

# Gulf General Atomic

Incorporated

P.O. Box 608, San Diego, California 92112

①

AD 678568

GAMD-8497

Addendum

Category A

## AN EULERIAN METHOD FOR CALCULATING STRENGTH DEPENDENT DEFORMATION

DDO

DEC 3 1968

Work done by:

J. K. Dienes  
L. J. Hageman

Report written by:

J. K. Dienes  
L. J. Hageman

This document, which was prepared primarily for internal use at Gulf General Atomic, may contain preliminary or incomplete data. It is informal and is subject to revision or correction; it does not, therefore, represent a final report.

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

Advanced Research Projects Agency  
ARPA Order No. 71-62  
Ballistic Research Laboratories  
Contract No. DA-04495-AMC-1481(X)  
GGA Project 6003

February 2, 1968

Reproduced by the  
CLEARINGHOUSE  
for Federal Scientific & Technical  
Information Springfield Va. 22151

**BEST  
AVAILABLE COPY**

ADDENDUM TO REPORT ON THE  
OIL-RPM COMPUTER CODE

This addendum describes a number of improvements to the version of OIL-RPM documented in GAMD-8497. In EDIT and MAP the changes represent primarily a streamlining of the program. The small change to CDT ensures stability in a more rational manner. The requirements for enlarging the grid are also described, and a flow chart of the equation of state subroutine (ES) is included.

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

## REVISED VERSIONS OF CDT, EDIT AND MAP

The improved version of CDT computes the time step using the speed  $|u| + C$  of a sound wave relative to the grid, where  $u$  denotes the maximum of the radial and axial velocities. This allows the user to set the stability fraction (STAB) to .6, or perhaps larger, and retains stability; where in the earlier version it was recommended that STAB be set to .4. This version of CDT can be directly substituted for the one listed previously.

However, to use the improved versions of EDIT and MAP the following statements must be added to subroutine INPUT after statement #50:

```
FRSTD = 1.  
IF(KUNIT.LE.0) KUNIT = 7  
IF(MSYMBL.LE.0) MSYMBL = 26
```

The prominent features of the new EDIT routine are:

- (1) It prints the angular distributions of the mass, momenta, and energy.
- (2) It uses a variable (KUNIT) instead of a constant to specify the unit number of the dump tape. (The user can start a second dump tape by setting KUNIT = 9 in the restart deck. The program will read from unit 7 but will write out on unit 9.)
- (3) An end of file mark is written at the end of each restart dump. (The next dump writes over this end of file mark.)
- (4) The general flow and organization of EDIT have been simplified making the routine easier to modify.

The new MAP routine is more efficiently coded, and uses only one variable, MSYMBL, to determine the degree of resolution for all the maps.

SUBROUTINE CDT

```

.....
DIMENSION AMX(2502),AIX(2502),U(2502) ,V(2502) ,P(2502) ,
1 X(52) ,XX(54) ,TAU(52) ,JPM(52) ,
2 Y(102) ,YY(104) ,FLEFT(102), YAMC(102), SIGC(102),
3 GAMC(102),
4 PK(15), Z(150) ,
5 XP(26,51),YP(26,51),
6 PL(204) ,UL(204) ,PR(204) ,
7 RSN(52), RST(52),
8 CMXP(5) ,CMYP(5) ,IJ(5) ,JK(5) ,
9 DX(52) ,DDX(54) ,DY(102) ,DDY(104) ,
$ SNB(52) ,STB(52) ,UK(52,3) ,VK(52,3) ,RHO(52,3)

```

\*\*\* DIMENSIONED ARRAYS  
\*\*\* Z-BLOCK IS SAVED ON TAPE.

```

COMMON Z
COMMON PK
COMMON YY, XX
COMMON DDX, DDY
COMMON AMX, AIX, U, V, P
COMMON TAU, JPM
COMMON UL, PL
COMMON XP, YP, CMXP, CMYP

```

\*\*\* NON-DIMENSIONED VARIABLES

```

COMMON AID ,AMMV ,AMMY ,AMPY ,AMUR ,AMUT ,AMVR ,
1AMVT ,DELEB ,DELER ,DELET ,DELM ,DTODX ,DXYMIN,EAMMP ,EAMPY ,
2E ,ERDUMP,I ,I3 ,IWS ,J ,K ,KA ,KB ,
3LL ,MD ,ME ,MZT ,NERR ,NK ,NPRINT,
4NR ,NRZ ,NULLE ,PIDTS ,SIEMIN,SNR ,SNT ,STR ,SOLID ,
5SUM ,TESTRH,TWOPI ,URR ,WS ,WSA ,WSB ,WSC ,WFLAGF,
6WFLAGL,WFLAGP

```

\*\*\* THE FOLLOWING EQUIVALENCES MAKE AVAILABLE  
X(0), Y(0), DX(0), DY(0)

```

EQUIVALENCE (XX(2), X(1)), (YY(2), Y(1))
EQUIVALENCE (DDX(2), DX(1)), (DDY(2), DY(1))

```

\*\*\* SPECIAL EQUIVALENCES FOR PH2 ONLY

```

EQUIVALENCE (UL,FLEFT), (UL(103),YAMC),
1 (PL,GAMC,PR), (PL(103),SIGC)

```

\*\*\* SPECIAL EQUIVALENCES FOR PH3 ONLY

```

EQUIVALENCE (UL,RSN),
1 (PL,RST), (P,UK),
2 (P(157),VK), (P(313),SNB),
3 (P(365),STB), (P(417),RHO)

```

\*\*\* SPECIAL EQUIVALENCES FOR EDIT

```

EQUIVALENCE (PR(1), IJ), (PR(6), JK)

```

\*\*\* Z-STORAGE EQUIVALENCES

```

EQUIVALENCE (Z( 1),PROB ),(Z( 2),CYCLE ),

```



```

10 TRIAL=0.
   SRATIO=10.**10
C     **WSC WILL BE MAXIMUM U OR V
   WSC=0.
   DO 60 I=1,I1
   K=I+1
   DO 60 J=1,I2
   RHOW=AMX(K)/(TAU(I)*DY(J))
   CALL ES
C     *** IF DENSITY OF CELL K IS LESS THAN RHOMIN, IT'S
C     VELOCITY OR SOUND SPEED IS NOT USED IN DETERMINING DT.
   IF (RHOW.LT.RHOMIN) GO TO 60
   IF (ABS(P(K)).LT.PMIN) P(K)=0.
   IF (CNAUT.GT,0.) GO TO 20
C
C     ***CALCULATE SOUND SPEED FOR POLYTOPIC GAS WITH
C     ***GAMMA EQUAL TO ESA+1.
   WSD=SQRT(GAMMA*ABS(P(K))/RHOW)
   GO TO 40
C
C     ***CHECK FOR NEGATIVE PRESSURE.
20 IF (P(K).GT.0.) GO TO 30
C     *** NEGATIVE PRESSURES NOT ALLOWED ALONG GRID BOUNDARY
C     AND NOT ALLOWED ANYWHERE UNTIL ACTIVE GRID REACHES
C     JSTR(INPUT PARAMETER FOR TURNING ON STRENGTH
C     CALCULATIONS).
   IF ((IMAX.NE.1.AND.I.EQ.IMAX).OR.J.EQ.JMAX.OR.I2.LT.JSTR) P(K)=0.
C
C     ***PRESSURE IS NEGATIVE OR ZERO
   WSD=CNAUT
   GO TO 40
C
C     ***PRESSURE IS POSITIVE.
30 WS=CNAUT+BBAR*SQRT(P(K))
   WSA=SQRT(GAMMA*P(K)/RHOW)
   WSD=AMAX1(WS,WSA)
C     *** WSB IS MAXIMUM OF RADIAL AND AXIAL VELOCITY OF CELL K.
40 WSB=AMAX1(ABS(U(K)),ABS(V(K)))
C     *** WSC STORES MAXIMUM VELOCITY OF CELLS USED TO DETERMINE
C     DT. PRINTED AS MAXUV.
   WSC=AMAX1(WSC,WSB)
C     ***WSD IS SOUND SPEED OF CELL K.
   WS=WSD+WSB
   IF (WS.LE.TRIAL) GO TO 50
C     *** TRIAL STORES MAXIMUM VELOCITY PLUS SOUNLD SPEED OF
C     CELL USED TO DETERMINE DT.
   TRIAL=WS
C     *** CMAX IS SOUND SPEED OF CELL CONTROLLING DT.
   CMAX=WSD
50 IF (WS.LE.0.) GO TO 60
   DXYMIN=AMIN1(DX(I),DY(J))
   RATIO=DXYMIN/WS
   IF (RATIO.GT.SRATIO) GO TO 60
C     *** I AND J OF CELL CONTROLLING DT STORED IN N10 AND N11
C     FOR PRINTOUT.
   N10=I
   N11=J

```

```

C          *** SRATIO IS SMALLEST VALUE CALCULATED FOR RATIO.
SRATIO=RATIO
C
C          ***END OF I, J LOOP
60      K=K+IMAX
C          *** IF TRIAL.LE.0. THERE IS PROBABLY AN ERROR IN THE INPUT
C          PARAMETERS FOR THE INITIAL VELOCITY, ENERGY OR DENSITY
C          OF THE PACKAGES.
65      IF (TRIAL.LE.0.) GO TO 180
C          *** IF FINAL.EQ.0.USE STAB FOR VALUE OF STABILITY FRACTION
C          IF FINAL.GT.0.USE A GEOMETRIC PROGRESSION WITH STAB
C          AS THE INITIAL VALUE AND FINAL AS THE FINAL VALUE.
IF (FINAL.LT.0.) GO TO 70
STAB=2.*STAB
STAB=AMIN1(STAB,FINAL)
70      DT=STAB*SRATIO
IF (NC.GT.0) GO TO 80
IF (DTMIN.GT.0..OR.DTMIN.LT.0.) GO TO 80
DTMIN = (10.**2)*DT
C
C          *** IS CONTROL-CELL ISOLATED
80      K=(N11-1)*IMAX+N10+1
WS=0.
IF (N10.GT.1) WS=AMX(K-1)
IF (N10.LT.IMAX) WS=AMX(K+1)+WS
IF (N11.GT.1) WS=AMX(K-IMAX)+WS
IF (N11.LT.JMAX) WS=AMX(K+IMAX)+WS
IF (WS.GT.0.) GO TO 100
C          *** ISOLATED, SO DESTROY IT.
WS=(AIX(K)+(U(K)**2+V(K)**2)*.5)*AMX(K)
EVAPM=EVAPM+AMX(K)
EVAPEN=EVAPEN+WS
ETH=ETH-WS
EVAPMU=EVAPMU+AMX(K)*U(K)
EVAPMV=EVAPMV+AMX(K)*V(K)
WRITE (6,300) N10,N11
AMX(K)=0.
AIX(K)=0.
P(K)=0.
U(K)=0.
V(K)=0.
C          *** RECALCULATE DT.
GO TO 10
C          *** INCREMENT TIME AND CYCLE.
100     T=T+DTNA
IF (T.LT.0.) GO TO 170
NC=NC+1
CYCLE=NC
C          *** RESET NPRINT. NPRINT=1 ON PRINT CYCLES.
NPRINT=0
C          *** DEFINE VELOCITY AND ENERGY CUTOFFS USED IN MAP AND PH2.
UMIN=TRIAL*ROEPS
SIEMIN=UMIN**2
PMIN=RHOZ*CNAUT*UMIN
IF (PMIN.LT.ROEPS) PMIN=UMIN*RHOZ*TRIAL
WRITE (6,310) N10,N11,T,DT,DTMIN,CMAX,U(K),V(K),DX(N10),DY(N11),UM

```



```

1IN,PMIN
C      *** AFTER STAB.GE.FINAL CHECK ON SIZE OF DT. DTMIN CAN
C      BE DEFINED IN INPUT DECK.
104    IF (STAB.LT.FINAL) GO TO 106
106    IF (DT.LE.DTMIN) GO TO 160
CONTINUE
DTNA=DT
C      *** TESTRH = .2*RHOZ
C      THE PRESSURE OF COLD,FREE SURFACE CELLS IS REDUCED BY A
C      FACTOR,F, WHICH ACCOUNTS FOR THE EFFECT OF FREE SURFACE
C      LOCATION ON THE PRESSURE GRADIENT. F IS THE DENSITY OF
C      THE LOWEST DENSITY ADJACENT CELL DIVIDED BY THE NORMAL
C      DENSITY,OR F IS TESTRH - WHICHEVER IS SMALLEST
WT=TESTRH
DO 150 I=1,I1
K=I+1
DO 150 J=1,I2
RHOH=AMX(K)/(DY(J)*TAU(I))
WTB=WT
IF (AIX(K).GE.ESESQ) GO TO 150
IF (RHOW.LT.SOLID) GO TO 150
IF (I.EQ.IMAX) GO TO 110
WTA=AMX(K+1)/(DY(J)*TAU(I+1))
IF (WTA.LT.WT) WTB=WTA
110    IF (I.EQ.1) GO TO 120
WTA=AMX(K-1)/(DY(J)*TAU(I-1))
IF (WTA.LT.WTB) WTB=WTA
120    IF (J.EQ.JMAX) GO TO 130
KA=K+IMAX
WTA=AMX(KA)/(DY(J+1)*TAU(I))
IF (WTA.LT.WTB) WTB=WTA
130    IF (J.EQ.1) GO TO 140
KB=K-IMAX
WTA=AMX(KB)/(DY(J-1)*TAU(I))
IF (WTA.LT.WTB) WTB=WTA
140    IF (WTB.LT.WT) P(K)=P(K)*WTB/RHOZ
150    K=K+IMAX
GO TO 200
C
C      *** DT TOO SMALL
160    NK=104
GO TO 190
C      *** T IS NEGATIVE
170    NK=102
GO TO 190
C      *** DT WILL BE NEGATIVE OR ZERO.
180    NK=65
GO TO 190
190    NR=3
CALL ERROR
C
C      ***FIND THE MAXIMUM PRESSURE ON EACH COLUMN AND
C      ***STORE ITS CELL NUMBER AS JPM. THIS WILL BE USED
C      ***IN DETERMINING THE REGION IN WHICH PHASE 3 IS
C      ***USED. WSA WILL BE A RUNNING MAXIMUM OF THE
C      ***PRESSURE IN THE GRID.
200    WSA=-1.E30

```

```

C      DO 270 I=1,I1
      *** WS WILL BE LOCAL MAXIMUM OF COLUMN I.
      WS=-1.E30
      K=(I2-1)*IMAX+I+1
      JP=I2
      JINTL=1
C      *** START AT TOP OF COLUMN AND LOOK FOR PRESSURE PEAK.
210   DO 220 J=JINTL,I2
      IF (P(K).LT.WS) GO TO 230
      WS=P(K)
C      *** JP IS J-INDEX OF CELL WITH PEAK PRESSURE.
      JP=JP-1
220   K=K-IMAX
C      *** IF YOU FALL THROUGH, THEN THERE WAS NO MAXIMUM IN THIS
C      COLUMN
      GO TO 260
C
C      3328      *** COME HERE IF PRESSURE HAS PASSED A LOCAL MAXIMUM
C      *** PTEMP IS PEAK PRESSURE OF COLUMN I.
C
230   PTEMP=P(K+IMAX)
      IF (PTEMP.LT.WSA) GO TO 240
C      *** WSA WILL BE PEAK PRESSURE IN ACTIVE GRID (ABSOLUTE
C      MAXIMUM).
      WSA=PTEMP
      GO TO 250
C
C      3329      *** PTEMP IS LOCAL MAXIMUM BUT IS LESS THAN ABSOLUTE
C      MAXIMUM
240   IF (PTEMP.GT.0.3*WSA) GO TO 250
C
C      *** THIS LOCAL MAXIMUM IS NOT BIG ENOUGH TO USE FOR JPM
C
      JINTL=J+1
C
      JP=JP-1
C
C      *** WE MAY HAVE REACHED BOTTOM OF COLUMN
      IF (JINTL.GE.I2) GO TO 260
C      *** CONTINUE DOWN COLUMN SEARCHING FOR SUFFICIENTLY LARGE
C      LOCAL MAXIMUM.
      WS=P(K)
      K=K-IMAX
      GO TO 210
C      *** IF POSITION OF PEAK PRESSURE IN COLUMN I DOES NOT
C      ADVANCE FROM ONE CYCLE TO THE NEXT, DO NOT CHANGE
C      VALUE OF JPM.
250   JP=JP+1
      IF (JP.LE.JPM(I)) GO TO 270
      JPM(I)=JP
C
C      ***IF JPM IS ZERO THE SHOCK HAS NEVER REACHED THIS
C      ***LOCATION. IF IT IS NONZERO THE SHOCK HAS PASSED
C      ***AND WE MUST CONTINUE TO INCREASE I UNTIL THE
C      ***RIGHT BOUNDARY OF THE SHOCK IS REACHED.
260   IF (JPM(I).LE.0) GO TO 280
C      *** END OF I LOOP.

```

```

270 CONTINUE
C     *** IF PEAK PRESSURE OF COLUMN I HAS GONE BELOW A THIRD
C     THE GRID MAXIMUM, AND IF JPM(I)=0. FROM THE PREVIOUS
C     CYCLE, WE HAVE REACHED THE RIGHT EDGE OF THE SHOCK.
280 CONTINUE
C     *** JPM(I) MUST BE MONOTONIC DECREASING
      K=I1-1
      DO 290 IWS=1,K
      I=I1-IWS
290  IF (JPM(I).LT.JPM(I+1)) JPM(I)=JPM(I+1)
      RETURN
C
300  FORMAT (/4H CDT,I3,I4,2X,31HISOLATED CONTROL CELL DESTROYED/)
310  FORMAT (/4H CDT,I3,I4,4H  T=,1PE12.6,5H  DY=,1PE9.3,8H  DTMIN=,1PE
19.3,4H  C=,1PE9.3,4H  U=,1PE9.3,4H  V=,1PE9.3,5H  DX=,1PE9.3,5H  D
2Y=,1PE9.3/13X,5HUMIN=,1PE9.3,7H  PMIN=1PE9.3)
      END

```

SUBROUTINE EDIT

```

.....
DIMENSION AMX(2502),AIX(2502),U(2502) ,V(2502) ,P(2502) ,
1 X(52) ,XX(54) ,TAU(52) ,JPM(52) ,
2 Y(102) ,YY(104) ,FLEFT(102),YAMC(102),SIGC(102),
3 GAMC(102),
4 PK(15), Z(150) ,
5 XP(26,51),YP(26,51),
6 PL(204) ,UL(204) ,PR(204) ,
7 RSN(52), RST(52),
8 CMXP(5) ,CMYP(5) ,IJ(5) ,JK(5) ,
9 DX(52) ,DDX(54) ,DY(102) ,DDY(104) ,
$ SNB(52) ,STB(52) ,UK(52,3) ,VK(52,3) ,RH0(52,3)

```

\*\*\* B L A N K C O M M O N  
DIMENSIONED VARIABLES

```

COMMON Z
COMMON PK
COMMON YY, XX
COMMON DDX, DDY
COMMON AMX, AIX, U, V, P
COMMON TAU, JPM
COMMON UL, PL,
COMMON XP, YP, CMXP, CMYP

```

NON-DIMENSIONED VARIABLES

```

COMMON AID ,AMMV ,AMMY ,AMPY ,AMUR ,AMUT ,AMVR ,
1AMVT ,DELEB ,DELER ,DELET ,DELM ,DTODX ,DXYMIN,EAMMP ,EAMPY ,
2E ,ERDUMP,I ,IWS ,I3 ,J ,K ,KA ,
3KB ,LL ,MD ,ME ,MZT ,NERR ,NK ,NPRINT,
4NR ,NRZ ,NULLE ,PIDTS ,SIEMIN,SNR ,SNT ,STR ,SOLID ,
5SUM ,TESTRH,TWOPI ,URR ,WS ,WSA ,WSB ,WSC ,WFLAGF,
6WFLAGL,WFLAGP
COMMON LAST

```

\*\*\* THE FOLLOWING EQUIVALENCES DEFINE STORAGE FOR  
X(0), Y(0), DX(0), DY(0)

```

EQUIVALENCE (XX(2), X(1)) ,(YY(2), Y(1)),
1 (DDX(2),DX(1)) ,(DDY(2),DY(1))

```

\*\*\* SPECIAL EQUIVALENCES FOR PH2 ONLY

```

EQUIVALENCE (UL,FLEFT), (UL(103),YAMC),
1 (PL,GAMC,PR), (PL(103),SIGC)

```

\*\*\* SPECIAL EQUIVALENCES FOR PH3 ONLY

```

EQUIVALENCE (UL,RSN),
1 (PL,RST), (P,UK),
2 (P(157),VK), (P(313),SNB),
3 (P(365),STB), (P(417),RH0)

```

\*\*\* SPECIAL EQUIVALENCES FOR EDIT

```

EQUIVALENCE (PR(1), IJ), (PR(6), JK), (UL(103),CRAD)

```

\*\*\* Z-STORAGE EQUIVALENCES

C

EQUIVALENCE (Z( 1),PROB ),(Z( 2),CYCLE ),  
 1(Z( 3),DT ),(Z( 4),NUMSP ),(Z( 5),NFRELP),(Z( 6),NDUMP7),  
 2(Z( 7),ICSTOP),(Z( 8),PIDY ),(Z( 9),TOPMU ),(Z( 10),RTMU ),  
 3(Z( 11),STK1 ),(Z( 12),NUMREZ),(Z( 13),ETH ),(Z( 14),UN14 ),  
 4(Z( 15),RHINIT),(Z( 16),PROJI ),(Z( 17),KUNIT ),(Z( 18),XMAX ),  
 5(Z( 19),NZ ),(Z( 20),NREZ ),(Z( 21),AMDM ),(Z( 22),UVMAX ),  
 6(Z( 23),UN23 ),(Z( 24),DMIN ),(Z( 25),JSTR ),(Z( 26),DTNA ),  
 7(Z( 27),CVIS ),(Z( 28),STK2 ),(Z( 29),STEZ ),(Z( 30),NC ),  
 8(Z( 31),UN31 ),(Z( 32),NRC ),(Z( 33),IMAX ),(Z( 34),IMAXA ),  
 9(Z( 35),JMAX ),(Z( 36),JMAXA ),(Z( 37),KMAX ),(Z( 38),KMAXA )

EQUIVALENCE  
 1(Z( 39),BOTM ),(Z( 40),BOTMV ),(Z( 41),NUMSPT),(Z( 42),CZERO ),  
 2(Z( 43),NUMSCA),(Z( 44),PRLIM ),(Z( 45),PRDELTA),(Z( 46),PRFACT)

EQUIVALENCE  
 1(Z( 47),I1 ),(Z( 48),I2 ),(Z( 49),IPCYCL),(Z( 50),TSTOP ),  
 2(Z( 51),RHOFIL),(Z( 52),TARGV ),(Z( 53),N3 ),(Z( 54),IVARDY),  
 3(Z( 55),VT ),(Z( 56),N6 ),(Z( 57),RTM ),(Z( 58),RTMV ),  
 4(Z( 59),UN59 ),(Z( 60),N10 ),(Z( 61),N11 ),(Z( 62),GAMMA ),  
 5(Z( 63),TOPM ),(Z( 64),BOTMU ),(Z( 65),SN ),(Z( 66),TOPMV ),  
 6(Z( 67),PRYBOT),(Z( 68),PRYTOP),(Z( 69),PRXRT ),(Z( 70),CYCPH3),  
 7(Z( 71),REZFCT),(Z( 72),TARGI ),(Z( 73),PROJU ),(Z( 74),BBOUND),  
 8(Z( 75),EVAP ),(Z( 76),ECK ),(Z( 77),NECYCL),(Z( 78),II ),  
 9(Z( 79),JJ ),(Z( 80),NMP ),(Z( 81),Y2 ),(Z( 82),EZPH1 )

EQUIVALENCE  
 1(Z( 83),IVARDX),(Z( 84),T ),(Z( 85),NMPMAX),(Z( 86),PMIN ),  
 2(Z( 87),INTER),(Z( 88),TAYBOT),(Z( 89),TAYTOP),(Z( 90),UN90 ),  
 3(Z( 91),MC ),(Z( 92),MR ),(Z( 93),MZ ),(Z( 94),MB )

EQUIVALENCE  
 1(Z( 95),REZ ),(Z( 96),NODUMP),(Z( 97),UN97 ),(Z( 98),UN98 ),  
 2(Z( 99),UN99 ),(Z(100),EVAPM ),(Z(101),EVAPEN),(Z(102),EVAPMU),  
 3(Z(103),EVAPMV),(Z(104),EZPH2 ),(Z(105),SNL ),(Z(106),STL ),  
 4(Z(107),TAXRT ),(Z(108),MSYMBL),(Z(109),UN109 ),(Z(110),ROEPS ),  
 5(Z(111),RHINI ),(Z(112),VINI ),(Z(113),FINAL ),(Z(114),FRSTD ),  
 6(Z(115),RHOZ ),(Z(116),ESA ),(Z(117),ESEZ ),(Z(118),ESB ),  
 7(Z(119),ESCAPA),(Z(120),ESESP ),(Z(121),ESESQ ),(Z(122),ESES ),  
 8(Z(123),ESALPH),(Z(124),ESBETA),(Z(125),ESCAPB),(Z(126),UN126 ),  
 9(Z(127),SS1 ),(Z(128),SS2 ),(Z(129),UMIN ),(Z(130),SS4 )

EQUIVALENCE  
 1(Z(131),PRTIME),(Z(132),EOR ),(Z(133),EOT ),(Z(134),EOB ),  
 2(Z(135),EMOR ),(Z(136),DXF ),(Z(137),DYF ),(Z(138),RHOMIN),  
 3(Z(139),STAB),(Z(140),XIENRG),(Z(141),XKENRG),(Z(142),XTENRG),  
 4(Z(143),STT ),(Z(144),DTMIN ),(Z(145),TRNSFC),(Z(146),EMOT ),  
 5(Z(147),JPROJ ),(Z(148),CNAUT ),(Z(149),BBAR ),(Z(150),EMOB )

C  
C  
C  
C  
C  
C  
C  
C

.....  
 END OF COMMON  
 .....

DIMENSION PROPI(50), AMK(15), GK(15), TAB(15), CRAD(52)  
 \*\*\* SPECIAL EQUIV. FOR EDIT  
 EQUIVALENCE (UL,PROPI), (PL(51), AMK), (PL(66),GK),  
 1 (PL(81),TAB)  
 EQUIVALENCE (PR(1),TIETAR),(PR(2),TKETAR),(PR(3),TETAR ),  
 1 (PR(4),TARMAS),(PR(5),TARMV ),(PR(6),TARMVP),

```

2          (PR(7),RAMOMA),(PR(8),PRAMO), (PR(9),TIEPRO),
3          (PR(10),TKEPRO),(PR(11),TEPRO),(PR(12),PRMAS),
4          (PR(13),PRMV ),(PR(14),PRMVP),(PR(15),RAMOMB),
5          (PR(16),PRAMOB)

```

```

C
C
C
C
C
C
C
C

```

.....

```

*** ENERGY SUM (ESUM) AND RELATIVE ERROR IN SUM (RELERR)
    COMPUTED. ECK IS LARGEST ERROR COMPUTED AND ON PRINT
    CYCLES IS PRINTED AND COMPARED TO DMIN, MAXIMUM
    ALLOWABLE ERROR.

```

```

ESUM=0.
DO 10 K=2,KMAX
10  ESUM=ESUM+AMX(K)*(.5*(U(K)**2+V(K)**2)+AIX(K))
    RELERR=(ESUM-ETH)/ETH
    IF (ABS(RELERR).LT.ABS(ECK)) GO TO 20
    ECK=RELERR
    NECYCL=NC
20  CONTINUE

```

```

C

```

```

*** NERR = 1 WHEN ERROR CALLS EDIT.
IF (NERR.EQ.1) GO TO 150

```

```

C

```

.....

```

*** NPRINT = 1 WHEN EDIT IS CALLED TO DO AN INTERMEDIATE
    PRINT. SKIP TESTS ON TIME TO STOP, PRINT, REZONE,ETC.
    WHICH ALREADY HAVE BEEN DONE FOR THIS CYCLE.

```

```

C
C
C
C
C
C

```

```

IF (NPRINT.EQ.1) GO TO 190
*** I3=1 SIGNALS A SHORT PRINT

```

```

C

```

```

I3=1
*** IF THIS IS FIRST CYCLE OF RUN, WFLAGF=1.
IF (WFLAGF.GT.0.) GO TO 120
*** IS THIS THE TIME OR CYCLE TO STOP EXECUTION
IF (ICSTOP.LE.NC.AND.ICSTOP.GT.0) GO TO 140
IF (T*(1.+ROEPS).GE.TSTOP.AND.TSTOP.GT.0.) GO TO 140
*** SHOULD THE GRID BE REZONED
IF ((REZ.NE.0..AND.REZFCT.NE.0..AND.NUMREZ.GT.0).OR.SS4.NE.0.) GO
1TO 145

```

```

C

```

```

ASSIGN 414 TO LOCA
ASSIGN 110 TO LOCB

```

```

C

```

.....

```

*** ARE WE PRINTING ON TIME OR CYCLE INTERVALS

```

```

C

```

```

IF (PRDEL.NE.0.) GO TO 50
IF (IPCYCL.NE.0) GO TO 100
GO TO 420

```

```

40

```

```

*** PRINTING ON TIME. IS IT TIME TO PRINT
IF (T*(1.+ROEPS).GE.PRTIME) GO TO 70
*** NO. BUT WILL NEXT CYCLE BYPASS THE PRINT TIME
IF (PRTIME.GE.T+DT) GO TO 60
DT=PRTIME-T

```

```

C
50
C

```

```

DTNA=DT
60 GO TO LOCA, (414,412)
C   *** YES, IT IS TIME TO PRINT. NPRINT=1 FLAGS THIS AS A
C   PRINT CYCLE.
70 NPRINT=1
C   *** AVOID TRUNCATION
T=PRTIME
C   *** IS IT TIME TO RESCALE PRINT INTERVAL
IF (T*(1.+ROEPS).LT.PRLIM.OR.NUMSCA.LE.0) GO TO 80
C   *** CHANGE PRINT INTERVAL AND THE TIME FOR THE NEXT
C   RESCALING.
PRDEL=PRDEL*PRFACT
PRLIM=PRLIM*PRFACT
NUMSCA=NUMSCA-1
C   *** DEFINE TIME FOR NEXT PRINT.
80 PRTIME=T+PRDEL
IWS=(PRTIME+.5*PRDEL)/PRDEL
WS=IWS
PRTIME=WS*PRDEL
C   *** WILL WE BYPASS TIME TO PRINT
IF (PRTIME.GE.T+DT) GO TO 90
C   *** YES, ADJUST DT
DT=PRTIME-T
DTNA=DT
90 GO TO LOCB, (110,412)
C   *** PRINTING ON CYCLES. IS THIS A PRINT CYCLE
100 IF (MOD(NC,PCYCL).NE.0) GO TO LOCA, (414,412)
C   *** YES. NPRINT = 1 FLAGS THIS AS A PRINT CYCLE.
NPRINT=1
C   *** IS THIS THE CYCLE TO RESCALE PRINT INTERVAL
IF (NC.LT.PRLIM.OR.NUMSCA.LE.0) GO TO LOCB, (110,414)
C   *** YES. MULTIPLY NUMBER OF CYCLES BETWEEN PRINTS BY PRFACT
PCYCL=INT(PRFACT)*PCYCL
PRLIM=PRFACT*PRLIM
NUMSCA=NUMSCA-1
GO TO LOCB, (110,412)
C   *** TEST FOR SHORT OR LONG PRINT
C   *** NUMSP COUNTS NUMBER OF SHORT PRINTS SINCE LAST LONG
C   PRINT. NUMSPT COUNTS NUMBER OF PRINTS SINCE LAST
C   TAPE DUMP.
110 NUMSP=NUMSP+1
NUMSPT=NUMSPT+1
IF (NUMSP.NE.NFRELP) GO TO 130
NUMSP=0
C   *** I3=I1 SIGNALS A LONG PRINT
120 I3=I1
C   *** PRINT OF RESTART CYCLE WILL BE SHORT IF PK(3).LT.-1.
IF (PK(3).LT.-1..AND.WFLAGF.GT.0.) I3=1
130 IF (NUMSPT.NE.NDUMP7) GO TO 190
GO TO 150
C   *** SET WFLAGL=1. TO SAY THIS IS LAST CYCLE OF RUN
140 WFLAGL=1.
145 I3=I1
NPRINT=1
NUMSP=0
C
C

```

```

C          *** TAPE DUMP
C
C          .....
150  NUMSPT=0
    IF (NFRELP.EQ.NDUMP7) NUMSP=0
    IF (NODUMP.NE.0) GO TO 170
    BACKSPACE KUNIT
    IF (FRSTD.GT.0.) GO TO 155
    BACKSPACE KUNIT
155  WS=555.0
    WRITE (KUNIT) WS,CYCLE,N3
    WRITE (KUNIT) (Z(L),L=1,MZT)
    WRITE (KUNIT) (U(K),V(K),AMX(K),AIX(K),P(K),K=1,KMAX)
    WRITE (KUNIT) X(0),(X(K),TAU(K),JPM(K),K=1,IMAX)
    WRITE (KUNIT) (Y(K),K=0,JMAX)
C          *** ARE TRACER POINTS BEING GENERATED
    IF (Y2.GT.(-1.)) GO TO 160
    WRITE (KUNIT) ((XP(I,J),YP(I,J),I=1,II),J=1,JJ)
160  WRITE (KUNIT) (DX(I),I=1,IMAX)
    WRITE (KUNIT) (DY(I),I=1,JMAX)
    WS=666.0
    WRITE (KUNIT) WS,WS,WS
    FRSTD = 0.
    END FILE KUNIT
170  CONTINUE
C          *** ERDUMP=1. WHEN ERROR CALLS EDIT FOR A TAPE DUMP ONLY
    IF (ERDUMP.GT.0.) RETURN
C          .....
C          *** COMPUTE AND PRINT ENERGY, MASS AND MOMENTUM TOTALS.
C          .....
C          *** INITIALIZE PR ARRAY, TEMPORARY STORAGE FOR ENERGY,MASS
C          AND MOMENTUM TOTALS PRINTED OUT.
190  DO 200 I=1,16
200  PR(I)=0.
C
C  RAMOMA=RADIAL MOMENTUM ABOVE JPROJ
C  RAMOMA=RADIAL MOMENTUM BELOW JPROJ
C  PRAMOA=POSITIVE RADIAL MOMENTUM ABOVE JPROJ
C  PRAMOB=POSITIVE RADIAL MOMENTUM BELOW JPROJ
C
    IF (JPROJ.NE.0) GO TO 205
    N=2
    GO TO 220
205  N=IMAX*JPROJ+1
    DO 210 K=2,N
    WS=AMX(K)
    PRMAS=PRMAS+WS
    TIEPRO=TIEPRO+WS*AIX(K)
    TKEPRO=TKEPRO+.5*WS*(U(K)**2+V(K)**2)
    WSA=WS*V(K)
    PRMV=PRMV+WSA
    IF (WSA.GT.0.) PRMVP=PRMVP+WSA
    RAMOMB=RAMOMB+AMX(K)*U(K)
    IF (U(K).GT.0.) PRAMOB=PRAMOB+AMX(K)*U(K)
210  CONTINUE

```



```

N=N+1
220 DO 230 K=N,KMAX
    WS=AMX(K)
    TARMAS=TARMAS+WS
    TIETAR=TIETAR+WS*AIX(K)
    TKETAR=TKETAR+.5*WS*(U(K)**2+V(K)**2)
    WSA=WS*V(K)
    TARMV=TARMV+WSA
    IF (WSA.GT.0.) TARMVP=TARMVP+WSA
    RAMOMA=RAMOMA+AMX(K)*U(K)
    IF (U(K).GT.0.) PRAMO=PRAMO+AMX(K)*U(K)
230 CONTINUE
    TETAR=TIETAR+TKETAR
    TEPRO=TIETAR+TKEPRO
    DO 240 J=1,8
    PR(J+16)=PR(J)+PR(J+8)
240 CONTINUE
    IF (IMAX.GT.1) GO TO 260
C
C     *** IF DOING A 1-D PROBLEM DIVIDE TOTALS BY NZ WHERE
C     NZ=4**(NUMBER OF TIMES THE GRID HAS BEEN REZONED.)
C
PROPI(1)=ETH/NZ
PROPI(2)=ECK/NZ
PROPI(4)=EZPH1/NZ
PROPI(5)=EZPH2/NZ
PROPI(6)=BBOUND/NZ
DO 250 J=1,24
250 PROPI(J+6)=PR(J)/NZ
PROPI(31)=BOTM/NZ
PROPI(32)=RTM/NZ
PROPI(33)=TOPM/NZ
PROPI(34)=EVAPM/NZ
PROPI(35)=EMOB/NZ
PROPI(36)=EMOR/NZ
PROPI(37)=EMOT/NZ
PROPI(38)=EVAPEN/NZ
PROPI(39)=BOTMU/NZ
PROPI(40)=RTMU/NZ
PROPI(41)=TOPMU/NZ
PROPI(42)=EVAPMU/NZ
PROPI(43)=BOTMV/NZ
PROPI(44)=RTMV/NZ
PROPI(45)=TOPMV/NZ
PROPI(46)=EVAPMV/NZ
PROPI(47)=EOB/NZ
PROPI(48)=EOR/NZ
PROPI(49)=EOT/NZ
WRITE (6,520) PROB,T,NC,PROPI(1),PROPI(2),NECYCL,(PROPI(J),J=4,6)
WRITE (6,530) (PROPI(J),J=7,49)
GO TO 270
260 WRITE (6,520) PROB,T,NC,ETH,ECK,NECYCL,EZPH1,EZPH2,BBOUND
WRITE (6,530) ((PR(J),J=1,24),BOTM,RTM,TOPM,EVAPM,EMOB,EMOR,EMOT,E
1VAPEN,BOTMU,RTMU,TOPMU,EVAPMU,BOTMV,RTMV,TOPMV,EVAPMV,EOB,EOR,EOT)
270 WRITE (6,570) (JPM(I),I=1,I1)
C     *** ENERGY TOTALS STORED FOR LATER USE IN TRACER POINT
C     PLOTS.

```

```
XIENRG=PR(17)
XKENRG=PR(18)
XTENRG=PR(19)
```

```
C . . . . .
NKT = 12
```

```
C *** TABS ARE TANGENT ALPHAS
TAB(1) = 0.02
TAB(2) = 0.04
TAB(3) = 0.06
TAB(4) = 0.08
TAB(5) = 0.10
TAB(6) = 0.15
TAB(7) = 0.20
TAB(8) = 0.25
TAB(9) = 0.30
TAB(10)= 0.40
TAB(11)= 0.50
TAB(12)= 1.00
```

```
C
NK1 = NKT+2
DO 275 I=1,NK1
AMK(I) = 0.
PK(I) = 0.
275 QK(I) = 0.
```

```
C
DO 280 K=2,KMAXA
IF (AMX(K))440,280,276
276 I=NK1
IF (V(K))279,279,277
277 WSA = ABS(U(K))/V(K)
```

```
C *** SEARCH FOR TAN ANGLE MADE BY VELOCITY VECTOR OF CELL.
```

```
C
DO 278 I=1,NKT
IF (TAB(I)-WSA) 278,279,279
278 CONTINUE
I=NK+1
```

```
C *** SUM MASS BETWEEN ANGLES.
279 AMK(I) = AMK(I) + AMX(K)
C *** SUM RADIAL MOMENTA BETWEEN ANGLES.
PK(I) = PK(I) + U(K)*AMX(K)
C *** SUM AXIAL MOMENTA BETWEEN ANGLES.
QK(I) = QK(I) + V(K)*AMX(K)
```

```
280 CONTINUE
WRITE(6,605)
WRITE(6,610)(AMK(I),I=1,NK1)
WRITE(6,615)(PK(I), I=1,NK1)
WRITE(6,620)(QK(I), I=1,NK1)
IF (NUMSPT.EQ.0) WRITE(6,540) NC
```

```
C *** ARE TRACER POINTS BEING GENERATED
IF (Y2.GT.(-1.)) GO TO 305
```

```
C . . . . .
C *** PRINT TRACER POINT COORDINATES IN CM.
C . . . . .
```

```

WRITE (6,580)
N=0
DO 300 J=1,JJ
DO 300 I=1,II
IF (XP(I,J).LE.0..AND.YP(I,J).LE.0.) GO TO 300
IP=INT(XP(I,J))
JP=INT(YP(I,J))
KK=JP*IMAX+IP+2
IF (AMX(KK).GT.0.) GO TO 290
XP(I,J)=0.
YP(I,J)=0.
GO TO 300
290 N=N+1
CMXP(N)=X(IP)+DX(IP+1)*(XP(I,J)-INT(XP(I,J)))
CMYP(N)=Y(JP)+DY(JP+1)*(YP(I,J)-INT(YP(I,J)))
C
C      *** IJ, JK = THE I AND J OF THE CELL THE TRACER POINT
C      ORIGINATED IN . (TRACER POINTS CHANGE POSITION IN
C      XP AND YP ARRAYS WHEN THEY ARE WEEDED OUT
C      DURING REZONE.)
C
IJ(N)=2**((NRZ+1)*(I-1)+1
JK(N)=2**((NRZ+1)*(J-1)+1
IF (N.LT.5) GO TO 300
WRITE (6,500) (IJ(M),JK(M),CMXP(M),CMYP(M),M=1,N)
N=0
300 CONTINUE
IF (N.EQ.0) GO TO 305
WRITE (6,500) (IJ(M),JK(M),CMXP(M),CMYP(M),M=1,N)
305 IF (IMAX.EQ.1) GO TO 360
C
C      . . . . .
C
C      *** PRINT SYMBOLIC CONTOUR MAPS OF COMPRESSION, PRESSURE,
C      VELOCITY, AND INTERNAL ENERGY UNLESS DOING A 1-D
C      PROBLEM.
C
C      . . . . .
C
CALL MAP
C
C      . . . . .
C
C      *** COMPUTE CRATER DEPTH AND VOLUME. AID SUMS DEPTH.
C
C      . . . . .
WRITE(6,490)
AID = 0.
C      *** START AT AXIS
DO 330 I =1, I1
CRAD(I) = .5*DX(I)+X(I-1)
PL(I) = 0.
UL(I) = 0.
DO 320 J =1, I2
K=(J-1)*IMAX + I + 1
C      *** WS IS COMPRESSION
WS = AMX(K)/(TAU(I)*DY(J)*RHOZ)
IF(WS.LT.(.99)) GO TO 310

```

```

      GO TO 325
310 AID = AID + 1.-WS
C      *** NOT AT BOTTOM OF CRATER YET
320 CONTINUE
325 IAID = INT(AID)
C      *** UL(I) IS CM. DEPTH OF CRATER IN COLUMN I
C      *** PL(I) IS CELL DEPTH OF CRATER IN COLUMN I
      UL(I) = Y(IAID) + DY(IAID+1)*(AID-FLOAT(IAID)) - Y(JPROJ)
      IF(UL(I).GT.0..OR.UL(I).LT.0.) PL(I) = AID
      AID = 0.
330 CONTINUE
C      *** PRINT CRATER DEPTHS
      DO 340 I=1,I1
      IF(UL(I).LT.0..OR.UL(I).GT.0.) GO TO 335
      GO TO 340
335 WRITE(6,495) I, PL(I), CRAD(I), UL(I)
340 CONTINUE
C      *** COMPUTE CRATER VOLUME AND VOLUME OF HEMISPHERE WITH
C      RADIUS=UL(I).
      WSB=0.
      DO 345 I=1,I1
      IF(UL(I).LT.0.) GO TO 350
C      *** WSB GIVES CRATER VOLUME
      WSB = UL(I)*TAU(I)+WSB
345 CONTINUE
350 CONTINUE
C      *** PRINT CRATER VOLUME ONLY WHEN GREATER THAN ZERO
      IF(WSB.GT.0.) GO TO 355
      GO TO 360
C      *** WSC GIVES VOLUME OF HEMISPHERE
355 WSC=2.0944*(UL(I))**3
      WRITE(6,498) WSB, WSC
C      . . . . .
C      . . . . .
C      *** SHORT PRINT MEANS I3=1 AND PROPERTIES ARE PRINTED ONLY
C      FOR CELLS IN FIRST COLUMN. LONG PRINT MEANS I3=I1 AND
C      PROPERTIES ARE PRINTED FOR ALL CELLS IN ACTIVE GRID.
C      . . . . .
C      . . . . .
360 DO 410 I=1,I3
      KSPACE=0
      WFLAGP=1.
      J=I2+1
      K=I2*IMAX+1+I
      DO 400 L=1,I2
      J=J-1
      K=K-IMAX
365 IF (AMX(K)) 440,390,370
370 IF (WFLAGP.EQ.0.) GO TO 380
      WRITE (6,550) I,X(I),DX(I)
      WFLAGP=0.
380 WS=AMX(K)/((TAU(I)*DY(J))
      WSA=WS/RHOZ
      WSC=P(K)
      WRITE (6,510) J,U(K),V(K),WSC,AMX(K),WS,AIX(K),WSA,Y(J)
      KSPACE=0
      GO TO 400

```

```

390 KSPACE=KSPACE+1
    IF (KSPACE.GT.1) GO TO 400
    WRITE (6,560)
400 CONTINUE
410 CONTINUE
C
    IF (NPRINT.EQ.1) GO TO 412
    ASSIGN 412 TO LOCA
    ASSIGN 412 TO LOCB
    IF (PRDEL.T.NE.0.) GO TO 50
    GO TO 100
C
    *** CHECK ON SIZE OF ENERGY DISCREPANCY
412 IF (ABS(ECK).GT.DMIN) GO TO 430
C
    *** IF LAST CYCLE, REWIND TAPE
414 IF (WFLAGL.EQ.0.) GO TO 416
    REWIND 7
416 WFLAGP=0.
    WFLAGF=0.
C
    *** NERR=1 WHEN EDIT IS CALLED BY ERROR.
    IF (NERR.EQ.1) RETURN
C
    *** SHOULD GRID BE REZONED ON THIS CYCLE
    IF ((REZ.NE.0..AND.REZFCT.NE.0..AND.NUMREZ.GT.0).OR.SS4.NE.0.) GO
1TO 419
C
    . . . . .
C
    RETURN
C
    . . . . .
C
    *** R E Z O N E
C
    . . . . .
419 CALL REZONE
C
    *** MUST CALL CDT TO RECALCULATE PRESSURES
    TNOW=T
    DTNOW=DT
    REZ=0.
    SS4=0.
    CALL CDT
    T=TNOW
    DT=DTNOW
    DTNA=DT
    NUMREZ=NUMREZ-1
C
    *** NREZ = NUMBER OF REZONES ALLOWED (INPUT VALUE OF NUMREZ)
    NUMREZ = NUMBER OF REZONES ALLOWED MINUS THE NUMBER
    OF REZONES PERFORMED SINCE T=0.
C
    NRZ=NREZ-NUMREZ
C
    *** NZ USED IN PRINTOUT OF TOTALS FOR 1-D PROBLEMS
    NZ=4,**NRZ
C
    GO TO 145
C
    . . . . .
C
    *** ERROR CONDITIONS

```

```

C . . . . . *** PRINT DELTA NOT SPECIFIED IN INPUT . . . . .
C
420 NK=40
    GO TO 450
C
430 NK=412
    GO TO 450
C
440 NK=365
450 NR=5
    CALL ERROR

C
C          FORMATS
C
490 FORMAT (1H0,17X,35HDEPTH OF CRATER MEASURED FROM JPROJ//12X,1HI,5X
1,18HJ OF CRATER BOTTOM,12X,1HR,11X,17HDEPTH IN CM. D(I)//)
495 FORMAT (I13,9X,0PF6.1,13X,1PE10.4,9X,1PE10.4)
498 FORMAT (//6X,13HCRATER VOLUME,11X,43HCRATER VOLUME BASED ON (2/3)
1* PI * D(1)**3/7X,1PE10.4,26X,1PE10.4)
500 FORMAT (5(I4,I4,1P2E9.2))
510 FORMAT (I4,1X,1P2E14.6,3E15.6,E14.6,E15.6,E14.6)
520 FORMAT(8H1PROBLEM,6X,4HTIME,8X,5HCYCLE,3X,13HTOT.EN.THEOR.3X,
1 19HMAX.REL.ERROR-CYCLE,3X,18HIE SET TO ZERO-PH1,3X,
2 18HIE SET TO ZERO-PH2,3X,12HPLASTIC-WORK/1F8.4,2X,1PE13.7,
3 3X,I4,4X,1PE13.7,3X,1PE13.7,1X,I4,6X,1PE13.7,8X,1PE13.7,6X,
4 1PE13.7/)
530 FORMAT (18X,2HIE,14X,2HKE,7X,13HTOT.EN. (SUM),7X,4HMASS,12X,2HMV,8
1X,12HMV(POSITIVE),8X,2HMU,8X,12HMU(POSITIVE)/11H J.GT.JPROJ,1P8E15
2.7/11H J.LE.JPROJ,1P8E15.7/14X,12H-----,3X,12H-----,
33X,12H-----,3X,12H-----,3X,12H-----,3X,12H---
4-----,3X,12H-----,3X,12H-----,3X/7H TOTALS,4X,1P
58E15.7//9H BOUNDARY,9X,6HBOTTOM,9X,5HRIGHT,10X,3HTOP,8X,12H$EVAPO
6RATEDS//9H MASS OUT,2X,1P4E15.7/11H ENERGY OUT,1P4E15.7/7H MU OUT,
74X,1P4E15.7/7H MV OUT,4X,1P4E15.7//11H WORK DONE ,1P3E15.7//)
540 FORMAT (1H0//21H TAPE 7 DUMP ON CYCLEI5//)
550 FORMAT (1H //4H I =I3,6X,6HR(I) =F12.3,6X,7HDR(I) =E14.7//3H J8X
1,1HU13X,1HV13X,3H P 12X,3HAMX12X,3HRHO11X,3HAIX12X,4HCOMP11X,2H Z/
2)
560 FORMAT (1H0)
570 FORMAT (//22H J OF PRESSURE-MAXIMUM/(25I5))
580 FORMAT(//103H TRACER POINTS - INITIAL LOCATION IN CELL COORDINATES
1 (I,J) - CURRENT LOCATION IN CM. COORDINATES (X,Y)// 5(4H I,3X,
21HJ,5X,1HX,8X,1HY,3X))
605 FORMAT(//41H ANGULAR DISTRIBUTION OF MASS AND MOMENTA/130H TAN U/V
1 0-.02 .02-.04 .04-.06 .06-.08 .08-.10 .10-.15 .15-.20 .20-
2.25 .25-.30 .30-.40 .40-.50 .50-1.0 1.0-UP V.LE.0./)
610 FORMAT(5H MASS,14(1X,1PE8.2))
615 FORMAT(5H MU ,14(1X,1PE8.2))
620 FORMAT(5H MV ,14(1X,1PE8.2))
    END

```

SUBROUTINE MAP

C  
C  
.....  
DIMENSION AMX(2502), AIX(2502), U(2502) , V(2502) , P(2502) ,  
1 X(52) , XX(54) , TAU(52) , JPM(52) ,  
2 Y(102) , YY(104) , FLEFT(102), YAMC(102), SIGC(102),  
3 GAMC(102),  
4 PK(15), Z(150) ,  
5 XP(26,51), YP(26,51),  
6 PL(204) , UL(204) , PR(204) ,  
7 RSN(52), RST(52),  
8 CMXP(5) , CMYP(5) , IJ(5) , JK(5) ,  
9 DX(52) , DDX(54) , DY(102) , DDY(104) ,  
S SNB(52) , STB(52) , UK(52,3) , VK(52,3) , RHO(52,3)

C  
C  
\*\*\* B L A N K C O M M O N  
DIMENSIONED VARIABLES  
COMMON Z  
COMMON PK  
COMMON YY, XX  
COMMON DDX, DDY  
COMMON AMX, AIX, U, V, P  
COMMON TAU, JPM  
COMMON UL, PL,  
COMMON XP, YP, CMXP, CMYP

C  
NON-DIMENSIONED VARIABLES  
COMMON AID , AMMV , AMMY , AMPY , AMUR , AMUT , AMVR ,  
1 AMVT , DELEB , DELER , DELET , DELM , DTDIX , DXYMIN , EAMMP , EAMPY ,  
2 ERDUMP , I , IWS , I3 , J , K , KA ,  
3 LL , MD , ME , MZT , NERR , NK , NPRINT ,  
4 NR , NRZ , NULLE , PIDTS , SIEMIN , SNR , SNT , STR , SOLID ,  
5 SSUM , TESTRH , TWOPI , URR , WS , WSA , WSB , WSC , WFLAGF ,  
6 WFLAGL , WFLAGP  
COMMON LAST

C  
C  
C  
C  
C  
\*\*\* THE FOLLOWING EQUIVALENCES DEFINE STORAGE FOR  
X(0), Y(0), DX(0), DY(0)

EQUIVALENCE (XX(2), X(1)) , (YY(2), Y(1)),  
1 (DDX(2), DX(1)) , (DDY(2), DY(1))

C  
C  
C  
\*\*\* SPECIAL EQUIVALENCES FOR PH2 ONLY

EQUIVALENCE (UL, FLEFT), (UL(103), YAMC),  
1 (PL, GAMC, PR), (PL(103), SIGC)

C  
C  
C  
\*\*\* SPECIAL EQUIVALENCES FOR PH3 ONLY

EQUIVALENCE (UL, RSN),  
1 (PL, RST), (P, UK),  
2 (P(157), VK), (P(313), SNB),  
3 (P(365), STB), (P(417), RHO)

C  
C  
C  
\*\*\* SPECIAL EQUIVALENCES FOR EDIT

EQUIVALENCE (PR(1), IJ), (PR(6), JK), (UL(103), CRAD)

C  
C  
C  
\*\*\* Z-STORAGE EQUIVALENCES

C

EQUIVALENCE  
 1(Z( 3),DT ),(Z( 4),NUMSP ),(Z( 5),NFRELP),(Z( 6),NDUMP7),  
 2(Z( 7),ICSTOP),(Z( 8),PIDY ),(Z( 9),TOPMU ),(Z( 10),RTMU ),  
 3(Z( 11),STK1 ),(Z( 12),NUMREZ),(Z( 13),ETH ),(Z( 14),UN14 ),  
 4(Z( 15),RHINIT),(Z( 16),PROJI ),(Z( 17),KUNIT ),(Z( 18),XMAX ),  
 5(Z( 19),NZ ),(Z( 20),NREZ ),(Z( 21),AMDM ),(Z( 22),UVMAX ),  
 6(Z( 23),UN23 ),(Z( 24),DMIN ),(Z( 25),JSTR ),(Z( 26),DTNA ),  
 7(Z( 27),CVIS ),(Z( 28),STK2 ),(Z( 29),STEZ ),(Z( 30),NC ),  
 8(Z( 31),UN31 ),(Z( 32),NRC ),(Z( 33),IMAX ),(Z( 34),IMAXA ),  
 9(Z( 35),JMAX ),(Z( 36),JMAXA ),(Z( 37),KMAX ),(Z( 38),KMAXA )

EQUIVALENCE  
 1(Z( 39),BOTM ),(Z( 40),BOTMV ),(Z( 41),NUMSPT),(Z( 42),CZERO ),  
 2(Z( 43),NUMSCA),(Z( 44),PRLIM ),(Z( 45),PRDELTA),(Z( 46),PRFACT)

EQUIVALENCE  
 1(Z( 47),I1 ),(Z( 48),I2 ),(Z( 49),IPCYCL),(Z( 50),TSTOP ),  
 2(Z( 51),RHOFIL),(Z( 52),TARGV ),(Z( 53),N3 ),(Z( 54),IVARDY),  
 3(Z( 55),VT ),(Z( 56),N6 ),(Z( 57),RTM ),(Z( 58),RTMV ),  
 4(Z( 59),UN59 ),(Z( 60),N10 ),(Z( 61),N11 ),(Z( 62),GAMMA ),  
 5(Z( 63),TOPM ),(Z( 64),BOTMU ),(Z( 65),SN ),(Z( 66),TOPMV ),  
 6(Z( 67),PRYBOT),(Z( 68),PRYTOP),(Z( 69),PRXRT ),(Z( 70),CYCPH3),  
 7(Z( 71),REZFCT),(Z( 72),TARGI ),(Z( 73),PROJU ),(Z( 74),BBOUND),  
 8(Z( 75),EVAP ),(Z( 76),ECK ),(Z( 77),NECYCL),(Z( 78),II ),  
 9(Z( 79),JJ ),(Z( 80),NMP ),(Z( 81),Y2 ),(Z( 82),EZPH1 )

EQUIVALENCE  
 1(Z( 83),IVARDX),(Z( 84),T ),(Z( 85),NMPMAX),(Z( 86),PMIN ),  
 2(Z( 87),INTER ),(Z( 88),TAYBOT),(Z( 89),TAYTOP),(Z( 90),UN90 ),  
 3(Z( 91),MC ),(Z( 92),MR ),(Z( 93),MZ ),(Z( 94),MB )

EQUIVALENCE  
 1(Z( 95),REZ ),(Z( 96),NODUMP),(Z( 97),UN97 ),(Z( 98),UN98 ),  
 2(Z( 99),UN99 ),(Z(100),EVAPM ),(Z(101),EVAPEN),(Z(102),EVAPMU),  
 3(Z(103),EVAPMV),(Z(104),EZPH2 ),(Z(105),SNL ),(Z(106),STL ),  
 4(Z(107),TAXRT ),(Z(108),MSYMBL),(Z(109),UN109 ),(Z(110),ROEPS ),  
 5(Z(111),RHINI ),(Z(112),VINI ),(Z(113),FINAL ),(Z(114),FRSTD ),  
 6(Z(115),RHOZ ),(Z(116),ESA ),(Z(117),ESEZ ),(Z(118),ESB ),  
 7(Z(119),ESCAPA),(Z(120),ESESP ),(Z(121),ESESQ ),(Z(122),ESES ),  
 8(Z(123),ESALPH),(Z(124),ESBETA),(Z(125),ESCAPB),(Z(126),UN126 ),  
 9(Z(127),SS1 ),(Z(128),SS2 ),(Z(129),UMIN ),(Z(130),SS4 )

EQUIVALENCE  
 1(Z(131),PRTIME),(Z(132),EOR ),(Z(133),EOT ),(Z(134),EOB ),  
 2(Z(135),EMOR ),(Z(136),DXF ),(Z(137),DYF ),(Z(138),RHOMIN),  
 3(Z(139),STAB),(Z(140),XIENRG),(Z(141),XKENRG),(Z(142),XTENRG),  
 4(Z(143),STT ),(Z(144),DTMIN ),(Z(145),TRNSFC),(Z(146),EMOT ),  
 5(Z(147),JPROJ ),(Z(148),CNAUT ),(Z(149),BBAR ),(Z(150),EMOB )

.....

END OF COMMON

.....

DIMENSION PROP(52), WSMAX(5), VALUE(41)

\*\*\* SPECIAL EQUIVALENCE FOR MAP

EQUIVALENCE (UL,PROP), (UL(52),WSMAX), (UL(157),VALUE)

DIMENSION ALE(41)

DATA ALE/ 2H .,2H -,2H A,2H B,2H C,2H D,2H E,2H F,

C  
C  
C  
C  
C  
C  
C  
C  
C  
C



```

1      2H G,2H H,2H I,2H J,2H K,2H L,2H M,2H N,2H O,
2      2H P,2H Q,2H R,2H S,2H T,2H U,2H V,2H W,2H X,
3      2H Y,2H Z,2H +,2H *,2H 1,2H 2,2H 3,2H 4,2H 5,
4      2H 6,2H 7,2H 8,2H 9,2H 0,2H /
      DIMENSION XUM(41)
      DATA XUM/      2H .,2H -,2H-A,2H-B,2H-C,2H-D,2H-E,2H-F,
1      2H-G,2H-H,2H-I,2H-J,2H-K,2H-L,2H-M,2H-N,2H-O,
2      2H-P,2H-Q,2H-R,2H-S,2H-T,2H-U,2H-V,2H-W,2H-X,
3      2H-Y,2H-Z,2H-+,2H-*,2H-1,2H-2,2H-3,2H-4,2H-5,
4      2H-6,2H-7,2H-8,2H-9,2H-0,2H /
      IDL=MINO(I1,54)
      JDL=12
      IF (NC.NE.0) GO TO 1
      IDL=MINO(IMAX,54)
      JDL=JMAX
C
C      *** FIND MAXIMUM VALUE IN ACTIVE GRID OF EACH PROPERTY
C
C      *** COMPRESSION
1      WSMIN=10E20
      WSMAX(1)=0.
      DO 2 J=1,JDL
      DO 2 I=1,IDL
      K=(J-1)*IMAX+I+1
      IF(AMX(K).EQ.0.) GO TO 2
      COMP = AMX(K)/(DY(J)*TAU(I)*RHOZ)
      WSMAX(1) = AMAX1(WSMAX(1),COMP)
      WSMIN = AMIN1(WSMIN,COMP)
2      CONTINUE
      IF(WSMAX(1).GT.WSMIN) GO TO 3
      WSMIN = 0.
C      *** PRESSURE
3      WSMAX(2)=0.
      DO 4 J=1,JDL
      DO 4 I=1,IDL
      K=(J-1)*IMAX+I+1
4      WSMAX(2) = AMAX1(WSMAX(2),ABS(P(K)))
C      *** RADIAL VELOCITY
      WSMAX(3)=0.
      DO 6 J=1,JDL
      DO 6 I=1,IDL
      K=(J-1)*IMAX+I+1
6      WSMAX(3) = AMAX1(WSMAX(3),ABS(U(K)))
C      *** AXIAL VELOCITY
      WSMAX(4)=0.
      DO 8 J=1,JDL
      DO 8 I=1,IDL
      K=(J-1)*IMAX+I+1
8      WSMAX(4) = AMAX1(WSMAX(4),ABS(V(K)))
C      *** SPECIFIC INTERNAL ENERGY
      WSMAX(5)=0.
      DO 10 J=1,JDL
      DO 10 I=1,IDL
      K=(J-1)*IMAX+I+1
10     WSMAX(5) = AMAX1(WSMAX(5),ABS(AIX(K)))
C
C      *** STORE INFORMATION TO BE PLOTTED IN PROP ARRAY

```

```

C           A ROW AT A TIME.
C
C           NPROP = 1
C           *** COMPRESSION
C           J=JDL
C           MS=MSYMBL+1
C           WRITE(6,500)
15          DO 20 I=1,IDL
C           K=(J-1)*IMAX+I+1
20          PROP(I) = AMX(K)/(TAU(I)*DY(J)*RHOZ)
C           GO TO 110
C
C           *** PRESSURE
C
C           J=JDL
30          MS=MSYMBL
C           WRITE(6,510)
35          DO 40 I=1,IDL
C           K=(J-1)*IMAX+I+1
40          PROP(I) = P(K)
C           GO TO 110
C
C           *** RADIAL VELOCITY
C
C           J=JDL
50          WRITE(6,520)
55          DO 60 I=1,IDL
C           K=(J-1)*IMAX+I+1
60          PROP(I) = U(K)
C           GO TO 110
C
C           *** AXIAL VELOCITY
C
C           J=JDL
70          WRITE(6,530)
75          DO 80 I=1,IDL
C           K=(J-1)*IMAX+I+1
80          PROP(I) = V(K)
C           GO TO 110
C
C           *** SPECIFIC INTERNAL ENERGY
C
C           J=JDL
90          WRITE(6,540)
95          DO 100 I=1,IDL
C           K=(J-1)*IMAX+I+1
100         PROP(I) = AIX(K)
C
C           *** WHEN PRINTING FIRST (TOP) ROW OF MAP, COMPUTE
C           SCALE FACTOR AND PRINT KEY.
110        IF(J.LT.JDL) GO TO 300
C
C           IF(WSMAX(NPROP).GT.0.) GO TO 180
C           *** SKIP CALCULATION OF SCALE FACTOR
C           GO TO 300
C
C           *** COMPUTE SCALE FACTOR AND PRINT MAXIMUM VALUE OF

```

```

C          EACH SYMBOL USED
C
180  SCALE = WSMAX(NPROP)/FLOAT(MS)
    IF ((AINT(SCALE*1000.)).LT.(SCALE*1000.)) GO TO 190
    GO TO 200
190  SCALE = AINT(SCALE*1000.+1)/1000.
200  CONTINUE
C
    IF(NPROP.EQ.1) GO TO 220
    VALUE(1) = 0.
    VALUE(2) = SCALE/10.
    DO 210 I=1,MS
210  VALUE(I+2) = FLOAT(I)*SCALE
    GO TO 240
C          *** VALUES FOR COMPRESSION MAP
220  VALUE(1) = WSMIN
    DO 230 I=1,MS
230  VALUE(I+1) = FLOAT(I)*SCALE
C          *** PRINT DEFINITIONS OF MAP SYMBOLS
240  ILIM1 = 1
    ILIM2 = 10
    MSP=MSYMBL + 2
250  IF (MSP.LT.ILIM2) ILIM2 = MSP
    IF (NPROP.NE.1) GO TO 260
    WRITE(6,550) (ALE(I),I=ILIM1,ILIM2)
    WRITE(6,560) (VALUE(I),I=ILIM1,ILIM2)
    GO TO 270
260  WRITE(6,570) (ALE(I),I=ILIM1,ILIM2)
    WRITE(6,580) (VALUE(I),I=ILIM1,ILIM2)
270  IF (MSP.EQ.ILIM2) GO TO 280
    ILIM1=ILIM2+1
    ILIM2=ILIM2+10
    GO TO 250
280  WRITE(6,590)
C
C          *** ASSIGN APPROPRIATE SYMBOL TO EACH CELL IN ROW J.
C
300  DO 370 I=1,IDL
    K=(J-1)*IMAX+I+1
    IF (AMX(K).GT.0.) GO TO 310
    MA = 41
    GO TO 360
310  IF(NPROP.EQ.1) GO TO 340
    IF(ABS(PROP(I)).GT.0.) GO TO 320
    MA = 1
    GO TO 360
320  IF(ABS(PROP(I)).GT.VALUE(2)) GO TO 330
    MA = 2
    GO TO 360
330  FLOTMA = ABS(PROP(I))/SCALE + 2.
    MA = INT(FLOTMA)
    IF(FLOTMA.GT.AINT(FLOTMA)) MA=MA+1
    MA = MAX0(MA,3)
    GO TO 360
C          *** DEFINE MA FOR COMPRESSION MAP
340  IF(PROP(I).GT.WSMIN) GO TO 350
    MA=1

```

```

GO TO 360
350 FLOTMA = ABS(PROP(I))/SCALE + 1.
    MA = INT(FLOTMA)
    IF(FLOTMA.GT.AINT(FLOTMA)) MA = MA+1
    MA = MAX0(MA,2)
C     *** STORE CHARACTER TO BE PLOTTED FOR CELL K
360 PR(I) = ALE(MA)
    IF(PROP(I).LT.0.) PR(I) = XUM(MA)
C     *** END OF I-LOOP
370 CONTINUE
C     *** PRINT J ROW OF MAP
    IF(MOD(J,5).NE.0) GO TO 380
    WRITE(6,600) J, (PR(I),I=1,IDL)
    GO TO 390
380 WRITE(6,610) (PR(I), I=1,IDL)
390 J=J-1
C     *** HAVE WE REACED BOTTOM ROW
    IF(J.EQ.0) GO TO 395
    GO TO (15,35,55,75,95),NPROP
C     *** PRINT AND LABEL X-AXIS OF MAP
395 PR(1) = ALE(29)
    WRITE(6,600) J, (PR(1),I=1,IDL)
    WRITE(6,620) (I, I=0,IDL,5)
C
    NPROP = NPROP + 1
    GO TO (400,30,50,70,90,400),NPROP
C
400 RETURN
C     *** FORMATS
500 FORMAT(1H1,4X,15HCOMPRESSION //)
510 FORMAT(1H1,4X,15HPRESSURE //)
520 FORMAT(1H1,4X,15HRADIAL VELOCITY//)
530 FORMAT(1H1,4X,15HAXIAL VELOCITY //)
540 FORMAT(1H1,4X,24HSPECIFIC INTERNAL ENERGY//)
550 FORMAT(16H SYMBOL ,10(4X,A2,4X))
560 FORMAT(16H MAXIMUM VALUE ,10(F6.3,4X))
570 FORMAT(16H SYMBOL ,10(3X,A2,5X))
580 FORMAT(16H MAXIMUM VALUE ,1P10E10.2)
590 FORMAT(//)
600 FORMAT(I10,2H I,54A2)
610 FORMAT(10X,2H I,54A2)
620 FORMAT(I12,10I10////)
END

```

## REQUIREMENTS FOR ENLARGING THE GRID

The RPM code, as it is listed in this report, can calculate at most 2500 cells. The number of rows (JMAX) cannot exceed 100, and the number of columns (IMAX) cannot exceed 50. To increase the grid size the user needs only to redimension most of the arrays in Blank Common and redefine some of the equivalences. Given IMAX and JMAX, the parameter definitions below show how the dimensions and equivalences should be redefined.

PARAMETERS: IJD = IMAX \* JMAX + 2

ID = IMAX + 2

IDP = IMAX + 4

JD = JMAX + 2

JDP = JMAX + 4

ITP = (IMAX + 2)/2

JTP = (JMAX + 2)/2

JD2 = 2 \* (JMAX + 2)

ID3 = 3 \* IDP

ID4 = 6 \* IDP

ID5 = 9 \* IDP

ID6 = 12 \* IDP

DIMENSIONS:

0 AMX(IJD), AIX(IJD), U(IJD), V(IJD), P(IJD),  
1 X(ID), XX(IDP), TAU(ID), JFM(ID);  
2 Y(JD), YY(JDP), FLEFT(JD), YAMC(JD), SIGC(JD),  
3 GAMC(JD),  
4 PK(15), Z(150),  
5 XP(ITP, JTP), YP(ITP, JTP),  
6 PL(JD2), UL(JD2), PR(JD2),  
7 RSN(ID), RST(ID),  
8 CMXP(5), CMYP(5), IJ(5), JK(5),  
9 DX(ID), DDX(IDP), DY(JD), DDY(JDP),  
\* SNB(ID), STB(ID), UK(ID,3), VK(ID,3), RHO(ID,3)

EQUIVALENCES:

(UL,FLECT), (UL(JDP),YAMC)

(PL,GAMC,PR), (PL(JDP),SIGC)

(P(ID3),VK), (P(ID4),SNB),

(P(ID5),STB), (P(ID6),RHO)

(PR(1),IJ), (PR(6),JK), (UL(JDP),CRAD)

