG/RAVG
      DENSMID = DENS + DPDR*(REQL-RTIP(NS))/(GAMM*RGAS*TLOC(NS))
      END IF
C
C      WRITE(6,*) ' ORIG DENS=',DENS,'MEAN DENS=',DENSMID,
C     & 'RHUB,REQL,RTIP', RHUB(NS),REQL,RTIP(NS)
C
      IF(NLOOP.LT.5) GO TO 160
C   END OF THE ITERATION TO ESTIMATE THE MEAN DENSITY
C
C*************************************************************************
C   JUMP TO HERE IF IFHUB = 'M'
  160 CONTINUE
C
      WRITE(6,201) XHUB(NS),XTIP(NS),RHUB(NS),RTIP(NS),QLEN,
     &           PITCH_ANGL(NS)*RADDEG,QANGL*RADDEG,DENS
  201 FORMAT(8F12.4)
C
C
```

```
      CONTINUE
C
C*********************************************************************
C   STORE THE HUB AND CASING COORDINATES FOR THE STAGE.
      DO 98 NS = 1,NSS
      XDIF = XTIP(NS) - XHUB(NS)
      RDIF = RTIP(NS) - RHUB(NS)
      SPAN(NS) = SQRT(XDIF*XDIF + RDIF*RDIF)
      XHUBALL(NSTG,NS) = XHUB(NS)
      XTIPALL(NSTG,NS) = XTIP(NS)
      RHUBALL(NSTG,NS) = RHUB(NS)
      RTIPALL(NSTG,NS) = RTIP(NS)
      CONTINUE
C
C*********************************************************************
C    CALCULATE THE ASPECT RATIOS .
      RTIPIN(NSTG)   = RTIP(NLE1)
      RTIPEXIT(NSTG) = RTIP(NTE2)
      AXCHRD1(NSTG)  = SDIST(NTE1) - SDIST(NLE1)
      AXCHRD2(NSTG)  = SDIST(NTE2) - SDIST(NLE2)
      SPANIN         = 0.5*(SPAN(NLE1) + SPAN(NTE1))
      SPANEXIT       = 0.5*(SPAN(NLE2) + SPAN(NTE2))
      ASPN(NSTG)     = SPANIN/AXCHRD1(NSTG)
      ASPR(NSTG)     = SPANEXIT/AXCHRD2(NSTG)
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
C   SET THE BLADE NUMBERS USING THE MODIFIED ZWEIFEL COEFFICIENT FOR BOTH
C   TURBINES AND COMPRESSORS.
C
C      IF(TURBO_TYP.EQ.'T') THEN
      COSIN1  = COS(BIN_ROW1(NSTG))
      COSIN2  = COS(BIN_ROW2(NSTG))
      COSOUT1 = COS(BOUT_ROW1(NSTG))
      COSOUT2 = COS(BOUT_ROW2(NSTG))
      COS1    = AMIN1(COSIN1,COSOUT1)
      COS2    = AMIN1(COSIN2,COSOUT2)
      RAVG1    = 0.5*(RMEAN(NLE1) + RMEAN(NTE1))
      RAVG2    = 0.5*(RMEAN(NLE2) + RMEAN(NTE2))
      ROVMAVG1 = 0.5*(RHOLOC(NLE1)*VM_LOC(NLE1)
     &        + RHOLOC(NTE1)*VM_LOC(NTE1))
      ROVMAVG2 = 0.5*(RHOLOC(NLE2)*VM_LOC(NLE2)
     &        + RHOLOC(NTE2)*VM_LOC(NTE2))

C
      IF(TURBO_TYP.EQ.'T') THEN
      PCX1Z   =  ZWEIFEL*( POABS(NLE1) - PLOC(NTE1) )*RAVG1*1.0E05
     &       /( ROVMAVG1*ABS(RMEAN(NTE1)*VT_LE1 - RMEAN(NTE1)*VT_TE1) )
C
      PCX2Z   =  ZWEIFEL*( POREL(NLE2) - PLOC(NTE2) )*RAVG2*1.0E05
     &       /( ROVMAVG2*ABS(RMEAN(NTE2)*VT_LE2 - RMEAN(NTE2)*VT_TE2) )
      END IF
C
      IF(TURBO_TYP.EQ.'C') THEN
      WRITE(6,*) 'DPOREL' , (POREL(NLE1) - PLOC(NLE1))
      WRITE(6,*)' DRVTHETA', VT_LE1,VT_TE1
      PCX1Z   =  ZWEIFEL*( POREL(NLE1) - PLOC(NLE1) )*RAVG1*1.0E05
     &       /( ROVMAVG1*ABS(RMEAN(NTE1)*VT_LE1 - RMEAN(NTE1)*VT_TE1) )
C
      PCX2Z   =  ZWEIFEL*( POABS(NLE2) - PLOC(NLE2) )*RAVG2*1.0E05
     &       /( ROVMAVG2*ABS(RMEAN(NTE2)*VT_LE2 - RMEAN(NTE2)*VT_TE2) )
      END IF
C


C      PCX1Z   = 0.5*ZWEIFEL/ABS(RMEAN(NLE1)*TAN(BIN_ROW1(NSTG))
C     &             - RMEAN(NTE1)*TAN(BOUT_ROW1(NSTG)))*RAVG1
C     &              /COS1**2
C
C      PCX2Z   = 0.5*ZWEIFEL/ABS(RMEAN(NLE2)*TAN(BIN_ROW2(NSTG))
C     &            - RMEAN(NTE2)*TAN(BOUT_ROW2(NSTG)))*RAVG2
C     &             /COS2**2
C
```

```
      WRITE(6,*)
      WRITE(6,*)     '  PITCH TO AXIAL CHORD RATIOS BASED ON THE ZWEIFEL
     & COEFFICIENT '
      WRITE(6,*)     '  MODIFIED TO ALLOW FOR RADIUS AND STREAM SURFACE
     & THICKNESS CHANGES.'
      WRITE(6,*)     '  PCX1Z, PCX2Z ', PCX1Z,PCX2Z
      WRITE(6,*)
C
C   SKIP THIS , USE THE ZWEIFEL COEFFICIENT AS ABOVE FOR COMPRESSORS.
C
      GO TO 1234
C
C   SET THE BLADE NUMBERS USING THE DIFFUSION FACTOR FOR COMPRESSORS.

      IF(TURBO_TYP.EQ.'C') THEN
      SOL1    =  0.5*ABS (TAN(BIN_ROW1(NSTG)) - TAN(BOUT_ROW1(NSTG)) )
     *             *COS(BIN_ROW1(NSTG))*COS(BOUT_ROW1(NSTG))
     &            /(D_FAC*COS(BOUT_ROW1(NSTG)) + COS(BIN_ROW1(NSTG))
     &            - COS(BOUT_ROW1(NSTG)))
      CX_CT    = COS( (BIN_ROW1(NSTG) + BOUT_ROW1(NSTG))/2 )
      PCX1D    = 1./(SOL1*CX_CT)
C
      SOL2    =  0.5*ABS (TAN(BIN_ROW2(NSTG)) - TAN(BOUT_ROW2(NSTG)) )
     *             *COS(BIN_ROW2(NSTG))*COS(BOUT_ROW2(NSTG))
     &            /(D_FAC*COS(BOUT_ROW2(NSTG)) + COS(BIN_ROW2(NSTG))
     &            - COS(BOUT_ROW2(NSTG)))
      CX_CT    = COS( (BIN_ROW2(NSTG) + BOUT_ROW2(NSTG))/2 )
      PCX2D    = 1./(SOL2*CX_CT)
      END IF
C
      WRITE(6,*)
      WRITE(6,*) ' PITCH TO AXIAL CHORD RATIOS FROM DIFFUSION FACTOR.'
      WRITE(6,*) ' PCX1D, PCX2D ', PCX1D, PCX2D
      WRITE(6,*)
C
      CONTINUE
C
      PCX1   = PCX1Z
      PCX2   = PCX2Z
C
      IF(PCX1.GT.2.0) PCX1 = 2.0
      IF(PCX2.GT.2.0) PCX2 = 2.0
      PITCH1     = PCX1*AXCHRD1(NSTG)
      PITCH2     = PCX2*AXCHRD2(NSTG)
      RSET_PC1   = 0.5*(RMEAN(NLE1) + RMEAN(NTE1))
      RSET_PC2   = 0.5*(RMEAN(NLE2) + RMEAN(NTE2))
      NROW1      = 2*PI*RSET_PC1/PITCH1
      NROW2      = 2*PI*RSET_PC2/PITCH2
C
C    OUTPUT TO SCREEN.
      WRITE(6,*)
      WRITE(6,*) ' PITCH TO AXIAL CHORD RATIOS USED. '
      WRITE(6,*) ' PCX1, PCX2, PITCH1, PITCH2, NROW1, NROW2 ',
     &             PCX1,PCX2,PITCH1,PITCH2,NROW1,NROW2
      WRITE(6,*)
C
      NR1 = 2*NSTG -1
      NR2 = 2*NSTG
      NBLADE(NR1) = NROW1
      NBLADE(NR2) = NROW2
      WRITE(6,*) 'STAGE No, ROW No, No. BLADES',NSTG, NR1, NBLADE(NR1)
      WRITE(6,*) 'STAGE No, ROW No, No. BLADES',NSTG, NR2, NBLADE(NR2)
      WRITE(6,*)
C
C*********************************************************************
C*********************************************************************
C    OUTPUT TO SCREEN
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*) ' CONDITIONS FOR THE FIRST BLADE ROW OF THE STAGE. '
      IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE STATOR '
      IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR ROTOR'
      WRITE(6,*)'*****************************************************'
```

```
      WRITE(6,*) 'FIRST BLADE INLET AND EXIT ANGLES ',
     &            BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG
      WRITE(6,*) ' FIRST BLADE AXIAL VELOCITY     ', VXIN(NSTG)
      WRITE(6,*) ' FIRST BLADE INLET MACH NUMBER  ', MACH_REL(NLE1)
      WRITE(6,*) ' FIRST BLADE EXIT MACH NUMBER   ', MACH_REL(NTE1)
      WRITE(6,*) ' FIRST BLADE EXIT DENSITY       ', RHOMID(NSTG)
      WRITE(6,*) ' FIRST BLADE EXIT PRESSURE      ', PMID(NSTG)
      WRITE(6,*) ' FIRST BLADE INLET STAGN PRESS  ', POABS(NLE1)
      WRITE(6,*) ' FIRST BLADE EXIT STAGN PRESS   ', POABS(NTE1)
      WRITE(6,*) ' FIRST BLADE REL INLET STAG PRES', POREL(NLE1)
      WRITE(6,*) ' FIRST BLADE EXIT TEMPERATURE   ', TMID(NSTG)
      WRITE(6,*) ' FIRST BLADE EXIT STAGN TEMP    ', TO_MID
      WRITE(6,*) ' FIRST BLADE TIP RADIUS =       ', RTIPIN(NSTG)
      WRITE(6,*) ' FIRST BLADE INLET SPAN =       ', SPANIN
      WRITE(6,*) ' FIRST BLADE AXIAL CHORD=       ', AXCHRD1(NSTG)
      WRITE(6,*) ' FIRST BLADE ASPECT RATIO =     ', ASPN(NSTG)
      WRITE(6,*)
C
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)  ' CONDITIONS FOR THE SECOND BLADE ROW OF THE STAGE.'
      IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE ROTOR '
      IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR STATOR'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*) ' SECOND BLADE INLET AND EXIT ANGLES ',
     &            BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
      WRITE(6,*) ' SECOND BLADE AXIAL VELOCITY    ', VXOUT(NSTG)
      WRITE(6,*) ' SECOND BLADE INLET MACH NUMBER ', MACH_REL(NLE2)
      WRITE(6,*) ' SECOND BLADE EXIT MACH NUMBER  ', MACH_REL(NTE2)
      WRITE(6,*) ' SECOND BLADE DENSITY           ', RHOEXIT(NSTG)
      WRITE(6,*) ' SECOND BLADE EXIT PRESSURE     ', PEXIT(NSTG)
      WRITE(6,*) ' SECOND BLADE INLET STAGN PRES. ', POABS(NLE2)
      WRITE(6,*) ' SECOND BLADE REL INLET STAG PR.', POREL(NLE2)
      WRITE(6,*) ' SECOND BLADE ROW EXIT STAG PRES', POABS(NTE2)
      WRITE(6,*) ' SECOND BLADE TEMPERATURE       ', TEXIT(NSTG)
      WRITE(6,*) ' SECOND BLADE STAGN EXIT TEMP.  ', TO_EXIT
      WRITE(6,*) ' SECOND BLADE TIP RADIUS =      ', RTIPEXIT(NSTG)
      WRITE(6,*) ' SECOND BLADE EXIT SPAN =       ', SPANEXIT
      WRITE(6,*) ' SECOND BLADE AXIAL CHORD =     ', AXCHRD2(NSTG)
      WRITE(6,*) ' SECOND BLADE ASPECT RATIO=     ', ASPR(NSTG)
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
C
C
      WRITE(6,*)
      WRITE(6,*)  ' THE RADII THROUGH THE STAGE ARE -  '
      WRITE(6,210) RHUB(NLE1),RTIP(NLE1),RHUB(NLE2),RTIP(NLE2)
      FORMAT('ROW 1 HUB RADIUS ',F8.4,'  ROW 1 TIP RADIUS ',F8.4,
     & '  ROW 2 HUB RADIUS ',F8.4,'  ROW 2 TIP RADIUS ',F8.4)
      WRITE(6,*)
C
C*****************************************************************
C*****************************************************************
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
      WRITE(6,125)
      FORMAT(' DO YOU WANT TO CHANGE THE ANGLES FOR THIS STAGE ? ANSWER
     & "Y" OR "N" ')
           READ(5,*)     ANSANGL
           WRITE(6,*)    ANSANGL
           WRITE(10,122) ANSANGL
      FORMAT(A1,T25, ' DO YOU WANT TO CHANGE THE ANGLES FOR THIS STAGE ?
     & "Y" or "N"')
      WRITE(6,*) ' CHANGE ANGLES ANSWER WAS ', ANSANGL
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)
C
      IF(ANSANGL.EQ.'Y'.OR.ANSANGL.EQ.'y') GO TO 500
C
      NROW2        = 2*NSTG
```

```
      PEXIT(NROW2) = PEXIT(NSTG)
C*************************************************************************
C*************************************************************************
C     GO TO 1000 IF THIS IS THE LAST STAGE
C
      IF (NSTG.EQ.NSTAGES) GO TO 1000
C*************************************************************************
C*************************************************************************
C   IF NOT THE LAST STAGE
C   SET PARAMETERS FOR THE NEXT STAGE SAME AS FOR PRESENT STAGE.
C
      ETA(NSTG+1)    = ETA(NSTG)
      RHUB(NSTG+1)   = RHUB(NSTG)
      RDES(NSTG+1)   = RDES(NSTG)
      DHOIS(NSTG+1) = DHOIS(NSTG)
      DHO(NSTG+1)    = DHO(NSTG)
      U(NSTG+1)      = U(NSTG)
      HO(NSTG+1)     = HO(NSTG) + DHO(NSTG)
      S(NSTG+1)      = SEXIT(NSTG)
C
      ALPHA_IN(NSTG+1)  = ALPHA_IN(NSTG)
      ALPHA_OUT(NSTG+1) = ALPHA_OUT(NSTG)
      BIN_ROW1(NSTG+1)  = BIN_ROW1(NSTG)
      BIN_ROW2(NSTG+1)  = BIN_ROW2(NSTG)
      BOUT_ROW1(NSTG+1) = BOUT_ROW1(NSTG)
      BOUT_ROW2(NSTG+1) = BOUT_ROW2(NSTG)
      DEVN1(NSTG+1)      = DEVN1(NSTG)
      DEVN2(NSTG+1)      = DEVN2(NSTG)
      AINC1(NSTG+1)      = AINC1(NSTG)
      AINC2(NSTG+1)      = AINC2(NSTG)
C
      REACN(NSTG+1) = REACN(NSTG)
      ASPN(NSTG+1)  = ASPN(NSTG)
      ASPR(NSTG+1)  = ASPR(NSTG)
      AXCHRD1(NSTG+1) = AXCHRD1(NSTG)
      AXCHRD2(NSTG+1) = AXCHRD2(NSTG)
      PHI(NSTG+1)    = PHI(NSTG)
      PSI(NSTG+1)    = PSI(NSTG)
      QLE_ROW1(NSTG+1) = QLE_ROW1(NSTG)
      QTE_ROW1(NSTG+1) = QTE_ROW1(NSTG)
      QLE_ROW2(NSTG+1) = QLE_ROW2(NSTG)
      QTE_ROW2(NSTG+1) = QTE_ROW2(NSTG)
      ROWGAP(NSTG+1)   = ROWGAP(NSTG)
      STAGEGAP(NSTG+1) = STAGEGAP(NSTG)
      FBLOCK_LE(NSTG+1)= FBLOCK_LE(NSTG)
      FBLOCK_TE(NSTG+1)= FBLOCK_TE(NSTG)
C
C*************************************************************************
C*************************************************************************
C   RETURN TO 1100 TO START ON NEXT STAGE
C
      GO TO 1100
C
C*************************************************************************
C*************************************************************************
C
      CONTINUE
C
C*************************************************************************
C*************************************************************************
C
C      FORM CONTINUOUS STREAM SURFACES ON THE HUB AND TIP
C
C*************************************************************************
C*************************************************************************
C
      NALL = 0
      DO 333 NSTG = 1,NSTAGES
C
      IF(NSTG.EQ.1) THEN
C
      NEND = NTE2_STG(1)
      IF(NSTAGES.EQ.1) NEND = NSS_STG(1)
```

```
      DO 334 NS = 1,NEND
           NALL = NALL + 1
           XSURFHUB(NALL) = XHUBALL(NSTG,NS)
           RSURFHUB(NALL) = RHUBALL(NSTG,NS)
           XSURFTIP(NALL) = XTIPALL(NSTG,NS)
           RSURFTIP(NALL) = RTIPALL(NSTG,NS)
           IF(NS.EQ.NLE1_STG(NSTG)) NLE1_ALL(NSTG) = NALL
           IF(NS.EQ.NTE1_STG(NSTG)) NTE1_ALL(NSTG) = NALL
           IF(NS.EQ.NLE2_STG(NSTG)) NLE2_ALL(NSTG) = NALL
           IF(NS.EQ.NTE2_STG(NSTG)) NTE2_ALL(NSTG) = NALL
      CONTINUE
C
           WRITE(6,*) ' STAGE No ', NSTG, 'NALL =', NALL
           WRITE(6,*) ' NLE1,NTE1,NLE2,NTE2', NLE1_ALL(NSTG),
     &                NTE1_ALL(NSTG),NLE2_ALL(NSTG),NTE2_ALL(NSTG)
C
C  END OF NSTAGE = 1 LOOP
      END IF
C
      IF(NSTG.EQ.1) GO TO 333
C*******************************************************************************
C*******************************************************************************
C
      DXLAST = XSURFHUB(NALL) - XSURFHUB(NALL-1)
      DRLAST = RSURFHUB(NALL) - RSURFHUB(NALL-1)
      DSLAST = SQRT(DXLAST*DXLAST + DRLAST*DRLAST)
C
C  REMOVE ANY OVERLAPPING POINTS ON THE UPSTREAM STAGE STREAM SURFACE.
C  CHECK FOR AND REMOVE ANY OVERLAP, THIS IS DONE ON THE HUB ONLY, SO PROBLEMS
C  CAN STILL ARISE AT THE TIP.

      NS = 1
      NE = NTE2_STG(NSTG)
      IF(NSTG.EQ.NSTAGES) NE = NSS_STG(NSTG)
C
      DO 335 N = NS,NE
C
      DXNEXT  = XHUBALL(NSTG,N) - XSURFHUB(NALL)
      DRNEXT  = RHUBALL(NSTG,N) - RSURFHUB(NALL)
      PROJN   = (DXNEXT*DXLAST  + DRNEXT*DRLAST)/DSLAST
C
C   SKIP ANY OVERLAPPING POINTS
      DISTMIN = 0.01*(AXCHRD1(NSTG) + AXCHRD2(NSTG))

      IF(PROJN.GT.DISTMIN.OR.N.EQ.NLE1_STG(NSTG)
     &                   .OR.N.EQ.NLE2_STG(NSTG)) THEN
      NALL = NALL + 1
           XSURFHUB(NALL) = XHUBALL(NSTG,N)
           RSURFHUB(NALL) = RHUBALL(NSTG,N)
           XSURFTIP(NALL) = XTIPALL(NSTG,N)
           RSURFTIP(NALL) = RTIPALL(NSTG,N)
           IF(N.EQ.NLE1_STG(NSTG)) NLE1_ALL(NSTG) = NALL
           IF(N.EQ.NTE1_STG(NSTG)) NTE1_ALL(NSTG) = NALL
           IF(N.EQ.NLE2_STG(NSTG)) NLE2_ALL(NSTG) = NALL
           IF(N.EQ.NTE2_STG(NSTG)) NTE2_ALL(NSTG) = NALL
           DXLAST = XSURFHUB(NALL) - XSURFHUB(NALL-1)
           DRLAST = RSURFHUB(NALL) - RSURFHUB(NALL-1)
           DSLAST = SQRT(DXLAST*DXLAST + DRLAST*DRLAST)
      END IF
C
      CONTINUE
C
      CONTINUE
C   NINTPTS  IS THE NUMBER OF POINTS ON THE CONTINUOUS STREAM SURFACE.
      NINTPTS = NALL
C
C*******************************************************************************
C*******************************************************************************
c   WRITE THE STREAM SURFACE COORDINATES TO THE SCREEN
C
C
      WRITE(6,*)
      WRITE(6,*) ' COORDINATES OF THE CONTINUOUS STREAM SURFACE BEFORE
```

```
     & SMOOTHING, NINTPTS = ',NINTPTS
      WRITE(6,*) ' XSURFHUB'
      WRITE(6,304) (XSURFHUB(N),N=1,NINTPTS)
      WRITE(6,*) ' XSURFTIP'
      WRITE(6,304) (XSURFTIP(N),N=1,NINTPTS)
      WRITE(6,*)  ' RSURFHUB'
      WRITE(6,304) (RSURFHUB(N),N=1,NINTPTS)
      WRITE(6,*)  ' RSURFTIP'
      WRITE(6,304) (RSURFTIP(N),N=1,NINTPTS)
      FORMAT(8F10.5)
      WRITE(6,*)
C
C*******************************************************************************
C*******************************************************************************
C   SET THE SURFACE DISTANCE ON THE STREAM SURFACE FOR USE IN THE SMOOTHING.
C   AND CHECK FOR ANY OVERLAPPING POINTS ON THE HUB AND TIP STREAM SURFACES.
C   ATTEMPT TO CORRECT FOR OVERLAPPING POINTS BY INTERCHANGING POINTS.
C
      DO NSTG = 1,NSTAGES
      WRITE(6,*)
      WRITE(6,*) '  LEADING AND TRAILING EDGE POINTS ON THE CONTINUOUS
     &STREAM SURFACE, STAGE No ',NSTG
      WRITE(6,*)  NLE1_ALL(NSTG),NTE1_ALL(NSTG),
     &           NLE2_ALL(NSTG),NTE2_ALL(NSTG)
      END DO
C
      SDISTHUB(1) = 0.0
      SDISTTIP(1) = 0.0
      XDIF_HUB = 1.0
      RDIF_HUB = 1.0
      XDIF_TIP = 1.0
      RDIF_TIP = 1.0
      IFWARNH  = 0
      IFWARNT  = 0
      DO 336 N = 2,NINTPTS
C
      CONTINUE
      XDIF = XSURFHUB(N) - XSURFHUB(N-1)
      RDIF = RSURFHUB(N) - RSURFHUB(N-1)
      SDISTHUB(N) = SDISTHUB(N-1) + SQRT(XDIF*XDIF + RDIF*RDIF)
      PROJ_HUB    = XDIF*XDIF_HUB + RDIF*RDIF_HUB

      IF(PROJ_HUB.LT.0.0) THEN
           WRITE(6,*)
           WRITE(6,*) ' CONTINUOUS STREAM SURFACE POINT NUMBER ', N
           WRITE(6,*) ' INTERCHANGING POINTS ON THE HUB'
           TEMP          = XSURFHUB(N-1)
           XSURFHUB(N-1) = XSURFHUB(N)
           XSURFHUB(N)   = TEMP
           TEMP          = RSURFHUB(N-1)
           RSURFHUB(N-1) = RSURFHUB(N)
           RSURFHUB(N)   = TEMP
           IFWARNH       = 1
           GO TO 401
      END IF
      XDIF_HUB = XDIF
      RDIF_HUB = RDIF
C
      CONTINUE
      XDIF = XSURFTIP(N) - XSURFTIP(N-1)
      RDIF = RSURFTIP(N) - RSURFTIP(N-1)
      SDISTTIP(N) = SDISTTIP(N-1) + SQRT(XDIF*XDIF + RDIF*RDIF)
      PROJ_TIP = XDIF*XDIF_TIP + RDIF*RDIF_TIP

      IF(PROJ_TIP.LT.0.0) THEN
           WRITE(6,*)
           WRITE(6,*) ' CONTINUOUS STREAM SURFACE POINT NUMBER ', N
           WRITE(6,*) ' INTERCHANGING POINTS ON THE TIP'
           TEMP          = XSURFTIP(N-1)
           XSURFTIP(N-1) = XSURFTIP(N)
           XSURFTIP(N)   = TEMP
           TEMP          = RSURFTIP(N-1)
           RSURFTIP(N-1) = RSURFTIP(N)
```

```
2241              RSURFTIP(N)    = TEMP
2242              IFWARNT        = 1
2243              GO TO 402
2244            END IF
2245            XDIF_TIP = XDIF
2246            RDIF_TIP = RDIF
2247
2248            CONTINUE
2249      C
2250      C   SMOOTH THE HUB AND CASING STREAM SURFACE COORDINATES USING SMOOTH2.
2251      C
2252      C      CALL SMOOTH(1,NINTPTS,NSMOOTH,SFAC,SDISTHUB,XSURFHUB)
2253      C      CALL SMOOTH(1,NINTPTS,NSMOOTH,SFAC,SDISTHUB,RSURFHUB)
2254            CALL SMOOTH2(1,NINTPTS,NSMOOTH,SFAC,XSURFHUB,RSURFHUB)
2255      C      CALL SMOOTH(1,NINTPTS,NSMOOTH,SFAC,SDISTTIP,XSURFTIP)
2256      C      CALL SMOOTH(1,NINTPTS,NSMOOTH,SFAC,SDISTTIP,RSURFTIP)
2257            CALL SMOOTH2(1,NINTPTS,NSMOOTH,SFAC,XSURFTIP,RSURFTIP)
2258      C
2259            WRITE(6,*)
2260            WRITE(6,*) ' COORDINATES OF THE CONTINUOUS STREAM SURFACE AFTER
2261           & SMOOTHING, NINTPTS = ',NINTPTS
2262            WRITE(6,*) ' XSURFHUB'
2263            WRITE(6,304) (XSURFHUB(N),N=1,NINTPTS)
2264            WRITE(6,*) ' XSURFTIP'
2265            WRITE(6,304) (XSURFTIP(N),N=1,NINTPTS)
2266            WRITE(6,*)  ' RSURFHUB'
2267            WRITE(6,304) (RSURFHUB(N),N=1,NINTPTS)
2268            WRITE(6,*)  ' RSURFTIP'
2269            WRITE(6,304) (RSURFTIP(N),N=1,NINTPTS)
2270            WRITE(6,*)
2271      C
2272      C*************************************************************************
2273      C*************************************************************************
2274      C
2275            WRITE(6,*)
2276           &'DO YOU WANT TO OUTPUT ALL BLADE ROWS TO THE FILE "stagen.dat"? '
2277            WRITE(6,*) '  ANSWER  "Y"   or  "N" . '
2278            READ(5,*)          ANSOUT
2279            IF(ANSOUT.EQ.'y') ANSOUT='Y'
2280            WRITE(6,*) ' ANSOUT = ', ANSOUT
2281            WRITE(10,337)      ANSOUT
2282            FORMAT(A1,T25,' IS OUTPUT REQUESTED FOR ALL BLADE ROWS ? ')
2283      C
2284            NROW_OUT = 0
2285            DO 339 NOUT = 1,NROWS
2286      C
2287            IF(ANSOUT.EQ.'Y'.OR.ANSOUT.EQ.'y') THEN
2288                IFOUT(NOUT) = 'Y'
2289                NROW_OUT    = NROWS
2290            ELSE
2291                 IF(NOUT.EQ.1) THEN
2292                 WRITE(6,*)' FOR EACH ONE OF',NROWS,' BLADE ROWS'
2293                 WRITE(6,*)' TYPE  "Y"  or  "N"  TO CHOOSE WHETHER TO OUTPUT'
2294                 WRITE(6,*)' THE  BLADE ROW DATA OR NOT.'
2295                 END IF
2296      C
2297                 WRITE(6,*)
2298                 WRITE(6,*) ' INPUT "Y" or "N"  FOR ROW NUMBER ',NOUT
2299      C
2300                 READ(5,*)      IFOUT(NOUT)
2301                 IF(IFOUT(NOUT).EQ.'y') IFOUT(NOUT) = 'Y'
2302                 WRITE(10,338) IFOUT(NOUT)
2303                 FORMAT(A1,T25, ' IS OUTPUT REQUESTED FOR THIS BLADE ROW ?')
2304                 IF(IFOUT(NOUT).EQ.'Y')  NROW_OUT = NROW_OUT + 1
2305            END IF
2306      C
2307            CONTINUE
2308
2309
2310      C*************************************************************************
2311      C*************************************************************************
2312      C*************************************************************************
2313      C  NOW  PREPARE AND WRITE OUT DATA FOR STAGEN
```

```
C*******************************************************************************
C*******************************************************************************
C*******************************************************************************
C
C      WRITE OUTPUT FOR  STAGEN
C
      OPEN(UNIT=9, FILE= 'stagen.dat')
C
      WRITE(9,9000) RGAS, GAMM
      FORMAT(2F12.4,T25, ' GAS CONSTANT, GAMMA')
C
      WRITE(9,9002) IM, KM
      FORMAT(2I10, T25, ' IM, KM ' )
C
      WRITE(9,9001)  FPRAT,FPMAX
      FORMAT(2F12.4,T25, '  FPRAT,  FPMAX')
C
      WRITE(9,9003) FRRAT,FRMAX
      FORMAT(2F12.4,T25, '  FRRAT,  FRMAX')
C
      WRITE(9,9005) 0
      FORMAT(I10,T25, ' IFDEFAULTS ')
C
      WRITE(9,9004) NROW_OUT, NOSECT
      FORMAT(2I10,T25,  ' NOWS, N SECTIONS ')
C
      WRITE(9,9006) 1.0
      FORMAT(F10.3,T25, ' SCALING FACTOR ')
C
C*******************************************************************************
C*******************************************************************************
      DO 3500  NR = 1, NROWS
C
C     SKIP THE OUTPUT FOR THIS ROW IF "IFOUT" IS NOT = "Y".
      IF(IFOUT(NR).NE.'Y') GO TO 3500
C
      NROW   = NR
      NSTG   = (NR-1)/2 + 1
      IF(MOD(NROW,2).EQ.0) THEN
      NLEALL = NLE2_ALL(NSTG)
      NTEALL = NTE2_ALL(NSTG)
      ELSE
      NLEALL = NLE1_ALL(NSTG)
      NTEALL = NTE1_ALL(NSTG)
      END IF
C
C*******************************************************************************
C*******************************************************************************
C    RESTORE 1D VARIABLES FOR THIS STAGE
C
      NSS   = NSS_STG(NSTG)
      DO NS = 1,NSS
           XMEAN(NS) = XMEANALL(NSTG,NS)
           RMEAN(NS) = RMEANALL(NSTG,NS)
           VM_LOC(NS)= VMLOCALL(NSTG,NS)
           RHUB(NS)  = RHUBALL(NSTG,NS)
           RTIP(NS)  = RTIPALL(NSTG,NS)
           XHUB(NS)  = XHUBALL(NSTG,NS)
           XTIP(NS)  = XTIPALL(NSTG,NS)
      END DO
C
C
      NLE1 = NLE1_STG(NSTG)
      NTE1 = NTE1_STG(NSTG)
      NLE2 = NLE2_STG(NSTG)
      NTE2 = NTE2_STG(NSTG)
C
      IF(MOD(NROW,2).EQ.0)THEN
           NLE = NLE2
           NTE = NTE2
      ELSE
           NLE = NLE1
           NTE = NTE1
```

```
      END IF
C
      IF(TURBO_TYP.EQ.'T'.AND.MOD(NR,2).EQ.0) ROWTYP = 'R'
      IF(TURBO_TYP.EQ.'T'.AND.MOD(NR,2).NE.0) ROWTYP = 'S'
      IF(TURBO_TYP.EQ.'C'.AND.MOD(NR,2).EQ.0) ROWTYP = 'S'
      IF(TURBO_TYP.EQ.'C'.AND.MOD(NR,2).NE.0) ROWTYP = 'R'
C
C
      WRITE(9,*) '*********************STARTING DATA FOR A NEW BLADE ROW
     &**************************'
      WRITE(9,126) NROW
      FORMAT('  BLADE ROW NUMBER = ',T25, I5)
      WRITE(9,127) ROWTYP
      FORMAT('  BLADE ROW  TYPE  = ',T25, A1)
C
C    INPUT THE NUMBER OF STREAMWISE GRID POINTS, UPSTREAM, ON AND DOWNSTREAM
C    OF THE BLADE ROW.
C
      NPOINTS_UP  = NINTUP
      IF(NR.EQ.1)     NPOINTS_UP  = NINTUP + NADDUP
      NPOINTS_DWN = NINTDWN
      IF(NR.EQ.NROWS) NPOINTS_DWN = NINTDWN + NADDWN
      NPOINTS_ON  = NINTON
      WRITE(9,1008)     NPOINTS_UP, NPOINTS_ON, NPOINTS_DWN
      FORMAT(3I5,T20, ' NPOINTS_UP, NPOINTS_ON, NPOINTS_DWN ')
C
C    SET THE RELATIVE SPACINGS OF THE GRID POINTS.
C
      WRITE(9,1001)   0.0,   0.5
      WRITE(9,1001)   0.1,   0.7
      WRITE(9,1001)   0.2,   1.0
      WRITE(9,1001)   0.3,   1.4
      WRITE(9,1001)   0.4,   2.0
      WRITE(9,1001)   0.5,   3.0
      WRITE(9,1001)   0.6,   3.0
      WRITE(9,1001)   0.7,   3.0
      WRITE(9,1001)   0.8,   2.5
      WRITE(9,1001)   0.9,   2.0
      WRITE(9,1001)   1.0,   1.5
      FORMAT(2F15.4,T35,' FRACTION AXIAL CHORD,  RELATIVE GRID SPACING')
C
      WRITE(9,1013) NBLADE(NR)
      FORMAT(I10,T20, ' NUMBER OF BLADES IN ROW. ')
C
      PIN   = PINLET(NSTG)*1.0E05
      PMIDD = PMID(NSTG)*1.0E05
      PEX   = PEXIT(NSTG)*1.0E05
      IF(TURBO_TYP.EQ.'T'.AND. ROWTYP.EQ.'S')
     & WRITE(9,1011) 0.0,PIN,PIN,PMIDD,PMIDD

      IF(TURBO_TYP.EQ.'C'.AND. ROWTYP.EQ.'R')
     & WRITE(9,1011) RPM,PIN,PIN,PMIDD,PMIDD

      IF(TURBO_TYP.EQ.'T'.AND. ROWTYP.EQ.'R')
     & WRITE(9,1011) RPM,PMIDD,PMIDD,PEX,PEX

      IF(TURBO_TYP.EQ.'C'.AND. ROWTYP.EQ.'S')
     & WRITE(9,1011) 0.0,PMIDD,PMIDD,PEX,PEX
C
      FORMAT(5F10.2,T55,'RPM, STATIC PRESSURES THROUGH ROW')
C
      RPMHUB = 0.0
      IF(ROWTYP.EQ.'R') RPMHUB = RPM
      WRITE(9,1014)  0, 0, 1, 1, 1 ,1, 0.0, RPMHUB
      FORMAT(6I5,F10.5,F10.2,T55,'TIP GAPS, WALL ROTNS and RPMHUB')
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
C**********************************************************************
C  NOW LOOP OVER ALL BLADE SECTIONS TO BE GENERATED.
C
C
```

```
      RDESIGNLE = RMEAN(NLE)
      RDESIGNTE = RMEAN(NTE)
C
C     MAKE AN INITIAL GUESS OF THE BLADE THICKNESSES, ETC.
C
      DO 4400 NS = 1,NOSECT
      IF(NSTG.EQ.1) THEN
           IF(ROWTYP.EQ.'S') TKMAX_S(NSTG,NS)  = TKMAXS
           IF(ROWTYP.EQ.'S') XTKMAX_S(NSTG,NS) = XTKMAXS
           IF(ROWTYP.EQ.'R') TKMAX_R(NSTG,NS)  = TKMAXR
           IF(ROWTYP.EQ.'R') XTKMAX_R(NSTG,NS) = XTKMAXR
      ELSE
           IF(ROWTYP.EQ.'S') TKMAX_S(NSTG,NS)  = TKMAX_S(NSTG-1,NS)
           IF(ROWTYP.EQ.'S') XTKMAX_S(NSTG,NS) = XTKMAX_S(NSTG-1,NS)
           IF(ROWTYP.EQ.'R') TKMAX_R(NSTG,NS)  = TKMAX_R(NSTG-1,NS)
           IF(ROWTYP.EQ.'R') XTKMAX_R(NSTG,NS) = XTKMAX_R(NSTG-1,NS)
      END IF
      CONTINUE
C
C*************************************************************************
C*************************************************************************
C     START A LOOP OVER ALL BLADE SECTIONS TO BE GENERATED
C
      DO 4500 NSECT = 1, NOSECT
C
      FRACSPAN = FLOAT(NSECT-1)/FLOAT(NOSECT-1)
C
c   SET THE LEADING AND TRAILING EDGE RADII FOR THIS SECTION.
      RSECTLE  = RHUB(NLE) + FRACSPAN*(RTIP(NLE) - RHUB(NLE))
      RSECTTE  = RHUB(NTE) + FRACSPAN*(RTIP(NTE) - RHUB(NTE))
C
C    SET THE COORDINATES AT THIS SPANWISE POSITION.
      DO 1012 N= 1,NINTPTS
      XSECT(N) = XSURFHUB(N) + FRACSPAN*(XSURFTIP(N)-XSURFHUB(N))
      RSECT(N) = RSURFHUB(N) + FRACSPAN*(RSURFTIP(N)-RSURFHUB(N))
      CONTINUE

      WRITE(9,*) '********************** ROW NUMBER',NR,'**************
     &**********************'
      WRITE(9,*) '**********STARTING NEW BLADE SECTION, SECTION NUMBER',
     & NSECT,'***************'
      WRITE(9,*) '*************************BLANK LINE*******************
     &**********************'
C
      WRITE(9,1015) 1
      FORMAT(I5, T25, ' INTYPE- TYPE OF BLADE GEOMETRY INPUT')
      WRITE(9,1007)  6 , 200,  4
      FORMAT(3I5,T25, ' NPIN, NXPTS, NSMOOTH ')
C
C
C*************************************************************************
C*************************************************************************
C     SET THE BLADE ANGLES AT THE DESIGN RADIUS
C
C     SET BLADE METAL ANGLES AT THE DESIGN RADIUS FOR THE FIRST ROW
      IF(MOD(NR,2).GT.0)  THEN
      IF(BIN_ROW1(NSTG).GT.BOUT_ROW1(NSTG)) THEN
           BLE_DES   = BIN_ROW1(NSTG)  - AINC1(NSTG)*DEGRAD
           BTE_DES   = BOUT_ROW1(NSTG) - DEVN1(NSTG)*DEGRAD
      ELSE
           BLE_DES   = BIN_ROW1(NSTG)  + AINC1(NSTG)*DEGRAD
           BTE_DES   = BOUT_ROW1(NSTG) + DEVN1(NSTG)*DEGRAD
      ENDIF
      END IF
C
C     SET BLADE METAL ANGLES AT THE DESIGN RADIUS FOR THE SECOND ROW
      IF(MOD(NR,2).EQ.0) THEN
      IF(BOUT_ROW2(NSTG).GT.BIN_ROW2(NSTG)) THEN
           BLE_DES  = BIN_ROW2(NSTG)   + AINC2(NSTG)*DEGRAD
           BTE_DES  = BOUT_ROW2(NSTG)  + DEVN2(NSTG)*DEGRAD
      ELSE
           BLE_DES  = BIN_ROW2(NSTG)   - AINC2(NSTG)*DEGRAD
           BTE_DES  = BOUT_ROW2(NSTG)  - DEVN2(NSTG)*DEGRAD
```

```
      ENDIF
      ENDIF

C*************************************************************************
C*************************************************************************
C     VARY THE ANGLES WITH SPAN FOR A FREE VORTEX DESIGN.
C
C    FIRST FOR A STATOR
      IF(ROWTYP.EQ.'S') THEN
               BLE   = (1.0 - FRAC_TWIST)*BLE_DES +
     &                 FRAC_TWIST*ATAN( TAN(BLE_DES)*RDESIGNLE/RSECTLE )
               BTE   = (1.0 - FRAC_TWIST)*BTE_DES +
     &                 FRAC_TWIST*ATAN( TAN(BTE_DES)*RDESIGNTE/RSECTTE )
      WRITE(6,*)
      WRITE(6,*) ' STAGE No ',NSTG,'STATOR  ROW NUMBER',NR,'SECTION No',
     & NSECT
      IF(TURBO_TYP.EQ.'C') THEN
           WRITE(6,*) ' INCIDENCE ANGLE = ', AINC2(NSTG)
           WRITE(6,*) ' DEVIATION ANGLE = ', DEVN2(NSTG)
      ELSE
           WRITE(6,*) ' INCIDENCE ANGLE = ', AINC1(NSTG)
           WRITE(6,*) ' DEVIATION ANGLE = ', DEVN1(NSTG)
      END IF
           WRITE(6,*) ' BLADE INLET METAL ANGLE = ', BLE*RADDEG
           WRITE(6,*) ' BLADE EXIT METAL ANGLE  = ', BTE*RADDEG
C   END OF ROWTYP = 'S' LOOP
      END IF
C
C   NEXT FOR A ROTOR
      IF(ROWTYP.EQ.'R') THEN
               RRAT     = RDESIGNLE/RSECTLE
               PHILEE   = PHI_LOC(NLE)
               TAN_BABS = TAN(BLE_DES) + 1.0/PHILEE
               TAN_BABS = TAN_BABS*RRAT
               TAN_BLE  = TAN_BABS - 1.0/PHILEE/RRAT
               BLE      = (1.0 - FRAC_TWIST)*BLE_DES
     &                  + FRAC_TWIST*ATAN(TAN_BLE)
C
               RRAT     = RDESIGNTE/RSECTTE
               PHITEE   = PHI_LOC(NTE)
               TAN_BABS = TAN(BTE_DES) + 1.0/PHITEE
               TAN_BABS = TAN_BABS*RRAT
               TAN_BTE  = TAN_BABS - 1.0/PHITEE/RRAT
               BTE      = (1.0 - FRAC_TWIST)*BTE_DES
     &                  + FRAC_TWIST*ATAN(TAN_BTE)
C
      WRITE(6,*)
      WRITE(6,*) ' STAGE No ',NSTG,'ROTOR  ROW NUMBER',NR,'SECTION No',
     & NSECT
      IF(TURBO_TYP.EQ.'C') THEN
           WRITE(6,*) ' INCIDENCE ANGLE = ', AINC1(NSTG)
           WRITE(6,*) ' DEVIATION ANGLE = ', DEVN1(NSTG)
      ELSE
           WRITE(6,*) ' INCIDENCE ANGLE = ', AINC2(NSTG)
           WRITE(6,*) ' DEVIATION ANGLE = ', DEVN2(NSTG)
      END IF
           WRITE(6,*) ' BLADE INLET METAL ANGLE = ', BLE*RADDEG
           WRITE(6,*) ' BLADE EXIT METAL ANGLE  = ', BTE*RADDEG
C    END OF ROWTYP = 'R' LOOP
      END IF
C
C*************************************************************************
C*************************************************************************
C    VARY THE TANGENT OF THE BLADE ANGLE WITH MERIDIONAL DISTANCE.
C    THE DISTANCE IS TRANSFORMED BY  "EXPO"  WHICH IS SET IN THE DEFAULTS.
C
      TAN1 = TAN(BLE)
      TAN6 = TAN(BTE)
      TAN2 = TAN1 +     (TAN6 - TAN1)/5.
      TAN3 = TAN1 + 2.0*(TAN6 - TAN1)/5.
      TAN4 = TAN1 + 3.0*(TAN6 - TAN1)/5.
      TAN5 = TAN1 + 4.0*(TAN6 - TAN1)/5.
      WRITE(9,1003) 0.0,         ATAN(TAN1)*RADDEG
```

```
      WRITE(9,1003) 0.2**EXPO,   ATAN(TAN2)*RADDEG
      WRITE(9,1003) 0.4**EXPO   , ATAN(TAN3)*RADDEG
      WRITE(9,1003) 0.6**EXPO   , ATAN(TAN4)*RADDEG
      WRITE(9,1003) 0.8**EXPO   , ATAN(TAN5)*RADDEG
      WRITE(9,1003) 1.0,          ATAN(TAN6)*RADDEG
      FORMAT(2F12.4,T25,' BLADE CENTRE LINE ANGLES ' )
      BETADWN  = ATAN(TAN6)*RADDEG
C
C*******************************************************************************
C*******************************************************************************
C
      IF(NSECT.EQ.1) THEN
C
      WRITE(6,*)
      WRITE(6,*)'*****************************************************
     &*************************************'
      WRITE(6,*)'*****************************************************
     &*************************************'
      IF(ROWTYP.EQ.'R')
     &WRITE(6,*) 'STAGE NUMBER',NSTG,' ROW NUMBER',NR,' THIS IS A ROTOR'
      IF(ROWTYP.EQ.'S')
     &WRITE(6,*)'STAGE NUMBER',NSTG,' ROW NUMBER',NR,' THIS IS A STATOR'
      WRITE(6,*)'*****************************************************
     &*************************************'
      WRITE(6,*)'*****************************************************
     &*************************************'
      ANSTK = 'N'
C
      WRITE(6,*)
      WRITE(6,*)  ' THE CURRENT VALUES WERE OF BLADE THICKNESS AND POINT
     & OF MAXIMUM THICKNESS ARE:'
C
      IF(ROWTYP.EQ.'S') THEN
           DO NS = 1,NOSECT
                WRITE(6,1032) NS,TKMAX_S(NSTG,NS),XTKMAX_S(NSTG,NS)
           END DO
      END IF
C
      IF(ROWTYP.EQ.'R')  THEN
           DO NS = 1,NOSECT
                WRITE(6,1032) NS,TKMAX_R(NSTG,NS),XTKMAX_R(NSTG,NS)
           END DO
      END IF
C
      FORMAT(' SECTION No.',I5,
     &       '  MAX THICKNESS, POSITION OF MAX THICKNESS',2F12.4)
C
      WRITE(6,*)
      WRITE(6,*)   'DO YOU WANT TO ACCEPT THESE ? ANSWER "Y"  or  "N".'
      READ(5,*)     ANSTK
      IF(ANSTK.EQ.'y') ANSTK = 'Y'
      IF(ANSTK.NE.'Y') ANSTK = 'N'
      WRITE(6,*)   'ANSTK = ', ANSTK
C
      IF(ROWTYP.EQ.'S') WRITE(10,1030) ANSTK,NSTG
      FORMAT(A1,T6,'STATOR No.',I3,   ' SET ANSTK = "Y" TO USE THE SAME
     & BLADE SECTIONS AS THE LAST STAGE')
      IF(ROWTYP.EQ.'R') WRITE(10,1031) ANSTK,NSTG
      FORMAT(A1,T6,'ROTOR No. ',I3,   ' SET ANSTK = "Y" TO USE THE SAME
     & BLADE SECTIONS AS THE LAST STAGE')
C
C   END OF NSECT = 1 LOOP
      END IF
C
C*******************************************************************************
C*******************************************************************************
C
      IF(ROWTYP.EQ.'S') THEN
C
      IF(ANSTK.EQ.'N'.OR.ANSTK.EQ.'n') THEN
C
      WRITE(6,*)'*****************************************************'
      WRITE(6,*)' STAGE NUMBER', NSTG, ' SECTION NUMBER ',NSECT
```

```
      WRITE(6,*)' INPUT NEW VALUES OF "TKMAX" AND "XTKMAX" FOR A STATOR'
      WRITE(6,*)' THE CURRENT VALUES ARE- ',
     &                    TKMAX_S(NSTG,NSECT), XTKMAX_S(NSTG,NSECT)
      WRITE(6,*)' PRESS "A" TO ACCEPT THESE OR TYPE IN NEW VALUES.'
      READ(5,*,ERR = 1009) TKMAX_S(NSTG,NSECT), XTKMAX_S(NSTG,NSECT)
      CONTINUE
      WRITE(6,*) ' THE NEW VALUES ARE-     ',
     &            TKMAX_S(NSTG,NSECT),XTKMAX_S(NSTG,NSECT)
      WRITE(6,*)'*************************************************'
C
      WRITE(10,123)TKMAX_S(NSTG,NSECT),XTKMAX_S(NSTG,NSECT),NSTG,NSECT
      FORMAT(2F8.4,T25,' MAX THICKNESS AND ITS LOCATION FOR STATOR',I3,
     &                 ' SECTION No.',I3)
C     END OF ANSTK = "N" LOOP
      END IF
C
      WRITE(9,1002) TKLE,TKTE,TKMAX_S(NSTG,NSECT),XTKMAX_S(NSTG,NSECT),
     &              XMODLE,XMODTE,TK_TYP
C
C     END OF ROWTYP = "S" LOOP .
      END IF
C
C*******************************************************************************
C*******************************************************************************
C
      IF(ROWTYP.EQ.'R') THEN
C
      IF(ANSTK.EQ.'N'.OR.ANSTK.EQ.'n') THEN
      WRITE(6,*)'*************************************************'
      WRITE(6,*)' STAGE NUMBER ', NSTG, ' SECTION NUMBER ',NSECT
      WRITE(6,*)' INPUT NEW VALUES OF "TKMAX"  AND "XTKMAX" FOR A ROTOR'
      WRITE(6,*)' THE CURRENT VALUES ARE- ',
     &                    TKMAX_R(NSTG,NSECT), XTKMAX_R(NSTG,NSECT)
      WRITE(6,*)' PRESS "A" TO ACCEPT THESE OR TYPE IN NEW VALUES.'
      READ(5,*,ERR = 1010) TKMAX_R(NSTG,NSECT), XTKMAX_R(NSTG,NSECT)
      CONTINUE
      WRITE(6,*) ' THE NEW VALUES ARE-     ',
     &            TKMAX_R(NSTG,NSECT),XTKMAX_R(NSTG,NSECT)
      WRITE(6,*)'*************************************************'
C
      WRITE(10,124)TKMAX_R(NSTG,NSECT),XTKMAX_R(NSTG,NSECT),NSTG,NSECT
      FORMAT(2F8.4,T25,' MAX THICKNESS AND ITS LOCATION FOR ROTOR ',I3,
     &                 ' SECTION No.',I3)
C     END OF ANSTK = "N" LOOP
      END IF
C
      WRITE(9,1002) TKLE,TKTE,TKMAX_R(NSTG,NSECT),XTKMAX_R(NSTG,NSECT),
     &              XMODLE,XMODTE,TK_TYP
C     END OF ROWTYP = "R" LOOP.
      END IF
      FORMAT(7F10.4,T75,' BLADE PROFILE SPECIFICATION')
C
C*******************************************************************************
C*******************************************************************************
C*******************************************************************************
C
      ROTN = 0.0
      XROT = 0.5
      YROT = 0.5
      IF(IF_ROT.EQ.'Y') THEN
      WRITE(6,*)
      WRITE(6,*) ' INPUT THE ANGLE BY WHICH THIS SECTION WILL BE TWISTED
     & IN THE CLOCKWISE DIRECTION, IN DEGREES.'
      READ(5,*,ERR = 1051) ROTN
      CONTINUE
      WRITE(10,1052) ROTN
      FORMAT(F10.4,T25, ' ANGLE OF CLOCKWISE ROTATION OF THIS SECTION.')
      END IF
C
C*******************************************************************************
C*******************************************************************************
C*******************************************************************************
```

```
C
      FCHORD   = 1.0
      FPERP    = 0.0
      FTKSCALE = 1.0
      WRITE(9,1004)         FCHORD, FPERP, FTKSCALE
      FORMAT(3F10.4,T50, ' FCHORD, FPERP, FTKSCALE')
C
C
      WRITE(9,1006)         ROTN,XROT,YROT
      FORMAT(3F10.4,T50, ' ROTN,XROT,YROT ' )
C
C
      XCUP   = 0.25
      XCDWN  = 0.25
      IF(NR.EQ.1)       XCUP = 0.5
      IF(NR.EQ.NROWS) XCDWN = 0.5
      BETUP  = ATAN(TAN1)*RADDEG
      BETDWN = BETADWN
      WRITE(9,1005)         XCUP, XCDWN, BETUP, BETDWN
      FORMAT(4F10.4,T50, ' XCUP, XCDWN, BETUP, BETDWN')
C
C
      WRITE(9,*)     ' BLANK LINE '
      WRITE(9,1023)    NINTPTS
      FORMAT(I5,T20,' NUMBER OF POINTS ON THE STREAM SURFACE.')
      WRITE(9,1016)  (XSECT(N),N=1,NINTPTS)
      WRITE(9,1017)  (RSECT(N),N=1,NINTPTS)
      WRITE(9,1018)   XSECT(NLEALL),XSECT(NTEALL),
     &                RSECT(NLEALL),RSECT(NTEALL)
      FORMAT(8F12.6)
      FORMAT(8F12.6)
      FORMAT(4F12.6,T50,' LEADING AND TRAILING EDGE COORDINATES')
C
C
      FCENTROID = 1.0
      FTANG     = 0.0
      FLEAN     = 0.0
      FSWEEP    = 0.0
      FAXIAL    = 0.0
      WRITE(9,1019)        FCENTROID, FTANG, FLEAN, FSWEEP, FAXIAL
      FORMAT(5F10.4,T50,' FCENTROID, FTANG, FLEAN, FSWEEP, FAXIAL')
C
      FSCALE = 1.0
      FCONST = 0.0
      WRITE(9,1020)         FSCALE, FCONST
      FORMAT(2F10.4,T50, ' FSCALE, FCONST ' )
C
C     END OF DATA FOR ONE STREAM SURFACE
C
      CONTINUE
C
C*************************************************************************
C*************************************************************************
C*************************************************************************

C     END OF DATA FOR THIS BLADE ROW
C
      CONTINUE
C
C     FIND THE INLET ENDWALL SLOPES, WHICH ARE USED TO SET THE INLET PITCH ANGLE .
      DXHUB    = XSURFHUB(2) - XSURFHUB(1)
      DXTIP    = XSURFTIP(2) - XSURFTIP(1)
      DRHUB    = RSURFHUB(2) - RSURFHUB(1)
      DRTIP    = RSURFTIP(2) - RSURFTIP(1)
      DSHUB    = SQRT(DXHUB*DXHUB + DRHUB*DRHUB)
      DSTIP    = SQRT(DXTIP*DXTIP + DRTIP*DRTIP)
      PITCHHUB = ATAN2(DRHUB,DXHUB)*RADDEG
      PITCHTIP = ATAN2(DRTIP,DXTIP)*RADDEG
C*************************************************************************
C*************************************************************************
C*************************************************************************
C
      WRITE(9,*) ' PUPHUB, PUPTIP, PDHUB,PDTIP '
```

```
2824         PIN = PINLET(1)*1.0E05
2825         PEX = PEXIT(NSTAGES)*1.0E05
2826         WRITE(9,1021)   PIN, PIN, PEX, PEX
2827         FORMAT(4F15.3,T65, ' INLET AND EXIT STATIC PRESSURES' )
2828         WRITE(9,*)          ' BLANK LINE '
2829   C
2830         IF(FLO_TYP.EQ.'AXI'.AND.TURBO_TYP.EQ.'C')
2831        &         YAWIN = BOUT_ROW2(1)*RADDEG
2832         IF(FLO_TYP.EQ.'AXI'.AND.TURBO_TYP.EQ.'T')
2833        &         YAWIN  =  BIN_ROW1(1)*RADDEG
2834         IF(FLO_TYP.EQ.'MIX') YAWIN =  ALPHA_IN(1)*RADDEG
2835         WRITE(9,1022)  2
2836         FORMAT(I5,T25,'NUMBER OF POINTS FOR INLET BOUNDARY CONDITIONS')
2837         WRITE(9,*)  0.0, 1.0,          '          FRAC SPAN AT INLET '
2838         WRITE(9,*)  PSTAGIN, PSTAGIN,  '          STAGNATION PRESSURE '
2839         WRITE(9,*)  TOIN, TOIN,        '          STAGNATION TEMPERATURE'
2840         WRITE(9,*)  0.0  , 0.0,        '          TANGENTIAL VELOCITY'
2841         WRITE(9,*)  VM_INLET, VM_INLET,'          MERIDIONAL VELOCITY '
2842         WRITE(9,*)  YAWIN,  YAWIN,     '          YAW ANGLE IN'
2843         WRITE(9,*)  PITCHHUB,PITCHTIP, '          PITCH ANGLE IN'
2844   C
2845   C*************************************************************************
2846   C    END OF OUTPUT TO "STAGEN.DAT" .
2847   C*************************************************************************
2848   C
2849         WRITE(6,*)
2850         WRITE(6,*) ' DESIGN NOW COMPLETED.'
2851         WRITE(6,*)
2852   C
2853         IF(IFWARNH.EQ.1) THEN
2854             WRITE(6,*)'WARNING! THE STREAM SURFACE POINTS WERE OVERLAPPING
2855        & ON THE HUB.'
2856             WRITE(6,*)'SOME POINTS HAVE BEEN MOVED AND THE BLADE SPACINGS
2857        &WILL HAVE CHANGED.'
2858         END IF
2859         IF(IFWARNT.EQ.1) THEN
2860             WRITE(6,*)'WARNING! THE STREAM SURFACE POINTS WERE OVERLAPPING
2861        & ON THE CASING.'
2862             WRITE(6,*)'SOME POINTS HAVE BEEN MOVED AND THE BLADE SPACINGS
2863        &WILL HAVE CHANGED.'
2864         END IF
2865   C
2866         WRITE(6,*)
2867         WRITE(6,*)
2868        &' FILE "stagen.dat" WRITTEN AS INPUT TO PROGRAM "stagen". '
2869         WRITE(6,*)
2870         WRITE(6,*) ' FILE "meangen.out" IS A COPY OF THE INPUT JUST USED.'
2871         WRITE(6,*)
2872         STOP
2873         END
2874   C*************************************************************************
2875   C*************************************************************************
2876   C*************************************************************************
2877
2878   C
2879   C*************************************************************************
2880   C
2881         SUBROUTINE PROPS (J,IM,HO,S,P,T,RHO,WET,V,G,VS,NMAIN,
2882                           IPROPS,IWET)
2883   C
2884   C     ROUTINE TO FIND FLUID PROPERTIES CORRESPONDING TO GIVEN
2885   C     VALUES OF STAGNATION ENTHALPY (J/KG) AND ENTROPY (J/KG K).
2886   C
2887         PARAMETER(NG=99, NST=20, NSC= 11)
2888   C
2889         DIMENSION HO(NG),V(NG),S(NG),P(NG),T(NG),G(NG),VS(NG),
2890                   RHO(NG),WET(NG)
2891   C
2892         COMMON /SET7/ HOIN,SI,RGAS,CPGAS,POIN,TOIN,GAMM
2893   C
2894         IF(IPROPS.NE.1) GO TO 1
2895   C
2896   C     PERFECT GAS PROPERTIES.
```

```
C
      IWET = 0
      DO 11 I=1,IM
      G(I)   = GAMM
      GG     = G(I)/(G(I)-1.0)
      H      = HO(I) - 0.5*V(I)*V(I)
      P(I)   = POIN*((H/HOIN)**GG)*EXP((SI-S(I))/RGAS)
      T(I)   = H/CPGAS
      RHO(I) = P(I)/RGAS/T(I)*100000.0
      VS(I)  = SQRT(G(I)*RGAS*T(I))
      WET(I) = 0.0
      GO TO 12
C
C     STEAM PROPERTIES.
C
      CONTINUE
C
      DO 10 I=1,IM
      HKJ= (HO(I)-0.5*V(I)*V(I))/1000.
      SKJ=S(I)/1000.
      IF(SKJ.GT.8.63) SKJ=8.63
      IF(HKJ.LT.(2000.+293.*(SKJ-6.8)))HKJ=2000.+293.*(SKJ-6.8)
      HABS=HKJ*0.2308 - 448.25
      HSAT= 4647.0 - 405.2*SKJ + 18.7*SKJ*SKJ
      IF(SKJ.LT.6.7) HSAT=-3533.35 + 2039.03*SKJ -164.06*SKJ*SKJ
      IF(HKJ.LT.HSAT) GO TO 5
      HR=(HKJ -3125.)/475.
      SR=(SKJ -7.1)/0.7
      C1=.9991-.02728*SR+.04982*SR*SR-.01596*SR*SR*SR
      C2=.001964-.00655*SR+.007398*SR*SR-.01471*SR*SR*SR
      C3=.02477+.000779*SR-.01044*SR*SR+.002993*SR*SR*SR
      C4=-.004313-.001639*SR-.000766*SR*SR+.01267*SR*SR*SR
      FM=C1+C2*HR+C3*HR*HR+C4*HR*HR*HR
      RHO(I)=(HABS/303.23)**3.333 *EXP(2.2*(7.268-SKJ))/0.2029
      P(I)=3.0435*RHO(I)*HABS/303.23
      RHO(I)=RHO(I)*FM-0.0025
      C1=1.001-.02005*SR+.04543*SR*SR-.01537*SR*SR*SR
      C2=-.008486-.008498*SR+.006452*SR*SR-.006273*SR*SR*SR
      C3=.02274+.003286*SR-.01253*SR*SR-.008903*SR*SR*SR
      C4=-.008879-.001671*SR+.007301*SR*SR+.01092*SR*SR*SR
      FM=C1+C2*HR+C3*HR*HR+C4*HR*HR*HR
      P(I)=P(I)*FM
      C1=1.016- .03847*SR+.03418*SR*SR-.01174*SR*SR*SR
      C2=-.002987+.0009509*SR+.001699*SR*SR-.004955*SR*SR*SR
      C3=-.001568-.0008933*SR-.00569*SR*SR-.004383*SR*SR*SR
      C4=-.004404-.00373*SR+.00566*SR*SR+.006968*SR*SR*SR
      FM=C1+C2*HR+C3*HR*HR+C4*HR*HR*HR
      T(I)= HABS*2.2*FM
      VS(I)= SQRT(130000.*P(I)/RHO(I))
      G(I) =1.3
      WET(I)=0.0
      GO TO 10
      IF(SKJ.LT.6.7) GO TO 20
      DSDH=-.00033 +.00055*(SKJ+.00033*HKJ-.66)/(-0.1+.00055*HKJ)
      DSDH= 1.0/DSDH
      A= 4647.0 -HKJ +SKJ*DSDH
      B= -405.2-1.0*DSDH
      C= 18.7
      SSAT=-(B +SQRT(B*B-4*A*C))/(2*C)
      HSAT= 4647. -405.2*SSAT+18.7*SSAT*SSAT
      SR = (SSAT-7.608)*1.272
      P(I)=.47843-1.1055*SR+1.3003*SR*SR-1.039*SR*SR*SR+.65315*SR*SR*SR*
      SR-.3305*SR*SR*SR*SR*SR+.09311*SR*SR*SR*SR*SR*SR
      TSAT=80.207-57.067*SR+12.07*SR*SR-2.997*SR*SR*SR+.0689*SR*SR*SR*SR
      +.6145*SR*SR*SR*SR*SR
      ROSAT=3.3751+7.2791*SR+7.7026*SR*SR+5.4648*SR*SR*SR+2.9851*SR*SR*S
      R*SR+1.1687*SR*SR*SR*SR*SR +.2245*SR*SR*SR*SR*SR*SR
      GO TO 21
      DSDH=-.0017+.0008*(SKJ+.0017*HKJ-3.4)/(-0.6+.0008*HKJ)
      DSDH=1.0/DSDH
      A= -3533.35 -HKJ +SKJ*DSDH
      B=  2039.03 - DSDH
      C=  -164.06
```

```
      SSAT=-(B +SQRT(B*B-4*A*C))/(2*C)
      HSAT = -3533.35 +2039.03*SSAT - 164.06*SSAT*SSAT
      SR = (SSAT-6.355) * 2.1542
      P(I)=19.064-24.34*SR+13.601*SR*SR-3.424*SR*SR*SR+0.1561*SR*SR*SR*
      SR + 0.2642*SR*SR*SR*SR*SR -0.32049*SR*SR*SR*SR*SR*SR
      TSAT=209.99 -63.872*SR +3.4481*SR*SR +2.7869*SR*SR*SR +2.0565*SR*
      SR*SR*SR -0.91474*SR*SR*SR*SR*SR -1.6936*SR*SR*SR*SR*SR*SR
      ROSAT=0.10432 + 0.13022*SR +0.087203*SR*SR +0.035654*SR*SR*SR+0.00
          5*SR*SR*SR*SR +0.0052544*SR*SR*SR*SR*SR+.0040945*SR*SR*SR*SR*
      SR*SR
      GO TO 22
      IF(SSAT.LT.8.35) GO TO 22
      ROSAT = 28.6 + 75.6*(SSAT-8.4) +110.0*(SSAT-8.4)**2
      P(I)  =(49.5 -140.0*(SSAT-8.4) +163.0*(SSAT-8.4)**2)* 0.001
      T(I)  = TSAT + 273.16
      WET(I)=(HSAT-HKJ)/(HSAT-4.19*TSAT)
      RHO(I)=1.0/ROSAT/(1.0-WET(I))
      G(I)=1.12
      IF(WET(I).LT.0.01) G(I)=1.3-18.0*WET(I)
      VS(I)=SQRT(G(I)*100000.*P(I)/RHO(I))
      IWET=1
      CONTINUE
      CONTINUE
      RETURN
      END
C*********************************************************************
C*************************************************************************************
C*************************************************************************************
C
      SUBROUTINE SMOOTH(N1,N2,NSMOOTH,FSMOOTH,FRAC,VAR)
C
C   THIS ROUTINE MAKES THE QUANTITY TO BE SMOOTHED VARY LINEARLY WITH SURFACE DISTANCE
.
C
      PARAMETER(NG=99, NST=20, NSC= 11)
C
      DIMENSION FRAC(NG),VAR(NG),TEMP(NG)
C
      DO 10 ITS = 1,NSMOOTH
C
      DO N = N1,N2
      TEMP(N) = VAR(N)
      END DO
C
      DO N = N1+1, N2-1
      FLEFT  = FRAC(N)   - FRAC(N-1)
      FRIGHT = FRAC(N+1) - FRAC(N)
      AVG = ( FLEFT*TEMP(N+1) + FRIGHT*TEMP(N-1) )/(FLEFT + FRIGHT)
      VAR(N) = (1.0-FSMOOTH)*VAR(N) + FSMOOTH*AVG
      END DO
C
      CONTINUE
C
      RETURN
      END
C*************************************************************************************
C*************************************************************************************
C*************************************************************************************
C
      SUBROUTINE SMOOTH2(N1,N2,NSMOOTH,FSMOOTH,XVAL,RVAL)
C
C     THIS SUBROUTINE SMOOTHS BY MOVING EACH POINT ALONG A PERPENDICUAR TOWARDS THE
LINE
C     JOINING ITS TWO ADJACENT POINTS.
C
      PARAMETER(NG=99, NST=20, NSC= 11)
C
      DIMENSION XVAL(NG),RVAL(NG),TEMPX(NG),TEMPR(NG)
C
      DO 10 ITS = 1,NSMOOTH
C
      DO N = N1,N2
      TEMPX(N) = XVAL(N)
```

```
3041          TEMPR(N) = RVAL(N)
3042          END DO
3043    C
3044          DO N = N1, N2-2
3045          XVEC = TEMPX(N+2) - TEMPX(N)
3046          RVEC = TEMPR(N+2) - TEMPR(N)
3047          SVEC = SQRT(XVEC*XVEC + RVEC*RVEC)
3048          XVEC = XVEC/SVEC
3049          RVEC = RVEC/SVEC
3050          XDIF = TEMPX(N+1) - TEMPX(N)
3051          RDIF = TEMPR(N+1) - TEMPR(N)
3052          PROJ = XVEC*XDIF + RVEC*RDIF
3053          XNORM = XDIF - PROJ*XVEC
3054          RNORM = RDIF - PROJ*RVEC
3055          XVAL(N+1) = TEMPX(N+1) - FSMOOTH*XNORM
3056          RVAL(N+1) = TEMPR(N+1) - FSMOOTH*RNORM
3057          END DO
3058    C
3059          CONTINUE
3060    C
3061          RETURN
3062          END
3063    C*******************************************************************************
3064
```