

```

1 C
2 C THE DIMENSIONS ARE SET BY THE FOLLOWING PARAMETER STATEMENT.:
3 C NG = TOTAL NUMBER OF POINTS USED ON THE STREAM SURFACE.
4 C NST = NUMBER OF STAGES.
5 C NSC = NUMBER OF BLADE SECTIONS TO BE GENERATED.
6 C PARAMETER(NG=99, NST=20, NSC= 11)
7 C
8 DIMENSION
9 & HO(NG), V(NG), S(NG), P(NG), T(NG), G(NG), VS(NG), RHO(NG),
10 & WET(NG), PSI(NG), PHI(NG), RHOMID(NG), RHOEXIT(NG), HEXIT(NG),
11 & DHO(NG), U(NG), VXOUT(NG), SMID(NG), SEXIT(NG), VMRAT(NG),
12 & VXMID(NG), HOEXIT(NG), HOMID(NG), HMID(NG), PO(NG), PMID(NG),
13 & PEXIT(NG), ETA(NG), ASPN(NG), ASPR(NG), SPAN(NG),
14 & RMEANALL(NST,NG), XMEANALL(NST,NG), VMLOCALL(NST,NG),
15 & RHUBALL(NST,NG), RTIPALL(NST,NG), NLE1_ALL(NST), NTE1_ALL(NST),
16 & XHUBALL(NST,NG), XTIPALL(NST,NG), NLE2_ALL(NST), NTE2_ALL(NST),
17 & NSS_STG(NST), NLE1_STG(NST), NTE1_STG(NST),
18 & NLE2_STG(NST), NTE2_STG(NST), ALPHA_IN(NG), ALPHA_OUT(NG),
19 & ROWGAP(NST), STAGEGAP(NST), DEVN1(NG), DEVN2(NG), AINC1(NG),
20 & AINC2(NG)
21 C
22 COMMON /SET7/ HOIN,SI, RGAS,CPGAS,POIN,TOIN,GAMM
23 C
24 DIMENSION RHUB(NG), RTIP(NG), XHUB(NG), XTIP(NG), DHOIS(NG), VX(NG),
25 & REACN(NG), RDES(NG), VIN(NG), AXCHRD1(NG), AXCHRD2(NG),
26 & XSURFHUB(NG), XSURFTIP(NG), RSURFHUB(NG), RSURFTIP(NG),
27 & SDISTHUB(NG), SDISTTIP(NG)
28 C
29 DIMENSION NBLADE(NG), XSECT(NG), RSECT(NG),
30 & PSTATIN(NG), PSTATOUT(NG), PROTOU(NG),
31 & HINLET(NG), SINLET(NG), PINLET(NG), VM(NG),
32 & XMEAN(NG), RMEAN(NG), VMER(NG), SDIST(NG), PITCH_ANGL(NG),
33 & FBLOCK_LE(NST), FBLOCK_TE(NST), FBLOCK(NG)
34 C
35 DIMENSION BIN_ROW1(NG), BOUT_ROW1(NG), BIN_ROW2(NG), BOUT_ROW2(NG),
36 & QLE_ROW1(NG), QTE_ROW1(NG), QLE_ROW2(NG), QTE_ROW2(NG)
37 C
38 DIMENSION RHINLET(NG), HOINLET(NG), VXIN(NG),
39 & TIN(NG), TMID(NG), TEXIT(NG), POREL(NG), POABS(NG),
40 & VRELIN(NG), VABSIN(NG), VRELMID(NG), VABSMID(NG),
41 & VRELEX(NG), VABSEX(NG), RHUBIN(NG), RHUBMID(NG),
42 & RHUBEXIT(NG), RTIPIN(NG), RTIPMID(NG), RTIPEXIT(NG),
43 & TKMAX_S(NST,NSC), XTKMAX_S(NST,NSC), TINLET(NG),
44 & TKMAX_R(NST,NSC), XTKMAX_R(NST,NSC)
45 C
46 DIMENSION VABS(NG), VREL(NG), HOLOC(NG), SLOC(NG), PLOC(NG),
47 & TLOC(NG), RHOLOC(NG), PHI_LOC(NG), VM_LOC(NG), U_LOC(NG),
48 & VTLOC(NG)
49 C
50 REAL MACH_REL(NG), MACH_ABS(NG)
51 C
52 CHARACTER*10 IFSAME_RAD, IF_RDES, IFHUB,
53 & IFSAME_ADM, IFSAME_FLO, IFSAME_ANG, RADTYPE
54 CHARACTER*1 INTYPE, ASP_TYP, ROWTYP, TURBO_TYP, ANSTK, ANSFLO,
55 & IFSAME_ALL, ANSSS, ANS, ANSANG, MIXTYP, ANSOUT, ANSIN,
56 & IFOUT(NG), IF_ROT
57 CHARACTER*3 FLO_TYP
58 CHARACTER*72 DUMMY_LINE
59 C
60 OPEN(UNIT=10, FILE= 'meangen.out')
61 OPEN(UNIT=5, FILE= '/dev/tty')
62 C
63 PI = 3.14159
64 DEGRAD = PI/180.
65 RADDEG = 180./PI
66 DEG = PI/180.
67 C
68 WRITE(6,*) '*****'
69 &*****'
70 WRITE(6,*) '*****'
71 &*****'
72 WRITE(6,*)
73 WRITE(6,*) ' WELCOME TO MEANGEN '
```

```

74      WRITE(6,*)
75      WRITE(6,*) 'THIS IS AN INTERACTIVE PROGRAM FOR THE ONE-DIMENSIONAL'
76      WRITE(6,*) '                DESIGN OF AXIAL TURBOMACHINES.'
77      WRITE(6,*)
78      WRITE(6,*) 'ANSWER THE QUESTIONS AS THEY APPEAR ON THE SCREEN '
79      WRITE(6,*) 'AND THE PROGRAM WILL WRITE A DATA SET FOR THE'
80      WRITE(6,*) 'BLADE GEOMETRY PROGRAM "STAGEN" WHICH IN TURN WILL'
81      WRITE(6,*) 'GENERATE A 3D DATASET FOR "MULTALL-OPEN".'
82      WRITE(6,*)
83      WRITE(6,*) '*****'
84      &*****'
85      WRITE(6,*) '*****'
86      &*****'
87      C
88      WRITE(6,*) ' INPUT FROM SCREEN OR FILE ? '
89      WRITE(6,*) ' ANSWER "S" or "F" .'
90      READ(5,*) ANSIN
91      IF(ANSIN.EQ.'F'.OR.ANSIN.EQ.'f') THEN
92          ANSIN = 'F'
93          CLOSE(5)
94          OPEN(UNIT=5, FILE='meangen.in' )
95      END IF
96      C
97      WRITE(6,*) '*****'
98      WRITE(6,*) '*****'
99      WRITE(6,*)
100     WRITE(6,*) '          IS THIS A COMPRESSOR OR A TURBINE ?'
101     WRITE(6,*) '          ANSWER "C" or "T" .'
102         READ(5,*) TURBO_TYP
103         IF(TURBO_TYP.EQ.'t') TURBO_TYP= 'T'
104         IF(TURBO_TYP.EQ.'c') TURBO_TYP= 'C'
105         WRITE(10,101) TURBO_TYP
106     FORMAT(A1,T25,' TURBO_TYP,"C" FOR A COMPRESSOR,"T" FOR A TURBINE')
107     WRITE(6,*) '*****'
108     WRITE(6,*) '*****'
109     C
110     WRITE(6,*)
111     C
112     WRITE(6,*) ' DO YOU WANT TO DESIGN AN AXIAL FLOW MACHINE WITH A
113     &CONSTANT RADIUS AT A FIXED SPANWISE POSITION ON EACH STAGE ?'
114     WRITE(6,*) ' AND WITH REPEATING FLOW CONDITIONS.'
115     WRITE(6,*) ' OR A MIXED FLOW MACHINE WITH SIGNIFICANT CHANGES IN
116     & RADIUS THROUGH A STAGE ?'
117     WRITE(6,*) ' ANSWER "AXI" or "MIX" '
118     READ(5,*) FLO_TYP
119     IF(FLO_TYP.EQ.'axi') FLO_TYP = 'AXI'
120     IF(FLO_TYP.EQ.'mix') FLO_TYP = 'MIX'
121     WRITE(10,5) FLO_TYP
122     FORMAT(A3,T25,' FLO_TYP FOR AXIAL OR MIXED FLOW MACHINE ')
123     C
124     WRITE(6,*)
125     WRITE(6,*)
126     & 'THE BLADE ROTATION MUST BE IN THE POSITIVE THETA DIRECTION.'
127     WRITE(6,*)
128     WRITE(6,*)
129     & 'ALL FLOW ANGLES ARE POSITIVE IF THE ASSOCIATED FLOW VECTOR HAS
130     & A POSITIVE COMPONENT IN THE DIRECTION OF ROTATION.'
131     ROTN = 1.0
132     WRITE(6,*) '*****'
133     WRITE(6,*) '*****'
134     WRITE(6,*)
135
136     C*****
137     C*****
138     C    SET DEFAULTS
139     C*****
140     C*****
141     C
142     IF(TURBO_TYP.EQ.'T') THEN !    DEFAULTS FOR TURBINES.
143         TKLE = 0.04 ! LEADING EDGE THICKNESS/AXIAL CHORD.
144         TKTE = 0.04 ! TRAILING EDGE THICKNESS/AXIAL CHORD.
145         TKMAXS = 0.30 ! STATOR MAXIMUM THICKNESS/AXIAL CHORD.
146         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 FROM 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.

```

```

207 FPRAT = 1.25 ! GRID EXPANSION RATIO IN THE PITCHWISE DIRECTION.
208 FPMAX = 20.0 ! MAXIMUM GRID EXPANSION IN THE PITCHWISE DIRECTION.
209 FRRAT = 1.25 ! GRID EXPANSION RATIO IN THE SPANWISE DIRECTION.
210 FRMAX = 20.0 ! MAXIMUM GRID EXPANSION IN THE SPANWISE DIRECTION.
211 C
212 C*****
213 C*****
214 C END OF SETTING DEFAULTS.
215 C*****
216 C*****
217 C
218 WRITE(6,*)
219 WRITE(6,*) '*****'
220 WRITE(6,*) '*****'
221 WRITE(6,*)
222 & 'INPUT THE GAS CONSTANT IN J/KG K, AND GAS SPECIFIC HEAT RATIO.'
223 WRITE(6,*)
224 & 'THE DEFAULT VALUES ARE THOSE FOR AIR, RGAS = 287.15, GAMMA=1.4.'
225 WRITE(6,*) ' TYPE "A" TO ACCEPT THESE, OR TYPE IN NEW VALUES.'
226 READ(5,*,ERR=1111) RGAS, GAMM
227 CONTINUE
228 WRITE(10,102) RGAS, GAMM
229 FORMAT(2F10.3, T25, ' GAS PROPERTOES, RGAS, GAMMA ')
230 CPGAS = RGAS*GAMM/(GAMM-1.)
231 FGA = GAMM/(GAMM - 1.0)
232 C
233 WRITE(6,*)
234 WRITE(6,*) '*****'
235 WRITE(6,*) '*****'
236 WRITE(6,*)
237 WRITE(6,*)
238 & 'INPUT THE INLET STAGNATION PRESSURE IN BAR AND INLET TEMPERATURE
239 & IN DEG K'
240 READ(5,*) POIN, TOIN
241 WRITE(10,103) POIN, TOIN
242 FORMAT(2F10.3, T25, ' POIN, TOIN ')
243 HOIN = CPGAS*TOIN
244 PSTAGIN = POIN*1.0E05
245 WRITE(6,*) '*****'
246 WRITE(6,*) '*****'
247 C
248 WRITE(6,*)
249 WRITE(6,*) ' INPUT THE NUMBER OF STAGES IN THE MACHINE.'
250 READ(5,*) NSTAGES
251 WRITE(10,104) NSTAGES
252 FORMAT(I5, T25, ' NUMBER OF STAGES IN THE MACHINE ')
253 WRITE(6,*) ' NUMBER OF STAGES = ', NSTAGES
254 NROWS = 2*NSTAGES
255 WRITE(6,*)
256 WRITE(6,*) '*****'
257 WRITE(6,*) '*****'
258 WRITE(6,*)
259 WRITE(6,*) ' BASE THE DESIGN ON THE HUB, MEAN OR TIP RADIUS ? '
260 WRITE(6,*) ' INPUT "H" , "M" or "T" '
261 READ(5,*) IFHUB
262 IF(IFHUB.EQ.'h') IFHUB = 'H'
263 IF(IFHUB.EQ.'m') IFHUB = 'M'
264 IF(IFHUB.EQ.'t') IFHUB = 'T'
265 WRITE(10,106) IFHUB
266 FORMAT(A1, T25, ' CHOICE OF DESIGN POINT RADIUS, HUB, MID or TIP')
267 WRITE(6,*)
268 WRITE(6,*) '*****'
269 WRITE(6,*) '*****'
270 C
271 C*****
272 C*****
273 C CALL PROPS TO GET THE INLET STAGNATION CONDITIONS
274 C
275 HO(1) = HOIN
276 PO(1) = POIN
277 S(1) = 0.0
278 SI = 0.0
279 VIN(1) = 10.0

```

```

280 C
281 CALL PROPS(1,1,HO(1),S(1),PO(1),T(1),RHO(1),WET(1),
282 & VIN(1),G(1),VS(1),1,IPROPS,IWET)
283 C
284 C
285 WRITE(6,*) '*****'
286 WRITE(6,*) '*****'
287 WRITE(6,*)
288 WRITE(6,*) ' INLET STAGNATION PRESSURE = ', PO(1), ' BAR.'
289 WRITE(6,*) ' INLET STAGNATION TEMPERATURE = ', T(1), ' K. '
290 WRITE(6,*) ' INLET STAGNATION DENSITY = ', RHO(1), ' Kg/M3. '
291 WRITE(6,*)
292 WRITE(6,*) '*****'
293 WRITE(6,*) '*****'
294 C
295 C
296 WRITE(6,*) '*****'
297 WRITE(6,*) '*****'
298 WRITE(6,*)
299 WRITE(6,*) ' INPUT THE ROTATION SPEED IN RPM, IT MUST BE POSITIVE.'
300 READ(5,*) RPM
301 WRITE(6,*) ' RPM = ', RPM
302 WRITE(10,107) RPM
303 FORMAT(F12.3,T25, ' ROTATION SPEED, RPM ')
304 WRITE(6,*) '*****'
305 WRITE(6,*) '*****'
306 WRITE(6,*)
307 C
308 C
309 WRITE(6,*) '*****'
310 WRITE(6,*) '*****'
311 WRITE(6,*)
312 WRITE(6,*) ' INPUT THE REQUIRED INLET MASS FLOW RATE IN kg/sec.'
313 READ(5,*) FLOWIN
314 WRITE(10,108) FLOWIN
315 FORMAT(F12.3,T25, ' MASS FLOW RATE, FLOWIN. ')
316 WRITE(6,*) ' INLET FLOW = ', FLOWIN
317 FLOW = FLOWIN
318 WRITE(6,*)
319 WRITE(6,*) '*****'
320 WRITE(6,*) '*****'
321 C
322 OMEGA = RPM*PI/30.
323 DHO(1) = 0.0
324 DHOTOTAL = 0.0
325 C
326 C*****
327 C*****
328 C START THE LOOP OVER NSTG STAGES. RETURN TO "1100" AFTER EVERY STAGE.
329 C*****
330 C*****
331 C*****
332 NSTG = 0
333 CONTINUE
334 C
335 C*****
336 C*****
337 NSTG = NSTG + 1
338 C
339 NROW = 2*NSTG -1
340 C
341
342 WRITE(6,*)
343 WRITE(6,*) '*****'
344 &*****
345 WRITE(6,*) '*****'
346 &*****
347 WRITE(6,*) ' STARTING STAGE NUMBER ', NSTG
348 WRITE(6,*) '*****'
349 &*****
350 WRITE(6,*) '*****'
351 &*****
352 WRITE(6,*)

```

```

353 C
354 C
355 IF(NSTG.GT.1) THEN
356 C
357 IFSAME_ALL = 'N'
358 C
359 WRITE(6,*) '*****'
360 WRITE(6,*) '*****'
361 WRITE(6,*)
362 WRITE(6,*) ' ARE THE ANGLES, MASS FLOW, DESIGN RADIUS, '
363 WRITE(6,*) ' EFFICIENCY, ETC, FOR THIS STAGE "ALL" THE SAME AS '
364 WRITE(6,*) ' FOR THE LAST STAGE ? '
365 WRITE(6,*)
366 IF(FLO_TYP.EQ.'AXI') THEN
367 WRITE(6,*) ' ANSWER "Y" or "N", OR ANSWER "C" TO CHANGE FROM '
368 WRITE(6,*) ' "FLO_TYP" = "AXI" TO "FLO_TYP" = "MIX". '
369 ELSE
370 WRITE(6,*) ' ANSWER "Y" or "N", OR ANSWER "C" TO CHANGE FROM '
371 WRITE(6,*) ' "FLO_TYP" = "MIX" TO "FLO_TYP" = "AXI". '
372 END IF
373 C
374 READ(5,*) IFSAME_ALL
375 WRITE(6,*) ' IFSAME_ALL= ', IFSAME_ALL
376 WRITE(10,109) IFSAME_ALL
377 IF(IFSAME_ALL.EQ.'y') IFSAME_ALL = 'Y'
378 IF(IFSAME_ALL.EQ.'n') IFSAME_ALL = 'N'
379 IF(IFSAME_ALL.EQ.'c') IFSAME_ALL = 'C'
380 C
381 IF(IFSAME_ALL.EQ.'C'.AND.(FLO_TYP.EQ.'AXI')) THEN
382 FLO_TYP = 'MIX'
383 GO TO 2000
384 END IF
385 IF(IFSAME_ALL.EQ.'C'.AND.(FLO_TYP.EQ.'MIX')) THEN
386 FLO_TYP = 'AXI'
387 MIXTYP = 'N'
388 GO TO 500
389 END IF
390 C
391 FORMAT(A1,T25,' IFSAME_ALL, SET = "Y" TO REPEAT THE LAST STAGE INP
392 &UT TYPE AND VELOCITY TRIANGLES, SET = "C" TO CHANGE INPUT TYPE.')
393 C
394 IF(FLO_TYP.EQ.'AXI'.AND.(IFSAME_ALL.EQ.'Y')) GO TO 600
395 IF(FLO_TYP.EQ.'MIX'.AND.(IFSAME_ALL.EQ.'Y')) GO TO 700
396 C
397 C END OF NSTAGE GT 1 LOOP
398 END IF
399 C
400
401 C RE ENTER HERE IF CHANGING THE ANGLES FOR THIS STAGE.
402 C
403 CONTINUE
404 C
405 C*****
406 C*****
407 C
408 IF(FLO_TYP.EQ.'MIX') GO TO 2000
409 C
410 C*****
411 C*****
412 C
413 C SET THE VELOCITY TRIANGLES FOR THE STAGE FOR FLO_TYP = "AXI".
414 C
415 WRITE(6,*)
416 WRITE(6,*)
417 WRITE(6,*) '*****'
418 WRITE(6,*) '*****'
419 WRITE(6,*)
420 WRITE(6,*) ' YOU MAY SPECIFY THE STAGE VELOCITY TRIANGLES IN ONE
421 &OF 3 WAYS '
422 WRITE(6,*)
423 WRITE(6,*) ' METHOD "A"-SPECIFY THE REACTION, FLOW COEFFICIENT AND
424 & STAGE LOADING COEFFICIENT '
425 WRITE(6,*)

```

```

426     WRITE(6,*) ' METHOD "B"- SPECIFY THE FLOW COEFFICIENT, THE STATOR
427 &EXIT ANGLE AND THE ROTOR EXIT ANGLE '
428     WRITE(6,*)
429     WRITE(6,*) ' METHOD "C"- SPECIFY THE FLOW COEFFICIENT, THE ROTOR
430 &INLET ANGLE AND THE ROTOR EXIT ANGLE '
431     WRITE(6,*)
432     WRITE(6,*) ' METHOD "D" SPECIFY THE STAGE REACTION,THE FIRST BLADE
433 &ROW INLET ANGLE AND THE FIRST BLADE ROW EXIT ANGLE '
434     WRITE(6,*)
435 C
436     WRITE(6,*)
437     WRITE(6,*) '
438     WRITE(6,*) '
439 C
440     READ(5,*) INTYPE
441     IF(INTYPE.EQ.'a') INTYPE ='A'
442     IF(INTYPE.EQ.'b') INTYPE ='B'
443     IF(INTYPE.EQ.'c') INTYPE ='C'
444     IF(INTYPE.EQ.'d') INTYPE ='D'
445 C
446     WRITE(10,110) INTYPE
447     FORMAT(A1,T25,' INTYPE, TO CHOOSE THE METHOD OF DEFINING THE VELOC
448 &ITY TRIANGLES')
449 C
450     WRITE(6,*)
451     WRITE(6,*) ' ***** '
452     WRITE(6,*) ' ***** '
453 C
454 C
455 C
456 C
457     IF(INTYPE.EQ.'A') THEN
458     WRITE(6,*) ' INTYPE = "A" CHOSEN.'
459 C
460     IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE.'
461     IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR.'
462     WRITE(6,*)
463     WRITE(6,*) ' INPUT THE REACTION, FLOW COEFFICIENT AND STAGE LOADIN
464 &G COEFFICIENT. '
465     READ(5,*) REACN(NSTG), PHI(NSTG), PSI(NSTG)
466     WRITE(6,*) REACN(NSTG), PHI(NSTG), PSI(NSTG)
467     WRITE(10,111) REACN(NSTG), PHI(NSTG), PSI(NSTG)
468     FORMAT(3F7.3,T25,' REACTION, FLOW COEFF., LOADING COEFF. ')
469 C
470     IF(TURBO_TYP.EQ.'T') THEN
471     BIN_ROW1(NSTG) =
472 & ATAN( (1.- REACN(NSTG) - 0.5*PSI(NSTG))/PHI(NSTG))
473     BOUT_ROW2(NSTG) =
474 & ATAN( TAN(BIN_ROW1(NSTG)) - 1.0/PHI(NSTG) )
475     BOUT_ROW1(NSTG) =
476 & ATAN( TAN(BIN_ROW1(NSTG)) + PSI(NSTG)/PHI(NSTG))
477     BIN_ROW2(NSTG) =
478 & ATAN( TAN(BOUT_ROW1(NSTG)) - 1.0/PHI(NSTG) )
479     WRITE(6,*)
480     WRITE(6,*) ' CALCULATED FLOW ANGLES FOR THIS STAGE '
481     WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
482 & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
483     END IF
484 C
485     IF(TURBO_TYP.EQ.'C') THEN
486     BIN_ROW2(NSTG) =
487 & ATAN( (1.-REACN(NSTG) + 0.5*PSI(NSTG))/PHI(NSTG))
488     BOUT_ROW1(NSTG) =
489 & ATAN( TAN(BIN_ROW2(NSTG)) - 1.0/PHI(NSTG) )
490     BIN_ROW1(NSTG) =
491 & ATAN( TAN(BOUT_ROW1(NSTG)) - PSI(NSTG)/PHI(NSTG))
492     BOUT_ROW2(NSTG) =
493 & ATAN( TAN(BIN_ROW1(NSTG)) + 1.0/PHI(NSTG))
494     WRITE(6,*)
495     WRITE(6,*) ' CALCULATED FLOW ANGLES FOR THIS STAGE '
496     WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,

```



```

497      &          BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
498      END IF
499 C  END OF INTYPE = "A" .
500      END IF
501 C
502 C
*****
503 C
*****
504 C
      IF(INTYPE.EQ.'B') THEN
505      WRITE(6,*) ' INTYPE = "B" CHOSEN. '
506 C
      IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE.'
507      IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR.'
508      WRITE(6,*)
509      WRITE(6,*) ' INPUT THE FLOW COEFFICIENT, STATOR EXIT ANGLE AND ROT
510      &OR EXIT ANGLE'
511      READ(5,*) PHI(NSTG), B2NOZ, B2ROT
512      WRITE(10,112) PHI(NSTG), B2NOZ, B2ROT
513      FORMAT(3F12.3, T25, ' FLOW COEFF, STATOR ANGLES ')
514 C
      IF(TURBO_TYP.EQ.'T') THEN
515      BOUT_ROW1(NSTG) = B2NOZ*DEGRAD
516      BOUT_ROW2(NSTG) = B2ROT*DEGRAD
517      BIN_ROW2(NSTG) = ATAN( TAN(BOUT_ROW1(NSTG)) - 1.0/PHI(NSTG) )
518      BIN_ROW1(NSTG) = ATAN( TAN(BOUT_ROW2(NSTG)) + 1.0/PHI(NSTG) )
519      REACN(NSTG) =
520      & -0.5*PHI(NSTG)*(TAN(BIN_ROW2(NSTG)) + TAN(BOUT_ROW2(NSTG)))
521      PSI(NSTG) =
522      & 2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW1(NSTG)))
523      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
524      & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
525      WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
526      END IF
527 C
      IF(TURBO_TYP.EQ.'C') THEN
528      BOUT_ROW2(NSTG) = B2NOZ*DEGRAD
529      BOUT_ROW1(NSTG) = B2ROT*DEGRAD
530      BIN_ROW1(NSTG) = ATAN( TAN(BOUT_ROW2(NSTG)) - 1.0/PHI(NSTG) )
531      BIN_ROW2(NSTG) = ATAN( TAN(BOUT_ROW1(NSTG)) + 1.0/PHI(NSTG) )
532      REACN(NSTG) =
533      & -0.5*PHI(NSTG)*(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
534      PSI(NSTG) =
535      & -2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW2(NSTG)))
536      WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
537      & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
538      WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
539      END IF
540 C
541 C  END OF INTYPE = "B" .
542      END IF
543 C
544 C
*****
545 C
*****
546 C
      IF(INTYPE.EQ.'C') THEN
547      WRITE(6,*) ' INPUT TYPE "C" CHOSEN.'
548 C
      IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE.'
549      IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR.'
550      WRITE(6,*)
551      WRITE(6,*) ' INPUT THE ROTOR INLET ANGLE, THE ROTOR EXIT ANGLE AND
552      & THE FLOW COEFFICIENT.'
553      READ(5,*) B1ROT, B2ROT, PHI(NSTG)
554      WRITE(10,113) B1ROT, B2ROT, PHI(NSTG)
555      FORMAT(3F12.3, T25, ' ROTOR ANGLES, FLOW COEFF. ')
556 C
      IF(TURBO_TYP.EQ.'T') THEN
557      BIN_ROW2(NSTG) = B1ROT*DEGRAD
558      BOUT_ROW2(NSTG) = B2ROT*DEGRAD

```



```

566     BIN_ROW1(NSTG) = ATAN( TAN(BOUT_ROW2(NSTG)) + 1.0/PHI(NSTG))
567     BOUT_ROW1(NSTG) = ATAN( TAN(BIN_ROW2(NSTG)) + 1.0/PHI(NSTG))
568     REACN(NSTG) =
569     & -0.5*PHI(NSTG)*(TAN(BIN_ROW2(NSTG)) + TAN(BOUT_ROW2(NSTG)))
570     PSI(NSTG) =
571     & 2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW1(NSTG)))
572     WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
573     & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
574     WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
575     END IF
576 C
577     IF(TURBO_TYP.EQ.'C') THEN
578     BIN_ROW1(NSTG) = B1ROT*DEGRAD
579     BOUT_ROW1(NSTG) = B2ROT*DEGRAD
580     BIN_ROW2(NSTG) = ATAN( TAN(BOUT_ROW1(NSTG)) + 1.0/PHI(NSTG))
581     BOUT_ROW2(NSTG) = ATAN( TAN(BIN_ROW1(NSTG)) + 1.0/PHI(NSTG))
582     REACN(NSTG) =
583     & -0.5*PHI(NSTG)*(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
584     PSI(NSTG) =
585     & -2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW2(NSTG)))
586     WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
587     & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
588     WRITE(6,*) ' REACTION= ', REACN(NSTG), ' LOADING = ', PSI(NSTG)
589     END IF
590 C     END OF INTYPE = "C" OPTION.
591     END IF
592 C
593 C
*****
594 C
*****
595 C
596     IF(INTYPE.EQ.'D') THEN
597     WRITE(6,*) 'INPUT TYPE "D" CHOSEN.'
598 C
599     IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE.'
600     IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR.'
601     WRITE(6,*)
602     WRITE(6,*) ' INPUT THE FIRST BLADE ROW INLET ANGLE, THE FIRST BLADE
603     & ROW EXIT ANGLE AND THE STAGE REACTION.'
604     READ(5,*) BIN_ROW1(NSTG), BOUT_ROW1(NSTG), REACN(NSTG)
605     WRITE(10,114) BIN_ROW1(NSTG), BOUT_ROW1(NSTG), REACN(NSTG)
606     FORMAT(3F12.3,T25,' FIRST ROW ANGLES, REACTION')
607     BIN_ROW1(NSTG) = BIN_ROW1(NSTG)*DEGRAD
608     BOUT_ROW1(NSTG) = BOUT_ROW1(NSTG)*DEGRAD
609 C
610     IF(TURBO_TYP.EQ.'T') THEN
611     PHI(NSTG) =
612     & 2*(1.0-REACN(NSTG))/(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
613     BIN_ROW2(NSTG) = ATAN( TAN(BOUT_ROW1(NSTG)) - 1.0/PHI(NSTG))
614     BOUT_ROW2(NSTG) = ATAN( TAN(BIN_ROW1(NSTG)) - 1.0/PHI(NSTG))
615     PSI(NSTG) =
616     & 2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW1(NSTG)))
617     WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
618     & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
619     WRITE(6,*) ' PHI= ', PHI(NSTG), ' LOADING = ', PSI(NSTG)
620     END IF
621 C
622     IF(TURBO_TYP.EQ.'C') THEN
623     PHI(NSTG) =
624     & -2*REACN(NSTG)/(TAN(BIN_ROW1(NSTG)) + TAN(BOUT_ROW1(NSTG)))
625     BIN_ROW2(NSTG) = ATAN( TAN(BOUT_ROW1(NSTG)) + 1.0/PHI(NSTG))
626     BOUT_ROW2(NSTG) = ATAN( TAN(BIN_ROW1(NSTG)) + 1.0/PHI(NSTG))
627     PSI(NSTG) =
628     & -2*(1.0 - REACN(NSTG) - PHI(NSTG)*TAN(BIN_ROW2(NSTG)))
629     WRITE(6,*) BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG,
630     & BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
631     WRITE(6,*) ' PHI= ', PHI(NSTG), ' LOADING = ', PSI(NSTG)
632     END IF
633 C     END OF INTYPE = "D" OPTION.
634     END IF
635 C
636 C*****

```

```

637 C*****
638 C
639 C NEXT SET THE ROTATIONAL SPEED AND DESIGN RADIUS. FOR FLO_TYP = "AXI" .
640 C
641 C*****
642 C*****
643 C
644 WRITE(6,*)
645 WRITE(6,*) ' YOU MAY SET THE DESIGN RADIUS IN ONE OF 2 WAYS'
646 WRITE(6,*)
647 WRITE(6,*) ' METHOD "A". INPUT THE DESIGN RADIUS DIRECTLY.'
648 WRITE(6,*) ' METHOD "B". INPUT STAGE ENTHALPY CHANGE.'
649 WRITE(6,*)
650 WRITE(6,*) ' CHOOSE METHOD "A" OR "B".'
651 WRITE(6,*) ' TYPE IN "A" or "B". '
652 WRITE(6,*)
653 READ(5,*) RADTYPE
654 IF(RADTYPE.EQ.'a') RADTYPE = 'A'
655 IF(RADTYPE.EQ.'b') RADTYPE = 'B'
656 WRITE(6,*) ' RADTYPE = ', RADTYPE
657 WRITE(10,115) RADTYPE
658 FORMAT(A1,T25,' RADTYPE, TO CHOOSE THE DESIGN POINT RADIUS')
659 C
660 IF(RADTYPE.EQ.'A') THEN
661 WRITE(6,*) ' INPUT THE DESIGN RADIUS IN METRES.'
662 READ(5,*) RDES(NSTG)
663 WRITE(6,*) ' DESIGN POINT RADIUS =', RDES(NSTG)
664 WRITE(10,116) RDES(NSTG)
665 FORMAT(F12.3,T25,' THE DESIGN POINT RADIUS ')
666 U(NSTG) = RDES(NSTG)*OMEGA
667 DHO(NSTG) = PSI(NSTG)*U(NSTG)*U(NSTG)
668 END IF
669 C
670 IF(RADTYPE.EQ.'B') THEN
671 WRITE(6,*) ' INPUT THE STAGE ACTUAL ENTHALPY CHANGE IN KJ/Kg. '
672 WRITE(6,*) ' THE ENTHALPY CHANGE IS ALWAYS TREATED AS POSITIVE.'
673 READ(5,*) DHO(NSTG)
674 WRITE(6,*) DHO(NSTG)
675 WRITE(10,117) DHO(NSTG)
676 FORMAT(F12.3,T25,' STAGE ENTHALPY CHANGE, KJ/Kg')
677 DHO(NSTG) = 1000.0*DHO(NSTG)
678 U(NSTG) = SQRT(DHO(NSTG)/PSI(NSTG))
679 RDES(NSTG) = U(NSTG)/OMEGA
680 END IF
681 C
682 IF(TURBO_TYP.EQ.'T') DHO(NSTG) = -DHO(NSTG)
683 C
684 C*****
685 C*****
686 C
687 WRITE(6,*) '*****'
688 WRITE(6,*) '*****'
689 WRITE(6,*)
690 WRITE(6,*) ' STAGE NUMBER. ', NSTG
691 WRITE(6,*) ' INPUT THE FIRST AND SECOND BLADE ROW AXIAL CHORDS IN
692 & METRES '
693 WRITE(6,*) ' THE CURRENT VALUES ARE ', AXCHRD1(NSTG),AXCHRD2(NSTG)
694 WRITE(6,*) ' Press A to accept these, or type in new values.'
695 READ(5,*,ERR=20,END=20) AXCHRD1(NSTG),AXCHRD2(NSTG)
696 WRITE(6,*) ' STATOR AND ROTOR AXIAL CHORDS ARE ',
697 & AXCHRD1(NSTG),AXCHRD2(NSTG)
698 WRITE(10,120) AXCHRD1(NSTG),AXCHRD2(NSTG)
699 FORMAT(2F12.3,T25,' BLADE AXIAL CHORDS IN METRES.')
700 WRITE(6,*)
701 WRITE(6,*) '*****'
702 WRITE(6,*) '*****'
703 C
704 C*****
705 C*****
706 C
707 WRITE(6,*) ' INPUT THE INTER ROW GAP AND INTER STAGE GAP AS FRACTI
708 &ONS OF THE FIRST ROW AXIAL CHORD '
709 WRITE(6,*) ' THE CURRENT VALUES ARE ', ROWGAP(NSTG),STAGEGAP(NSTG)

```

```

710      WRITE(6,*) ' Press A to accept these, or type in new values.'
711      READ(5,*,ERR=218,END=218) ROWGAP(NSTG), STAGEGAP(NSTG)
712      CONTINUE
713      WRITE(6,*) ' ROWGAP =', ROWGAP(NSTG), 'STAGEGAP = ', STAGEGAP(NSTG)
714      WRITE(10,119) ROWGAP(NSTG), STAGEGAP(NSTG)
715      FORMAT(2F12.3, T25, ' ROW GAP AND STAGE GAP ')
716      WRITE(6,*)
717      WRITE(6,*) ' *****'
718      WRITE(6,*) ' *****'
719  C
720  C*****
721  C*****
722  C JUMP TO HERE IF IFSAME ALL = "Y" .
723  C SET THE AXIAL POSITIONS OF THE BLADES FOR IN_TYPE = "AXI" .
724  C
725      CONTINUE
726  C
727      IF(NSTG.EQ.1) THEN
728          XMEAN(1) = -1.0*AXCHRD1(NSTG)
729          XMEAN(2) = -0.5*AXCHRD1(NSTG)
730          XMEAN(3) = 0.0
731          XMEAN(4) = XMEAN(3) + AXCHRD1(NSTG)
732          XMEAN(5) = XMEAN(4) + 0.5*ROWGAP(NSTG)*AXCHRD1(NSTG)
733          XMEAN(6) = XMEAN(4) + ROWGAP(NSTG)*AXCHRD1(NSTG)
734          XMEAN(7) = XMEAN(6) + AXCHRD2(NSTG)
735          XMEAN(8) = XMEAN(7) + 0.5*STAGEGAP(NSTG)*AXCHRD2(NSTG)
736          XMEAN(9) = XMEAN(7) + STAGEGAP(NSTG)*AXCHRD2(NSTG)
737          IF(NSTG.EQ.NSTAGES) XMEAN(8) = XMEAN(7) + 0.5*AXCHRD2(NSTG)
738          IF(NSTG.EQ.NSTAGES) XMEAN(9) = XMEAN(7) + AXCHRD2(NSTG)
739      ELSE
740  C
741  C SET THE LOCATION OF THE FIRST POINT IF THE INPUT TYPE HAS BEEN CHANGED FROM "MIX"
742  C TO "AXI"
743      XMEAN1 = XMEAN(7)
744      IF(IFSAME_ALL.EQ.'C') XMEAN1 = XMEANTE
745  C
746      XMEAN(1) = XMEAN1 + 0.333*STAGEGAP(NSTG)*AXCHRD1(NSTG)
747      XMEAN(2) = XMEAN1 + 0.667*STAGEGAP(NSTG)*AXCHRD1(NSTG)
748      XMEAN(3) = XMEAN1 + STAGEGAP(NSTG)*AXCHRD1(NSTG)
749      XMEAN(4) = XMEAN(3) + AXCHRD2(NSTG)
750      XMEAN(5) = XMEAN(4) + 0.5*ROWGAP(NSTG)*AXCHRD2(NSTG)
751      XMEAN(6) = XMEAN(4) + ROWGAP(NSTG)*AXCHRD2(NSTG)
752      XMEAN(7) = XMEAN(6) + AXCHRD2(NSTG)
753      XMEAN(8) = XMEAN(7) + 0.5*STAGEGAP(NSTG)*AXCHRD2(NSTG)
754      XMEAN(9) = XMEAN(8) + STAGEGAP(NSTG)*AXCHRD2(NSTG)
755      IF(NSTG.EQ.NSTAGES) XMEAN(8) = XMEAN(7) + 0.5*AXCHRD2(NSTG)
756      IF(NSTG.EQ.NSTAGES) XMEAN(9) = XMEAN(7) + AXCHRD2(NSTG)
757  C
758  C
759      DO N=1,9
760          RMEAN(N) = RDES(NSTG)
761          VMROT(N) = 1.0
762      END DO
763  C
764      NSS = 9
765      NLE1 = 3
766      NTE1 = 4
767      NLE2 = 6
768      NTE2 = 7
769      PHI_REF = PHI(NSTG)
770  C
771      WRITE(6,*)
772      WRITE(6,*) ' MEAN LINE COORDINATES, STAGE NUMBER', NSTG
773      WRITE(6,*) ' XMEAN ', (XMEAN(N), N=1, NSS)
774      WRITE(6,*) ' RMEAN ', (RMEAN(N), N=1, NSS)
775      WRITE(6,*) ' VMROT ', (VMROT(N), N=1, NSS)
776      WRITE(6,*)
777      WRITE(6,*) ' NLE1, NTE1, NLE2, NTE2 ', NLE1, NTE1, NLE2, NTE2
778      WRITE(6,*)
779  C
780  C
781  C*****

```

```

782 C*****
783 C    GO TO 3000 IF FLO_TYP = "AXI"
784 C
785     GO TO 3000
786 C
787 C*****
788 C*****
789 C    RE ENTER HERE IF FLO_TYP = "MIX"
790 C
791     CONTINUE
792 C
793 C*****
794 C*****
795 C*****
796 C*****
797 C    NOW INPUT DATA FOR FLO_TYP = "MIX" .
798 C
799 C*****
800 C*****
801     WRITE(6,*) ' FOR FLO_TYP= "MIX" YOU HAVE A CHOICE OF TWO INPUT METH
802 &ODS'
803     WRITE(6,*)
804     WRITE(6,*) ' EITHER INPUT ALL 4 BLADE ANGLES.'
805     WRITE(6,*)
806     WRITE(6,*) 'OR'
807     WRITE(6,*)
808     WRITE(6,*) 'INPUT THE ABSOLUTE FLOW ANGLES AT STAGE INLET AND EXIT'
809     WRITE(6,*) 'AND THE FLOW COEFFICIENT AND STAGE LOADING COEFFICIENT'
810     WRITE(6,*)
811     WRITE(6,*) ' INPUT  "A"  FOR THE FIRST METHOD,  "B" FOR THE SECOND'
812     READ(5,*) MIXTYP
813     WRITE(10,410) MIXTYP
814     FORMAT(A1,T25,' MIXTYP = INPUT TYPE FOR FLO_TYP = "MIX" .')
815     IF(MIXTYP.EQ.'a') MIXTYP = 'A'
816     IF(MIXTYP.EQ.'b') MIXTYP = 'B'
817 C
818 C*****
819 C*****
820 C
821     IF(MIXTYP.EQ.'A') THEN
822 C
823     WRITE(6,*) ' INPUT THE STATOR INLET AND EXIT FLOW ANGLES, IN DEG.'
824     READ(5,*) STATOR_IN, STATOR_OUT
825     WRITE(10,411) STATOR_IN, STATOR_OUT
826     FORMAT(2F10.3,T25,' ANGLES, STATOR_IN, STATOR_OUT')
827 C
828     WRITE(6,*)
829     WRITE(6,*) ' INPUT THE ROTOR INLET AND EXIT RELATIVE FLOW ANGLES,
830 & IN DEGREES.'
831     READ(5,*) ROTOR_IN, ROTOR_OUT
832     WRITE(10,412) ROTOR_IN, ROTOR_OUT
833     FORMAT(2F10.3,T25,' ANGLES, ROTOR_IN, ROTOR_OUT')
834 C
835     IF(TURBO_TYP.EQ.'C') THEN
836         BIN_ROW1(NSTG) = ROTOR_IN*DEGRAD
837         BOUT_ROW1(NSTG) = ROTOR_OUT*DEGRAD
838         BIN_ROW2(NSTG) = STATOR_IN*DEGRAD
839         BOUT_ROW2(NSTG) = STATOR_OUT*DEGRAD
840         PHI_STG1 = 1.0/(TAN(BIN_ROW2(NSTG))
841 & - TAN(BOUT_ROW1(NSTG)))
842     ELSE
843         BIN_ROW1(NSTG) = STATOR_IN*DEGRAD
844         BOUT_ROW1(NSTG) = STATOR_OUT*DEGRAD
845         BIN_ROW2(NSTG) = ROTOR_IN*DEGRAD
846         BOUT_ROW2(NSTG) = ROTOR_OUT*DEGRAD
847         PHI_STG1 = 1.0/(TAN(BOUT_ROW1(NSTG))
848 & - TAN(BIN_ROW2(NSTG)))
849     END IF
850 C
851     END OF MIXTYP = "A" LOOP.
852     END IF
853 C*****
854 C*****

```

```

855 C*****
856 C*****

857
858     IF(MIXTYP.EQ.'B') THEN
859 C
860     IF(NSTG.EQ.1) THEN
861     WRITE(6,*)
862     WRITE(6,*) ' INPUT THE FLOW COEFFICIENT AT THE FIRST ROTOR LEADING
863 & EDGE.'
864     READ(5,*)      PHI_REF
865     WRITE(10,82) PHI_REF
866     FORMAT(F10.4,T25,
867 & ' FLOW COEFFICIENT AT THE FIRST ROTOR LEADING EDGE.')
868     WRITE(6,*) ' FLOW COEFFICIENT = ', PHI_REF
869     WRITE(6,*)
870     END IF
871 C
872 C*****
873 C*****
874 C
875     WRITE(6,*) ' INPUT THE INLET AND EXIT ABSOLUTE FLOW ANGLES FOR THE
876 & WHOLE STAGE.'
877     READ(5,*)      ALPHA_IN(NSTG), ALPHA_OUT(NSTG)
878     WRITE(10,81)    ALPHA_IN(NSTG), ALPHA_OUT(NSTG)
879     FORMAT(2F10.3,T25,' STAGE INLET AND OUTLET ABSOLUTE FLOW ANGLES.')
880     WRITE(6,*) ' STAGE INLET & EXIT ANGLES',
881 &
882     ALPHA_IN(NSTG) = ALPHA_IN(NSTG)*DEGRAD
883     ALPHA_OUT(NSTG) = ALPHA_OUT(NSTG)*DEGRAD
884 C
885 C*****
886 C*****
887 C
888     WRITE(6,*)
889     WRITE(6,*) ' INPUT THE STAGE LOADING COEFFICIENT BASED ON THE BLADE
890 & SPEED AT THE ROTOR LEADING EDGE '
891     READ(5,*)      PSI(NSTG)
892     WRITE(10,83) PSI(NSTG)
893     FORMAT(F10.4,T25, ' STAGE LOADING COEFFICIENT AT THE ROTOR LEADING
894 & EDGE.')
895     WRITE(6,*) ' STAGE LOADING COEFFICIENT = ', PSI(NSTG)
896 C
897 C END OF MIXTYP = "B" LOOP.
898     END IF
899 C
900 C*****
901 C*****
902 C*****
903 C*****
904 C RE ENTER HERE IF FLO_TYP = "MIX" AND THE ANGLES, ETC,
905 C WERE THE SAME AS FOR THE LAST STAGE.
906 C
907     CONTINUE
908 C*****
909 C*****
910 C NOW INPUT THE STREAM SURFACE COORDINATES. FOR FLO_TYP = "MIX" .
911 C
912     WRITE(6,*) '*****'
913     WRITE(6,*) '*****'
914     WRITE(6,*) ' THE STREAM SURFACE COORDINATES AND MERIDIONAL VELOCITY
915 & RATIOS MUST NOW BE INPUT '
916 C
917     WRITE(6,*)
918     WRITE(6,*) ' THE NEW VALUES MUST FORM A SMOOTH CONTINUATION OF
919 & THE LAST STREAM SURFACE '
920     WRITE(6,*)
921     WRITE(6,*) '*****'
922     WRITE(6,*) '*****'
923 C
924     WRITE(6,*)
925     WRITE(6,*) 'INPUT THE NUMBER OF POINTS ON THE MEAN STREAM SURFACE'
926     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 ? '

```

```

1000      WRITE(6,*) ' ANSWER "Y" or "N" '
1001      READ(5,*) ANSSS
1002      WRITE(6,*) ' ANSSS =', ANSSS
1003  C
1004      IF(ANSSS.EQ.'Y'.OR.ANSSS.EQ.'y') GO TO 700
1005  C
1006      WRITE(10,128) NSS
1007      WRITE(10,*) 'THE FOLLOWING LINE OF DATA CONTAINS THE STREAM SURFACE
1008 & AXIAL COORDINATES.'
1009      WRITE(10,129) (XMEAN(NP),NP=1,NSS )
1010      WRITE(10,*) 'THE FOLLOWING LINE OF DATA CONTAINS THE STREAM SURFACE
1011 & RADIAL COORDINATES.'
1012      WRITE(10,129) (RMEAN(NP),NP=1,NSS )
1013      WRITE(10,*) 'THE FOLLOWING LINE OF DATA CONTAINS THE MERIDIONAL VEL
1014 & OCITY RATIOS.'
1015      WRITE(10,129) (VMRAT(NP),NP=1,NSS )
1016      WRITE(10,130) NLE1,NTE1,NLE2,NTE2
1017      WRITE(10,138) ANSSS
1018      FORMAT(A1,T25, ' DO YOU WANT TO CHANGE THE STREAM SURFACE COORDINA
1019 & TES ?')
1020 128 FORMAT(I5,T25, ' NUMBER OF POINTS ON THE STREAM SURFACE.')
1021      FORMAT(8F10.4)
1022      FORMAT(4I5,T25, ' LEADING AND TRAILING EDGE POINTS ON THE MEAN STRE
1023 & AM SURFACE.')
1024  C
1025  C*****
1026  C*****
1027  C*****
1028  C*****
1029  C RE ENTER HERE IF FLO_TYP = "AXI"
1030  C
1031      CONTINUE
1032  C
1033  C*****
1034  C*****
1035  C FOR BOTH VALUES OF "FLO_TYP"
1036  C CALCULATE THE STREAM SURFACE DISTANCES.
1037  C
1038      SDIST(1) = 0.0
1039      DO NP = 2,NSS
1040          XDIF = XMEAN(NP) - XMEAN(NP-1)
1041          RDIF = RMEAN(NP) - RMEAN(NP-1)
1042          SDIF = SQRT(XDIF*XDIF + RDIF*RDIF)
1043          IF(SDIF.LT.1.001*ABS(RDIF)) SDIF = 1.001*ABS(RDIF)
1044          SDIST(NP) = SDIST(NP-1) + SDIF
1045      END DO
1046  C
1047  C*****
1048  C*****
1049  C
1050      WRITE(6,*) 'INPUT THE BLOCKAGE FACTOR AT THE LEADING EDGES OF THE
1051 & FIRST BLADE ROW AND AT THE TRAILING EDGE OF THE'
1052      WRITE(6,*) 'SECOND BLADE ROW.'
1053      WRITE(6,*) 'THIS IS THE SUM OF THE DISPLACEMENT THICKNESSES OF THE
1054 & HUB AND CASING BOUNDARY LAYERS DIVIDED BY THE BLADE SPAN. '
1055      WRITE(6,*) 'THE CURRENT VALUES ARE ', FBLOCK_LE(NSTG),
1056 & FBLOCK_TE(NSTG)
1057      WRITE(6,*) 'Press "A" to accept these, or type in new values.'
1058      READ(5,*, ERR= 219,END=219) FBLOCK_LE(NSTG), FBLOCK_TE(NSTG)
1059      CONTINUE
1060      WRITE(6,*) ' FBLOCK_LE = ', FBLOCK_LE(NSTG), ' FBLOCK_TE = ',
1061 & FBLOCK_TE(NSTG)
1062      WRITE(10,220) FBLOCK_LE(NSTG), FBLOCK_TE(NSTG)
1063      FORMAT(2F10.5,T25, ' BLOCKAGE FACTORS, FBLOCK_LE, FBLOCK_TE ')
1064      WRITE(6,*)
1065      WRITE(6,*) '*****'
1066      WRITE(6,*) '*****'
1067  C
1068  C*****
1069  C*****
1070  C
1071  C SET THE BLOCKAGE FACTORS
1072  C MAKE THE BLOCKAGE VARY LINEARLY WITH STREAM SURFACE DISTANCE.

```



```

1073         DBLOCK_DS = (FBLOCK_TE(NSTG)-FBLOCK_LE(NSTG))
1074         & / (SDIST(NTE2) - SDIST(NLE1))
1075         DO N = 1,NSS
1076             FBLOCK(N) = FBLOCK_LE(NSTG)+DBLOCK_DS*(SDIST(N)-SDIST(NLE1))
1077         END DO
1078 C
1079         WRITE(6,*)
1080         WRITE(6,*) ' BLOCKAGE FACTOR THROUGH THE STAGE '
1081         WRITE(6,*) (FBLOCK(N),N=1,NSS)
1082         WRITE(6,*)
1083 C
1084 C CALCULATE THE MERIDIONAL PITCH ANGLES.
1085         DO NP = 2,NSS-1
1086             PITCH_ANGL(NP) = ASIN( (RMEAN(NP+1) - RMEAN(NP-1))
1087             & / (SDIST(NP+1)-SDIST(NP-1)) )
1088         END DO
1089         PITCH_ANGL(1) = ASIN( (RMEAN(2)-RMEAN(1))/(SDIST(2)-SDIST(1)) )
1090         PITCH_ANGL(NSS) = ASIN( (RMEAN(NSS)-RMEAN(NSS-1))
1091         & / (SDIST(NSS)-SDIST(NSS-1)) )
1092 C
1093 C
1094         WRITE(6,*) 'STREAM SURFACE COORDINATES AND SLOPE BEFORE SMOOTHING'
1095         WRITE(6,*) ' XMEAN RMEAN SDIST PITCH ANGLE '
1096         DO NP=1,NSS
1097             WRITE(6,131) XMEAN(NP),RMEAN(NP),SDIST(NP),PITCH_ANGL(NP)*RADDEG
1098         END DO
1099         WRITE(6,*)
1100 C
1101 C SMOOTH THE MEAN STREAM SURFACE COORDINATES AND PITCH ANGLE.
1102 C CHANGED TO USE SMOOTH2 29/09/2017
1103         CALL SMOOTH2(1,NSS,NSMOOTH,SFAC,XMEAN,RMEAN)
1104 C         CALL SMOOTH(1, NSS, 4, 0.25, SDIST, XMEAN)
1105 C         CALL SMOOTH(1, NSS, 4, 0.25, SDIST, RMEAN)
1106         CALL SMOOTH(1, NSS, 4, 0.25, SDIST, PITCH_ANGL)
1107 C
1108         WRITE(6,*)
1109         WRITE(6,*) ' STREAM SURFACE COORDINATES AND SLOPE AFTER SMOOTHING'
1110         WRITE(6,*) ' XMEAN RMEAN SDIST PITCH ANGLE '
1111         DO NP=1,NSS
1112             WRITE(6,131) XMEAN(NP),RMEAN(NP),SDIST(NP),PITCH_ANGL(NP)*RADDEG
1113             FORMAT(4F12.4)
1114         END DO
1115 C
1116 C*****
1117 C*****
1118 C*****
1119 C*****
1120 C SET THE REFERENCE MERIDIONAL VELOCITY
1121 C
1122         IF(MIXTYP.EQ.'A') THEN
1123 C
1124             IF(NSTG.EQ.1.AND.TURBO_TYP.EQ.'C') THEN
1125                 PHI_REF = PHI_STG1*VMRAT(NLE1)/VMRAT(NLE2)*RMEAN(NLE2)/RMEAN(NLE1)
1126             ELSE
1127                 PHI_REF = PHI_STG1
1128             END IF
1129 C
1130         END IF
1131 C*****
1132 C*****
1133 C SET THE REFERENCE MERIDIONAL VELOCITY.
1134         IF(NSTG.EQ.1.AND.TURBO_TYP.EQ.'C')
1135         & VM_REF = PHI_REF*RMEAN(NLE1)*OMEGA
1136         IF(NSTG.EQ.1.AND.TURBO_TYP.EQ.'T')
1137         & VM_REF = PHI_REF*RMEAN(NLE2)*OMEGA
1138 C*****
1139 C*****
1140 C SET THE LOCAL MERIDIONAL VELOCITY AND BLADE SPEED
1141 C
1142         WRITE(6,*)
1143         WRITE(6,*) ' NS U_LOCAL VM_LOCAL PHI_LOCAL '
1144         DO NS = 1,NSS
1145             VM_LOC(NS) = VM_REF*VMRAT(NS)

```

```

1146     U_LOC(NS)      = 0.0
1147     IF(TURBO_TYP.EQ.'C'.AND.NS.GE.NLE1.AND.NS.LE.NTE1)
1148 &     U_LOC(NS)= RMEAN(NS)*OMEGA
1149     IF(TURBO_TYP.EQ.'T'.AND.NS.GE.NLE2.AND.NS.LE.NTE2)
1150 &     U_LOC(NS) = RMEAN(NS)*OMEGA
1151     PHI_LOC(NS)     = VM_LOC(NS)/(RMEAN(NS)*OMEGA)
1152     WRITE(6,132) NS, U_LOC(NS),VM_LOC(NS),PHI_LOC(NS)
1153     FORMAT(I5,3F12.4)
1154     END DO
1155     WRITE(6,*)
1156 C
1157 C     SET THE ABSOLUTE TANGENTIAL VELOCITIES. THIS IS NOT CORRECT FOR MIXTYP = "B".
1158     IF(MIXTYP.EQ.'B') GO TO 133
1159     WRITE(6,*) ' SETTING TANGENTIAL VELs, STAGE', NSTG
1160     VT_LE1 = VM_LOC(NLE1)*TAN(BIN_ROW1(NSTG)) + U_LOC(NLE1)
1161     VT_TE1 = VM_LOC(NTE1)*TAN(BOU_ROW1(NSTG)) + U_LOC(NTE1)
1162     VT_LE2 = VM_LOC(NLE2)*TAN(BIN_ROW2(NSTG)) + U_LOC(NLE2)
1163     VT_TE2 = VM_LOC(NTE2)*TAN(BOU_ROW2(NSTG)) + U_LOC(NTE2)
1164     VT_IN  = VT_LE1*RMEAN(NLE1)/RMEAN(1)
1165     VT_OUT = VT_TE2*RMEAN(NTE2)/RMEAN(NSS)
1166     CONTINUE
1167 C     SET THE ABSOLUTE TANGENTIAL VELOCITIES FOR MIXTYP = "B" .
1168     IF(MIXTYP.EQ.'B') THEN
1169         VT_LE1 = VM_LOC(NLE1)*TAN(ALPHA_IN(NSTG))
1170         VT_IN  = VT_LE1*RMEAN(NLE1)/RMEAN(1)
1171         VT_TE2 = VM_LOC(NTE2)*TAN(ALPHA_OUT(NSTG))
1172         VT_OUT = VT_TE2*RMEAN(NTE2)/RMEAN(NSS)
1173     END IF
1174 C
1175 C*****
1176 C*****
1177     IF(MIXTYP.EQ.'A') THEN
1178 C
1179     IF(TURBO_TYP.EQ.'C') THEN
1180         ALPHA_IN(NSTG) = ATAN(1/PHI_LOC(NLE1) + TAN(BIN_ROW1(NSTG)))
1181         ALPHA_OUT(NSTG) = BOU_ROW2(NSTG)
1182         PSI(NSTG) = (VT_TE1 - VT_LE1)/U_LOC(NLE1)
1183     ELSE
1184         ALPHA_IN(NSTG) = BIN_ROW1(NSTG)
1185         ALPHA_OUT(NSTG) = ATAN(1./PHI_LOC(NTE2) + TAN(BOU_ROW2(NSTG)))
1186         PSI(NSTG) = (VT_LE2 - VT_TE2)/U_LOC(NLE2)
1187     END IF
1188 C
1189 C     END OF MIXTYP = "A" LOOP.
1190     END IF
1191 C
1192 C*****
1193 C*****
1194 C
1195     IF(FLO_TYP.EQ.'AXI') GO TO 5500
1196 C
1197 C*****
1198 C*****
1199 C     CALCULATE THE BLADE ANGLES FOR FLO_TYP = "MIX"
1200 C
1201     IF(TURBO_TYP.EQ.'C') THEN
1202         RDES(NSTG) = RMEAN(NLE1)
1203         U(NSTG) = U_LOC(NLE1)
1204         VMER(NSTG) = VM_LOC(NLE1)
1205         DHO(NSTG) = PSI(NSTG)*U_LOC(NLE1)*U_LOC(NLE1)
1206         DRVT = DHO(NSTG)/OMEGA
1207         VT_TE1 = (RMEAN(NLE1)*VT_LE1 + DRVT)/RMEAN(NTE1)
1208         VT_LE2 = VT_TE1*RMEAN(NTE1)/RMEAN(NLE2)
1209         IF(MIXTYP.EQ.'B') THEN
1210             BOU_ROW1(NSTG) = ATAN((VT_TE1 - U_LOC(NTE1))/VM_LOC(NTE1))
1211             BIN_ROW2(NSTG) = ATAN(VT_LE2/VM_LOC(NLE2))
1212             BIN_ROW1(NSTG) = ATAN((VT_LE1 - U_LOC(NLE1))/VM_LOC(NLE1))
1213             BOU_ROW2(NSTG) = ATAN(VT_TE2/VM_LOC(NTE2))
1214         END IF
1215     END IF
1216 C
1217 C
1218     IF(TURBO_TYP.EQ.'T') THEN

```

```

1219      RDES(NSTG)      = RMEAN(NLE2)
1220      U(NSTG)         = U_LOC(NLE2)
1221      VMER(NSTG)      = VM_LOC(NLE2)
1222      DHO(NSTG)       = -PSI(NSTG)*U_LOC(NLE2)*U_LOC(NLE2)
1223      DRVT            = DHO(NSTG)/OMEGA
1224      VT_LE2          = (VT_TE2*RMEAN(NTE2) - DRVT)/RMEAN(NLE2)
1225      VT_TE1          = VT_LE2*RMEAN(NLE2)/RMEAN(NTE1)
1226      IF(MIXTYP.EQ.'B') THEN
1227          BOUT_ROW1(NSTG) = ATAN(VT_TE1/VM_LOC(NTE1))
1228          BIN_ROW2(NSTG)  = ATAN((VT_LE2 - U_LOC(NLE2))/VM_LOC(NLE2))
1229          BIN_ROW1(NSTG)  = ATAN(VT_LE1/VM_LOC(NLE1))
1230          BOUT_ROW2(NSTG) = ATAN((VT_TE2 - U_LOC(NTE2))/VM_LOC(NTE2))
1231      END IF
1232  END IF
1233  C
1234  C*****
1235  C*****
1236  C      RE ENTER HERE IF FLO_TYPE = "AXI"
1237  C
1238      CONTINUE
1239  C
1240  C*****
1241  C*****
1242  C
1243  C      NEXT SECTION FOR BOTH TYPES OF INPUT I.E. BOTH "FLO_TYP" = "MIX" and = "AXI"
1244  C
1245  C      STORE THE STREAM SURFACE COORDINATES AND VELOCITY RATIO.
1246  C
1247      NSS_STG(NSTG) = NSS
1248      NLE1_STG(NSTG) = NLE1
1249      NTE1_STG(NSTG) = NTE1
1250      NLE2_STG(NSTG) = NLE2
1251      NTE2_STG(NSTG) = NTE2
1252  C
1253      DO NS = 1, NSS
1254          XMEANALL(NSTG, NS) = XMEAN(NS)
1255          RMEANALL(NSTG, NS) = RMEAN(NS)
1256          VMLOCALL(NSTG, NS) = VM_LOC(NS)
1257      END DO
1258  C
1259  C*****
1260  C*****
1261  C
1262  C      CALL PROPS FOR STEAM CONDITIONS AT STAGE INLET '
1263  C
1264      CALL PROPS(1, 1, HO(NSTG), S(NSTG), PO(NSTG), T(NSTG), RHO(NSTG),
1265      &          WET(NSTG), VIN(1), G(NSTG), VS(NSTG), 1, IPROPS, IWET)
1266  C
1267  C      WRITE OUTPUT TO SCREEN
1268  C
1269      WRITE(6,*) ' STAGE INLET STAGNATION ENTHALPY = ', HO(NSTG)
1270      WRITE(6,*) ' STAGE INLET ENTROPY = ', S(NSTG)
1271      WRITE(6,*) ' STAGE INLET PRESSURE = ', PO(NSTG), ' BAR.'
1272      WRITE(6,*) ' STAGE INLET TEMPERATURE = ', T(NSTG), ' K.'
1273      WRITE(6,*) ' STAGE INLET DENSITY = ', RHO(NSTG),
1274      &          ' Kg/M3.'
1275      WRITE(6,*)
1276      WRITE(6,*) ' ROTATIONAL SPEED = ', RPM
1277      WRITE(6,*) ' BLADE SPEED = ', U(NSTG), ' Metres/sec.'
1278      WRITE(6,*) ' DESIGN RADIUS = ', RDES(NSTG), ' Metres.'
1279      WRITE(6,*) ' STAGE STAGNATION ENTHALPY CHANGE = ',
1280      &          DHO(NSTG)/1000., ' KJ/Kg.'
1281      WRITE(6,*)
1282      ROIN = RHO(1)
1283  C
1284  C
1285  C*****
1286  C*****
1287  C*****
1288  C
1289  C      NOW INPUT THE DETAILS OF THE STAGE
1290  C
1291      IF(IFSAME_ALL.EQ.'Y') GO TO 666

```

```

1292 C
1293 WRITE(6,*)
1294 WRITE(6,*) ' STARTING INPUT FOR STAGE NUMBER ', NSTG
1295 WRITE(6,*)
1296 WRITE(6,*)
1297 WRITE(6,*) '*****'
1298 WRITE(6,*) '*****'
1299 WRITE(6,*)
1300 CONTINUE
1301 WRITE(6,*) ' STAGE NUMBER. ', NSTG
1302 WRITE(6,*) ' INPUT A GUESS OF THE STAGE EFFICIENCY'
1303 WRITE(6,*) ' THE CURRENT VALUE IS ', ETA(NSTG)
1304 WRITE(6,*) ' Press A to accept this, or type in a new value.'
1305 READ(5,*,ERR=13,END=13) ETA(NSTG)
1306 CONTINUE
1307 WRITE(6,*) ' THE GUESSED STAGE EFFICIENCY IS ', ETA(NSTG)
1308 IF(ETA(NSTG).GT.1.0) THEN
1309 WRITE(6,*) ' VALUE MUST BE A DECIMAL NOT A PERCENTAGE, e.g. 0.8.'
1310 GO TO 1113
1311 END IF
1312 WRITE(10,118) ETA(NSTG)
1313 FORMAT(F12.3,T25, ' GUESS OF THE STAGE ISENTROPIC EFFICIENCY')
1314 C
1315 IF(TURBO_TYP.EQ.'T') DHOIS(NSTG) = DHO(NSTG)/ETA(NSTG)
1316 IF(TURBO_TYP.EQ.'C') DHOIS(NSTG) = DHO(NSTG)*ETA(NSTG)
1317 C
1318 WRITE(6,*) '*****'
1319 WRITE(6,*) '*****'
1320 C
1321 C*****
1322 C*****
1323 C
1324 WRITE(6,*) '*****'
1325 WRITE(6,*) '*****'
1326 WRITE(6,*)
1327 WRITE(6,*) ' INPUT A GUESS OF THE DEVIATION ANGLES FOR THE FIRST'
1328 WRITE(6,*) ' AND SECOND BLADE ROWS '
1329 WRITE(6,*) ' THIS IS THE DIFFERENCE BETWEEN THE FLOW ANGLE AND'
1330 WRITE(6,*) ' THE METAL ANGLE AT THE TRAILING EDGE'
1331 WRITE(6,*) ' THE DEVIATION ANGLES INPUT SHOULD ALWAYS BE POSITIVE'
1332 WRITE(6,*) ' THE CURRENT VALUES ARE', DEVN_1,DEVN_2,'DEGREES'
1333 WRITE(6,*) ' Press A to accept these, or type in new values'
1334 READ(5,*,ERR=21,END=21) DEVN_1, DEVN_2
1335 CONTINUE
1336 WRITE(6,*) ' THE DEVIATION ANGLES ARE ', DEVN_1, DEVN_2
1337 C
1338 WRITE(10,121) DEVN_1, DEVN_2
1339 FORMAT(2F8.3,T25, ' ESTIMATE OF THE FIRST AND SECOND ROW DEVIATION
1340 &ANGLES')
1341 DEVN1(NSTG) = DEVN_1
1342 DEVN2(NSTG) = DEVN_2
1343 WRITE(6,*)
1344 WRITE(6,*) '*****'
1345 WRITE(6,*) '*****'
1346 WRITE(6,*)
1347 C
1348 C*****
1349 C*****
1350 C
1351 WRITE(6,*) '*****'
1352 WRITE(6,*) '*****'
1353 WRITE(6,*)
1354 WRITE(6,*) ' INPUT THE REQUIRED INCIDENCE ANGLES FOR THE FIRST'
1355 WRITE(6,*) ' AND SECOND BLADE ROWS '
1356 WRITE(6,*) ' THIS IS THE DIFFERENCE BETWEEN THE FLOW ANGLE AND'
1357 WRITE(6,*) ' THE METAL ANGLE AT THE LEADING EDGE'
1358 WRITE(6,*) ' IT CAN BE EITHER POSITIVE OR NEGATIVE'
1359 WRITE(6,*) ' THE CURRENT VALUES ARE', AINC_1, AINC_2,'DEGREES'
1360 WRITE(6,*) ' Press A to accept these, or type in new values'
1361 READ(5,*,ERR=22,END=22) AINC_1, AINC_2
1362 CONTINUE
1363 WRITE(6,*) ' THE INCIDENCE ANGLES ARE ', AINC_1, AINC_2
1364 C

```

```

1365         WRITE(10,222) AINC_1, AINC_2
1366     FORMAT(2F8.3,T25,' FIRST AND SECOND ROW INCIDENCE ANGLES')
1367     AINC1(NSTG) = AINC_1
1368     AINC2(NSTG) = AINC_2
1369     WRITE(6,*)
1370     WRITE(6,*) '*****'
1371     WRITE(6,*) '*****'
1372     WRITE(6,*)
1373 C
1374 C*****
1375 C*****
1376 C
1377     IF(IFSAME_ALL.EQ.'Y') GO TO 774
1378 C
1379 C*****
1380 C*****
1381 C     NEXT FOR BOTH VALUES OF "FLO_TYP", CHOOSE THE BLADE TWIST. THE TWIST CAN BE ANY
1382 C     MULTIPLE OF THAT REQUIRED BY A FREE-VORTEX DESIGN.
1383 C
1384     WRITE(6,*)
1385     WRITE(6,*) ' INPUT "FRAC_TWIST", THE FRACTION OF FREE-VORTEX TWIST
1386 & THAT YOU WANT TO USE ON THIS STAGE.'
1387     WRITE(6,*) ' FRAC_TWIST = 1.0 GIVES FULL FREE-VORTEX TWIST. '
1388     WRITE(6,*) ' FRAC_TWIST = 0.0 GIVES NO TWIST, SO THE BLADE ANGLES
1389 & ARE THE SAME AT ALL SPANWISE POSITIONS.'
1390     WRITE(6,*) 'VALUES OF FRAC_TWIST GREATER THAN 1.0 OR LESS THAN 0.0
1391 & CAN ALSO BE USED.'
1392     IF(NSTG.EQ.1) FRAC_TWIST = 1.0
1393     WRITE(6,*) ' THE CURRENT VALUE IS ', FRAC_TWIST
1394     WRITE(6,*) ' Press A to accept this, or type in a new value.'
1395     READ(5,*,ERR=140) FRAC_TWIST
1396     CONTINUE
1397 C
1398     WRITE(10,141) FRAC_TWIST
1399     FORMAT(F10.5,T25,' BLADE TWIST OPTION, FRAC_TWIST')
1400 C
1401 C*****
1402 C*****
1403 C     NEXT INPUT THE OPTION TO ROTATE THE BLADE SECTIONS. EACH SECTION GENERATED
1404 C     CAN BE ROTATED BY A DIFFERNT ANGLE.
1405 C
1406     IF_ROT = 'N'
1407     WRITE(6,*) ' DO YOU WISH TO ROTATE THE SECTIONS GENERATED BY ANGLES
1408 & TO BE INPUT LATER ? '
1409     WRITE(6,*) ' THE ROTATION CAN BE DIFFERENT FOR EACH BLADE SECTION'
1410     WRITE(6,*) ' INPUT Y or N '
1411     READ(5,*,ERR=143) IF_ROT
1412     CONTINUE
1413     IF(IF_ROT.EQ.'y') IF_ROT = 'Y'
1414     WRITE(10,144) IF_ROT
1415     FORMAT(A1,T25,' BLADE ROTATION OPTION , Y or N' )
1416 C
1417 C*****
1418 C*****
1419 C
1420     CONTINUE
1421 C
1422 C*****
1423 C*****
1424 C
1425     WRITE(6,*)
1426     WRITE(6,*) '*****'
1427     WRITE(6,*) '*****'
1428     WRITE(6,*)
1429     WRITE(6,*)
1430     WRITE(6,*) ' INPUT THE QO LINE ANGLES, MEASURED FROM THE AXIAL DIR
1431 & SECTION AT THE LE AND TE OF THE FIRST BLADE ROW .'
1432     WRITE(6,*) ' THE CURRENT VALUES ARE', QLE_ROW1(NSTG), QTE_ROW1(NSTG)
1433     WRITE(6,*) ' Press A to accept these, or type in new values.'
1434     READ(5,*,ERR=775,END=775) QLE_ROW1(NSTG), QTE_ROW1(NSTG)
1435     CONTINUE
1436     WRITE(6,*) ' QLE_ROW1, QTE_ROW1 = ', QLE_ROW1(NSTG), QTE_ROW1(NSTG)
1437     WRITE(10,776) QLE_ROW1(NSTG), QTE_ROW1(NSTG)

```

```

1438     FORMAT(2F8.3,T25,' QO ANGLES AT LE AND TE OF ROW 1 ')
1439     WRITE(6,*)
1440     WRITE(6,*) '*****'
1441     WRITE(6,*) '*****'
1442     WRITE(6,*)
1443 C
1444     WRITE(6,*)
1445     WRITE(6,*) '*****'
1446     WRITE(6,*) '*****'
1447     WRITE(6,*)
1448     WRITE(6,*)
1449     WRITE(6,*) ' INPUT THE QO LINE ANGLES, MEASURED FROM THE AXIAL DIR
1450 &SECTION AT THE LE AND TE OF THE SECOND BLADE ROW .'
1451     WRITE(6,*) ' THE CURRENT VALUES ARE',QLE_ROW2(NSTG),QTE_ROW2(NSTG)
1452     WRITE(6,*) ' Press A to accept these, or type in new values.'
1453     READ(5,*,ERR=777,END=777) QLE_ROW2(NSTG), QTE_ROW2(NSTG)
1454     CONTINUE
1455     WRITE(6,*) ' QLE_ROW2, QTE_ROW2 =', QLE_ROW2(NSTG), QTE_ROW2(NSTG)
1456     WRITE(10,779) QLE_ROW2(NSTG), QTE_ROW2(NSTG)
1457     FORMAT(2F8.3,T25,' QO ANGLES AT LE AND TE OF ROW 2 ')
1458     WRITE(6,*)
1459     WRITE(6,*) '*****'
1460     WRITE(6,*) '*****'
1461     WRITE(6,*)
1462 C
1463 C*****
1464 C*****
1465 C RETURN TO HERE IF IFSAME_ALL = "Y" SO NOT CHANGING THE FLOW ANGLES, ETC.
1466 C
1467     CONTINUE
1468 C
1469 C*****
1470 C*****
1471 C
1472     VX(NSTG) = VM_LOC(NLE1)
1473     IF(TURBO_TYP.EQ.'C') PHI(NSTG) = VM_LOC(NLE1)/U_LOC(NLE1)
1474     IF(TURBO_TYP.EQ.'T') PHI(NSTG) = VM_LOC(NLE2)/U_LOC(NLE2)
1475 C
1476     DHOTOTAL = DHOTOTAL + DHOIS(NSTG)
1477 C
1478 C*****
1479 C*****
1480 C*****
1481 C
1482     TEXITT = (HO(NSTG) + DHO(NSTG))/CPGAS
1483     IF(TURBO_TYP.EQ.'T') SEXIT(NSTG) = S(NSTG) -
1484 & (1. - ETA(NSTG))*DHOIS(NSTG)/TEXITT
1485     IF(TURBO_TYP.EQ.'C') SEXIT(NSTG) = S(NSTG) +
1486 & (1. - ETA(NSTG))*DHO(NSTG)/TEXITT
1487 C
1488 C CALCULATE PROPERTIES AT STAGE EXIT
1489 C
1490     DO 155 NS = NTE2,NSS
1491         VTLOC(NS) = VT_TE2*RMEAN(NTE2)/RMEAN(NS)
1492         WTLOC = VTLOC(NS) - U_LOC(NS)
1493         VMLOC = VM_LOC(NS)
1494         VABS(NS) = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
1495         VREL(NS) = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
1496         HOLOC(NS) = HO(NSTG) + DHO(NSTG)
1497         SLOC(NS) = SEXIT(NSTG)
1498     CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),
1499 & RHOLOC(NS),WET(NS),VABS(NS),G(NS),
1500 & VS(NS),1,IPROPS,IWET)
1501     HSTATIC = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1502     HOREL = HSTATIC + 0.5*VREL(NS)*VREL(NS)
1503     POREL(NS) = PLOC(NS)*(HOREL/HSTATIC)**FGA
1504     POABS(NS) = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
1505     MACH_REL(NS) = VREL(NS)/VS(NS)
1506     MACH_ABS(NS) = VABS(NS)/VS(NS)
1507 C
1508     IF(NS.EQ.NTE2) THEN
1509         VABSEX(NSTG) = VABS(NS)
1510         VRELEX(NSTG) = VREL(NS)

```

```

1511         HOEXIT(NSTG) = HOLOC(NS)
1512         HEXIT(NSTG) = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1513         RHOEXIT(NSTG) = RHOLOC(NS)
1514         PEXIT(NSTG) = PLOC(NS)
1515         TEXTIT(NSTG) = TLOC(NS)
1516         TO_EXIT = HOLOC(NS)/CPGAS
1517         PO_EXIT = PEXIT(NSTG)
1518         & * (HOEXIT(NSTG)/HEXIT(NSTG))**FGA
1519         VXOUT(NSTG) = VMLOC
1520     END IF
1521 C
1522     CONTINUE
1523 C
1524 C CALCULATE PROPERTIES AT STAGE INLET
1525 C
1526     DO 153 NS = 1,NLE1
1527         VTLOC(NS) = VT_IN*RMEAN(1)/RMEAN(NS)
1528         WTLOC = VTLOC(NS) - U_LOC(NS)
1529         VMLOC = VM_LOC(NS)
1530         VABS(NS) = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
1531         VREL(NS) = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
1532         HOLOC(NS) = HO(NSTG)
1533         SLOC(NS) = S(NSTG)
1534         CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
1535 & WET(NS),VABS(NS),G(NS),VS(NS),1,IPROPS,IWET)
1536         HSTATIC = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1537         HOREL = HSTATIC + 0.5*VREL(NS)*VREL(NS)
1538         POREL(NS) = PLOC(NS)*(HOREL/HSTATIC)**FGA
1539         POABS(NS) = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
1540         MACH_REL(NS) = VREL(NS)/VS(NS)
1541         MACH_ABS(NS) = VABS(NS)/VS(NS)
1542 C
1543         IF(NS.EQ.NLE1) THEN
1544             VABSIN(NSTG) = VABS(NS)
1545             VRELIN(NSTG) = VREL(NS)
1546             HOINLET(NSTG) = HO(NSTG)
1547             HINLET(NSTG) = HO(NSTG) - 0.5*VABS(NS)*VABS(NS)
1548             RHOINLET(NSTG) = RHOLOC(NS)
1549             PINLET(NSTG) = PLOC(NS)
1550             TINLET(NSTG) = TLOC(NS)
1551             SINLET(NSTG) = SLOC(NS)
1552             VXIN(NSTG) = VMLOC
1553             FGA = G(NS)/(G(NS) - 1.0)
1554             PO_INLET = PINLET(NSTG)
1555             & * (HOINLET(NSTG)/HINLET(NSTG))**FGA
1556         END IF
1557         IF(NS.EQ.1.AND.NSTG.EQ.1) VM_INLET = VMLOC
1558     CONTINUE
1559 C
1560 C CALCULATE PROPERTIES WITHIN THE FIRST BLADE ROW
1561 C
1562     DO 154 NS = NLE1+1,NTE1
1563         FRAC = (SDIST(NS) - SDIST(NLE1))/(SDIST(NTE1) - SDIST(NLE1))
1564         VTLOC(NS) = VT_LE1 + FRAC*(VT_TE1 - VT_LE1)
1565         WTLOC = VTLOC(NS) - U_LOC(NS)
1566         VMLOC = VM_LOC(NS)
1567         VABS(NS) = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
1568         VREL(NS) = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
1569         HOLOC(NS) = HO(NSTG) + U_LOC(NS)*(VTLOC(NS) - VT_LE1)
1570         SLOC(NS) = SINLET(NSTG) + 0.5*FRAC*(SEXIT(NSTG) - S(NSTG))
1571         CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
1572 & WET(NS),VABS(NS),G(NS),VS(NS),1,IPROPS,IWET)
1573         HSTATIC = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1574         HOREL = HSTATIC + 0.5*VREL(NS)*VREL(NS)
1575         POREL(NS) = PLOC(NS)*(HOREL/HSTATIC)**FGA
1576         POABS(NS) = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
1577         MACH_REL(NS) = VREL(NS)/VS(NS)
1578         MACH_ABS(NS) = VABS(NS)/VS(NS)
1579 C
1580         IF(NS.EQ.NTE1) THEN
1581             VABSMID(NSTG) = VABS(NS)
1582             VRELMID(NSTG) = VREL(NS)
1583             HOMID(NSTG) = HOLOC(NS)

```



```

1584             HMID(NSTG)      = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1585             RHOMID(NSTG)     = RHOLOC(NS)
1586             PMID(NSTG)       = PLOC(NS)
1587             TMID(NSTG)        = TLOC(NS)
1588             SMID(NSTG)        = SLOC(NS)
1589             TO_MID            = HOMID(NSTG)/CPGAS
1590
1591     END IF
1592     CONTINUE
1593
1594 C
1595 C     CALCULATE PROPERTIES IN THE GAP BETWEEN ROWS
1596 C
1597 DO 157 NS = NTE1+1,NLE2
1598     VTLOC(NS) = VT_TE1*RMEAN(NTE1)/RMEAN(NS)
1599     WTLOC      = VTLOC(NS) - U_LOC(NS)
1600     VMLOC      = VM_LOC(NS)
1601     VABS(NS)   = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
1602     VREL(NS)   = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
1603     HOLOC(NS)  = HOMID(NSTG)
1604     SLOC(NS)   = SMID(NSTG)
1605     CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
1606 &             WET(NS),VABS(NS),G(NS),VS(NS),1,IProps,IWET)
1607     HSTATIC    = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1608     HOREL      = HSTATIC + 0.5*VREL(NS)*VREL(NS)
1609     POREL(NS)  = PLOC(NS)*(HOREL/HSTATIC)**FGA
1610     POABS(NS)  = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
1611
1612     CONTINUE
1613
1614 C
1615 C     CALCULATE PROPERTIES WITHIN THE SECOND BLADE ROW
1616 C
1617 DO 156 NS = NLE2,NTE2
1618     FRAC = (SDIST(NS)-SDIST(NLE2))/(SDIST(NTE2)-SDIST(NLE2))
1619     VTLOC(NS) = VT_LE2 + FRAC*(VT_TE2 - VT_LE2)
1620     WTLOC      = VTLOC(NS) - U_LOC(NS)
1621     VMLOC      = VM_LOC(NS)
1622     VABS(NS)   = SQRT(VTLOC(NS)*VTLOC(NS) + VMLOC*VMLOC)
1623     VREL(NS)   = SQRT(WTLOC*WTLOC + VMLOC*VMLOC)
1624     HOLOC(NS)  = HOMID(NSTG) + U_LOC(NS)*(VTLOC(NS) - VT_LE2)
1625     SLOC(NS)   = SMID(NSTG) + 0.5*FRAC*(SEXIT(NSTG) - S(NSTG))
1626     CALL PROPS(1,1,HOLOC(NS),SLOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),
1627 &             WET(NS),VABS(NS),G(NS),VS(NS),1,IProps,IWET)
1628     HSTATIC    = HOLOC(NS) - 0.5*VABS(NS)*VABS(NS)
1629     HOREL      = HSTATIC + 0.5*VREL(NS)*VREL(NS)
1630     POREL(NS)  = PLOC(NS)*(HOREL/HSTATIC)**FGA
1631     POABS(NS)  = PLOC(NS)*(HOLOC(NS)/HSTATIC)**FGA
1632     MACH_REL(NS) = VREL(NS)/VS(NS)
1633     MACH_ABS(NS) = VABS(NS)/VS(NS)
1634
1635     CONTINUE
1636
1637 C
1638 C     CALCULATE THE STAGE REACTION.
1639 C
1640 IF(TURBO_TYP.EQ.'T') REACN(NSTG) =
1641 & (TLOC(NLE2) - TLOC(NTE2))/(TLOC(NLE1)-TLOC(NTE2))
1642 IF(TURBO_TYP.EQ.'C') REACN(NSTG) =
1643 & (TLOC(NTE1) - TLOC(NLE1))/(TLOC(NTE2)-TLOC(NLE1))
1644
1645 C
1646 C     OUTPUT TO SCREEN
1647 C
1648 WRITE(6,*) ' LOCAL FLOW PROPERTIES THROUGH THE STAGE'
1649 WRITE(6,*) '          VM          P          T          RHO
1650 & HO          S'
1651 DO NS = 1,NSS
1652     WRITE(6,158) VM_LOC(NS),PLOC(NS),TLOC(NS),RHOLOC(NS),HOLOC(NS),
1653 &             SLOC(NS)
1654     END DO
1655     FORMAT(6F12.4)
1656
1657 C
1658 C*****
1659 C*****
1660
1661 WRITE(6,*) ' *****'
1662 WRITE(6,*) ' *****'
1663 WRITE(6,*)
1664 WRITE(6,*) ' STAGE NUMBER ', NSTG

```

```

1657 WRITE(6,*)
1658 WRITE(6,*) ' MASS FLOW RATE = ', FLOW
1659 WRITE(6,*) ' ROTATIONAL SPEED, RPM = ', RPM
1660 WRITE(6,*) ' THE FLOW COEFFICIENT,  $V_x/U$  = ', PHI(NSTG)
1661 WRITE(6,*) ' THE STAGE LOADING COEFFICIENT  $DH/U^{*2}$  = ', PSI(NSTG)
1662 WRITE(6,*) ' THE STAGE REACTION BASED ON ENTHALPY = ',
1663 & REACN(NSTG)
1664 WRITE(6,*) ' THE AXIAL VELOCITY = ', VX(NSTG)
1665 WRITE(6,*) ' DESIGN POINT ROTATIONAL SPEED = ', U(NSTG)
1666 WRITE(6,*) ' DESIGN POINT RADIUS = ',
1667 & RDES(NSTG)
1668 WRITE(6,*) ' STAGE ISENTROPIC ENTHALPY CHANGE = ',
1669 & DHOIS(NSTG)/1000.
1670 WRITE(6,*) ' STAGE ACTUAL ENTHALPY CHANGE = ',
1671 & DHO(NSTG)/1000.
1672 WRITE(6,*)
1673 WRITE(6,*) ' ***** '
1674 C
1675 C*****
1676 C*****
1677 C OUTPUT TO SCREEN
1678 C
1679 WRITE(6,*) ' XHUB XTIP RHUB RTIP
1680 & QLEN PITCH QANGL DENSITY '
1681 C
1682 DO 95 NS = 1,NSS
1683 C
1684 C SET THE QO LINE ANGLES
1685 C
1686 IF(NS.LE.NLE1) THEN
1687 DENS = RHOLOC(NS)
1688 FRAC = (SDIST(NS) - SDIST(1))/(SDIST(NLE1) - SDIST(1))
1689 QANGL = FRAC*QLE_ROW1(NSTG)*DEGRAD
1690 & + (1.-FRAC)*(0.5*PI+PITCH_ANGL(1))
1691 END IF
1692 IF(NS.GE.NLE1.AND.NS.LE.NTE1) THEN
1693 FRAC = (SDIST(NS) - SDIST(NLE1))/(SDIST(NTE1) - SDIST(NLE1))
1694 DENS = RHOLOC(NS)
1695 QANGL = DEGRAD*(FRAC*QTE_ROW1(NSTG)+(1.-FRAC)*QLE_ROW1(NSTG))
1696 END IF
1697 IF(NS.GT.NTE1.AND.NS.LE.NLE2) THEN
1698 FRAC = (SDIST(NS) - SDIST(NTE1))/(SDIST(NLE2) - SDIST(NTE1))
1699 DENS = RHOLOC(NS)
1700 QANGL = DEGRAD*(FRAC*QLE_ROW2(NSTG)+(1.-FRAC)*QTE_ROW1(NSTG))
1701 END IF
1702 IF(NS.GT.NLE2.AND.NS.LE.NTE2) THEN
1703 FRAC = (SDIST(NS) - SDIST(NLE2))/(SDIST(NTE2) - SDIST(NLE2))
1704 DENS = RHOLOC(NS)
1705 QANGL = DEGRAD*(FRAC*QTE_ROW2(NSTG)+(1.-FRAC)*QLE_ROW2(NSTG))
1706 END IF
1707 IF(NS.GT.NTE2) THEN
1708 FRAC = (SDIST(NS) - SDIST(NTE2))/(SDIST(NSS) - SDIST(NTE2))
1709 DENS = RHOLOC(NS)
1710 QANGL = FRAC*(0.5*PI + PITCH_ANGL(NSS))
1711 & + (1.-FRAC)*QTE_ROW2(NSTG)*DEGRAD
1712 END IF
1713 C
1714 C SET THE HUB AND TIP COORDINATES
1715 C
1716 DENSMID = DENS
1717 NLOOP = 0
1718 C*****
1719 C START THE ITERATION TO ESTIMATE THE MEAN DENSITY.
1720 C
1721 CONTINUE
1722 C
1723 NLOOP = NLOOP + 1
1724 C
1725 VOLFLO = FLOW/DENSMID
1726
1727 QLEN1 = VOLFLO/(VM_LOC(NS)*SIN(QANGL- PITCH_ANGL(NS)))
1728 & /(2*PI*RMEAN(NS))
1729 C INCREASE QLEN TO ALLOW FOR BLOCKAGE.

```

```

1730      QLEN      = QLEN1/(1.0 - FBLOCK(NS))
1731 C
1732      AAXIAL = VOLFLO*SIN(QANGL)
1733      &        /(PI*VM_LOC(NS)*SIN(QANGL-PITCH_ANG(NS)))
1734 C INCREASE THE ANNULUS AREA TO ALLOW FOR BLOCKAGE
1735      AAXIAL = AAXIAL/(1.0 - FBLOCK(NS))
1736 C
1737      IF(IFHUB.EQ.'M') THEN
1738          XHUB(NS) = XMEAN(NS) - 0.5*QLEN*COS(QANGL)
1739          XTIP(NS) = XMEAN(NS) + 0.5*QLEN*COS(QANGL)
1740          RHUB(NS) = RMEAN(NS) - 0.5*QLEN*SIN(QANGL)
1741          RTIP(NS) = RMEAN(NS) + 0.5*QLEN*SIN(QANGL)
1742          GO TO 161
1743      END IF
1744 C
1745      IF(IFHUB.EQ.'H') THEN
1746          RHUB(NS) = RMEAN(NS)
1747          XHUB(NS) = XMEAN(NS)
1748          RTIP(NS) = SQRT(RHUB(NS)*RHUB(NS) + AAXIAL)
1749          XTIP(NS) = XHUB(NS) + (RTIP(NS) - RHUB(NS))/TAN(QANGL)
1750      IF(QANGL.GT.0.99*PI.OR.QANGL.LT.0.01*PI) THEN
1751          RTIP(NS) = RHUB(NS) + QLEN*SIN(QANGL)
1752          XTIP(NS) = XHUB(NS) + QLEN*COS(QANGL)
1753      END IF
1754      END IF
1755 C
1756      IF(IFHUB.EQ.'T') THEN
1757          RTIP(NS) = RMEAN(NS)
1758          XTIP(NS) = XMEAN(NS)
1759          RHUB(NS) = SQRT(RTIP(NS)*RTIP(NS) - AAXIAL)
1760          XHUB(NS) = XTIP(NS) - (RTIP(NS) - RHUB(NS))/TAN(QANGL)
1761      IF(QANGL.GT.0.99*PI.OR.QANGL.LT.0.01*PI) THEN
1762          RHUB(NS) = RTIP(NS) - QLEN*SIN(QANGL)
1763          XHUB(NS) = XTIP(NS) - QLEN*COS(QANGL)
1764      END IF
1765      END IF
1766 C*****
1767 C MAKE AN ESTIMATE OF THE MEAN DENSITY
1768 C AND USE IT TO CALCULATE THE VOLUME FLOW. ITERATE 5 TIMES .
1769 C
1770      IF(IFHUB.EQ.'H') THEN
1771          REQL      = SQRT((RTIP(NS)*RTIP(NS) + RHUB(NS)*RHUB(NS))/2.0)
1772          RAVG      = 0.5*(RHUB(NS) + REQL)
1773          VTAVG     = 0.5*(VTLOC(NS) + VTLOC(NS)*RHUB(NS)/REQL)
1774          DENSAVG  = 0.5*(DENS + DENSMID)
1775          DPDR     = DENSAVG*VTAVG*VTAVG/RAVG
1776          DENSMID  = DENS + DPDR*(REQL-RHUB(NS))/(GAMM*RGAS*TLOC(NS))
1777      END IF
1778 C
1779      IF(IFHUB.EQ.'T') THEN
1780          REQL      = SQRT((RTIP(NS)*RTIP(NS) + RHUB(NS)*RHUB(NS))/2.0)
1781          RAVG      = 0.5*(RTIP(NS) + REQL)
1782          VTAVG     = 0.5*(VTLOC(NS) + VTLOC(NS)*RTIP(NS)/REQL)
1783          DENSAVG  = 0.5*(DENS + DENSMID)
1784          DPDR     = DENSAVG*VTAVG*VTAVG/RAVG
1785          DENSMID  = DENS + DPDR*(REQL-RTIP(NS))/(GAMM*RGAS*TLOC(NS))
1786      END IF
1787 C
1788 C      WRITE(6,*) ' ORIG DENS=',DENS,'MEAN DENS=',DENSMID,
1789 C      & 'RHUB,REQL,RTIP', RHUB(NS),REQL,RTIP(NS)
1790 C
1791      IF(NLOOP.LT.5) GO TO 160
1792 C END OF THE ITERATION TO ESTIMATE THE MEAN DENSITY
1793 C
1794 C*****
1795 C JUMP TO HERE IF IFHUB = 'M'
1796 C CONTINUE
1797 C
1798      WRITE(6,201) XHUB(NS),XTIP(NS),RHUB(NS),RTIP(NS),QLEN,
1799      & PITCH_ANG(NS)*RADDEG,QANGL*RADDEG,DENS
1800      FORMAT(8F12.4)
1801 C
1802 C

```

```

1803      CONTINUE
1804  C
1805  C*****
1806  C   STORE THE HUB AND CASING COORDINATES FOR THE STAGE.
1807      DO 98 NS = 1,NSS
1808          XDIF = XTIP(NS) - XHUB(NS)
1809          RDIF = RTIP(NS) - RHUB(NS)
1810          SPAN(NS) = SQRT(XDIF*XDIF + RDIF*RDIF)
1811          XHUBALL(NSTG,NS) = XHUB(NS)
1812          XTIPALL(NSTG,NS) = XTIP(NS)
1813          RHUBALL(NSTG,NS) = RHUB(NS)
1814          RTIPALL(NSTG,NS) = RTIP(NS)
1815      CONTINUE
1816  C
1817  C*****
1818  C   CALCULATE THE ASPECT RATIOS .
1819          RTIPIN(NSTG) = RTIP(NLE1)
1820          RTIPEXIT(NSTG) = RTIP(NTE2)
1821          AXCHRD1(NSTG) = SDIST(NTE1) - SDIST(NLE1)
1822          AXCHRD2(NSTG) = SDIST(NTE2) - SDIST(NLE2)
1823          SPANIN = 0.5*(SPAN(NLE1) + SPAN(NTE1))
1824          SPANEXIT = 0.5*(SPAN(NLE2) + SPAN(NTE2))
1825          ASPN(NSTG) = SPANIN/AXCHRD1(NSTG)
1826          ASPR(NSTG) = SPANEXIT/AXCHRD2(NSTG)
1827  C
1828  C*****
1829  C*****
1830  C*****
1831  C   SET THE BLADE NUMBERS USING THE MODIFIED ZWEIFEL COEFFICIENT FOR BOTH
1832  C   TURBINES AND COMPRESSORS.
1833  C
1834  C   IF(TURBO_TYP.EQ.'T') THEN
1835       COSIN1 = COS(BIN_ROW1(NSTG))
1836       COSIN2 = COS(BIN_ROW2(NSTG))
1837       COSOUT1 = COS(BOU_ROW1(NSTG))
1838       COSOUT2 = COS(BOU_ROW2(NSTG))
1839       COS1 = AMIN1(COSIN1,COSOUT1)
1840       COS2 = AMIN1(COSIN2,COSOUT2)
1841       RAVG1 = 0.5*(RMEAN(NLE1) + RMEAN(NTE1))
1842       RAVG2 = 0.5*(RMEAN(NLE2) + RMEAN(NTE2))
1843       ROVMAVG1 = 0.5*(RHOLOC(NLE1)*VM_LOC(NLE1)
1844       &          + RHOLOC(NTE1)*VM_LOC(NTE1))
1845       ROVMAVG2 = 0.5*(RHOLOC(NLE2)*VM_LOC(NLE2)
1846       &          + RHOLOC(NTE2)*VM_LOC(NTE2))
1847
1848  C
1849       IF(TURBO_TYP.EQ.'T') THEN
1850           PCX1Z = ZWEIFEL*( POABS(NLE1) - PLOC(NTE1) )*RAVG1*1.0E05
1851       &          /( ROVMAVG1*ABS(RMEAN(NTE1)*VT_LE1 - RMEAN(NTE1)*VT_TE1) )
1852  C
1853           PCX2Z = ZWEIFEL*( POREL(NLE2) - PLOC(NTE2) )*RAVG2*1.0E05
1854       &          /( ROVMAVG2*ABS(RMEAN(NTE2)*VT_LE2 - RMEAN(NTE2)*VT_TE2) )
1855       END IF
1856  C
1857       IF(TURBO_TYP.EQ.'C') THEN
1858           WRITE(6,*) 'DPOREL' , (POREL(NLE1) - PLOC(NLE1))
1859           WRITE(6,*) 'DRVTHETA', VT_LE1,VT_TE1
1860           PCX1Z = ZWEIFEL*( POREL(NLE1) - PLOC(NLE1) )*RAVG1*1.0E05
1861       &          /( ROVMAVG1*ABS(RMEAN(NTE1)*VT_LE1 - RMEAN(NTE1)*VT_TE1) )
1862  C
1863           PCX2Z = ZWEIFEL*( POABS(NLE2) - PLOC(NLE2) )*RAVG2*1.0E05
1864       &          /( ROVMAVG2*ABS(RMEAN(NTE2)*VT_LE2 - RMEAN(NTE2)*VT_TE2) )
1865       END IF
1866  C
1867
1868  C   PCX1Z = 0.5*ZWEIFEL/ABS(RMEAN(NLE1)*TAN(BIN_ROW1(NSTG))
1869  C   &          - RMEAN(NTE1)*TAN(BOU_ROW1(NSTG))) *RAVG1
1870  C   &          /COS1**2
1871  C
1872  C   PCX2Z = 0.5*ZWEIFEL/ABS(RMEAN(NLE2)*TAN(BIN_ROW2(NSTG))
1873  C   &          - RMEAN(NTE2)*TAN(BOU_ROW2(NSTG))) *RAVG2
1874  C   &          /COS2**2
1875  C

```

```

1876      WRITE(6,*)
1877      WRITE(6,*)      ' PITCH TO AXIAL CHORD RATIOS BASED ON THE ZWEIFEL
1878      & COEFFICIENT '
1879      WRITE(6,*)      ' MODIFIED TO ALLOW FOR RADIUS AND STREAM SURFACE
1880      & THICKNESS CHANGES.'
1881      WRITE(6,*)      ' PCX1Z, PCX2Z ', PCX1Z,PCX2Z
1882      WRITE(6,*)
1883  C
1884  C SKIP THIS , USE THE ZWEIFEL COEFFICIENT AS ABOVE FOR COMPRESSORS.
1885  C
1886      GO TO 1234
1887  C
1888  C SET THE BLADE NUMBERS USING THE DIFFUSION FACTOR FOR COMPRESSORS.
1889
1890      IF(TURBO_TYP.EQ.'C') THEN
1891      SOL1 = 0.5*ABS (TAN(BIN_ROW1(NSTG)) - TAN(BOU_ROW1(NSTG)) )
1892      *
1893      *COS(BIN_ROW1(NSTG))*COS(BOU_ROW1(NSTG))
1894      &
1895      /(D_FAC*COS(BOU_ROW1(NSTG)) + COS(BIN_ROW1(NSTG))
1896      &
1897      - COS(BOU_ROW1(NSTG)))
1898      CX_CT = COS ( (BIN_ROW1(NSTG) + BOU_ROW1(NSTG))/2 )
1899      PCX1D = 1./(SOL1*CX_CT)
1900  C
1901      SOL2 = 0.5*ABS (TAN(BIN_ROW2(NSTG)) - TAN(BOU_ROW2(NSTG)) )
1902      *
1903      *COS(BIN_ROW2(NSTG))*COS(BOU_ROW2(NSTG))
1904      &
1905      /(D_FAC*COS(BOU_ROW2(NSTG)) + COS(BIN_ROW2(NSTG))
1906      &
1907      - COS(BOU_ROW2(NSTG)))
1908      CX_CT = COS ( (BIN_ROW2(NSTG) + BOU_ROW2(NSTG))/2 )
1909      PCX2D = 1./(SOL2*CX_CT)
1910      END IF
1911  C
1912      WRITE(6,*)
1913      WRITE(6,*) ' PITCH TO AXIAL CHORD RATIOS FROM DIFFUSION FACTOR.'
1914      WRITE(6,*) ' PCX1D, PCX2D ', PCX1D, PCX2D
1915      WRITE(6,*)
1916  C
1917      CONTINUE
1918  C
1919      PCX1 = PCX1Z
1920      PCX2 = PCX2Z
1921  C
1922      IF(PCX1.GT.2.0) PCX1 = 2.0
1923      IF(PCX2.GT.2.0) PCX2 = 2.0
1924      PITCH1 = PCX1*AXCHRD1(NSTG)
1925      PITCH2 = PCX2*AXCHRD2(NSTG)
1926      RSET_PC1 = 0.5*(RMEAN(NLE1) + RMEAN(NTE1))
1927      RSET_PC2 = 0.5*(RMEAN(NLE2) + RMEAN(NTE2))
1928      NROW1 = 2*PI*RSET_PC1/PITCH1
1929      NROW2 = 2*PI*RSET_PC2/PITCH2
1930  C
1931  C OUTPUT TO SCREEN.
1932      WRITE(6,*)
1933      WRITE(6,*) ' PITCH TO AXIAL CHORD RATIOS USED. '
1934      WRITE(6,*) ' PCX1, PCX2, PITCH1, PITCH2, NROW1, NROW2 ',
1935      &
1936      PCX1,PCX2,PITCH1,PITCH2,NROW1,NROW2
1937      WRITE(6,*)
1938  C
1939      NR1 = 2*NSTG -1
1940      NR2 = 2*NSTG
1941      NBLADE(NR1) = NROW1
1942      NBLADE(NR2) = NROW2
1943      WRITE(6,*) 'STAGE No, ROW No, No. BLADES',NSTG, NR1, NBLADE(NR1)
1944      WRITE(6,*) 'STAGE No, ROW No, No. BLADES',NSTG, NR2, NBLADE(NR2)
1945      WRITE(6,*)
1946  C
1947  C*****
1948  C*****
1949  C OUTPUT TO SCREEN
1950  C
1951      WRITE(6,*) '*****'
1952      WRITE(6,*) ' CONDITIONS FOR THE FIRST BLADE ROW OF THE STAGE. '
1953      IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE STATOR '
1954      IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR ROTOR'
1955      WRITE(6,*) '*****'

```

```

1949     WRITE(6,*) ' FIRST BLADE INLET AND EXIT ANGLES ',
1950 &      BIN_ROW1(NSTG)*RADDEG, BOUT_ROW1(NSTG)*RADDEG
1951     WRITE(6,*) ' FIRST BLADE AXIAL VELOCITY ', VXIN(NSTG)
1952     WRITE(6,*) ' FIRST BLADE INLET MACH NUMBER ', MACH_REL(NLE1)
1953     WRITE(6,*) ' FIRST BLADE EXIT MACH NUMBER ', MACH_REL(NTE1)
1954     WRITE(6,*) ' FIRST BLADE EXIT DENSITY ', RHOMID(NSTG)
1955     WRITE(6,*) ' FIRST BLADE EXIT PRESSURE ', PMID(NSTG)
1956     WRITE(6,*) ' FIRST BLADE INLET STAGN PRESS ', POABS(NLE1)
1957     WRITE(6,*) ' FIRST BLADE EXIT STAGN PRESS ', POABS(NTE1)
1958     WRITE(6,*) ' FIRST BLADE REL INLET STAG PRES', POREL(NLE1)
1959     WRITE(6,*) ' FIRST BLADE EXIT TEMPERATURE ', TMID(NSTG)
1960     WRITE(6,*) ' FIRST BLADE EXIT STAGN TEMP ', TO_MID
1961     WRITE(6,*) ' FIRST BLADE TIP RADIUS = ', RTIPIN(NSTG)
1962     WRITE(6,*) ' FIRST BLADE INLET SPAN = ', SPANIN
1963     WRITE(6,*) ' FIRST BLADE AXIAL CHORD= ', AXCHRD1(NSTG)
1964     WRITE(6,*) ' FIRST BLADE ASPECT RATIO = ', ASPN(NSTG)
1965     WRITE(6,*)
1966 C
1967 C
1968     WRITE(6,*) ' ***** '
1969     WRITE(6,*) ' CONDITIONS FOR THE SECOND BLADE ROW OF THE STAGE.'
1970     IF(TURBO_TYP.EQ.'T') WRITE(6,*) ' THIS IS A TURBINE ROTOR '
1971     IF(TURBO_TYP.EQ.'C') WRITE(6,*) ' THIS IS A COMPRESSOR STATOR'
1972     WRITE(6,*) ' ***** '
1973     WRITE(6,*) ' SECOND BLADE INLET AND EXIT ANGLES ',
1974 &      BIN_ROW2(NSTG)*RADDEG, BOUT_ROW2(NSTG)*RADDEG
1975     WRITE(6,*) ' SECOND BLADE AXIAL VELOCITY ', VXOUT(NSTG)
1976     WRITE(6,*) ' SECOND BLADE INLET MACH NUMBER ', MACH_REL(NLE2)
1977     WRITE(6,*) ' SECOND BLADE EXIT MACH NUMBER ', MACH_REL(NTE2)
1978     WRITE(6,*) ' SECOND BLADE DENSITY ', RHOEXIT(NSTG)
1979     WRITE(6,*) ' SECOND BLADE EXIT PRESSURE ', PEXIT(NSTG)
1980     WRITE(6,*) ' SECOND BLADE INLET STAGN PRES. ', POABS(NLE2)
1981     WRITE(6,*) ' SECOND BLADE REL INLET STAG PR.', POREL(NLE2)
1982     WRITE(6,*) ' SECOND BLADE ROW EXIT STAG PRES', POABS(NTE2)
1983     WRITE(6,*) ' SECOND BLADE TEMPERATURE ', TEXTIT(NSTG)
1984     WRITE(6,*) ' SECOND BLADE STAGN EXIT TEMP. ', TO_EXIT
1985     WRITE(6,*) ' SECOND BLADE TIP RADIUS = ', RTIPEXIT(NSTG)
1986     WRITE(6,*) ' SECOND BLADE EXIT SPAN = ', SPANEXIT
1987     WRITE(6,*) ' SECOND BLADE AXIAL CHORD = ', AXCHRD2(NSTG)
1988     WRITE(6,*) ' SECOND BLADE ASPECT RATIO= ', ASPR(NSTG)
1989     WRITE(6,*) ' ***** '
1990     WRITE(6,*) ' ***** '
1991 C
1992 C
1993     WRITE(6,*)
1994     WRITE(6,*) ' THE RADII THROUGH THE STAGE ARE - '
1995     WRITE(6,210) RHUB(NLE1),RTIP(NLE1),RHUB(NLE2),RTIP(NLE2)
1996     FORMAT('ROW 1 HUB RADIUS ',F8.4,' ROW 1 TIP RADIUS ',F8.4,
1997 & ' ROW 2 HUB RADIUS ',F8.4,' ROW 2 TIP RADIUS ',F8.4)
1998     WRITE(6,*)
1999 C
2000 C*****
2001 C*****
2002 C
2003     WRITE(6,*) ' ***** '
2004     WRITE(6,*) ' ***** '
2005     WRITE(6,*)
2006     WRITE(6,125)
2007     FORMAT(' DO YOU WANT TO CHANGE THE ANGLES FOR THIS STAGE ? ANSWER
2008 & "Y" OR "N" ')
2009     READ(5,*) ANSANGL
2010     WRITE(6,*) ANSANGL
2011     WRITE(10,122) ANSANGL
2012     FORMAT(A1,T25, ' DO YOU WANT TO CHANGE THE ANGLES FOR THIS STAGE ?
2013 & "Y" or "N"')
2014     WRITE(6,*) ' CHANGE ANGLES ANSWER WAS ', ANSANGL
2015     WRITE(6,*) ' ***** '
2016     WRITE(6,*) ' ***** '
2017     WRITE(6,*)
2018 C
2019     IF(ANSANGL.EQ.'Y'.OR.ANSANGL.EQ.'y') GO TO 500
2020 C
2021     NROW2 = 2*NSTG

```

```

2022      PEXIT(NROW2) = PEXIT(NSTG)
2023 C*****
2024 C*****
2025 C      GO TO 1000 IF THIS IS THE LAST STAGE
2026 C
2027      IF (NSTG.EQ.NSTAGES) GO TO 1000
2028 C*****
2029 C*****
2030 C      IF NOT THE LAST STAGE
2031 C      SET PARAMETERS FOR THE NEXT STAGE SAME AS FOR PRESENT STAGE.
2032 C
2033      ETA(NSTG+1) = ETA(NSTG)
2034      RHUB(NSTG+1) = RHUB(NSTG)
2035      RDES(NSTG+1) = RDES(NSTG)
2036      DHOIS(NSTG+1) = DHOIS(NSTG)
2037      DHO(NSTG+1) = DHO(NSTG)
2038      U(NSTG+1) = U(NSTG)
2039      HO(NSTG+1) = HO(NSTG) + DHO(NSTG)
2040      S(NSTG+1) = SEXIT(NSTG)
2041 C
2042      ALPHA_IN(NSTG+1) = ALPHA_IN(NSTG)
2043      ALPHA_OUT(NSTG+1) = ALPHA_OUT(NSTG)
2044      BIN_ROW1(NSTG+1) = BIN_ROW1(NSTG)
2045      BIN_ROW2(NSTG+1) = BIN_ROW2(NSTG)
2046      BOUT_ROW1(NSTG+1) = BOUT_ROW1(NSTG)
2047      BOUT_ROW2(NSTG+1) = BOUT_ROW2(NSTG)
2048      DEVN1(NSTG+1) = DEVN1(NSTG)
2049      DEVN2(NSTG+1) = DEVN2(NSTG)
2050      AINC1(NSTG+1) = AINC1(NSTG)
2051      AINC2(NSTG+1) = AINC2(NSTG)
2052 C
2053      REACN(NSTG+1) = REACN(NSTG)
2054      ASPN(NSTG+1) = ASPN(NSTG)
2055      ASPR(NSTG+1) = ASPR(NSTG)
2056      AXCHRD1(NSTG+1) = AXCHRD1(NSTG)
2057      AXCHRD2(NSTG+1) = AXCHRD2(NSTG)
2058      PHI(NSTG+1) = PHI(NSTG)
2059      PSI(NSTG+1) = PSI(NSTG)
2060      QLE_ROW1(NSTG+1) = QLE_ROW1(NSTG)
2061      QTE_ROW1(NSTG+1) = QTE_ROW1(NSTG)
2062      QLE_ROW2(NSTG+1) = QLE_ROW2(NSTG)
2063      QTE_ROW2(NSTG+1) = QTE_ROW2(NSTG)
2064      ROWGAP(NSTG+1) = ROWGAP(NSTG)
2065      STAGEGAP(NSTG+1) = STAGEGAP(NSTG)
2066      FBLOCK_LE(NSTG+1) = FBLOCK_LE(NSTG)
2067      FBLOCK_TE(NSTG+1) = FBLOCK_TE(NSTG)
2068 C
2069 C*****
2070 C*****
2071 C      RETURN TO 1100 TO START ON NEXT STAGE
2072 C
2073      GO TO 1100
2074 C
2075 C*****
2076 C*****
2077 C
2078      CONTINUE
2079 C
2080 C*****
2081 C*****
2082 C
2083 C      FORM CONTINUOUS STREAM SURFACES ON THE HUB AND TIP
2084 C
2085 C*****
2086 C*****
2087 C
2088      NALL = 0
2089      DO 333 NSTG = 1,NSTAGES
2090 C
2091      IF(NSTG.EQ.1) THEN
2092 C
2093      NEND = NTE2_STG(1)
2094      IF(NSTAGES.EQ.1) NEND = NSS_STG(1)

```



```

2095 DO 334 NS = 1,NEND
2096     NALL = NALL + 1
2097     XSURFHUB(NALL) = XHUBALL(NSTG,NS)
2098     RSURFHUB(NALL) = RHUBALL(NSTG,NS)
2099     XSURFTIP(NALL) = XTIPALL(NSTG,NS)
2100     RSURFTIP(NALL) = RTIPALL(NSTG,NS)
2101     IF(NS.EQ.NLE1_STG(NSTG)) NLE1_ALL(NSTG) = NALL
2102     IF(NS.EQ.NTE1_STG(NSTG)) NTE1_ALL(NSTG) = NALL
2103     IF(NS.EQ.NLE2_STG(NSTG)) NLE2_ALL(NSTG) = NALL
2104     IF(NS.EQ.NTE2_STG(NSTG)) NTE2_ALL(NSTG) = NALL
2105 CONTINUE
2106 C
2107     WRITE(6,*) ' STAGE No ', NSTG, 'NALL =', NALL
2108     WRITE(6,*) ' NLE1,NTE1,NLE2,NTE2', NLE1_ALL(NSTG),
2109 &             NTE1_ALL(NSTG),NLE2_ALL(NSTG),NTE2_ALL(NSTG)
2110 C
2111 C END OF NSTAGE = 1 LOOP
2112 END IF
2113 C
2114 IF(NSTG.EQ.1) GO TO 333
2115 C*****
2116 C*****
2117 C
2118     DXLAST = XSURFHUB(NALL) - XSURFHUB(NALL-1)
2119     DRLAST = RSURFHUB(NALL) - RSURFHUB(NALL-1)
2120     DSLAST = SQRT(DXLAST*DXLAST + DRLAST*DRLAST)
2121 C
2122 C REMOVE ANY OVERLAPPING POINTS ON THE UPSTREAM STAGE STREAM SURFACE.
2123 C CHECK FOR AND REMOVE ANY OVERLAP, THIS IS DONE ON THE HUB ONLY, SO PROBLEMS
2124 C CAN STILL ARISE AT THE TIP.
2125
2126     NS = 1
2127     NE = NTE2_STG(NSTG)
2128     IF(NSTG.EQ.NSTAGES) NE = NSS_STG(NSTG)
2129 C
2130 DO 335 N = NS,NE
2131 C
2132     DXNEXT = XHUBALL(NSTG,N) - XSURFHUB(NALL)
2133     DRNEXT = RHUBALL(NSTG,N) - RSURFHUB(NALL)
2134     PROJN = (DXNEXT*DXLAST + DRNEXT*DRLAST)/DSLAST
2135 C
2136 C SKIP ANY OVERLAPPING POINTS
2137     DISTMIN = 0.01*(AXCHRD1(NSTG) + AXCHRD2(NSTG))
2138
2139     IF(PROJN.GT.DISTMIN.OR.N.EQ.NLE1_STG(NSTG)
2140 &     .OR.N.EQ.NLE2_STG(NSTG)) THEN
2141     NALL = NALL + 1
2142     XSURFHUB(NALL) = XHUBALL(NSTG,N)
2143     RSURFHUB(NALL) = RHUBALL(NSTG,N)
2144     XSURFTIP(NALL) = XTIPALL(NSTG,N)
2145     RSURFTIP(NALL) = RTIPALL(NSTG,N)
2146     IF(N.EQ.NLE1_STG(NSTG)) NLE1_ALL(NSTG) = NALL
2147     IF(N.EQ.NTE1_STG(NSTG)) NTE1_ALL(NSTG) = NALL
2148     IF(N.EQ.NLE2_STG(NSTG)) NLE2_ALL(NSTG) = NALL
2149     IF(N.EQ.NTE2_STG(NSTG)) NTE2_ALL(NSTG) = NALL
2150     DXLAST = XSURFHUB(NALL) - XSURFHUB(NALL-1)
2151     DRLAST = RSURFHUB(NALL) - RSURFHUB(NALL-1)
2152     DSLAST = SQRT(DXLAST*DXLAST + DRLAST*DRLAST)
2153 END IF
2154 C
2155 CONTINUE
2156 C
2157 CONTINUE
2158 C NINTPTS IS THE NUMBER OF POINTS ON THE CONTINUOUS STREAM SURFACE.
2159 NINTPTS = NALL
2160 C
2161 C*****
2162 C*****
2163 C WRITE THE STREAM SURFACE COORDINATES TO THE SCREEN
2164 C
2165 C
2166 WRITE(6,*)
2167 WRITE(6,*) ' COORDINATES OF THE CONTINUOUS STREAM SURFACE BEFORE

```

```

2168      & SMOOTHING, NINTPTS = ',NINTPTS
2169      WRITE(6,*) ' XSURFHUB'
2170      WRITE(6,304) (XSURFHUB(N),N=1,NINTPTS)
2171      WRITE(6,*) ' XSURFTIP'
2172      WRITE(6,304) (XSURFTIP(N),N=1,NINTPTS)
2173      WRITE(6,*) ' RSURFHUB'
2174      WRITE(6,304) (RSURFHUB(N),N=1,NINTPTS)
2175      WRITE(6,*) ' RSURFTIP'
2176      WRITE(6,304) (RSURFTIP(N),N=1,NINTPTS)
2177      FORMAT(8F10.5)
2178      WRITE(6,*)
2179  C
2180  C*****
2181  C*****
2182  C  SET THE SURFACE DISTANCE ON THE STREAM SURFACE FOR USE IN THE SMOOTHING.
2183  C  AND CHECK FOR ANY OVERLAPPING POINTS ON THE HUB AND TIP STREAM SURFACES.
2184  C  ATTEMPT TO CORRECT FOR OVERLAPPING POINTS BY INTERCHANGING POINTS.
2185  C
2186      DO NSTG = 1,NSTAGES
2187      WRITE(6,*)
2188      WRITE(6,*) '  LEADING AND TRAILING EDGE POINTS ON THE CONTINUOUS
2189  &STREAM SURFACE, STAGE No ',NSTG
2190      WRITE(6,*)  NLE1_ALL(NSTG),NTE1_ALL(NSTG) ,
2191  &              NLE2_ALL(NSTG),NTE2_ALL(NSTG)
2192      END DO
2193  C
2194      SDISTHUB(1) = 0.0
2195      SDISTTIP(1) = 0.0
2196      XDIF_HUB = 1.0
2197      RDIF_HUB = 1.0
2198      XDIF_TIP = 1.0
2199      RDIF_TIP = 1.0
2200      IFWARNH = 0
2201      IFWARNT = 0
2202      DO 336 N = 2,NINTPTS
2203  C
2204      CONTINUE
2205      XDIF = XSURFHUB(N) - XSURFHUB(N-1)
2206      RDIF = RSURFHUB(N) - RSURFHUB(N-1)
2207      SDISTHUB(N) = SDISTHUB(N-1) + SQRT(XDIF*XDIF + RDIF*RDIF)
2208      PROJ_HUB = XDIF*XDIF_HUB + RDIF*RDIF_HUB
2209
2210      IF(PROJ_HUB.LT.0.0) THEN
2211          WRITE(6,*)
2212          WRITE(6,*) ' CONTINUOUS STREAM SURFACE POINT NUMBER ', N
2213          WRITE(6,*) ' INTERCHANGING POINTS ON THE HUB'
2214          TEMP = XSURFHUB(N-1)
2215          XSURFHUB(N-1) = XSURFHUB(N)
2216          XSURFHUB(N) = TEMP
2217          TEMP = RSURFHUB(N-1)
2218          RSURFHUB(N-1) = RSURFHUB(N)
2219          RSURFHUB(N) = TEMP
2220          IFWARNH = 1
2221          GO TO 401
2222      END IF
2223      XDIF_HUB = XDIF
2224      RDIF_HUB = RDIF
2225  C
2226      CONTINUE
2227      XDIF = XSURFTIP(N) - XSURFTIP(N-1)
2228      RDIF = RSURFTIP(N) - RSURFTIP(N-1)
2229      SDISTTIP(N) = SDISTTIP(N-1) + SQRT(XDIF*XDIF + RDIF*RDIF)
2230      PROJ_TIP = XDIF*XDIF_TIP + RDIF*RDIF_TIP
2231
2232      IF(PROJ_TIP.LT.0.0) THEN
2233          WRITE(6,*)
2234          WRITE(6,*) ' CONTINUOUS STREAM SURFACE POINT NUMBER ', N
2235          WRITE(6,*) ' INTERCHANGING POINTS ON THE TIP'
2236          TEMP = XSURFTIP(N-1)
2237          XSURFTIP(N-1) = XSURFTIP(N)
2238          XSURFTIP(N) = TEMP
2239          TEMP = RSURFTIP(N-1)
2240          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 C     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 C     CALL SMOOTH2(1,NINTPTS,NSMOOTH,SFAC,XSURFTIP,RSURFTIP)
2258 C
2259 C     WRITE(6,*)
2260 C     WRITE(6,*) ' COORDINATES OF THE CONTINUOUS STREAM SURFACE AFTER
2261 & SMOOTHING, NINTPTS = ',NINTPTS
2262 C     WRITE(6,*) ' XSURFHUB'
2263 C     WRITE(6,304) (XSURFHUB(N),N=1,NINTPTS)
2264 C     WRITE(6,*) ' XSURFTIP'
2265 C     WRITE(6,304) (XSURFTIP(N),N=1,NINTPTS)
2266 C     WRITE(6,*) ' RSURFHUB'
2267 C     WRITE(6,304) (RSURFHUB(N),N=1,NINTPTS)
2268 C     WRITE(6,*) ' RSURFTIP'
2269 C     WRITE(6,304) (RSURFTIP(N),N=1,NINTPTS)
2270 C     WRITE(6,*)
2271 C
2272 C*****
2273 C*****
2274 C
2275 C     WRITE(6,*)
2276 C     &'DO YOU WANT TO OUTPUT ALL BLADE ROWS TO THE FILE "stagen.dat"? '
2277 C     WRITE(6,*) ' ANSWER "Y" or "N" . '
2278 C     READ(5,*) ANSOUT
2279 C     IF(ANSOUT.EQ.'y') ANSOUT='Y'
2280 C     WRITE(6,*) ' ANSOUT = ', ANSOUT
2281 C     WRITE(10,337) ANSOUT
2282 C     FORMAT(A1,T25,' IS OUTPUT REQUESTED FOR ALL BLADE ROWS ? ')
2283 C
2284 C     NROW_OUT = 0
2285 C     DO 339 NOUT = 1,NROWS
2286 C
2287 C     IF(ANSOUT.EQ.'Y'.OR.ANSOUT.EQ.'y') THEN
2288 C         IFOUT(NOUT) = 'Y'
2289 C         NROW_OUT = NROWS
2290 C     ELSE
2291 C         IF(NOUT.EQ.1) THEN
2292 C             WRITE(6,*) ' FOR EACH ONE OF',NROWS,' BLADE ROWS'
2293 C             WRITE(6,*) ' TYPE "Y" or "N" TO CHOOSE WHETHER TO OUTPUT'
2294 C             WRITE(6,*) ' THE BLADE ROW DATA OR NOT.'
2295 C             END IF
2296 C
2297 C             WRITE(6,*)
2298 C             WRITE(6,*) ' INPUT "Y" or "N" FOR ROW NUMBER ',NOUT
2299 C
2300 C             READ(5,*) IFOUT(NOUT)
2301 C             IF(IFOUT(NOUT).EQ.'y') IFOUT(NOUT) = 'Y'
2302 C             WRITE(10,338) IFOUT(NOUT)
2303 C             FORMAT(A1,T25,' IS OUTPUT REQUESTED FOR THIS BLADE ROW ?')
2304 C             IF(IFOUT(NOUT).EQ.'Y') NROW_OUT = NROW_OUT + 1
2305 C         END IF
2306 C
2307 C     CONTINUE
2308 C
2309 C
2310 C*****
2311 C*****
2312 C*****
2313 C     NOW PREPARE AND WRITE OUT DATA FOR STAGEN

```

```

2314 C*****
2315 C*****
2316 C*****
2317 C
2318 C      WRITE OUTPUT FOR  STAGEN
2319 C
2320 C      OPEN(UNIT=9, FILE= 'stagen.dat')
2321 C
2322 C      WRITE(9,9000) RGAS, GAMM
2323 C      FORMAT(2F12.4,T25, ' GAS CONSTANT, GAMMA')
2324 C
2325 C      WRITE(9,9002) IM, KM
2326 C      FORMAT(2I10, T25, ' IM, KM ' )
2327 C
2328 C      WRITE(9,9001) FPRAT, FPMAX
2329 C      FORMAT(2F12.4,T25, ' FPRAT, FPMAX')
2330 C
2331 C      WRITE(9,9003) FRRAT, FRMAX
2332 C      FORMAT(2F12.4,T25, ' FRRAT, FRMAX')
2333 C
2334 C      WRITE(9,9005) 0
2335 C      FORMAT(I10,T25, ' IFDEFAULTS ')
2336 C
2337 C      WRITE(9,9004) NROW_OUT, NOSECT
2338 C      FORMAT(2I10,T25, ' NROWS, N SECTIONS ')
2339 C
2340 C      WRITE(9,9006) 1.0
2341 C      FORMAT(F10.3,T25, ' SCALING FACTOR ')
2342 C
2343 C*****
2344 C*****
2345 C      DO 3500 NR = 1, NROWS
2346 C
2347 C      SKIP THE OUTPUT FOR THIS ROW IF "IFOUT" IS NOT = "Y".
2348 C      IF(IFOUT(NR).NE.'Y') GO TO 3500
2349 C
2350 C      NROW = NR
2351 C      NSTG = (NR-1)/2 + 1
2352 C      IF(MOD(NROW,2).EQ.0) THEN
2353 C      NLEALL = NLE2_ALL(NSTG)
2354 C      NTEALL = NTE2_ALL(NSTG)
2355 C      ELSE
2356 C      NLEALL = NLE1_ALL(NSTG)
2357 C      NTEALL = NTE1_ALL(NSTG)
2358 C      END IF
2359 C
2360 C*****
2361 C*****
2362 C      RESTORE 1D VARIABLES FOR THIS STAGE
2363 C
2364 C      NSS = NSS_STG(NSTG)
2365 C      DO NS = 1, NSS
2366 C      XMEAN(NS) = XMEANALL(NSTG,NS)
2367 C      RMEAN(NS) = RMEANALL(NSTG,NS)
2368 C      VM_LOC(NS) = VMLOCALL(NSTG,NS)
2369 C      RHUB(NS) = RHUBALL(NSTG,NS)
2370 C      RTIP(NS) = RTIPALL(NSTG,NS)
2371 C      XHUB(NS) = XHUBALL(NSTG,NS)
2372 C      XTIP(NS) = XTIPALL(NSTG,NS)
2373 C      END DO
2374 C
2375 C
2376 C      NLE1 = NLE1_STG(NSTG)
2377 C      NTE1 = NTE1_STG(NSTG)
2378 C      NLE2 = NLE2_STG(NSTG)
2379 C      NTE2 = NTE2_STG(NSTG)
2380 C
2381 C      IF(MOD(NROW,2).EQ.0) THEN
2382 C      NLE = NLE2
2383 C      NTE = NTE2
2384 C      ELSE
2385 C      NLE = NLE1
2386 C      NTE = NTE1

```

```

2387         END IF
2388 C
2389         IF (TURBO_TYP.EQ.'T'.AND.MOD(NR,2).EQ.0) ROWTYP = 'R'
2390         IF (TURBO_TYP.EQ.'T'.AND.MOD(NR,2).NE.0) ROWTYP = 'S'
2391         IF (TURBO_TYP.EQ.'C'.AND.MOD(NR,2).EQ.0) ROWTYP = 'S'
2392         IF (TURBO_TYP.EQ.'C'.AND.MOD(NR,2).NE.0) ROWTYP = 'R'
2393 C
2394 C
2395         WRITE(9,*) '*****STARTING DATA FOR A NEW BLADE ROW
2396 &*****'
2397         WRITE(9,126) NROW
2398         FORMAT(' BLADE ROW NUMBER = ',T25, I5)
2399         WRITE(9,127) ROWTYP
2400         FORMAT(' BLADE ROW TYPE = ',T25, A1)
2401 C
2402 C INPUT THE NUMBER OF STREAMWISE GRID POINTS, UPSTREAM, ON AND DOWNSTREAM
2403 C OF THE BLADE ROW.
2404 C
2405         NPOINTS_UP = NINTUP
2406         IF (NR.EQ.1) NPOINTS_UP = NINTUP + NADDUP
2407         NPOINTS_DWN = NINTDWN
2408         IF (NR.EQ.NROWS) NPOINTS_DWN = NINTDWN + NADDWN
2409         NPOINTS_ON = NINTON
2410         WRITE(9,1008) NPOINTS_UP, NPOINTS_ON, NPOINTS_DWN
2411         FORMAT(3I5,T20, ' NPOINTS_UP, NPOINTS_ON, NPOINTS_DWN ')
2412 C
2413 C SET THE RELATIVE SPACINGS OF THE GRID POINTS.
2414 C
2415         WRITE(9,1001) 0.0, 0.5
2416         WRITE(9,1001) 0.1, 0.7
2417         WRITE(9,1001) 0.2, 1.0
2418         WRITE(9,1001) 0.3, 1.4
2419         WRITE(9,1001) 0.4, 2.0
2420         WRITE(9,1001) 0.5, 3.0
2421         WRITE(9,1001) 0.6, 3.0
2422         WRITE(9,1001) 0.7, 3.0
2423         WRITE(9,1001) 0.8, 2.5
2424         WRITE(9,1001) 0.9, 2.0
2425         WRITE(9,1001) 1.0, 1.5
2426         FORMAT(2F15.4,T35, ' FRACTION AXIAL CHORD, RELATIVE GRID SPACING')
2427 C
2428         WRITE(9,1013) NBLADE(NR)
2429         FORMAT(I10,T20, ' NUMBER OF BLADES IN ROW. ')
2430 C
2431         PIN = PINLET(NSTG)*1.0E05
2432         PMIDD = PMID(NSTG)*1.0E05
2433         PEX = PEXIT(NSTG)*1.0E05
2434         IF (TURBO_TYP.EQ.'T'.AND. ROWTYP.EQ.'S')
2435 & WRITE(9,1011) 0.0,PIN,PIN,PMIDD,PMIDD
2436
2437         IF (TURBO_TYP.EQ.'C'.AND. ROWTYP.EQ.'R')
2438 & WRITE(9,1011) RPM,PIN,PIN,PMIDD,PMIDD
2439
2440         IF (TURBO_TYP.EQ.'T'.AND. ROWTYP.EQ.'R')
2441 & WRITE(9,1011) RPM,PMIDD,PMIDD,PEX,PEX
2442
2443         IF (TURBO_TYP.EQ.'C'.AND. ROWTYP.EQ.'S')
2444 & WRITE(9,1011) 0.0,PMIDD,PMIDD,PEX,PEX
2445 C
2446         FORMAT(5F10.2,T55, 'RPM, STATIC PRESSURES THROUGH ROW')
2447 C
2448         RPMHUB = 0.0
2449         IF (ROWTYP.EQ.'R') RPMHUB = RPM
2450         WRITE(9,1014) 0, 0, 1, 1, 1, 1, 0.0, RPMHUB
2451         FORMAT(6I5,F10.5,F10.2,T55, 'TIP GAPS, WALL ROTNS and RPMHUB')
2452 C
2453 C*****
2454 C*****
2455 C*****
2456 C*****
2457 C NOW LOOP OVER ALL BLADE SECTIONS TO BE GENERATED.
2458 C
2459 C

```

```

2460 RDESIGNLE = RMEAN(NLE)
2461 RDESIGNTE = RMEAN(NTE)
2462 C
2463 C MAKE AN INITIAL GUESS OF THE BLADE THICKNESSES, ETC.
2464 C
2465 DO 4400 NS = 1, NOSECT
2466 IF (NSTG.EQ.1) THEN
2467     IF (ROWTYP.EQ.'S') TKMAX_S(NSTG,NS) = TKMAXS
2468     IF (ROWTYP.EQ.'S') XTKMAX_S(NSTG,NS) = XTKMAXS
2469     IF (ROWTYP.EQ.'R') TKMAX_R(NSTG,NS) = TKMAXR
2470     IF (ROWTYP.EQ.'R') XTKMAX_R(NSTG,NS) = XTKMAXR
2471 ELSE
2472     IF (ROWTYP.EQ.'S') TKMAX_S(NSTG,NS) = TKMAX_S(NSTG-1,NS)
2473     IF (ROWTYP.EQ.'S') XTKMAX_S(NSTG,NS) = XTKMAX_S(NSTG-1,NS)
2474     IF (ROWTYP.EQ.'R') TKMAX_R(NSTG,NS) = TKMAX_R(NSTG-1,NS)
2475     IF (ROWTYP.EQ.'R') XTKMAX_R(NSTG,NS) = XTKMAX_R(NSTG-1,NS)
2476 END IF
2477 CONTINUE
2478 C
2479 C*****
2480 C*****
2481 C START A LOOP OVER ALL BLADE SECTIONS TO BE GENERATED
2482 C
2483 DO 4500 NSECT = 1, NOSECT
2484 C
2485     FRACSPAN = FLOAT(NSECT-1)/FLOAT(NOSECT-1)
2486 C
2487 c SET THE LEADING AND TRAILING EDGE RADII FOR THIS SECTION.
2488     RSECTLE = RHUB(NLE) + FRACSPAN*(RTIP(NLE) - RHUB(NLE))
2489     RSECTTE = RHUB(NTE) + FRACSPAN*(RTIP(NTE) - RHUB(NTE))
2490 C
2491 C SET THE COORDINATES AT THIS SPANWISE POSITION.
2492 DO 1012 N= 1,NINTPTS
2493     XSECT(N) = XSURFHUB(N) + FRACSPAN*(XSURFTIP(N)-XSURFHUB(N))
2494     RSECT(N) = RSURFHUB(N) + FRACSPAN*(RSURFTIP(N)-RSURFHUB(N))
2495 CONTINUE
2496
2497 WRITE(9,*) '***** ROW NUMBER',NR,'*****
2498 &*****'
2499 WRITE(9,*) '*****STARTING NEW BLADE SECTION, SECTION NUMBER',
2500 & NSECT, '*****'
2501 WRITE(9,*) '*****BLANK LINE*****
2502 &*****'
2503 C
2504 WRITE(9,1015) 1
2505 FORMAT(I5, T25, ' INTYPE- TYPE OF BLADE GEOMETRY INPUT')
2506 WRITE(9,1007) 6, 200, 4
2507 FORMAT(3I5,T25, ' NPIN, NXPTS, NSMOOTH ')
2508 C
2509 C
2510 C*****
2511 C*****
2512 C SET THE BLADE ANGLES AT THE DESIGN RADIUS
2513 C
2514 C SET BLADE METAL ANGLES AT THE DESIGN RADIUS FOR THE FIRST ROW
2515 IF (MOD(NR,2).GT.0) THEN
2516 IF (BIN_ROW1(NSTG).GT.BOUT_ROW1(NSTG)) THEN
2517     BLE_DES = BIN_ROW1(NSTG) - AINC1(NSTG)*DEGRAD
2518     BTE_DES = BOUT_ROW1(NSTG) - DEVN1(NSTG)*DEGRAD
2519 ELSE
2520     BLE_DES = BIN_ROW1(NSTG) + AINC1(NSTG)*DEGRAD
2521     BTE_DES = BOUT_ROW1(NSTG) + DEVN1(NSTG)*DEGRAD
2522 ENDIF
2523 END IF
2524 C
2525 C SET BLADE METAL ANGLES AT THE DESIGN RADIUS FOR THE SECOND ROW
2526 IF (MOD(NR,2).EQ.0) THEN
2527 IF (BOUT_ROW2(NSTG).GT.BIN_ROW2(NSTG)) THEN
2528     BLE_DES = BIN_ROW2(NSTG) + AINC2(NSTG)*DEGRAD
2529     BTE_DES = BOUT_ROW2(NSTG) + DEVN2(NSTG)*DEGRAD
2530 ELSE
2531     BLE_DES = BIN_ROW2(NSTG) - AINC2(NSTG)*DEGRAD
2532     BTE_DES = BOUT_ROW2(NSTG) - DEVN2(NSTG)*DEGRAD

```

```

2533         ENDIF
2534     ENDIF
2535
2536 C*****
2537 C*****
2538 C     VARY THE ANGLES WITH SPAN FOR A FREE VORTEX DESIGN.
2539 C
2540 C     FIRST FOR A STATOR
2541     IF(ROWTYP.EQ.'S') THEN
2542         BLE = (1.0 - FRAC_TWIST)*BLE_DES +
2543 &         FRAC_TWIST*ATAN( TAN(BLE_DES)*RDESIGNLE/RSECTLE )
2544         BTE = (1.0 - FRAC_TWIST)*BTE_DES +
2545 &         FRAC_TWIST*ATAN( TAN(BTE_DES)*RDESIGNTE/RSECTTE )
2546     WRITE(6,*)
2547     WRITE(6,*) ' STAGE No ',NSTG,'STATOR  ROW NUMBER',NR,'SECTION No',
2548 & NSECT
2549     IF(TURBO_TYP.EQ.'C') THEN
2550         WRITE(6,*) ' INCIDENCE ANGLE = ', AINC2(NSTG)
2551         WRITE(6,*) ' DEVIATION ANGLE = ', DEVN2(NSTG)
2552     ELSE
2553         WRITE(6,*) ' INCIDENCE ANGLE = ', AINC1(NSTG)
2554         WRITE(6,*) ' DEVIATION ANGLE = ', DEVN1(NSTG)
2555     END IF
2556     WRITE(6,*) ' BLADE INLET METAL ANGLE = ', BLE*RADDEG
2557     WRITE(6,*) ' BLADE EXIT METAL ANGLE = ', BTE*RADDEG
2558 C END OF ROWTYP = 'S' LOOP
2559     END IF
2560 C
2561 C     NEXT FOR A ROTOR
2562     IF(ROWTYP.EQ.'R') THEN
2563         RRAT = RDESIGNLE/RSECTLE
2564         PHILEE = PHI_LOC(NLE)
2565         TAN_BABS = TAN(BLE_DES) + 1.0/PHILEE
2566         TAN_BABS = TAN_BABS*RRAT
2567         TAN_BLE = TAN_BABS - 1.0/PHILEE/RRAT
2568         BLE = (1.0 - FRAC_TWIST)*BLE_DES
2569 &         + FRAC_TWIST*ATAN(TAN_BLE)
2570 C
2571         RRAT = RDESIGNTE/RSECTTE
2572         PHITEE = PHI_LOC(NTE)
2573         TAN_BABS = TAN(BTE_DES) + 1.0/PHITEE
2574         TAN_BABS = TAN_BABS*RRAT
2575         TAN_BTE = TAN_BABS - 1.0/PHITEE/RRAT
2576         BTE = (1.0 - FRAC_TWIST)*BTE_DES
2577 &         + FRAC_TWIST*ATAN(TAN_BTE)
2578 C
2579     WRITE(6,*)
2580     WRITE(6,*) ' STAGE No ',NSTG,'ROTOR  ROW NUMBER',NR,'SECTION No',
2581 & NSECT
2582     IF(TURBO_TYP.EQ.'C') THEN
2583         WRITE(6,*) ' INCIDENCE ANGLE = ', AINC1(NSTG)
2584         WRITE(6,*) ' DEVIATION ANGLE = ', DEVN1(NSTG)
2585     ELSE
2586         WRITE(6,*) ' INCIDENCE ANGLE = ', AINC2(NSTG)
2587         WRITE(6,*) ' DEVIATION ANGLE = ', DEVN2(NSTG)
2588     END IF
2589     WRITE(6,*) ' BLADE INLET METAL ANGLE = ', BLE*RADDEG
2590     WRITE(6,*) ' BLADE EXIT METAL ANGLE = ', BTE*RADDEG
2591 C END OF ROWTYP = 'R' LOOP
2592     END IF
2593 C
2594 C*****
2595 C*****
2596 C     VARY THE TANGENT OF THE BLADE ANGLE WITH MERIDIONAL DISTANCE.
2597 C     THE DISTANCE IS TRANSFORMED BY "EXPO" WHICH IS SET IN THE DEFAULTS.
2598 C
2599     TAN1 = TAN(BLE)
2600     TAN6 = TAN(BTE)
2601     TAN2 = TAN1 + (TAN6 - TAN1)/5.
2602     TAN3 = TAN1 + 2.0*(TAN6 - TAN1)/5.
2603     TAN4 = TAN1 + 3.0*(TAN6 - TAN1)/5.
2604     TAN5 = TAN1 + 4.0*(TAN6 - TAN1)/5.
2605     WRITE(9,1003) 0.0, ATAN(TAN1)*RADDEG

```



```

2606      WRITE(9,1003) 0.2**EXPO,      ATAN(TAN2)*RADDEG
2607      WRITE(9,1003) 0.4**EXPO ,      ATAN(TAN3)*RADDEG
2608      WRITE(9,1003) 0.6**EXPO ,      ATAN(TAN4)*RADDEG
2609      WRITE(9,1003) 0.8**EXPO ,      ATAN(TAN5)*RADDEG
2610      WRITE(9,1003) 1.0,              ATAN(TAN6)*RADDEG
2611      FORMAT(2F12.4,T25,' BLADE CENTRE LINE ANGLES ' )
2612      BETADWN = ATAN(TAN6)*RADDEG
2613 C
2614 C*****
2615 C*****
2616 C
2617      IF(NSECT.EQ.1) THEN
2618 C
2619      WRITE(6,*)
2620      WRITE(6,*) '*****
2621 &*****'
2622      WRITE(6,*) '*****
2623 &*****'
2624      IF(ROWTYP.EQ.'R')
2625 &WRITE(6,*) 'STAGE NUMBER',NSTG,' ROW NUMBER',NR,' THIS IS A ROTOR'
2626      IF(ROWTYP.EQ.'S')
2627 &WRITE(6,*) 'STAGE NUMBER',NSTG,' ROW NUMBER',NR,' THIS IS A STATOR'
2628      WRITE(6,*) '*****
2629 &*****'
2630      WRITE(6,*) '*****
2631 &*****'
2632      ANSTK = 'N'
2633 C
2634      WRITE(6,*)
2635      WRITE(6,*) ' THE CURRENT VALUES WERE OF BLADE THICKNESS AND POINT
2636 & OF MAXIMUM THICKNESS ARE:'
2637 C
2638      IF(ROWTYP.EQ.'S') THEN
2639          DO NS = 1,NOSECT
2640              WRITE(6,1032) NS,TKMAX_S(NSTG,NS),XTKMAX_S(NSTG,NS)
2641          END DO
2642      END IF
2643 C
2644      IF(ROWTYP.EQ.'R') THEN
2645          DO NS = 1,NOSECT
2646              WRITE(6,1032) NS,TKMAX_R(NSTG,NS),XTKMAX_R(NSTG,NS)
2647          END DO
2648      END IF
2649 C
2650      FORMAT(' SECTION No.',I5,
2651 & ' MAX THICKNESS, POSITION OF MAX THICKNESS',2F12.4)
2652 C
2653      WRITE(6,*)
2654      WRITE(6,*) 'DO YOU WANT TO ACCEPT THESE ? ANSWER "Y" or "N".'
2655      READ(5,*) ANSTK
2656      IF(ANSTK.EQ.'Y') ANSTK = 'Y'
2657      IF(ANSTK.NE.'Y') ANSTK = 'N'
2658      WRITE(6,*) 'ANSTK = ', ANSTK
2659 C
2660      IF(ROWTYP.EQ.'S') WRITE(10,1030) ANSTK,NSTG
2661      FORMAT(A1,T6,'STATOR No.',I3, ' SET ANSTK = "Y" TO USE THE SAME
2662 & BLADE SECTIONS AS THE LAST STAGE')
2663      IF(ROWTYP.EQ.'R') WRITE(10,1031) ANSTK,NSTG
2664      FORMAT(A1,T6,'ROTOR No. ',I3, ' SET ANSTK = "Y" TO USE THE SAME
2665 & BLADE SECTIONS AS THE LAST STAGE')
2666 C
2667 C END OF NSECT = 1 LOOP
2668      END IF
2669 C
2670 C*****
2671 C*****
2672 C
2673      IF(ROWTYP.EQ.'S') THEN
2674 C
2675      IF(ANSTK.EQ.'N'.OR.ANSTK.EQ.'n') THEN
2676 C
2677      WRITE(6,*) '*****
2678      WRITE(6,*) ' STAGE NUMBER', NSTG, ' SECTION NUMBER ',NSECT

```

```

2679     WRITE(6,*) ' INPUT NEW VALUES OF "TKMAX" AND "XTKMAX" FOR A STATOR'
2680     WRITE(6,*) ' THE CURRENT VALUES ARE- ',
2681     &          TKMAX_S(NSTG,NSECT) , XTKMAX_S(NSTG,NSECT)
2682     WRITE(6,*) ' PRESS "A" TO ACCEPT THESE OR TYPE IN NEW VALUES.'
2683     READ(5,*,ERR = 1009) TKMAX_S(NSTG,NSECT) , XTKMAX_S(NSTG,NSECT)
2684     CONTINUE
2685     WRITE(6,*) ' THE NEW VALUES ARE- ',
2686     &          TKMAX_S(NSTG,NSECT) , XTKMAX_S(NSTG,NSECT)
2687     WRITE(6,*) ' *****'
2688 C
2689     WRITE(10,123) TKMAX_S(NSTG,NSECT) , XTKMAX_S(NSTG,NSECT) , NSTG,NSECT
2690     FORMAT(2F8.4,T25, ' MAX THICKNESS AND ITS LOCATION FOR STATOR',I3,
2691     &          ' SECTION No.',I3)
2692 C     END OF ANSTK = "N" LOOP
2693     END IF
2694 C
2695     WRITE(9,1002) TKLE,TKTE,TKMAX_S(NSTG,NSECT) , XTKMAX_S(NSTG,NSECT) ,
2696     &          XMODLE,XMODTE,TK_TYP
2697 C
2698 C     END OF ROWTYP = "S" LOOP .
2699     END IF
2700 C
2701 C*****
2702 C*****
2703 C
2704     IF(ROWTYP.EQ.'R') THEN
2705 C
2706         IF(ANSTK.EQ.'N'.OR.ANSTK.EQ.'n') THEN
2707             WRITE(6,*) ' *****'
2708             WRITE(6,*) ' STAGE NUMBER ', NSTG, ' SECTION NUMBER ',NSECT
2709             WRITE(6,*) ' INPUT NEW VALUES OF "TKMAX" AND "XTKMAX" FOR A ROTOR'
2710             WRITE(6,*) ' THE CURRENT VALUES ARE- ',
2711             &          TKMAX_R(NSTG,NSECT) , XTKMAX_R(NSTG,NSECT)
2712             WRITE(6,*) ' PRESS "A" TO ACCEPT THESE OR TYPE IN NEW VALUES.'
2713             READ(5,*,ERR = 1010) TKMAX_R(NSTG,NSECT) , XTKMAX_R(NSTG,NSECT)
2714             CONTINUE
2715             WRITE(6,*) ' THE NEW VALUES ARE- ',
2716             &          TKMAX_R(NSTG,NSECT) , XTKMAX_R(NSTG,NSECT)
2717             WRITE(6,*) ' *****'
2718 C
2719             WRITE(10,124) TKMAX_R(NSTG,NSECT) , XTKMAX_R(NSTG,NSECT) , NSTG,NSECT
2720             FORMAT(2F8.4,T25, ' MAX THICKNESS AND ITS LOCATION FOR ROTOR ',I3,
2721             &          ' SECTION No.',I3)
2722 C         END OF ANSTK = "N" LOOP
2723             END IF
2724 C
2725             WRITE(9,1002) TKLE,TKTE,TKMAX_R(NSTG,NSECT) , XTKMAX_R(NSTG,NSECT) ,
2726             &          XMODLE,XMODTE,TK_TYP
2727 C         END OF ROWTYP = "R" LOOP.
2728             END IF
2729             FORMAT(7F10.4,T75, ' BLADE PROFILE SPECIFICATION')
2730 C
2731 C*****
2732 C*****
2733 C*****
2734 C
2735         ROTN = 0.0
2736         XROT = 0.5
2737         YROT = 0.5
2738         IF(IF_ROT.EQ.'Y') THEN
2739             WRITE(6,*)
2740             WRITE(6,*) ' INPUT THE ANGLE BY WHICH THIS SECTION WILL BE TWISTED
2741             & IN THE CLOCKWISE DIRECTION, IN DEGREES.'
2742             READ(5,*,ERR = 1051) ROTN
2743             CONTINUE
2744             WRITE(10,1052) ROTN
2745             FORMAT(F10.4,T25, ' ANGLE OF CLOCKWISE ROTATION OF THIS SECTION.')
2746             END IF
2747 C
2748 C*****
2749 C*****
2750 C*****

```

```

2751 C
2752 FCHORD = 1.0
2753 FPERP = 0.0
2754 FTKSCALE = 1.0
2755 WRITE(9,1004) FCHORD, FPERP, FTKSCALE
2756 FORMAT(3F10.4,T50, ' FCHORD, FPERP, FTKSCALE')
2757 C
2758 C
2759 WRITE(9,1006) ROTN,XROT,YROT
2760 FORMAT(3F10.4,T50, ' ROTN,XROT,YROT ' )
2761 C
2762 C
2763 XCUP = 0.25
2764 XCDWN = 0.25
2765 IF(NR.EQ.1) XCUP = 0.5
2766 IF(NR.EQ.NROWS) XCDWN = 0.5
2767 BETUP = ATAN(TAN1)*RADDEG
2768 BETDWN = BETADWN
2769 WRITE(9,1005) XCUP, XCDWN, BETUP, BETDWN
2770 FORMAT(4F10.4,T50, ' XCUP, XCDWN, BETUP, BETDWN')
2771 C
2772 C
2773 WRITE(9,*) ' BLANK LINE '
2774 WRITE(9,1023) NINTPTS
2775 FORMAT(I5,T20, ' NUMBER OF POINTS ON THE STREAM SURFACE.')
```

2776 WRITE(9,1016) (XSECT(N),N=1,NINTPTS)

2777 WRITE(9,1017) (RSECT(N),N=1,NINTPTS)

2778 WRITE(9,1018) XSECT(NLEALL),XSECT(NTEALL),

2779 & RSECT(NLEALL),RSECT(NTEALL)

2780 FORMAT(8F12.6)

2781 FORMAT(8F12.6)

2782 FORMAT(4F12.6,T50, ' LEADING AND TRAILING EDGE COORDINATES')

2783 C

2784 C

2785 FCENTROID = 1.0

2786 FTANG = 0.0

2787 FLEAN = 0.0

2788 FSWEEP = 0.0

2789 FAXIAL = 0.0

2790 WRITE(9,1019) FCENTROID, FTANG, FLEAN, FSWEEP, FAXIAL

2791 FORMAT(5F10.4,T50, ' FCENTROID, FTANG, FLEAN, FSWEEP, FAXIAL')

2792 C

2793 FSCALE = 1.0

2794 FCONST = 0.0

2795 WRITE(9,1020) FSCALE, FCONST

2796 FORMAT(2F10.4,T50, ' FSCALE, FCONST ')

2797 C

2798 C

2799 C

2800 CONTINUE

2801 C

2802 C*****

2803 C*****

2804 C*****

2805

2806 C

2807 C

2808 CONTINUE

2809 C

2810 C

2811 FIND THE INLET ENDWALL SLOPES, WHICH ARE USED TO SET THE INLET PITCH ANGLE .

2812 DXHUB = XSURFHUB(2) - XSURFHUB(1)

2813 DXTIP = XSURFTIP(2) - XSURFTIP(1)

2814 DRHUB = RSURFHUB(2) - RSURFHUB(1)

2815 DRTIP = RSURFTIP(2) - RSURFTIP(1)

2816 DSHUB = SQRT(DXHUB*DXHUB + DRHUB*DRHUB)

2817 DSTIP = SQRT(DXTIP*DXTIP + DRTIP*DRTIP)

2817 PITCHHUB = ATAN2(DRHUB,DXHUB)*RADDEG

2818 PITCHTIP = ATAN2(DRTIP,DXTIP)*RADDEG

2819 C*****

2820 C*****

2821 C*****

2822 C

2823 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.

```

```

2897 C
2898 IWET = 0
2899 DO 11 I=1,IM
2900 G(I) = GAMM
2901 GG = G(I)/(G(I)-1.0)
2902 H = HO(I) - 0.5*V(I)*V(I)
2903 P(I) = POIN*((H/HOIN)**GG)*EXP((SI-S(I))/RGAS)
2904 T(I) = H/CPGAS
2905 RHO(I) = P(I)/RGAS/T(I)*100000.0
2906 VS(I) = SQRT(G(I)*RGAS*T(I))
2907 WET(I) = 0.0
2908 GO TO 12
2909 C
2910 C STEAM PROPERTIES.
2911 C
2912 CONTINUE
2913 C
2914 DO 10 I=1,IM
2915 HKJ= (HO(I)-0.5*V(I)*V(I))/1000.
2916 SKJ=S(I)/1000.
2917 IF(SKJ.GT.8.63) SKJ=8.63
2918 IF(HKJ.LT.(2000.+293.*(SKJ-6.8))) HKJ=2000.+293.*(SKJ-6.8)
2919 HABS=HKJ*0.2308 - 448.25
2920 HSAT= 4647.0 - 405.2*SKJ + 18.7*SKJ*SKJ
2921 IF(SKJ.LT.6.7) HSAT=-3533.35 + 2039.03*SKJ -164.06*SKJ*SKJ
2922 IF(HKJ.LT.HSAT) GO TO 5
2923 HR=(HKJ -3125.)/475.
2924 SR=(SKJ -7.1)/0.7
2925 C1=.9991-.02728*SR+.04982*SR*SR-.01596*SR*SR*SR
2926 C2=.001964-.00655*SR+.007398*SR*SR-.01471*SR*SR*SR
2927 C3=.02477+.000779*SR-.01044*SR*SR+.002993*SR*SR*SR
2928 C4=-.004313-.001639*SR-.000766*SR*SR+.01267*SR*SR*SR
2929 FM=C1+C2*HR+C3*HR*HR+C4*HR*HR*HR
2930 RHO(I)=(HABS/303.23)**3.333 *EXP(2.2*(7.268-SKJ))/0.2029
2931 P(I)=3.0435*RHO(I)*HABS/303.23
2932 RHO(I)=RHO(I)*FM-0.0025
2933 C1=1.001-.02005*SR+.04543*SR*SR-.01537*SR*SR*SR
2934 C2=-.008486-.008498*SR+.006452*SR*SR-.006273*SR*SR*SR
2935 C3=.02274+.003286*SR-.01253*SR*SR-.008903*SR*SR*SR
2936 C4=-.008879-.001671*SR+.007301*SR*SR+.01092*SR*SR*SR
2937 FM=C1+C2*HR+C3*HR*HR+C4*HR*HR*HR
2938 P(I)=P(I)*FM
2939 C1=1.016- .03847*SR+.03418*SR*SR-.01174*SR*SR*SR
2940 C2=-.002987+.0009509*SR+.001699*SR*SR-.004955*SR*SR*SR
2941 C3=-.001568-.0008933*SR-.00569*SR*SR-.004383*SR*SR*SR
2942 C4=-.004404-.00373*SR+.00566*SR*SR+.006968*SR*SR*SR
2943 FM=C1+C2*HR+C3*HR*HR+C4*HR*HR*HR
2944 T(I)= HABS*2.2*FM
2945 VS(I)= SQRT(130000.*P(I)/RHO(I))
2946 G(I) =1.3
2947 WET(I)=0.0
2948 GO TO 10
2949 IF(SKJ.LT.6.7) GO TO 20
2950 DSDH=-.00033 +.00055*(SKJ+.00033*HKJ-.66)/(-0.1+.00055*HKJ)
2951 DSDH= 1.0/DSDH
2952 A= 4647.0 -HKJ +SKJ*DSDH
2953 B= -405.2-1.0*DSDH
2954 C= 18.7
2955 SSAT=- (B +SQRT(B*B-4*A*C))/(2*C)
2956 HSAT= 4647. -405.2*SSAT+18.7*SSAT*SSAT
2957 SR = (SSAT-7.608)*1.272
2958 P(I)=.47843-1.1055*SR+1.3003*SR*SR-1.039*SR*SR*SR+.65315*SR*SR*SR*
2959 SR-.3305*SR*SR*SR*SR*SR+.09311*SR*SR*SR*SR*SR*SR
2960 TSAT=80.207-57.067*SR+12.07*SR*SR-2.997*SR*SR*SR+.0689*SR*SR*SR*SR
2961 +.6145*SR*SR*SR*SR*SR
2962 ROSAT=3.3751+7.2791*SR+7.7026*SR*SR+5.4648*SR*SR*SR+2.9851*SR*SR*S
2963 R*SR+1.1687*SR*SR*SR*SR*SR +.2245*SR*SR*SR*SR*SR*SR
2964 GO TO 21
2965 DSDH=-.0017+.0008*(SKJ+.0017*HKJ-3.4)/(-0.6+.0008*HKJ)
2966 DSDH=1.0/DSDH
2967 A= -3533.35 -HKJ +SKJ*DSDH
2968 B= 2039.03 - DSDH
2969 C= -164.06

```

```

2970      SSAT=-(B +SQRT(B*B-4*A*C))/(2*C)
2971      HSAT = -3533.35 +2039.03*SSAT - 164.06*SSAT*SSAT
2972      SR = (SSAT-6.355) * 2.1542
2973      P(I)=19.064-24.34*SR+13.601*SR*SR-3.424*SR*SR*SR+0.1561*SR*SR*SR*
2974      SR + 0.2642*SR*SR*SR*SR*SR -0.32049*SR*SR*SR*SR*SR*SR
2975      TSAT=209.99 -63.872*SR +3.4481*SR*SR +2.7869*SR*SR*SR +2.0565*SR*
2976      SR*SR*SR -0.91474*SR*SR*SR*SR*SR -1.6936*SR*SR*SR*SR*SR*SR
2977      ROSAT=0.10432 + 0.13022*SR +0.087203*SR*SR +0.035654*SR*SR*SR+0.00
2978      5*SR*SR*SR*SR +0.0052544*SR*SR*SR*SR*SR+.0040945*SR*SR*SR*SR*
2979      SR*SR
2980      GO TO 22
2981      IF(SSAT.LT.8.35) GO TO 22
2982      ROSAT = 28.6 + 75.6*(SSAT-8.4) +110.0*(SSAT-8.4)**2
2983      P(I) =(49.5 -140.0*(SSAT-8.4) +163.0*(SSAT-8.4)**2)* 0.001
2984      T(I) = TSAT + 273.16
2985      WET(I)=(HSAT-HKJ)/(HSAT-4.19*TSAT)
2986      RHO(I)=1.0/ROSAT/(1.0-WET(I))
2987      G(I)=1.12
2988      IF(WET(I).LT.0.01) G(I)=1.3-18.0*WET(I)
2989      VS(I)=SQRT(G(I)*100000.*P(I)/RHO(I))
2990      IWET=1
2991      CONTINUE
2992      CONTINUE
2993      RETURN
2994      END
2995      C*****
2996      C*****
2997      C*****
2998      C
2999      SUBROUTINE SMOOTH(N1,N2,NSMOOTH,FSMOOTH,FRAC,VAR)
3000      C
3001      C THIS ROUTINE MAKES THE QUANTITY TO BE SMOOTHED VARY LINEARLY WITH SURFACE DISTANCE
3002      C
3003      C PARAMETER(NG=99, NST=20, NSC= 11)
3004      C
3005      C DIMENSION FRAC(NG),VAR(NG),TEMP(NG)
3006      C
3007      C DO 10 ITS = 1,NSMOOTH
3008      C
3009      C DO N = N1,N2
3010      C TEMP(N) = VAR(N)
3011      C END DO
3012      C
3013      C DO N = N1+1, N2-1
3014      C FLEFT = FRAC(N) - FRAC(N-1)
3015      C FRIGHT = FRAC(N+1) - FRAC(N)
3016      C AVG = ( FLEFT*TEMP(N+1) + FRIGHT*TEMP(N-1) )/(FLEFT + FRIGHT)
3017      C VAR(N) = (1.0-FSMOOTH)*VAR(N) + FSMOOTH*AVG
3018      C END DO
3019      C
3020      C CONTINUE
3021      C
3022      C RETURN
3023      C END
3024      C*****
3025      C*****
3026      C*****
3027      C
3028      C SUBROUTINE SMOOTH2(N1,N2,NSMOOTH,FSMOOTH,XVAL,RVAL)
3029      C
3030      C THIS SUBROUTINE SMOOTHS BY MOVING EACH POINT ALONG A PERPENDICUAR TOWARDS THE
3031      C LINE JOINING ITS TWO ADJACENT POINTS.
3032      C
3033      C PARAMETER(NG=99, NST=20, NSC= 11)
3034      C
3035      C DIMENSION XVAL(NG),RVAL(NG),TEMPX(NG),TEMPR(NG)
3036      C
3037      C DO 10 ITS = 1,NSMOOTH
3038      C
3039      C DO N = N1,N2
3040      C 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

```