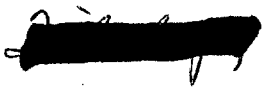


C-1396



BRL CR 39

BRL

AD726460

CONTRACT REPORT NO. 39

3SR-350

HELP, A MULTI-MATERIAL EULERIAN PROGRAM FOR COMPRESSIBLE FLUID
AND ELASTIC-PLASTIC FLOWS IN TWO SPACE DIMENSIONS AND TIME
VOLUME II: FORTRAN LISTING OF HELP

Prepared by
Systems, Science and Software
La Jolla, California

May 1971

This document has been approved for public release and sale;
its distribution is unlimited.

U.S. ARMY ABERDEEN RESEARCH AND DEVELOPMENT CENTER
BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as
an official Department of the Army position, unless
so designated by other authorized documents.

*The use of trade names or manufacturers' names in this report
does not constitute indorsement of any commercial product.*

BALLISTIC RESEARCH LABORATORIES

BRL CONTRACT REPORT NO. 39

3SR-350

MAY 1971

HELP, A MULTI-MATERIAL EULERIAN PROGRAM FOR COMPRESSIBLE FLUID
AND ELASTIC-PLASTIC FLOWS IN TWO SPACE DIMENSIONS AND TIME

VOLUME II: FORTRAN LISTING OF HELP

L. J. Hageman
J. M. Walsh

Prepared By:

SYSTEMS, SCIENCE AND SOFTWARE
La Jolla, California

This document has been approved for public release and sale;
its distribution is unlimited.

Contract No. DAAG 07-68-C-0931
and
RDT&E Project No. 1T650212D620

ABERDEEN PROVING GROUND, MARYLAND

BALLISTIC RESEARCH LABORATORIES

BRL CONTRACT REPORT NO. 39

3SR-350

LJHageman/JMWalsh
Aberdeen Proving Ground, Md.
May 1971

HELP, A MULTI-MATERIAL EULERIAN PROGRAM FOR COMPRESSIBLE FLUID
AND ELASTIC-PLASTIC FLOWS IN TWO SPACE DIMENSIONS AND TIME

VOLUME II: FORTRAN LISTING OF HELP

ABSTRACT

Volume II is a complete FORTRAN IV listing of the HELP program, the background description of which is contained in the preceding Volume I.

The language used in the following version of the HELP code is FORTRAN IV with one exception. To minimize the length of the listing, the common, dimension and equivalence statements are listed only once at the beginning and are assumed to be inserted in routines in place of the nonstandard statement, INCLUDE COMDIM. This program has been run on the CDC-6600 as well as the UNIVAC-1108.

```

1 C THIS SECTION IS INSERTED IN EACH ROUTINE IN PLACE OF "INCLUDE COMDIM"
2 C .....PH3 20
3 DIMENSION AMX(2500), AIX(2500), U(2500),
4 1 V(2500), P(2500), MFLAG(2500)
5 C
6 DIMENSION X(50), XX(52), DX(50),
7 1 DDY(52), TAU(50), PROP(50),
8 2 CRAD(50), PR(50)
9 C
10 DIMENSION Y(100), YY(102), DY(100),
11 1 DDY(102), FLEFT(100), YAMC(100),
12 2 SIGC(100), GAMC(100), UL(200),
13 3 PL(200)
14 C
15 DIMENSION SNB(50), STB(50), UK(50,3),
16 1 VK(50,3), RHOC(50,3), SZZ(50,3),
17 2 SRR(50,3), SRZ(50,3), EZZ(50,3),
18 3 ERR(50,3), ERZ(50,3)
19 C
20 DIMENSION FRACFP(4,250), FRACRT(4,250), TX(4,400),
21 1 TY(4,400), PRS(3,3), DNS(3,3),
22 2 PACX(3,10), PACY(3,10), MPACK(3),
23 3 MPAC(3), RHOIN(3), SSIEN(3),
24 4 UUR(3), VVA(3), CSQR(3),
25 5 VOL(3), WSQR(3), DELP(3),
26 6 AMDM(3), RMU(3), CZERO(3),
27 7 STEZ(3), STK1(3), STK2(3),
28 8 SAMMY(3), SDELM(3), SDELER(3),
29 9 SDELEB(3), SDELET(3), SGAMC(3,100),
30 * SSIGC(3,100), NMP(4)
31 C
32 DIMENSION VALUE(41), PK(15)

```

*** 2-STORAGE EQUIVALENCES

PH3 560

```

33 C
34 C
35 C
36 COMMON Z(1) ,CYCLE ,DT ,NUMSP ,NPRELP ,NDUMP,
37 1 ICSTOP ,PIDY ,TOPMU ,RTMU ,UN11 ,NUMREZ ,ETH ,
38 2 KUNITR ,IPR ,PRCNT ,KUNITW ,XMAX ,NZ ,NREZ ,
39 3 IGM ,UN22 ,UN23 ,DMIN ,UN25 ,DTNA ,CVIS ,
40 4 UN28 ,UN29 ,NC ,UN31 ,NRC ,IMAX ,IMAXA ,
41 5 JMAX ,JMAXA ,KMAX ,KMAXA ,BOTM ,BOTHV ,NUMSPT ,
42 6 MAPS ,NUMSCA ,PRLIM ,PRDEL ,PRFACT ,I1 ,I2 ,
43 7 IPCYCL ,TSTOP ,WFLAGF ,WFLAGL ,UN53 ,IVARDY ,VT ,
44 8 N6 ,RTH ,RTHV ,UN59 ,N10 ,N11 ,GAMMA ,
45 9 TOPM ,BOTMU ,UN65 ,TOPMV ,NSIDES ,NHAT ,CYCHX ,
46 COMMON CYCPH3 ,REZFCT ,INTRAC ,NMCLS ,BROUND ,UN75 ,
47 1 ECK ,NECYCL ,NTPMX ,GLUED ,UVMAX ,HTCC ,UN82 ,
48 2 IVARDX ,T ,EMIN ,PMIN ,INTER ,I ,J ,
49 3 K ,M ,N ,UN93 ,UN94 ,REZ ,NODUMP ,
50 4 UN97 ,UN98 ,UN99 ,EVAPH ,EVAPEN ,EVAPMU ,EVAPMV ,
51 5 EZPH2 ,UN105 ,UN106 ,UN107 ,UN108 ,UN109 ,ROEPS ,
52 6 UN111 ,UN112 ,FINAL ,UN114 ,M88B ,M88 ,UN117 ,
53 7 NADD ,MINX ,MAXX ,MINY ,MAXY ,IEXTX ,JEXTY ,

```

```

54      8 UN125 ,UN126 ,SS1 ,SS2 ,UMIN ,SS4 ,PRIME,
55      9 EOR ,EOT ,EOB ,EMOR ,OXF ,DYF ,UN138
56      COMMON STAB ,XIENRG ,XENRG ,XTENRG ,UN143 ,DTMIN ,
57      1 UN145 ,EMOT ,JCENTR ,RADIUS ,BBAR ,EMOB
58      C
59      C      *** DIMENSIONED ARRAYS
60      C
61      1      PK,
62      2      YY,      XX,
63      3      DDX,      CDY,
64      4      AMX,      AIX,      U,      V,      P,      MFLAG,
65      5      TAU,      UL,      PL,
66      6      TX,      TY,      NMP,      RHOIN,
67      7      CZERO,      STK1,      STK2,      STEZ,      RMU,
68      8      AMDM,      SSIEN,      UUR,      VVA,      MPAC,      MPACK,
69      9      PACX,      PACY
70      C
71      C      *** NON-DIMENSIONED VARIABLES
72      C
73      COMMON CYC,      ENERGY,      ERDUMP,      I3,      IFS1,      IFS2,
74      1      KA,      KR,      MA, MFK,      MO,      MR,      NERR,
75      2      NK,NVOID,      NPRINT,      NR,      PIDTS,      PRESUR,      RHOW,
76      3      SDT,      SUM,      T*OPI,      URR,      VABOVE,      WS,
77      4      WSA,      WSB,      WSC,      WSX,      WSY,      LAST
78      C
79      C
80      COMMON /MXCELL/      SIE(3,250),      XMASS(3,250),      RHO(4,250),
81      1      SAMPY(3,250),      SAMMP(3,250),      PLW(3),
82      2      RHOZ(30),      CNAUT(30),      MAT(30)
83      C
84      COMMON /ELPL/      STRSZZ(2500),      STRSRR(2500),      STRSRZ(2500)
85      C
86      COMMON /TRACRS/      XP(1000),      YP(1000)
87      C
88      C      *** NOTE THESE MISCELLANEOUS VARIABLES ARE DIMENSIONED
89      C      FOR A GRID WITH 3 MATERIAL PACKAGES AND 50 COLUMNS
90      C      AND 100 ROWS.
91      C
92      COMMON/MISC/      VL(3),XMAS(2,3),SSIE(2,3),XRH(2,3),PQ(4,8),
93      1      MFGREZ(150),REZAMX(150),REZAI(150),
94      2      REZAMS(3,150), REZSIE(3,150), REZRHO(4,150)
95      C
96      EQUIVALENCE (Z(1),PROB)
97      C
98      C      *** THE FOLLOWING EQUIVALENCES MAKE AVAILABLE
99      C      X(0), Y(0), DX(0), DY(0)
100     C
101     EQUIVALENCE (XX(2), X(1)), (YY(2), Y(1))
102     EQUIVALENCE (DDX(2), DX(1)), (DDY(2), DY(1))
103     C
104     C      *** SPECIAL EQUIVALENCES FOR CDT ONLY
105     C
106     EQUIVALENCE (PL,DELP), (PL(6),DNS), (PL(21),PRS), (PL(36),CSQR),
107     1 (PL(42),VOL), (PL(48),WSQR)

```

```

108 C
109 C
110 C      *** SPECIAL EQUIVALENCES FOR EDIT AND MAP
111 C
112 EQUVALENCE (UL,PROP), (VALUE,PR,CRAD)
113 EQUVALENCE (JCENTR,JPROJ)
114 C
115 C      *** SPECIAL EQUIVALENCES FOR PH3 ONLY
116 C
117 EQUVALENCE (P,SNB), (P(51),STB), (P(101),RHOC),
118 | (P(251),UK,ERR), (P(401),VK,EZZ), (P(551),ERZ),
119 2 (P(701),SRZ), (P(851),SRR), (P(1001),SZZ)
120 C
121 C      *** SPECIAL EQUIVALENCES FOR INFACE
122 C
123 EQUVALENCE (P,FRACFP), (P(1001),FRACRT)
124 C
125 C      *** SPECIAL EQUIVALENCES FOR PH2 ONLY
126 C
127 EQUVALENCE (UL,FLEFT), (UL(101),YAMC),
128 1 (PL,GAMC), (PL(101),SIGC),
129 2 (P,SDELET), (P(4),SDELER),
130 3 (P(7),SDELEB), (P(10),SDELM),
131 4 (P(13),SAMMY), (P(16),SGAMC),
132 5 (P(316),SSIGC)
133 C
134 C .....
135 C
136 C      END OF COMMON
137 C
138 C .....
139 C
140 C
141 END
142 BLOCK DATA
143 C      *** DEFINES NORMAL DENSITY AND SOUND SPEED COEFFICIENT
144 C      FOR THE 19 MATERIALS LISTED IN EQST.
145 COMMON /MXCELL/ SIE(3,250), XMASS(3,250), RHO(4,250),
146 1 SAMPY(3,250), SAMMP(3,250), PLW(3),
147 2 RHOZ(30), CNAUT(30), MAT(30)
148 DATA (RHOZ(K),K=1,19)
149 1 /19.17,8.9,7.8,2.79,1.8,4.5,8.9,10.2,11.7,
150 2 11.3, .9, 2.7, 2.7, 1.97, 1.7, 2.3, 2.8, 2.7, 2.2/
151 DATA (CNAUT(K),K=1,19)
152 1 /4.01E5, 3.95E5, 4.03E5, 5.27E5, 8.06E5, 4.78E5,
153 2 4.63E5, 5.15E5, 2.13E5, 2.03E5, 2.89E5,
154 3 2.58E5, 2.58E5, 2.24E5, 1.63E5, 3.49E5, 5.51E5,
155 4 3.85E5, 3.37E5/
156 END
157 SUBROUTINE ADPTR
158 C      *** ADD MATERIAL TRACER PARTICLES IN A SPECIFIED REGION.
159 INCLUDE COMDIM
160 C
161 C ..... FIND THE AREA IN WHICH TRACERS ARE TO BE ADDED.

```



```

162 C
163 IF(NADD.EQ.0)RETURN
164 IF(MINX.LT.1)MINX=1
165 IF(MINX.GT.IMAX)MINX=IMAX
166 IF(MINY.LT.1)MINY=1
167 IF(MINY.GT.JMAX)MINY=JMAX
168 IF(MAXX.GT.IMAX)MAXX=IMAX
169 IF(MAXY.GT.JMAX)MAXY=JMAX
170 WRITE(6,BINADD,MINX,MAXX,MINY,MAXY
171 8 FORMAT(//34H SUBROUTINE ADDTCR HAS BEEN CALLED,/,8H NADD=,
172 114,7H MINX=,13,7H MAXX=,13,7H MINY=,13,7H MAXY=,13,/)
173 XMIN=X(MINX-1)
174 XXMA=X(MAXX)
175 YMIN=Y(MINY-1)
176 YMAX=Y(MAXY)
177 C
178 C ***** DELETE ANY DUPLICATED TRACERS.
179 C
180 DO 6 L=1,NVOID
181 NP=NMP(L)
182 IF(NP.LE.1)GO TO 6
183 N=1
184 1 TXTMP=TX(L,N)
185 TYTMP=TY(L,N)
186 2 N=N+1
187 IF(ABS(TXTMP-TX(L,N)).LE.0.0.AND.ABS(TYTMP-TY(L,N)).LE.0.0)GO TO 3
188 IF(N.LT.NP)GO TO 1
189 GO TO 6
190 3 NP=NP-1
191 NMP(L)=NP
192 IF(N.LE.NP)GO TO 4
193 TX(L,N)=0.
194 TY(L,N)=0.
195 GO TO 6
196 4 DO 5 M=N,NP
197 TX(L,M)=TX(L,M+1)
198 5 TY(L,M)=TY(L,M+1)
199 TX(L,NP+1)=0.
200 TY(L,NP+1)=0.
201 IF(N.LT.NP)GO TO 2
202 6 CONTINUE
203 C
204 C ***** NADD.LT.0 INTERPOLATE USING CELL COORDINATES.
205 C ***** NADD.GT.0 INTERPOLATE USING PHYSICAL COORDINATES.
206 C
207 ITFLAG=0
208 IF(NADD.LT.0)ITFLAG=1
209 IF(IVARX.EQ.0.AND.IVARDY.EQ.0)ITFLAG=1
210 NAD=IABS(NADD)
211 DO 110 LPAS=1,2
212 DO 100 NN1=1,NVOID
213 NP=NMP(NN1)
214 IF(NP.LE.0)GO TO 100
215 TXSAV=TX(NN1,1)

```

```

216 ..... TYSAV=TY(INN1,1)
217 ..... MMSAV=1
218 ..... K1=0
219 ..... MPLUS=0
220 ..... DO 90 I315=1,NP
221 ..... MN1=I315
222 ..... IF(INN1,EQ,NVOID) MN1=NP-MN1+1
223 ..... MP=MN1+MPLUS
224 ..... K2=K1
225 ..... K1=0
226 ..... C
227 ..... C ..... DETERMINE WHICH CELL TRACERS LIE IN.
228 ..... C
229 ..... ITP=INT(TX(INN1,MP))
230 ..... JTP=INT(TY(INN1,MP))
231 ..... C
232 ..... C ..... IF THIS CELL IS NOT IN THE GRID, GO TO 80.
233 ..... C
234 ..... IF(ITP.GT.IMAX.OR.JTP.GT.JMAX)GO TO 80
235 ..... C
236 ..... C ..... FIND THE PHYSICAL CO-ORDINATES OF THE TRACER.
237 ..... C
238 ..... XTX=X(ITP)+(TX(INN1,MP)-FLOAT(ITP))*DX(ITP+1)
239 ..... YTY=Y(JTP)+(TY(INN1,MP)-FLOAT(JTP))*DY(JTP+1)
240 ..... TXT=TX(INN1,MP)
241 ..... YTY=TY(INN1,MP)
242 ..... C
243 ..... C ..... IF THE TRACER IS NOT IN THE AREA IN WHICH TRACERS
244 ..... C ..... ARE TO BE ADDED, GO TO 80.
245 ..... C
246 ..... IF(XTX.LT.XMIN.OR.XTX.GE.XXMA)GO TO 80
247 ..... IF(YTY.LT.YMIN.OR.YTY.GE.YMAX)GO TO 80
248 ..... K1=1
249 ..... C
250 ..... C ..... IF THE PREVIOUS TRACER WAS NOT IN THE AREA, GO TO 70.
251 ..... IF(K2.EQ.0) GO TO 70
252 ..... C
253 ..... C
254 ..... C ..... IF TWO CONSECUTIVE TRACERS ARE BOTH ON THE
255 ..... C ..... AXIS, DO NOT INTERPOLATE BETWEEN THEM.
256 ..... C
257 ..... IF((XTX.LE.0..OR.YTY.LE.0.) .AND. (TXT1.LE.0..OR.YTY1.LE.0.))GOTO70
258 ..... NADD=INT(FLOAT(NTRACR)*SQRT((TXT1-TXT)**2+(YTY1-YTY)**2))
259 ..... NADD=MIN0(NADD,NAD)
260 ..... IF(NADD.LE.0)GO TO 70
261 ..... IF(LPAS.EQ.2)*PLUS=MPLUS+NADD
262 ..... NNMP=NNMP(INN1)+NADD
263 ..... C
264 ..... C ..... IF THERE IS NO MORE ROOM FOR NEW TRACERS, GO TO 120.
265 ..... C
266 ..... IF(NNMP.GT.NTPMX)GO TO 120
267 ..... NNMP(INN1)=NNMP
268 ..... C
269 ..... C ..... SHIFT ALL TRACERS WHICH FOLLOW UP IN THE ARRAY

```

```

270 C ***** BY NADD PLACES.
271 C
272     IF(LPAS.NE.2)GO TO 70
273     DO 10 I11=MM1,NP
274         MT1=NMP(NN1)+MM1-I11
275         MT2=MT1-NADD
276         TX(NN1,MT1)=TX(NN1,MT2)
277     10 TY(NN1,MT1)=TY(NN1,MT2)
278     DO 60 I11=1,NADD
279         MT1=I11+MP-1
280         IF(I11FLAG.EQ.1)GO TO 55
281 C
282 C ***** FIND THE PHYSICAL CO-ORDINATES OF THE NEW TRACERS
283 C ***** WHICH ARE TO BE ADDED.
284 C
285         XADD=XTX1+FLOAT(I111)*(XTX-XTX1)/FLOAT(NADD+1)
286         YADD=YTY1+FLOAT(I111)*(YTY-YTY1)/FLOAT(NADD+1)
287 C
288 C ***** FIND WHICH CELL THE NEW TRACERS WILL LIE IN.
289 C
290         DO 20 I=1,IMAX
291             IF(XADD.LT.>(I))GO TO 30
292             20 CONTINUE
293             30 DO 40 J=1,JMAX
294                 IF(YADD.LT.>(J))GO TO 50
295                 40 CONTINUE
296 C
297 C ***** DETERMINE TX AND TY FOR THE NEW TRACERS.
298 C
299         50 TX(NN1,MT1)=FLOAT(I-1)+(XADD-X(I-1))/DX(I)
300             TY(NN1,MT1)=FLOAT(J-1)+(YADD-Y(J-1))/DY(J)
301             GO TO 60
302 C
303 C ***** FIND CELL COORDINATES OF NEW TRACERS
304 C ***** IF INTERPOLATING BETWEEN CELL COORDINATES.
305 C
306         55 TX(NN1,MT1)=TXT1+FLOAT(I111)*(TXT-TXT1)/FLOAT(NADD+1)
307             TY(NN1,MT1)=TYT1+FLOAT(I111)*(TYT-TYT1)/FLOAT(NADD+1)
308             60 CONTINUE
309             70 XTX1=XTX
310                 YTY1=YTY
311                 TXT1=TX
312                 TYT1=TY
313             80 CONTINUE
314                 IF(MM1.EQ.MMSAV)GO TO 90
315                 IF(ABS(TXT-TXSAV).LE.0..AND.ABS(TYT-TYSAV).LE.0.)KI=-1
316                 IF(KI.GE.0)GO TO 90
317                 KI=0
318                 IF(MM1.GE.NP)GO TO 90
319 C
320 C ***** THIS WAS THE LAST POINT OF A SUBPACKAGE. FIND THE
321 C ***** COORDINATES OF THE FIRST POINT OF THE NEXT SUBPACKAGE.
322 C
323         MMSAV=MM1+1

```

```

324 ..... MT1=MMSAV+MPLUS
325 ..... TXSAV=TX(NN1,MT1)
326 ..... TYSAV=TY(NN1,MT1)
327 ..... 90 CONTINUE
328 ..... IF(LPAS.NE.2)NMP(NN1)=NP
329 ..... 100 CONTINUE
330 ..... 110 CONTINUE
331 ..... NADD=0
332 ..... RETURN
333 ..... 120 WRITE(6,130)NN1,NN1,NNMP,NTPMX
334 ..... 130 FORMAT(1X,47HERROR IN ADDTCR. NUMBER OF TRACERS IN PACKAGE ,13,
335 ..... 122H EXCEEDED VTPMX. NMP1,13,3H) =,14,9H NTPMX =,14)
336 ..... NMP(NN1)=NP
337 ..... RETURN
338 ..... END
339 ..... SUBROUTINE CARDS
340 ..... C ..... ** READS INPUT FROM CARDS INTO BLANK COMMON.
341 ..... DIMENSION TABLE(1),CARD(7),LABLE(1)
342 ..... COMMON TABLE
343 ..... EQUIVALENCE(TABLE(1),LABLE(1))
344 ..... INPERR=0
345 ..... *RITE (6,80)
346 ..... 10 READ (5,90) IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
347 ..... WRITE (6,100) IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
348 ..... IF (NUMWPC.LT.1) GO TO 50
349 ..... IF (LOC.LT.1) GO TO 70
350 ..... DO 30 I=1,NUMWPC
351 ..... J=LOC+I-1
352 ..... IF (IEND.NE.2) GO TO 20
353 ..... LABLE(J)=IFIX(CARD(I))
354 ..... GO TO 30
355 ..... 20 TABLE(J)=CARD(I)
356 ..... 30 CONTINUE
357 ..... 40 IF (IEND.NE.1) GO TO 10
358 ..... IF (INPERR.EQ.0) RETURN
359 ..... STOP
360 ..... 50 IF (LOC.NE.0) GO TO 70
361 ..... DO 60 J=1,7
362 ..... IF (ICARD(J).NE.0.) GO TO 70
363 ..... 60 CONTINUE
364 ..... WRITE (6,120)
365 ..... GO TO 40
366 ..... 70 WRITE (6,110)
367 ..... INPERR=1
368 ..... GO TO 40
369 ..... C ..... FORMATS
370 ..... C
371 ..... 80 FORMAT (//18H INPUT CARDS//)
372 ..... 90 FORMAT (11,15,11,0P7E9.4)
373 ..... 100 FORMAT (11H 14,17,13,1P7E14.6)
374 ..... 110 FORMAT (//42H *** ERROR ON PRECEDING DATA CARD ****//)
375 ..... 120 FORMAT (//18H BLANK CARD ****//)
376 ..... END
377 ..... SUBROUTINE CDT

```

```

378 ----- INCLUDE CONDIM
379 C
380 C
381 C          ***CHECK COURANT CONDITION AND PARTICLE VELOCITY.
382 C          ***RECORD I AND J OF ZONE WHERE DT IS CONTROLLED.
383 C          ***FIRST CALCULATE PRESSURES FROM EQ. OF ST.
384 C
385          DO 5 K=1,KMAX
386          PI(K) = 0.
387          5 CONTINUE
388 C
389          6 TRIAL=0.
390          SRATIO= 1.E+38
391 C          ***WSC WILL BE MAXIMUM U OR V
392          WSC=0.
393          DO 60 I=1,11
394          K=I+1
395          DO 60 J=1,12
396          VCELL = TAU(I)*DY(J)
397          RHOW = AMX(K)/VCELL
398          IF(RHOW.LE.0.) GO TO 60
399          MFK=MFLAG(K)
400          IF(MFK.GT.100) GO TO 7
401          N=MAT(MFK)
402          4 ENERGY = AIX(K)
403          IF(N.EQ.20) ENERGY=AMAX1(EHIN,AIX(K))
404          WS=1.0
405          CALL EQST
406          PI(K)=PRESUR
407          IF(MFK.LT.100) GO TO 30
408          3 IF(RHOW.GT.RHO(LSAVE,M)) RHO(LSAVE,M)= RHOW
409          IF(PI(K).LT.0.) PI(K)=0.
410          GO TO 30
411 C          *** MIXED CELL. ITERATE TO FIND CELL PRESSURE.
412          7 M=MFK-100
413          DO 10 L=1,NMAT
414          DO 8 N=1,3
415          DNS(N,L) = 0.
416          PRS(N,L) = 0.
417          8 CONTINUE
418          CSQR(L)=0.
419          WSGR(L)=0.
420          10 CONTINUE
421 C          *** TO BEGIN INTERATION COMPUTE TWO PRESSURES FOR EACH
422 C          MATERIAL IN THE CELL USING (1) DENSITIES CALCULATED ON
423 C          LAST CYCLE AND (2) THOSE DENSITIES INCREASED BY ONE
424 C          PERCENT.
425          ITC=0
426          NM=0
427          DO 11 L=1,NMAT
428          IF(XMASS(L,M).LE.0. .OR. RHO(L,M).LE.0.) GO TO 11
429          DNS(2,L) = RHO(L,M)
430          LSAVE=L
431          M=MAT(L)

```

```

432      NM=NM+1
433      11 CONTINUE
434      C      *** IF MIXED CELL HAS ONLY ONE MATERIAL (NM=1),
435      C      COMPUTE PRESSURE AS IF CELL WERE PURE AND
436      C      ADJUST MATERIAL DENSITY ACCORDING TO SIGN OF PRESSURE
437      IF(NM-1) 60,115,12
438      115 CONTINUE
439      GO TO 4
440      C
441      12 DO 13 L=1,NMAT
442      IF(XMASS(L,M).LE.0.) GO TO 13
443      RHO=DNS(2,L)
444      N=MAT(L)
445      WS=1.0
446      IF(RHOW/RHOZ(N).LT.1.0) WS=-1.0
447      ENERGY=SIE(L,M)
448      IF(N.EQ.20) ENERGY=AMAX1(EMIN,SIE(L,M))
449      CALL EQST
450      PRS(2,L) = PRESUR
451      C      *** ALTER CURRENT DENSITY BY 1 PERCENT, COMPUTE ANOTHER
452      C      PRESSURE POINT
453      DNS(3,L) = RHO(L,M)*1.01
454      RHO=DNS(3,L)
455      WS=1.0
456      IF(RHOW/RHOZ(N).LT.1.0) WS=-1.0
457      CALL EQST
458      PRS(3,L) = PRESUR
459      13 CONTINUE
460      C      *** DETERMINE CSQR CORRESPONDING TO TWO PTS JUST DETERMINED
461      VSUM=0.
462      NFLAG=0
463      133 WSUM=0.
464      DO 14 L=1,NMAT
465      IF(XMASS(L,M).LE.0.) GO TO 14
466      IF(NFLAG.GT.0) GO TO 135
467      CSQR(L) = (PRS(2,L)-PRS(3,L))/(DNS(2,L)-DNS(3,L))
468      VOL(L) = XMASS(L,M)/RHO(L,M)
469      VSUM=VSUM+VOL(L)
470      135 CONTINUE
471      WSQR(L) = 1.0/(RHO(L,M)**2 * CSQR(L))
472      WSUM = WSUM+XMASS(L,M)*WSQR(L)
473      14 CONTINUE
474      C
475      DP=(VSUM-VCCELL)/WSUM
476      C      *** NORMALIZE DENSITIES, COMPUTE THIRD PRESSURE POINT.
477      DO 145 L=1,NMAT
478      IF(XMASS(L,M).LE.0.) GO TO 145
479      DV = -DP*WSQR(L)
480      VL(L)=1.0/RHO(L,M)+DV
481      IF(VL(L).GT.0.) GO TO 145
482      CSQR(L)=CSQR(L)*2.
483      NFLAG=NFLAG+1
484      IF(NFLAG.LE.IPR) GO TO 133
485      WRITE(6,300) 1,J,ITC,PAV,(XMASS(N,M),DNS(2,N),SIE(N,M),PRS(2,N),

```

```

486      1      CSQR(N),N=1,NMAT)
487      WRITE(6,295)
488      NK=145
489      GO TO 180
490      C
491      145 CONTINUE
492      C
493      DO 15 L=1,NMAT
494      IF(XMASS(L,M).LE.0.) GO TO 15
495      DNS(1,L)=1.0/VL(L)
496      RHOH = DNS(1,L)
497      N=MAT(L)
498      WS=1.0
499      IF(RHOH/RHOZ(N).LT.1.0) WS=-1.0
500      ENERGY = SIE(L,M)
501      IF(N.EQ.20) ENERGY=AMAX1(EMIN,SIE(L,M))
502      CALL EQST
503      PRS(1,L) = PRESUR
504      15 CONTINUE
505      IF(INTER.EQ.0) GO TO 16
506      WRITE(6,301) ((LL,DNS(LL,L),PRS(LL,L),LL=1,3),L=1,NMAT)
507      C      *** BEGIN ITERATION - COMPUTE CSQR USING LAST POINT AND
508      C      CLOSEST OF OTHER TWO.
509      16 ITC=ITC+1
510      NFLAG=0
511      C      *** IF P OF ALL MATERIAL .LT. PMIN, SKIP OUT AND
512      C      ADJUST DENSITIES BY A CONSTANT FACTOR TO EXACTLY FILL
513      C      THE CELL.
514      DO 165 L=1,NMAT
515      IF(PRS(1,L).GT.PMIN .AND. XMASS(L,M).GT.0.) GO TO 168
516      165 CONTINUE
517      GO TO 28
518      C
519      168 CONTINUE
520      WSUM=0.
521      PSUM=0.
522      DO 18 L=1,NMAT
523      IF(XMASS(L,M).LE.0.) GO TO 18
524      IF(NFLAG.GT.0) GO TO 175
525      WSA=ABS(PRS(1,L)-PRS(3,L))
526      WSB=ABS(PRS(1,L)-PRS(2,L))
527      IF(WSB.LT.WSA) GO TO 17
528      DNS(2,L) = DNS(3,L)
529      PRS(2,L) = PRS(3,L)
530      17 WSX=PRS(2,L)-PRS(1,L)
531      WSY=DNS(2,L)-DNS(1,L)
532      IF(ABS(WSX).LE.1.0E-05*ABS(PRS(2,L)) .OR.
533      1 ABS(WSY).LE.1.0E-05*ABS(DNS(2,L)) ) GO TO 175
534      CSQR(L)=WSX/WSY
535      175 CONTINUE
536      WSQR(L)=1.0/(DNS(1,L)**2 * CSQR(L))
537      WSUM=XSUM+XMASS(L,M)*WSQR(L)
538      PSUM=PSUM+PRS(1,L)*XMASS(L,M)*WSQR(L)
539      18 CONTINUE

```

```

540 C *** COMPUTE WEIGHTED AVERAGE PRESSURE, NEW DENSITIES FOR
541 C NEXT PRESSURE POINT
542 PAV=PSUM/WSUM
543 DO 19 L=1,NMAT
544 IF(XMASS(L,M).LE.0.) GO TO 19
545 DV=WSQR(L)*(PRS(1,L)-PAV)
546 VL(L)=1.0/DNS(1,L)+DV
547 IF(VL(L).GT.0.) GO TO 19
548 CSQR(L)=CSQR(L)+2.
549 NFLAG=NFLAG+1
550 IF(NFLAG.LE.IPR) GO TO 168
551 WRITE(6,300) I,J,ITC,PAV,(XMASS(N,M),DNS(1,N),SIE(N,M),PRS(1,N),
552 1 CSQR(N),N=1,NMAT)
553 WRITE(6,295)
554 NK=19
555 GO TO 180
556 C
557 19 CONTINUE
558 C
559 DO 20 L=1,NMAT
560 IF(XMASS(L,M).LE.0.) GO TO 20
561 DNS(3,L) = DNS(2,L)
562 DNS(2,L) = DNS(1,L)
563 PRS(3,L) = PRS(2,L)
564 PRS(2,L) = PRS(1,L)
565 C
566 DNS(1,L)=1.0/VL(L)
567 RHO* = DNS(1,L)
568 N=MAT(L)
569 WS=1.0
570 IF(RHO*/RHOZ(N).LT.1.0) WS=-1.0
571 ENERGY=SIE(L,M)
572 IF(N.EQ.20) ENERGY=AMAX1(EMIN,SIE(1,M))
573 CALL EQST
574 PRS(1,L) = PRESUR
575 20 CONTINUE
576 IF(INTER.EQ.0) GO TO 21
577 WRITE(6,300) I,J,ITC,PAV,(XMASS(L,M),DNS(1,L),SIE(L,M),PRS(1,L),
578 1 CSQR(L),L=1,NMAT)
579 C *** TEST IF ITERATION COMPLETE
580 21 PSUM=0.
581 DO 22 L=1,NMAT
582 IF(XMASS(L,M).LE.0.) GO TO 22
583 IF(ABS((PAV-PRS(1,L))/PAV).GT.PRCNT) GO TO 24
584 IF(PRS(1,L).GT.PMIN) PSUM=1.
585 22 CONTINUE
586 P(K)=0.
587 IF(PSUM.GT.0.) P(K)=PAV
588 GO TO 28
589 C
590 24 IF(ITC.LT.IPR) GO TO 16
591 WRITE(6,295)
592 WRITE(6,300) I,J,ITC,PAV,(XMASS(L,M),DNS(1,L),SIE(L,M),PRS(1,L),
593 1 CSQR(L),L=1,NMAT)

```



```

594      NK=28
595      GO TO 180
596      C
597      28 CONTINUE
598      C      *** STORE NEW DENSITIES.
599      DO 29 L=1,NMAT
600      IF(XMASS(L,M).LE.0.) GO TO 29
601      RHO(L,M) = DNS(I,L)
602      29 CONTINUE
603      GO TO 30
604      C
605      C
606      30 CONTINUE
607      IF (ABS(P(K)).LT.PMIN) P(K)=0.
608      C
609      C      *** CHECK FOR NEGATIVE PRESSURE.
610      IF(P(K).LT.0. .AND. ((I.EQ.IMAX .OR. J.EQ.JMAX
611      .OR. (CVIS.LT.0..AND.J.EQ.1))) P(K)=0.
612      WS=0.
613      WSA=0.
614      C      *** CHECK IF CELL IS MIXED.
615      IF(MFK.GT.100) GO TO 33
616      C
617      C      ***** PURE CELL *****
618      C
619      C      *** DETERMINE IF MATERIAL IS POLYTROPIC GAS.
620      IF(N=20) 32,31,32
621      31 WSA= SQRT(GAMMA*ABS(P(K))/RHO)
622      GO TO 38
623      32 WS= CNAUT(N)
624      GO TO 37
625      C
626      C
627      C      ***** MIXED CELL *****
628      33 M=MFK-100.
629      XM=0.
630      CN=C.
631      DO 35 N=1,NMAT
632      XMS= XMASS(N,M)
633      IF(XMS.LE.0.) GO TO 35
634      MN=MAT(N)
635      IF(MN.NE.20) GO TO 34
636      C      *** POLYTROPIC GAS
637      C      *****NOTE-- HERE WE ASSUME THERE IS ONLY ONE SUCH GAS
638      C      IN THE ENTIRE GRID.
639      WSA= SQRT(GAMMA*ABS(P(K))/RHO(N,M))
640      GO TO 35
641      C      *** NON-POLYTROPIC MATERIAL.
642      34 XM=XM+XMS
643      CN=CN + CNAUT(MN)*XMS
644      35 CONTINUE
645      IF(XM.LE.0.) GO TO 38
646      WS=CN/XM
647      37 IF(P(K).GT.0.) WS=WS + RBAR*SQRT(P(K))

```

```

648      38 WS= AMAXI(WS,WSA)
649      C      *** WS IS SOUND SPEED OF CELL K.
650      C      *** WSA IS MAXIMUM OF RADIAL AND AXIAL VELOCITY OF CELL K.
651      C      *** WSC STORES MAXIMUM VELOCITY OF CELLS USED TO DETERMINE
652      C      DT. PRINTED AS MAXUV.
653      40 WSB=AMAXI(ABS(U(K)),ABS(V(K)))
654      WSC=AMAXI(WSC,WSB)
655      WS=WS+WSB
656      C      *** TRIAL STORES SUM OF VELOCITY AND SOUND SPEED USED
657      C      TO DETERMINE DT. PRINTED AS MAXCUV.
658      IF (WS.LE.TRIAL) GO TO 50
659      TRIAL=WS
660      50 IF (WS.LE.0.) GO TO 60
661      DXYMIN=AMIN1(DX(I),DY(J))
662      RATIO=DXYMIN/WS
663      IF (RATIO.GT.SRATIO) GO TO 60
664      C      *** I AND J OF CELL CONTROLLING DT STORED IN N10 AND N11
665      C      FOR PRINTOUT.
666      N10=I
667      N11=J
668      C      *** SRATIO IS SMALLEST VALUE CALCULATED FOR RATIO.
669      SRATIO=RATIO
670      C
671      C      ***END OF I, J LOOP
672      60 K=K+IMAX
673      UVMAX=WSC
674      C      *** SET FREE SURFACE NEG. PRESSURES TO ZERO.
675      DO 64 K=2,KMAX
676      MFK=MFLAG(K)
677      IF(MFK.LT.100) GO TO 64
678      M=MFK-100
679      IF(RHO(INVOID,M).LE.0.) GO TO 64
680      IF(P(K),LT.0.) P(K)=0.
681      64 CONTINUE
682      C
683      C      *** IF TRIAL.LE.0. THERE IS PROBABLY AN ERROR IN THE INPUT
684      C      PARAMETERS FOR THE INITIAL VELOCITY, ENERGY OR DENSITY
685      C      OF THE PACKAGES, OR IN THE X,Y,DX,DY ARRAYS.
686      65 IF (TRIAL.LE.0.) GO TO 170
687      C      *** IF FINAL.EQ.0.USE STAB FOR VALUE OF STABILITY FRACTION
688      C      IF FINAL.GT.0.USE A GEOMETRIC PROGRESSION WITH STAB
689      C      AS THE INITIAL VALUE AND FINAL AS THE FINAL VALUE.
690      IF (FINAL.EQ.0.) GO TO 70
691      STAB=2.*STAB
692      STAB=AMIN1(STAB,FINAL)
693      70 DT=STAB*SRATIO
694      IF(DT.LE.0.) GO TO 150
695      IF (STAB.LT.FINAL) GO TO 80
696      C      *** AFTER STAB,GE,FINAL CHECK ON SIZE OF DT. DTMIN IS AN
697      C      INPUT PARAMETER AND CAN BE SET TO 0.
698      75 IF (DT.LE.DTHIN) GO TO 150
699      80 CONTINUE
700      C
701      C      *** IS CONTROL-CELL ISOLATED

```

```

702 K=(N11-1)*IMAX+N10+1
703 WS=0.
704 IF (N10.GT.1) WS=AMX(K-1)
705 IF (N10.LT.IMAX) WS=AMX(K+1)+WS
706 IF (N11.GT.1) WS=AMX(K-IMAX)+WS
707 IF (N11.LT.JMAX) WS=AMX(K+IMAX)+WS
708 IF (WS.GT.0.) GO TO 90
709 C *** ISOLATED, SO DESTROY IT.
710 WS=(AIX(K)+(U(K)**2+V(K)**2)*.5)*AMX(K)
711 EVAPM=EVAPM+AMX(K)
712 EVAPEN=EVAPEN+WS
713 ETH=ETH-WS
714 EVAPMU=EVAPMU+AMX(K)*U(K)
715 EVAPMV=EVAPMV+AMX(K)*V(K)
716 WRITE (6,290) N10,N11,T,DT,TRIAL,WSC,UMIN,PMIN
717 AMX(K)=0.
718 AIX(K)=0.
719 P(K)=0.
720 U(K)=0.
721 V(K)=0.
722 MFK=MFLAG(K)
723 IF (MFK.LT.100) GO TO 6
724 M=MFK-100
725 DO 82 N=1,NMAT
726 XMASS(N,M)=0.
727 SIE(N,M)=0.
728 82 CONTINUE
729 C *** RECALCULATE DT.
730 GO TO 6
731 C *** INCREMENT TIME AND CYCLE.
732 90 T=T+DTNA
733 95 IF (T.LT.C.) GO TO 160
734 NC=NC+1
735 CYCLE=NC
736
737 C *** RESET NPRINT, NPRINT=1 ON PRINT CYCLES.
738 NPRINT=0
739 C *** DEFINE VELOCITY AND ENERGY CUTOFFS USED IN MAP AND PH2.
740 UMIN=TRIAL*ROEPS
741 SIEMIN=UMIN**2
742 DO 140 L=1,NMAT
743 N=MAT(L)
744 IF (N.EQ.20) GO TO 140
745 WS = RHOZ(N)*CNAUT(N)*UMIN
746 PMIN = AMINI(PMIN,*S)
747 140 CONTINUE
748 WRITE(6,280) NC
749 WRITE (6,290) N10,N11,T,DT,TRIAL,WSC,UMIN,PMIN
750 DTNA=DT
751 GO TO 190
752 C
753 C *** DT TOO SMALL
754 150 NK=75
755 GO TO 180

```

```

756 C          *** T IS NEGATIVE
757 160 NK=95
758      GO TO 180
759 C          *** DT WILL BE NEGATIVE OR ZERO.
760 170 NK=65
761      GO TO 180
762 180 NR=3
763      CALL ERROR
764 C
765      190 RETURN
766 C
767      280 FORMAT(1X,5HCYCLE,15)
768 290 FORMAT (/4H CDT,13,14,4H T=,1PE13.7,5H DT=,1PE13.7,9H MAXCUV=,1
769      1PE13.7,8H MAXUV=,1PE13.7,7H UMIN=,1PE13.7,7H PMIN=,1PE13.7)
770      295 FORMAT(/33H TROUBLE WITH PRESSURE ITERATION )
771      300 FORMAT(4H I = 13, 4H J = 13, 6H ITC = 13, 6H PAV = 1PE20.8/
772      210X, 10H MASS ,10X, 10H DENSITY ,10X,10H SIE ,10X,
773      3      10H PRESSURE,10X 10H CSQR /(1P5E20.8))
774      301 FORMAT(16,1P2E20.8)
775      END
776      SUBROUTINE COMPR(L)
777 C . . . . .
778 C      SUBROUTINE COMPR TAKES TWO CELLS L AND M, COMBINES THEM
779 C      INTO ONE CELL K AND ZEROS OUT THE OLD CELLS.
780 C . . . . .
781      INCLUDE COMDIM
782      MFL=MFLAG(L)
783      MFM=MFLAG(M)
784      MFLAG(M)=0
785      MFLAG(L)=0
786      WSA=AMX(L)+AMX(M)
787      IF(ABS(WSA).GT.0.)GO TO 100
788 C . . . . .
789 C      BOTH CELLS ARE EMPTY OF MASS.
790 C . . . . .
791      IF(MFM.EQ.0)GO TO 60
792 C . . . . .
793 C      ERROR FOUND IF CELL M IS NOT MIXED OR VOID.
794 C . . . . .
795      IF(MFM.LT.100)GO TO 80
796      IF(MFL.EQ.0)GO TO 50
797 C . . . . .
798 C      ERROR FOUND IF CELL L IS NOT MIXED OR VOID.
799 C . . . . .
800      IF(MFL.LT.100)GO TO 80
801      IF(MFL.GT.MFM)GO TO 10
802 C . . . . .
803 C      BOTH CELLS MIXED. ZERO OUT ONE AFTER TRANSFERING DENSITIES TO OTHER.
804 C . . . . .
805      MM=MFL-100
806      NN=MFM-100
807      MFLAG(K)=MFL
808      GO TO 20
809      10 MM=MFM-100

```

```

810      NN=MFL-100
811      MFLAG(K)=MFM
812      20 IF(RHO(I,NN).GT.0.)RHO(I,MM)=RHO(I,NN)
813      RHO(I,NN)=-1.
814      C . . . . .
815      C . . . . . ERROR FOUND IF MIXED CELL NOT FREE SURFACE.
816      C . . . . .
817      IF(ABS(RHO(INVOID,NN)-1.)GT.0.)GO TO 80
818      RHO(INVOID,NN)=0.
819      GO 25 II=I,MMAT
820      IF(RHO(II,NN).GT.0.)RHO(II,MM)=RHO(II,NN)
821      IF(II.NE.1)RHO(II,NN)=0.
822      25 SIF(II,MM)=0.
823      C . . . . .
824      C . . . . . ERROR FOUND IF MIXED CELL NOT FREE SURFACE.
825      C . . . . .
826      30 IF(ABS(RHO(INVOID,MM)-1.)GT.0.)GO TO 80
827      40 AMX(K)=0.
828      AIX(K)=0.
829      U(K)=0.
830      V(K)=0.
831      STRSRR(K)=0.
832      STRSRZ(K)=0.
833      STRSZZ(K)=0.
834      GO TO 120
835      C . . . . .
836      C . . . . . CELL L IS VOID AND CELL M IS MIXED.
837      C . . . . .
838      50 MFLAG(K)=MFM
839      MM=MFM-100
840      GO TO 30
841      C . . . . .
842      C . . . . . CELL M IS VOID.
843      C . . . . .
844      60 IF(MFL.NE.0)GO TO 70
845      C . . . . .
846      C . . . . . BOTH CELLS ARE VOID.
847      C . . . . .
848      MFLAG(K)=0
849      GO TO 40
850      C . . . . .
851      C . . . . . ERROR FOUND IF CELL L IS NOT VOID OR MIXED.
852      C . . . . .
853      70 IF(MFL.LT.100)GO TO 80
854      C . . . . .
855      C . . . . . CELL M IS VOID AND CELL L IS MIXED.
856      C . . . . .
857      MFLAG(K)=MFL
858      NN=MFL-100
859      GO TO 30
860      C . . . . .
861      C . . . . . ERROR EXIT. A CELL WAS FOUND WHICH HAS ZERO MASS AND WHICH WAS NOT
862      C . . . . . VOID OR FREE SURFACE. OR, BOTH CELLS PURE BUT OF DIFFERENT MATERIAL.
863      C . . . . .

```

```

864      80 PRINT 90,L,MFL,AMX(L),M,MFM,AMX(M)
865      90 FORMAT(1X,20HERROR IN COMPR, L=,14,3X,4HMFL=,14,3X,7HAMX(L)=,
866      |E15.5,5X,2HM=,14,3X,4HMFM=,14,3X,7HAMX(M)=,E15.5)
867      CALL EXIT
868      C . . . . .
869      C ONE OR BOTH CELLS CONTAIN MASS.
870      C . . . . .
871      100 IF(MFM.GT.100.OR.MFL.GT.100)GO TO 140
872      C . . . . .
873      C ERROR FOUND IF BOTH CELLS PURE BUT OF DIFFERENT MATERIAL.
874      C . . . . .
875      IF(MFM.NE.MFL)GO TO 80
876      C . . . . .
877      C BOTH CELLS PURE AND CONTAIN SAME MATERIAL. COMBINE THE TWO CELLS TO
878      C FORM A NEW PURE CELL SO THAT MASS, MOMENTUM AND ENERGY ARE CONSERVED.
879      C . . . . .
880      MFLAG(K)=MFL
881      UUK=(U(L)*AMX(L)+U(M)*AMX(M))/WSA
882      VVK=(V(L)*AMX(L)+V(M)*AMX(M))/WSA
883      WS=UUK**2+VVK**2
884      110 WSB=AMX(L)*(U(L)**2+V(L)**2)+AMX(M)*(U(M)**2+V(M)**2)
885      U(K)=UUK
886      V(K)=VVK
887      AIX(K)=AIX(L)*AMX(L)+AIX(M)*AMX(M)
888      STRSRR(K)=(AMX(L)*STRSRR(L)+AMX(M)*STRSRR(M))/WSA
889      STRSRZ(K)=(AMX(L)*STRSRZ(L)+AMX(M)*STRSRZ(M))/WSA
890      STRSZZ(K)=(AMX(L)*STRSZZ(L)+AMX(M)*STRSZZ(M))/WSA
891      AMX(K)=WSA
892      E=AIX(K)+.5*WSB
893      AIX(K)=E/AMX(K)-.5*WS
894      C . . . . .
895      C ZERO OUT THE ORIGINAL TWO CELLS.
896      C . . . . .
897      120 IF(M.EQ.K)GO TO 130
898      AIX(M)=0.
899      AMX(M)=0.
900      U(M)=0.
901      V(M)=0.
902      STRSRR(M)=0.
903      STRSRZ(M)=0.
904      STRSZZ(M)=0.
905      130 IF(L.EQ.K)RETURN
906      AIX(L)=0.
907      AMX(L)=0.
908      U(L)=0.
909      V(L)=0.
910      STRSRR(L)=0.
911      STRSRZ(L)=0.
912      STRSZZ(L)=0.
913      RETURN
914      C . . . . .
915      C ONE OR BOTH OF THE TWO CELLS ARE MIXED. SET UP TWO
916      C TEMPORARY MIXED CELLS AND ZERO OUT EXISTING MIXED CELLS.
917      C . . . . .

```

```

918 140 DO 150 II=1,2
919    DO 150 JJ=1,NMAT
920      XMAS(II,JJ)=0.
921      SSIE(II,JJ)=0.
922 150 XRH(II,JJ)=0.
923      RVM=0.
924      RVL=0.
925      IF(MFM.LT.100)GO TO 160
926 C . . . . .
927 C CELL M IS MIXED. TRANSFER PROPERTIES TO TEMPORARY MIXED CELL.
928 C . . . . .
929      N=MFM-100
930      RVM=RHO(NVOID,N)
931      JJ=1
932      GO TO 170
933 160 JJ=2
934 C . . . . .
935 C CELL L IS MIXED. TRANSFER PROPERTIES TO TEMPORARY MIXED CELL.
936 C . . . . .
937      N=MFL-100
938      RVL=RHO(NVOID,N)
939 170 DO 180 II=1,NMAT
940      XMAS(JJ,II)=XMASS(II,N)
941      SSIE(JJ,II)=SIE(II,N)
942      XRH(JJ,II)=RHO(II,N)
943      XMASS(II,N)=0.
944      SIE(II,N)=0.
945 180 RHO(II,N)=0.
946      RHO(II,N)=-1.
947      RHO(NVOID,N)=0.
948      IF(JJ.EQ.2)GO TO 190
949      IF(MFL.GT.100)GO TO 160
950 190 IF(MFM.GT.100)GO TO 200
951      IF(MFM.EQ.0)GO TO 210
952 C . . . . .
953 C CELL M IS PURE. TRANSFER PROPERTIES TO TEMPORARY MIXED CELL.
954 C . . . . .
955      XMAS(1,MFM)=AMX(M)
956      SSIE(1,MFM)=ATX(M)
957      JJ=(M-2)/IMAX+1
958      II=M-1-(JJ-1)*IMAX
959      XRH(1,MFM)=AMX(M)/(TAU(II)*DY(JJ))
960 200 IF(MFL.GT.100)GO TO 210
961      IF(MFL.EQ.0)GO TO 210
962 C . . . . .
963 C CELL L IS PURE. TRANSFER PROPERTIES TO TEMPORARY MIXED CELL.
964 C . . . . .
965      XMAS(2,MFL)=AMX(L)
966      SSIE(2,MFL)=ATX(L)
967      JJ=(L-2)/IMAX+1
968      II=L-1-(JJ-1)*IMAX
969      XRH(2,MFL)=AMX(L)/(TAU(II)*DY(JJ))
970 210 MFLAGIK)=N+100
971 C . . . . .

```

```

972 C FORM A NEW MIXED CELL K BY COMBINING THE TWO TEMPORARY MIXED
973 C CELLS SO THAT MASS, MOMENTUM AND ENERGY ARE CONSERVED.
974 C .....
975 RHO(NVOID,N)=1.
976 IF (ABS(RVM).LE.0..AND.ABS(RVL).LE.0.)RHO(NVOID,N)=0.
977 UUK=(U(L)*AMX(L)+U(M)*AMX(M))/WSA
978 VVK=(V(L)*AMX(L)+V(M)*AMX(M))/WSA
979 WS=UUK**2+VVK**2
980 DO 230 II=1,NHAT
981 WSB=XMAS(1,II)+XMAS(2,II)
982 IF (ABS(WSB).LE.0.)GO TO 220
983 RHO(II,N)=(XMAS(1,II)*XRH(1,II)+XMAS(2,II)*XRH(2,II))/WSB
984 WSC=XMAS(1,II)*(U(M)**2+V(M)**2)+XMAS(2,II)*(U(L)**2+V(L)**2)
985 SIE(II,N)=SSIE(1,II)*XMAS(1,II)+SSIE(2,II)*XMAS(2,II)
986 XMASS(II,N)=V.SB
987 E=SIE(II,N)+.5*WSC
988 SIE(II,N)=E/WSB=.5*VS
989 GO TO 230
990 220 XMASS(II,N)=0.
991 SIE(II,N)=0.
992 RHO(II,N)=.5*(XRH(1,II)+XRH(2,II))
993 230 CONTINUE
994 GO TO 110
995 END
996 SUBROUTINE EDIT
997 INCLUDE COMDIM
998 C
999 C *** SPECIAL EQUIV. FOR EDIT
1000 C
1001 C *** ERDUMP=1, WHEN ERROR CALLS EDIT FOR A TAPE DUMP ONLY
1002 IF (ERDUMP.GT.0.) GO TO 150
1003 C *** ENERGY SUM (ESUM) AND RELATIVE ERROR IN SUM (RELERR)
1004 C COMPUTED. ECK IS LARGEST ERROR COMPUTED AND ON PRINT
1005 C CYCLES IS PRINTED AND COMPARED TO DMIN, MAXIMUM
1006 C ALLOWABLE ERROR.
1007 ESUM=0.
1008 DO 10 K=2,KMAX
1009 10 ESUM=ESUM+AMX(K)*(.5*(U(K)**2+V(K)**2)+AIX(K))
1010 RELERR=(ESUM-ETH)/ETH
1011 IF (ABS(RELERR).LT.ABS(ECK)) GO TO 20
1012 ECK=RELERR
1013 NECYCL=NC
1014 20 CONTINUE
1015 C *** ADDTCR CALLED WHEN NADD (INPUT) .GT.0. NADD ALSO TELLS
1016 C NUMBER OF TRACERS TO ADD BETWEEN ANY TWO EXISTING
1017 C TRACERS IN THE SPECIFIED REGION.
1018 IF (NADD.GT.0) CALL ADDTCR
1019 C *** NPRINT = 1 WHEN EDIT IS CALLED TO DO AN INTERMEDIATE
1020 C PRINT. SKIP TESTS ON TIME TO STOP, PRINT, REZONE, ETC.
1021 C WHICH ALREADY HAVE BEEN DONE FOR THIS CYCLE.
1022 IF (NPRINT.EQ.1) GO TO 190
1023 C *** I3=1 SIGNALS A SHORT PRINT
1024 I3=1
1025 C *** IF THIS IS FIRST CYCLE OF RUN, *FLAGF=1.

```



```

1024 IF (WFLAGF.GT.0.) GO TO 120
1027 C      *** IS THIS THE TIME OR CYCLE TO STOP EXECUTION
1028 IF (ICSTOP.LE.NC.AND.ICSTOP.GT.0) GO TO 30
1029 IF (T*(1.+ROEPS).GE.TSTOP.AND.TSTOP.GT.0) GO TO 30
1030 C      *** SHOULD THE GRID BE REZONED
1031 IF(((REZ.NE.C..AND.REZFCT.NE.D.),OR.SS4.NE.D.),AND.NUMREZ.GT.0)
1032 GO TO 190
1033 C
1034 GO TO 40
1035 C      *** SET WFLAGL=1. TO SAY THIS IS LAST CYCLE OF RUN
1036 30 WFLAGL=1.
1037 I3=11
1038 NPRINT=1
1039 NUMSPT=NDUMP7
1040 NUMSP=0
1041 GO TO 190
1042 40 ASSIGN 140 TO LOCA
1043 ASSIGN 110 TO LOCB
1044 C      *** ARE WE PRINTING ON TIME OR CYCLE INTERVALS
1045 IF (PRDEL.NE.0.) GO TO 50
1046 45 IF (IPCYCL.NE.0) GO TO 100
1047 GO TO 430
1048 C      *** PRINTING ON TIME. IS IT TIME TO PRINT
1049 50 IF (T*(1.+ROEPS).GE.PRTIME) GO TO 70
1050 C      *** NO. BUT WILL NEXT CYCLE BYPASS THE PRINT TIME
1051 IF (PRTIME.GE.T+DT) GO TO 60
1052 DT=PRTIME-T
1053 DTNA=DT
1054 60 GO TO LOCA, (140,130)
1055 C      *** YES, IT IS TIME TO PRINT. NPRINT=1 FLAGS THIS AS A
1056 C      PRINT CYCLE.
1057 70 NPRINT=1
1058 C      *** AVOID TRUNCATION
1059 T=PRTIME
1060 C      *** IS IT TIME TO RESCALE PRINT INTERVAL
1061 IF (T*(1.+ROEPS).LT.PRLIM.OR.NUMSCA.LE.0) GO TO 80
1062 C      *** CHANGE PRINT INTERVAL AND THE TIME FOR THE NEXT
1063 C      RESCALING.
1064 PRDEL=PRDEL*PFRACT
1065 PRLIM=PRLIM*PFRACT
1066 NUMSCA=NUMSCA-1
1067 C      *** DEFINE TIME FOR NEXT PRINT.
1068 80 PRTIME=T+PRDEL
1069 IVS=(PRTIME+.5*PRDEL)/PRDEL
1070 WS=IVS
1071 PRTIME=WS*PRDEL
1072 C      *** WILL WE BYPASS TIME TO PRINT
1073 IF (PRTIME.GE.T+DT) GO TO 90
1074 C      *** YES, ADJUST DT
1075 DT=PRTIME-T
1076 DTNA=DT
1077 90 GO TO LOCB, (110,130)
1078 C      *** PRINTING ON CYCLES. IS THIS A PRINT CYCLE
1079 100 IF (MOD(INC,IPCYCL).NE.0) GO TO 60

```

```

1080 C          *** YES. NPRINT = 1 FLAGS THIS AS A PRINT CYCLE.
1081 NPRINT=1
1082 C          *** IS THIS THE CYCLE TO RESCALE PRINT INTERVAL
1083 IF (INC.LT.PRLIM.OR.NUMSCA.LE.0) GO TO 90
1084 C
1085 C          *** YES. MULTIPLY NUMBER OF CYCLES BETWEEN PRINTS BY PRFACT
1086 C
1087 IFCYCL=INT(PRFACT)*IFCYCL
1088 PRLIM=PRFACT*PRLIM
1089 NUMSCA=NUMSCA-1
1090 GO TO LOCB, (110,130)
1091 C          *** TEST FOR SHORT OR LONG PRINT
1092 C          *** NUMSP COUNTS NUMBER OF SHORT PRINTS SINCE LAST LONG
1093 C          PRINT. NUMSPT COUNTS NUMBER OF CYCLES SINCE LAST
1094 C          TAPE DUMP.
1095 110 NUMSP=NUMSP+1
1096 NUMSPT=NUMSPT+1
1097 IF (NUMSP.NE.NFRFLP) GO TO 190
1098 NUMSP=0
1099 C          *** I3=I1 SIGNALS A LONG PRINT
1100 120 I3=I1
1101 C          *** PRINT OF RESTART CYCLE WILL BE SHORT IF PK(3).EQ.-2.
1102 IF (PK(3).EQ.-2. .AND. WFLAGF.GT.0.) I3=1
1103 GO TO 190
1104 C          *** CHECK FOR ENERGY DISCREPANCY
1105 130 IF (ABSIECK).GT.DMIN) GO TO 440
1106 C          *** IF LAST CYCLE, REWIND TAPE
1107 140 IF (WFLAGL.EQ.0.) GO TO 470
1108 REWIND KUNITW
1109 GO TO 470
1110 150 NUMSPT=0
1111 IF (NODUMP.NE.0) GO TO 170
1112 BACKSPACE KUNITW
1113 WS=555.0
1114 WRITE (KUNITW) WS, CYCLE
1115 WRITE (KUNITW) (Z(I), I=1, 150)
1116 WRITE (KUNITW) (U(I), V(I), AMX(I), AIX(I), P(I), MFLAG(I), I=1, KMAX)
1117 WRITE (KUNITW) (STRSZ(I), STRSRR(I), STRSRZ(I), I=1, KMAX)
1118 WRITE (KUNITW) (X(I), DX(I), TAU(I), I=1, IMAX)
1119 WRITE (KUNITW) (Y(I), DY(I), I=1, JMAX)
1120 WRITE (KUNITW) (CZERO(M), STK1(M), STK2(M), STEZ(M), RMU(M),
1121 1 ANDM(M), RHOIN(M), SSIEN(M), UUR(M), VVA(M), MAT(M), PLW(M),
1122 2 M=1, NMAT)
1123 WRITE (KUNITW) (MPAC(I), MPACK(I), I=1, MBBB)
1124 WRITE (KUNITW) ((PACX(I, L), PACY(I, L), I=1, MBRB), L=1, MBO)
1125 WRITE (KUNITW) ((XNASS(M, L), RHO(M, L), STE(M, L), SAMPY(M, L),
1126 1 SAMMP(M, L), M=1, NMAT), RHO(NVOID, L), L=1, NMXCLS)
1127 DO 160 N=1, NVOID
1128 NP=NMP(N)
1129 WRITE (KUNITW) NP, (TX(N, L), TY(N, L), L=1, NP)
1130 160 CONTINUE
1131 NP=((IMAX/2+1)*(JMAX/2+1)
1132 WRITE (KUNITW) NP, (XP(L), YP(L), L=1, NP)
1133 WS=666.0

```

```

1134 WRITE(KUNITW) WS, WS
1135 WRITE (6,550) NC
1136 IF (#FLAGL.EQ.0.) GO TO 170
1137 END FILE KUNITW
1138 170 CONTINUE
1139 IF (ERDUMP.GT.0.) GO TO 470
1140 GO TO 280
1141 C      *** INITIALIZE PR ARRAY, TEMPORARY STORAGE FOR ENERGY, MASS
1142 C      AND MOMENTUM TOTALS PRINTED OUT.
1143 190 DO 200 J=1,8
1144     DO 200 M=1,NVOID
1145 200 PQ(M,J)= 0.
1146 C
1147     DO 235 K=2,KMAX
1148     MF= MFLAG(K)
1149     IF(MF.GE.100) GO TO 210
1150     M1= MF
1151     M2= MF
1152     WS= AMX(K)
1153     WSI= AIX(K)*WS
1154     GO TO 215
1155 C
1156 210 MC= MF-100
1157     M1= 1
1158     M2= NHAT
1159 215 DO 230 M=M1,M2
1160 C
1161     IF(MF.LT.100) GO TO 220
1162     WS= XMASS(M,MC)
1163     WSI= SIE(M,MC)*WS
1164 220 PQ(M,1)= PQ(M,1) + WSI
1165     PQ(M,2)= PQ(M,2) + .5*WS*(U(K)**2 + V(K)**2)
1166     PQ(M,4)= PQ(M,4) + WS
1167     WSA= WS*V(K)
1168     PQ(M,5)= PQ(M,5) + WSA
1169     IF (WSA.GT.0.) PQ(M,6)= PQ(M,6) + WSA
1170     PQ(M,7)= PQ(M,7) + WS*U(K)
1171 230 CONTINUE
1172 235 CONTINUE
1173 C
1174     DO 240 M=1,NMAT
1175     PQ(M,3)= PL3(M)
1176 240 PQ(M,3)= PQ(M,1) + PQ(M,2)
1177 C      TOTALS
1178     DO 245 J=1,8
1179     SUM=0.
1180     DO 242 M=1,NMAT
1181 242 SUM= SUM + PQ(M,J)
1182 245 PQ(NVOID,J)= SUM
1183     IF (IMAX.GT.1) GO TO 260
1184 C
1185 C      *** IF DOING A 1-D PROBLEM DIVIDE TOTALS BY NZ WHERE
1186 C      NZ=4*(NUMBER OF TIMES THE GRID HAS BEEN REZONED.)
1187 C

```

```

1188      PROP (1)=ETH/NZ
1189      PROP (2)=ECK/NZ
1190      PROP (5)=EZPH2/NZ
1191      PROP (6)=BBOUND/NZ
1192      DO 250 J=1,24
1193 250    PROP (J+6)=PR(J)/NZ
1194      PROP (31)=BOTM/NZ
1195      PROP (32)=RTM/NZ
1196      PROP (33)=TOPM/NZ
1197      PROP (34)=EVAPM/NZ
1198      PROP (35)=EMOB/NZ
1199      PROP (36)=EMOR/NZ
1200      PROP (37)=EMOT/NZ
1201      PROP (38)=EVAPEN/NZ
1202      PROP (39)=BOTMU/NZ
1203      PROP (40)=RTMU/NZ
1204      PROP (41)=TOPMU/NZ
1205      PROP (42)=EVAPMU/NZ
1206      PROP (43)=BOTMV/NZ
1207      PROP (44)=RTMV/NZ
1208      PROP (45)=TOPMV/NZ
1209      PROP (46)=EVAPMV/NZ
1210      PROP (47)=EOB/NZ
1211      PROP (48)=EOR/NZ
1212      PROP (49)=EOT/NZ
1213      WRITE(6,530) PROB,T,NC,PROP(1),PROP(2),NECYCL,PROP(5),PROP(6)
1214      WRITE(6,540) (PROP(J),J=7,49)
1215      GO TO 270
1216 260    WRITE(6,530) PROB,T,NC,ETH,ECK,NECYCL,EZPH2,BBOUND
1217      WRITE(6,540) (M,(PQ(M,J),J=1,8),M=1,NMAT)
1218      WRITE(6,545) (PQ(NVOID,J),J=1,8),BOTM,RTM,TOPM,EVAPM,EMOB,
1219      1 EMOR,EMOT,EVAPEN,BOTMU,RTMU,TOPMU,EVAPMU,BOTMV,RTMV,TOPMV,
1220      2 EVAPMV,EOB,EOR,EOT
1221 270    CONTINUE
1222      C      *** ENERGY TOTALS STORED FOR LATER USE IN TRACER POINT
1223      C      PLOTS.
1224      XIENRG=PQ(NVOID,1)
1225      XKENRG=PQ(NVOID,2)
1226      XIENRG=PQ(NVOID,3)
1227      C      *** IS THIS A TAPE DUMP OR REZONE CYCLE
1228      IF (NUMSPT.EQ.NDUMP7.OR.(REZ.NE.O..AND.REZECT.NE.O..AND.NUMREZ.GT.
1229      10)) GO TO 150
1230      C      *** PRINT COORDINATES OF BOUNDARY TRACERS FOR EACH
1231      C      MATERIAL PACKAGE.
1232 280    WRITE(6,590)
1233      DO 300 N=1,NMAT
1234      NP=NMP(N)
1235      WRITE(6,510)N
1236 300    WRITE(6,600) (L,TX(N,L),TY(N,L),L=1,NP)
1237      C      *** PRINT COORDINATES OF FREE SURFACE TRACERS.
1238      NP=NMP(NVOID)
1239      WRITE(6,610)
1240 300    WRITE(6,600) (L,TX(NVOID,L),TY(NVOID,L),L=1,NP)
1241      C      *** PRINT SYMBOLIC CONTOUR MAPS OF COMPRESSION, PRESSURE.

```

```

1242 C VELOCITY, AND INTERNAL ENERGY
1243 IF (MAPS.GT.0) CALL MAP
1244 C *** SHORT PRINT MEANS I3=1 AND PROPERTIES ARE PRINTED ONLY
1245 C FOR CELLS IN FIRST COLUMN. LONG PRINT MEANS I3=11 AND
1246 C PROPERTIES ARE PRINTED FOR ALL CELLS IN ACTIVE GRID.
1247 370 DO 420 I=1,I3
1248 KSPACE=0
1249 WFLAGP=1.
1250 J=I2+1
1251 K=I2*(IMAX+I+1)
1252 DO 410 L=1,I2
1253 J=J-1
1254 K=K-IMAX
1255 375 MFK=MFLAG(K)
1256 380 IF (WFLAGP.EQ.0.) GO TO 390
1257 WRITE (6,560) I,X(I),DX(I)
1258 WRITE(6,565)
1259 WFLAGP=0.
1260 390 WS=AMX(K)/(TAU(I)*DY(J))
1261 MN=0
1262 WSA=0.
1263 IF (MFK.GT.100 .OR. MFK.EQ.0) GO TO 395
1264 MN=MAT(MFK)
1265 WSA = WS/RHOI(MFK)
1266 IF (MN.EQ.20) GO TO 395
1267 WSA = WS/RHOZ(MN)
1268 395 WRITE(6,520) J,MFK,U(K),V(K),P(K),AIX(K),WSA,AMX(K),STRSZZ(K),
1269 I STRSHR(K), STRSRZ(K),Y(J)
1270 IF (MFK.LT.100) GO TO 398
1271 M=MFK-100
1272 WRITE(6,630) RHO(NVOID,M)
1273 DO 397 N=1,NMAT
1274 MN = MAT(N)
1275 WSA=XHASS(N,M)/RHO(N,M)
1276 WSA=WSA/(TAU(I)*DY(J))
1277 WSC = RHO(N,M)/RHOI(MN)
1278 IF (MN.EQ.20) GO TO 396
1279 WSC=RHO(N,M)/RHOZ(MN)
1280 396 CONTINUE
1281 WRITE(6,620) MN,WSA,RHO(N,M),SIE(N,M),WSC,XHASS(N,M)
1282 397 CONTINUE
1283 WRITE(6,565)
1284 398 CONTINUE
1285 KSPACE=0
1286 410 CONTINUE
1287 420 CONTINUE
1288 IF (NPRINT.EQ.1) GO TO 130
1289 ASSIGN 130 TO LOCA
1290 ASSIGN 130 TO LOCB
1291 IF (PRDELTA.NE.0.) GO TO 50
1292 GO TO 100
1293 C *** PRINT DELTA NOT SPECIFIED IN INPUT
1294 430 MK=45
1295 GO TO 460

```

```

1296 C          *** ENERGY CHECK
1297 440 NK=130
1298 GO TO 460
1299 460 NR=5
1300 CALL ERROR
1301 470 WFLAGP=0.
1302 WFLAGF=0.
1303 C          *** SHOULD GRID BE REZONED ON THIS CYCLE
1304 IF(((REZ.NE.0..AND.REZFACT.NE.0.)..OR.SS4.NE.0.)..AND.NUMREZ.GT.0)
1305 GO TO 480
1306 RETURN
1307 480 CALL REZONE
1308 C          *** MUST CALL CDT TO RECALCULATE PRESSURES
1309 TNOW=T
1310 DTNOW=DT
1311 REZ=D.
1312 SS4=0.
1313 CALL CDT
1314 T=TNOW
1315 DT=DTNOW
1316 DTNA=DT
1317 NUMKEZ=NUMREZ-1
1318 C
1319 C          *** NREZ = NUMBER OF REZONES ALLOWED (INPUT VALUE OF NUMREZ)
1320 C          NUMREZ = NUMBER OF REZONES ALLOWED MINUS THE NUMBER
1321 C          OF REZONES PERFORMED SINCE T=0.
1322 C
1323 NRZ=NREZ-NUMREZ
1324 C          *** NZ USED IN PRINTOUT OF TOTALS FOR 1-D PROBLEMS
1325 NZ=4..*NRZ
1326 C
1327 NUMSPT=NDUMP7
1328 GO TO 120
1329 C
1330 C          FORMATS
1331 C
1332 510 FORMAT(/9H PACKAGE ,12/ 7X, 5I6X,1HN,6X,1HX,7X,1HY//)
1333 520 FORMAT(14,15,1X,1P4E12.4,1PE13.5,1P4E12.4,1PE9.3)
1334 530 FORMAT(8H1PROBLEM,6X,4HTIME,8X,5HCYCLE,3X,13HTOT.EN.THEOR,3X,
1335 1 19HMAX.REL.ERROR-CYCLE,3X,18HIE SET TO ZERO-PH2,3X,
1336 2 20HELASTIC PLASTIC WORK/1F8.4,2X,1PE13.7,
1337 3 17,1PE17.7,E16.7,15,E19.7,E21.7//)
1338 540 FORMAT (12H PACKAGE NO.,6X,2HIE,14X,2HKE,8X,13HTOT.EN. (SUM),6X,
1339 1 4HMASS,12X,2HNV,8X,12HNV(POSITIVE),8X,2HNU,8X,12HPLASTIC-WORK/
1340 2 (1H ,15,5X,1P8E15.7))
1341 545 FORMAT(14X,8(12H-----,3X)/ 7H TOTALS,4X,1P
1342 58E15.7///9H BOUNDARY,9X,6HBOTTOM,9X,5HRIGHT,10X,3HTOP,8X,12HSEVAP0
1343 6RATEDS//9H MASS OUT,2X,1P4E15.7//11H ENERGY OUT,1P4E15.7//7H MU OUT,
1344 74X,1P4E15.7//7H MV OUT,4X,1P4E15.7//11H WORK DONE ,1P3E15.7//)
1345 550 FORMAT (1H0//21H TAPE 7 DUMP ON CYCLE15//)
1346 560 FORMAT (1H ///4H I =13,6X,6HR(1) =F12.3,6X,7HDR(1) =E14.7//)
1347 565 FORMAT(/3H J,7H MFLAG,6X,1HU,11X,1HV,11X,1HP,16X,3HSIE,9X,4HCOMP
1348 1 8X, 5HMASS, 8X, 3HSZZ, 9X, 3HSRR, 9X, 3HSRZ, 9X, 1HZ//)
1349 570 FORMAT (1H0)

```

```

1350 580 FORMAT (//22H J OF PRESSURE-MAXIMUM/(2515)) 4600
1351 590 FORMAT(//30X, 53HCELL-COORDINATES OF TRACERS FOR EACH MATERIAL PAC
1352 1KAGE/ )
1353 600 FORMAT(9X,16,2F8.2,16,2F8.2,16, 2F8.2, 16, 2F8.2, 16, 2F8.2)
1354 610 FORMAT(/21H FREE SURFACE TRACERS/ 7X, 5(6X,1HN,6X,1HX,7X,1HY)/ )
1355 620 FORMAT(16X,14,7X,F6.3,4X,1P4E13,5)
1356 630 FORMAT(16X,5H MAT ,5X,10HFRAC. VOL.,7X,3HRHO,10X,3HSIE,9X,
1357 1 4HCOMP,9X,4HMASS,9X,13HRHO(NVOID,M)=,F3.1)
1358 END
1359 SUBROUTINE EQST 4620-
1360 C *** COMPUTE PRESSURE AS A FUNCTION OF ENERGY AND DENSITY
1361 C USING THE TILLOTSON EQUATION OF STATE.
1362 INCLUDE COMDIM
1363 DIMENSION ESA(30), ESB(30), ESCAPA(30), ESCAPB(30), ESEZ(30),
1364 1 ESALPH(30), ESBETA(30), ESES(30), ESESP(30)
1365 C
1366 C *** MATERIALS WHOSE EQ. OF STATE CONSTANTS ARE SPECIFIED.
1367 C IN THE FOLLOWING DATA STATEMENTS
1368 C 1 - TUNGSTEN (W)
1369 C 2 - COPPER (CU)
1370 C 3 - IRON (FE)
1371 C 4 - ALUMINUM (AL)
1372 C 5 - BERYLLIUM (BE)
1373 C 6 - TITANIUM (TI)
1374 C 7 - NICKLE (NI)
1375 C 8 - MOLYBDENUM (MO)
1376 C 9 - THORIUM (TH)
1377 C 10 - LEAD (PB)
1378 C 11 - POLYMERS
1379 C 12 - GRANITE
1380 C 13 - ANDESITE
1381 C 14 - WET TUFF
1382 C 15 - DRY TUFF
1383 C 16 - OIL SHALE
1384 C 17 - DOLOMITE
1385 C 18 - LIMESTONE
1386 C 19 - HALITE
1387 C
1388 DATA (ESA(K),K=1,19)
1389 1 / 4*.5, .55, 3*.5, 2*.4, .6, 8*.5/
1390 C
1391 DATA (ESB(K),K=1,19)
1392 1 / 1.04, 2*1.5, 1.63, .62, .60, 1.33, 1.02, .86, 2.4,
1393 1 2.0, 4*1.3, 1.0, 3*.6/
1394 C
1395 DATA (ESCAPA(K),K=1,19)
1396 1 / 3.08E12, 1.39E12, 1.28E12, .75E12, 1.17E12, 1.03E12,
1397 1 1.91E12, 2.71E12, .53E12, .466E12, .075E12, .60E12,
1398 2 .34E12, .10E12, .045E12, .28E12, .85E12, .4E12,
1399 3 .25E12/
1400 C
1401 DATA (ESCAPB(K),K=1,19)
1402 1 / 2.5E12, 1.1E12, 1.05E12, .65E12, .55E12, .5E12,
1403 1 1.5E12, 1.65E12, .5E12, .15E12, .02E12, 0.0,

```

```

1404      2      .28E12, .06E12, .03E12, .11E12, .30E12, .67E12,
1405      3      .30E12/
1406      C
1407      DATA (ESEZ(K),K=1,19)
1408      1      / .225E12, .325E12, .095E12, .05E12, .175E12, .07E12,
1409      1      .09E12, .045E12, .025E12, .02E12, .07E12, .16E12,
1410      2      .16E12, .11E12, .06E12, .11E12, .10E12, .10E12,
1411      3      .05E12/
1412      C
1413      DATA (ESALPH(K),K=1,19)
1414      1      / 10., 7*5., 9., 2*10., 8*5./
1415      C
1416      DATA (ESBETA(K),K=1,19)
1417      1      / 10., 7*5., .88, 2., 9*5./
1418      C
1419      DATA (ESES(K),K=1,19)
1420      1      / 1.11E10, 1.38E10, 2.44E10, 3.0E10, 10.0E10, 3.5E10,
1421      1      2.85E10, 2.8E10, 2.0E10, .26E10, 2.4E10, 2*3.5E10,
1422      2      3*3.2E10, 2*2.5E10, 2.0E10/
1423      C
1424      DATA (EESP(K),K=1,19)
1425      1      / 5.6E10, 6.9E10, 10.2E10, 19.0E10, 46.0E10, 12.5E10,
1426      1      9.4E10, 9.0E10, 8.0E10, .97E12, 18.0E12, 2*.18E12,
1427      2      .16E12, .18E12, .16E12, 2*.14E12, .15E12/
1428      C      *** IS MATERIAL N AN IDEAL GAS
1429      IF(N.EQ.20) GO TO 30
1430      C      *** STORE CONSTANTS FOR MATERIAL N.
1431      A=ESA(N)
1432      B=ESB(N)
1433      CAPA=ESCAPA(N)
1434      CAPB=ESCAPB(N)
1435      C      *** CAPB IS TENTATIVELY SET TO -CAPB WHEN
1436      C      RHOZ .LT. RHOZR IN A MIXED CELL (WS=-1.)
1437      IF(WS.LE.-1.) CAPB=-CAPB
1438      EZ = ESEZ(N)
1439      ALPHA = ESALPH(N)
1440      BETA = ESBETA(N)
1441      ES = ESES(N)
1442      ESP = EESP(N)
1443      RHOZR = RHOZ(N)
1444      IF(RHOZR.LE.0.) GO TO 80
1445      WS=ESP-ES
1446      IF(WS.LE.0.) WS=1.
1447      SSI=1./*S
1448      ETA = RHOZ/RHOZR
1449      IF(ETA.LE.0.) GO TO 80
1450      VOV = 1./ETA
1451      EXPMIN= 1. - CAPA/(ABS(CAPB)+CAPB)
1452      IF(EXPMIN.LE.0.) CAPB= ABS(CAPB)
1453      C
1454      IF(ENERGY.LE.0.) GO TO 20
1455      C      *** P1, P4 = THERMAL PRESSURE TERMS.
1456      P1=ENERGY*RHOV*A
1457      P4=B/(ENERGY/(EZ*ETA**2)+.) * ENERGY * RHOZ.

```



```

1458 C      *** P5 = MECHANICAL PRESSURE TERM.
1459 10 P5=CAPA*(ETA-1.)
1460 P2=-1.
1461 C      *** IF RHO4 .LT. RHOZR AND ENERGY IS BETWEEN ES AND ESP,
1462 C      A COMBINATION OF THE EXPANDED AND CONDENSED EQUATIONS
1463 C      OF STATE IS USED.
1464 IF(ETA.GE.1.1) GO TO 50
1465 C      *** ESP= ENERGY TO VAPORIZE MATERIAL, MUST EXCEED ES.
1466 IF(ENERGY.GT.ESP) GO TO 40
1467 C      *** ES = ENERGY TO BRING MATERIAL TO VAPOR TEMPERATURE.
1468 IF(ENERGY.GT.ES) P2=1.
1469 C      *** P2 = 1. SIGNALS USE OF BOTH EXPANDED AND COMPRESSED
1470 C      FORMULATIONS, OTHERWISE P2=-1.
1471 GO TO 50
1472 C      *** THERMAL PRESSURE TERMS = 0. WHEN ENERGY IS ZERO OR
1473 C      NEGATIVE.
1474 20 P1=0.
1475 P4=0.
1476 IF(MFK.LT.100 .AND. ETA.LE.AMDM(MFK)) GO TO 80
1477 GO TO 10
1478 C      *** IDEAL GAS
1479 30 PRESUR =(GAMMA-1.)*RHO4*ENERGY
1480 GO TO 90
1481 C      *** EXPANDED STATE.
1482 40 P8=(1.-VOW)
1483 P9=EXP(ALPHA*P8)
1484 P12=EXP(-BETA*P8**2)
1485 PRESUR=P1+(P4+P5*P9)*P12
1486 IF(P2.LT.0.) GO TO 70
1487 P1=SS1*(ENERGY-ES)
1488 PRESUR = P1*PRESUR+(1.-P1)*P3
1489 GO TO 70
1490 C      *** IF (2B-A)/(2B+A) .LT. RHO4 .LT. RHOZR AND
1491 C      THE CELL IS MIXED (CAPB.LT.0), LINEARLY
1492 C      DECREASE CAPB BY A FACTOR BETWEEN -1 AND +1.
1493 C      IF RHO4 .LT. (2B-A)/(2B+A) LEAVE CAPB
1494 C      UNCHANGED, VIZ., CAPB=-ABS(CAPB).
1495 50 IF((CAPB.LT.0..AND.EXPMIN*VOW.LT.2.-EXPMIN)
1496 1 CAPB=CAPB*(1.-EXPMIN*VOW)/(EXPMIN-1.)
1497 P6=CAPB*((ETA-1.)*2)
1498 P7=P6*P5
1499 C      *** PRESSURE PLATEAU FOR PURE CELLS THAT ARE UNDERDENSE
1500 C      AND COLD.
1501 IF(MFK.LT.100 .AND. ETA.LT.EXPMIN) P7= .5*CAPA*(EXPMIN-1.)
1502 PRESUR=P1+P4+P7
1503 IF(P2.LT.0.) GO TO 60
1504 C      *** USING COMBINATION OF CONDENSED AND EXPANDED EQUATIONS
1505 C      OF STATE.
1506 P3=PRESUR
1507 GO TO 40
1508 C      *** USING CONDENSED EQUATION OF STATE.
1509 60 IF(PRESUR.GE.0.) GO TO 90
1510 C      *** WHEN USING CONDENSED EQ. OF STATE, SET PRESUR = 0
1511 C      IF MATERIAL IS EXPANDED(ETA.LT.AMDM), OR IF J.LT.N6.

```

```

1512 C          (N6 IS INPUT PARAMETER).
1513 IF(MFK.LT.100 .AND. (J.LE.N6 .OR. ETA.LE.AMDM(MFK))) GO TO 80
1514 GO TO 90
1515 C          *** ALLOW NEGATIVE PRESSURES ONLY WHEN USING CONDENSED
1516 C          EQ. OF STATE.
1517 70 IF(PRESUR.GE.0. .OR. MFK.GT.100) GO TO 90
1518 80 PRESUR = 0.
1519 90 RETURN
1520 END
1521 SUBROUTINE ERROR
1522 C          *** PRINT ERROR MESSAGE AND CELL QUANTITIES BEFORE
1523 C          EXITING ON AN ERROR CONDITION DETECTED BY THE
1524 C          PROGRAM.
1525 INCLUDE COMDIM
1526 C
1527 IF (NERR.EQ.1) GO TO 420
1528 GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140),NR
1529 10 WRITE (6,210) NK
1530 GO TO 410
1531 20 WRITE (6,220) NK
1532 GO TO 410
1533 30 WRITE (6,230) NK
1534 GO TO 410
1535 40 WRITE (6,240) NK
1536 GO TO 410
1537 50 WRITE (6,250) NK
1538 GO TO 410
1539 60 WRITE (6,260) NK
1540 GO TO 410
1541 70 WRITE (6,270) NK
1542 GO TO 410
1543 80 WRITE (6,280) NK
1544 GO TO 410
1545 90 WRITE (6,290) NK
1546 GO TO 410
1547 100 WRITE (6,300) NK
1548 GO TO 410
1549 110 WRITE (6,310) NK
1550 GO TO 410
1551 120 WRITE (6,320) NK
1552 GO TO 410
1553 130 WRITE (6,330) NK
1554 GO TO 410
1555 140 WRITE (6,340) NK
1556 GO TO 410
1557 410 WRITE (6,350) I,J,K,(M,Z(M),Z(M),M=1,150)
1558 C          *** IF NR=1, ERROR IS IN INPUT DECK
1559 IF(NR.EQ.1) GO TO 420
1560 C          *** IF NR=5 AND NK=130, EDIT PRINT HAS JUST BEEN DONE. BY
1561 C          SETTING ERDUMP=1.,EDIT WILL DO A TAPE DUMP BUT NOT
1562 C          ANOTHER PRINT.
1563 IF(FLAGL.GT.0.) GO TO 420
1564 IF (NR.EQ.5.AND.NK.EQ.130) ERDUMP=1.
1565 NERR = 1

```

```

1566      I3=11
1567      NPRINT=1
1568      WFLAGL=1.
1569      NUMSPT=NDUMP7
1570      CALL EDIT
1571      420 CALL EXIT
1572  C
1573  C
1574      210 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1575           ) INPUT )
1576      220 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1577           ) SETUP )
1578      230 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1579           ) CDT )
1580      240 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1581           ) ES )
1582      250 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1583           ) EDIT )
1584      260 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1585           ) MAP )
1586      270 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1587           ) PH1 )
1588      280 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1589           ) PH3 )
1590      290 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1591           ) INFACE )
1592      300 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1593           ) NEWMIX )
1594      310 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1595           ) NEWRHO )
1596      320 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1597           ) FLUX )
1598      330 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1599           ) PH2 )
1600      340 FORMAT (1H1,5X,3BH*** ERROR EXIT - SEE STATEMENT NUMBER ,15,10H IN
1601           ) REZONE )
1602      350 FORMAT (//5X,6H      I=,13,6H      J=I,6H      K=I3//16X,7HZ=BLOCK//6X,
1603           115H REAL FORMAT ,5X,15H INTEGER FORMAT/2X,1H1,8X,4HZ(11,17X,4HZ(
1604           21)//(14,2X,E15.6,5X,115))
1605      END
1606      SUBROUTINE FLUX
1607      INCLUDE COMDIM
1608  C***** COMPUTE FLUXES ACROSS TOP AND RIGHT BOUNDARIES FOR *****
1609  C      EACH MATERIAL IN A MIXED CELL.
1610  C
1611      MA=MA-100
1612      MR=MR-100
1613  C      *** BEGIN LOOP ON MATERIALS
1614  C
1615      200 DO 500 NT= 1,NMAT
1616          VABOVE=0.
1617          URR=0.
1618  C
1619  C      *** IF CELL DOESNT CONTAIN MATERIAL N, SKIP OUT

```

```

1620 C IF(RHO(NT,M).LE.D.) GO TO 500
1621 C
1622 C *** CALCULATE FLUX OF MATERIAL N ACROSS TOP BOUNDARY.
1623 C IF CELL ABOVE DOESNT CONTAIN MATERIAL N, SKIP OUT.
1624 C
1625 RHOA=D.
1626 IF(J.EQ.JMAX .AND. V(K).GT.D.) GO TO 230
1627 IF(J.EQ.JMAX .AND. V(K).LE.D.) GO TO 300
1628 C
1629 IF(MA.GT.C ) GO TO 210
1630 C *** CELL ABOVE PURE
1631 IF(MA+100 .NE. NT) GO TO 300
1632 RHOA = AMX(KA)/(TAU(1)*DY(J+1))
1633 VC=TAU(1)*DY(J+1)
1634 GO TO 220
1635 C *** CELL ABOVE MIXED
1636 C *** RHO(NT,MA) = C. INDICATES CELL DOES NOT CONTAIN
1637 C MATERIAL NT.
1638 210 IF(RHO(NT,MA).LE.D.) GO TO 300
1639 RHOA = RHO(NT,MA)
1640 VC=XMASS(NT,MA)/RHOA
1641 C *** IF CELL ABOVE CONTAINS A FREE SURFACE, SET IFS2=1
1642 IFS2=D
1643 IF(RHO(NVOID,MA).GT.D.) IFS2 = 1
1644 IF(AMX(KA).LE.C.) GO TO 230
1645 220 IF(AMX(K).LE.D.) GO TO 235
1646 WSA=(V(K)+V(KA))*0.5
1647 WS=DT/DY(J)
1648 WSB=1.0+(V(KA)-V(K))*WS
1649 IF(ABS(V(K))*WS.GT.STAB .OR. ABS(V(KA))*WS.GT.STAB) WSB=1.0
1650 VABOVE = WSA/WSB
1651 GO TO 240
1652 230 VABOVE = V(K)
1653 GO TO 240
1654 235 VABOVE=V(KA)
1655 C
1656 C *** IF ONLY ONE CELL CONTAINS A FREE SURFACE, USE
1657 C DENSITY OF THE OTHER ONE TO DEFINE THE MASS FLUX.
1658 240 IF(ABS(VABOVE).LE.UMIN) GO TO 300
1659 IF(VABOVE.GT.C. .AND. XMASS(NT,M).LE.D.) GO TO 300
1660 IF(MA.LE.D .OR. J.EQ.JMAX) GO TO 245
1661 IF(VABOVE.LT.C. .AND. XMASS(NT,MA).LE.D.) GO TO 300
1662 IF(IFS1.GT.IFS2) GO TO 250
1663 IF(IFS2.GT.IFS1) GO TO 260
1664 C
1665 C *** IF BOTH OR NEITHER CONTAIN A FREE SURFACE USE DENSITY
1666 C OF DONOR CELL TO DEFINE MASS FLUX.
1667 C
1668 245 IF(VABOVE.GT.C.)GO TO 260
1669 IF(J.EQ.JMAX) GO TO 300
1670 250 RHOT = RHOA
1671 GO TO 270
1672 260 RHOT = RHO(NT,M)
1673 VC=XMASS(NT,M)/RHOT

```

```

1674 C
1675 C      *** ACCUMULATE FLUXES ACROSS TOP BOUNDARY
1676 C
1677 270 SMPY=RHOT*VABOVE*FRAC(TP(NT,M))*SDT*CYC
1678 WS=CYCMX/CYC*ABS(SMPY)
1679 IF(WS.LT.ROEPS*RHOT*VC) SMPY=0.
1680 SAMPY(NT,M)=SMPY+SAMPY(NT,M)
1681 C
1682 C      *** CALCULATE FLUX OF MATERIAL N ACROSS RIGHT BOUNDARY.
1683 C      IF CELL ON RIGHT DOESNT CONTAIN MATERIAL N SKIP OUT.
1684 C
1685 300 RHOR=0.
1686 IF(I.EQ.IMAX .AND. U(K).GT.0.) GO TO 330
1687 IF(I.EQ.IMAX .AND. U(K).LE.0.) GO TO 500
1688 IF(MR.GT.0) GO TO 310
1689 C      *** CELL ON RIGHT PURE
1690 IF(MR+100 .NE. NT) GO TO 500
1691 RHOR = AMX(KR)/(TAU(I+1)*DY(J))
1692 VC=TAU(I+1)*DY(J)
1693 GO TO 320
1694 C      *** CELL ON RIGHT MIXED
1695 C      *** RHO(NT,MR)=0. INDICATES CELL DOES NOT CONTAIN
1696 C      MATERIAL NT.
1697 310 IF(RHO(NT,MR) .LE. 0.) GO TO 500
1698 RHOR = RHO(NT,MR)
1699 VC=XMASS(NT,MR)/RHOR
1700 C      *** IF CELL ON RIGHT CONTAINS A FREE SURFACE SET IFS2=1
1701 IFS2=0
1702 IF(RHO(NVOID,MR).GT.0.) IFS2 = 1
1703 IF(AMX(KR).LE.0.) GO TO 330
1704 320 IF(AMX(K).LE.0.) GO TO 335
1705 WSA=(U(K)+U(KR))*S
1706 WS=DT/DX(I)
1707 WSB=1.0+(U(KR)-U(K))*WS
1708 IF(ABS(U(K))*WS.GT.STAB .OR. ABS(U(KR))*WS.GT.STAB) WSB=1.0
1709 URR = WSA/WSB
1710 GO TO 340
1711 330 URR = U(K)
1712 GO TO 340
1713 335 URR=U(KR)
1714 C
1715 C      *** IF ONLY ONE CELL CONTAINS A FREE SURFACE,USE DENSITY
1716 C      OF THE OTHER CELL TO DEFINE MASS FLUX.
1717 340 IF(ABS(URR).LE.UNIN) GO TO 500
1718 IF(URR.GT.0. .AND. XMASS(NT,M).LE.0.) GO TO 500
1719 IF(MR.LE.0 .OR. I.EQ.IMAX) GO TO 345
1720 IF(URR.LT.0. .AND. XMASS(NT,MR).LE.0.) GO TO 500
1721 IF(IFS1.GT.IFS2) GO TO 350
1722 IF(IFS2.GT.IFS1) GO TO 360
1723 C
1724 C      *** IF BOTH OR NEITHER CELL CONTAIN A FREE SURFACE, USE
1725 C      DENSITY OF DONOR CELL TO DEFINE MASS FLUX.
1726 C
1727 345 IF(URR.GT.0.) GO TO 360

```

```

1728      IF(I.EQ.IMAX) GO TO 500
1729      350 RHOT = RHOR
1730      GO TO 370
1731      360 RHOT = RHO(INT,M)
1732      VC=XMASS(INT,M)/RHOT
1733      C
1734      C          *** ACCUMULATE FLUX ACROSS RIGHT BOUNDARY.
1735      C
1736      370 SMMP=RHOT*URR*FRACRT(INT,M)*SDT*CYC
1737      WS=CYCMX/CYC*ABS(SMMP)
1738      IF(WS.LT.ROEPS*RHOT*VC) SMMP=0.
1739      SAMMP(INT,M)=SMMP+SAMMP(INT,M)
1740      C
1741      C          *** END OF LOOP ON MATERIALS
1742      C
1743      500 CONTINUE
1744      RETURN
1745      END
1746      SUBROUTINE GLUE
1747      INCLUDE COMDIM
1748      C          *** ADJUST VELOCITIES OF CELLS OVER-ACCELERATED BY THE
1749      C          PRESSURE EFFECTS IN PHI BY 'GLUEING' THEM TO
1750      C          AN APPROPRIATE NEIGHBOR.
1751      C
1752      C          *** GLUE CELL -K- AND CELL -NK-
1753      C
1754      C          *** TWO PASSES THROUGH GLUE
1755      C          1 - GLUE ALL CELLS WITH UNREAL VELOCITIES TO
1756      C          HIGHEST PRESSURE NEIGHBOR.
1757      C          2 - GLUE ALL FREE SURFACE CELLS TO
1758      C          HIGHEST DENSITY NEIGHBOR.
1759      C
1760      DO 400 IG=1,2
1761      DO 300 J=1,12
1762      DO 300 I=1,11
1763      K=(J-1)*IMAX+I+1
1764      JP=J
1765      IF(J.LE.JPROJ .OR. JPROJ.EQ.0) GO TO 1409
1766      C          *** THE FOLLOWING REDEFINITION OF INDEX K IS NEEDED
1767      C          TO GIVE SYMMETRIC TREATMENT TO UPPER AND LOWER
1768      C          HALVES OF A SPHERICAL PACKAGE.
1769      JP=[2-J+JPROJ+1]
1770      K=(JP-1)*IMAX+I+1
1771      1409 IF(AMX(K).LE.0.) GO TO 300
1772      M=MFLAG(K)
1773      C          *** GLUE CELLS WITH UNREAL VELOCITIES (MFLAG NEGATIVE)
1774      C          OR WITH NEGATIVE INTERNAL ENERGY
1775      IF(M.LT.0 .OR. AIX(K).LT.0.)GO TO 100
1776      C
1777      IF(IG.EQ.1 .OR. M.LT.100) GO TO 300
1778      MM=M-100
1779      IF(RHO(INVOID,MM) .LE.0.) GO TO 300
1780      C
1781      100 NK=0

```

```

1782 C      *** DEFINE K-INDEX OF NEIGHBOR CELLS.
1783     KA=1
1784     KB=1
1785     KR=1
1786     KL=1
1787     IF(JP.NE.JMAX) KA=K+IMAX
1788     IF(JP.NE.1) KB=K-IMAX
1789     IF(I.NE.IMAX) KR=K+1
1790     IF(I.NE.1) KL=K-1
1791     IF(M.LT.0) GO TO 130
1792     IF(AIX(K).LT.0.)GO TO 110
1793 C
1794 C      *** ASSURE THAT F.S. CELLS ARE GLUED TO CELLS OF LIKE
1795 C      MATERIAL
1796 C
1797 C
1798     IF(JP.EQ.JMAX) GO TO 102
1799     MA=MFLAG(K+IMAX)
1800     IF(MA.GT.100) GO TO 101
1801     IF(XMASS(MA,MM).LE.0.) KA=1
1802     GO TO 102
1803 C
1804     101 MA=MA-100
1805     XM=0.
1806     DO 1015 N=1,NMAT
1807     1015 XM=XM+XMASS(N,MM)*XMASS(N,MA)
1808     IF(XM.GT.0.) GO TO 102
1809     KA=1
1810 C
1811     102 IF(I.EQ.IMAX) GO TO 104
1812     MR=MFLAG(K+1)
1813     IF(MR.GT.100) GO TO 103
1814     IF(XMASS(MR,MM).LE.0.) KR=1
1815     GO TO 104
1816 C
1817     103 MR=MR-100
1818     XM=0.
1819     DO 1035 N=1,NMAT
1820     1035 XM=XM+XMASS(N,MM)*XMASS(N,MR)
1821     IF(XM.GT.0.) GO TO 104
1822     KR=1
1823 C
1824     104 IF(JP.EQ.1) GO TO 106
1825     MB=MFLAG(K-IMAX)
1826     IF(MB.GT.100) GO TO 105
1827     IF(XMASS(MB,MM).LE.0.) KB=1
1828     GO TO 106
1829 C
1830     105 MB=MB-100
1831     XM=0.
1832     DO 1055 N=1,NMAT
1833     1055 XM=XM+XMASS(N,MM)*XMASS(N,MB)
1834     IF(XM.GT.0.) GO TO 106
1835     KB=1

```

```

1836 C
1837 106 IF(I.EQ.1) GO TO 108
1838 ML=MFLAG(K-1)
1839 IF(ML.GT.100) GO TO 107
1840 IF(XMASS(ML,MM).LE.0.) KL=1
1841 GO TO 108
1842 C
1843 107 ML=ML-100
1844 XM=0.
1845 DO 1075 N=1,NMAT
1846 1075 XM=XM+XMASS(N,MM)*XMASS(N,ML)
1847 IF(XM.GT.0.) GO TO 108
1848 KL=1
1849 108 CONTINUE
1850 GO TO 130
1851 C
1852 C *** GLUE CELLS WITH NEG. INTERNAL ENERGY TO NEIGHBOR
1853 C OF HIGHEST RELATIVE VELOCITY
1854 110 CONTINUE
1855 WS=0.
1856 IF(AMX(KA).LE.0.) GO TO 115
1857 WS = (U(K)-U(KA))**2 + (V(K)-V(KA))**2
1858 NK=KA
1859 115 IF(AMX(KR).LE.0.) GO TO 120
1860 WSA = (U(K)-U(KR))**2 + (V(K)-V(KR))**2
1861 IF(WSA.LT.WS) GO TO 120
1862 WS=WSA
1863 NK=KR
1864 120 IF(AMX(KB).LE.0.) GO TO 125
1865 WSA = (U(K)-U(KB))**2 + (V(K)-V(KB))**2
1866 IF(WSA.LT.WS) GO TO 125
1867 NK=KB
1868 125 IF(AMX(KL).LE.0.) GO TO 150
1869 WSA = (U(K)-U(KL))**2 + (V(K)-V(KL))**2
1870 IF(WSA.LT.WS) GO TO 150
1871 NK=KL
1872 GO TO 150
1873 C
1874 C *** GLUE CELLS WITH UNREAL VELOCITIES TO NEIGHBOR WITH
1875 C HIGHEST PRESSURE.
1876 C
1877 130 MFLAG(K)=ABS(MFLAG(K))
1878 PMAX = AMAX1(ABS(P(KA)),ABS(P(KR)),ABS(P(KB)),ABS(P(KL)))
1879 C
1880 IF(ABS(P(KA)).LT.PMAX) GO TO 135
1881 NK=KA
1882 GO TO 150
1883 135 IF(ABS(P(KR)).LT.PMAX) GO TO 140
1884 NK=KR
1885 GO TO 150
1886 140 IF(ABS(P(KB)).LT.PMAX) GO TO 145
1887 NK=KB
1888 GO TO 150
1889 145 NK=KL

```



```

1890 C
1891 150 IF(NK.LE.1) GO TO 300
1892 IF(ABS(U(NK)).LE.0. .AND. ABS(V(NK)).LE.0. .AND.
1893 ABS(AIX(NK)).LE.0.) GO TO 300
1894 C *** COMPUTE TOTAL KINETIC ENERGY OF THE TWO CELLS BEING
1895 C GLUED (K AND NK).
1896 WS1=AMX(K)*(U(K)**2+V(K)**2)+AMX(NK)*
1897 (U(NK)**2+V(NK)**2)
1898 C *** NEW VELOCITIES TO CONSERVE MOMENTUM
1899 WSU=(AMX(K)*U(K)+AMX(NK)*U(NK))/(AMX(K)+AMX(NK))
1900 WSV=(AMX(K)*V(K)+AMX(NK)*V(NK))/(AMX(K)+AMX(NK))
1901 C *** NEW KINETIC ENERGY
1902 WS2=(AMX(K)+AMX(NK))*(WSU**2+WSV**2)
1903 C *** GAIN IN KINETIC ENERGY (ALWAYS NEGATIVE)
1904 WS=(WS2-WS1)/2.
1905 ENER = AMX(K)*AIX(K) + AMX(NK)*AIX(NK) - WS
1906 AIX(K) = ENER/(AMX(K)+AMX(NK))
1907 AIX(NK) = AIX(K)
1908 C
1909 M=MFLAG(K)
1910 MFLAG(NK)=ABS(MFLAG(NK))
1911 MN=MFLAG(NK)
1912 IF(M.LT.100)GO TO 1197
1913 M=M-100
1914 DO 170 N=1,M,MAT
1915 IF(XMASS(N,M).LE.0.)GO TO 170
1916 SIE(N,M)=AIX(K)
1917 170 CONTINUE
1918 1197 IF(MN.LT.100)GO TO 197
1919 MN=MN-100
1920 DO 175 N=1,MN,MAT
1921 IF(XMASS(N,MN).LE.0.)GO TO 175
1922 SIE(N,MN)=AIX(NK)
1923 175 CONTINUE
1924 C *** NOTE - AFTER GLUEING, THE VELOCITY COMPONENTS AND THE
1925 C SPECIFIC INTERNAL ENERGIES OF THE TWO CELLS ARE EQUAL.
1926 197 U(K)=WSU
1927 U(NK)=WSU
1928 V(K)=WSV
1929 V(NK)=WSV
1930 WRITE(6,4000) I,JP,K,NK
1931 4000 FORMAT(2I6,5HCELLS, 15,5H AND,15,7H GLUED)
1932 300 CONTINUE
1933 400 CONTINUE
1934 C
1935 RETURN
1936 END
1937 INCLUDE COMDIM
1938 C
1939 C
1940 C *** INITIALIZE BLANK COMMON STORAGE
1941 LAST=1
1942 IQ=0
1943 5 IQ=IQ+1

```

```

1944      Z(IQ)=0.
1945      IF(LAST.GT.0) GO TO 5
1946      C
1947      CALL INPUT
1948      C      *** COMPUTE TIME STEP AND PRESSURES
1949      10 CALL CDT
1950      C      *** CHECK ENERGY CONSERVATION, PRINT, DUMP
1951      CALL EDIT
1952      C      *** WFLAGL - LAST CYCLE FLAG- SET IN EDIT
1953      IF(WFLAGL.GT.0) CALL EXIT
1954      C      *** WORK PHASE
1955      CALL PH1
1956      C      *** DEBUG EDIT PRINT
1957      IF(INTER.GT.0 .AND. NPRINT.GT.0) CALL EDIT
1958      C      *** DOES CALCULATION INVOLVE STRENGTH EFFECTS
1959      IF(CYCPH3.LT.0) GO TO 20
1960      C      *** STRENGTH PHASE
1961      CALL PH3
1962      C      *** DEBUG EDIT PRINT
1963      IF(INTER.GT.0 .AND. NPRINT.GT.0) CALL EDIT
1964      C      *** INTERFACE MOTION, MIXED CELL FLUXES
1965      20 CALL INFACE
1966      C      *** TRANSPORT PHASE
1967      CALL PH2
1968      C
1969      GO TO 10
1970      C
1971      END
1972      SUBROUTINE INFACE
1973      C      *** COMPUTE FRACTIONAL AREAS OF MIXED CELLS TO BE
1974      C      USED IN DEFINITION OF MASS FLUX TERMS.
1975      C      *** MOVE TRACER PARTICLES.
1976      C      *** CREATE MIXED CELLS.
1977      C      *** FLAG CELLS BECOMING PURE AND ADJUST THEIR
1978      C      FLUXES TO EXACTLY EVACUATE THOSE MATERIALS THAT
1979      C      ARE LEAVING.
1980      INCLUDE COMDIM
1981      EQUIVALENCE (MSY,RTY), (USA,TPX), (WSC,FRACK)
1982      ICY = INT(CYCMX)
1983      SDT=DT/CYCMX
1984      C
1985      C      *** BEGIN SUBCYCLE LOOP
1986      C
1987      C
1988      DO 875 LJ = 1,ICY
1989      CYC=FLOAT(LJ)-1.
1990      C
1991      C
1992      C      *** COMPUTE FRACTIONAL AREAS FROM INTERCEPTS WITH CELL
1993      C      BOUNDARIES USING UPDATED POSITIONS OF TRACERS.
1994      C      THESE AREAS WILL BE USED TO COMPUTE FLUXES ON THIS
1995      C      SUBCYCLE OF INFACE.
1996      C
1997      C      *** INITIALIZE ARRAYS

```

```

1998 C
1999 DO 24 N=1,NVOID
2000 DO 22 M=1,NMXCLS
2001 FRACFP(N,M) = 0.
2002 FRACRT(N,M) = 0.
2003 22 CONTINUE
2004 24 CONTINUE
2005 C      *** TRAVEL AROUND EACH MATERIAL PACKAGE AND USE PAIRS OF
2006 C      TRACERS THAT STRADDLE CELL BOUNDARY(S) TO COMPUTE
2007 C      FRACTIONAL AREAS.
2008 DO 36 N=1,NVOID
2009 NN=NMP(N)
2010 IF(NN.EQ.0)GO TO 36
2011 C
2012 C      *** STORE THE COORDINATES OF THE FIRST TRACER
2013 C      OF THE FIRST SUBPACKAGE.
2014 TX3= TX(N,1)
2015 TY3= TY(N,1)
2016 M1=1
2017 M2=2
2018 C      *** START LOOP ON M2
2019 200 TX1= TX(N,M1)
2020 TY1=TY(N,M1)
2021 TX2=TX(N,M2)
2022 TY2=TY(N,M2)
2023 C
2024 C      *** (TX3,TY3) IS THE FIRST POINT OF THIS PACKAGE
2025 C      OR SUBPACKAGE. IF TX2=TX3 AND TY2=TY3,
2026 C      THE PACKAGE HAS BEEN COMPLETELY CIRCUMSCRIBED.
2027 C
2028 IF(TX2.NE.TX3 .OR. TY2.NE.TY3) GO TO 211
2029 C      *** TEST TO SEE IF THERE ARE MORE SUBPACKAGES.
2030 IF(M2.GE.NN) GO TO 36
2031 C      *** WILL NEXT TRACER BE EQUAL TO THE FIRST ONE
2032 IF(TX(N,M2+1).NE.TX3 .OR. TY(N,M2+1).NE.TY3) GO TO 34
2033 C      *** PRESET FRACTIONAL AREAS OF CELL K HERE IF WE ENTER
2034 C      IT FROM AN AXIS.
2035 211 IF(TX1.LE.0. .AND. TX2.GT.0.) GO TO 212
2036 IF(TY1.LE.0. .AND. TY2.GT.0.) GO TO 213
2037 GO TO 215
2038 212 J= INT(TY1)+1
2039 IF(J.GT.12) GO TO 215
2040 MXAXIS=M1
2041 K=(J-1)*IMAX+2
2042 M=IABS(MFLAG(K))
2043 IF(M.GT.100) GO TO 2120
2044 M0=M
2045 I=1
2046 CALL NEWMIX
2047 2120 M=M-100
2048 FRACFP(N,M)=TAU(I)
2049 FRACRT(N,M)=DY(J)*X(I)*TWOPI
2050 IF(I.GH.EQ.1)FRACRT(N,M)=DY(J)
2051 MR=IABS(MFLAG(K-IMAX))-100

```

```

2052      IF(MR.LT.0 .OR. J.EQ.1) GO TO 215
2053      FRACFP(N,MB)=TAU(I)
2054      GO TO 215
2055      213 I= INT(TX1)+1
2056      IF(I.GT.11) GO TO 215
2057      MYAXIS=M1
2058      K=I+1
2059      M=IABS(MFLAG(K))
2060      IF(M.GT.100) GO TO 2130
2061      NO=M
2062      J=1
2063      CALL NEWMIX
2064      2130 M=M-100
2065      ML=IABS(MFLAG(K-1))-100
2066      IF(ML.LT.0 .OR. I.EQ.1) GO TO 214
2067      FRACRT(N,ML)=DY(I)*X(I-1)*TWOPI
2068      IF(IGH.EQ.1)FRACRT(N,ML)=DY(I)
2069      214 FRACFP(N,M)=TAU(I)
2070      FRACRT(N,M)=DY(I)*X(I)*TWOPI
2071      IF(IGH.EQ.1)FRACRT(N,M)=DY(I)
2072      215 CONTINUE
2073      C      ***IF BOTH POINTS ARE ON THE SAME AXIS. SKIP OUT
2074      IF(TY1+TY2.LE.0.)GO TO 33
2075      IF(TX1+TX2 .LE. 0.) GO TO 33
2076      C
2077      ITX1 = INT(TX1)
2078      ITY1 = INT(TY1)
2079      ITX2 = INT(TX2)
2080      ITY2 = INT(TY2)
2081      ITXB=ITX1
2082      ITYB=ITY1
2083      C      *** IF BOTH POINTS ARE OUTSIDE THE ACTIVE GRID, SKIP OUT.
2084      IF( (TX1.GT.FLOAT(I1)).OR.TY1.GT.FLOAT(I2)) .AND.(TX2.GT.FLOAT(I1)
2085      1 .OR.TY2.GT.FLOAT(I2)) ) GO TO 33
2086      C      *** IF BOTH POINTS ARE IN THE SAME CELL, SKIP OUT.
2087      23 IF( (ITX1.EQ.ITX2 .AND. ITY1.EQ.ITY2) GO TO 33
2088      I=ITX1
2089      IF( (ITX1.LT.ITX2) I=I+1
2090      J=ITY1
2091      IF( (ITY1.LT.ITY2) J=J+1
2092      RTX=FLOAT(I)
2093      TPY=FLOAT(J)
2094      C      *** IF CELL DIMENSIONS ARE CONSTANT, CAN USE CELL UNITS TO
2095      C      COMPUTE SLOPE - OTHERWISE, MUST CONVERT TRACER
2096      C      COORDINATES TO CM. UNITS.
2097      IF( (IVARDX.EQ.0 .AND. IVARDY.EQ.0) GO TO 231
2098      C      *** COMPUTE CM. VALUES OF COORDINATES
2099      XCM1= X(ITXH) + (ITX1-AINT(ITX1))*DX(ITXB+1)
2100      XCM2=X(ITX2)+(ITX2-AINT(ITX2))*DX(ITX2+1)
2101      YCM1= Y(ITYB) + (ITY1-AINT(ITY1))*DY(ITYB+1)
2102      YCM2=Y(ITY2)+(ITY2-AINT(ITY2))*DY(ITY2+1)
2103      C      *** COMPUTE SLOPE OF LINE THROUGH THESE TWO POINTS
2104      SLOPE=(YCM2-YCM1)/(XCM2-XCM1)
2105      C      *** COMPUTE INTERSECTIONS WITH CELL BOUNDARIES

```

```

2106      RTY=YCM2+SLOPE*(X(I)-XCM2)
2107      TPX=XCM2+(Y(J)-YCM2)/SLOPE
2108      IT=I
2109      JT=J
2110  C          *** CONVERT INTERSECTIONS TO CELL UNITS
2111      IF(I*TY1.EQ.*ITY2) GO TO 235
2112      DO 234 I=1,11
2113      IF(TPX-X(I)) 232,233,234
2114  232  TPX=FLOAT(I-1)+(TPX-X(I-1))/(X(I)-X(I-1))
2115      GO TO 235
2116  233  TPX=FLOAT(I)
2117      GO TO 235
2118  234  CONTINUE
2119      NK=234
2120      NR=9
2121      CALL ERROR
2122  235  CONTINUE
2123  C
2124      IF(ITX1.EQ.*ITX2) GO TO 239
2125      DO 238 J=1,12
2126      IF(RTY-Y(J)) 236,237,238
2127  236  RTY=FLOAT(J-1)+(RTY-Y(J-1))/(Y(J)-Y(J-1))
2128      GO TO 239
2129  237  RTY=FLOAT(J)
2130      GO TO 239
2131  238  CONTINUE
2132      NK=238
2133      NR=9
2134      PRINT 7777, RTY,Y(J),Y(12),SLOPE,YCM1,YCM2,ITY1,ITY2
2135  7777  FORMAT (I11,8E16.8)
2136      CALL ERROR
2137  239  CONTINUE
2138      I=IT
2139      J=JT
2140      GO TO 239D
2141  C
2142  231  SLOPE=(ITY2-TY1)/(TX2-TX1)
2143      RTY=TY2+SLOPE*(RTX-TX2)
2144      TPX=TX2+(TPY-TY2)/SLOPE
2145  239D  FRACY=0.
2146      FRACX=0.
2147      IF(I*TY1.EQ.*ITY2) GO TO 242
2148      IF(ITX1.EQ.*ITX2) GO TO 28
2149  C          *** IF POINTS STRADDLE BOTH A TOP AND A RIGHT CELL
2150  C          BOUNDARY, PROCESS FIRST THE ONE CLOSEST TO THE FIRST
2151  C          TRACER (TX1, TY1).
2152      DRT=SQRT((RTX-TX1)**2 + (RTY-TY1)**2)
2153      DTP=SQRT((TPX-TX1)**2 + (TPY-TY1)**2)
2154      IF(DRT.GT.DTP) GO TO 28
2155  C
2156  242  FRACY=RTY-AINT(RTY)
2157      J=INT(RTY)
2158      IF(FRACY.GT.0.) J=J+1
2159      IF(FRACY.LE.0. .AND. ITX1.LT.ITX2) J=J+1

```

```

2160      IF(I.LE.0 .OR. J.LE.0 .OR. I.GT.IMAX .OR. J.GT.JMAX) GO TO 33
2161      K=(J-1)*IMAX+1+1
2162      M=IABS(MFLAG(K))
2163      MB=IABS(MFLAG(K-IMAX))-100
2164      ML=IABS(MFLAG(K-1))-100
2165      KR=K+1
2166      MBR=IABS(MFLAG(KR-IMAX))-100
2167      C      *** IF CELL ON LEFT OF POINT WAS PREVIOUSLY A PURE CELL,
2168      C      CALL NEAMIX TO SET UP STORAGE FOR IT IN MIXED CELL
2169      C      ARRAYS, AND COMPUTE FLUX FOR SUBCYCLES COMPLETED.
2170      IF(M.GT.100) GO TO 240
2171      MO=M
2172      CALL NEWMIX
2173      C      *** IF MATERIAL N JUST ENTERED CELL ON LEFT OF POINT,
2174      C      CALL NEVRHO TO ASSIGN A DENSITY TO IT.
2175      240 M=M-100
2176      IF(RHO(N,M).LE.0. .AND. N.NE.NVOID) CALL NEVRHO
2177      IF(N.EQ.NVOID) RHO(NVOID,M)=1.0
2178      IF(I.GE.IMAX) GO TO 25
2179      MR=IABS(MFLAG(KR))
2180      C      *** IF CELL ON RIGHT OF POINT WAS PREVIOUSLY A PURE CELL,
2181      C      CALL NEWMIX TO SET UP STORAGE FOR IT IN MIXED CELL
2182      C      ARRAYS, AND COMPUTE FLUX FOR SUBCYCLES COMPLETED.
2183      IF(MR.GT.100) GO TO 241
2184      MO=MR
2185      MT=M
2186      IT=1
2187      KT=K
2188      K=KR
2189      J=J+1
2190      CALL NEWMIX
2191      MR=M
2192      I=IT
2193      K=KT
2194      M=MT
2195      C      *** IF MATERIAL N JUST ENTERED CELL ON RIGHT OF POINT,
2196      C      ASSIGN TO IT THE SAME DENSITY AS THAT ASSIGNED TO
2197      C      CELL ON LEFT.
2198      241 MR=MR-100
2199      IF(RHO(N,MR).LE.0.) RHO(N,MR)=RHO(N,M)
2200      C      *** COMPUTE FRACTIONAL AREA FOR RADIAL TRANSPORT OF
2201      C      MATERIAL N.
2202      25 IF(ITX1.LT.ITX2) GO TO 26
2203      C      *** ENTERING CELL K - LEAVING CELL KR
2204      WS = (RTY-FLOAT(J-1))*DY(J)*X(I)*TWOPI
2205      IF(IGH.EQ.1) WS=(RTY-FLOAT(J-1))*DY(J)
2206      C      *** PRESET FRACTIONAL AREAS OF CELL K WHICH WE ARE ENTERING
2207      WSA=DY(J)*X(I)*TWOPI
2208      IF(IGH.EQ.1) WSA=DY(J)
2209      IF(FRACT(N,M).LT.WSA .AND. FRACT(N,M)+WS .GT. WSA) GO TO 253
2210      IF(MB.LT.0 .OR. J.EQ.1) GO TO 251
2211      IF(FRACTP(N,MB).GT.0. .AND. FRACTP(N,MB).LT.TAU(I)) GO TO 255
2212      FRACTP(N,MB) = TAU(I)
2213      251 IF(ML.LT.0 .OR. I.EQ.1) GO TO 252

```

```

2214      WSA = DY(J)*X(I-1)*TWOPI
2215      IF(IGM.EQ.1)WSA=DY(J)
2216      IF(FRACRT(N,ML).GT.0. .AND. FRACRT(N,ML).LT.WSA) GO TO 255
2217      FRACRT(N,ML) = WSA
2218 252 IF(FRACTP(N,M).GT.0. .AND. FRACTP(N,M).LT.TAU(I)) GO TO 255
2219      FRACTP(N,M)=TAU(I)
2220 255 WSA = DY(J)*X(I)*TWOPI
2221      IF(IGM.EQ.1)WSA=DY(J)
2222  C      *** RESET FRACTIONAL AREAS OF CELL KR WHICH WE ARE LEAVING
2223      IF(FRACRT(N,M).GT.0. .AND. FRACRT(N,M)+WS .LT.WSA) GO TO 256
2224 253 IF(WS.GE.VSA.OR.I.EQ.IMAX)GO TO 256
2225      IF(FRACTP(N,MR).GT.0. .AND. FRACTP(N,MR).LT.TAU(I+1)) GO TO 256
2226      FRACTP(N,MR)=0.
2227      WSA = DY(J)*X(I+1)*TWOPI
2228      IF(IGM.EQ.1)WSA=DY(J)
2229      IF(FRACRT(N,MR).GT.0. .AND. FRACRT(N,MR).LT.WSA) GO TO 256
2230      FRACRT(N,MR) = 0.
2231      IF(MBR.LT.0 .OR. J.EQ.1) GO TO 256
2232      IF(FRACTP(N,MR).GT.0. .AND. FRACTP(N,MR).LT.TAU(I+1)) GO TO 256
2233      FRACTP(N,MR) = 0.
2234 256 ITX|=ITX|-1
2235      GO TO 275
2236  C      *** ENTLRING CELL KR - LEAVING CELL K
2237 26 WS = (FLOAT(J)-RTY)*DY(J)*X(I)*TWOPI
2238      IF(IGM.EQ.1)WS=(FLOAT(J)-RTY)*DY(J)
2239  C      PRFSET FRACTIONAL AREAS OF CELL KR
2240      IF(I.GE.IMAX) GO TO 265
2241      WSA=DY(J)*X(I)*TWOPI
2242      IF(IGM.EQ.1)WSA=DY(J)
2243      IF(FRACRT(N,M).LT.WSA .AND. FRACRT(N,M)+WS .GT. WSA) GO TO 263
2244      IF(FRACTP(N,MR).GT.0. .AND. FRACTP(N,MR).LT.TAU(I+1)) GO TO 265
2245      FRACTP(N,MR) = TAU(I+1)
2246      WSA = DY(J)*X(I+1)*TWOPI
2247      IF(IGM.EQ.1)WSA=DY(J)
2248      IF(FRACRT(N,MR).GT.0. .AND. FRACRT(N,MR).LT.WSA) GO TO 265
2249      FRACRT(N,MR) = WSA
2250      IF(MBR.LT.0 .OR. J.EQ.1) GO TO 265
2251      IF(FRACTP(N,MR).GT.0. .AND. FRACTP(N,MR).LT.TAU(I+1)) GO TO 265
2252      FRACTP(N,MR) = TAU(I+1)
2253  C      *** RESET FRACTIONAL AREAS OF CELL K WHICH WE ARE LEAVING
2254 265 WSA = DY(J)*X(I)*TWOPI
2255      IF(IGM.EQ.1)WSA=DY(J)
2256      IF(FRACRT(N,M).GT.0. .AND. FRACRT(N,M)+WS .LT. WSA) GO TO 27
2257      IF(INT(TX(N,MYAXIS))+1.EQ.I.AND.J.EQ.1) GO TO 27
2258 263 IF(MB.LT.0 .OR. J.EQ.1) GO TO 266
2259      IF(FRACTP(N,MR).GT.0. .AND. FRACTP(N,MR).LT.TAU(I)) GO TO 27
2260      FRACTP(N,MR) = 0.
2261 266 WSA = DY(J)*X(I-1)* TWOPI
2262      IF(IGM.EQ.1)WSA=DY(J)
2263      IF(INT(TY(N,MXAXIS))+1.EQ.J.AND.I.EQ.1) GO TO 27
2264      IF(ML.LT.0 .OR. I.EQ.1) GO TO 267
2265      IF(FRACRT(N,ML).GT.0. .AND. FRACRT(N,ML).LT.WSA) GO TO 27
2266      FRACRT(N,ML) = 0.
2267 267 IF(FRACTP(N,M).GT.0. .AND. FRACTP(N,M).LT.TAU(I)) GO TO 27

```

```

2268      FRACFP(N,M)=0.
2269      27 ITXI=ITXI+1
2270      C      *** UPDATE FRACRT(N,M)
2271      275 FRACRT(N,M) = FRACRT(N,M)+WS
2272      WSA = DY(J)*X(I)*TWOPI
2273      IF(IGM.EQ.1)WSA=DY(J)
2274      IF(FRACRT(N,M).GT.WSA) FRACRT(N,M)=FRACRT(N,M)-WSA
2275      C      *** AFTER INCREMENTING ITXI, GO BACK TO 23 TO SEE IF
2276      C      MORE CELL BOUNDARIES BETWEEN THESE TWO TRACERS
2277      C      NEED TO BE PROCESSED.
2278      GO TO 23
2279      C
2280      C      *** IF THESE TWO PTS STRADDLE A TOP BOUNDARY,
2281      C      FIND INTERCEPT AND CALCULATE A FRACTIONAL AREA (FRACFP)
2282      C
2283      28 FRACX=TPX-AINT(TPX)
2284      I=INT(TPX)
2285      IF(FRACX.GT.0.) I=I+1
2286      IF(FRACX.LE.0. .AND. ITY1.GT.ITY2) I=I+1
2287      IF(I.LE.0 .OR. J.LE.0 .OR. I.GT.IMAX .OR. J.GT.JMAX) GO TO 33
2288      K=(J-1)*IMAX+I+1
2289      M=IABS(MFLAG(K))
2290      MB=IABS(MFLAG(K-IMAX))-100
2291      ML=IABS(MFLAG(K-1))-100
2292      KA=K+IMAX
2293      MLA=IABS(MFLAG(KA-1))-100
2294      C      *** IF CELL BELOW POINT WAS PREVIOUSLY A PURE CELL,
2295      C      CALL NEWMIX TO SET UP STORAGE FOR IT IN MIXED CELL
2296      C      ARRAYS, AND COMPUTE FLUX FOR SUBCYCLES COMPLETED.
2297      IF(M.GT.100)GO TO 28D
2298      MD=M
2299      CALL NEWMIX
2300      C      *** IF MATERIAL N JUST ENTERED CELL BELOW POINT,
2301      C      CALL NEWRHO TO ASSIGN A DENSITY TO IT.
2302      28D M=M-100
2303      IF(RHO(N,M).LE.0. .AND. N.NE.NVOID) CALL NEWRHO
2304      IF(N.EQ.NVOID) RHO(NVOID,M)=1.0
2305      IF(J.GE.JMAX) GO TO 29
2306      MA=IABS(MFLAG(KA))
2307      C      *** IF CELL ABOVE POINT WAS PREVIOUSLY A PURE CELL,
2308      C      CALL NEWMIX TO SET UP STORAGE FOR IT IN MIXED CELL
2309      C      ARRAYS, AND COMPUTE FLUX FOR SUBCYCLES COMPLETED.
2310      IF(MA.GT.100) GO TO 28I
2311      MO=MA
2312      MT=M
2313      JT=J
2314      KT=K
2315      K=KA
2316      J=J+1
2317      CALL NEWMIX
2318      MA=M
2319      M=MT
2320      K=KT
2321      J=JT

```



```

2322 C          *** IF MATERIAL N JUST ENTERED CELL ABOVE POINT,
2323 C          ASSIGN TO IF THE SAME DENSITY AS THAT ASSIGNED TO
2324 C          CELL BELOW.
2325 281 MA=MA-100
2326 IF(RHO(N,MA).LE.0.) RHO(N,MA)=RHO(N,M)
2327 C          *** IS MATERIAL ON RIGHT OR LEFT OF INTERCEPT
2328 29 WSX=INT(IPX)
2329 WSX=X(1:WSX) + DX(1:WSX+1)*FRACX
2330 IF(IT1.LT.IT2) GO TO 295
2331 C          *** ENTERING CELL K - LEAVING CELL KA
2332 WS=(X(1)**2 - WSX**2)*PIDY
2333 IF(IGM.EQ.1)WS=X(1)-WSX
2334 C          *** PRESET FRACTIONAL AREAS OF CELL K WHICH WE ARE ENTERING
2335 IF(FRACT(N,M).LT.TAU(1) .AND. FRACTP(N,M)+WS .GT.TAU(1))GO TO 293
2336 WSA=DY(J)*X(1)*TWOPI
2337 IF(IGM.EQ.1)WSA=DY(J)
2338 IF(FRACT(N,M).GT.0. .AND. FRACT(N,M).LT.WSA) GO TO 293
2339 FRACT(N,M)=WSA
2340 IF(MB.LT.0 .OR. J.EQ.1) GO TO 292
2341 IF(FRACTP(N,MB).GT.0. .AND. FRACTP(N,MB).LT.TAU(1)) GO TO 293
2342 FRACTP(N,MB) = TAU(1)
2343 292 IF(ML.LT.0 .OR. I.EQ.1) GO TO 293
2344 WSA=DY(J)*X(I-1)*TWOPI
2345 IF(IGM.EQ.1)WSA=DY(J)
2346 IF(FRACT(N,ML).GT.0. .AND. FRACT(N,ML).LT.WSA) GO TO 293
2347 FRACT(N,ML)=WSA
2348 C          *** RESET FRACTIONAL AREAS OF CELL KA WHICH WE ARE LEAVING
2349 293 IF(FRACTP(N,M).GT.0. .AND. FRACTP(N,M)+WS .LT. TAU(1)) GO TO 299
2350 IF(J.EQ.JMAX)GO TO 299
2351 IF(INT(IT(N,MXAX(5))).EQ.J.AND.I.EQ.1) GO TO 299
2352 IF(MLA.LT.0 .OR. I.EQ.1) GO TO 294
2353 WSA=DY(J+1)*X(I-1)*TWOPI
2354 IF(IGM.EQ.1)WSA=DY(J+1)
2355 IF(FRACT(N,MLA).GT.0. .AND. FRACT(N,MLA).LT.WSA) GO TO 299
2356 FRACT(N,MLA)=0.
2357 294 IF(FRACTP(N,MA).GT.0. .AND. FRACTP(N,MA).LT.TAU(1)) GO TO 299
2358 FRACTP(N,MA)=0.
2359 2941 WSA=DY(J+1)*X(I)*TWOPI
2360 IF(IGM.EQ.1)WSA=DY(J+1)
2361 IF(FRACT(N,MA).GT.0. .AND. FRACT(N,MA).LT.WSA) GO TO 299
2362 FRACT(N,MA)=0.
2363 299 ITY1=ITY1-1
2364 GO TO 31
2365 C          *** ENTERING CELL KA - LEAVING CELL K
2366 295 WS=(WSX**2 - X(I-1)**2)*PIDY
2367 IF(IGM.EQ.1)WS=WSX-X(I-1)
2368 C          PRESET FRACTIONAL AREAS OF CELL KA
2369 IF(J.GE.JMAX) GO TO 297
2370 IF(FRACTP(N,M).LT.TAU(1) .AND. FRACTP(N,M)+WS .GT.TAU(1))GO TO 297
2371 IF(MLA.LT.0 .OR. I.EQ.1) GO TO 296
2372 WSA=DY(J+1)*X(I-1)*TWOPI
2373 IF(IGM.EQ.1)WSA=DY(J+1)
2374 IF(FRACT(N,MLA).GT.0. .AND. FRACT(N,MLA).LT.WSA) GO TO 297
2375 FRACT(N,MLA)=WSA

```

```

2376      296 IF(FRACFP(N,MA).GT.0. .AND. FRACFP(N,MA).LT.TAU(I)) GO TO 297
2377      FRACFP(N,MA)=TAU(I)
2378      WSA=DY(J+1)*X(I)*THOPI
2379      IF(IGH.EQ.1)WSA=DY(J+1)
2380      IF(FRACRT(N,MA).GT.0. .AND. FRACRT(N,MA).LT.WSA) GO TO 297
2381      FRACRT(N,MA) = WSA
2382      C      *** RESET FRACTIONAL AREAS OF CELL K WHICH WE ARE LEAVING
2383      297 IF(FRACFP(N,M).GT.0. .AND. FRACFP(N,M)+WS .LT. TAU(I)) GO TO 30
2384      IF(WS.GE.TAU(I)) GO TO 30
2385      WSA=DY(J)*X(I)*TWOPI
2386      IF(IGH.EQ.1)WSA=DY(J)
2387      IF(FRACRT(N,M).GT.0. .AND. FRACRT(N,M).LT.WSA) GO TO 30
2388      FRACRT(N,M)=0.
2389      IF(INT(TXIN,NYAXIS)+1.EQ.1.AND.J.EQ.1) GO TO 30
2390      IF(MB.LT.0 .OP. J.EQ.1) GO TO 298
2391      IF(FRACFP(N,MB).GT.0. .AND. FRACFP(N,MB).LT.TAU(I)) GO TO 30
2392      FRACFP(N,MB) = 0.
2393      298 IF(ML.LT.0 .OR. I.EQ.1) GO TO 30
2394      WSA=DY(J)*X(I-1)*TWOPI
2395      IF(IGH.EQ.1)WSA=DY(J)
2396      IF(FRACRT(N,ML).GT.0. .AND. FRACRT(N,ML).LT.WSA) GO TO 30
2397      FRACRT(N,ML)=0.
2398      30 ITYI=ITYI+1
2399      C      *** UPDATE FRACFP(N,M)
2400      31 FRACFP(N,M)=FRACFP(N,M)+WS
2401      IF(FRACFP(N,M).GT.TAU(I))FRACFP(N,M)=FRACFP(N,M)-TAU(I)
2402      32 CONTINUE
2403      C      *** AFTER INCREMENTING ITYI, GO BACK TO 23 TO SEE IF
2404      C      MORE CELL BOUNDARIES BETWEEN THESE TWO TRACERS
2405      C      NEED TO BE PROCESSED.
2406      GO TO 23
2407      C      *** COME HERE WHEN ALL BOUNDARIES BETWEEN TWO POINTS HAVE
2408      C      BEEN PROCESSED OR WHEN TWO POINTS ARE IN THE SAME CELL.
2409      33 CONTINUE
2410      C      *** RESET FRACTIONAL AREAS OF CELL K HERE IF WE LEAVE IT
2411      C      ON AN AXIS.
2412      IF(TX2.LE.0. .AND. TX2.LT.TX1) GO TO 342
2413      IF(TY2.LE.0. .AND. TY2.LT.TY1) GO TO 344
2414      GO TO 35
2415      342 J=INT(TY2)+1
2416      IF(J.EQ.INT(TY3)+1) GO TO 35
2417      IF(J.GT.I2) GO TO 35
2418      K=(J-1)*IMAX+2
2419      M=IABS(MFLAG(K))-100
2420      IF(FRACFP(N,M).LT.TAU(I)) GO TO 35
2421      FRACFP(N,M)=0.
2422      WSA=DY(J)*X(I)*TWOPI
2423      IF(IGH.EQ.1)WSA=DY(J)
2424      IF(FRACRT(N,M).LT.WSA)GO TO 35
2425      FRACRT(N,M)=0.
2426      GO TO 35
2427      344 I=INT(TX2)+1
2428      IF(I.GT.I1) GO TO 35
2429      K=I+1

```

```

2430      M=IAPS(MFLAG(K))-100
2431      NL=IAPS(MFLAG(K-1))-100
2432      IF(ML.LT.0 .OR. I.EQ.1) GO TO 346
2433      WSA=DY(J)*X(I)-1)*TWOPI
2434      IF(IGM.EQ.1)WSA=DY(J)
2435      IF(FRACRT(N,NL).LT.WSA)GO TO 35
2436      FRACRT(N,NL)=C.
2437      346 IF(FRACTP(N,M).LT.TAU(I)) GO TO 35
2438      FRACTP(N,M)=C.
2439      GO TO 35
2440      C
2441      C
2442      C      *** REINITIALIZE FOR NEXT SUBPACKAGE.
2443      34 M2=M2+1
2444      TX3= TX(N,M2)
2445      TY3= TY(N,M2)
2446      C
2447      C      *** INCREMENT THE TRACER INDICES.
2448      35 M1=M2
2449      M2=M2+1
2450      C      *** LOOP BACK TO BEGINNING.
2451      IF(M2.LE.NN) GO TO 200
2452      C
2453      C      *** IF WE FALL THROUGH TO THIS POINT, IT MEANS THAT THE LAST
2454      C      SUBPACKAGE OF THE NTH PACKAGE DID NOT MAKE A COMPLETE LOOP.
2455      WRITE (6,9100) N
2456      9100 FORMAT (68H0**** INFACE DETECTED AN INCOMPLETE SUBPACKAGE OF MATE
2457      *RIAL PACKAGE ,13/31H0 CHECK REMAINING PACKAGES.)
2458      PRINT 9102,M1,M2,NN,TX1,TY1,TX2,TY2,TX3,TY3
2459      9102 FORMAT(10H0M1,M2,NN=,316/25H0TX1,TY1,TX2,TY2,TX3,TY3=,
2460      * 311X,1P2E17.6))
2461      CALL EXIT
2462      C      *** END OF LOOP ON MATERIALS (N)
2463      36 CONTINUE
2464      C
2465      IF(INTER.NE.9) GO TO 362
2466      DO 361 K=2,NMAX
2467      M=MFLAG(K)
2468      IF(M.LT.100) GO TO 361
2469      M=N-100
2470      J=(K-2)/IMAX+1
2471      I=(K-1)-IMAX*(J-1)
2472      WS=DY(J)*X(I)*TWOPI
2473      IF(IGM.EQ.1)WS=DY(J)
2474      WRITE(6,1666) I,J,TAU(I),WS,(FRACTP(N,M),FRACRT(N,M),N=1,NVOID)
2475      1666 FORMAT(19H I,J,TAU,XDY2P1, = ,216,1P2E20.8/10X,10H FRACTP,10X,
2476      1 10H FRACRT/(1P2E20.8))
2477      361 CONTINUE
2478      362 CONTINUE
2479      C      *** MAKE FLAGS .GT. 100 NEGATIVE. THEN MAKE POSITIVE
2480      C      FLAGS OF CELLS WITH AN INTERCEPT ON ONE OF ITS
2481      C      BOUNDARIES. THUS AFTER THIS LOOP, A CELL WITH
2482      C      A NEGATIVE FLAG NO LONGER IS CUT BY AN INTERFACE
2483      C      AND HAS BECOME PURE. ITS FLAG WILL NOT BE REDEFINED

```

```

2484 C UNTIL THE END OF PH2.
2485 C
2486 DO 382 I=1,I1
2487 DO 38 J=1,I2
2488 K=(J-1)*IMAX+I+1
2489 M=IABS(MFLAG(K))
2490 IF(M.LT.100) GO TO 38
2491 MFLAG(K)=-M
2492 MB=IABS(MFLAG(K-IMAX))-100
2493 ML=IABS(MFLAG(K-1))-100
2494 M=M-100
2495 C
2496 DO 37 N=1,NVOID
2497 IF(N.NE.NVOID .AND. XMASS(N,M).LE.C.) GO TO 37
2498 WSA=FRACFP(N,M)
2499 WSY=TAU(I)
2500 IF(WSA.GT.0. .AND. WSA.LT.WSY) GO TO 378
2501 WSB=FRACRT(N,M)
2502 WSX=DY(J)*X(I)*TAOP1
2503 IF(IGM.EQ.1)WSX=DY(J)
2504 IF(WSB.GT.0. .AND. WSB.LT.WSX) GO TO 378
2505 IF(IMB.LT.0 .OR. J.EQ.1) GO TO 377
2506 WSA=FRACFP(N,MB)
2507 IF(WSA.GT.0. .AND. WSA.LT.WSY) GO TO 378
2508 377 IF(ML.LT.0 .OR. I.EQ.1) GO TO 37
2509 WSB=FRACRT(N,ML)
2510 WSX=DY(J)*X(I-1)*TWOP1
2511 IF(IGM.EQ.1)WSX=DY(J)
2512 IF(WSB.GT.0. .AND. WSB.LT.WSX) GO TO 378
2513 37 CONTINUE
2514 GO TO 38
2515 378 MFLAG(K)=M+100
2516 38 CONTINUE
2517 382 CONTINUE
2518 IF(INTER.EQ.0) GO TO 380
2519 WRITE(6,678)
2520 DO 384 J=1,JMAX
2521 NB=(J-1)*IMAX+2
2522 NE=NB+IMAX-1
2523 WRITE(6,679) (MFLAG(K),K=NB,NE)
2524 384 CONTINUE
2525 678 FORMAT(12H MFLAG ARRAY/)
2526 679 FORMAT(32I4)
2527 380 CONTINUE
2528 C *** LOOK FOR CELLS THAT WILL BE MIXED ON NEXT CYCLE
2529 C (POSITIVE FLAGS), BUT SHOULD BE EVACUATED OF ONE OR
2530 C MORE MATERIALS. THE DENSITY OF MATERIALS TO BE
2531 C EVACUATED WILL BE SET TO ZERO.
2532 DO 390 I=1,I1
2533 DO 39 J=1,I2
2534 K=(J-1)*IMAX+I+1
2535 M=MFLAG(K)
2536 C *** IF FLAG IS NEGATIVE, EVACUATION FLAGS SET IN ANOTHER
2537 C LOOP.

```

```

2538      IF(M.LT.100) GO TO 39
2539      M=M-100
2540      MB=ABS(MFLAG(K-IMAX))-100
2541      ML=ABS(MFLAG(K+1))-100
2542      DO 385 N=1,MMAT
2543      IF(XMASS(N,M).LE.0.) GO TO 385
2544      WSA=FRACFP(N,M)
2545      WSY=TAU(I)
2546      IF(WSA.GT.0. .AND. WSA.LT.WSY) GO TO 385
2547      WSR=FRACRT(N,M)
2548      WSX=DY(J)*X(I)*TWOPI
2549      IF(IGM.EQ.1)WSX=DY(J)
2550      IF(WSB.GT.0. .AND. WSB.LT.WSX) GO TO 385
2551      IF(MB.LT.0 .OR. J.EQ.1) GO TO 381
2552      WSA=FRACFP(N,MB)
2553      IF(WSA.GT.0. .AND. WSA.LT.WSY) GO TO 385
2554      381 IF(ML.LT.0 .OR. J.EQ.1) GO TO 383
2555      WSR=FRACRT(N,ML)
2556      WSX=DY(J)*X(I-1)*TWOPI
2557      IF(IGM.EQ.1)WSX=DY(J)
2558      IF(WSB.GT.0. .AND. WSR.LT.WSX) GO TO 385
2559      383 RHO(N,M)=0.
2560      385 CONTINUE
2561      39 CONTINUE
2562      390 CONTINUE
2563      C
2564      C      *** REDEFINE FRACFP AND FRACRT FOR CELLS THAT ARE
2565      C      NO LONGER MIXED (FLAG NEGATIVE) BUT WILL STILL
2566      C      BE PROCESSED AS MIXED CELLS UNTIL THE END OF PH2.
2567      C
2568      DO 480 I=1,II
2569      DO 48 J=1,I2
2570      K=(J-1)*IMAX+I+1
2571      C      *** IF CELL HAS NEGATIVE FLAG IT LOST ALL INTERFACES
2572      C      DURING THIS SUBCYCLE
2573      MN=0
2574      NM=0
2575      IF(MFLAG(K).GE.0) GO TO 48
2576      MA=ABS(MFLAG(K+IMAX))
2577      MR=ABS(MFLAG(K+1))
2578      MB=ABS(MFLAG(K-IMAX))
2579      ML=ABS(MFLAG(K-1))
2580      C      *** CELL WILL BE PURE OF SAME MATERIAL AS ONE OF ITS PURE
2581      C      NEIGHBORS OR ONE OF ITS MIXED NEIGHBORS ON THE
2582      C      LEFT OR BELOW.
2583      IF(I.EQ.IMAX .AND. J.EQ.JMAX) GO TO 4115
2584      IF(MA.GT.100 .OR. J.EQ.JMAX) GO TO 410
2585      NM=MA
2586      GO TO 415
2587      410 IF(MR.GT.100 .OR. I.EQ.IMAX) GO TO 411
2588      NM=MR
2589      GO TO 415
2590      411 IF(I.EQ.1 .AND. J.EQ.1) GO TO 415
2591      4115 IF(MB.GT.100 .OR. J.EQ.1) GO TO 412

```

```

2592      NM=MB
2593      GO TO 415
2594      412 IF(ML.GT.100 .OR. I.EQ.1) GO TO 413
2595      NM=ML
2596      GO TO 415
2597      413 IF(J.EQ.1) GO TO 414
2598      MN=MB-100
2599      GO TO 4143
2600      414 IF(I.EQ.1) GO TO 4140
2601      MN=ML-100
2602      GO TO 4143
2603      4140 WRITE(6,960) I,J
2604      CALL EXIT
2605      C
2606      4143 DO 4146 N=1,NVOID
2607      IF(J.EQ.1) GO TO 4144
2608      IF(FRACTP(N,MN).GT.0.) GO TO 4148
2609      GO TO 4146
2610      4144 IF(FRACT(N,MN).GT.0.) GO TO 4148
2611      4146 CONTINUE
2612      WRITE(6,960) I,J
2613      CALL EXIT
2614      C
2615      4148 NM=N
2616      415 IF(NM.EQ.0) NM=NVOID
2617      M=IABS(MFLAG(K))-100
2618      MB=MB-100
2619      ML=ML-100
2620      C
2621      43 DO 46 N=1,NVOID
2622      IF(N.NE.NM) GO TO 44
2623      FRACTP(N,M) = TAU(I)
2624      FRACRT(N,M) = DY(J)*X(I)+TWOPI
2625      IF(I.GH.EQ.1)FRACRT(N,M)=DY(J)
2626      IF(MB.LT.0 .OR. J.EQ.1) GO TO 431
2627      FRACTP(N,MB)=TAU(I)
2628      431 IF(ML.LT.0 .OR. I.EQ.1) GO TO 46
2629      FRACRT(N,ML)=DY(J)*X(I-1)+TWOPI
2630      IF(I.GH.EQ.1)FRACRT(N,ML)=DY(J)
2631      GO TO 46
2632      44 RHO(N,M)=0.
2633      46 CONTINUE
2634      C      *** END OF LOOP ON K FOR CELLS WITH NEGATIVE FLAG
2635      48 CONTINUE
2636      480 CONTINUE
2637      IF(I.J.GT.1) GO TO 49
2638      C
2639      C      *** IF THIS IS FIRST SUBCYCLE OF INFACE, COMPUTE
2640      C      FLUXES OF MATERIAL TO BE EVACUATED USING FLUX
2641      C      TERMS FROM LAST CYCLE. INITIALIZE FLUX TERMS
2642      C      OF OTHER MATERIALS.
2643      C
2644      DO 4860 I=1,I1
2645      DO 488 J=1,I2

```

```

2646      K=(J-1)*IMAX+1
2647      M=IABS(MFLAG(K))-100
2648      IF(M.LT.0) GO TO 488
2649      MB=0
2650      IF(J.GT.1) MB=MFLAG(K-IMAX)-100
2651      ML=MFLAG(K-1)-100
2652      C
2653      DO 486 N=1,NMAT
2654      IF(RHO(N,M).GT.0. .OR. XMASS(N,M).LE.0.) GO TO 486
2655      TFLUX=0.
2656      FR=0.
2657      FT=0.
2658      FB=0.
2659      DIFF=XMASS(N,M)
2660      C
2661      MR=IABS(MFLAG(K+1))
2662      IF(SAMMP(N,M).LE.0. .OR. (MR.LT.100 .AND. I.LT.IMAX)) GO TO 482
2663      FR=SAMMP(N,M)
2664      TFLUX=TFLUX+FR
2665      XMASS(N,M)=-DIFF
2666      C
2667      482 MA=IABS(MFLAG(K+IMAX))
2668      IF(SAMPY(N,M).LE.0. .OR. (MA.LT.100 .AND. J.LT.JMAX)) GO TO 483
2669      FT=SAMPY(N,M)
2670      TFLUX=TFLUX+FT
2671      XMASS(N,M)=-DIFF
2672      C
2673      483 IF(I.EQ.1 .OR. ML.LT.0 .OR. RHO(N,ML).LE.0. .OR.
2674      1  SAMMP(N,ML).GE.0.) GO TO 484
2675      FL=-SAMMP(N,ML)
2676      TFLUX=TFLUX+FL
2677      XMASS(N,ML)=-ABS(XMASS(N,ML))
2678      C
2679      484 IF(J.EQ.1 .OR. MB.LE.0 .OR. RHO(N,MB).LE.0. .OR.
2680      1  SAMPY(N,MB).GE.0.) GO TO 485
2681      FB=-SAMPY(N,MB)
2682      TFLUX=TFLUX+FB
2683      XMASS(N,MB)=-ABS(XMASS(N,MB))
2684      C
2685      485 IF(TFLUX.LE.0.) GO TO 486
2686      C
2687      C
2688      C
2689      WS=DIFF/TFLUX
2690      C
2691      SAMMP(N,M)=FR*WS
2692      SAMPY(N,M)=FT*WS
2693      IF(FL.GT.0.) SAMMP(N,ML)=-FL*WS
2694      IF(FB.GT.0.) SAMPY(N,MB)=-FB*WS
2695      486 CONTINUE
2696      488 CONTINUE
2697      4800 CONTINUE
2698      DO 490 M=1,NMXCLS
2699      DO 489 N=1,NMAT

```

```

2700      XM=XMASS(N,M)
2701      IF(XM.LT.0.) GO TO 489
2702      SAMPY(N,M) = C.
2703      SAMMP(N,M) = C.
2704      489 XMASS(N,M) = ABS(XM)
2705      490 CONTINUE
2706      C
2707      C.....
2708      C
2709      C      *** COMPUTE FLUXES FOR CELLS CONTAINING INTERFACES.
2710      C
2711      C      *** CYC IS FACTOR IN FLUX EQUATIONS AND IS .GT. 1 FOR
2712      C      CELLS THAT BECOME MIXED AFTER FIRST SUBCYCLE.
2713      C      (CYC.GT.1 ONLY WHEN FLUX CALLED FROM NEWMIX)
2714      49 CYC = 1.
2715      DO 100 I=1,11
2716      DO 60 J=1,12
2717      K=(J-1)*IMAX+I+1
2718      C      *** IF CELL K IS NOT MIXED, SKIP OUT.
2719      MFK=IABS(MFLAG(K))
2720      IF(MFK.LT.100) GO TO 60
2721      C      *** DEFINE LOCATION IN MIXED ARRAYS OF INFO. ON CELL K
2722      M=MFK-100
2723      C      *** DEFINE INDICES OF CELLS ABOVE AND ON RIGHT
2724      KR=K+1
2725      KA=K+IMAX
2726      MA=IABS(MFLAG(KA))
2727      MR=IABS(MFLAG(KR))
2728      C      *** IF CELL K CONTAINS A FREE SURFACE SET IFS1 = 1
2729      IFS1=0
2730      IF(RHO(NVOID,H).GT.0.) IFS1=1
2731      CALL FLUX
2732      50 CONTINUE
2733      C      *** END OF J-LOOP
2734      60 CONTINUE
2735      C      *** END OF I-LOOP
2736      100 CONTINUE
2737      C
2738      C.....
2739      C
2740      C      *** MOVE TRACERS.
2741      C
2742      NVP=NVOID+1
2743      C      *** WHEN NTCC.GT.1, PASSIVE CELL CENTERED TRACERS (XP,YP)
2744      C      ARE BEING PROCESSED.
2745      C      WHEN NTCC=0, THESE TRACERS HAVE NOT BEEN GENERATED.
2746      DO 730 N=1,NVP
2747      NN=NMP(N)
2748      IF(N.GT.NVOID) NN=NTCC
2749      IF(NN.EQ.0) GO TO 730
2750      DO 720 L=1,NN
2751      C      *** FIND I AND J OF CELL TRACER IS IN BEFORE IT IS MOVED.
2752      IF(I.LT.NVP) GO TO 491
2753      I=INT(XP(L))+1

```



```

2754     J=INT(YP(L))+1
2755     GO TO 492
2756     491 CONTINUE
2757         I=INT(TX(N,L))+1
2758         IF(TX(N,L).GE.FLOAT(IMAX).AND.ABS(TX(N,L)-AINT(TX(N,L))).LE.D.)
2759             I=I-1
2760         J=INT(TY(N,L))+1
2761     C         *** IF TRACER IS OUTSIDE GRID, SKIP OUT.
2762     492 CONTINUE
2763         IF(I.GT.I1 .OR. J.GT.I2 .OR. J.LT.D) GO TO 720
2764     C
2765         K=(J-1)*IMAX+1+1
2766         M=IABS(MFLAG(K))-100
2767     C
2768     C         *** STORE FRACTIONAL PARTS OF THE COORDINATES IN FRX,FRY.
2769     C
2770         IF(N.LT.NVP) GO TO 493
2771         FRX = XP(L)-AINT(XP(L))
2772         FRY = YP(L)-AINT(YP(L))
2773         GO TO 494
2774     493 CONTINUE
2775         FRX = TX(N,L) - AINT(TX(N,L))
2776         FRY = TY(N,L) - AINT(TY(N,L))
2777     494 CONTINUE
2778         WSX=ABS(FRX-.5)
2779         WSY=ABS(FRY-.5)
2780     C         *** DEFINE INDICES OF NEIGHBOR CELLS.
2781         KH=K+1
2782         KV=K+IMAX
2783         IF(FRY.LT.(.5)) KV=K-IMAX
2784         KD=KV+1
2785         IF(FRX.GE.(.5)) GO TO 500
2786         KH=K-1
2787         KD=KV-1
2788     500 CONTINUE
2789     C         *** INDICES OF CELLS OUTSIDE THE GRID
2790         IF(J.GT.I) GO TO 505
2791         KV=MAXO(K,KV)
2792         KD=MAXO(KH,KD)
2793     505 IF(J.LT.JMAX) GO TO 510
2794         KV=MINDO(K,KV)
2795         KD=MINDO(KH,KD)
2796     510 IF(I.GT.I) GO TO 515
2797         KH=MAXO(K,KH)
2798         KD=MAXO(KD,KV)
2799     515 IF(I.LT.IMAX) GO TO 520
2800         KH=MINDO(K,KH)
2801         KD=MINDO(KV,KD)
2802     520 CONTINUE
2803     C         *** DEFINE WEIGHTING FACTORS
2804         WH=0.
2805         WV=0.
2806         WD=0.
2807         WK=0.

```

```

2808      WFS=0.
2809      IF(AMX(KH).GT.0.) WH=WSX*(1.0-WSY)
2810      IF(AMX(KV).GT.0.) WV=WSY*(1.0-WSX)
2811      IF(AMX(KD).GT.0.) WD=WSX*WSY
2812      IF(AMX(K).GT.0.) WK=(1.0-WSY)*(1.0-WSX)
2813      C      *** SUM WEIGHTING FACTORS
2814      WFS = WH + WV + WD + WK
2815      IF(WFS.LE.0.) GO TO 720
2816      C      *** CALCULATE RADIAL VELOCITY OF THE TRACER.
2817      IF(I.GT.1 .OR. FRX.GE.(.5)) GO TO 603
2818      WH=-WH
2819      WD=-WD
2820      603 CONTINUE
2821      UEFF=(U(KH)*WH + U(KV)*WV + U(KD)*WD + U(K)*WK)/WFS
2822      WH=ABS(WH)
2823      WD=ABS(WD)
2824      C      *** CALCULATE AXIAL VELOCITY OF THE TRACER.
2825      IF(J.GT.1 .OR. CVIS.LT.0. .OR. FRY.GE.(.5)) GO TO 604
2826      WV=-WV
2827      WD=-WD
2828      604 CONTINUE
2829      VEFF=(V(KH)*WH + V(KV)*WV + V(KD)*WD + V(K)*WK)/WFS
2830      605 CONTINUE
2831      C      *** STORE NEW TRACER COORDINATES
2832      IF(ABS(UEFF).LT.UMIN) UEFF=0.
2833      IF(ABS(VEFF).LT.UMIN) VEFF=0.
2834      DISTX = UEFF*SDT
2835      DISTY = VEFF*SDT
2836      POSX = X(I-1) + DX(I)*FRX + DISTX
2837      POSY = Y(J-1) + DY(J)*FRY + DISTY
2838      C      *** DO NOT ALLOW TRACERS TO MOVE OFF OF AXIS.
2839      IF(TX(N,L).LE.0. .AND. N.LT.NVP) GO TO 709
2840      IF(POSX.GT.X(I))GO TO 705
2841      IF(POSX.LT.X(I-1).AND. I.GT.1) GO TO 707
2842      IF(N.EQ.NVP) GO TO 704
2843      TX(N,L) = TX(N,L) + DISTX/DX(I)
2844      GO TO 709
2845      704 XP(L)=XP(L)+DISTX/DX(I)
2846      GO TO 709
2847      705 IF(N.EQ.NVP) GO TO 706
2848      TX(N,L) = FLOAT(I) +(POSX-X(I))/DX(I+1)
2849      GO TO 709
2850      706 XP(L)=FLOAT(I)+(POSX-X(I))/DX(I+1)
2851      GO TO 709
2852      707 IF(N.EQ.NVP) GO TO 708
2853      TX(N,L) = FLOAT(I-2) + 1.0 - (X(I-1)-POSX)/DX(I-1)
2854      GO TO 709
2855      708 XP(L)=FLOAT(I-2)+1.0-(X(I-1)-POSX)/DX(I-1)
2856      C
2857      C      *** DO NOT ALLOW TRACERS TO MOVE OFF OF AXIS.
2858      709 IF(TY(N,L).LE.0. .AND. N.LT.NVP) GO TO 606
2859      IF(POSY.GT.Y(J)) GO TO 711
2860      IF(POSY.LT.Y(J-1) .AND. J.GT.1) GO TO 713
2861      IF(N.EQ.NVP) GO TO 710

```

```

2862      TY(N,L) = TY(N,L) + DISTY/DY(J)
2863      GO TO 715
2864      710 YP(L)=YP(L)+DISTY/DY(J)
2865      GO TO 715
2866      711 IF(N.EQ.NVPI) GO TO 712
2867      TY(N,L) = FLOAT(J) + (POSY-Y(J))/DY(J+1)
2868      GO TO 715
2869      712 YP(L)=FLOAT(J)+(POSY-Y(J))/DY(J+1)
2870      GO TO 715
2871      713 IF(N.EQ.NVPI) GO TO 714
2872      TY(N,L)=FLOAT(J-2) + 1.0 + (Y(J-1)-POSY)/DY(J-1)
2873      GO TO 715
2874      714 YP(L)=FLOAT(J-2)+1.0-(Y(J-1)-POSY)/DY(J-1)
2875      715 CONTINUE
2876      IF(N.LT.NVPI) GO TO 606
2877      IF(XP(L).LT.0.) XP(L)=0.
2878      IF(YP(L).GT.0. .AND. XP(L).LE.FLOAT(IMAX) .AND. YP(L).LE.FLOAT(JMAX))
2879      I GO TO 607
2880      XP(L)=0.
2881      YP(L)=0.
2882      GO TO 607
2883      606 CONTINUE
2884      C      *** IF TRACER CROSSED AXIS, PUT IT BACK ON AXIS.
2885      IF(TX(N,L).LT.0.) TX(N,L) = 0.
2886      IF(TY(N,L).LT.0.) TY(N,L)=0.
2887      607 CONTINUE
2888      C      *** END OF LOOP ON L
2889      720 CONTINUE
2890      C      *** END OF LOOP ON MATERIALS (MN)
2891      730 CONTINUE
2892      C      *** END OF SUBCYCLE LOOP
2893      875 CONTINUE
2894      C      *** IF INFACE IS NOT BEING SUBCYCLED (ICY=1),
2895      C      THE FLUX TERMS OF CELLS BEING EVACUATED OF A
2896      C      MATERIAL HAVE ALREADY BEEN ADJUSTED (SEE BELOW
2897      C      STATEMENT 480).
2898      IF(ICY.EQ.1) GO TO 9600
2899      C
2900      C..... ADJUST FLUX TERMS OF CELLS EVACUATED BY A MATERIAL
2901      C      INTERFACE.
2902      C
2903      DO 9500 I=1,11
2904      DO 950 J=1,12
2905      K=(J-1)*IMAX+1+1
2906      M=IABS(MFLAG(K))
2907      IF(N.LT.100) GO TO 950
2908      M=M-100
2909      C      *** ADJUST FLUX OF EACH MATERIAL
2910      DO 940 N=1,NMAT
2911      C      *** IF XMASS(N,M) .GT. 0. , BUT RHO(N,M) = 0., MATERIAL
2912      C      N IS TO BE EVACUATED.
2913      IF(XMASS(N,M).LE.0.) GO TO 940
2914      IF(RHO(N,M).GT.0.) GO TO 940
2915      MB=0

```

```

2916 IF(J.GT.1) MB=MFLAG(K-IMAX)-100
2917 ML=MFLAG(K-1)-100
2918 TFLUX=0.
2919 FR=0.
2920 FT=0.
2921 FL=0.
2922 FB=0.
2923 C
2924 IF(SAMMP(N,M).LE.0.) GO TO 880
2925 FR=SAMMP(N,M)
2926 TFLUX=TFLUX+FR,
2927 C
2928 880 IF(SAMPY(N,M).LE.0.) GO TO 885
2929 FT=SAMPY(N,M)
2930 TFLUX=TFLUX+FT
2931 C
2932 885 IF(I.EQ.1 .OR. ML.LT.0 .OR. RHO(N,ML).LE.0. .OR.
2933 I SAMMP(N,ML).GE.0.) GO TO 890
2934 FL=-SAMMP(N,ML)
2935 TFLUX=TFLUX+FL
2936 C
2937 890 IF(J.EQ.1 .OR. MB.LE.0 .OR. RHO(N,MB).LE.0. .OR.
2938 I SAMPY(N,MB).GE.0.) GO TO 895
2939 FB=-SAMPY(N,MB)
2940 TFLUX=TFLUX+FB
2941 C
2942 895 IF(TFLUX.GT.0.) GO TO 900
2943 GO TO 940
2944 C
2945 900 SAMMY(N)=0.
2946 SGAMC(N,J)=0.
2947 IF(IABS(MFLAG(K-IMAX)).LT.100 .OR. J.EQ.1) GO TO 905
2948 MB=IABS(MFLAG(K-IMAX))-100
2949 SAMMY(N)=SAMPY(N,MB)
2950 C
2951 905 IF(IABS(MFLAG(K-1)).LT.100 .OR. I.EQ.1) GO TO 910
2952 ML=IABS(MFLAG(K-1))-100
2953 SGAMC(N,J)=SAMMP(N,ML)
2954 C
2955 910 DIFF=XMASS(N,M)-SAMPY(N,M)-SAMMP(N,M)+SAMMY(N)+SGAMC(N,J)
2956 C
2957 WS=DIFF/TFLUX
2958 C
2959 SANMP(N,M)=FR*WS+SAMMP(N,M)
2960 SAMPY(N,M)=FT*WS+SAMPY(N,M)
2961 IF(FL.GT.0.) SAMMP(N,ML)=-FL*WS+SAMMP(N,ML)
2962 IF(FB.GT.0.) SAMPY(N,MB)=-FB*WS+SAMPY(N,MB)
2963 WRITE(6,9991) I, J, N, M, ML, MB, DIFF, XMASS(N,M), SAMMY(N),
2964 I SGAMC(N,J), SAMPY(N,M), SAMMP(N,M), SAMPY(N,MB), SAMMP(N,ML)
2965 9991 FORMAT(25H EVACUATION OF MATERIAL N/14H I,J,N,M,ML,MB,6I6/
2966 1 36H DIFF,XMASS(N,M),SAMMY(N),SGAMC(N,J),1P4E20.8/
2967 2 46H SAMPY(N,M),SAMMP(N,M),SAMPY(N,MB),SAMMP(N,ML),
2968 3 1P4E20.8)
2969 990 FORMAT(63H TROUBLE REFINING MATERIAL OF CELL THAT HAS BECOME PURE,

```

```

2970      I 1,J = ,214)
2971      C          *** END OF LOOP ON MATERIAL.
2972      740 CONTINUE
2973      C          *** END OF LOOP ON K (CELLS).
2974      950 CONTINUE
2975      9500 CONTINUE
2976      9600 RETURN
2977      END
2978      SUBROUTINE INPUT
2979      C          *** READ RESTART TAPE.
2980      C          *** CALL CARDS TO READ INPUT DECK. PRINT INPUT VARIABLES.
2981      C          *** DEFINE CONSTANTS
2982      C          *** CALL SETUP TO DEFINE CELL QUANTITIES AT TIME=0.
2983      INCLUDE COMDIM
2984      C
2985      KUNITR=7
2986      KUNITW=7
2987      C          *** READ AND PRINT ID HEADING CARD (FIRST CARD IN
2988      C          INPUT DECK)
2989      READ (5,370) IWS
2990      WRITE (6,370) IWS
2991      C          *** CARDS ROUTINE WILL READ AND PRINT FIRST DATA CARD.
2992      CALL CARDS
2993      C          *** PK(3).LT.0. MEANS THIS PROBLEM IS BEING RESTARTED FROM
2994      C          THE RESTART TAPE AND SETUP IS NOT NEEDED.
2995      IF (PK(3).LT.0.) GO TO 70.
2996      CALL CARDS
2997      C          *** Z(1)=PROB IS DEFINED BY THE SECOND CARD OF A SETUP
2998      C          DECK, BUT IS NOT DEFINED IN A RESTART DECK.
2999      S      IF (PROB.EQ.0.) GO TO 230
3000      CALL SETUP
3001      GO TO 70
3002      10    CONTINUE
3003      CALL CARDS
3004      C          *** INITIALIZE P-STORAGE.
3005      20    DO 30 K=1,KMAXA
3006      30    PK)=0.0
3007      C          *** SET T AND NC SO THEY WILL EQUAL ZERO ON FIRST EDIT
3008      C          PRINT AFTER BEING INCREMENTED BY CDT.
3009      T=T-DTNA
3010      NC=NC-1
3011      C          *** CHECK FATAL INPUT ERRORS.
3012      36    IF (IMAX.EQ.0.OR.JMAX.EQ.0) GO TO 280
3013      C          *** DEFINE CONSTANTS USED THROUGHOUT CALCULATION.
3014      CYCLE=NC
3015      NUMSP=0
3016      WFLAGF=1.
3017      WFLAGL=0.
3018      NRZ=NRZ-NUMRFZ
3019      TWOPI=2.*PIDY
3020      VT=10.**(-5)
3021      SSZ=1.
3022      C          *** PRINT VALUES OF MOST INPUT PARAMETERS.
3023      WRITE(6,310) ICSTOP, IEXTX, INTER, IMAX, IPCYCL, IPR, IVARDX,

```

```

3024      1      IVARDY, I1, I2, JEXTY, JMAX, JPROJ, MAPS, MAXX,
3025      2      MAXY, MINX, MINY, NADD, NDUMP7, NFRELP, NMXCLS,
3026      3      NODUMP, NSIDES, NTCC, NTPMX, NTRACR, NUMREZ, NUMSCA,
3027      4      N6
3029      WRITE(6, 320)      RBAR, CVIS, CYCHX, CYCPH3, DMIN, DTMIN,
3029      1      DXF, DYF, EMIN, FINAL, GAMMA, GLUED, PRCNT,
3030      2      PRDEL, PRFACT, PRLIN, RADIUS, REZFC,
3031      3      ROCPS, SSZ, SS4, STAB, TSTOP, VT
3032      C      *** PRINT INITIAL CONDITIONS
3033      WRITE(6, 300)
3034      DO 38 L=1, NMAT
3035      MN = MAT(L)
3036      WRITE(6, 390) L, RHOZ(MN), KHOIN(L), SSIEN(L), UUR(L), VVAL(L)
3037      38 CONTINUE
3038      C      *** PRINT DX, DY ARRAYS WHEN THE CELL DIMENSIONS ARE
3039      C      VARIABLE.
3040      IF (IVARDY.EQ.0) GO TO 40
3041      WRITE (6, 330)
3042      WRITE (6, 350) (I, DX(I)), I=1, INAX)
3043      40 IF (IVARDY.EQ.0) GO TO 50
3044      WRITE (6, 340)
3045      WRITE (6, 350) (J, DY(J)), J=1, JMAX)
3046      50 CONTINUE
3047      C      *** WHEN T.GT.C., PROBLEM IS BEING RESTARTED.
3048      IF (T.GT.C.) GO TO 60
3049      C      *** DEFINE TIME OF FIRST EDIT PRINT AFTER CYCLE D.
3050      PRTIME=PRDEL
3051      GO TO 300
3052      C      *** PRDEL = D. WHEN PRINTING ON CYCLES RATHER TIME.
3053      60 IF (PRDEL.EQ.C.) GO TO 300
3054      C      *** DEFINE TIME OF FIRST EDIT PRINT AFTER RESTART CYCLE.
3055      IWS=T/PRDEL+1.
3056      PRTIME=FLOAT(IWS)*PRDEL
3057      GO TO 300
3058      C      *** READ DUMP TAPE
3059      70 CONTINUE
3060      IWS=0
3061      NMAT=INT(PK(4)+.5)
3062      80 REWIND KUNITR
3063      90 READ (KUNITR) PR(1), PR(2)
3064      C      *** FIRST WORD OF FIRST RECORD OF EACH DUMP SHOULD BE
3065      C      555.C. TEST THIS THREE TIMES BEFORE EXITING.
3066      IF (PR(1)-555.0) 100, 110, 100
3067      100 IWS=IWS+1
3068      IF (MOD(IWS, 3)) 220, 220, 80
3069      110 IF (PR(2)) 100, 120, 120
3070      C      *** WHEN SETTING UP A PROBLEM PR(2) = PK(2) = 0. WHEN
3071      C      RESTARTING A PROBLEM, THE RESTART TAPE IS READ UNTIL
3072      C      PR(2).GE.PK(2), THE RESTART CYCLE NUMBER.
3073      120 IF (PK(2)-PR(2)) 150, 150, 130
3074      130 NREC = NMAT+12
3075      DO 140 L=2, NREC
3076      140 READ (KUNITR)
3077      GO TO 90

```

```

3078      150 READ (KUNITR) (Z(I),I=1,150)
3079          NVOID = NMAT + J
3080      C          *** MAKE SURE PROBLEM NUMBER ON TAPE (PROB) MATCHES
3081      C          PROBLEM NUMBER ON INPUT CARDS (PK(I)).
3082          IF(ABS(PROB-PK(I)) - .01) 151,151,210
3083      151 IF(PK(3)+3.) 160,152,160
3084      152 KUNITR=7
3085      160 READ (KUNITR) (U(I),V(I),AMX(I),AIX(I), P(I), MFLAG(I),I=1,KMAV)
3086          READ (KUNITR) (STRSZ(I), STRSRR(I), STRSRZ(I), I=1,KMAX)
3087          READ (KUNITR) (X(I), DX(I), TAU(I), I=1,IMAX)
3088          READ (KUNITR) (Y(I), DY(I), I=1,JMAX)
3089          READ (KUNITR) (CZERO(M), STK1(M), STK2(M), STEZ(M), RMU(M),
3090      1 ANDM(M), RHOIN(M), SSIEN(M), UUR(M), VVA(M), MAT(M), PLW(M),
3091      2 M=1,NMAT)
3092          READ (KUNITR) (MPAC(I),MPACK(I),I=1,MBBB)
3093          READ (KUNITR) ((PACX(I,L),PACY(I,L),I=1,MBBB),L=1,MBB)
3094          READ (KUNITR) ((XMASS(M,L), RHO(M,L), SIE(M,L), SAMPY(M,L),
3095      1 SAMP(M,L),M=1,NMAT),RHO(NVOID,L),L=1,NMXCLS)
3096          DO 165 N=1,NVOID
3097          READ (KUNITR) NP, (TX(N,L),TY(N,L),L=1,NP)
3098          NMP(N)=NP
3099      165 CONTINUE
3100          IF(PK(3).EQ.(-3)) GO TO 173
3101          READ (KUNITR) NP, (XP(L),YP(L),L=1,NP)
3102      173 CONTINUE
3103          READ (KUNITR) PR(1), PR(2)
3104      C          *** PK(3)=-3, FOR A -CLAM- START
3105      C          CELL-CENTERED TRACERS ARE NOT GENERATED BY THE
3106      C          'CLAM' GENERATOR.
3107          IF(PK(3).EQ.(-3.)) GO TO 200
3108      C          *** THE FIRST WORD OF THE LAST RECORD OF EACH DUMP SHOULD
3109      C          BE 555.0 OR 666.0.
3110      175 IF(PR(1)-555.0) 240,10,180
3111      180 IF(PR(2)-666.0) 250,10,250
3112      C          *** INITIALIZE Z ARRAY WHEN IT IS READ IN FROM CLAM TAPE.
3113      200 DO 205 I=1,150
3114      205 Z(I)=0.
3115          CALL CARDS
3116          CALL SETUP
3117          GO TO 20
3118      C          *** PROBLEM NUMBER ON THE RESTART TAPE IS NOT THE SAME AS
3119      C          THE PROBLEM NUMBER ON THE INPUT CARD.
3120      210 NK=150
3121          GO TO 290
3122      C          *** CANNOT FIND FIRST WORD OF FIRST RECORD.
3123      220 NK=100
3124          GO TO 290
3125      C          *** NOT A RESTART AND YET Z(1) = 0.
3126      230 NK=5
3127          GO TO 290
3128      C          *** FIRST WORD OF LAST RECORD IS INCORRECT.
3129      240 NK=175
3130          GO TO 290
3131      C          *** SECOND WORD OF LAST RECORD IS INCORRECT.

```

```

3132 250 NK=180
3133 GO TO 290
3134 C *** IMAX OR JMAX IS ZERO
3135 280 NK=36
3136 290 NR=1
3137 C *** PRINT FIRST TWO WORDS OF DUMP (PR(1),PR(2))
3138 AND Z(151),Z(152),Z(153).
3139 WRITE(6,360) PR(1), Z(151), PR(2), Z(152), Z(153)
3140 CALL ERROR
3141 300 RETURN
3142 C
3143 310 FORMAT(//7X,6HICSTOP,6X,6H IEXTX,6X,6H INTER,6X,6H IMAX,6X,
3144 1 6HIPCYCL,6X,6H IPR,6X,6HIVARDX,6X,6HIVARDY,6X,
3145 2 6H I1,6X,6H I2//1X,10I12//7X,
3146 3 6H JEXTY,6X,6H JMAX,6X,6H JPROJ,6X,6H MAPS,6X,
3147 4 6H MAXX,6X,6H MAXY,6X,6H MINX,6X,6H MINY,6X,
3148 5 6H NADD,6X,6HNDUMP7//1X,10I12//7X,
3149 6 6HNFRELP,6X,6HNMXCLS,6X,6HNODUMP,6X,6HNSIDES,6X,
3150 7 6H NTCC,6X,6H NTPMX,6X,6HNTRACR,6X,6HNUMREZ,6X,
3151 8 6HNUMSCA,6X,6H N6//1X,10I12//)
3152 320 FORMAT(//7X,6H ,6X,6H BBAR,6X,6H CVIS,6X,6H CYCHX,6X,
3153 1 6HCYCPH3,6X,6H DMIN,6X,6H DTMIN,6X,6H DXF,6X,
3154 2 6H DYF,6X,6H EMIN//1X,1P9E12.4// 7X,
3155 3 6H ,6X,6H FINAL,6X,6H GAMMA,6X,6H GLUED,6X,
3156 4 6H PRCNT,6X,6HPRDELTA,6X,6HPRFACT,6X,6H PRLIM,6X,
3157 5 6HRADIUS,6X,6HREZFACT//1X,1P9E12.4// 7X,
3158 6 6H ,6X,6H ROEPS,6X,6H SS2,6X,6H SS4,6X,
3159 7 6H STAB,6X,6H TSTOP,6X,6H VT//1X,1P6E12.4//)
3160 330 FORMAT (//7(3H I,6X,2HDR,7X))
3161 340 FORMAT (//7(3H J,6X,2HDZ,7X))
3162 350 FORMAT (7(14,2X,1PE9.3,3X))
3163 360 FORMAT (11H1,5X,72H*** CHECK FIRST RECORD OF THE DUMP AND FIRST DAT
3164 1A CARD OF THE INPUT DECK // 4X,7HON TAPE,41X,8HON CARDS / 4X,
3165 24HWS =,F6.1,4X,7H(555.0),24X, 8HZ(151) =,F8.4,3X,16H(PROBLEM NUMBE
3166 3R) / 8H CYCLE =,F6.1,4X,18H(CYCLE BEING READ),13X, 8HZ(152) =F5.1,
3167 46X,15H(RESTART CYCLE) / 37X,
3168 512X, 8HZ(153) =,F5.1,6X,14H(RESTART FLAG))
3169 370 FORMAT (11,71H
3170 1 )
3171 380 FORMAT(64H PACKAGE NORMAL I N I T I A L C O N D I T
3172 1 I O N S/76H NUMBER DENSITY DENSITY S.I.E.
3173 2 U V//13X, 6H(RHOZ),6X, 7H(RHOIN)//)
3174 390 FORMAT(15,F13.3,F13.3,9X,1PE10.4,5X,1PE10.4,5X,1PE10.4)
3175 END
3176 SUBROUTINE LOCJIXYLOC,LOC,FAC,IDR)
3177 C *** GIVEN CH. COORDINATES OF A POINT, LOCJIX TELLS IN WHICH
3178 C ROW OR COLUMN OF THE GRID THE POINT IS LOCATED.
3179 INCLUDE COMDIM
3180 C
3181 C***** FAC=0. WHEN FINDING TRACER COEFFICIENTS.
3182 C***** FAC=.5 WHEN CALLED FROM SETUP.
3183 C***** FAC=1. WHEN FINDING CELL POINT IS IN.
3184 C***** IDR=0 WHEN FINDING AN X COORDINATE.
3185 C***** IDR=1 WHEN FINDING A Y COORDINATE.

```



```

3186 C
3187 IF(IIDR.EQ.0)GO TO 40
3188 C
3189 C***** FIND A Y COORDINATE
3190 C
3191 DO 10 LCO=1,JMAX
3192 LOC=LCO
3193 YTEMP=Y(LOC-1)+(1.-FAC)*DY(LOC)
3194 IF(XYLOC.LT.YTEMP)GO TO 30
3195 10 CONTINUE
3196 C
3197 C***** POINT IS OUTSIDE GRID. FIND CELL POINT WOULD
3198 C***** FALL IN IF GRID WOULD BE EXPANDED UPWARDS.
3199 C
3200 DYTEMP=.5*(DY(JMAX)+DY(JMAX-1))
3201 YTEMP=Y(JMAX)-FAC*DYTEMP
3202 20 LOC=LOC+1
3203 YTEMP=YTEMP+DYTEMP
3204 IF(XYLOC.GE.YTEMP)GO TO 20
3205 30 LOC=LOC-1
3206 RETURN
3207 C
3208 C***** FIND AN X COORDINATE
3209 C
3210 40 DO 50 LCO=1,IMAX
3211 LOC=LCO
3212 XTEMP=X(LOC-1)+(1.-FAC)*DX(LOC)
3213 IF(XYLOC.LT.XTEMP)GO TO 70
3214 50 CONTINUE
3215 C
3216 C***** POINT IS OUTSIDE GRID. FIND CELL POINT WOULD
3217 C***** FALL IN IF GRID WOULD BE EXPANDED TO RIGHT.
3218 C
3219 DXTEMP=.5*(DX(IMAX)+DX(IMAX-1))
3220 XTEMP=X(IMAX)-FAC*DXTEMP
3221 60 LOC=LOC+1
3222 XTEMP=XTEMP+DXTEMP
3223 IF(XYLOC.GE.XTEMP)GO TO 60
3224 70 LOC=LOC-1
3225 RETURN
3226 END
3227 SUBROUTINE MAP
3228 C
3229 C
3230 C
3231 INCLUDE COMDIN
3232 C
3233 C
3234 DIMENSION WSMAX(5)
3235 DIMENSION ALE(41)
3236 DATA ALE/ 2H .,2H =,2H A,2H B,2H C,2H D,2H E,2H F,
3237 1 2H G,2H H,2H I,2H J,2H K,2H L,2H M,2H N,2H O,
3238 2 2H P,2H Q,2H R,2H S,2H T,2H U,2H V,2H W,2H X,
3239 3 2H Y,2H Z,2H *,2H *,2H I,2H ?,2H 3,2H 4,2H 5,

```

```

3240      4      2H 6,2H 7,2H 8,2H 9,2H 0,2H /
3241      DIMENSION XUM(4)
3242      DATA XUM/      2H .,2H -,2H-A,2H-B,2H-C,2H-D,2H-E,2H-F,
3243      1      2H-G,2H-H,2H-I,2H-J,2H-K,2H-L,2H-M,2H-N,2H-O,
3244      2      2H-P,2H-Q,2H-R,2H-S,2H-T,2H-U,2H-V,2H-W,2H-X,
3245      3      2H-Y,2H-Z,2H+,2H-,2H-1,2H-2,2H-3,2H-4,2H-5,
3246      4      2H-6,2H-7,2H-8,2H-9,2H-0,2H /
3247      MSYMBL=26
3248      IDL=MIND(11,54)
3249      JDL=12
3250      IF (NC.NE.0) GO TO 1
3251      IDL=MIND(IMAX,54)
3252      JDL=JMAX
3253      C
3254      C      *** FIND MAXIMUM VALUE IN ACTIVE GRID OF EACH PROPERTY
3255      C
3256      C      *** COMPRESSION
3257      I      WSMIN=10E20
3258      WSMAX(1)=0.
3259      DO 2 J=1,JDL
3260      DO 2 I=1,IDL
3261      K=(J-1)*IMAX+I+1
3262      IF (ABS(AMX(K)).LE.0.) GO TO 2
3263      IF (MFLAG(K).GT.100) GO TO 2
3264      M=MFLAG(K)
3265      N=MAT(M)
3266      WS=RHOZ(N)
3267      IF (N.EQ.20) WS=RHOIN(M)
3268      COMP= AMX(K)/(DY(J)*TAU(I)*WS)
3269      WSMAX(1) = AMAX1(WSMAX(1),COMP)
3270      WSMIN = AMIN1(WSMIN,COMP)
3271      2      CONTINUE
3272      IF (WSMAX(1).GT.WSMIN) GO TO 3
3273      WSMIN = 0.
3274      C      *** PRESSURE
3275      3      WSMAX(2)=0.
3276      DO 4 J=1,JDL
3277      DO 4 I=1,IDL
3278      K=(J-1)*IMAX+I+1
3279      4      WSMAX(2) = AMAX1(WSMAX(2),ABS(P(K)))
3280      C      *** RADIAL VELOCITY
3281      WSMAX(3)=0.
3282      DO 6 J=1,JDL
3283      DO 6 I=1,IDL
3284      K=(J-1)*IMAX+I+1
3285      6      WSMAX(3) = AMAX1(WSMAX(3),ABS(U(K)))
3286      C      *** AXIAL VELOCITY
3287      WSMAX(4)=0.
3288      DO 8 J=1,JDL
3289      DO 8 I=1,IDL
3290      K=(J-1)*IMAX+I+1
3291      8      WSMAX(4) = AMAX1(WSMAX(4),ABS(V(K)))
3292      WSMAX(5)=0.
3293      DO 10 J=1,JDL

```

```

3294 C          *** SPECIFIC INTERNAL ENERGY
3295 DO 10 I=1,IDL
3296 K=(J-1)*IMAX+I+1
3297 10 WSMAX(5) = AMAX1(WSMAX(5),ABS(AIX(K)))
3298 C
3299 C          *** STORE INFORMATION TO BE PLOTTED IN PROP ARRAY
3300 C          A ROW AT A TIME.
3301 C
3302 NPROP = 1
3303 C          *** COMPRESSION
3304 J=JDL
3305 MS=MSYMBL+1
3306 WRITE(6,500)
3307 15 DO 20 I=1,IDL
3308 PROP(I)=0.
3309 K=(J-1)*IMAX+I+1
3310 IF(ABS(AMX(K)).LE.0.)GO TO 20
3311 IF(MFLAG(K).GT.100) GO TO 20
3312 M=MFLAG(K)
3313 N=NAT(M)
3314 WS=RHOZ(N)
3315 IF(N.EQ.20) WS=RHOIN(M)
3316 PROP(I)=AMX(K)/(TAU(I))*DY(J)*WS)
3317 20 CONTINUE
3318 GO TO 110
3319 C
3320 C          *** PRESSURE
3321 C
3322 30 J=JDL
3323 NS=NSYMBL
3324 IF(WSMAX(NPROP) .LE. 0.) GO TO 396
3325 WRITE(6,510)
3326 35 DO 40 I=1,IDL
3327 K=(J-1)*IMAX+I+1
3328 40 PROP(I) = P(K)
3329 GO TO 110
3330 C
3331 C          *** RADIAL VELOCITY
3332 C
3333 50 J=JDL
3334 IF(WSMAX(NPROP) .LE. 0.) GO TO 396
3335 WRITE(6,520)
3336 55 DO 60 I=1,IDL
3337 K=(J-1)*IMAX+I+1
3338 60 PROP(I) = U(K)
3339 GO TO 110
3340 C
3341 C          *** AXIAL VELOCITY
3342 C
3343 70 J=JDL
3344 IF(WSMAX(NPROP) .LE. 0.) GO TO 396
3345 WRITE(6,530)
3346 75 DO 80 I=1,IDL
3347 K=(J-1)*IMAX+I+1

```

```

3348      80  PROP(I) = V(K)
3349      GO TO 110
3350      C
3351      C          *** SPECIFIC INTERNAL ENERGY
3352      C
3353      90  J=JDL
3354      IF(WSMAX(NPROP) .LE. 0.) GO TO 396
3355      92  WRITE(6,540)
3356      95  DO 100 I=1,IDL
3357      K=(J-1)*IMAX+I+1
3358      100 PROP(I) = AIX(K)
3359      C
3360      C          *** WHEN PRINTING FIRST (TOP) ROW OF MAP, COMPUTE
3361      C          SCALE FACTOR AND PRINT KEY.
3362      110 IF(J.LT.JDL) GO TO 300
3363      C
3364      C          *** COMPUTE SCALE FACTOR AND PRINT MAXIMUM VALUE OF
3365      C          EACH SYMBOL USED
3366      C
3367      C
3368      180 SCALE = WSMAX(NPROP)/FLOAT(MS)
3369      IF(NPROP.EQ.1) SCALE=(WSMAX(1)-WSMIN)/FLOAT(MS)
3370      IF ((AINT(SCALE*1000.)),LT.(SCALE*1000.)) GO TO 190
3371      GO TO 200
3372      190 SCALE = AINT(SCALE*1000.+1)/1000.
3373      200 CONTINUE
3374      C
3375      IF(NPROP.EQ.1) GO TO 220
3376      VALUE(1) = 0.
3377      VALUE(2) = SCALE/10.
3378      VALUE2=VALUE(2)
3379      DO 210 I=1,MS
3380      210 VALUE(I+2) = FLOAT(I)*SCALE
3381      GO TO 240
3382      C          *** VALUES FOR COMPRESSION MAP
3383      220 VALUE(1) = WSMIN
3384      DO 230 I=1,MS
3385      230 VALUE(I+1) = FLOAT(I)*SCALE + WSMIN
3386      C          *** PRINT DEFINITIONS OF MAP SYMBOLS
3387      240 ILIM1 = 1
3388      ILIM2 = 10
3389      MSP=MSYHBL + 2
3390      250 IF (MSP.LT.ILIM2) ILIM2 = MSP
3391      IF (NPROP.NE.1) GO TO 260
3392      WRITE(6,550) (ALE(I),I=ILIM1,ILIM2)
3393      WRITE(6,560) (VALUE(I),I=ILIM1,ILIM2)
3394      GO TO 270
3395      260 WRITE(6,570) (ALE(I),I=ILIM1,ILIM2)
3396      WRITE(6,580) (VALUE(I),I=ILIM1,ILIM2)
3397      270 IF (MSP.EQ.ILIM2) GO TO 280
3398      ILIM1=ILIM2+1
3399      ILIM2=ILIM2+10
3400      GO TO 250
3401      280 WRITE(6,590)

```

```

3402 C
3403 C      *** ASSIGN APPROPRIATE SYMBOL TO EACH CELL IN ROW J.
3404 C
3405 300 DO 370 I=1,IDL
3406     K=(J-1)*IMAX+I+1
3407     IF (ANX(K).GT.0.) GO TO 310
3408     MA = 41
3409     GO TO 360
3410 310 IF(NPROP.EQ.1) GO TO 340
3411     IF(ABS(PROP(I)).GT.C.) GO TO 320
3412     MA = 1
3413     GO TO 360
3414     320 IF(ABS(PROP(I)).GT.VALUE2) GO TO 330
3415     MA = 2
3416     GO TO 360
3417     330 FLOTMA = ABS(PROP(I))/SCALE + 2.
3418     MA = INT(FLOTMA)
3419     IF(FLOTMA.GT.AINT(FLOTMA)) MA=MA+1
3420     MA = MAX0(MA,3)
3421     GO TO 360
3422 C      *** DEFINE MA FOR COMPRESSION MAP
3423 340 IF(+FLAG(K).LT.100) GO TO 345
3424     MA=30
3425     GO TO 360
3426     345 IF(PROP(I).GT.WSMIN) GO TO 350
3427     MA=1
3428     GO TO 360
3429     350 FLOTMA=ABS(PROP(I)-WSMIN)/SCALE+1.
3430     MA = INT(FLOTMA)
3431     IF(FLOTMA.GT.AINT(FLOTMA)) MA = MA+1
3432     MA = MAX0(MA,2)
3433 C      *** STORE CHARACTER TO BE PLOTTED FOR CELL K
3434 360 PR(I) = ALE(MA)
3435     IF(PROP(I).LT.0.) PR(I) = XUM(MA)
3436 C      *** END OF I-LOOP
3437 370 CONTINUE
3438 C      *** PRINT J ROW OF MAP
3439     IF(MOD(J,5).NE.0) GO TO 380
3440     WRITE(6,600) J, (PR(I),I=1,IDL)
3441     GO TO 390
3442 380 WRITE(6,610) (PR(I), I=1,IDL)
3443 390 J=J-1
3444 C      *** HAVE WE REACED BOTTOM ROW
3445     IF(J.EQ.0) GO TO 395
3446     GO TO (15,35,55,75,95),NPROP
3447 C      *** PRINT AND LABEL X-AXIS OF MAP
3448 395 PR(1) = ALE(29)
3449     WRITE(6,600) J, (PR(I),I=1,IDL)
3450     WRITE(6,620) (I, I=0,IDL,5)
3451 C
3452 396 NPROP = NPROP+1
3453     GO TO (400,30,50,70,90,400),NPROP
3454 C
3455 400 RETURN

```

```

3456 C          *** FORMATS
3457 500  FORMAT(1H1,4X,15HCOMPRESSION  //)
3458 510  FORMAT(1H1,4X,15HPRESSURE   //)
3459 520  FORMAT(1H1,4X,15HRADIAL VELOCITY//)
3460 530  FORMAT(1H1,4X,15HAXIAL VELOCITY //)
3461 540  FORMAT(1H1,4X,24HSPECIFIC INTERNAL ENERGY//)
3462 550  FORMAT(16H      SYMBOL      ,10(4X,A2,4X))
3463 560  FORMAT(16H  MAXIMUM VALUE ,10(F6.3,4X))
3464 570  FORMAT(16H      SYMBOL      ,10(3X,A2,5X))
3465 580  FORMAT(16H  MAXIMUM VALUE ,1P10E10.2)
3466 590  FORMAT(//)
3467 600  FORMAT(110,2H 1,54A2)
3468 610  FORMAT(10X,2H 1,54A2)
3469 620  FORMAT(112,10110///// )
3470      END
3471      END
3472      SUBROUTINE NEWMIX
3473 C          *** SETS UP STORAGE IN MIXED CELL ARRAYS FOR A CELL THAT
3474 C          HAS JUST BECOME MIXED-(WHOSE BOUNDARY HAS JUST BEEN CUT
3475 C          BY AN INTERFACE). CALLED FROM INFACE.
3476      INCLUDE COMDIM
3477 C          ***          CFL K HAS BECOME MIXED, SEARCH FOR
3478 C          AVAILABLE STORAGE LOCATION IN MIXED ARRAYS
3479      DO 620 M=1,NMXCLS
3480          IF(RHO(1,M).LT.0.) GO TO 630
3481      620 CONTINUE
3482 C          *** IF YOU FALL THROUGH, THERE IS NO AVAILABLE
3483 C          STORAGE. PRINT MESSAGE, CALL EXIT.
3484          WRITE(6,1000) I, J, K
3485      1000 FORMAT(46HIRAN OUT OF STORAGE FOR MIXED CELLS. (I,J,K)=,4I6)
3486          NK=10
3487          NK=620
3488          CALL ERROR
3489 C          *** REDEFINE MFLAG. RHO(1,M) .GE. 0. WILL INDICATE
3490 C          M STORAGE IS BEING USED.
3491      630 CONTINUE
3492          IF(RHO(1,M).GE.-1.) GO TO 635
3493          NR=10
3494          NK=630
3495          CALL ERROR
3496      635 MFLAG(K)=M+100
3497          DO 640 NN=1,NMAT
3498              RHO(NN,M)=0.
3499              XMASS(NN,M)=C.
3500              SIE(NN,M)=0.
3501      640 CONTINUE
3502          IF(MO.EQ.0) GO TO 700
3503          RHO(MO,M) = ANX(K)/(TAU(1)*DY(J))
3504          XMASS(MO,M)=ANX(K)
3505          SIE(MO,M)=AIX(K)
3506 C          *** MAKE UP SURCYCLES IF NECESSARY
3507          IF(CYC.LT.1.) GO TO 700
3508 C          *** DEFINE      FRAC1P, FRAC1 SO FLUX VARIABLES
3509 C          CAN BE DEFINED FOR SURCYCLES ALREADY COMPLETED.

```

```

3510      FRACFP(MO,M) = TAU(I)
3511      FRACRT(MO,M) = DY(J)*X(I)*TWOPI
3512      IF(IGM.EQ.1)FPACRT(MO,M)=DY(J)
3513      C      *** STORE FLAGS OF CELL ABOVE AND CELL ON RIGHT.
3514      KA=K+IMAX
3515      KR=K+1
3516      MA=IABS(MFLAG(KA))
3517      MR=IABS(MFLAG(KR))
3518      CALL FLUX
3519      FRACFP(MO,M)=C.
3520      FRACRT(MO,M)=0.
3521      700 M=M+100
3522      RETURN
3523      END
3524      SUBROUTINE NEVRHO
3525      C      *** DEFINES THE DENSITY OF A MATERIAL WHOSE INTERFACE
3526      C      HAS JUST ENTERED CELL K. (CALLED FROM INFACE.)
3527      INCLUDE COMDIM
3528      EQUIVALENCE (WSY,RTY), (WSA,TPX), (WSC,FRACX)
3529      C
3530      C
3531      C      *** IS POINT ON RIGHT OR TOP BOUNDARY OF CELL K.
3532      C
3533      IF(FRACX.GT.0.) GO TO 100
3534      C
3535      C..... RIGHT BOUNDARY. CONSIDER CELL ON RIGHT FIRST.
3536      C
3537      KT=K+1
3538      IT=I+1
3539      JT=J
3540      IF(IT.GT.IMAX) GO TO 20
3541      5 MT=IABS(MFLAG(KT))
3542      IF(MT.EQ.0) GO TO 20
3543      C      *** IS CELL KT MIXED OR PURE.
3544      IF(MT.LT.100) GO TO 10
3545      C      *** MIXED. DOES IT CONTAIN MATERIAL N.
3546      IF(RHO(N,MT-100).LE.0.) GO TO 20
3547      C      *** YES. USE ITS DENSITY.
3548      MT=MT-100
3549      GO TO 220
3550      C      *** PURE. DOES IT CONTAIN MATERIAL N.
3551      10 IF(MT.EQ.N) GO TO 230
3552      C
3553      C      *** CELL KT DOES NOT CONTAIN MATERIAL N.
3554      C      CONSIDER ANOTHER NEIGHBOR CELL.
3555      20 IF(KT.EQ.K+1) GO TO 30
3556      IF(KT.EQ.K-IMAX .OR. KT.EQ.K+IMAX) GO TO 50
3557      C      *** NONE OF THE NEIGHBOR CELLS CONTAIN MATERIAL N
3558      C      -- CALL ERROR THEN EXIT
3559      NK=20
3560      NR=11
3561      CALL ERROR
3562      C      *** CELL ON RIGHT DOES NOT CONTAIN MATERIAL N.
3563      C      CONSIDER CELL BELOW OR ABOVE DEPENDING ON WHICH

```

```

3564 C THE INTERCEPT IS CLOSER TO.
3565 30 IF(RTY-AINT(RTY) .LT. (.5)) GO TO 40
3566 C *** USE CELL ABOVE
3567 KT=K+IMAX
3568 IT=I
3569 JT=J+1
3570 IF(JT.GT.JMAX) GO TO 20
3571 GO TO 5
3572 C *** USE CELL BELOW
3573 40 KT=K-IMAX
3574 IT=I
3575 JT=J-1
3576 IF(JT.LT.1) GO TO 20
3577 GO TO 5
3578 C *** CELL ABOVE OR BELOW DOES NOT CONTAIN MATERIAL N,
3579 C CONSIDER DIAGONAL CELL.
3580 50 KT=KT+1
3581 IT=I+1
3582 IF(IT.GT.IMAX) GO TO 20
3583 GO TO 5
3584 C
3585 C..... TOP BOUNDARY. CONSIDER CELL ABOVE FIRST.
3586 C
3587 100 KT=K+IMAX
3588 IT=I
3589 JT=J+1
3590 IF(JT.GT.JMAX) GO TO 120
3591 105 MT=IABS(MFLAG(KT))
3592 IF(MT.EQ.0) GO TO 120
3593 C *** IS CELL KT MIXED OR PURE.
3594 IF(MT.LT.100) GO TO 110
3595 C *** MIXED. DOES IT CONTAIN MATERIAL N.
3596 IF(RHO(N,MT-100).LE.0.) GO TO 120
3597 C *** YES. USE ITS DENSITY.
3598 MT=MT-100
3599 GO TO 220
3600 C *** PURE. DOES IT CONTAIN MATERIAL N.
3601 110 IF(MT.EQ.N) GO TO 230
3602 C *** CELL KT DOES NOT CONTAIN MATERIAL N. CONSIDER
3603 C ANOTHER NEIGHBOR CELL
3604 120 IF(KT.EQ.K+IMAX) GO TO 130
3605 IF(KT.EQ.K-1.OR.KT.EQ.K-1) GO TO 150
3606 C *** NO NEIGHBOR CELL CAN GIVE DENSITY VALUE. THERE MUST
3607 C BE AN ERROR.
3608 NK=120
3609 NR=11
3610 CALL ERROR
3611 C *** CELL ABOVE DOES NOT CONTAIN MATERIAL N. CONSIDER
3612 C CELL ON RIGHT OR LEFT DEPENDING ON WHICH THE INTERCEPT
3613 C IS CLOSER TO.
3614 130 IF(TPX-AINT(TPX) .LT. (.5)) GO TO 140
3615 C *** USE CELL ON RIGHT
3616 KT=K+1
3617 IT=I+1

```



```

3618      JT=J
3619      IF(IT.GT.IMAX) GO TO 120
3620      GO TO 105
3621      C          *** USE CELL ON LEFT
3622      140 KT=K-1
3623          IT=I-1
3624          JT=J
3625          IF(IT.LT.1) GO TO 120
3626          GO TO 105
3627      C          *** CELL ON RIGHT OR LEFT DOES NOT CONTAIN MATERIAL N.
3628      C          CONSIDER DIAGONAL CELL.
3629      150 KT=K+IMAX
3630          JT=J+1
3631          IF(JT.GT.JMAX) GO TO 120
3632          GO TO 105
3633      C
3634      C          *** CELL KT IS MIXED AND CONTAINS MATERIAL N.
3635      C
3636      220 RHO(N,M) = RHO(N,MT)
3637      GO TO 300
3638      C
3639      C          *** CELL KT IS PURE AND CONTAINS MATERIAL N
3640      C
3641      230 RHO(N,M) = AMX(KT)/(TAU(IT)*DY(JT))
3642      C
3643      300 RETURN
3644      END
3645      SUBROUTINE PHI
3646      C          *** COMPUTES EFFECTS OF PRESSURE TERMS TO UPDATE CELL
3647      C          VELOCITIES AND INTERNAL ENERGY.
3648      INCLUDE COMDIM
3649      C
3650      C
3651      C
3652      C
3653      C          *** NRT AND NRC ARE USED TO ADVANCE THE ACTIVE GRID.
3654      NRT=0
3655      NRC=0
3656      C          *** VEL=1. FLAGS FIRST PASS. ON SECOND PASS, VEL = 0.
3657      VEL=1.0
3658      C          *** RC = DISTANCE FROM AXIS TO CENTER OF CELL K.
3659      C          RR = DISTANCE FROM AXIS TO CENTER OF CELL K+1.
3660      10  RC=DX(1)/2.0
3661          RR=RC+(DX(1)+DX(2))/2.0
3662          IF(IGM.EQ.1)RC=1.
3663          IF(IGM.EQ.1)RR=RC
3664          K=2
3665      C          *** FOR ALL CELLS IN COLUMN NEXT TO AXIS, SET PRESSURE
3666      C          AT LEFT SIDE OF CELL = PRESSURE IN CELL, AND SET
3667      C          RADIAL VELOCITY AT LEFT SIDE OF CELL = 0.
3668      DO 20 J=1,JMAX
3669          PL(J)=P(K)
3670          UL(J)=0.0
3671      20  K=K+IMAX

```

```

3672      DO 140 I=1,11
3673      K=I+1
3674      C      *** DEFINE PRESSURE AND AXIAL VELOCITY AT BOTTOM
3675      C      BOUNDARY OF GRID.
3676      VBLO=V(K)
3677      PHLO=P(K)
3678      C      *** IF BOTTOM BOUNDARY OF GRID IS REFLECTIVE, SET
3679      C      AXIAL VELOCITY AT THAT BOUNDARY = 0.
3680      IF (CVIS.GT.(-.5)) VLO=0.
3681      TAUDTS=TAU(I)*DT
3682      DO 130 J=1,12
3683      N=K+IMAX
3684      PIDTS=1.0/(PIDY*DT*DY(J))
3685      IF (IGM.EQ.1)PIDTS=2./(DY(J)*DT)
3686      IF (AMX(K).LE.0.) GO TO 3C
3687      IF (I.LT.IMAX) GO TO 5D
3688      C      *** FOR ALL CELLS IN LAST COLUMN OF GRID, SET PRESSURE
3689      C      AT RIGHT OF CELL = PRESSURE IN CELL. COMPUTE
3690      C      ENERGY LOST ACROSS RIGHT BOUNDARY AND SUBTRACT IT
3691      C      FROM ETH, THEORETICAL ENERGY TOTAL.
3692      PRR=P(K)
3693      E=PRR*U(K)/PIDTS*RC
3694      ETH=ETH-E
3695      EOR=EOR+E
3696      GO TO 4D
3697      C      *** CELL K IS EMPTY
3698      3D      PL(J)=0.
3699      UL(J)=U(K+1)*PR
3700      PBLO=0.
3701      VBLO=V(N)
3702      GO TO 13D
3703      4D      URR=RC*U(K)
3704      GO TO 7D
3705      C      *** IF CELL ON RIGHT IS EMPTY SET SPECIAL P AND U
3706      5D      IF (AMX(K+1).GT.0.) GO TO 6D
3707      PRK=0.
3708      UPR=U(K)*RC
3709      GO TO 7D
3710      6D      PRK=(P(K)+P(K+1))/2.
3711      UPR=(U(K)*RC+U(K+1)*RR)/2.
3712      7D      IF (J.LT.JMAX) GO TO 8D
3713      C      *** FOR ALL CELLS IN TOP ROW OF GRID, SET PRESSURE AND
3714      C      AXIAL VELOCITY AT TOP OF CELL = PRESSURE AND AXIAL
3715      C      VELOCITY IN CELL. COMPUTE ENERGY LOST ACROSS TOP
3716      C      BOUNDARY.
3717      PABOVE=P(K)
3718      E=PABOVE*V(K)/2.*TAUDTS
3719      ETH=ETH-E
3720      EOT=EOT+E
3721      VABOVE=V(K)
3722      GO TO 11D
3723      C      *** IF CELL ABOVE IS EMPTY SET SPECIAL P AND V
3724      8D      IF (AMX(N).GT.0.) GO TO 9D
3725      PABOVE=0.

```

```

3726      VABOVE=V(K)
3727      GO TO 100
3728  90    PABOVE=(P(K)+P(N))/2.
3729      VAROVE=(V(K)+V(N))/2.
3730  100   IF (J.GT.1) GO TO 110
3731  C     *** WHEN BOTTOM BOUNDARY TRANSMITTIVE (CVIS=-1.), COMPUTE
3732  C     ENERGY LOST ACROSS BOTTOM BOUNDARY AND SUBTRACT
3733  C     IT FROM 'ETH', THEORETICAL ENERGY TOTAL.
3734  C     SKIP OUT IF BOTTOM BOUNDARY REFLECTIVE (CVIS=0.) .
3735      IF (CVIS.GT.-.5) GO TO 110
3736      E=PBLO*V(K)/2.*TAUDTS
3737      ETH=ETH+E
3738      EOB=EOB-E
3739  110   IF (VEL.EQ.0.) GO TO 120
3740  C     *** COMPUTE UPDATED VELOCITIES ON FIRST PASS (VEL = 1.)
3741      V(K)=V(K)+(PBLO-PABOVE)*TAUDTS/(AMX(K))
3742      U(K)=U(K)+(PL(J)-PRR)/(AMX(K))*RC/PIDTS*2.0
3743  120   CONTINUE
3744  C     *** ON FIRST PASS ONE HALF THE NEW AIX(K) IS CALCULATED
3745  C     USING GRADIENTS BASED ON OLD VELOCITIES. ON SECOND
3746  C     PASS THE OTHER HALF OF THE NEW AIX(K) IS CALCULATED
3747  C     USING GRADIENTS BASED ON NEW VELOCITIES. NOTE, SOME
3748  C     CELLS ARE 'GLUED' AFTER SECOND PASS TO CORRECT
3749  C     HIGH NEGATIVE INTERNAL ENERGIES.
3750      WS=(VBLO-VAROVE)*TAUDTS/2.
3751      WS=(UL(J)-URR)/PIDTS+WS
3752      WSA = WS*P(K)
3753      AIX(K) = AIX(K)+*WSA/AMX(K)
3754  C
3755      MFK=MFLAG(K)
3756  C     *** IS CELL K PURE
3757      IF(MFK.LT.100) GO TO 124
3758  C     *** CELL K MIXED. PARTITION CHANGE IN INTERNAL ENERGY
3759  C     IN PROPORTION TO FRACTIONAL VOLUME.
3760      M=MFK-100
3761  C     *** IF CELL CONTAINS A FREE SURFACE, DEFINE TOTAL VOLUME
3762  C     (VCELL) TO BE SUM OF VOLUMES OF MATERIALS IN CELL.
3763      VCELL=0.
3764      DO 119 N=1,NMAT
3765      IF(RHO(N,M).LE.0.) GO TO 119
3766      VCELL=VCELL+XMASS(N,M)/RHO(N,M)
3767  119   CONTINUE
3768  C
3769  121   DO 122 N=1,NMAT
3770      IF(XMASS(N,M).LE.0.) GO TO 122
3771      WS = XMASS(N,M)/RHO(N,M)
3772      WS=WS/VCELL
3773  C     *** CHANGE IN INTERNAL ENERGY FOR MATERIAL N.
3774      WSB = WSA+*WS
3775      SIE(N,M) = SIE(N,M) + WSB/XMASS(N,M)
3776  122   CONTINUE
3777  124   CONTINUE
3778      VBLO=VAROVE
3779      PL(J)=PRR

```

```

3780      UL(J)=URR
3781      PBLO=PABOVE
3782      C      *** RC, N, RR REDEFINED FOR NEXT CELL IN ROW J.
3763      130 K=K+IMAX
3784      RC=RR
3785      RR=(X(I+1)+X(I+2))/2.
3786      IF(IGH.EQ.1)RC=1.
3787      IF(IGH.EQ.1)RR=RC
3788      140 CONTINUE
3789      IF(VEL.EQ.0.) GO TO 141
3790      VEL=0.0
3791      GO TO 10
3792      141 CONTINUE
3793      C
3794      WS=0.
3795      DO 1414 K=1,KMAX
3796      M=MFLAG(K)
3797      IF(M.GT.100) GO TO 1412
3798      WS=AMAX1(WS,AIX(K))
3799      GO TO 1414
3800      1412 M=M-100
3801      DO 1413 N=1,MHAT
3802      WS=AMAX1(WS,STE(N,M))
3803      1413 CONTINUE
3804      1414 CONTINUE
3805      WS=SQRT(WS)
3806      UVMAX=AMAX1(WS,UVMAX)
3807      DO 142 I=1,I1
3808      DO 142 J=1,I2
3809      K=(J-1)*IMAX+I+1
3810      IF(ABS(U(K)).GT.UVMAX .OR. ABS(V(K)).GT.UVMAX) MFLAG(K)=-MFLAG(K)
3811      142 CONTINUE
3812      C      *** GLUE SPECIAL CELLS TO CORRECT FOR UNREALISTIC VELOCITIES
3813      CALL GLUE
3814      DO 190 I=1,I1
3815      K=I+1
3816      DO 180 J=1,I2
3817      170 IF (I.NE.1) GO TO 180
3818      C      *** ENLARGE ACTIVE GRID IN I-DIRECTION IF A CELL IN THE I1
3819      C      COLUMN HAS NONZERO VELOCITY OR ENERGY.
3820      IF (U(K).NE.0..OR.V(K).NE.0..OR.AIX(I).NE.0.) NRC=1
3821      180 K=K+IMAX
3822      LL=K-2*IMAX
3823      C      *** ENLARGE ACTIVE GRID IN J-DIRECTION IF A CELL IN THE I2
3824      C      ROW HAS NONZERO VELOCITY OR ENERGY.
3825      IF (U(LL).NE.0..OR.V(LL).NE.0..OR.AIX(LL).NE.0.) NRT=1
3826      190 CONTINUE
3827      I1=I1+NRC
3828      I2=I2+NRT
3829      C      *** DONT ALLOW ACTIVE GRID TO EXCEED IMAX BY JMAX GRID.
3830      IF (I1-IMAX) 210,210,200
3831      200 I1=IMAX
3832      210 IF (I2-JMAX) 230,230,220
3833      220 I2=JMAX

```

```

3834 230 RETURN
3835 C
3836 240 FORMAT (4H PH1,2I4,4H M=,1PE15.8,6H SIE=,1PE15.8,4H U=,1PE15.8,
3837 14H V=,1PE15.8,16H SIE SET TO ZERO)
3838 END
3839 SUBROUTINE PH2
3840 C      *** ACCOUNTS FOR MASS FLUX AND ASSOCIATED TRANSPORTS OF
3841 C      MOMENTUM AND ENERGY.
3842 INCLUDE COMDIM
3843 DIMENSION SYMBO(8)
3844 DATA SYMBO/2H 1,2H 2,2H 3,2H 4,2H 5,2H .2H M,2H +/
3845 C      *** INITIALIZE ACTIVE GRID COUNTERS AND DEFINE CONSTANTS.
3846 C
3847 NRT=0
3848 NRC=0
3849 SUME=0.
3850 REZ=0.
3851 PIDTS=1.0/(PIDY*DT)
3852 TWOPDT=2.0*PIDY*DT
3853 DO 150 K=1,KMAX
3854 150 P(K)=0.
3855 C
3856 C      *** DEFINE RADIAL MOMENTUM AT AXIS. SET AXIAL MOMENTUM,
3857 C      ENERGY, AND MASS FLUX AT AXIS TO 0.
3858 C
3859 K=2
3860 DO 200 J=1,JMAX
3861 IF (AMX(K).LE.0.) GO TO 160
3862 IF (U(K).LT.0.) GO TO 170
3863 160 FLEFT(J) = 0.
3864 GO TO 190
3865 C      *** DEFINE MASS FLUX AS THOUGH LEFT CELL BOUNDARY WERE NOT
3866 C      THE AXIS - AFTER MU DEFINED, MASS FLUX SET TO ZERO.
3867 C
3868 170 CONTINUE
3869 178 GAMC(J) = AMX(K)*U(K)*DT/DX(1)
3870 IF (GAMC(J) + AMX(K) .GE. 0.) GO TO 180
3871 GAMC(J) = -AMX(K)
3872 180 FLEFT(J) = 2.*GAMC(J)*U(K)/SS2
3873 C
3874 190 GAMC(J) = 0.
3875 YAMC(J) = 0.
3876 SIGC(J) = 0.
3877 200 K=K+1MAX
3878 C
3879 C.....
3880 C
3881 C      PH2 COMPUTES MASS FLUX TERMS OF PURE CELLS ONLY.
3882 C      THE MASS FLUX TERMS OF THE MIXED CELLS HAVE
3883 C      ALREADY BEEN COMPUTED IN INFAC AND FLUX.
3884 C      PH2 DOES THE ACTUAL TRANSPORT FOR BOTH PURE AND MIXED
3885 C      CELLS.
3886 C
3887 C.....

```

```

3884 C
3889 C
3890 C      *** BEGIN LOOP ON I = PH2 CALCULATES A COLUMN AT A TIME.
3891 C
3892 DO 1150 I=1,I1
3893 IF (IGM.EQ.1) P1OTS=2./DT
3894 IF (IGM.EQ.1) T*VOPDT=DT/X(I)
3895 J=1
3896 K=I+1
3897 DO 205 N=1,NMAT
3898 SDELEB(N)=0.
3899 SAMMY(N)=0.
3900 205 CONTINUE
3901 IF (AMX(K) .1220,230,210)
3902 210 MFK=IABS(MFLAG(K))
3903 C      *** DEFINE FLUXES AT BOTTOM GRID BOUNDARY.
3904 220 IF ((-V(K)) .GT. UMIN) GO TO 240
3905 230 AMMV = 0.
3906 GO TO 290
3907 240 IF (MFK.GT.100) GO TO 245
3908 AMMY=AMX(K)*V(K)*DT/DY(J)
3909 IF (AMMY+AMX(K).LT.0.) AMMY=-AMX(K)
3910 DELEB=(AIX(K)+(U(K)**2 + V(K)**2)**.5)*AMMY
3911 GO TO 247
3912 245 DELEB=0.
3913 AMMY=0.
3914 M=MFK-100
3915 DO 246 N=1,NMAT
3916 SAMMY(N)=RHO(N,M)*V(K)*DT*TAU(I)
3917 IF (SAMMY(N)+XMASS(N,M).LT.0.) SAMMY(N)=-XMASS(N,M)
3918 SDELEB(N)=(SIE(N,M)+(U(K)**2+V(K)**2)**.5)*SAMMY(N)
3919 AMMY=AMMY+SAMMY(N)
3920 DELEB=DELEB+SDELEB(N)
3921 246 CONTINUE
3922 247 IF (CVIS.GE.0.) GO TO 280
3923 AMMU=AMMY*U(K)
3924 AMMV=AMMY*V(K)
3925 EMOB = EMOB + DELEB
3926 ETH = ETH + DELEB
3927 BOTM = BOTM - AMMY
3928 BOTMV = BOTMV - AMMV
3929 BOTMU = BOTMU - AMMU
3930 GO TO 300
3931 C      *** REFLECTIVE BOTTOM BOUNDARY
3932 280 IF (V(K).GE.0.) GO TO 230
3933 AMMV = 2.*AMMY*V(K)/SS2
3934 290 AMMY = 0.
3935 AMMU = 0.
3936 DELEB = 0.
3937 300 CONTINUE
3938 C
3939 C
3940 C      *** BEGIN LOOP ON J.
3941 DO 1140 J=1,I2

```

```

3942      VABOVE=D.
3943      URR=D.
3944      AMPY=D.
3945      AMMP=D.
3946      C      *** DEFINE FLUXES AT TOP BOUNDARY OF CELL.
3947      C      *** L IS INDEX OF CELL ABOVE CELL K.
3948      L = K+JMAX
3949      C
3950      MFK=IABS(MFLAG(K))
3951      MFL=IABS(MFLAG(L))
3952      IF(MFK.GT.100) GO TO 470
3953      C      *** SET FLUX TO ZERO IF EITHER CELL IS FLAGGED ZERO.
3954      IF(MFK.EQ.0 .OR. (MFL.EQ.0 .AND. J.LT.JMAX)) GO TO 360
3955      C..... CELL K PURE.
3956      C
3957      C      *** SPECIAL TESTS FOR TOP BOUNDARY OF GRID.
3958      310 IF(J.EQ.JMAX .AND. V(K).GT.0.) GO TO 320
3959      IF(J.EQ.JMAX .AND. V(K).LE.0.) GO TO 360
3960      IF(AMX(K)) 1220,360,314
3961      C
3962      314 IF(AMX(L)) 1220,315,316
3963      C      *** CELL ABOVE EMPTY, SHOULD TRANSPORT INTO IT BE ALLOWED?
3964      315 IF(V(K).GT.0. .AND. MFL.GT.100) GO TO 320
3965      GO TO 360
3966      C      *** BOTH NON-EMPTY
3967      316 WSA=(V(K)+V(L))*S
3968      WS=DT/DY(J)
3969      WSR=1.0+(V(L)-V(K))*WS
3970      IF(ABS(V(K))*WS.GT.STAB .OR. ABS(V(L))*WS.GT.STAB) WSR=1.0
3971      VABOVE=WSA/WSR
3972      C
3973      IF(ABS(VABOVE).LT.UMIN) GO TO 360
3974      IF(VABOVE) 319,360,324
3975      319 IF(MFL.GT.100) GO TO 350
3976      C      *** DONOR CELL(ABOVE) IS PURE
3977      M=L
3978      DTODY=DT/DY(J+1)
3979      GO TO 326
3980      C      *** CELL ABOVE EMPTY.
3981      320 VABOVE=V(K)
3982      C      *** DONOR CELL (K) IS PURE
3983      324 M=K
3984      DTODY=DT/DY(J)
3985      C      *** FLUX DEFINITION FOR PURE DONOR.
3986      326 AMPY=AMX(M)*VABOVE*DTODY
3987      GO TO 455
3988      C      *** DONOR CELL (ABOVE) IS MIXED. TRANSPORT ONLY CELL K MAT.
3989      350 IF(XMASS(MFK,MFL-100).LE.0.) GO TO 360
3990      AMPY=RHO(MFK,MFL-100)*VABOVE*DT*TAU(I)
3991      FAMPY=SIE(MFK,MFL-100) + .5*(U(L)**2 + V(L)**2)
3992      M=L
3993      GO TO 458
3994      C      *** NO FLUX ACROSS TOP BOUNDARY OF CELL K.
3995      360 AMPY=C.

```

```

3996      GO TO 460
3997      C      *** DONOR CELL PURE.
3998      455 EAMPY = AIX(IH) + .5*(U(IH)**2 + V(IH)**2)
3999      C
4000      458 UAMPY = U(IH)
4001      VAMPY = V(IH)
4002      IF(ABS(AMPY).GT.RUEPS*AMX(K) .AND. AMX(K).GT.0.) GO TO 459
4003      IF(ABS(AMPY).GT.RUEPS*AMX(L) .AND. AMX(L).GT.0.) GO TO 459
4004      AMPY=0.
4005      459 DELET= AMPY*EAMPY
4006      C      *** L IS INDEX OF CELL TO RIGHT OF CELL K.
4007      460 L=K+1
4008      C      *** DEFINE FLUXES AT RIGHT BOUNDARY OF CELL.
4009      MFL=IABS(MFLAG(L))
4010      C      *** SET FLUX TO ZERO IF EITHER CELL IS FLAGGED ZERO.
4011      IF(MFK.EQ.0 .OR. (MFL.EQ.0 .AND. I.LT.IMAX)) GO TO 560
4012      C
4013      C      *** SPECIAL TESTS FOR RIGHT BOUNDARY OF GRID.
4014      510 IF(I.EQ.IMAX .AND. U(K).GT.0.) GO TO 520
4015      IF(I.EQ.IMAX .AND. U(K).LE.0.) GO TO 560
4016      IF(AMX(K)) 1220,560,514
4017      C      *** IS CELL ON RIGHT EMPTY?
4018      514 IF(AMX(L)) 1220,515,516
4019      C      *** YES. DOES IT CONTAIN A MATERIAL INTERFACE?
4020      515 IF(U(K).GT.0. .AND. MFL.GT.100) GO TO 520
4021      GO TO 560
4022      C      *** BOTH CELLS ARE NON-EMPTY. COMPUTE TRANSPORT VELOCITY.
4023      516 WSA=(U(K)+U(L))*5
4024      WSB=DT/DX(I)
4025      WSB=1.0+(U(L)-U(K))*WS
4026      IF(ABS(U(K))*WS.GT.STAB .OR. ABS(U(L))*WS.GT.STAB) WSB=1.0
4027      URR=WSA/WSB
4028      C
4029      IF(ABS(URR).LT.UMIN) GO TO 560
4030      IF(URR) 519, 560, 524
4031      519 IF(MFL.GT.100) GO TO 550
4032      C      *** DONOR CELL ON RIGHT IS PURE.
4033      M=L
4034      AREA=TAU(I+1)
4035      GO TO 526
4036      C      *** CELL ON RIGHT EMPTY.
4037      520 URR=U(K)
4038      C      *** DONOR CELL K IS PURE.
4039      524 N=K
4040      AREA=TAU(I)
4041      C      *** MASS FLUX IF DONOR IS PURE
4042      526 AMMP=AMX(N)/AREA*TAOPDT*X(I)*URR
4043      GO TO 655
4044      C      *** DONOR CELL ON RIGHT IS MIXED. TRANSPORT ONLY CELL K
4045      C      MATERIAL.
4046      550 IF(XMASS(MFK,MFL-100).LE.0.) GO TO 560
4047      EAMP=RHQ(MFK,MFL-100)*URR*TAOPDT*X(I)*DY(J)
4048      AMMP=51E(MFK,MFL-100) + .5*(U(L)**2 + V(L)**2)
4049      M=L

```



```

4050      GO TO 658
4051      C      *** NO FLUX ACROSS RIGHT BOUNDARY OF CELL K.
4052      560 AMMP=0.
4053      GO TO 960
4054      C      *** DONOR CELL PURE.
4055      655 EAMMP = A*(X(M) + .5*(U(M)**2 + V(M)**2)
4056      C
4057      658 UAMMP = U(M)
4058      VAMMP = V(M)
4059      IF (ABS(AMMP).GT.ROEPS*AMX(K) .AND. AMX(K).GT.0.) GO TO 659
4060      IF (ABS(AMMP).GT.ROEPS*AMX(L) .AND. AMX(L).GT.0.) GO TO 659
4061      AMMP=0.
4062      659 DELER= AMMP*EAMMP
4063      GO TO 960
4064      C***** CELL K MIXED. COMPUTE ENERGY FLUX FOR EACH MATERIAL.
4065      670 MK=MFK-100
4066      DELER=0.
4067      DELET=0.
4068      AMPY=0.
4069      AMKP=0.
4070      KT=K
4071      KR=K
4072      MT=MK
4073      MR=MK
4074      DO 680 N=1,NMAT
4075      IF (SAMPY(N,MK)) 690,680,700
4076      680 CONTINUE
4077      GO TO 705
4078      690 KT=K+1MAX
4079      MT=1ABS(MFLAG(KT))-100
4080      WSX=AIX(KT)
4081      700 UAMPY=U(KT)
4082      VAMPY=V(KT)
4083      705 DO 710 N=1,NMAT
4084      IF (SAMMP(N,MK)) 720,710,730
4085      710 CONTINUE
4086      GO TO 735
4087      720 KR=K+1
4088      MR=1ABS(MFLAG(KR))-100
4089      WSY=AIX(KR)
4090      730 UAMMP=U(KR)
4091      VAMMP=V(KR)
4092      735 WSA=.5*(U(KT)**2 + V(KT)**2)
4093      WSB=.5*(U(KR)**2 + V(KR)**2)
4094      DO 600 N=1,NMAT
4095      SDELET(N)=0.
4096      SDELER(N)=0.
4097      IF (ABS(SAMPY(N,MK)).LE.C.) GO TO 760
4098      IF (MT.GT.0) WSX=SIE(N,MT)
4099      SDELET(N)=SAMPY(N,MK)*(WSX+WSA)
4100      DELET=DELET+SDELET(N)
4101      AMPY=AMPY+SAMPY(N,MK)
4102      760 IF (ABS(SAMMP(N,MK)).LE.0.) GO TO 780
4103      IF (MR.GT.0) WSY=SIE(N,MR)

```

```

4104      SDELER(N)=SAMMP(N,MK)*(WSY+WSB)
4105      DELER=DELER+SDELER(N)
4106      AMMP=AMMP+SAMMP(N,MK)
4107      780 SDELM(N)=-SAMPY(N,MK)-SAMMP(N,MK)+SGAMC(N,J)+SAMMY(N)
4108      800 CONTINUE
4109      H=MK
4110      960 DELM=-AMPY-AMMP+GAMC(J)+AMMY
4111      970 IF(ABS(AMPY).LE.0.) GO TO 980
4112      C          CALCULATE ENERGY AND MOMENTUM FLUX AT TOP
4113      AMUT=AMPY*UAMPY
4114      AMVT=AMPY*VAMPY
4115      C          IS THIS AT TOP BOUNDARY
4116      IF (J.NE.JMAX) GO TO 99C
4117      C          YES, TOP. ADJUST ENERGY.
4118      ETH=ETH-DELET
4119      EMOT=EMOT+DELET
4120      TOPM=TOPM+AMPY
4121      TOPMV=TOPMV+AMVT
4122      TOPMU=TOPMU+AMUT
4123      C          IS AMPY LARGE ENOUGH TO TRIGGER REZONE
4124      IF (AMPY/(TAU(I)*DY(J)).GE.VT) REZ=1.
4125      GO TO 99D
4126      C          AMPY=0. SET MOMENTUM AND ENERGY FLUX=0.
4127      980 AMUT=0.
4128      AMVT=0.
4129      DELET=0.
4130      990 IF(ABS(AMMP).LE.0.) GO TO 1000
4131      C          CALCULATE ENERGY + MOMENTUM FLUX AT RIGHT
4132      AMUR=AMMP*UAMMP
4133      AMVR=AMMP*VAMMP
4134      C          IS THIS AT RIGHT BOUNDARY
4135      IF (I.NE.IMAX) GO TO 1010
4136      C          YES, RIGHT. ADJUST ENERGY.
4137      ETH=ETH-DELER
4138      EMOR=EMOR+DELER
4139      RTN=RTN+AMMP
4140      PTMV=RTMV+AMVR
4141      RTHU=RTMU+AMUR
4142      C          IS AMMP LARGE ENOUGH TO TRIGGER REZONE
4143      IF (AMMP/(TAU(I)*DY(J)).GE.VT) REZ=1.
4144      GO TO 1010
4145      C          AMMP=0. SET MOMENTUM AND ENERGY FLUX=0.
4146      1000 AMUR=0.
4147      AMVR=0.
4148      DELER=0.
4149      C          REPARTITION ENERGY + MOMENTUM
4150      1010 CONTINUE
4151      1020 WSA=.5*(U(K)**2+V(K)**2)
4152      SIGMU=-AMUT-AMUR+AMMU+FLFT(J)
4153      SIGMV=-AMVT-AMVR+AMMV+YAMC(J)
4154      WS=(FLM+AMX(K)
4155      UNEW=(SIGMU+AMX(K)*U(K))/WS
4156      DELU=UNEW-U(K)
4157      U(K)=UNEW

```

```

4158 1030 VNEW=(SIGMV+AMX(K)*V(K))/WS
4159 DELV=VNEW-V(K)
4160 V(K)=VNEW
4161 SIENEW=0.
4162 C      *** IS CELL K PURE?
4163 IF (MFK.LT.100) GO TO 1040
4164 C..... CELL K MIXED. COMPUTE NEW S.I.E. FOR EACH MATERIAL
4165 C      AND FOR ENTIRE CELL.
4166 WS=0.
4167 TIE=0.
4168 MK=MFK-100
4169 DO 1038 N=1,NMAT
4170 IF (ABS(SDELM(N)).LE.0.) GO TO 1037
4171 C      *** TOTAL ENERGY CHANGE FOR MATERIAL N = SUSR
4172 SWSB = -SDELET(N)-SDELER(N)+SSIGC(N,J)+SDELEB(N)
4173 C      *** NEW VALUE FOR MASS OF MATERIAL N = SWS(N)
4174 SWS = SDELM(N)+XMASS(N,M)
4175 IF (ABS(SWS).GT.0.) GO TO 1034
4176 SIE(N,MK)=0.
4177 XMASS(N,MK)=0.
4178 GO TO 1038
4179 1034 SWSA=((SIE(N,MK)+WSA)*XMASS(N,MK) + SWSB)/SWS - .5*(U(K)**2 +
4180 | V(K)**2)
4181 DELI=SWSA-SIE(N,MK)
4182 IF (ABS(DELI).GT.UMIN**2) GO TO 1035
4183 SUME=SUME+DELI*S+S
4184 GO TO 1036
4185 1035 SIE(N,MK)=SWSA
4186 1036 XMASS(N,MK)=SWS
4187 1037 WS=WS+XMASS(N,MK)
4188 TIE=TIE + XMASS(N,MK)*SIE(N,MK)
4189 1038 CONTINUE
4190 IF (ABS(WS).LE.0.) GO TO 1050
4191 SIENEW=TIE/WS
4192 GO TO 1050
4193 C
4194 C..... CELL K IS PURE
4195 C
4196 1040 WSR=-DELET-DELER+DELEB+SIGC(J)
4197 IF (ABS(WS).LE.0.) GO TO 1050
4198 SIENEW=((AIX(K)+WSA)*AMX(K)+WSB)/WS - .5*(U(K)**2+V(K)**2)
4199 DELI=SIENEW-AIX(K)
4200 IF (ABS(DELI).GT.UMIN**2) GO TO 1050
4201 SUME=SUME+DELI*WS
4202 GO TO 1060
4203 1050 AIX(K)=SIENEW
4204 1060 AMX(K)=WS
4205 1090 IF (I.NE.II) GO TO 1100
4206 IF (ABS(U(K)).GT.0..OR.ABS(V(K)).GT.0..OR.ABS(AIX(K)).GT.0.) NRC=,
4207 C      *** SPECIAL INTERMEDIATE PRINT FOR CHECKING ENERGY
4208 C      CONSERVATION - PRINTS ONLY IF INTER = 7 IN INPUT DECK.
4209 1100 IF (INTER.NE.7) GO TO 1130
4210 ENERGY=DELER+DELET-SIGC(J)
4211 DO 1110 N=1,IMAX

```

```

4266      ANMU=AMUT
4267      AMMV=AMVT
4268      DELEB=DELET
4269      C
4270      C          *** END OF J-LOOP.
4271      C
4272      1140  K=K+IMAX
4273           LL=K-IMAX
4274           IF (ABS(U(LL)),GT.0.,OR,ABS(V(LL)),GT.0.,OR,ABS(AIX(LL)),GT.0.)
4275           J  NRT=1
4276      C
4277      C          *** END OF I-LOOP.
4278      C
4279      1150  CONTINUE
4280      C          *** ADVANCE ACTIVE GRID.
4281           11=11+NRC
4282           12=12+NRT
4283           IF (IMAX-11) 1160,1170,1180
4284      1160  11=IMAX
4285      1170  CONTINUE
4286      1180  IF (JMAX-12) 1190,1200,1210
4287      1190  12=JMAX
4288      1200  CONTINUE
4289      1210  GO TO 1230
4290      C          *** NEGATIVE MASS
4291      1220  NK=315.
4292           NR=13
4293      1225  CALL ERROR
4294      1230  SUM=0.0
4295      C          *** MAKE ADJUSTMENTS FOR OVER-EMPTIED CELLS
4296           DO 1280 JP=1,12
4297           DO 1270 I=1,11
4298           J=JP
4299           IF (J.LE.JPROJ,OR,JPROJ.EQ.0) GO TO 1281
4300           J=12-JP+JPROJ+1
4301      1281  K=(J-1)*IMAX+I+1
4302      C
4303           MFK=ABS(HFLAG(K))
4304           IF (MFK.GT.100) GO TO 1226
4305           IF (AMX(K).GE.0.) GO TO 1270
4306      C          *** PURE CELL OVER-EMPTIED
4307           WRITE(6,1700)I,J
4308           NK=1226
4309           NR=13
4310           CALL ERROR
4311      1226  DO 1227 L=1,NMAT
4312           IF (XMASS(L,MFK-100).LT.0.) GO TO 1228
4313      1227  CONTINUE
4314           GO TO 1270
4315      C          *** MATERIAL L ( IN A MIXED CELL ) OVER-EMPTIED.
4316      1228  N=L
4317           M=MFK-100
4318           MSY=XMASS(N,M)
4319           MSY=51E(N,M)

```

```

4212 ENERGY=ENERGY+SIGC(INN)
4213 1110 CONTINUE
4214 DO 1120 LJD=2,KMAX
4215 ENERGY=ENERGY+AMX(LJD)*(AIX(LJD)+.5*(U(LJD)**2+V(LJD)**2))
4216 1120 CONTINUE
4217 WRITE (6,1300) I,J,ENERGY
4218 WRITE (6,1310) AMPY,AMMP,AMHY,GAMC(J)
4219 WRITE (6,1320) DELET,DELER,DELEB,SIGC(J)
4220 IF(IABS(MFLAG(K)).LT.100) GO TO 1130
4221 WRITE(6,1321) (H,SDELET(N),SDELER(N),SDELEB(N),SSIGC(N,J),
4222 I N=1,NMAT)
4223 1321 FORMAT(13,8H SDELET=IPE15.8,6X,7HSDELER=IPE15.8,6X,7HSDELEB=,
4224 I IPE15.8,6X,6HSSIGC=IPE15.8)
4225 1130 CONTINUE
4226 C *** IS CELL K PURE
4227 IF(MFK.GT.100) GO TO 1131
4228 IF(MFK.GT.0) GO TO 1134
4229 DO 1133 N=1,NMAT
4230 SAMMY(N)=0.
4231 SDELEB(N)=0.
4232 SGAMC(N,J)=0.
4233 SSIGC(N,J)=0.
4234 1133 CONTINUE
4235 GO TO 1138
4236 C
4237 C***** CELL K MIXED. STORE FLUXES TO BE USED IN TRANSPORT
4238 C OF EACH MATERIAL FOR CELLS ABOVE AND ON RIGHT.
4239 C
4240 1131 CONTINUE
4241 DO 1132 N=1,NMAT
4242 SAMMY(N) = SAMPY(N,MK)
4243 SDELEB(N) = SDELET(N)
4244 SGAMC(N,J) = SAMNP(N,MK)
4245 SSIGC(N,J) = SDELER(N)
4246 1132 CONTINUE
4247 GO TO 1138
4248 C
4249 C***** CELL K PURE. STORE TOTAL FLUXES TO BE USED BY CELLS
4250 C ABOVE AND ON RIGHT.
4251 C
4252 C *** IS CELL ABOVE PURE?
4253 1134 IF(IABS(MFLAG(K+IMAX)).LT.100) GO TO 1136
4254 SAMMY(MFK) = AMPY
4255 SDELEB(MFK)=DELET
4256 C *** IS CELL ON RIGHT PURE?
4257 1136 IF(IABS(MFLAG(K+1)).LT.100) GO TO 1138
4258 SGAMC(MFK,J) = AMMP
4259 SSIGC(MFK,J)=DELER
4260 C
4261 1138 GAMC(J)=AMHP
4262 FLEFT(J)=AMUF
4263 YANC(J)=AMVR
4264 SIGC(J)=DELER
4265 AMHY=AMPY

```

```

4320      XMASS(N,M)=0.
4321      SIE(N,M)=0.
4322      WS=AMX(K)-WSX
4323      AIX(K)=(AIX(K)*AMX(K)-WSX*WSY)/WS
4324      AMX(K)=WS
4325 1229  UNEV=0.
4326      VNEV=0.
4327      SIENEV=0.
4328      WS=C.
4329  C      *** FIND NEIGHBOR WITH GREATEST AMOUNT OF THAT MATERIAL
4330  C      *** WHICH HAS OVER-EMPTIED.
4331      KT=K+IMAX
4332      IF(J.EQ.JMAX) GO TO 1236
4333      IT=J
4334      JT=J+1
4335 1231  MT=IABS(MFLAG(KT))
4336      IF(MT.LT.100) GO TO 1232
4337      WSA=XMASS(N,MT-100)
4338      GO TO 1234
4339 1232  IF(MT.NE.N) GO TO 1236
4340      WSA=AMX(KT)
4341 1234  IF(WSA.LE.WS) GO TO 1236
4342      WS=WSA
4343      NWS=KT
4344 1236  IF(KT.NE.K+IMAX) GO TO 1238
4345      KT=K-1
4346      IF(I.EQ.1) GO TO 1238
4347      IT=I-1
4348      JT=J
4349      GO TO 1231
4350 1238  IF(KT.NE.K-1) GO TO 1240
4351      KT=K-IMAX
4352      IF(J.EQ.1) GO TO 1240
4353      IT=I
4354      JT=J-1
4355      GO TO 1231
4356 1240  IF(KT.NE.K-IMAX) GO TO 1242
4357      KT=K+1
4358      IF(I.EQ.IMAX) GO TO 1242
4359      IT=I+1
4360      JT=J
4361      GO TO 1231
4362 1242  IF(WS.GE.ABS(WSX)) GO TO 1244
4363      WRITE(6,1450)I,J,MFK
4364      WS=(U(K)**2 + V(K)**2)/2.0
4365      EVAPM=EVAPM + WSX
4366      WS=WSX*(WSY+WS)
4367      EVAPEN=EVAPEN+WS
4368      ETH = ETH-WS
4369      EVAPMU=EVAPMU+WSX*U(K)
4370      EVAPMV=EVAPMV+WSX*V(K)
4371      WRITE(6,1460)I,J,N,MFK,WSX,WSY
4372      GO TO 1269
4373  C      *** REMOVE MASS FROM CHOSEN NEIGHBOR (NWS) AND ADJUST

```

```

4374 C KINETIC AND INTERNAL ENERGIES .
4375 1244 MFN=IABS(MFLAG(NWS))
4376 AM=AMX(NWS)+WSX
4377 VNEW=(AMX(NWS)*V(NWS) + WSX*V(K))/AM
4378 UNEW=(AMX(NWS)*U(NWS) + WSX*U(K))/AM
4379 WSA=.5*(U(NWS)**2 + V(NWS)**2)
4380 WSH=.5*(UNEW**2 + VNEW**2)
4381 WSC=.5*(U(K)**2+V(K)**2)
4382 IF(INTER.EQ.0) GO TO 1246
4383 WRITE(6,888) I,J,MFK,MFN,L,NWS,WSX,WSY,AM,V(K),V(NWS),VNEW
4384 888 FORMAT(18H I,J,MFK,MFN,L,NWS,6I6/28H WSX,WSY,AM,V(K),V(NWS),VNEW,
4385 1 1P6E15.7)
4386 1246 CONTINUE
4387 EK=ASX*(WSY + WSC)
4388 IF(MFN.GT.100) GO TO 1260
4389 C *** CELL (NWS) IS PURE.
4390 ENWS=AMX(NWS)*(AIX(NWS) + WSA)
4391 SIENEW = (EK+ENWS)/AM - WSB
4392 GO TO 1258
4393 C *** CELL (NWS) IS MIXED.
4394 1260 MFN=MFN-100
4395 ESUM=0.
4396 SUM=0.
4397 XM=XMASS(N,MFN) + WSX
4398 IF(INTER.EQ.0) GO TO 1261
4399 WRITE(6,889) IT,JT,MFN,(XMASS(L,MFN),SIE(L,MFN),L=1,NMAT)
4400 889 FORMAT(22H IT,JT,MFN, XMASS, SIE, 3I6,/40X,(1P2E20.8))
4401 1261 CONTINUE
4402 DO 1266 L=1,NMAT
4403 IF(L.EQ.N) GO TO 1262
4404 IF(ABS(XMASS(L,MFN)).LE.0.) GO TO 1266
4405 SIE(L,MFN)=SIE(L,MFN)+(WSA-WSB)
4406 GO TO 1264
4407 1262 TE = XMASS(L,MFN)*(SIE(L,MFN) + WSA) + EK
4408 SIE(L,MFN)=TE/XM - WSB
4409 XMASS(L,MFN)=XM
4410 1264 ESUM=ESUM+SIE(L,MFN)*XMASS(L,MFN)
4411 SUM=XMASS(L,MFN) +SUM
4412 1266 CONTINUE
4413 IF(INTER.EQ.0) GO TO 1267
4414 WRITE(6,889) IT,JT,MFN,(XMASS(L,MFN),SIE(L,MFN),L=1,NMAT)
4415 1267 CONTINUE
4416 SIENEW=ESUM/SUM
4417 1268 AMX(NWS)=AM
4418 U(NWS)=UNEW
4419 V(NWS)=VNEW
4420 AIX(NWS)=SIENEW
4421 IF(INTER.EQ.0) GO TO 1269
4422 WRITE(6,887) SIENEW
4423 887 FORMAT(7H SIENEW,1PE20.8)
4424 1269 CONTINUE
4425 C
4426 IF(ABS(AMX(K)).GT.0.) GO TO 1226
4427 AIX(K)=0.

```

```

4428      U(K)=0.
4429      V(K)=0.
4430      IF(MFK.LT.100) MFLAG(K)=0
4431 1270 CONTINUE
4432 1280 CONTINUE
4433 C      *** EVAPORATE MASS NOT EVACUATED DUE TO MACHINE ROUND-OFF.
4434      DO 1650 K=2,KMAX
4435      M=ABS(MFLAG(K))
4436      IF(M.LT.100) GO TO 1650
4437      M=M-100
4438      TIE=0.
4439      DO 1620 N=1,NMAT
4440      IF(RHO(N,M).GT.0. .OR. XMASS(N,M).LE.0.) GO TO 1615
4441      WS=(U(K)**2+V(K)**2)/2.0
4442      DIFF=XMASS(N,M)
4443      EVAPM=EVAPM+DIFF
4444      WS=DIFF*(SIE(N,M)+WS)
4445      EVAPEN=EVAPEN+WS
4446      ETH=ETH-WS
4447      EVAPMU=EVAPMU+DIFF*U(K)
4448      EVAPMV=EVAPMV+DIFF*V(K)
4449      J=(K-2)/IMAX+1
4450      I=(K-1)-IMAX*(J-1)
4451      WRITE(6,1430) I,J,N,XMASS(N,M),RHO(N,M),SIE(N,M)
4452      AMX(K)=AMX(K)-DIFF
4453      XMASS(N,M)=0.
4454      SIE(N,M)=0.
4455 1615 TIE=TIE+SIE(N,M)*XMASS(N,M)
4456 1620 CONTINUE
4457      AIX(K)=TIE/AMX(K)
4458 1650 CONTINUE
4459 C      *** REDEFINE FLAGS OF CELLS THAT BECAME PURE
4460      DO 1288 K=2,KMAX
4461      IF(MFLAG(K).GE.0) GO TO 1288
4462      MK=-MFLAG(K)-100
4463      MFLAG(K)=0
4464      RHO(NVOID,MK)=0.
4465      DO 1284 N=1,NMAT
4466      IF(RHO(N,MK).GT.0.) MFLAG(K)=N
4467      RHO(N,MK)=0.
4468      XMASS(N,MK)=0.
4469      SIE(N,MK)=0.
4470 1284 CONTINUE
4471      IF(MFLAG(K).GT.0) GO TO 1285
4472      AMX(K)=0.
4473      AIX(K)=0.
4474      U(K)=0.
4475      V(K)=0.
4476 1285 CONTINUE
4477      RHO(1,MK)=-1.
4478 1288 CONTINUE
4479 C      *** PRINT SYMBOLIC MAP OF ACTIVE GRID DISPLAYING
4480 C      THE LOCATION OF THE MATERIAL PACKAGES(NUMBERED CELLS)
4481 C      AND THE MIXED CELLS (LABELED 'H').

```



```

4482      IF(NPRINT.EQ.0 .AND.NC.GT.0) GO TO 1508
4483      WRITE(6,1370)
4484      IOL=11
4485      JDL=12
4486      IF(NC.GT.0.) GO TO 1501
4487      IDL=MIND(IMAX,54)
4488      JDL=JMAX
4489      C
4490      1501 J=JDL
4491      1502 DO 1504 I=1,IDL
4492          PR(I)=SYMBOL(6)
4493          K=(J-1)*IMAX+I+1
4494          MFK = MFLAG(K)
4495          IF(MFK.LT.100) PR(I) = SYMBOL(MFK)
4496          IF(MFK.EQ.0) PR(I) = SYMBOL(6)
4497          IF(MFK.GT.100) PR(I) = SYMBOL(7)
4498      1504 CONTINUE
4499      C
4500      IF(MOD(J,5).NE.0) GO TO 1505
4501      WRITE(6,1380) J,(PR(I),I=1,IDL)
4502      GO TO 1506
4503      1505 WRITE(6,1390) (PR(I),I=1,IDL)
4504      1506 J=J-1
4505      IF(J.EQ.0) GO TO 1507
4506      GO TO 1502
4507      1507 PR(I)=SYMBOL(8)
4508      WRITE(6,1380) J,(PR(I),I=1,IDL)
4509      WRITE(6,1400) (I,I=0,IDL,5)
4510      1508 CONTINUE
4511      C      *** ETH = THEORETICAL ENERGY SUM, USED IN EDIT FOR
4512      C      ENERGY CHECK.
4513      C      *** EZPHZ = ENERGY SET TO ZERO IN PH2 SINCE TIME=0.
4514      C      *** SUME = SUM OF THE ENERGY FLUXES IGNORED ON THIS CYCLE.
4515      ETH=ETH-SUME
4516      EZPHZ=EZPHZ-SUME
4517      RETURN
4518      C
4519      1290 FORMAT (5H NEGM,13,14,4H M=,1PE14.7,6H DELM=,1PE14.7,6H BOT=,1PE
4520      114.7,7H LEFT=,1PE14.7,6H TOP=,1PE14.7,5H RT=,1PE14.7)
4521      1300 FORMAT (5H I= 13,6X,5H J= 13,6X,9H ENERGY=1PE15.8)
4522      1310 FORMAT (7H AMPY=1PE15.8,6X,6H AMHP=1PE15.8,6X,6H AMHY=1PE15.8,9H
4523      1GAMC(J)=1PE15.8)
4524      1320 FORMAT (7H DELET=1PE15.8,6X,6HDELER=1PE15.8,6X,6HDELEB=1PE15.8,9H
4525      1SIGC(J)=1PE15.8)
4526      1330 FORMAT (4H PH2,214,4H M=,1PE15.8,6H SIE=,1PE15.8,4H U=,1PE15.8,
4527      14H V=,1PE15.8,19H SIE SET TO ZERO)
4528      1340 FORMAT (4H PH2,214,4H M=,1PE15.8,6H SIE=,1PE15.8,4H U=,1PE15.8,
4529      14H V=,1PE15.8,19H CELL EVAPORATED)
4530      1350 FORMAT (12H ADJUST FLUX,4H M=,1PE14.7,6H DELM=,1PE14.7,6H BOT=,
4531      11PE14.7,7H LEFT=,1PE14.7,6H TOP=,1PE14.7,5H RT=,1PE14.7)
4532      1360 FORMAT (12H ADJUST MASS,4H M=,1PE14.7,6H DELM=,1PE14.7,6H BOT=,
4533      11PE14.7,7H LEFT=,1PE14.7,6H TOP=,1PE14.7,5H RT=,1PE14.7)
4534      1370 FORMAT(11H,3INDISPLAY OF MIXED AND PURE CELLS,5X,14HM = MIXED CELL
4535      1      5X,43HNUMERAL N = PURE CELL OF PACKAGE N MATERIAL//)

```

```

4536 1380 FORMAT (I10,2H 1,54A2)
4537 1390 FORMAT (I0X,2H 1,54A2)
4538 1400 FORMAT (I12,I0I10//)
4539 1410 FORMAT(/20H MASS LEAK IN COLUMN,16,6H, ROW, 16)
4540 1420 FORMAT (1H1,26HFREE SURFACE TRACERS ADDED //
4541 1 (6(16,2X,F5.2,2X,F5.2)))
4542 1430 FORMAT(4H PH2,214,4H N=,14,4H M=,1PE15.8,6H RHO=,1PE15.8,
4543 1 6H SIE=,1PE15.8,19H MASS EVAPOATED)
4544 1440 FORMAT(64H ERROR CONDITION- MIXED CELL AT TRANSMITTIVE BOTTOM BOUN
4545 1DARY I=,14, 4H J=,14,11H MFLAG(K)=,14)
4546 1450 FORMAT(8H I,J,MFK, 316, 29H ISOLATED NEGATIVE MASS - PH2)
4547 1460 FORMAT(/4H PH2,214,4H N=,14,8H MFLAG=,14,6H MASS=,1PE16.8,
4548 1 6H SIE=,1PE16.8,21H NEG. MASS EVAPORATED/)
4549 1700 FORMAT(1X,216,5X,22HPURE CELL OVEREMPTIED.)
4550 END
4551 SUBROUTINE PH3
4552 C *** COMPUTES EFFECTS OF DEVIATORIC AND HOOP STRESSES
4553 C TO UPDATE CELL VELOCITIES AND ENERGIES.
4554 INCLUDE COMDIM
4555 M=I+1
4556 C *** CALCULATE FACTOR FOR VARIABLE DT.
4557 C
4558 C
4559 ICY=INT(CYCPH3)
4560 DN=0.
4561 DO 10 I=1,ICY
4562 10 DN=DN+FLOAT(I)
4563 C
4564 C *** LOOP THROUGH SUBCYCLES
4565 C
4566 DO 500 LJH=1,ICY
4567 DTFACT=FLOAT(ICY-LJH+1)/DN
4568 DTSTR=DT*DTFACT
4569 C *** INITIALIZE P ARRAY
4570 DO 18 K=1,KMAX
4571 18 P(K)=0.
4572 C *** DEFINE POINTERS USED FOR STORING STRAIN RATES
4573 C THREE ROWS AT A TIME.
4574 NKA=3
4575 NK=2
4576 NKB=1
4577 C *** DEFINE STRAIN RATES FOR FIRST TWO ROWS OF GRID.
4578 DO 30 J=1,2
4579 NT=J+1
4580 VFACT=1.0
4581 IF(J.EQ.1 .AND. CVIS,GE.C.) VFACT=-1.0
4582 C
4583 DO 27 I=1,I1
4584 K=(J-1)*INAX+I+1
4585 IK=I+1
4586 MFK=MFLAG(K)
4587 C *** WHEN MNB=20, MATERIAL IN CELL IS AN IDEAL GAS. SKIP OUT.
4588 MNB=0
4589 IF(MFK.LT.100) MNB=MAT(MFK)

```

```

4590      IF(AMX(K).LE.C. .OR. MNB.EQ.20) GO TO 27
4591      KA=K+JMAX
4592      KB=K-JMAX
4593      KR=K+J
4594      KL=K-J
4595      MFKA=MFLAG(KA)
4596      MFKB=MFLAG(KB)
4597      MFKR=MFLAG(KR)
4598      MFKL=MFLAG(KL)
4599      UFACT=1.0
4600      C
4601      WSX=1.0/(.5*DX(I-1)+DX(I)+.5*DX(I+1))
4602      C
4603      C..... ADJUST TERMS IF CELL ON RIGHT IS VOID OR OUTSIDE GRID.
4604      C
4605      IF(AMX(KR).GT.0..AND.I.LT.IMAX)GO TO 21
4606      WSX=1.0/(.5*(DX(I-1)+DX(I)))
4607      KR=K
4608      C      *** ADJUST TERMS IF CELL IS IN AXIS COLUMN.
4609      21 IF(I.GT.1) GO TO 22
4610      WSX=1.0/(1.5*DX(I)+.5*DX(2))
4611      KL=K
4612      UFACT=-1.0
4613      GO TO 23
4614      C
4615      C..... ADJUST TERMS IF CELL ON LEFT IS VOID.
4616      C
4617      22 IF(AMX(KL).GT.0.)GO TO 23
4618      WSX=1.0/(.5*(DX(I)+DX(I+1)))
4619      KL=K
4620      C
4621      23 WSY=1.0/(.5*DY(J-1)+DY(J)+.5*DY(J+1))
4622      C
4623      C..... ADJUST TERMS IF CELL ABOVE IS VOID OR OUTSIDE GRID.
4624      C
4625      IF(AMX(KA).GT.0..AND.J.LT.JMAX)GO TO 24
4626      WSY=1.0/(.5*(DY(J-1)+DY(J)))
4627      KA=K
4628      C      *** ADJUST TERMS IF CELL IS IN BOTTOM ROW OF GRID.
4629      24 IF(J.GT.1) GO TO 25
4630      WSY=1.0/(1.5*DY(I)+.5*DY(2))
4631      KB=K
4632      GO TO 26
4633      C
4634      C..... ADJUST TERMS IF CELL BELOW IS VOID.
4635      C
4636      25 IF(AMX(KB).GT.0.)GO TO 26
4637      WSY=1.0/(.5*(DY(J)+DY(J+1)))
4638      KB=K
4639      C
4640      26 DVODX=(V(KR)-V(KL))*WSX
4641      DUODX=(U(KR)-U(KL))*UFACT*WSX
4642      DVODY=(V(KA)-V(KB))*VFACT*WSY
4643      DUODY=(U(KA)-U(KB))*KSY

```

```

4644      UGX=U(K)/(X(I)+X(I-1))*2.
4645      THO3 = (DUODX + DVODY + UGX)/3.
4646      EZZ(IK,NT) = DVODY - THO3
4647      EKRIK,NT) = DUODX - THO3
4648      ERZ(IK,NT) = (DUODY + DVODX)/2.
4649      SZZ(IK,NT)=STFSZZ(K)
4650      SRR(IK,NT)=STFSRR(K)
4651      SRZ(IK,NT)=STFSRZ(K)
4652      27 CONTINUE
4653      C      *** DEFINE STRAIN RATES FOR CELLS IN DUMMY COLUMN LEFT OF
4654      C      AXIS.
4655      EZZ(1,NT) = EZZ(2,NT)
4656      ERR(1,NT) = ERR(2,NT)
4657      ERZ(1,NT) = ERZ(2,NT)
4658      SZZ(1,NT)=SZZ(2,NT)
4659      SRR(1,NT)=SRR(2,NT)
4660      SRZ(1,NT)=SRZ(2,NT)
4661      30 CONTINUE
4662      C      *** DEFINE STRAIN RATES FOR CELLS IN DUMMY ROW BELOW GRID.
4663      DO 35 IK=1,M
4664      EZZ(IK,1) = EZZ(IK,2)
4665      ERR(IK,1) = ERR(IK,2)
4666      ERZ(IK,1) = ERZ(IK,2)
4667      SZZ(IK,1)=SZZ(IK,2)
4668      SRR(IK,1)=SRR(IK,2)
4669      SRZ(IK,1)=SRZ(IK,2)
4670      35 CONTINUE
4671      C.....
4672      C
4673      C      *** COMPUTE NEW CELL-CENTERED STRESSES - MOVING ACROSS
4674      C      ROWS.
4675      C
4676      C.....
4677      DO 100 J=1,12
4678      K=(J-1)*IMAX+2
4679      DO 50 I=1,11
4680      MFK=MFLAG(K)
4681      JK=J+1
4682      C      *** WHEN MNB=20, MATERIAL IN CELL IS AN IDEAL GAS, SKIP OUT
4683      MNE=0
4684      IF(MFK.LT.100) MNB=MAT(MFK)
4685      IF((AM)(K).LE.0. .OR. MNB.EQ.20) GO TO 45
4686      CALL STPNG
4687      STPENG=FS
4688      36 IF(STPENG.LE.0.)GO TO 45
4689      TKC = 2.*STPENG**2
4690      C      *** DETERMINE POSITION OF MATERIAL WHICH WILL BE
4691      C      AT CELL CENTER AFTER PH2.
4692      DXT = -U(K)*DT
4693      DYT = -V(K)*DT
4694      ADYT = ABS(DYT)
4695      ADXT = ABS(DXT)
4696      C      *** HORIZONTAL, VERTICAL, DIAGONAL WEIGHTING FACTORS (AREAS)
4697      WK = (DX(I)-DXT)*(DY(J)-ADYT)

```

```

4698 WKH = ADXT*(DY(J)-ADYT)
4699 WKV = ADYT*((X(I)-ADXT)
4700 WKD = ADXT*ADYT
4701 WSUM = WK + WKH + WKV + WKD
4702 C      *** DETERMINE INDICES OF CELLS USED TO CALCULATE
4703 C      INTERPOLATED VALUES OF STRAIN RATES AND STRESSES.
4704     KHS = K-1
4705     KHE = 1K-1
4706     IF(DXT.LT.D.) GO TO 40
4707     KHS = K+1
4708     KHE = 1K+1
4709 C
4710 40 KVS = K-1MAX
4711     KVE = NK
4712     IF(DYT.LT.D.) GO TO 42
4713     KVS = K+1MAX
4714     KVE = NKA
4715 C
4716 42 KDS = KVS-1
4717     IF(DXT.GT.D.) KDS = KVS+1
4718 C      *** REFFINE INDICES IF NEIGHBOR CELLS ARE EMPTY,
4719 C      OUTSIDE GRID, OR ARE MIXED.
4720     MFH = MFLAG(KHS)
4721     MFV = MFLAG(KVS)
4722     IF(AMX(KHS).LT.D..OR.MFH.EQ.20.OR.KHE-1.GT.IMAX)KHE=1K
4723     IF(AMX(KVS).LT.D..OR.MFV.EQ.20.OR.KVS.GT.KMAX)KVE=NK
4724 C      *** CORRECT FOR CONVECTION BY INTERPOLATING STRAINS
4725 C      AND STRESSES AT POINT(DXT,DYT).
4726     EZZINT = (WK*EZZ(IK,NK) + WKH*EZZ(KHE,NK) + WKV*EZZ(IK,KVE) +
4727 1     WKD*EZZ(KHE,KVE))/WSUM
4728     ERRINT = (WK*ERR(IK,NK) + WKH*ERR(KHE,NK) + WKV*ERR(IK,KVE) +
4729 1     WKD*ERR(KHE,KVE))/WSUM
4730     ERZINT = (WK*ERZ(IK,NK) + WKH*ERZ(KHE,NK) + WKV*ERZ(IK,KVE) +
4731 1     WKD*ERZ(KHE,KVE))/WSUM
4732 C
4733     SZZINT = (WK*SZZ(IK,NK) + WKH*SZZ(KHE,NK) + WKV*SZZ(IK,KVE) +
4734 1     WKD*SZZ(KHE,KVE))/WSUM
4735     SRRINT = (WK*SRR(IK,NK) + WKH*SRR(KHE,NK) + WKV*SRR(IK,KVE) +
4736 1     WKD*SRR(KHE,KVE))/WSUM
4737     SRZINT = (WK*SRZ(IK,NK) + WKH*SRZ(KHE,NK) + WKV*SRZ(IK,KVE) +
4738 1     WKD*SRZ(KHE,KVE))/WSUM
4739 C      *** CALCULATE NEW STRAINS AND STRESSES
4740     IF(MFK.GT.100)GO TO 53
4741     YSA=RMU(MFK)
4742     GO TO 55
4743 53 MKF=MFK-100
4744     YSA=0.
4745     VCELL=TAU(I)*DY(J)
4746     DO 54 MM=1,NMAT
4747     IF(XMASS(MM,MKF).LE.C.) GO TO 54
4748     VOLM=XMASS(MM,MKF)/(RHO(MM,MKF)*VCELL)
4749     YSA=YSA+RMU(MM)*VOLM
4750 54 CONTINUE
4751 55 YS=2.*YSA*DTST

```

```

4752      STRSZZ(K) = SZZINT + EZZINT*WS
4753      STRSRR(K) = SRRINT + ERRINT*WS
4754      STRSRZ(K) = SRZINT + ERZINT*WS
4755      C
4756      C      *** HAS YIELD POINT BEEN EXCEEDED
4757      C
4758      TK1 = (STRSZZ(K)**2 + STRSRR(K)**2 + STRSRZ(K)**2 +
4759      I      STRSZZ(K)*STRSRR(K))*2.
4760      IF(TK1.GT.TKO) GO TO 441
4761      IF(TK1.LT.TKO*1.E-06) GO TO 45
4762      GO TO 50
4763      C      *** REDUCE STRESSES
4764      441 WS = SQRT(TKO/TK1)
4765      STRSZZ(K) = STRSZZ(K)*WS
4766      STRSRR(K) = STRSRR(K)*WS
4767      STRSRZ(K) = STRSRZ(K)*WS
4768      GO TO 50
4769      45 STRSZZ(K)=0.
4770      STRSRR(K)=0.
4771      STRSRZ(K)=0.
4772      C
4773      C      *** END OF I-LOOP FOR NEW STRESSES
4774      C
4775      50 K=K+1
4776      C
4777      C      *** DEFINE NK,NKA,NKB FOR NEXT ROW.
4778      C      COMPUTE ANOTHER ROW OF STRAIN RATES.
4779      C
4780      IF(J.EQ.12) GO TO 100
4781      NKA=NKA+1
4782      NK = NK+1
4783      NKB=NKB+1
4784      IF(NKA.GT.3) NKA=1
4785      IF(NK.GT.3) NK=1
4786      IF(NKB.GT.3) NKB=1
4787      IF(J+2.GT.JHAX) GO TO 100
4788      C      *** DEFINE STRAIN RATES IN THE ROW ABOVE THE NEXT ONE TO BE
4789      C      CALCULATED (J+2).
4790      KK=(J+1)*IMAX+2
4791      DO 90 I=1,11
4792      MFK=MFLAG(KK)
4793      IK=I+1
4794      C      *** WHEN MNB=20, MATERIAL IN CELL IS AN IDEAL GAS. SKIP OUT
4795      MNU=0
4796      IF(MFK.LT.100) MNB=MAT(MFK)
4797      IF(AMX(KK).LE.D. .OR. MNB.EQ.20) GO TO 85
4798      KR=KK+1
4799      KL=KK-1
4800      KB=KK-IMAX
4801      KA=KK+IMAX
4802      MFKA=MFLAG(KA)
4803      MFKB=MFLAG(KB)
4804      MFKR=MFLAG(KR)
4805      MFKL=MFLAG(KL)

```

```

4806          UFACT=1.0
4807          C
4808          WSX=1.0/(1.5*DX(I-1)+DX(I)+.5*DX(I+1))
4809          C
4810          C***** ADJUST TERMS IF CELL ON RIGHT IS VOID OR OUTSIDE GRID.
4811          C
4812          IF (AMX(KR).GT.0..AND.1.LT.IMAX) GO TO 60
4813          WSX=1.0/(1.5*(DX(I-1)+DX(I)))
4814          KR=KK
4815          C          *** ADJUST TERMS IF CELL IS IN AXIS COLUMN.
4816          60 IF (I.GT.1) GO TO 62
4817          WSX=1.0/(1.5*DX(I)+.5*DX(2))
4818          KL=KK
4819          UFACT=-1.0
4820          GO TO 64
4821          C
4822          C***** ADJUST TERMS IF CELL ON LEFT IS VOID.
4823          C
4824          62 IF (AMX(KL).GT.0.) GO TO 64
4825          WSX=1.0/(1.5*(DX(I)+DX(I+1)))
4826          KL=KK
4827          C
4828          C***** ADJUST TERMS IF CELL ABOVE IS VOID OR OUTSIDE GRID.
4829          C
4830          64 IF (AMX(KA).GT.0..AND.J.LT.JMAX) GO TO 66
4831          WSY=1.0/(1.5*(DY(J+2)+DY(J+1)))
4832          KA=KK
4833          C
4834          C***** ADJUST TERMS IF CELL BELOW IS VOID.
4835          C
4836          66 IF (AMX(KB).GT.0.) GO TO 70
4837          WSY=1.0/(1.5*(DY(J+2)+DY(J+3)))
4838          KB=KK
4839          C
4840          70 CONTINUE
4841          DVODY=(V(KA)-V(KB))*WSY
4842          DUODY=(U(KA)-U(KB))*WSX
4843          DVODX=(V(KR)-V(KL))*WSX
4844          DUODX=(U(KR)-U(KL))*UFACT*WSX
4845          UOX=U(KK)/(X(I)+X(I-1))*2.
4846          THO3=(DUODX+DVODY+UOX)/3.
4847          FZZ(IK,NKA)=DVODY-THO3
4848          ERR(IF,NKA)=DUODX-THO3
4849          ERZ(IK,NKA)=(DUODY+DVODX)/2.
4850          SZZ(IK,NKA)=STRSZZ(KK)
4851          SRK(IK,NKA)=STPSRK(KK)
4852          SRZ(IK,NKA)=STPSRZ(KK)
4853          GO TO 90
4854          C
4855          85 FZZ(IK,NKA)=0.
4856          ERR(IF,NKA)=0.
4857          ERZ(IK,NKA)=0.
4858          SZZ(IK,NKA)=0.
4859          SRK(IF,NKA)=0.

```

```

4860         SRZ(IK,NKA) = 0.
4861     C
4862     90 KK=KK+1
4863     C     *** DEFINE STRAIN RATES FOR CELLS IN DUMMY COLUMN ON
4864     C     LEFT OF AXIS.
4865         EZZ(1,NKA)= EZZ(2,NKA)
4866         ERR(1,NKA)= ERR(2,NKA)
4867         ERZ(1,NKA)= ERZ(2,NKA)
4868         SZZ(1,NKA)=SZZ(2,NKA)
4869         SPR(1,NKA)=SPR(2,NKA)
4870         SRZ(1,NKA)=SRZ(2,NKA)
4871     C
4872     C     *** END OF J-LOOP FOR NEW STRESSES
4873     C
4874     100 CONTINUE
4875     C
4876     C.....
4877     C     *** COMPUTE STRESSES AT CELL BOUNDARIES. THEN
4878     C     UPDATE VELOCITIES.
4879     C.....
4880     C
4881     C     *** DEFINE POINTERS USED FOR STORING OLD VELOCITIES
4882     C     THREE ROWS AT A TIME.
4883         NKA=3
4884         NK=2
4885         NKE=1
4886     C     *** STORE OLD VELOCITIES OF FIRST TWO ROWS.
4887         DO 200 NT=2,3
4888         KK=(NT-2)*IMAX+2
4889         DO 190 JK=2,K
4890         UK(JK,NT) = U(KK)
4891         VK(JK,NT) = V(KK)
4892         RHOC(JK,NT) = AMX(KK)/(TAU(JK-1)*DY(NT-1))
4893     190 KK=KK+1
4894     C     *** DEFINE VELOCITIES FOR CELLS IN COLUMN ON LEFT OF AXIS
4895         UK(1,NT) = -UK(2,NT)
4896         VK(1,NT) = VK(2,NT)
4897         RHOC(1,NT) = RHOC(2,NT)
4898     200 CONTINUE
4899     C     *** DEFINE VELOCITIES FOR CELLS IN DUMMY ROW BELOW GRID.
4900         VFACT=1.
4901         IF(CVIS.GE.0.) VFACT=-1.
4902         DO 210 JK=1,K
4903         UK(JK,1) = UK(JK,2)
4904         VK(JK,1) = VK(JK,2)*VFACT
4905         RHOC(JK,1) = RHOC(JK,2)
4906     210 CONTINUE
4907     C
4908         DO 300 J=1,12
4909         K=(J-1)*IMAX+2
4910     C     *** SET TO 0. STRESSES AT AXIS. SNB, STB SET TO 0. WHEN
4911     C     P ARRAY INITIALIZED.
4912         STJ=0.
4913         STI=0.

```



```

4914 C
4915 DO 250 I=1,11
4916 IK=I+1
4917 IKR=IK+1
4918 MFK=MFLAG(K)
4919 STP = 0.
4920 SNR = 0.
4921 SNT = 0.
4922 STT = 0.
4923 HOOB=0.
4924 DELI=0.
4925 IF(MFK.GT.100)GO TO 37
4926 N=HAT(MFK)
4927 IF(AMX(K).LE.0. .OR. N.EQ.20) GO TO 230
4928 SOLID=AMDH(MFK)*RHOZ(N)
4929 IF(RHOC(IK,NK).LT.SOLID) GO TO 230
4930 GO TO 211
4931 37 MFK=MFK-100
4932 IF(RHO(INVOID,MFK) .GT. 0.) GO TO 230
4933 DO 38 MM=1,NHAT
4934 IF(XMASS(MM,MFK).LE.0.) GO TO 38
4935 N=HAT(MM)
4936 SOLID=AMDH(MM)*RHOZ(N)
4937 IF(RHO(MM,MFK).LT.SOLID)GO TO 230
4938 38 CONTINUE
4939 C
4940 C *** COMPUTE STRESSES AT RIGHT OF CELL.
4941 C
4942 211 KR=K+1
4943 IF(I.EQ.IMAX) GO TO 212
4944 IF(I.EQ.11) GO TO 213
4945 MFKR=MFLAG(KR)
4946 IF(MFKR.GT.100)GO TO 41
4947 N=HAT(MFKR)
4948 IF(AMX(KR).LE.0. .OR. N.EQ.20) GO TO 213
4949 SOLID=AMDH(MFKR)*RHOZ(N)
4950 IF(RHOC(IKR,NK).LT.SOLID)GO TO 213
4951 GO TO 44
4952 41 MFKR=MFKR-100
4953 IF(RHO(INVOID,MFKR) .GT. 0.) GO TO 213
4954 DO 43 MM=1,NHAT
4955 IF(XMASS(MM,MFKR).LE.0.) GO TO 43
4956 N=HAT(MM)
4957 SOLID=AMDH(MM)*RHOZ(N)
4958 IF(RHO(MM,MFKR).LT.SOLID)GO TO 213
4959 43 CONTINUE
4960 44 CONTINUE
4961 C *** NORMAL CASE.
4962 SNR = (STRSRR(K) + STRSRR(KR))*0.5
4963 STR = (STRSRZ(K) + STRSRZ(KR))*0.5
4964 GO TO 214
4965 C *** CELL AT RIGHT GRID BOUNDARY.
4966 212 SNR=STRSRR(K)
4967 STR=STRSRZ(K)

```

```

4968      ETHCHG=(SNR*U(K)+STR*V(K))*DY(J)*DTSTR
4969      IF(IGM,NE.1)ETHCHG=ETHCHG*TWOPI*X(1)
4970      ETH=ETH+ETHCHG
4971      IKR = IK
4972      GO TO 214
4973 C          *** CELL ON RIGHT EMPTY OR CONTAINS FREE SURFACE, OR IS
4974 C          UNDERDENSE
4975      213 SNR=0.
4976          STR=0.
4977 C
4978 C          *** COMPUTE STRESSES AT TOP OF CELL.
4979 C
4980      214 KA=K+IMAX
4981          IF(J.EQ.JMAX) GO TO 215
4982          IF(J.EQ.I2) GO TO 216
4983          MFKA=MFLAG(KA)
4984          IF(MFKA.GT.100)GO TO 46
4985          N=MAT(MFKA)
4986          IF(AMX(KA).LE.0. .OR. N.EQ.20) GO TO 216
4987          SOLID=AMDH(MFKA)*RHOZ(N)
4988          IF(RHOC(IK,NKA).LT.SOLID)GO TO 216
4989          GO TO 49
4990      46 MFAK=MFKA-100
4991          IF(RHO(NVOID,MFAK).GT. 0.) GO TO 216.
4992          DO 47 MM=1,NMAT
4993          IF(XMASS(MM,MFAK).LE.0.) GO TO 47
4994          N=MAT(MM)
4995          SOLID=AMDH(MM)*RHOZ(N)
4996          IF(RHO(MM,MFAK).LT.SOLID)GO TO 216
4997      47 CONTINUE
4998      48 CONTINUE
4999 C          *** NORMAL CASE.
5000          SNT = (STRSZZ(K) + STRSZZ(KA))*0.5
5001          STT = (STRSRZ(K) + STRSRZ(KA))*0.5
5002          GO TO 217
5003 C          *** CELL AT TOP GRID BOUNDARY.
5004      215 SNT=STRSZZ(K)
5005          STT=STRSRZ(K)
5006          ETH=ETH+(SNT*V(K)+STT*U(K))*TAU(1)*DTSTR
5007          NKA=NK
5008          GO TO 217
5009 C          *** CELL ABOVE EMPTY OR CONTAINS FREE SURFACE OR IS
5010 C          UNDERDENSE
5011      216 SNT=0.
5012          STT=0.
5013 C
5014 C          *** COMPUTE HOOP STRESS.
5015 C
5016      217 HOOP = -(STRSZZ(K) + STRSHR(K))
5017 C
5018 C          *** DEFINE STRESSES AT BOTTOM IF CELL IN FIRST ROW.
5019 C
5020          IF(J.GT.1) GO TO 220
5021          SNB(IK) = STRSZZ(K)

```

```

5022      IF(CVIS.GE.D.) GO TO 220
5023      STB(K)= ST*SNZ(K)
5024      ETH=ETH+(SNB(IK)*V(K)+STB(IK)*U(K))*TAU(I)*DTSTR
5025      C
5026      C      *** COMPUTE NEW VALUES OF U, V, SIE FOR CELL K.
5027      C
5028      220 SNLX = SNL*X(I-1)
5029      WS = TWOPI*DTSTR/AMX(K)
5030      DELU = WS*(DY(J)*(SNR*X(I)-SNLX)+TAU(I)/TWOPI*(STT-STB(IK))
5031      1      - HODP*DX(I)*DY(J))
5032      STLX = STL*X(I-1)
5033      DELV = WS*((SNT-SNB(IK))*TAU(I)/TWOPI + DY(J)*(STR*X(I)-STLX))
5034      C
5035      UKT = UK(IK,NK)
5036      VKT = VK(IK,NK)
5037      WS = TAU(I)*.5*((UKT+UK(IK,NKA))*STT + (VKT+VK(IK,NKA))*SNT)
5038      1      - ((UKT+UK(IK,NKB))*STB(IK) + (VKT+VK(IK,NKB))*SNB(IK))
5039      C
5040      WSA = PIDY*DY(J)*(X(I)+((UK(IKR,NK)+UKT)*SNR + (VK(IKR,NK)+VKT)
5041      1      *STR) - (X(I-1)+((UK(IK-1,NK))+UKT)*SNL + (VK(IK-1,NK)
5042      2      *STL)))
5043      C
5044      IF(IGH.NE.1)GO TO 2221
5045      WSA=DTSTR/AMX(K)
5046      DELU=WSA*(DY(J)*(SNR-SNL)+TAU(I)*(STT-STB(IK)))
5047      DELV=WSA*(DY(J)*(STR-STL)+TAU(I)*(SNT-SNB(IK)))
5048      WSA=.5*DY(J)*((UK(IKR,NK)+UKT)*SNR+(VK(IKR,NK)+VKT)*STR
5049      1-(UK(IK-1,NK)+UKT)*SNL-(VK(IK-1,NK)+VKT)*STL)
5050      2221 WSA=(WSA+WS)*DTSTR/AMX(K)
5051      WSC = DELU*(UKT+DELU/2.)+DELV*(VKT+DELV/2.)
5052      DELI = WSB-WSC
5053      U(K) = U(K)+DELU
5054      V(K) = V(K)+DELV
5055      TKI=(STRSZ(K)**2 + STRSRR(K)**2 + STRSRZ(K)**2 +
5056      1      STRSZZ(K)*STRSRR(K))*2.
5057      CALL STRNG
5058      STRENG=WS
5059      221 TKC = 2.*STRENG**2
5060      IF(LIK.LT.100) GO TO 225
5061      MKF=MFK-100
5062      VCELL=TAU(I)*DY(J)
5063      DO 222 MM=1,NMAT
5064      IF(XMASS(MM,MKF).LE.D.) GO TO 222
5065      PV = 1.0/(RHO(MM,MKF)*VCELL)
5066      SIE(MM,MKF)=SIE(MM,MKF)+PV*DELI*AMX(K)
5067      IF(TK)*(.1.+RUFPS) .LT. TKC) GO TO 222
5068      PLW(MM) = PLW(MM)+PV*DELI*AMX(K)*XMASS(MM,MKF)
5069      222 CONTINUE
5070      GO TO 226
5071      225 IF(TK)*(.1.+RUFPS) .LT. TKC) GO TO 226
5072      C      *** PLW(M) IS THE TOTAL PLASTIC WORK OF MATERIAL PACKAGE M.
5073      C      AROUND IS THE TOTAL ELASTIC PLASTIC WORK OF THE GRID.
5074      PLW(MFK) = PLW(MFK)+DELI*AMX(K)
5075      226 AIZ(K) = AIZ(K)+DELI

```

```

5076 C
5077 BBOUND = BBOUND + DELI*AMX(K)
5078 C
5079 C *** CELL K IS DONE. SAVE STRESSES FOR CELL ON RIGHT
5080 C AND CELL ABOVE.
5081 C
5082 230 CONTINUE
5083 C
5084 239 SNL=SNR
5085 STL = STR
5086 SNB(1K) = SNT
5087 STG(1K) = STT
5088 C
5089 IF (INTER.NE.99) GO TO 250
5090 F=0.
5091 PW=0.
5092 DO 240 LJD=2,KMAX
5093 E=E+AMX(LJD)*(.5*(U(LJD)**2+V(LJD)**2)+ATX(LJD))
5094 240 CONTINUE
5095 WRITE (6,440) J,J,E
5096 DO 245 LJD=2,JK
5097 UBAR=.5*(UK(LJD,NK)+UK(LJD,NKA))
5098 VBAR=.5*(VK(LJD,NK)+VK(LJD,NKA))
5099 245 E=E-TAU(LJD-1)*(UBAR*STB(LJD)+VBAR*SNB(LJD))*DTSTR
5100 IKK=IK+1
5101 DO 247 LJD=IKK,11
5102 UBAR=.5*(UK(LJD,NKA)+UK(LJD,NKB))
5103 VBAR=.5*(VK(LJD,NKA)+VK(LJD,NKB))
5104 247 E=E-TAU(LJD-1)*(UBAR*STB(LJD)+VBAR*SNB(LJD))*DTSTR
5105 UBAR=.5*(UK(IK+1,NK)+UK(IK,NK))
5106 VBAR=.5*(VK(IK+1,NK)+VK(IK,NK))
5107 ETHCHG=DY(J)*(UBAR*SNL+VBAR*STL)*DTSTR
5108 IF (IGN.NE.1) ETHCHG=ETHCHG*TDOP1*X(1)
5109 E=F-ETHCHG
5110 WRITE (6,440) J,J,E
5111 PW=PW+DELI*AMX(K)
5112 WRITE (6,450) PW
5113 C *** END OF I-LOOP FOR NEW VELOCITIES
5114 C
5115 250 K=K+1
5116 C
5117 C *** DEFINE NK,NKA,NKB FOR NEXT ROW.
5118 C STORE ANOTHER ROW OF OLD VELOCITIES.
5119 C
5120 IF (J.EQ.12) GO TO 300
5121 NKA=NKA+1
5122 NK = NK+1
5123 NKB=NKB+1
5124 IF (NKA.GT.3) NKA=1
5125 IF (NK.GT.3) NK=1
5126 IF (NKB.GT.3) NKB=1
5127 IF (J+2.GT.JMAX) GO TO 300
5128 KK=(J+1)*IMAX+2
5129 GO 290 IK=2,11

```

```

5130      UK(IK,NKA) = U(KK)
5131      VK(IK,NKA) = V(KK)
5132      RHOC(IK,NKA) = AMX(KK)/(TAU(IK-1)*DY(J+2))
5133      290 KK=KK+1
5134      UK(1,NKA) = UK(2,NKA)
5135      VK(1,NKA) = VK(2,NKA)
5136      RHOC(1,NKA) = RHOC(2,NKA)
5137      C
5138      C          *** END OF J-LOOP FOR NEW V-LOCITIES.
5139      C
5140      300 CONTINUE
5141      C
5142      C          *** END OF ONE SUBCYCLE.
5143      C
5144      500 CONTINUE
5145      C          *** INITIALIZE P ARRAY.
5146      DO 600 K=2,KMAX
5147      600 P(K)=0.
5148      RETURN
5149      C
5150      440 FORMAT (4X,2HJ=12,4X,2HJ=12,4X,2HE=1PE13.7)
5151      450 FORMAT (4X,3HPW=1PE12.6)
5152      C
5153      END
5154      SUBROUTINE PROPRT
5155      C          *** DEFINES MIXED CELL VARIABLES FOR CELLS THAT ARE
5156      C          ON THE BOUNDARY OF A RECTANGULAR PACKAGE. CALLED
5157      C          FROM SETUP WHEN GENERATING A PROBLEM.
5158      INCLUDE COMBIN
5159      C          *** ASSIGN PROPERTIES TO BOUNDARY CELL K OF PACKAGE M.
5160      C          ROUTINE CALLED WHEN SETTING UP A RECTANGULAR PACKAGE.
5161      IF(MFK.GT.100) GO TO 15
5162      MA=40
5163      MFLAG(K)=100+40
5164      MO=MO+1
5165      IF(MO.LE.NHXCLS) GO TO 10
5166      NK=10
5167      NR=2
5168      WRITE(6,100)
5169      100 FORMAT(33H GENERATING TOO MANY MIXED CELLS.)
5170      10 RHO(M,MA)=0.
5171      IF(MFK.EQ.0) GO TO 15
5172      RHO(MFK,MA)=RHOIN(MFK)
5173      XMASS(MFK,MA)=AMX(K)
5174      SIE(MFK,MA)=AIX(K)
5175      15 RHO(M,MA)=RHOIN(M)
5176      IF(M.EQ.NVOID) RHO(M,MA)=1.0
5177      RETURN
5178      END
5179      SUBROUTINE REZONE
5180      C . . . . .
5181      C          SUBROUTINE REZONE REZONES THE GRID. IF TEXTX=1 AND IMAX IS AN
5182      C          EVEN NUMBER, THE GRID IS REZONED IN THE X DIRECTION. IF JEXTY=1
5183      C          AND JMAX IS AN EVEN NUMBER, THE GRID IS REZONED IN THE Y DIRECTION.

```

```

5184 C .....
5185      INCLUDE COMDIA
5186      WRITE(6,10)IC,IEXTX,JEXTY
5187 10 FORMAT(//,5X,'REZONE CALLED ON CYCLE',15,5X,'IEXTX =',12,
5188      1X,'JEXTY =',12,/)
5189      IF(IEXTX.EQ.0.AND.JEXTY.EQ.0)RETURN
5190 C .....
5191 C      ZERO THE PRESSURE ARRAY.
5192 C .....
5193      DO 20 K=1,KMAX
5194      20 P(K)=0.
5195      IF(IEXTX.EQ.0)GO TO 170
5196      NIMAX=IMAX/2
5197      IF(2*NIMAX.EQ.IMAX)GO TO 40
5198 C .....
5199 C      ERROR FOUND. IEXTX=1 BUT IMAX NOT EVEN. GRID NOT REZONED.
5200 C .....
5201      WRITE(6,30)IMAX
5202 30 FORMAT(5X,'IMAX =',14,' WHICH IS NOT AN EVEN NUMBER. THE GRID WAS
5203      1 NOT REZONED IN THE X DIRECTION.')
5204      GO TO 170
5205 C .....
5206 C      GRID TO BE REZONED IN A DIRECTION.
5207 C .....
5208      40 NIMAX1=NIMAX+1
5209 C .....
5210 C      REDEFINE THE X COORDINATES OF THE TRACERS.
5211 C .....
5212      DO 65 N=1,NVOID
5213      NP=NMP(N)
5214      IF(NP.LE.0)GO TO 65
5215      DO 60 M=1,NP
5216      I=INT(TX(N,M))
5217      IF(I.LT.IMAX.AND.IVARDX.EQ.1)GO TO 50
5218      TX(N,M)=.5*TX(N,M)
5219      GO TO 60
5220 50 TXM=TX(N,N)-FLOAT(I)
5221      J=1-2*(I/2)
5222      L=J+1
5223      TX(N,M)=.5*FLOAT(I-J)+(TXM*DX(I+1)+FLOAT(J)*DX(I))/(DX(L)+DX(L+1))
5224 60 CONTINUE
5225 65 CONTINUE
5226      NP=(IMAX/2+1)*(JMAX/2+1)
5227      DO 70 N=1,NP
5228      I=INT(XP(N))
5229      IF(I.LT.IMAX.AND.IVARDX.EQ.1)GO TO 68
5230      XP(N)=.5*XP(N)
5231      GO TO 70
5232 68 XPM=XP(N)-FLOAT(I)
5233      J=1-2*(I/2)
5234      L=J+1
5235      XP(N)=.5*FLOAT(I-J)+(XPM*DX(J+1)+FLOAT(J)*DX(I))/(DX(L)+DX(L+1))
5236 70 CONTINUE
5237 C .....

```

```

5238 C FIND THE BOTTOM OF THE BOTTOM PACKAGE WHICH EXTENDS OUT OF THE GRID.
5239 C .....
5240 YMIN=FLOAT(JMAX-1)
5241 IF(MBRR.LE.0)GO TO 76
5242 DO 74 I=1,MBRR
5243 MB=MFAC(MBRR)
5244 IF(MB.LE.0)GO TO 74
5245 DO 72 J=1,MB
5246 IF(PACY(I,J).LT.YMIN)YMIN=PACY(I,J)
5247 72 CONTINUE
5248 74 CONTINUE
5249 C .....
5250 C STORE CELL PROPERTIES AT RIGHT EDGE OF OLD GRID AND ABOVE
5251 C YMIN SO THAT THEY CAN BE USED FOR FILLING THE NEW AREA.
5252 C .....
5253 76 DO 78 J=1,JMAX
5254 MFGREZ(J)=0
5255 REZAMX(J)=0.
5256 78 REZAI(X(J)=0.
5257 JYMIN=INT(YMIN)+1
5258 IF(JYMIN.GE.JMAX)GO TO 90
5259 DO 84 J=JYMIN,JMAX
5260 K=J*IMAX+1
5261 MFK=MFLAG(K)
5262 MFGREZ(J)=MFK
5263 IF(MFK.LE.0)GO TO 84
5264 REZAMX(J)=AMX(K)/(TAU(IMAX)*DY(J))
5265 IF(MFK.GT.100)GO TO 80
5266 REZAI(X(J)=SSIEN(MFK)
5267 GO TO 84
5268 80 MFK=MFK-100
5269 DO 82 N=1,NMAT
5270 REZXMS(N,J)=XMASS(N,MFK)/AMX(K)
5271 REZSIE(N,J)=C.
5272 IF(SIF(N,MFK).GT.0.)REZSIE(N,J)=SSIEN(N)
5273 REZRHO(N,J)=RHO(N,MFK)
5274 82 REZAI(X(J)=REZAI(X(J)+XMASS(N,MFK)*REZSIE(N,J)
5275 REZAI(X(J)=REZAI(X(J)/AMX(K)
5276 REZRHO(NVOID,J)=RHO(NVOID,MFK)
5277 84 CONTINUE
5278 C .....
5279 C COMPRES THE OLD GRID IN X DIRECTION BY COMBINING PAIRS OF CELLS.
5280 C .....
5281 90 DO 100 J=1,JMAX
5282 DO 100 I=1,NIMAX
5283 K=(J-1)*IMAX+1+1
5284 L=(J-1)*IMAX+2*1
5285 M=L+1
5286 CALL COMPR(L)
5287 100 CONTINUE
5288 C .....
5289 C REDEFINE X, DX, AND TAU
5290 C .....
5291 DO 110 I=1,NIMAX

```

```

5292      110 X(I)=X(2*I)
5293          DO 120 I=NIMAX1,IMAX
5294      120 X(I)=2.*X(I-1)-X(I-2)
5295          WS=0.
5296          DO 130 I=1,IMAX
5297              DX(I)=X(I)-WS
5298              TAU(I)=PI*Y*(X(I)**2-WS*WS)
5299              IF(IGH.EQ.1)TAU(I)=DX(I)
5300      130 WS=X(I)
5301      C * * * * *
5302      C   FILL THE NEW AREA WITH THE PROPERTIES SAVED ABOVE.
5303      C * * * * *
5304          MO=0
5305          CYC=0.
5306          DO 150 J=1,JMAX
5307              DO 150 I=NIMAX1,IMAX
5308                  K=(J-1)*IHAX+I+1
5309                  MFLAG(K)=MFGREZ(J)
5310                  AMX(K)=REZAMX(J)*TAU(I)*DY(J)
5311                  AIX(K)=REZAIK(J)
5312                  U(K)=0.
5313                  V(K)=0.
5314                  IF(MFLAG(K).LT.100)GO TO 150
5315                  CALL NEVMIX
5316                  MFK=MFLAG(K)-100
5317                  DO 140 N=1,NMAT
5318                      RHO(N,MFK)=REZRHO(N,J)
5319                      SIE(N,MFK)=REZSIE(N,J)
5320      140 XMASS(N,MFK)=REZXMS(N,J)*AMX(K)
5321                      RHO(INVOID,MFK)=REZRHO(INVOID,J)
5322      150 CONTINUE
5323      C * * * * *
5324      C   UPDATE THE TOTAL THEORITICAL ENERGY AND ACTIVE GRID COUNTER.
5325      C * * * * *
5326          DO 160 J=1,JMAX
5327              DO 160 I=NIMAX1,IMAX
5328                  K=(J-1)*IHAX+I+1
5329      160 ETH=ETH+AMX(K)*AIX(K)
5330                  II=II/2
5331      C * * * * *
5332      C   GRID HAS BEEN REZONED IN THE X DIRECTION.
5333      C * * * * *
5334      170 IF(JEXTY.EQ.0)RETURN
5335          NJMAX=JMAX/2
5336          IF(2.*NJMAX.EQ.JMAX)GO TO 190
5337      C * * * * *
5338      C   ERROR FOUND. JEXTY=1 BUT JMAX NOT EVEN. GRID NOT REZONED.
5339      C * * * * *
5340      C * * * * *
5341          WRITE(6,180)JMAX
5342      180 FORMAT(5X,'JMAX =',I4,' WHICH IS NOT AN EVEN NUMBER. THE GRID WAS
5343          NOT REZONED IN THE Y DIRECTION.')
5344          RETURN
5345      C * * * * *

```



```

5346 C GRID TO BE REZONED IN Y DIRECTION.
5347 C .....
5348 190 NJMAX1=NJMAX+1
5349 C .....
5350 C REDEFINE THE Y COORDINATES OF THE TRACERS.
5351 C .....
5352 DO 220 N=1,NVOID
5353 NP=NMP(N)
5354 IF(NP.LE.0)GO TO 220
5355 DO 210 M=1,NP
5356 I=INT(TY(N,M))
5357 IF(I.LT.JMAX.AND.IVARDY.EQ.1)GO TO 200
5358 TY(N,M)=.5*TY(N,M)
5359 GO TO 210
5360 200 TYM=TY(N,M)-FLOAT(I)
5361 J=I-2*(I/2)
5362 L=I-J+1
5363 TY(N,M)=.5*FLOAT(I-J)+(TYM*DY(I+1)+FLOAT(J)*DY(I))/(DY(L)+DY(L+1))
5364 210 CONTINUE
5365 220 CONTINUE
5366 NP=(IMAX/2+1)*(JMAX/2+1)
5367 DO 222 M=1,NP
5368 I=INT(YP(M))
5369 IF(I.LT.JMAX.AND.IVARDY.EQ.1)GO TO 221
5370 YP(M)=.5*YP(M)
5371 GO TO 222
5372 221 YPH=YP(M)-FLOAT(I)
5373 J=I-2*(I/2)
5374 L=I-J+1
5375 YP(M)=.5*FLOAT(I-J)+(YPH*DY(I+1)+FLOAT(J)*DY(I))/(DY(L)+DY(L+1))
5376 222 CONTINUE
5377 C .....
5378 C STORE CELL PROPERTIES AT TOP EDGE OF OLD GRID SO THAT THEY
5379 C CAN BE USED FOR FILLING IN THE NEW AREA.
5380 C .....
5381 DO 225 I=1,IMAX
5382 MFGREZ(I)=0
5383 REZAMX(I)=0.
5384 225 REZATX(I)=0.
5385 DO 240 I=1,IMAX
5386 K=(JMAX-1)*IMAX+I+1
5387 MFK=MFLAG(K)
5388 MFGREZ(I)=MFK
5389 IF(MFK.LE.0)GO TO 240
5390 REZAMX(I)=AMX(K)/(TAU(I)*DY(JMAX))
5391 IF(MFK.GT.100)GO TO 230
5392 REZATX(I)=SSIFN(MFK)
5393 GO TO 240
5394 230 MFK=MFK-100
5395 DO 235 N=1,NHAT
5396 REZXMS(N,I)=XMASS(N,MFK)/AMX(K)
5397 REZSIE(N,I)=0.
5398 IF(SIE(N,MFK).GT.0.)REZSIE(N,I)=SSIFN(N)
5399 REZRHO(N,I)=RHO(N,MFK)

```

```

5400      235 REZAIX(I)=REZAIX(I)+XMASS(N,MFK)*REZSIE(N,I)
5401      REZAIX(I)=REZAIX(I)/AMX(K)
5402      REZRHO(NVOID,I)=RHO(NVOID,MFK)
5403      240 CONTINUE
5404      C .....
5405      C COMPRES THE OLD GRID IN Y DIRECTION BY COMBINING PAIRS OF CELLS.
5406      C .....
5407      DO 250 J=1,NJMAX
5408      DO 250 I=1,IMAX
5409      K=(J-1)*IMAX+I+1
5410      L=2*(J-1)*IMAX+I+1
5411      M=L+IMAX
5412      CALL COMPRS(L)
5413      250 CONTINUE
5414      C .....
5415      C REDEFINING Y AND DY.
5416      C .....
5417      DO 260 J=1,NJMAX
5418      260 Y(J)=Y(2*J)
5419      DO 270 J=NJMAX1,JMAX
5420      270 Y(J)=2*Y(J-1)-Y(J-2)
5421      DO 280 J=1,JMAX
5422      280 DY(J)=Y(J)-Y(J-1)
5423      C .....
5424      C FILL THE NEW AREA WITH THE PROPERTIES SAVED ABOVE.
5425      C .....
5426      MU=0
5427      CYC=0.
5428      DO 300 I=1,IMAX
5429      DO 300 J=NJMAX1,JMAX
5430      K=(J-1)*IMAX+I+1
5431      MFLAG(K)=MFGREZ(I)
5432      AMX(K)=REZAMX(I)*TAU(I)*DY(J)
5433      AIX(K)=REZAIX(I)
5434      U(K)=0.
5435      V(K)=0.
5436      IF(MFLAG(K).LT.100)GO TO 300
5437      CALL NEWMIX
5438      MFK=MFLAG(K)-100
5439      DO 290 N=1,NHAT
5440      RHO(N,MFK)=REZRHO(N,I)
5441      SIE(N,MFK)=REZSIE(N,I)
5442      290 XMASS(N,MFK)=REZXMS(N,I)*AMX(K)
5443      RHO(NVOID,MFK)=REZRHO(NVOID,I)
5444      300 CONTINUE
5445      C .....
5446      C UPDATE THE TOTAL THEORITICAL ENERGY AND ACTIVE GRID COUNTER.
5447      C .....
5448      DO 310 I=1,IMAX
5449      DO 310 J=NJMAX1,JMAX
5450      K=(J-1)*IMAX+I+1
5451      310 ETH=ETH+AIX(K)*AIX(K)
5452      J2=J2/2
5453      N6=N6/2

```

```

5454 C . . . . .
5455 C REZONE HAS FINISHED.
5456 C . . . . .
5457 RETURN
5458 END
5459 SUBROUTINE SETUP
5460 C *** SETUP DEFINES CELL QUANTITIES AT TIME=0, ALSO MATERIAL
5461 C TRACERS ARE GENERATED HERE AS WELL AS THE DX,DY,X,Y,
5462 C TAU AND MFLAG ARRAYS.
5463 C
5464 C *** THIS PROBLEM GENERATOR CAN SET UP ONE SPHERE
5465 C AND ANY NUMBER OF CYLINDERS,THE NUMBER OF MATERIAL
5466 C PACKAGES IS LIMITED ONLY BY THE DIMENSIONS OF THE
5467 C MIXED CELL ARRAYS.
5468 C
5469 C *** SETUP WRITES THE CYCLE 0, DUMP ON THE RESTART TAPE.
5470 C
5471 INCLUDE COMDIP
5472 DIMENSION MNAME(40)
5473 DIMENSION XSTPT(6),YSTRT(6),XEND(6),YEND(6),IPKSS(6),INCFS(6)
5474 DIMENSION NT(4),TEMP(4)
5475 DATA(MNAME(K),K=1,40)/6H TU,6HNGSTEN,6H ,6HCOPPER,
5476 1 6H ,6H IRON,6H AL,6HUMINUM,6H PER,6HYLLIUM,
5477 2 6H TI,6HTANIUM,6H ,6HNICKLE,6H MOLY,6HBCENUM,
5478 3 6H T,6HHORIUM,6H ,6H LEAD,6H PO,6HLYMERS,
5479 4 6H G,6HGANITE,6H AN,6HDESITE,6H WE,6HTUFF,
5480 5 6H UR,6HTUFF,6H OIL,6H SHALE,6H DO,6HLOMITE,
5481 6 6H LIM,6HESTONE,6H ,6HHALITE,6H IDE,6HAL GAS/
5482 DATA(MNAME(K),K=41,42)/6H ,6HBASALT/
5483 PIDY=3.1415927
5484 CYCLE=0.0
5485 DT=0.0
5486 NVOID=NMAT+1
5487 KMAX=IMAX*JMAX+1
5488 KMAXA=KMAX+1
5489 JMAXA=JMAX+1
5490 IMAXA=IMAX+1
5491 XMAX=X(JMAX)
5492 YMAX=Y(JMAX)
5493 IIMAX=IMAX*2**NUMREZ
5494 JJMAX=JMAX*2**NUMREZ
5495 IF(IPK(3).LE.-3.) GO TO 258
5496 C *** INITIALIZE PROPERTY ARRAYS
5497 DO 10 K=1,KMAX
5498 AIX(K)=0.
5499 AIX(K)=0.
5500 U(K)=0.
5501 V(K)=0.
5502 MFLAG(K)=0
5503 10 CONTINUE
5504 C
5505 DO 11 L=1,NM>CLS
5506 RHO(L,L)=-1.
5507 11 CONTINUE

```

```

5508 C
5509 DO 16 M=1,NVOID
5510 DO 12 L=1,NMXCLS
5511 FRACFP(M,L)=0.
5512 FRACRT(M,L)=0.
5513 12 CONTINUE
5514 DO 14 L=1,NTPMX
5515 TX(M,L)=0.
5516 TY(M,L)=0.
5517 14 CONTINUE
5518 16 CONTINUE
5519 C      *** COMPUTE Y VALUES FROM DY ARRAY.
5520 IF(IVARDY.EQ.0) GO TO 220
5521 C      *** DY VARIES
5522 J=0
5523 203 READ(5,501) (NT(L),L=1,4), (TEMP(L),L=1,4)
5524 DO 205 L=1,4
5525 NTL=NT(L)
5526 IF(NTL.EQ.999) GO TO 206
5527 DO 204 N=1,NTL
5528 J=J+1
5529 DY(J)=TEMP(L)
5530 204 CONTINUE
5531 205 CONTINUE
5532 GO TO 203
5533 206 IF(J-JMAX)207,209,207
5534 207 WRITE(6,208)
5535 208 FORMAT(50H1SETUP ERROR - DEFINING MORE OR LESS THAN JMAX DYS)
5536 CALL EXIT
5537 C
5538 209 CONTINUE
5539 Y(1)=DY(1)
5540 DO 210 J=2,JMAX
5541 Y(J)=Y(J-1)+DY(J)
5542 210 CONTINUE
5543 GO TO 240
5544 C      *** DY CONSTANT
5545 220 CONTINUE
5546 DO 230 J=1,JMAX
5547 Y(J) = DY*FLOAT(J)
5548 DY(J) = DY
5549 230 CONTINUE
5550 C      *** COMPUTE X VALUES FROM DX ARRAY.
5551 240 IF(IVARDX.EQ.0) GO TO 260
5552 C      *** DX VARIES
5553 I=0
5554 243 READ(5,501) (NT(L),L=1,4), (TEMP(L),L=1,4)
5555 DO 245 L=1,4
5556 NTL=NT(L)
5557 IF(NTL.EQ.999) GO TO 246
5558 DO 244 N=1,NTL
5559 I=I+1
5560 DX(I)=TEMP(L)
5561 244 CONTINUE

```

```

5562 245 CONTINUE
5563 GO TO 243
5564 246 IF(I-IMAX)247,249,247
5565 247 WRITE(6,248)
5566 248 FORMAT(50H)SETUP ERROR - DEFINING MORE OR LESS THAN IMAX DXS)
5567 CALL EXIT
5568 C
5569 249 CONTINUE
5570 X(I)=DX(I)
5571 DO 250 I=2,IMAX
5572 X(I)=X(I-1)+DX(I)
5573 250 CONTINUE
5574 GO TO 280
5575 C *** DX CONSTANT
5576 280 CONTINUE
5577 DO 270 I=1,IMAX
5578 X(I) = DXF*FLOAT(I)
5579 DX(I) = DXF
5580 270 CONTINUE
5581 C
5582 C *** COMPUTE CELL FACE AREA (TAU(I))
5583 280 WS=X(I)*2
5584 TAU(I)=PIDY*WS
5585 IF(IGH.EQ.1)TAU(I)=DX(I)
5586 DO 290 I=2,IMAX
5587 WSA=X(I)*2
5588 TAU(I)=PIDY*(WSA-WS)
5589 IF(IGH.EQ.1)TAU(I)=DX(I)
5590 WS=WSA
5591 290 CONTINUE
5592 FD=1./FLOAT(NTRACR)
5593 MD=1
5594 MBBB=0
5595 MBB=5
5596 C
5597 C *** BEGIN LOOP ON MATERIAL PACKAGES
5598 C
5599 DO 50 NN=1,NMAT
5600 READ(5,450) IGEOM
5601 450 FORMAT(11)
5602 IF(IGEOM=1) 295,296,30
5603 295 WRITE(6,455)
5604 455 FORMAT(43H BAD INPUT - SEE STATEMENT NO. 295 IN SETUP)
5605 CALL EXIT
5606 C *** SETUP A RECTANGLE
5607 296 M=NN
5608 C MAT(M) = CODE MATERIAL NUMBER FOR MTH PACKAGE.
5609 C USED TO IDENTIFY E.S. CONSTANTS.
5610 C IF A PACKAGE IS DIVIDED INTO SPATIALLY
5611 C DISCONNECTED SUBPACKAGES, READ IN THREE CARDS
5612 C FOR EACH SUBPACKAGE WITH A NEGATIVE VALUE
5613 C OF MAT FOR EACH EXCEPT THE LAST
5614 C MRT = I OF RIGHT COLUMN OF PACKAGE.
5615 C NTP = J OF TOP ROW OF PACKAGE

```

```

5616 C          MLF   = I OF LEFT COLUMN OF PACKAGE.
5617 C          MBT   = J OF BOTTOM ROW OF PACKAGE.
5618 C          UR    = RADIAL VELOCITY OF ALL CELLS IN PACKAGE.
5619 C          VA    = AXIAL VELOCITY OF ALL CELLS IN PACKAGE.
5620 C          RHOIN  = INPUT DENSITY FOR PACKAGE
5621 C          SIEN   = SPEC. INT. ENERGY OF ALL CELLS IN PACKAGE.
5622 C
5623          READ(5,500)MAT(M),PACRT,PACTP,PACLF,PACBT,UR,VA,SIEN,RHOIN(M),
5624          1 CZERO(M),STK1(M),STK2(M),STEZ(M),RMU(M),AMD(M))
5625          CALL LOCIJ(PACRT,MRT,.5,0)
5626          CALL LOCIJ(PACTP,MTP,.5,1)
5627          CALL LOCIJ(PACLF,MLF,.5,0)
5628          CALL LOCIJ(PACBT,MBT,.5,1)
5629          IF(MRT.EQ.0)MRT=1
5630          IF(MTP.EQ.0)MTP=1
5631          IF(PACRT.LE.0.)MRT=0
5632          IF(PACTP.LE.0.)MTP=0
5633          MLF=MLF+1
5634          MBT=MBT+1
5635          JPKS=0
5636          IF(MAT(M).LT.0)JPKS=1
5637          MAT(M)=ABS(MAT(M))
5638          UUR(M)=UR
5639          VVA(M)=VA
5640          SSIEN(M)=SIEN
5641          WRITE(6,400)
5642          MA1=MAT(M)*2 - 1
5643          MA2=MA1+1
5644          WRITE(6,410) M,MNAME(MA1),MNAME(MA2),RHOIN(M),SIEN,UR,VA,MLF,MRT,
5645          1          MRT,MTP,CZERO(M),STK1(M),STK2(M),STEZ(M),RMU(M),
5646          2          AMD(M)
5647          L=NMP(M)
5648          LSAVE=L+1
5649          FMLF=FLOAT(MLF-1)
5650          FMBT=FLOAT(MBT-1)
5651          FMRT=FLOAT(MRT)
5652          FMTP=FLOAT(MTP)
5653 C          *** N=NUMBER OF TRACERS ALONG BOTTOM BOUNDARY OF PACKAGE.
5654          IF(MBT.EQ.1)GO TO 18
5655          N=(MRT-MLF+1)*NTRACR+1
5656          IF(MRT.EQ.0)N=(IJMAX-MLF+1)*NTRACR+2
5657          DO 17 KK=1,N
5658          L=L+1
5659          TX(M,L)=FMLF+FLOAT(KK-1)*FD
5660          17 TY(M,L)=FMBT
5661          18 IF(MRT.EQ.0)GO TO 21
5662          IF(MBT.NE.1)GO TO 19
5663          L=L+1
5664          TX(M,L)=FMRT
5665          TY(M,L)=0.
5666          19 N=(MTP-MBT+1)*NTRACR-1
5667          IF(MTP.EQ.0)N=(JJMAX-MBT+1)*NTRACR+1
5668          DO 20 KK=1,N
5669          L=L+1

```

```

5670 TX(M,L)=FMRT
5671 20 TY(M,L)=FMRT+FLOAT(KK)*FD
5672 21 IF(MTP.EQ.0)GO TO 23
5673 N=(MRT-MLF+1)*NTRACR+1
5674 IF(MRT.EQ.0)N=(IIMAX-MLF+1)*NTRACR+2
5675 DO 22 KK=1,N
5676 L=L+1
5677 TX(M,L)=FMLF+FLOAT(N-KK)*FD
5678 22 TY(M,L)=FMTP
5679 23 IF(MLF.EQ.1)GO TO 25
5680 N=(MTP-MBT+1)*NTRACR
5681 IF(MTP.EQ.0)N=(JJMAX-MBT+1)*NTRACR+2
5682 DO 24 KK=1,N
5683 L=L+1
5684 TX(M,L)=FMLF
5685 24 TY(M,L)=FMRT+FLOAT(N-KK)*FD
5686 25 IF(L.LT.NTPMX)GO TO 26
5687 WRITE(6,670)I.
5688 CALL EXIT
5689 26 NMP(M)=L
5690 IF(L.EQ.0)GO TO 28
5691 L=L+1
5692 NMP(M)=L
5693 TX(M,L)=TX(M,LSAVE)
5694 TY(M,L)=TY(M,LSAVE)
5695 28 MMR=MRT
5696 IF(MRT.GT.IMAX.OR.MRT.EQ.0)MMR=IMAX
5697 MMT=MTP
5698 IF(MTP.GT.JMAX.OR.MTP.EQ.0)MMT=JMAX
5699 IF(MLF.GT.IMAX.OR.MBT.GT.JMAX)GO TO 29
5700 DO 27 I=MLF,MMR
5701 DO 27 J=MBT,MMT
5702 K=(J-1)*IMAX+I+1
5703 ANX(K)=RHQIN(M)*TAU(I)*DY(J)
5704 AIX(K)=SIEN
5705 U(K)=UR
5706 V(K)=VA
5707 27 MFLAG(K)=H
5708 IF(MRT.NE.0.AND.MRT.LT.IMAX.AND.MTP.NE.0.AND.MTP.LT.JMAX)GO TO 40
5709 C
5710 C..... PACKAGE EXTENDS BEYOND ORIGINAL GRID.
5711 C..... GENERATE COORDINATES FOR CORNERS OF PACKAGE.
5712 C
5713 29 MB=MBRH
5714 MB=MB+1
5715 MPAC(MB)=5
5716 MPACK(MB)=H
5717 PACX(MB,1)=FMLF
5718 PACX(MB,2)=FMRT+1.E-4
5719 PACY(MB,1)=FMRT
5720 PACY(MB,3)=FMTP
5721 IF(MRT.EQ.0)PACX(MB,2)=FLOAT(IIMAX+1)
5722 IF(MTP.EQ.0)PACY(MB,3)=FLOAT(JJMAX+1)
5723 IF(MLF.GT.1)PACX(MB,1)=PACX(MB,1)+1.E-4

```

```

5724 SF=1.
5725 IF(MBT.LE.JPROJ)SF=-1.
5726 IF(MBT.GT.1)PACY(MB,1)=PACY(MB,1)+SF*1.E-4
5727 SF=1.
5728 IF(MTP.LE.JPROJ)SF=-1.
5729 PACY(MB,3)=PACY(MB,3)+SF*1.E-4
5730 PACX(MB,3)=PACX(MB,2)
5731 PACX(MB,4)=PACX(MB,1)
5732 PACX(MB,5)=PACX(MB,1)
5733 PACY(MB,2)=PACY(MB,1)
5734 PACY(MB,4)=PACY(MB,3)
5735 PACY(MB,5)=PACY(MB,1)
5736 MBRB=MB
5737 GO TO 49
5738
5739 C .30 CONTINUE
5740 C *** SETUP A CIRCLE
5741 C N=NN
5742 C ISPHMX = NUMBER OF CELLS FROM CENTER OF SPHERE TO
5743 C ITS RIGHT EDGE.
5744 C JRADR = NUMBER OF CELLS FROM CENTER OF SPHERE TO
5745 C ITS BOTTOM EDGE.
5746 C JRADA = NUMBER OF CELLS FROM CENTER OF SPHERE TO
5747 C ITS TOP EDGE.
5748 C JSPHTP = NUMBER OF CELLS FROM BOTTOM OF GRID TO TOP
5749 C EDGE OF SPHERE (AT THE AXIS).
5750 C JSPHBT = NUMBER OF CELLS FROM BOTTOM OF GRID TO AND
5751 C INCLUDING BOTTOM EDGE OF SPHERE (AT THE AXIS).
5752 C JCENTR = NUMBER OF THE GRID LINE WHICH COINCIDES WITH
5753 C SPHERE CENTER AT THE AXIS.
5754 C YCENTR = DISTANCE OF SPHERE CENTER FROM GRID BOTTOM =
5755 C IN CENTIMETERS.
5756 C RADIUS = RADIUS OF SPHERE - IN CENTIMETERS.
5757 READ(5,800)MAT(N),UR,VA,SIEN,RHOIN(N),CZERO(N),STK1(N),STK2(N),
5758 1 STEZ(N),RMU(N),AMDM(N)
5759 800 FORMAT(16/4E10.4/6E10.4)
5760 UR=0.
5761 VVA(N)=VA
5762 SSIEN(N)=SIEN
5763 MA1=MAT(N)*2-1
5764 MA2=MA1+1
5765 YCENTR=Y(JCENTR)
5766 WRITE(6,420)
5767 WRITE(6,430) N,aname(MA1),aname(MA2),RHOIN(N),SIEN,UR,VA,RADIUS,
5768 1 YCENTR,CZERO(N),STK1(N),STK2(N),STEZ(N),RMU(N),AMDM(N)
5769 DO 51 ISPHMX=1,IMAX
5770 IF(RADIUS.LE.X(ISPHMX))GO TO 52
5771 51 CONTINUE
5772 52 YTOP=YCENTR+RADIUS
5773 DO 53 JSPHTP=1,JMAX
5774 IF(YTOP.LE.Y(JSPHTP))GO TO 54
5775 53 CONTINUE
5776 54 YBOT=YCENTR-RADIUS
5777 IF(YBOT.LT.0.)YBOT=0.

```



```

5778      DO 55 JSPHBT=1,JMAX
5779      IF(Y90T.LE.Y(JSPHBT))GO TO 56
5780      55 CONTINUE
5781      56 YC2=YCENTR**2
5782      RSQRD = RADIUS**2
5783      DO 43 I=1,ISPHMX
5784      K=(JSPHBT-1)*IMAX+I+1
5785      XL2=(X(I-1))**2
5786      XR2=(X(I))**2
5787      DO 46 J=JSPHBT,JSPHTP
5788      VOLSPH = TAU(I)*DY(J)
5789      IF(J.GT.JCENTR) GO TO 32
5790      WSY=YCENTR-Y(J)
5791      WSY=YCENTR-Y(J-1)
5792      GO TO 34
5793      32 WSY=Y(J-1)-YCENTR
5794      WSY=Y(J)-YCENTR
5795      34 WSB=XL2 + WSY**2
5796      WSA=XR2+WSY**2
5797      IF(WSB.GE.RSQRD) GO TO 46
5798      IF(WSA.LT.RSQRD) GO TO 45
5799      C      *** CELL CUT BY SPHERE BOUNDARY
5800      XL2T=AMAX1(XL2,RSQRD-WSY**2)
5801      XR2B=AMIN1(XR2,RSQRD+WSY**2)
5802      WSLF=XL2+WSY**2
5803      RECTAD=0.
5804      RECTDL=0.
5805      IF(IGM.EQ.1)GO TO 40
5806      IF(WSLF.LT.RSQRD) RECTAD=PI*DY*((RSQRD-WSY**2)-XL2)*WSY
5807      IF(WSY.GT.0.) RECTDL=PI*DY*(XR2B-XL2)*WSY
5808      WSA=SQRT((RSQRD-XR2B)**3)*2./3.
5809      WSB=SQRT((RSQRD-XL2T)**3)*2./3.
5810      GO TO 41
5811      40 IF(WSLF.LT.RSQRD)RECTAD=WSY*(SQRT(RSQRD-WSY**2)-SQRT(XL2T))
5812      IF(WSY.GT.0.)RECTDL=WSY*(SQRT(XR2B)-SQRT(XL2T))
5813      WSA=.5*(SQRT(XL2T)*SQRT(RSQRD-XL2T)+RSQRD*ASIN(SQRT(XL2T)
5814      1/RADIUS))/PI*DY
5815      WSB=.5*(SQRT(XR2B)*SQRT(RSQRD-XR2B)+RSQRD*ASIN(SQRT(XR2B)
5816      1/RADIUS))/PI*DY
5817      41 CONTINUE
5818      VOLSPH=PI*DY*(WSB-WSA)-RECTDL+RECTAD
5819      42 MFLAG(K)=40+100
5820      IF(MO.LE.NMXCLS) GO TO 43
5821      WRITE(6,553) I,J,IGEOM
5822      CALL EXIT
5823      43 M = MO
5824      RHO(I,M)=0.
5825      MO = MO + 1
5826      SIE(N,M) = SIE(N)
5827      V(K) = VA
5828      U(K)=UR
5829      XMASS(N,M) = VOLSPH*RHOIN(N)
5830      RHO(N,M) = RHOIN(N)
5831      IF(N.EQ.1) GO TO 44

```

```

5832 WS=TAU(I)*DY(J)
5833 XMASS(N-1,M)=(WS-VOLSPH)*RHOIN(N-1)
5834 SIE(N-1,M)=AIX(K)
5835 IF(XMASS(N-1,M).LT.0.) SIE(N-1,M)=0.
5836 IF(XMASS(N-1,M).LT.0.) XMASS(N-1,M)=0.
5837 RHO(N-1,M)=RHOIN(N-1)
5838 WS=XMASS(N,M)+XMASS(N-1,M)
5839 AIX(K)=(XMASS(N-1,M)*SIE(N-1,M) + SIEN*XMASS(N,M))/WS
5840 AMX(K)=WS
5841 GO TO 46
5842 C      *** SPHERE BOUNDARY IS A FREE SURFACE WHEN SPHERE IS
5843 C      FIRST PACKAGE GENERATED (N=1).
5844 44 RHO(NVOID,M)=1.0
5845 AMX(K)=XMASS(N,M)
5846 AIX(K)=SIEN
5847 GO TO 46
5848 C      *** CELL NOT CUT BY SPHERE BOUNDARY.
5849 45 AMX(K) = VOLSPH*RHOIN(N)
5850 V(K) = VA
5851 U(K)=UR
5852 AIX(K) = SIEN
5853 MFLAG(K)=N
5854 C      ***END OF LOOP ON ROWS(J)
5855 46 K=K+IMAX
5856 C      ***END OF LOOP ON COLUMNS(I)
5857 48 CONTINUE
5858 C      *** PLACE PARTICLES AROUND SPHERE
5859 JDIAM=JSPHTP+1-JSPHBT
5860 IDIAM=2*ISPHIX
5861 ANGLE=PIDY/2.+ASIN((YCENTR-YBOT)/RADIUS)
5862 NCELLS=.75*FLOAT(MAX0(JDIAM,IDIAM))*ANGLE/PIDY
5863 KMID=NTRACR*NCELLS+1
5864 KEND=2*KMID
5865 NMP(N)=KEND
5866 ANGLE=ANGLE/FLOAT(KEND-1)
5867 DO 57 I=1,KEND
5868 TX(N,I)=RADIUS*SIN(FLOAT(KEND-I)*ANGLE)
5869 C      *** TRACER COORDINATES INITIALLY DEFINED IN CM. UNITS.
5870 57 TY(N,I)=YCENTR+RADIUS*COS(FLOAT(KEND-I)*ANGLE)
5871 TX(N,KEND)=0.
5872 IF(YBOT.LE.0.) TY(N,1)=0.
5873 IF(YBOT.GT.0.) TX(N,1)=0.
5874 NA=N-1
5875 IF(N.EQ.1) NA=NVOID
5876 GO 63 NP=1,KEND
5877 DO 58 I=1,ISPHIX
5878 IF(TX(N,NP).LE.X(I)) GO TO 59
5879 58 CONTINUE
5880 59 DO 61 J=1,JSPHTP
5881 IF(TY(N,NP).LE.Y(J)) GO TO 62
5882 61 CONTINUE
5883 62 NPA=NMP(NA)+KEND+1-NP
5884 C      *** TRACER COORDINATES CONVERTED TO CELL UNITS.
5885 TX(N,NP)=FLOAT(I-1)+(TX(N,NP)-X(I-1))/DX(I)

```

```

5886 TY(N,NP)=FLOAT(J-1)+(TY(N,NP)-Y(J-1))/DY(J)
5887 C      *** NA IS PACKAGE NUMBER OF MATERIAL OUTSIDE THE SPHERE.
5888 C      NPA IS INDEX FOR TRACERS OF PACKAGE NA WHICH WILL
5889 C      BE ORDERED EXACTLY OPPOSITE FROM THE PACKAGE N
5890 C      TRACERS IN THIS REGION.
5891 TX(NA,NPA)=TX(N,NP)
5892 63 TY(NA,NPA)=TY(N,NP)
5893 NP=NMP(N)+1
5894 NMP(N)=NP
5895 TX(N,NP)=TX(N,1)
5896 TY(N,NP)=TY(N,1)
5897 NPA=NMP(NA)+1
5898 NPB=NPA+KEND
5899 NMP(NA)=NPB
5900 TX(NA,NPB)=TX(NA,NPA)
5901 TY(NA,NPB)=TY(NA,NPA)
5902 IF(NMP(NA).LE.NTPMX)GO TO 49
5903 WRITE(6,670)NA
5904 670 FORMAT(' ERROR IN SETUP. NMP(',12,') GT NTPMX. ')
5905 CALL EXIT
5906 C
5907 49 CONTINUE
5908 C
5909 C      *** JPKS=1 WHEN PACKAGE WE HAVE GENERATED IS A SUBPACKAGE
5910 C      AND THERE IS ANOTHER SUBPACKAGE TO BE DEFINED.
5911 IF(JPKS.EQ.1)GO TO 296
5912 C      *** END OF LOOP ON MATERIAL PACKAGES.
5913 50 CONTINUE
5914 C
5915 C      *** DEFINE VOID MATERIAL TRACERS THAT DEFINE STRAIGHT LINE
5916 C      BOUNDARIES OF THE FREE SURFACE. TRACERS SHOULD BE
5917 C      ORDERED SUCH THAT IN TRAVELLING BETWEEN ANY
5918 C      CONSECUTIVE PAIR OF TRACERS THE VOID IS ON THE LEFT.
5919 IF(NSIDES.EQ.0)GO TO 105
5920 L=NMP(NVOID)
5921 LSAVE=L+1
5922 READ 520,(XSTRT(MM),YSTRT(MM),XEND(MM),YEND(MM),MM=1,NSIDES)
5923 520 FORMAT(4E10.4)
5924 DO 71 MM=1,NSIDES
5925 INCFS(MM)=1
5926 71 IPKSS(MM)=0
5927 IPKSS(1)=2
5928 INCFS(1)=0
5929 IF(NSIDES.LE.1)GO TO 73
5930 IPKSS(1)=1
5931 DO 72 MM=2,NSIDES
5932 IF(ABS(XSTRT(MM)-XEND(MM-1)).LE.0..AND.ABS(YSTRT(MM)-YEND(MM-1))
5933 .LE.0.)GO TO 72
5934 IPKSS(MM)=1
5935 IF(XSTRT(MM).GT.0..AND.YSTRT(MM).GT.0.)IPKSS(MM)=0
5936 INCFS(MM)=0
5937 IF(XEND(MM-1).LE.0..OR.YEND(MM-1).LE.0.)IPKSS(MM-1)=2
5938 72 CONTINUE
5939 IF(XEND(NSIDES).LE.0..OR.YEND(NSIDES).LE.0.)IPKSS(NSIDES)=2

```

```

5940      73 CONTINUE
5941      DO 89 MM=1,NSIDES
5942          STRTX=XSTRT(MM)
5943          STRTY=YSTRT(MM)
5944          ENDX=XEND(MM)
5945          ENDY=YEND(MM)
5946          INCF=INCF5(MM)
5947          IPKS=IPK55(MM)
5948          CALL LOCIJ(STRTX,NSX,.5,0)
5949          CALL LOCIJ(STRTY,NSY,.5,1)
5950          CALL LOCIJ(ENDX,NDX,.5,0)
5951          CALL LOCIJ(ENDY,NDY,.5,1)
5952          IF(STRTX.GE.0.)GO TO 152
5953          NSX=0
5954          NSY=NSY+1
5955          NDX=NDX+1
5956          NDY=NDY+1
5957          GO TO 166
5958      152 IF(STRTY.GE.0.)GO TO 154
5959          NSY=0
5960          NDY=NDY+1
5961          GO TO 166
5962      154 IF(ENDX.GE.0.)GO TO 156
5963          NSX=NSX+1
5964          NDX=0
5965          GO TO 166
5966      156 IF(ENDY.GE.0.)GO TO 158
5967          NSX=NSX+1
5968          NDX=NDX+1
5969          NSY=NSY+1
5970          NDY=0
5971          GO TO 166
5972      158 IF(ABS(STRTX-ENDX).LE.0)GO TO 162
5973          IF(NSY.EQ.0)NSY=1
5974          IF(NDY.EQ.0)NDY=1
5975          IF(STRTX.GT.ENDX)GO TO 160
5976          IF(NDX.EQ.0)NDX=1
5977          NSX=NSX+1
5978          GO TO 166
5979      160 IF(NSX.EQ.0)NSX=1
5980          NSY=NSY+1
5981          NDY=NDY+1
5982          NDX=NDX+1
5983          GO TO 166
5984      162 IF(NSX.EQ.0)NSX=1
5985          IF(NDX.EQ.0)NDX=1
5986          IF(STRTY.GT.ENDY)GO TO 164
5987          IF(NDY.EQ.0)NDY=1
5988          NSX=NSX+1
5989          NDX=NDX+1
5990          NSY=NSY+1
5991          GO TO 166
5992      164 IF(NSY.EQ.0)NSY=1
5993          NDY=NDY+1

```

```

5994      166 IF (NDX.EQ.NSX.OR.NDY.EQ.NSY)GO TO 170
5995      WRITE(6,168)MM,NSX,NSY,NDX,NDY
5996      168 FORMAT(22H ERROR IN FREE SURFACE ,5X,5I6)
5997      CALL EXIT
5998      170 CONTINUE
5999      IF (IPKS.EQ.1)LSAVE=L+1
6000      IF (ABS(STRTX-ENDX).LE.0.)GO TO 84
6001      IF (NSX.EQ.0)GO TO 82
6002      IF (NSX.GT.NDX.AND.NDX.GT.0)GO TO 82
6003      C
6004      C..... ADD TRACERS FROM LEFT TO RIGHT.
6005      C
6006      N=(NDX-NSX+1)*NTRACR+1-INCF
6007      IF (NDX.EQ.0)N=(IIMAX-NSX+1)*NTRACR+2-INCF
6008      DO 81 KK=1,N
6009      L=L+1
6010      TX(NVOID,L)=FLOAT(NSX-1)+FLOAT(KK-1+INCF)*FD
6011      81 TY(NVOID,L)=FLOAT(NSY)
6012      GO TO 88
6013      C
6014      C..... ADD TRACERS FROM RIGHT TO LEFT.
6015      C
6016      82 N=(NSX-NDX+1)*NTRACR+1-INCF
6017      IF (NSX.EQ.0)N=(IIMAX-NDX+1)*NTRACR+2
6018      DO 83 KK=1,N
6019      L=L+1
6020      TX(NVOID,L)=FLOAT(NDX-1)+FLOAT(N-KK)*FD
6021      83 TY(NVOID,L)=FLOAT(NSY-1)
6022      GO TO 88
6023      84 IF (NSY.EQ.0)GO TO 86
6024      IF (NSY.GT.NCY.AND.NDY.GT.0)GO TO 86
6025      C
6026      C..... ADD TRACERS FROM BOTTOM TO TOP.
6027      C
6028      N=(NDY-NSY+1)*NTRACR+1-INCF
6029      IF (NDY.EQ.0)N=(JJMAX-NSY+1)*NTRACR+2-INCF
6030      DO 85 KK=1,N
6031      L=L+1
6032      TX(NVOID,L)=FLOAT(NSX-1)
6033      85 TY(NVOID,L)=FLOAT(NSY-1)+FLOAT(KK-1+INCF)*FD
6034      GO TO 88
6035      C
6036      C..... ADD TRACERS FROM TOP TO BOTTOM.
6037      C
6038      86 N=(NSY-NDY+1)*NTRACR+1-INCF
6039      IF (NSY.EQ.0)N=(JJMAX-NDY+1)*NTRACR+2
6040      DO 87 KK=1,N
6041      L=L+1
6042      TX(NVOID,L)=FLOAT(NSX)
6043      87 TY(NVOID,L)=FLOAT(NDY-1)+FLOAT(N-KK)*FD
6044      88 IF (IPKS.NE.2)GO TO 89
6045      L=L+1
6046      TX(NVOID,L)=TX(NVOID,LSAVE)
6047      TY(NVOID,L)=TY(NVOID,LSAVE)

```

```

6048      LSAVE=L+1
6049      89 CONTINUE
6050      NMP(NVOID)=L
6051      IF(L.LE.NTPMX)GO TO 105
6052      WRITE(6,530)
6053      530 FORMAT(' ERROR IN SETUP. NMP(NVOID) GT NTPMX')
6054      CALL EXIT
6055      105 CONTINUE
6056      C      *** DIDDLE TRACERS SLIGHTLY SO PACKAGE BOUNDARIES
6057      C      DONT FALL EXACTLY ON GRID LINES AND MIXED
6058      C      CELLS CAN BE MORE EASILY DEFINED.
6059      C
6060      DO 120 N=1,NVOID
6061      NN=NMP(N)
6062      IF(NN.EQ.0) GO TO 120
6063      DO 110 L=1,NN
6064      IF(TX(N,L).GT.0..AND.TX(N,L).LT.FLOAT(IIMAX))TX(N,L)=TX(N,L)+1.E-4
6065      SF=1.
6066      IF(INT(TY(N,L)).LE.JPROJ) SF=-1.
6067      IF(TY(N,L).GT.0..AND.TY(N,L).LT.FLOAT(JJMAX))
6068      TY(N,L)=TY(N,L)+1.E-4*SF
6069      110 CONTINUE
6070      120 CONTINUE
6071      C
6072      C      *** FLAG CELLS BORDERING PACKAGES
6073      DO 201 M=1,NVOID
6074      NP=NMP(M)
6075      IF(NP.EQ.0)GO TO 201
6076      DO 200 L=1,NP
6077      I=INT(TX(M,L))+1
6078      J=INT(TY(M,L))+1
6079      IF(I.GT.IMAX..OR.J.GT.JMAX)GO TO 200
6080      K=(J-1)*IMAX+I+1
6081      MFK=MFLAG(K)
6082      IF(MFK.GT.100) MA=MFK-100
6083      CALL PROPRT
6084      200 CONTINUE
6085      201 CONTINUE
6086      C      *** COMPUTE TOTAL ENERGY IN GRID (ETH)
6087      250 ETH=0.
6088      DO 300 K=1,KMAX
6089      ETH=ETH + AMX(K)*(1.5*(U(K)**2 + V(K)**2) + AIX(K))
6090      300 CONTINUE
6091      C
6092      IF(NTCC.EQ.0) GO TO 308
6093      C      *** SETUP PASSIVE CELL-CENTERED TRACERS WHEN NTCC.GT.0.
6094      NP=0
6095      DO 303 I=1,IMAX,2
6096      DO 303 J=1,JMAX,2
6097      K=(J-1)*IMAX+I+1
6098      IF(AMX(K).LE.0.) GO TO 303
6099      M=MFLAG(K)
6100      IF(M.LT.100) GO TO 302
6101      M=M-100

```

```

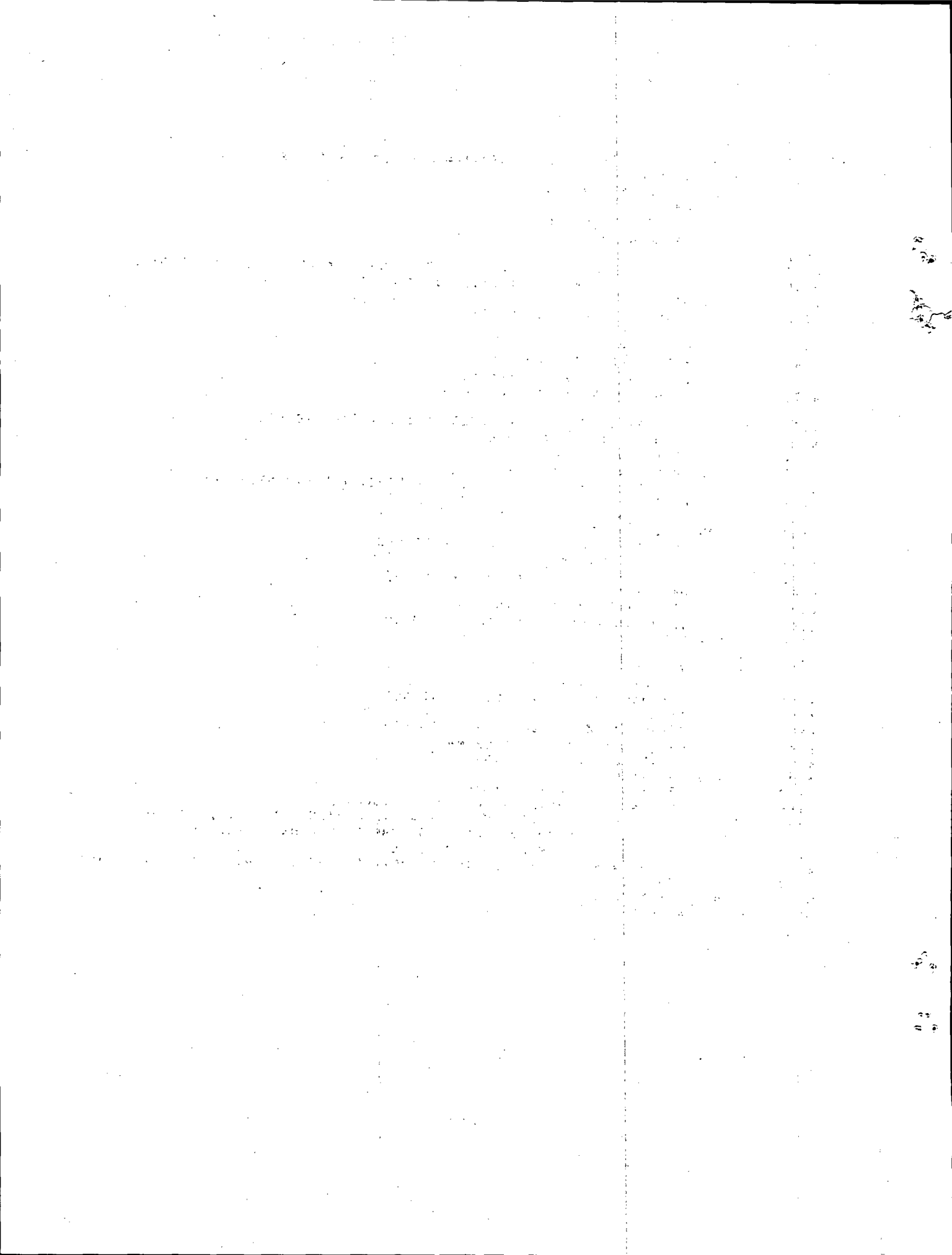
6102      PV=0.
6103      DO 301 N=1,MMAT
6104      IF(RHO(N,M).LE.0.) GO TO 301
6105      PV=PV+XMASS(N,M)/RHO(N,M)
6106 301 CONTINUE
6107      IF(PV.LT.(TAU(I)*DY(J)*.5)) GO TO 303
6108 302 NP=NP+1
6109      XP(NP) = FLOAT(I)-.5
6110      YP(NP) = FLOAT(J)-.5
6111 303 CONTINUE
6112      NTCC=NP
6113      WRITE(6,307) (L,XP(L),YP(L),L=1,NTCC)
6114 307 FORMAT(1X,'CELL CENTERED TRACKS'/(15,2F10.2,15,2F10.2,15,2F10.2,
6115      1 15,2F10.2,15,2F10.2))
6116 C
6117 308 CONTINUE
6118 C      *** WRITE TAPE DUMP FOR CYCLE 0.
6119      WS=555.0
6120      REWIND KUNITW
6121      WRITE (KUNITW)  WS, CYCLE
6122      WRITE (KUNITW)  (7(I),I=1,150)
6123      WRITE (KUNITW)  (U(I),V(I),AMX(I),AIX(I),P(I), MFLAG(I),I=1,KMAX)
6124      WRITE (KUNITW)  (STRSZ(I), STRSRP(I), STRSRZ(I), I=1,KMAX)
6125      WRITE (KUNITW)  (X(I), DX(I), TAU(I), I=1,IMAX)
6126      WRITE (KUNITW)  (Y(I), DY(I), I=1,JMAX)
6127      WRITE(KUNITW) (CZERG(M), STK1(M), STK2(M), STEZ(M), RMU(M),
6128      1 ANDN(M), RHOIN(M), SSIEN(M), OUR(M), YVA(M), MAT(M), PLW(M),
6129      2 M=1,MMAT)
6130      WRITE(KUNITW) (MPAC(I),MPACK(I),I=1,MBBB)
6131      WRITE(KUNITW) ((PACX(I,L),PACY(I,L),I=1,MBBB),L=1,MBB)
6132      WRITE(KUNITW) ((XMASS(M,L), RHO(M,L), SJE(M,L), SAMPY(M,L),
6133      1 SAMP(M,L), M=1,MMAT), RHO(INVOID,L),L=1,MMXCLS)
6134      DO 350 N=1,NVOID
6135      NP=NMP(N)
6136      WRITE(KUNITW) NP, (TX(N,L),TY(N,L),L=1,NP)
6137 350 CONTINUE
6138      NP=(IMAX/2+1)*(JMAX/2+1)
6139      WRITE(KUNITW) NP, (XP(L),YP(L),L=1,NP)
6140      WS=666.0
6141      WRITE(KUNITW) WS, WS
6142      RETURN
6143 400 FORMAT(// 40X,18HINITIAL CONDITIONS//12X,8HMATERIAL,
6144      16X,7HDENSITY,8X,3HSIE,11X,1HU,12X,1HV,7X,7HLEFT(1),3X,8HRIGHT(1),
6145      23X, 9HROTIG(J),2X,6HTOP(J)/)
6146 410 FORMAT(9H PACKAGE ,11,2A6,1X,1P4E13.5,3X,13,7X,13,7X,13,7X,13//
6147      1 29X, 2HY0,11X,2HY1,11X,2HY2,11X,2HE0,10X,3HRMU,10X,4HAMDM/
6148      2 23X, 1P6E13.5//)
6149 420 FORMAT(//40X,18HINITIAL CONDITIONS//12X,8HMATERIAL,6X,7HDENSITY,
6150      18X,3HSIE,11X,1HU,12X,1HV,7X,6HRADIUS,4X,6HCENTER/)
6151 430 FORMAT(9H PACKAGE ,11,2A6,1X, 4E13.5,2F10.3//
6152      1 29X,2HY0,11X,2HY1,11X,2HY2,11X,2HE0,10X,3HRMU,10X,4HAMDM/
6153      2 23X,1P6E13.5//)
6154 500 FORMAT(16,4E10.4/4E10.4/6E10.4)
6155 501 FORMAT(14,4,4,3E10.4)

```

```

6156      550 FORMAT(67H EXCEEDED STORAGE REQUESTED FOR MIXED CELLS(NMXCLS), 11.
6157      1J,IGEOM) = ,316)
6158      600 FORMAT(3I6,2FA.3)
6159      END
6160      SUBROUTINE STRNG
6161      INCLUDE COMDIM
6162      C
6163      C      *** STRNG IS CALLED FROM PH3 TO COMPUTE THE YIELD STRENGTH
6164      C      OF THE MATERIAL IN CELL K.
6165      400 STRENG=0.
6166      IF(MFK.GT.100) GO TO 405
6167      C      *** PURE CELL
6168      N=MAT(MFK)
6169      SOLID=AMDM(MFK)*RHOZ(N)
6170      DENSTY=AMX(K)/(TAU(I)*DY(J))
6171      IF(DENSTY.LT.SOLID) GO TO 440
6172      WSA=1.
6173      IF(AIX(K).LT.0.OR.ARS(STEZ(MFK)).LE.0) GO TO 402
6174      WSA=1.-AIX(K)/STEZ(MFK)
6175      IF(WSA.LT.0.) GO TO 440
6176      402 WSB=DENSTY/RHOZ(N)-1.
6177      STRENG=(CZERO(MFK)+WSB*(STK1(MFK)+STK2(MFK)+WSB))*WSA
6178      GO TO 440
6179      C      *** MIXED CELL
6180      405 MKF=MFK-100
6181      IF(RHO(NVOID,MKF).GT.0.) GO TO 440
6182      DO 407 MM=1,NMAT
6183      IF(XMASS(MM,MKF).LE.0.) GO TO 407
6184      M=MAT(MM)
6185      SOLID=AMDM(MM)*RHOZ(N)
6186      IF(RHO(MM,MKF).LT.SOLID) GO TO 440
6187      407 CONTINUE
6188      C
6189      VCELL=TAU(I)*DY(J)
6190      DO 409 MM=1,NMAT
6191      IF(XMASS(MM,MKF).LE.0.) GO TO 409
6192      WSA =1.
6193      IF(ABS(STEZ(MM)).LE.0.) GO TO 408
6194      WSA=1.-SIE(MM,MKF)/STEZ(MM)
6195      IF(WSA.LE.0.) GO TO 409
6196      408 N=MAT(MM)
6197      WSB=RHO(MM,MKF)/RHOZ(N)-1.
6198      VOLM = XMASS(MM,MKF)/(RHO(MM,MKF)*VCELL)
6199      C      *** STRENGTH OF MATERIAL IN A MIXED CELL IS A VOLUME
6200      C      WEIGHTED AVERAGE OF THE STRENGTHS OF ALL THE
6201      C      MATERIALS IN THE CELL.
6202      STRENG = STRENG + VOLM*(CZERO(MM)+WSB*(STK1(MM)+STK2(MM)+WSB))*WSA
6203      409 CONTINUE
6204      440 WS=STRENG
6205      RETURN
6206      END

```

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Commander Defense Documentation Center ATTN: TIPCR Cameron Station Alexandria, Virginia 22314	1	Commanding General U. S. Army Materiel Command ATTN: AMCRD-TE Washington, D. C. 20315
1	Director of Defense Research and Engineering (OSD) ATTN: Asst Dir/Tac Ord & Msl Sys Washington, D. C. 20301	1	Commanding General U. S. Army Materiel Command ATTN: AMCRD-TP Washington, D. C. 20315
1	Director Advanced Research Projects Agency ATTN: Dr. S. Lukasik Washington, D. C. 20301	1	Commanding General U. S. Army Aviation Systems Command ATTN: AMSAV-E 12th & Spruce Streets St. Louis, Missouri 63166
1	Director Weapons Systems Evaluation Group Washington, D. C. 20305	1	Commanding General U. S. Army Electronics Command ATTN: AMSEL-DL Fort Monmouth, New Jersey 07703
1	Director Institute for Defense Analysis 400 Army-Navy Drive Arlington, Virginia 22202	5	Commanding General U. S. Army Missile Command ATTN: AMSMI-R AMSMI-RSD AMCPM-PE AMCPM-LC AMCPM-SE Redstone Arsenal, Alabama 35809
1	Headquarters Defense Atomic Support Agency ATTN: STBS Washington, D. C. 20305	1	Commanding General U. S. Army Tank-Automotive Command ATTN: AMSTA-RHFL Warren, Michigan 48090
1	Commanding General U. S. Army Materiel Command ATTN: AMCDL Washington, D. C. 20315	2	Commanding Officer U. S. Army Mobility Equipment Research & Development Center ATTN: Tech Docu Ctr, Bldg 315 AMSME-RZT Fort Belvoir, Virginia 22060
1	Commanding General U. S. Army Materiel Command ATTN: AMCRD, Dr. J.V.R. Kaufman Washington, D. C. 20315		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Commanding General U. S. Army Munitions Command ATTN: AMSMU-RE Dover, New Jersey 07801	1	Commanding Officer U. S. Army Materials and Mechanics Research Center ATTN: AMXMR-ATL Watertown, Massachusetts 02172
1	Commanding Officer U. S. Army Edgewood Arsenal ATTN: SMUEA-W Edgewood Arsenal, Maryland 21010	1	Commanding General U. S. Army Natick Laboratories ATTN: AMXRE, Dr.D. Sieling Natick, Massachusetts 01762
1	Commanding Officer U. S. Army Frankford Arsenal ATTN: SMUFA-C2500 Philadelphia, Pennsylvania 19137	1	Office of Vice Chief of Staff ATTN: CSAVCS-W-TIS Department of the Army Washington, D. C. 20310
2	Commanding Officer U. S. Army Picatinny Arsenal ATTN: SMUPA-V SMUPA-VK Dover, New Jersey 07801	1	Chief of Research & Development ATTN: CRDMS, Dr. W. B. Taylor Department of the Army Washington, D. C. 20310
1	Commanding General U. S. Army Weapons Command ATTN: AMSWE-RE Rock Island, Illinois 61202	1	Director U. S. Army Research Office ATTN: CRDPES 3045 Columbia Pike Arlington, Virginia 22204
1	Director U. S. Army Advanced Materiel Concepts Agency 2461 Eisenhower Avenue Alexandria, Virginia 22314	1	Director U. S. Army Ballistic Missile Defense Agency ATTN: Dr. P. J. Friel 1320 Wilson Boulevard Arlington, Virginia 22209
1	Director U. S. Army Aeronautical Research Laboratory Moffett Naval Air Station California 94035	3	Commander U. S. Naval Air Systems Command ATTN: AIR-604 Washington, D. C. 20360
1	Commanding Officer U. S. Army Harry Diamond Laboratories ATTN: AMXDO-TD/002 Washington, D. C. 20438	3	Commander U. S. Naval Ordnance Systems Command ATTN: ORD-9132 Washington, D. C. 20360

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	Commander U. S. Naval Ordnance Laboratory ATTN: Mr. N. L. Coleburn Code 730, Lib Silver Spring, Maryland 20910	1	Director Lawrence Radiation Laboratory ATTN: Dr. M. Wilkins P. O. Box 808 Livermore, California 94550
1	Director U. S. Naval Research Laboratory ATTN: Code 6240, Mr. Atkins Washington, D. C. 20390	1	Director Los Alamos Scientific Laboratory P. O. Box 1663 Los Alamos, New Mexico 87544
1	Commander U. S. Naval Weapons Center ATTN: Code 753 China Lake, California 93555	1	Director National Aeronautics & Space Administration Langley Research Center Langley Station Hampton, Virginia 23365
1	AFATL (DLRW, Mr. Dittrich) Eglin AFB Florida 32542	1	Director National Aeronautics & Space Administration Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135
1	Air Force Weapons Laboratory Technical Library Kirtland AFB New Mexico 87117	1	Aerospace Corporation P. O. Box 95085 Los Angeles, California 90045
1	AFML (MAMD) Wright-Patterson AFB Ohio 45433	1	AVCO Corporation Research & Advance Development Division 201 Lowell Street Wilmington, Massachusetts 01887
1	Director U. S. Bureau of Mines ATTN: Mr. R. Watson 4800 Forbes Street Pittsburgh, Pennsylvania 15213	2	General Motors Technical Center ATTN: Dr. C. Maiden Mr. W. Isbell 12 Mile & Mound Roads Warren, Michigan 48090
1	Director Environmental Science Services Administration ATTN: Code R, Dr. J. Rinehart U. S. Department of Commerce Boulder, Colorado 80302		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	General Research Corporation ATTN: Dr. B. Cummings P. O. Box 3587 Santa Barbara, California 93105	1	Philco-Ford Corporation Aeronutronic Division ATTN: Dr. M. Boyer Ford Road Newport Beach, California 92663
1	Grumman Aircraft Corporation Bethpage New York 11714	1	Physics International Corporation ATTN: Dr. C. Godfrey 2700 Merced Street San Leandro, California 94577
1	Hughes Aircraft Company Research & Development Laboratories Cantincla and Teale Streets Culver City, California 90232	1	Sandia Corporation P. O. Box 5800 Albuquerque, New Mexico 87115
1	Kaman Nuclear A Division of Kaman Aircraft Corporation ATTN: Dr. D. Sachs Garden of the Gods Road Colorado Springs, Colorado 80907	3	Systems, Science and Software ATTN: Dr. J. Walsh Dr. J. Diencs Dr. T. Riney P. O. Box 1620 La Jolla, California 92037
1	Lockheed Aircraft Corporation Military Operations Reserve Division 2555 Hollywood Way Burbank, California 91503	1	TRW Systems Group ATTN: Mr. F. Pieper One Space Park Redondo Beach, California 90278
1	Lockheed Palo Alto Research Laboratories 3251 Hanover Street Palo Alto, California 94304	1	Brown University Division of Engineering ATTN: Prof. P. Symonds Providence, Rhode Island 02912
1	Martin-Marietta Corporation Research Institute Advanced Studies ATTN: Dr. R. Sedney 1450 S. Rolling Road Baltimore, Maryland 21212	2	Drexel Institute of Technology Wave Propagation Research Center ATTN: Prof. P. Chou 32rd & Chestnut Streets Philadelphia, Pennsylvania 19104

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	The Johns Hopkins University Mechanics Department ATTN: Prof. J. Bell Prof. R. Green 34th & Charles Street Baltimore, Maryland 21218	1	University of Notre Dame Dept of Met Eng & Mat Science ATTN: Prof. N. Fiore Notre Dame, Indiana 46556
1	Massachusetts Institute of Technology Aeroelastic & Structures Laboratory 77 Massachusetts Avenue Cambridge, Massachusetts 02139	1	University of Texas Department of Engineering Mechanics ATTN: Prof. H. Calvit Austin, Texas 78712
1	Lincoln Laboratory (MIT) 244 Wood Street Lexington, Massachusetts 02173	1	Washington State University Department of Physics ATTN: Prof. G. Duvall Pullman, Washington, 99163
			<u>Aberdeen Proving Ground</u>
1	Pennsylvania State University Dept of Engineering Mechanics ATTN: Prof. N. Davids University Park, Pennsylvania 16802		Ch, Tech Lib Marine Corps Ln Ofc CDC Ln Ofc
1	Southwest Research Institute ATTN: Dr. W. Baker 8500 Culebra Road San Antonio, Texas 78228		
2	Stanford Research Institute Poulter Laboratories ATTN: Dr. M. Evans Dr. M. Cowperthwaite Menlo Park, California 94025		
1	University of Illinois Department of Aeronautical Engineering ATTN: Prof. R. Strehlow Urbana, Illinois 61803		

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Systems, Science and Software P. O. Box 1620 La Jolla, California 92123		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE HELP, A Multi-Material Eulerian Program for Compressible Fluid and Elastic-Plastic Flows in Two Space Dimensions and Time, Volume II: Fortran Listing of HELP			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Contractor Report			
5. AUTHOR(S) (First name, middle initial, last name) L. J. Hageman J. M. Walsh			
6. REPORT DATE May 1971		7a. TOTAL NO. OF PAGES 123	7b. NO. OF REFS 0
8a. CONTRACT OR GRANT NO. DAAG 07-68-C-0931		9a. ORIGINATOR'S REPORT NUMBER(S) 3SR-350 Volume II	
b. PROJECT NO. RDT&E 1T650212D620		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) BRL Contract Report No. 39	
c.			
d.			
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY USA Aberdeen Research & Development Center Ballistic Research Laboratories Aberdeen Proving Ground, Maryland 21005	
13. ABSTRACT Volume II is a complete FORTRAN IV listing of the HELP program, the background description of which is contained in the preceding Volume I.			

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Numerical Finite-Difference Code Eulerian Computer Code Compressible Flow Elastic-Plastic Flow Hydrodynamic Code						