78
10-17
LA-6383

MASTER

m 685
UC-32 and UC-34b

Issued: January 1977

**PCS—An Euler-Lagrange Method for Treating
Convection in Pulsating Stars Using Finite
Difference Techniques in Two Spatial Dimensions**

by

Robert G. Deupree

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87545

An Affirmative Action/Equal Opportunity Employer

UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy \$5.00 Microfiche \$3.00

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

PCS—AN EULER-LAGRANGE METHOD FOR TREATING
CONVECTION IN PULSATING STARS USING FINITE
DIFFERENCE TECHNIQUES IN TWO SPATIAL DIMENSIONS

by

Robert G. Deupree

ABSTRACT

Finite difference techniques have been used to examine the coupling of radial pulsation and convection in stellar models having comparable time scales. Numerical procedures are emphasized, including diagnostics to help determine the range of free parameters.

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

I. INTRODUCTION

The computation of time-dependent convection in pulsating stars is probably one of the most difficult problems in multidimensional numerical hydrodynamics. Among the principal features that are encountered are: (1) The pulsation damping and driving mechanisms often balance each other to about 1%; consequently, great accuracy is required to obtain realistic pulsation growth rates. (2) The growth rates are so small that many pulsation periods must be computed; hence, high accuracy is required over a large number ($\sim 10^4$) of time steps. (3) In the interior regions of the model, departures from hydrostatic equilibrium (balance between pressure forces and gravity) are extremely small (residual may be $\sim 10^{-6}$ of the individual forces) and great accuracy is required. (4) Velocities may approach the sound speed; consequently some treatment for shock waves may be necessary. (5) Several optically thin zones must be included, requiring an implicit method of solution for the equation of energy conservation.¹ (6) There must be an adequate method of converting convective kinetic energy into heat. (7) One must have an explicit method of integrating the energy conservation equation in the deeper layers of the model.² (8) Great care must be taken to insure the numerical stability of the advective terms. (9) Numerical problems with a

steep energy gradient in the hydrogen ionization region² must be overcome.

(10) Adequate zoning in the hydrogen ionization region must be obtained to describe convection. The first five problems apply to purely radial pulsation features and have been dealt with at length in the literature.^{1,3,4} The remaining difficulties apply specifically to convection.

The physical approximations made have been discussed at some length in Ref.5 and can be summarized as follows.

1. Use of the diffusion approximation for radiative transfer.
2. ∇ (gravitational potential) = $-\frac{GM_r}{r^2} \hat{r}$.
3. Surface flux = $2 \sigma T_{\text{surf}}^4$.
4. Convection described by two-dimensional processes even though it is quite turbulent.
5. Conversion of convective kinetic energy into heat achieved by an "eddy viscosity" (as both molecular and radiative viscosity are far too small to make this conversion on any reasonable computational grid scale).

The model is a stellar envelope. The stellar core is treated as inert, with the boundary condition at the bottom of the envelope being constant luminosity emerging from the core and zero radial velocity at a fixed radius. The surface is a free surface with the emergent radiation being treated as a black body. Symmetry conditions are imposed horizontally across the side boundaries.

II. GENERAL PROCEDURE

The calculations are made on a two-dimensional grid defined in spherical coordinates covering a range in radius from the boundary of an inert core to the surface (and beyond), and a set range in colatitude, θ . The general configuration is shown in Fig. 1. There are NTHZ zones in θ , numbered from 2 to NTH (NTH = NTHZ + 1). The θ zones 1 and MAXTH (= NTH + 1) are "dummy" zones used to express the symmetry conditions across the two boundaries marked in the figure. The width of a θ zone, $\Delta\theta$, is constant. It is impossible to cover the entire range of θ ($0^\circ \leq \theta \leq 180^\circ$) with high resolution in θ (desirable values of $\Delta\theta$ are about 0.5°), so we have as free parameters TH(1) (value of θ at the midpoint of angular zone 1), DTH ($\Delta\theta$), and NTH.

The radial zone number of the surface is designated by NN, with one value for each colatitude grid point, as there is no reason for spherical symmetry to be maintained. For each grid point (radial zone #j, angular zone #k) we have the radial velocity component (v), θ velocity component (w) and other variables

Fig. 1

Outline of the general configuration. The radial coordinate is measured by zone number j [$j = 1, \dots, NM(l)$] and the angular coordinate by l ($l = 2, \dots, NTH$). Extra angular zones ($l = 1$ and $MAXTH$) are kept for the stipulation of the symmetry conditions across the angular boundaries. The angular width of a zone ($\Delta\theta$) is constant. Most quantities are evaluated at midpoints. The radial velocity component (v) is evaluated at angular midpoints and radial interfaces, and the theta velocity component (w) is evaluated at radial midpoints and angular interfaces.

(ρ , T , P , E) measured at the positions shown in Fig. 1. The radial coordinate, r (measured at radial and angular midpoints), is allowed to flow with the horizontal mean of the radial velocity, V_o , whereas the horizontal coordinate, θ , is fixed.

In general, the variables have three subscripts: the first referring to the radial zone number, the second to the angular zone number, and the third to the time step (a value of 1 here means the beginning of the time step, a value of 2 means the end of the time step).

The equations to be solved may be written figuratively as

$$\frac{\partial \rho}{\partial t} = f \quad \text{mass conservation} \quad (1)$$

$$\frac{\partial v}{\partial t} = g_1 \quad (2)$$

momentum conservation

$$\frac{\partial w}{\partial t} = g_2 \quad (3)$$

$$\frac{\partial E}{\partial t} = h \quad \text{energy conservation} \quad (4)$$

$$\frac{\partial r}{\partial t} = V_o \quad \text{radial coordinate motion} \quad (5)$$

I shall outline the method of solution using the main program of the program PCS. A listing is included. First, a number of variables are defined in DATA statements. The philosophy of using DATA statements arises from the problems of converting from one computer system to another: DATA statements are probably the easiest way to convert from single to double precision (required for an IBM machine). These variables will be defined as they occur.

Define NTH, NTHZ, MAXTH, DTHO (θ in degrees), DTH (θ in radians), TH (θ), SINE ($\sin\theta$), and COSI ($\cos\theta$). The following parameters are read from a punched card:

NPRINT - number of models computed between detailed printing of results

IPLT - same as NPRINT except for plotting

IPNE - same as NPRINT except for punching

IMOD - number of time steps to be computed

ISTRT - number governing reading in of model

IWORK - number governing whether PdV work is computed
 DTAI - fraction of Courant condition time step taken
 ALPHA - donor cell parameter for momenta equations
 ALPHAI - donor cell parameter for mass and energy equations
 CVISC - coefficient for eddy viscosity.

Next the opacity data is read in through UREAD. The initial model is read in next. If ISTPT = 1, this is a model set in hydrostatic and radiative equilibrium with respect to the difference equations of PCS (generated by another program); otherwise, a previous run of PCS is read in. MODTOT is the total number of time steps that have been computed to the beginning of this run. The horizontal average of the radial velocity component (VBAR) is computed in RSTEP. This concludes the general set-up stage (100).^{*} The following steps are carried out every time step:

1. Set horizontal symmetry conditions (110).
2. Set condition at inert core (120).
3. Find N1 = min (NN) and N2 = max (NN).
4. Compute artificial viscosity (430) and set horizontal pressure symmetry condition (420).
5. Integrate momentum conservation equations (2) and (3) (VMSOL).
6. Set $w = 0$ below IRADIL to avoid excessively small time steps imposed by the Courant condition due to the small value of $r\Delta\theta$ (IRADIL should be sufficiently small that convection is not expected to penetrate to that depth) (70).
7. Modify surface velocity (660).⁵
8. Compute work if IWORK > 0 (PDV).
9. Solve for the new horizontal average of the radial velocity and check to see if rezoning is required (RSTFP).
10. Integrate Eq. (4) explicitly in the layers interior to IEDIR (see Ref. 2 for discussion) (ESOL).
11. Integrate Eq. (1) (RHOSOL).
12. Using internal energy at new time (from ESOL) and density at new time (from RHOSOL), find temperature at new time in interior layers (ESOL1).
13. Advance r to half time step [RADV(1)].
14. Solve Eq. (4) implicitly in outer layers (DPSET) using the temperatures at the new time in zone IEDIR as the interior boundary condition.

^{*}

Numbers in parentheses refer to statement numbers in the program.

15. Advance r to new time [RADV(2)].
16. Evaluate interior mass at new time for use in gravitational force (FRAC).
17. Set quantities to zero in layers external to surface (450).
18. Advance time.
19. Compute new pressure and internal energy and evaluate Courant condition every ICOURKth time step (THERM).
20. Print surface values of velocity, temperature, radius, and luminosity (540).
21. Evaluate maximum convective velocity (130).
22. Print complete model results according to NPRINT (1000).
23. Compute kinetic energy if reference velocity is large enough (EKLK); reference velocity, governed by radial zone IVTST, should be chosen near node of 1H to be free of modal contamination.
24. Check to see if rezoning is required (REZONE).
25. Punch model results on cards according to IPNH (PNCARD).
26. Plot results according to IPLT (GCODY).
27. Do another time step according to JMOD.

The time spent in various subroutines or performing tasks is summarized in Table I. These results are for a case in which the energy equation required two iterations to converge, and the model is composed of 36 radial zones and two angular zones. The times of performance are given in seconds of CDC 7600 time. This rather shocking result for the Courant condition indicates that it should be evaluated as infrequently as possible; hence the use of ICOURK. I have found a practical limit to be about every fifth time step. A more desirable alternative would be a faster solution of the equation of state, and this is currently being examined. An increase of speed in the solution of the equation of state would also appear in the implicit solution of the energy equation, where the equation of state, including derivatives with respect to temperature, is evaluated repeatedly to determine the matrix coefficients.

A. Convective Free Parameters

The numerical restrictions are defined basically by three parameters: c (CVISC), α (ALPHA), and α_1 (ALPHA1). The last two refer to the donor cell contributions to the momentum and the mass and energy equations, respectively.


```

PROGRAM PCS(INP,FSET5=INP,OUT,FSET6=OUT,FSET7,FILM,FSET12=FILM)
COMMON / DRST / DRSTD
COMMON / QUANT / RHO(50,6,2), V(50,6,2), W(50,6,2), R(50), TH(6),
1 P(50,6), T(50,6), ENLHG(50,6), DELAD(50,6), ZL(50,4)
COMMON / ZMOD / X, V, ZMSTAR, ZMCORF, DT
COMMON / MUOPR / NTH, NTHZ, MAXTH, MZONE, MAXZ
COMMON / VSTF / DYMAX
COMMON / SFAC / NN(10), N1, N2, NSKP
COMMON / FSIN / SINE(10), COSI(10), DTH
COMMON / GRAV / ZMSKR(50)
COMMON / IPRT / IPRTM
COMMON / VISCOS / CVISC, ALPHA, ALPHA1, VCONV, IVISC0
COMMON / RADIUS / RNEW(50), VBAR(50)
DIMENSION NOTE(6), ISEQ(3)
DATA NLAS1, ICAL, ITER, IPRT, IOUT, IEDIR, IHADIL / 0, 1, 0, 0, 0, 4, 6 /
DATA G, PI, TIME, VMULT1, Tn, Z0, CPO, OMNO, TFM, FVO, IPTM / 0, 0.000E-2
1 3.14159E+0, 0.0E+0, 1.7E+2, 0.1, 0.0E+0, 2.0E+0, 0.0E+0, 1.1E+0,
1 1.1E+2, 4.5E-2, 4.5E+2, 1.2E-1 /
C CHANGE THESE CARDS WHEN CHANGING THE NUMBER OF ANGULAR ZONES
NTH=1
DTH=0.25
ICOURK=5
ICOUR=0
VCONV=0.
C CHANGE THE FOLLOWING CARDS WHEN THE NUMBER OF RADIAL ZONES CHANGES
MAXZ=50
MZONE=00
NTHZ=NTH=1
MAXTH=NTH+1
DTH=PI/182.*NTHZ
TH(1)=PI/TH/Tn
DO 299 L=2, MAXTH
299 TH(L)=TH(L-1)+DTH
DO 301 J=1, MAXTH
NOTE(J)=0
COSI(J)= COS(TH(J))
301 SINE(J)= SIN(TH(J))
COSI(1)=Tn- COS(TH(2))
SINE(MAXTH)=Tn- SIN(TH(NTH))
C IF ZONES DO NOT MAKE UP 90 DEGREES, SET SINE TO REAL VALUE IN LAST A
COSI(1)= COS(TH(1))
SINE(MAXTH)= SIN(TH(MAXTH))
C NOTE THE ABOVE
READ(5,4) NPRINT, IPLT, IPNH, IMOD, ISTRY, INORK, DIA1, ALPHA, ALPHA1,
1 CVISC
4 FORMAT(6I5, 1X, F4.2, 1X, F4.2, 1X, F4.2, 1X, E9, 3)
CALL UREAD
GO TO (40, 50) , ISTRY
40 CALL STATIC (NUMZN, DRSTD, VTEST, IVTST, ISEQ)
MODTOT=0
GO TO 60
50 CALL RMDM (TIME, DRSTD, VTEST, IVTST, MODTOT, INORK, ISEQ)
60 CONTINUE
WRITE(6,8) ISEQ(1), ISEQ(2), ISEQ(3), MODTOT, NPRINT, IPLT, IPNH, IMOD,
1 NTHZ, DTH0, ALPHA, ALPHA1, CVISC, INORK
8 FORMAT(///24H SUMMARY UPACITY TABLE , I4, 3X, 10H1 (IN 10**33)=, I4,
13X, 6HT EFF=, I4, 3X, 7HMODTOT=, I6, /, 7H PRINT=, I5, 10X, 5HPLOT=, I5, 10X,
26HPUNCH=, I5, 10X, 18HMODELS TO BE DONE=, I5, 5X, 24HNUMBER OF ANGULAR

```

```

3ONES=,13,5X,4MDTH=,F6,3,/,7M ALPHA=,F4,2,3X,7M ALPHA1=,F4,2,3X
46MCVISC=,E9,3,/,7M INORK=,13,///)
  IPLT=IPLT
  CALL RSREP (DT,NTH,N2)
180 CONTINUE
  DO 110 J=1,MAXZ
    RHO(J,1,1)=RHO(J,2,1)
    RHO(J,MAXTH,1)=RHO(J,NTH,1)
    V(J,1,1)=V(J,2,1)
    V(J,MAXTH,1)=V(J,NTH,1)
    N(J,1,1)=Z0
    N(J,MAXTH,1)=-N(J,4THZ,1)
    T(J,1)=T(J,2)
    T(J,MAXTH)=T(J,NTH)
    ENERG(J,1)=ENERG(J,2)
    ENERG(J,MAXTH)=ENERG(J,NTH)
110 CONTINUE
  DO 120 L=1,MAXTH
    V(1,L,1)=Z0
    V(1,L,2)=Z0
    N(1,L,1)=Z0
    N(1,L,2)=Z0
    RHO(1,L,2)=RHO(1,L,1)
120 CONTINUE
  N1=NN(2)
  N2=NN(2)
  DO 430 L=2,NTH
    K1=NN(L)
    IF (K1.GT,N2) N2=K1
    IF (K1.LT,N1) N1=K1
    DO 430 J=2,K1
      PP=P(J,L)
      QQ=VISC(RHO(J,L,1),PP,V(J,L,1),V(J+1,L,1),N(J,L,1),N(J,L-1,1))
430 P(J,L)=PP+QQ
      K=NN(2)
      IF (NN(NTH).GT,K) K=NN(NTH)
      DO 420 J=1,K
        P(J,1)=P(J,2)
        P(J,MAXTH)=P(J,NTH)
420 CONTINUE
C   USE THESE CARDS FOR ONE STEP TIME INTEGRATION SCHEME
  N2=NN(2)
  DO 680 L=2,NTH
    IF (NN(L).GT,N2) N2=NN(L)
680 CONTINUE
  ISGN0=1
  IF(V(IVTST,2,1).LT,Z0) ISGN0=-1
  CALL VMSOL ('MZONE,NTH,DT,IRADIL)
  DO 70 I=1,IRADIL
    DO 70 J=1,MAXTH
      N(I,J,1)=Z0
70 N(I,J,2)=Z0
  DO 670 L=1,MAXTH
    K=NN(L)
    DO 660 J=N1,K
      V(J,L,1)=VMULT*V(J+1,L,1)
660 V(J,L,2)=VMULT*V(J+1,L,2)
      V(K+1,L,1)=V(K,L,1)

```

```

670 V(K+1,L,2)=V(K,L,2)
    IF (IWORK,LE,0) GO TO 65
    ISGNN=1
    IF (V(IVTST,2,1).LT,Z0) ISGNN=-1
    IPRD=ISGNN*ISGNN
    IF (ISGNN.GT,0) IPRD=1
    CALL PDV(UT,IPRD)
65  CONTINUE
    CALL RSTEP (DT,NTH,N2)
    CALL ESOL (NTH,IEDIR,DT)
    CALL RHOSOL (MZONE,NTH,DT,IOUT,NOTE)
    IF (IOUT,NE,0) GO TO 541
    CALL ESOL1 (NTHZ,IEDIR)
    CALL RADV (1)
    CALL DPSFT (IOUT,IEDIR,NOTE)
    IF (IOUT,NE,0) GO TO 541
    CALL RADV (2)
    DO 440 L=1,MAXTH
440  NOTE(L)=0
    CALL FRAC (N2,NTH)
    NN(1)=NN(2)
    NN(MAXTH)=NN(NTH)
    DO 450 L=2,NTH
    KJ=NN(L)+1
    K2=KJ+5
    IF (K2.GT,MAXZ) K2=MAXZ
    DO 450 J=K1,K2
    T(J,L)=Z0
    ENERG(J,L)=Z0
    RHO(J,L,1)=Z0
    RHO(J,L,2)=Z0
    V(J,L,1)=Z0
    V(J,L,2)=Z0
450  P(J,L)=Z0
    N1=MAXZ
    N2=0
    DO 600 L=2,NTH
    K=NN(L)
    IF (K.GT,N2) N2=K
    IF (K.LT,N1) N1=K
    IF (T(K,L),LE,Z0) T(K,L)=T(K-1,L)
600  CONTINUE
C  END OF CHANGES FOR ONE STEP INTEGRATION SCHEME
    TIME=TIME+DT
    ICOUR=ICOUR+1
    IF (ICOUR,LE,0) GO TO 630
    DO 610 L=2,NTH
    K=NN(L)
    DO 610 J=2,K
610  CALL THERM (RHO(J,L,1),T(J,L),P(J,L),ENERG(J,L),DUT,DPT,DPR,
    A  DPR1,VAD,1)
    GO TO 620
630  CONTINUE
    ICOUR=ICOURK
    DAX=DHND*DT
    DO 240 L=2,NTH
    CALL THERM (RHO(1,L,1),T(1,L),P(1,L),ENERG(1,L),DUT,DPT,DPR,
    1  DPR1,VAD,1)

```

```

      K=NN(L)
      IF (V(K,L,1).EQ.Z0) V(K,L,1)=V(K-1,L,1)
      DO 240 J=2,K
      DD=AMIN1 (R(J)*DTH,R(J)-R(J-1))
      IF (Y(J,L).LE.Z0) IOUT=1
      IF (RHO(J,L,1).LE.Z0) IOUT=1
      IF (IOUT.NE.0) GO TO 541
      CALL THERM (RHO(J,L,1),T(J,L),P(J,L),ENERG(J,L),DUT,DPT,DPR,
1      DELAD(J,L),VAD,3)
      QQ=VAD+SQRT((V(J,L,1)-VBAR(J))**2+W(J,L,1)**2)
      IF (J.LE.IRADIL) DD=R(J)-R(J-1)
      DT=DTAI*DD/QQ
      IF (DT,LT,DAX) DAX=DT
240  CONTINUE
620  CONTINUE
      DT=1.D+0
      IF (TPERTM,GT,TPTM) GO TO 250
      IPERT=IPERT-1
      IF (IPERT.EQ.0.AND.TPARM,GT,TPM) GO TO 250
      GO TO 260
250  IPERT=5
260  IF (IPERT,GT,0) DT=3.D-1/DTAI
      DT=DT*DAX
      IF (DT,GT,FV0) DT=FV0
      ITER=ITER+1
      DO 540 L=2,NTH,2
      LP1=L+1
      K=NN(L)
      FJ=AN(L+1)
      WRITE (6,6) K,L,K1,LP1,TIME,R(K),I(K,L),V(K,L,1),ZL(K,L-1),
1      T(K1,LP1),V(K1,LP1,1),ZL(K1,L)
6  FORMAT (1X,4I3,1P8E13,5)
540  CONTINUE
      K=IRADIL+5
      VCONV=0.
      IVISCO=-1
      DO 130 L=2,NTH
      KK=NN(L)
      DO 130 J=K,KK
      VPRES=ABS(W(J,L,2))
      IF (VPRES,GT,VCONV) VCONV=VPRES
      VPRES=ABS(V(J,L,1)-VBAR(J))
      IF (VPRES,GT,VCONV) VCONV=VPRES
130  CONTINUE
      IF (VCONV,GT,2.E+5) IVISCO=1
      IF (VCONV,GT,6.E+5) IVISCO=2
      MODTOT=MODTOT+1
      IF ((ITER/NPRINT).LE,NLAST) GO TO 220
      NLAST=ITER/NPRINT
541  CONTINUE
      WRITE(6,1) MODTOT,TIME
1  FORMAT (15H1 STEP NUMBER ,I6,2X,5HTIME=,E14,6)
      DO 1000 L=2,NTH
      WRITE (6,2) TH(L)
2  FORMAT (//1X,6HTHETA=,F7,4,/,/,8X,1HR,13X,3HRHO,10X,1HP,13X,1HT,
1  10X,6HENERGY,11X,1HV,13X,1HW,11X,5HDELAD,11X,1HL)
      K1=NN(L)
      WRITE (6,3) (R(J),RHO(J,L,1),P(J,L),T(J,L),ENERG(J,L),V(J,L,1),

```

```

      1 W(J,L,1),DELAD(J,L),ZL(J,L-1),J=2,K1)
      3 FORMAT (9(1X,1PE13,5))
1000 CONTINUE
220 CONTINUE
    VVTEST=ABS(V(IVTST,2,1))
    IF (VVTEST .GT. VTEST) CALL EKIN (NTH,N2)
    CALL REZONE (NTH,N1)
    IF (IOUT.NE.0) STOP
    IF (ITER.NE.(IPNH*ICAL)) GO TO 221
    ICAL=ICAL+1
    CALL PNCHMD (TIME,DRSTD,VTEST,IVTST,MODIOT,IWORK,ISEQ)
221 CONTINUE
    IPLT1=IPLT1-1
    IF (IPLT1.NE.0) GO TO 230
    IPLT1=IPLT1
    K=NN(2)
    VS=V(K,2,1)
    CALL GOODY (TIME,VS,N1,NTH,IRADIL)
230 CONTINUE
    NN(1)=NN(2)
    NN(MAXT)=NN(NTH)
    IF (ITER.LT.IMOD) GO TO 100
    STOP
    END

```

TABLE I

COMPUTATIONAL SPEED OF VARIOUS SUBROUTINES OR FEATURES
IN SECONDS OF CDC 7600 TIME

Feature	Computation Time (s)
VWSOL	0.009
RSTEP	0.002
ESOL	0.005
RHOSOL	0.004
ESOLI	0.002
RADV	0.003
Solution of energy equation	0.089
Evaluation of Courant condition	0.024
Remainder	<u>0.010</u>
Total	0.148

They are applied only to the advective terms, which may be written figuratively:

$$\begin{aligned} \text{difference term} = & (1 - \alpha) \times \text{centered difference term} + \\ & \alpha \times \text{donor cell term} \quad (0 \leq \alpha \leq 1). \end{aligned}$$

The donor cell algorithm has been discussed by Hirt et al.,^{6,7} Ruppel and Cloutman,⁸ and Cloutman et al.⁹ The donor cell terms are quite dissipative, so values of α and α_1 should be kept as small as allowable for numerical stability. Hirt et al.⁷ give a prescription for a rectangular mesh

$$\alpha \geq \max_i \left(\left| \frac{u \delta t}{\delta x} \right|_i \text{ and } \left| \frac{v \delta t}{\delta y} \right|_i \right). \quad (6)$$

The value of α_1 is chosen to be different (less than) from that of α because the nonphysical effects are usually greater in the mass and energy equations and the momentum equations require larger values for stability.¹⁰

The constant c is taken as the free parameter for the kinematic eddy viscosity coefficient (ν). Somewhat in analogy with Deardorff's^{11,12} subgrid scale viscosity, I shall assume

$$\nu = C \Delta r \, r \Delta \theta. \quad (7)$$

The approach adopted for the velocity dependence of these three parameters is

$$\begin{aligned} c = \alpha = \alpha_1 = 0 & \quad v_m < v_0 \\ c = c_0 v_m, \alpha = \alpha_0, \alpha_1 = \alpha_{10} & \quad v_0 \leq v_m < v_1 \\ c = c_0 v_m, \alpha = 2\alpha_0, \alpha_1 = 2\alpha_{10} & \quad v_m \geq v_1, \end{aligned} \quad (8)$$

where v_m is the maximum value of the convective velocity ($\equiv \underline{v} - v_0 \hat{r}$) over the grid mesh. Characteristic values of (v_0, v_1) are (2,6.5) km/s. The primary difficulty with the computational procedure is the selection of values for the set (c_0, v_0, α_{10}) that will yield physically plausible results. Consequently, I shall now consider the diagnostics that allow one to find the range of realistic values for these constants, particularly c_0 . One has considerably more latitude in the selection of values for α_0 and α_{10} .

Obviously, both α_0 and α_{10} must be less than 0.5. If the value of either is too small, numerical instability is encountered. It usually manifests itself in the tendency for all the matter to flow out of one of the interior zones. Should this happen, a diagnostic (FORMAT 1 of subroutine RHOSOL) is printed, and execution terminated. It is more difficult to tell whether either of these is too large. Since the effects in this case can be more or less offset by decreasing c_0 , my policy has been to set α_0 and α_{10} at what I consider reasonable values, and to change c_0 . The values of α_0 and α_{10} must be largest for full amplitude pulsation, for which one may expect the range $0.15 \leq \alpha_{10} \leq 0.35$, $0.3 \leq \alpha_0 \leq 0.45$.

The selection of c_0 is much more sensitive. An example of the case in which c_0 is too small is given in Fig. 2. The surface luminosity suddenly and effectively drops to zero. When c_0 is too small, convective kinetic energy is not converted into heat, so less energy is radiated from the surface. When this occurs, it is necessary to go back at least one period to restart the calculation.

I have removed the difficulty of the Rayleigh number being too small for convection to develop by setting $c_0 = 0$ for small convective velocities. If c_0 is very large, the limiting convective velocity will be very close to v_0 . One may find during the calculation that c_0 requires an upward revision. It is easy to overestimate this revision, so that c_0 is too large for the present convective velocities. This leads to a large (and, with the present assumptions, a physically unrealistic) temperature inversion in the top one or two optically thin zones. A temperature inversion may also occur, however, if c_0 is sufficiently small that the convective velocities reach the sound speed near the surface, with the artificial viscosity providing the dissipation.

There appears to be no easy way to guess the correct values of these parameters, particularly c_0 , a priori. The diagnostics given above can indicate only when something has gone wrong; perhaps with experience the number of mistakes can be reduced.

B. Restrictions on the Time Step

There are three major restrictions on the time step: the Courant condition, accuracy limitations, and a viscosity restriction. The first two are well known in astrophysical hydrodynamics. The third is less familiar to astrophysicists. Suppose that $r \, d\theta$ is much less than dr and that the eddy kinematic viscosity coefficient is large. Then

Fig. 2

Demonstration of the results when the eddy viscosity is too small. The surface luminosity (flux radiated from the surface of the star) drastically decreases as convection converts internal energy into kinetic energy. Because the viscosity is too small, the kinetic energy is not converted into internal energy and hence cannot be radiated by the star.

$$\frac{\partial w}{\partial t} \approx v \frac{\partial^2 w}{r^2 \partial \theta^2} . \quad (9)$$

A plausible difference equation might be

$$w_{ij}^{n+1} = w_{ij}^n + \delta t \, v \frac{w_{i,j+1}^n - 2w_{ij}^n + w_{i,j-1}^n}{r^2 (\Delta \theta)^2} . \quad (10)$$

For the case with only two angular zones, this becomes

$$w_{ij}^{n+1} = w_{ij}^n - \delta t \, v \frac{2w_{ij}^n}{r^2 (\Delta \theta)^2} . \quad (11)$$

Obviously, if δt is large enough, $|w_{ij}^{n+1}| > |w_{ij}^n|$, with the values at n and $n + 1$ having opposite signs. This generates a numerical instability with characteristics of extremely rapid growth (less than 10 time steps are required before completely unrealistic velocities are reached) and alternating signs for velocities at adjacent grid points. The condition for stability in this simple case is

$$\delta t \leq \frac{(r \, \Delta \theta)^2}{v} . \quad (12)$$

Fortunately, this simple case appears to be the primary one encountered in these calculations (at least when two angular zones are used), undoubtedly because of the small value of $r \, d\theta$ in the interior regions. The exception occurs when the radial zoning becomes too small near the surface. To prevent undesirably small time steps in this latter situation, rezoning is necessary. Again, the indication of what is wrong lies in the rapid growth of the instability, alternating velocities, and unrealistic velocity amplitudes. The feature that actually terminates execution is the outflow of all the matter in one of the zones more than five zones interior to the surface (FORMAT 1 of subroutine REOSOL).

The various subroutines can be broken up into several groups. The next section will deal with those that explicitly integrate equations (VWSOL, RLVSC, RHOSOL, RADV, ESOL, and ESOL1). The subroutines involved in the implicit solution of the energy conservation equation (DPSET, DP3, COEF, and COEF1) are described in Sec. IV. The following sections consider rezoning (REZONE, INTRP, RSTEP, and REVAMP), the equation of state (THERM, EEERG, and PIERW), starting and ending

a run (UREAD, RDMD, STATIC, and PNCHMD), opacity calculation (OPACIT and ZKAP), assorted subsidiary calculations (PDV, EKIN, FRAC, and VISC), and plotting (GOODY and MNMX).

III. EXPLICIT INTEGRATIONS

This section contains an outline of the subroutines involved in the explicit integration of the two components of the momentum equation (VWSOL), mass equation (RHOSOL), energy equation in the interior (ESOL, ESOL1), and the increment in radius (RAEV). The energy equation must be explicitly integrated in the interior:² if the implicit Penney method¹³ is used in the interior, the error in the matrix reduction produces temperature gradients which ultimately generate unrealistically large velocities. Such might not be the case if all equations were solved simultaneously and implicitly. An implicit approach must be used in the zones close to the surface.¹

A. VWSOL

The two components of the momentum equation may be written¹⁴

$$\begin{aligned} \frac{\partial v}{\partial t} + v_1 \frac{\partial v}{\partial r} + \frac{w}{r} \frac{\partial v}{\partial \theta} - \frac{w^2}{r} = -\frac{1}{\rho} \frac{\partial P}{\partial r} - \frac{GM_r}{r^2} \\ + v \left[\frac{1}{2} \frac{\partial^2 (rv_1)}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 v_1}{\partial \theta^2} + \frac{\cot \theta}{r^2} \frac{\partial v_1}{\partial \theta} - \frac{2}{r^2} \frac{\partial w}{\partial \theta} - \frac{2 v_1}{r^2} - \frac{2 \cot \theta}{r^2} w \right], \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{\partial w}{\partial t} + v_1 \frac{\partial w}{\partial r} + \frac{w}{r} \frac{\partial w}{\partial \theta} + \frac{vw}{r} = -\frac{1}{\rho r} \frac{\partial P}{\partial \theta} + 2 \left[\frac{1}{r} \frac{\partial^2 (rw)}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} \right. \\ \left. + \frac{\cot \theta}{r^2} \frac{\partial w}{\partial \theta} + \frac{2}{r^2} \frac{\partial v_1}{\partial \theta} - \frac{w}{r^2 \sin^2 \theta} \right], \end{aligned} \quad (14)$$

where $v_1 = v - V_0$. v_1 must be applied to the advective terms if the coordinate system is to move with velocity V_0 (which I take to be the horizontal average of the radial velocity component). v_1 was used instead of v in the viscosity terms because it seemed reasonable to apply the eddy viscosity only to the convective motion.

After defining various quantities that depend only on the radial zoning, I test to see if the density in the zone is less than the permissible minimum. If

so, the velocities are set so that there is no flow out of the zone, although flow into the zone is, of course, allowed (150). If there is enough mass in the zone, the eddy viscosity coefficient is set. The value of the eddy viscosity is zero for those zones where only radial motion is allowed ($\leq$ IRADIL) and for those zones too close to the surface ($> N1-4$). The reasoning for the first is straightforward, as only pulsation is allowed in the deeper layers. The second deletion was based on calculations showing that unrealistic results occurred when viscosity was allowed too close to the surface. I have no concrete idea why this should be the case, but including the surface zones in the viscous treatment produces physically reasonable results, at least as long as the hydrogen ionization region does not penetrate these zones.

Next, the convective radial velocities are defined for several zones (VC1-VC5) and the viscous terms computed. The subroutine ELVSC computes

$$\left. \begin{aligned} \text{TERM 1} &= \frac{1}{r} \frac{\partial^2 (rv_1)}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 v_1}{\partial \theta^2} + \frac{\cot \theta}{r^2} \frac{\partial v_1}{\partial \theta} - \frac{2}{r^2} \frac{\partial w}{\partial \theta} \\ \text{TERM 2} &= \frac{1}{r} \frac{\partial^2 (rw)}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 w}{\partial \theta^2} + \frac{\cot \theta}{r^2} \frac{\partial w}{\partial \theta} + \frac{2}{r^2} \frac{\partial v_1}{\partial \theta} \end{aligned} \right\} \quad (15)$$

using the simplicity of the terms included in TERM1 and TERM2 in the calling sequence. The remaining terms in brackets of Eqs (13) and (14), are computed to make the two viscous terms [Q5V and Q5W in Eqs. (13) and (14), respectively]. The curvature terms

$$\left. \begin{aligned} Q4W &= \frac{vw}{r} \rightarrow \frac{v_{i+1/2,j} w_{i,j+1/2}}{r_i} \\ Q5W &= -\frac{w^2}{r} \rightarrow -\frac{w_{i,j+1/2}^2}{r_i} \end{aligned} \right\} \quad (16)$$

are computed. They are not space-centered, but have been negligible in all publications of this program to date. Next, the pressure and gravity forces

$$\left. \begin{aligned}
Q1V &= \left(\frac{1}{\rho} \frac{\partial P}{\partial r} + \frac{GM_r}{r^2} \right)_{i+1/2,j} + \frac{2}{\rho_{i,j} + \rho_{i+1,j}} \frac{P_{i+1,j} - P_{i,j}}{r_{i+1} - r_i} \\
&\quad + \frac{GM_{r_{i+1/2}}}{r_{i+1/2}^2} \\
Q1W &= \left(\frac{1}{\rho r} \frac{\partial P}{\partial \theta} \right)_{i,j+1/2} + \frac{2}{\rho_{i,j} + \rho_{i,j+1}} \left(P_{i,j+1} - P_{i,j} \right)
\end{aligned} \right\} \quad (17)$$

are computed. The spatial step needed for Q1W is included when the new velocities are calculated. The terms are set to zero if the density average in the denominator vanishes. (In fact, the density average should not vanish if the previously mentioned test has been made to see if the density was too small.) Finally, the advective terms are computed; first, the centered terms

$$\begin{aligned}
Q2v &= (v - v_0) \frac{\partial v}{\partial r} + \frac{1}{4} \left(v_{i+3/2,j} + v_{i+1/2,j} - v_{0_{i+3/2}} - v_{0_{i+1/2}} \right) \frac{v_{i+3/2,j} - v_{i+1/2,j}}{r_{i+3/2} - r_{i+1/2}} \\
&\quad + \frac{1}{4} \left(v_{i+1/2,j} + v_{i-1/2,j} - v_{0_{i+1/2}} - v_{0_{i-1/2}} \right) \frac{v_{i+1/2,j} - v_{i-1/2,j}}{r_{i+1/2} - r_{i-1/2}}, \quad (18)
\end{aligned}$$

$$\begin{aligned}
Q3v &= w \frac{\partial v}{r \partial \theta} + \frac{1}{4} \left(w_{i,j+1/2} + w_{i+1,j+1/2} \right) \frac{v_{i+1/2,j+1} - v_{i+1/2,j}}{r_{i+1/2} \Delta \theta} \\
&\quad + \frac{1}{4} \left(w_{i,j-1/2} + w_{i+1,j+1/2} \right) \frac{v_{i+1/2,j} - v_{i+1/2,j-1}}{r_{i+1/2} \Delta \theta}, \quad (19)
\end{aligned}$$

$$\begin{aligned}
Q2w &= (v - v_0) \frac{\partial w}{\partial r} + \frac{1}{4} \left(v_{i+1/2,j} + v_{i+1/2,j+1} - 2v_{0_{i+1/2}} \right) \frac{w_{i+1,j+1/2} - w_{i,j+1/2}}{r_{i+1} - r_i} \\
&\quad + \frac{1}{4} \left(v_{i-1/2,j} + v_{i-1/2,j+1} - 2v_{0_{i-1/2}} \right) \frac{w_{i,j+1/2} - w_{i-1,j+1/2}}{r_i - r_{i-1}}, \quad (20)
\end{aligned}$$

$$Q3w = \frac{w}{r} \frac{\partial w}{\partial \theta} \rightarrow \frac{1}{4r_i \Delta \theta} \left[(w_{i,j+1/2} + w_{i,j+3/2})(w_{i,j+3/2} - w_{i,j+1/2}) + (w_{i,j+1/2} + w_{i,j-1/2}) \right. \\ \left. \times (w_{i,j+1/2} - w_{i,j-1/2}) \right] . \quad (21)$$

The V terms are centered at $(i+\frac{1}{2}, j)$ and the w terms at $(i, j+\frac{1}{2})$.

If the peak convective velocity is sufficiently large, donor cell terms are added. Defining

$$v_{i+1/2,j}^1 = v_{i+1/2,j} - v_{0_{i+1/2}} \\ w_{i+1/2,j} = \frac{1}{4}(w_{i,j+1/2} + w_{i,j-1/2} + w_{i+1,j+1/2} + w_{i+1,j-1/2}) \\ v_{i,j+1/2} = \frac{1}{4}(v_{i+1/2,j} + v_{i+1/2,j+1} + v_{i-1/2,j} + v_{i-1/2,j+1}) \\ - 2v_{0_{i+1/2}} - 2v_{0_{i-1/2}} \quad (22)$$

then

$$Q2v_{nc} = \frac{1}{2} \left[(v_{i+1/2,j}^1 - |v_{i+1/2,j}^1|) \frac{v_{i+3/2,j} - v_{i+1/2,j}}{r_{i+3/2} - r_{i+1/2}} \right. \\ \left. + (v_{i+1/2,j}^1 + |v_{i+1/2,j}^1|) \frac{v_{i+1/2,j} - v_{i-1/2,j}}{r_{i+1/2} - r_{i-1/2}} \right] , \quad (23)$$

$$Q3v_{nc} = \frac{1}{2r_{i+1/2} \Delta \theta} \left[(w_{i+1/2,j} - |w_{i+1/2,j}|) (v_{i+1/2,j+1} - v_{i+1/2,j}) \right. \\ \left. + (w_{i+1/2,j} + |w_{i+1/2,j}|) (v_{i+1/2,j} - v_{i-1/2,j}) \right] , \quad (24)$$

$$Q2w_{nc} = \frac{1}{2} \left[\left(v_{i,j+1/2} - |v_{i,j+1/2}| \right) \frac{w_{i+1,j+1/2} - w_{i,j+1/2}}{r_{i+1} - r_i} + \left(v_{i,j+1/2} + |v_{i,j+1/2}| \right) \frac{w_{i,j+1/2} - w_{i-1,j+1/2}}{r_i - r_{i-1}} \right], \quad (25)$$

$$Q3w_{nc} = \frac{1}{2r_i \Delta \theta} \left[\left(w_{i,j+1/2} - |v_{i,j+1/2}| \right) \left(w_{i,j+3/2} - w_{i,j+1/2} \right) + \left(w_{i,j+1/2} + |v_{i,j+1/2}| \right) \left(w_{i,j+1/2} - w_{i,j-1/2} \right) \right]. \quad (26)$$

The centered and noncentered terms are added together, e.g.,

$$Q2v = (1-\alpha) Q2v + \alpha Q2v_{nc}. \quad (27)$$

The new velocities (VN and WN) are found, transposed into the old arrays (V and W), and the boundary conditions set (250).

B. PHCSOL

The equation of mass conservation is

$$\frac{\partial \rho}{\partial t} + (v - v_0) \frac{\partial \rho}{\partial r} + \frac{\rho}{r^2} \frac{\partial}{\partial r} (r^2 v) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \rho w) = 0. \quad (28)$$

The centered difference terms are written as

$$Q1R = \frac{\rho}{r^2} \frac{\partial}{\partial r} (r^2 v) \rightarrow \frac{\rho_{1,j}}{r_i^2 (r_{i+1/2} - r_{i-1/2})} \left[r_{i+1/2}^2 v_{i+1/2,j} - r_{i-1/2}^2 v_{i-1/2,j} \right], \quad (29)$$

$$Q2R = \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \rho w) \rightarrow \frac{1}{2r_i \sin \theta_j \Delta \theta} \times \left[\sin \theta_{j+1/2} w_{i,j+1/2} (\rho_{1,j} + \rho_{1,j+1}) - \sin \theta_{j-1/2} w_{i,j-1/2} (\rho_{1,j} + \rho_{1,j-1}) \right], \quad (30)$$

$$Q3R = \left(v - v_0 \right) \frac{\partial \rho}{\partial r} + \frac{1}{2} \left[\left(v_{i+1/2,j} - v_{0_{i+1/2}} \right) \frac{\rho_{i+1,j} - \rho_{i,j}}{r_{i+1} - r_i} + \left(v_{i-1/2,j} - v_{0_{i-1/2}} \right) \frac{\rho_{i,j} - \rho_{i-1,j}}{r_i - r_{i-1}} \right]. \quad (31)$$

Equations (29)-(31) are centered at (i,j). Defining

$$v_{i,j}^1 = \frac{1}{2} \left(v_{i+1/2,j} + v_{i-1/2,j} - v_{0_{i+1/2}} - v_{0_{i-1/2}} \right) \quad (32)$$

the noncentered radial part of the advective term becomes

$$Q3R_{nc} \rightarrow \frac{1}{2} \left[\left(v_{i,j}^1 - |v_{i,j}^1| \right) \frac{\rho_{i+1,j} - \rho_{i,j}}{r_{i+1} - r_i} + \left(v_{i,j}^1 + |v_{i,j}^1| \right) \frac{\rho_{i,j} - \rho_{i-1,j}}{r_i - r_{i-1}} \right]. \quad (33)$$

As Q2R in Eq. (30) is only partially an advective term, its alteration is a little more complicated. If

$$w_{i,j} = \frac{1}{2} \left(w_{i,j+1/2} + w_{i,j-1/2} \right), \quad (34)$$

then

$$Q2R_{nc} \rightarrow \frac{\rho_{i,j}}{r_i \sin \theta_j \Delta \theta} \left(\sin \theta_{j+1/2} w_{i,j+1/2} - \sin \theta_{j-1/2} w_{i,j-1/2} \right) + \frac{1}{2r_i \Delta \theta} \left[\left(w_{i,j} - |w_{i,j}| \right) \left(\rho_{i,j+1} - \rho_{i,j} \right) + \left(w_{i,j} + |w_{i,j}| \right) \left(\rho_{i,j} - \rho_{i,j-1} \right) \right]. \quad (35)$$

The first term is the nonadvective part and remains space-centered, while the second is treated with the donor cell prescriptions. The addition of centered and noncentered terms is done as in Eq. (27). If the new density is less than some specified value (RHOMN), the radial zone number is checked to see if it is too small (110). If so, too many zones are lost near the surface, and execution is terminated (through IOUT after returning to the main program and printing the present model: see Sec. II). If the radial zone number is close to the surface

(less than five zones away), the velocities are set to zero so that there is no motion out of the region where there is no mass. After all zones have been computed, the density at the new time is shifted into the array for the density at the old time, and the density at the half time is put into the array for the new time (200). Horizontal boundary conditions are set (250).

C. ESOL

The energy conservation equation in the interior is integrated

$$\begin{aligned} \frac{\partial E}{\partial t} + (v - v_0) \frac{\partial E}{\partial r} + \frac{w}{r} \frac{\partial E}{\partial \theta} + \frac{P}{\rho} \left[\frac{1}{2} \frac{\partial}{\partial r} (r^2 v) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta w) \right] = \\ - \frac{1}{\rho r^2} \frac{\partial}{\partial r} (r^2 F_r) - \frac{1}{\rho r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta F_\theta) \end{aligned} \quad (36)$$

where

$$\begin{aligned} F_r &= - \frac{4\sigma}{3\kappa\rho} \frac{\partial T^4}{\partial r} \\ F_\theta &= - \frac{4\sigma}{3\kappa\rho} \frac{1}{r} \frac{\partial T^4}{\partial \theta} \end{aligned} \quad (37)$$

To save computer time, $F_{r,i-1/2,j}$ (FLUX1) and $F_{r,i+1/2,j}$ (FLUXE) are stored. For the first radial zone, $F_{r,i-1/2,j} = F_0$, the constant flux emerging from the inert core (FLUX). After all horizontal zones for each radial zone have been computed, the radial zone number is advanced and FLUX1 is set equal to FLUXE. $F_{\theta,j,j-1/2}$ (FLUX1) and $F_{\theta,j,j+1/2}$ (FLUX2) are treated in a similar way. The difference expression for the radiative flux is

$$F_{r,i+1/2,j} = - \frac{4\sigma}{3} \left(\frac{1}{\kappa\rho} \right)_{i+1/2,j} \frac{T_{i+1}^4 - T_i^4}{r_{i+1} - r_i}, \quad (38)$$

and there are similar expressions for the horizontal flux. The quantity $\left(\frac{1}{\kappa\rho} \right)_{i+1/2,j}$ (FK1) is the Christy¹ weighted opacity computed by OPACIT (see Sec. VIII). The difference terms are expressed as

$$\text{TERM1} = \frac{1}{\rho r^2} \frac{\partial}{\partial r} (r^2 F_r) \rightarrow \frac{1}{\rho_{i,j} r_i^2} \frac{r_{i+1/2}^2 F_{r_{i+1/2,j}} - r_{i-1/2}^2 F_{r_{i-1/2,j}}}{r_{i+1/2} - r_{i-1/2}}, \quad (39)$$

$$\text{TERM2} = \frac{1}{\rho r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta F_\theta) \rightarrow \frac{1}{\rho_{i,j} r_i \sin \theta_j \Delta \theta} \left(\sin \theta_{j+1/2} F_{\theta_{i,j+1/2}} - \sin \theta_{j-1/2} F_{\theta_{i,j-1/2}} \right), \quad (40)$$

$$\text{TERM3} = \frac{P}{\rho} \left[\frac{1}{r^2} \frac{\partial}{\partial r} (r^2 v) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta w) \right] \rightarrow \frac{P_{i,j}}{\rho_{i,j}} \left[\frac{1}{r_i^2} \frac{r_{i+1/2}^2 v_{i+1/2,j} - r_{i-1/2}^2 v_{i-1/2,j}}{r_{i+1/2} - r_{i-1/2}} + \frac{1}{r_i \sin \theta_j \Delta \theta} (\sin \theta_{j+1/2} w_{i,j+1/2} - \sin \theta_{j-1/2} w_{i,j-1/2}) \right], \quad (41)$$

$$\text{TERM4} = (v - v_0) \frac{\partial E}{\partial r} \rightarrow \frac{1}{2} \left[\left(v_{i+1/2} - v_{0_{i+1/2}} \right) \frac{E_{i+1,j} - E_{i,j}}{r_{i+1} - r_i} + \left(v_{i-1/2,j} - v_{0_{i-1/2}} \right) \frac{E_{i,j} - E_{i-1,j}}{r_i - r_{i-1}} \right] \quad (42)$$

$$\text{TERM5} = \frac{w}{r} \frac{\partial E}{\partial \theta} \rightarrow \frac{1}{2} \left[w_{i,j+1/2} \frac{E_{i,j+1} - E_{i,j}}{r_i \Delta \theta} + w_{i,j-1/2} \frac{E_{i,j} - E_{i,j-1}}{r_i \Delta \theta} \right]. \quad (43)$$

Donor cell corrections have not been used here, and one must therefore keep in mind that large convective motions should not penetrate to these deeper layers (radial zone number < IEDIR). The final calculation (TERM6) is effectively a calculated zero and could (and probably should) be omitted. The energies at the advanced time step are stored as the array EKP.

D. ESOL1

Given the internal energy at the advanced time step in the interior zones (from ESCL) and the density at the advanced time step (computed by RHOSOL), the equation of state is used to obtain the temperatures at the advanced time step. In these deep layers, complete ionization is assumed, so that

$$E = 3/2 R(1/2 + 3/2 X + 1/4 Y) T + \frac{aT^4}{\rho} + \text{Ionization Energy} \quad (44)$$

$$\rightarrow 1.5 P1 + T + P \times T^4 + \text{UTERM.}$$

The derivative $\left(\frac{\partial E}{\partial T}\right)_\rho$ is found and, starting with the temperature at the beginning of the time step, the temperature is iterated by

$$E_{\text{correct}} = E(T_{\text{guess}}) + \left(\frac{\partial E}{\partial T}\right)_\rho \Delta T_{\text{guess}} \quad (45)$$

Convergence is considered obtained when $\Delta T_{\text{guess}}/T_{\text{guess}}$ is less than a certain value (FACMN). The value is very small because of the close balance between pressure forces and gravity in this region. If convergence fails in 20 tries, a diagnostic is printed, but execution is not terminated. In this case, something is probably drastically wrong and execution will soon be terminated by other means.

E. RADV

Equation (5) has been integrated by subprogram R_{STEP} (see Sec. V) for the value of the radial coordinate at the advanced time (RNEW). This subroutine replaces the current value of the radial coordinate by the value at the half time step (INDEX=1) or by the value at the advanced time (INDEX=2). The half time step value is needed for the implicit solution of the energy equation.

```

SUBROUTINE VWSOL (MAXZ,NTH,DT,IRADIL)
COMMON / RADIUS / RNEW(50),VBAR(50)
COMMON / DRST / DRSTD
COMMON / GRAV / ZMSMR(50)
COMMON / FSIN / SINE(10),COSI(10),DTH
COMMON / QUANT / RHO(50,6),RHOM(50,6),V(50,6),VN(50,6),W(50,6),
1 WN(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),MAST(50,6)
COMMON / SFACE / NN(10),N1,N2,NSKP
COMMON / VISCOS / CVI ,ALPHA,ALPHA1,VCONV,IVISCO
DATA G,PI,OHN,TW,RHOMIN,RHOMN,ZO,ON,OHNEG,OND / 6.668E-8,
A 3.14159, 0.5, 2., 1.E-11, 1.E-9, 0., 1., -1., 0.25 /
N=N2+5
IF (N.GT.MAXZ) N=MAXZ
RJP=OHN*(R(2)+R(1))
RJM=TW*(R(1)-R(2))
DO 220 J=2,N
RJM1=RJM
RJM=RJP
RJP=OHN*(R(J)+R(J+1))
WF=OHN*(R(J)+R(J+1))*DTH
DR1=R(J+1)-R(J)
DR2=R(J)-R(J-1)
R1=R(J)*DTH
DO 200 L=2,NTH
IF (RHO(J,L).LE.RHOMIN) GO TO 150
CVISC=ZO
IF (J.LT.(N1-4)) CVISC=CVI*R1*(RJP-RJM)*(VCONV/6.5E+5)
IF (J.LE.IRADIL) CVISC=ZO
VC=V(J,L)
WC=W(J,L)
VC1=VC-VBAR(J)
VC2=V(J+1,L)-VBAR(J+1)
VC3=V(J-1,L)-VBAR(J-1)
VC4=V(J,L+1)-VBAR(J)
VC5=V(J,L-1)-VBAR(J)
TERM1=RLVSC (RJP,RJM,RJM1,DTH,VC2,VC1,VC3,VC4,VC5,W(J,L+1),WC,
A R(J),COSI(L),SINE(L),OHNEG)
TERM2=RLVSC (R(J+1),R(J),R(J-1),DTH,W(J+1,L),WC,W(J-1,L),
A W(J,L+1),W(J,L-1),VC4,VC1,RJM,COSI(L),SINE(L),ON)
Q5V=CVISC*(TERM1-TW/RJM**2*(VC+COSI(L)/SINE(L)*WC))
Q5W=CVISC*(TERM2-WC/(R(J)*SINE(L))**2)
Q4V=-WC**2/R(J)
Q4W=VC*WC/R(J)
RHJP=OHN*(RHO(J,L)+RHO(J+1,L))
RHLP=OHN*(RHO(J,L)+RHO(J,L+1))
WTE=WC/WF
Q1V=ZO
Q1W=ZO
IF (RHO(J+1,L).LE.RHOMN) DR1=DRSTD
IF (RHJP,NE,ZO) Q1V=G*ZMSMR(J+1)/RJP**2*(P(J+1,L)-P(J,L))/RHJP/DR1
DR1=R(J+1)-R(J)
IF (RHLP,NE,ZO) Q1W=(P(J,L+1)-P(J,L))/RHLP
VJP=V(J+1,L)
RJP1=OHN*(R(J+2)+R(J+1))
VJM=V(J-1,L)
WLP=W(J,L+1)
WLM=W(J,L-1)

```

```

      Q2V=ONG*((VC+VJP-VBAR(J)-VBAR(J+1))*(VJP-VC)/(RJP1-RJP)+
1      (VC+VJM-VBAR(J)-VBAR(J+1))*(VC-VJM)/(RJP-RJM))
      Q3W=ONG*((WC+WLP)*(WLP-WC)+(WC+WLM)*(WC-WLM))/R1
      Q3V=ONG*((WC+W(J+1,L))*(V(J,L+1)-VC)+(WLM+W(J+1,L-1))*(VC-
1      V(J,L-1)))/WF
      Q2W=ONG*((VC+V(J,L+1)-TW*VBAR(J))*(W(J+1,L)-WC)/DR1+
1      (VJM+V(J-1,L+1)-TW*VBAR(J-1))*(WC-W(J-1,L))/DR2)
      IF (IVISCO,LT,0) GO TO 90
      ALPKP=ALPHA
      IF (IVISCO,GE,1) ALPHA=ALPHA*(VCONV/2,E+5)
      IF (ALPHA,GE,1.) ALPHA=1.
      VABS1=ABS(VC-VBAR(J))
      Q2V=(ON-ALPHA)*Q2V+ALPHA/TW*((VJP-VC)/(RJP1-RJP)*(VC-VBAR(J)+
A      VABS1)+(VC-VJM)/(RJP-RJM)*(VC-VBAR(J)+VABS1))
      VABS1=ABS(WC)
      Q3W=(ON-ALPHA)*Q3W+ALPHA/TW/R1*((WLP-WC)*(WC-VABS1)+(WC-WLM)*
A      (WC+VABS1))
      WCENT=(WC+W(J+1,L)+WLM+W(J+1,L-1))/TW/TW
      VABS1=ABS(WCENT)
      Q3V=(ON-ALPHA)*Q3V+ALPHA/TW/WF*((V(J,L+1)-VC)*(WCENT-VABS1)+
A      (VC-V(J,L-1))*(WCENT+VABS1))
      VCENT=(VC1+VC3+VC4+V(J-1,L+1)-VBAR(J-1))*ONG
      VABS1=ABS(VCENT)
      Q2W=(ON-ALPHA)*Q2W+ALPHA/TW*((W(J+1,L)-WC)/DR1*(VCENT-VABS1)
A      +(WC-W(J-1,L))/DR2*(VCENT+VABS1))
      ALPHA=ALPKP
90      VN(J,L)=V(J,L)-DT*(Q1V+Q2V+Q3V+Q4V+Q5V)
      WN(J,L)=W(J,L)-DT*(Q1W/R1+Q2W+Q3W+Q4W+Q5W)
      GO TO 200
150      VN(J,L)=Z0
      IF (WN(J,L),GT,Z0) WN(J,L)=Z0
      IF (WN(J,L-1),LT,Z0) WN(J,L-1)=Z0
200      CONTINUE
220      CONTINUE
      DO 250 J=2,N
      DO 240 L=2,NTH
      W(J,L)=WN(J,L)
240      V(J,L)=VN(J,L)
      V(J,1)=V(J,2)
      VN(J,1)=VN(J,2)
      V(J,NTH+1)=V(J,NTH)
      VN(J,NTH+1)=VN(J,NTH)
      W(J,1)=Z0
      WN(J,1)=Z0
      WN(J,NTH+1)=-WN(J,NTH-1)
250      W(J,NTH+1)=-W(J,NTH-1)
      RETURN
      END

```

```

FUNCTION RLWSC (RU,RC,RL,DTH,UU,UC,UL,UTHP,
1      UTHM,XTHP,XC,R1,CS,SN,TYP)
DATA TW / 2.E+0 /
T1=TW*((RU+J1)-RC*UC)/(RU-RC)-(RC+UC-RL*UL)/(RC-RL)/RC/(RU-RL)
T2=(UTHP-T1+UC+UTHM)/(RC*DTH)**2
T3=CS/SN/DTH*(UTHP-UC)/RC/RC
RLWSC=T1+T2+T3+TYP*TW*(XTHP-XC)/R1/DTH/R1
RETURN
END

```

```

SUBROUTINE RHOSOL (MAXZ,NTH,DT,IOUT,NOTE)
COMMON / QUANT / RHO(50,6),RHON(50,6),V(50,6),VN(50,6),W(50,6),
1 WN(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),AST(500)
COMMON / FSIN / SINE(10),COSI(10),DTH
COMMON / SPACE/ NN(10),N1,N2,NSKP
COMMON / RADIUS / RNEW(50),VBAR(50)
DIMENSION NOTE(6)
COMMON / VISCOS / CVISC,ALPHA1,ALPHA,VCONV,IVISCO
DATA ONH,RHOMN,ZO,TH,P45,P55 / 5.E-1,1.E-9,0.E+0,2.E+0,4.5E-1,
1 5.5E-1 /
N=NN2+5
IF (N.GT.MAXZ) N=MAXZ
DO 140 J=2,N
RJP=ONH*(R(J)+R(J+1))
RJM=ONH*(R(J)+R(J-1))
RAN=R(J)*DTH
DR1=R(J+1)-R(J)
DR2=R(J)-R(J-1)
DO 120 L=2,NTH
IF (RHON(J-1,L),LE,RHOMN) GO TO 110
RHOO=RHO(J,L)
VC=V(J,L)
WC=W(J,L)
SINLP=ONH*(SINE(L)+SINE(L+1))
SINLM=ONH*(SINE(L)+SINE(L-1))
Q1R=RHOO*(RJP**2+VC-RJM**2+V(J-1,L))/R(J)**2/(RJP-RJM)
Q3R=ONH*((VC-VBAR(J))*(RHO(J+1,L)-RHOO)/DR1+(V(J-1,L)-VBAR(J-1))
1 *(RHOO-RHO(J-1,L))/DR2)
Q2R=ONH*(SINLP*WC*(RHO(J,L+1)+RHOO)-SINLM*W(J,L-1)*(RHO(J,L-1)
1 +RHOO))/SINE(L)/RAN
IF (IVISCO,LT,0) GO TO 90
ALPKP=ALPHA
IF (IVISCO,GE,1) ALPHA=ALPHA*(VCONV/2.E+5)
IF (ALPHA,GE,1.) ALPHA=1.
VAVE=ONH*(VC-VBAR(J)+V(J-1,L)-VBAR(J-1))
VABS1=ABS(VAVE)
Q3R=(1.+ALPHA)*Q3R+ALPHA*ONH*((RHO(J+1,L)-RHOO)/DR1*
A (VAVE-VABS1)+(RHOO-RHO(J-1,L))/DR2*(VAVE+VABS1))
VAVE=ONH*(WC+W(J,L-1))
VABS1=ABS(VAVE)
Q2R=(1.+ALPHA)*Q2R+ALPHA*RHOO/SINE(L)/RAN*(SINLP*WC-SINLM*
A W(J,L-1))+ALPHA*ONH/RAN*((RHO(J,L+1)-RHOO)*(VAVE-VABS1)+
B (RHOO-RHO(J,L-1))*(VAVE+VABS1))
ALPHA=ALPKP
90 CONTINUE
DRHC=DT*(Q1R+Q2R+Q3R)
RHON(J,L)=RHOO+DRHC
IF (RHON(J,L),LT,RHOMN) GO TO 110
NN(L)=J
GO TO 120
110 RHON(J,L)=ZO
IF (J,GT,(N1-5)) GO TO 115
WRITE (6,1) J,L
1 FORMAT (1X,10H TROUBLE IN RHOSOL,2X,2I4)
IOUT=1
NN(L)=NN(L)+5
IF (NN(L),GT,MAXZ) NN(L)=MAXZ

```

```

      RETURN
115 IF (W(J,L).GT,ZO) W(J,L)=ZO
      IF (W(J,L-1).LT,ZO) W(J,L-1)=ZO
      WN(J,L)=W(J,L)
      WN(J,L-1)=W(J,L-1)
      VN(J,L)=ZO
      V(J,L)=ZO
120 CONTINUE
140 CONTINUE
      DO 200 L=2,NTH
      K1=NN(L)+1
      IF (RHO(K1,L).GT,ZO) NOTE(L)=1
      DO 200 J=2,K1
      R1=P45 *RHO(J,L)+P55 *RHON(J,L)
      RHO(J,L)=RHON(J,L)
200  RHON(J,L)=R1
      NN(1)=NN(2)
      NN(NTH+1)=NN(NTH)
      K1=NN(2)
      IF (NN(NTH).GT,K1) K1=NN(NTH)
      DO 250 J=1,K1
      RHO(J,1)=RHO(J,2)
      RHO(J,NTH+1)=RHO(J,NTH)
      RHON(J,1)=RHON(J,2)
250  RHON(J,NTH+1)=RHON(J,NTH)
      RETURN
      END

```

```

SUBROUTINE ESOL (NTH,INDEX,DT)
COMMON / RADIUS / RNF(50),VRAR(50)
COMMON / QIANT / RMO(50,6),RMON(50,6),V(50,6),VN(50,6),W(50,6),
1 W(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),WAST(500)
COMMON / ZMATT / EXP(20,4),FLUXE(4),FLUXI(8),A(012)
COMMON / PSIN / SINE(10),COSI(10),DTH
COMMON / MELT / FLUX
COMMON / VISCOS / CVI ,ALPHA1,ALPHA,VCONV,IVISCO
DATA SIG43,ONH,ZERO,ONG,TH / 7.55066667E-5, 0.5, 0., 0.25, 2. /
IF (INDEX,1E.0) RETURN
DO 10 I=2,NTH
EXP(I,I-1)=ENERG(I,I-1)
10 FLUXI(I-1)=FLUX
RJP=ONH*(R(2)+R(1))
DO 100 I=2,INDEX
RJM=RJP
RJP=ONH*(R(I+1)+R(I))
RB1=(RJP/R(I))**2
RB2=(RJM/R(I))**2
RTH=R(I)*DTH
FLUX1=ZERO
SINLP=ONH*(SINE(1)+SINE(2))
DO 80 J=2,NTH
CALL OPACIT (T(I,J),T(I+1,J),RMO(I,J),RMO(I+1,J),FK1,D1,D2,-1)
CALL OPACIT (T(I,J),T(I,J+1),RMO(I,J),RMO(I,J+1),FK2,D1,D2,-1)
FLUXE(J-1)=SIG43*FK1*(T(I+1,J)**4-T(I,J)**4)/(R(I+1)-R(I))
FLUX2=SIG43*FK2*(T(I,J+1)**4-T(I,J)**4)/RTH
SINLM=SINLP
SINLP=ONH*(SINE(J)+SINE(J+1))
TERM1=(RB1*FLUXE(J-1)-RB2*FLUXI(J-1))/(RJP-RJM)/RMO(I,J)
TERM2=(SINLP*FLUX2-SINLM*FLUX1)/SINE(J)/RTH/RMO(I,J)
TERM3=(RQ1*V(I,J)-RB2*V(I-1,J))/(RJP-RJM)*(SINLP*W(I,J)-SINLM*
1 W(I,J+1))/SINE(J)/RTH
TERM3=P(I,J)/RMO(I,J)*TERM3
E0=ENERG(I,J)
TERM4=ONH*((V(I,J)-VBAR(I))*(ENERG(I+1,J)-E0)/(R(I+1)-R(I))+
1 (V(I+1,J)-VBAR(I+1))*(E0-ENERG(I+1,J))/(R(I)-R(I+1)))
TERM5=ONH*(W(I,J)*(ENERG(I,J+1)-E0)+W(I,J+1)*(E0-ENERG(I,J+1)))/
1 RTH
CVISC=CVI+RTH*(RJP-RJM)
S11=(V(I,J)-VBAR(I)-V(I-1,J)+VBAR(I-1))/(RJP-RJM)
S12=ONG*(V(I,J+1)+V(I-1,J+1)-V(I,J-1)-V(I-1,J-1))/RTH
S13=ONG*((W(I+1,J)-W(I,J))/(R(I+1)-W(I))+W(I,J)-W(I-1,J))/
1 (R(I)-R(I-1)))
S14=(W(I,J+1)-W(I,J-1))/RTH
TERM6=-CVISC*(TH*S11**2+(S12+S13)**2+TH*S14**2)
80 EXP(I,J-1)=ENERG(I,J)-DT*(TERM1+TERM2+TERM3+TERM4+TERM5+TERM6)
DO 90 J=2,NTH
90 FLUXI(J-1)=FLUXE(J-1)
100 FLUX1=FLUX2
RETURN
END

```

```

SUBROUTINE ESOL1 (NTH,INDEX)
COMMON / QUANT / RHO(50,6),STUF(1500),R(50),TH(6),STF1(300),
1 TEMP(50,6),STF2(800)
COMMON / EXTRA / P1,R1,R2,UTERM
COMMON / ZMATT / EXP(20,4),A(920)
COMMON / TPARM / Y(200),DY(200)
DATA FACHN,ONH,FR,ON / 1.E-14,5.E-1, 4.E+0, 1.E+0 /,OPF/1.5E+0 /
IF (INDEX.LE.0) RETURN
DO 50 I=1,INDEX
DO 50 J=1,NTH
T=TEMP(I,J+1)
E=EXP(I,J)-UTERM
P=7.5690-15/RHO(I,J+1)
DO 20 L=1,20
P1=(OPF*P1+P*T**3)*T
DT=(E*F1)/(OPF*P1+FR*P*T**3)
T1=T+DT
FACT=ABS(T1/T-ON)
IF (FACT.LT,FACHN) GO TO 30
T=T1
20 CONTINUE
WRITE (6,1) T1,T
1 FORMAT (25H CONVERGENCE FAILED, T1=E14,6,4H T=E14,6)
30 K=NTH*(I-1)+J
50 Y(K)=T1
RETURN
END

```

```

SUBROUTINE RADV (INDEX)
COMMON / QUANT / STF(1000),R(50),TH(6),GARB(1400)
COMMON / RADIUS / RNEW(50),VBAR(50)
DATA P45,P55 / 4.5E-1, 5.5E-1 /
GO TO (10,20) ,INDEX
10 CONTINUE
DO 15 J=2,50
15 R(J)=P45*R(J)+P55*RNEW(J)
RETURN
20 CONTINUE
DO 25 J=2,50
25 R(J)=RNEW(J)
RETURN
END

```


IV. IMPLICIT SOLUTION OF THE ENERGY EQUATION

The implicit solution of the energy equation near the surface is outlined here. Subroutine DPSET prepares terms for the matrix solution and checks to see if a converged solution has been obtained. The matrix manipulations are done by DP3, using matrix coefficients generated by COEF and COEF1. The theory of the solution is that of the Henyey method¹³ and has been outlined previously for this case,² but will be reviewed here.

A. DPSET

First find the maximum and minimum values of the radial zone numbers of the surface (N_2 and N_1 , respectively). Set the initial guess for the temperature at the advanced time (Y) equal to the temperature at the present time (T) for all zones interior to the minimum radial zone of the surface (20). Do the same for the other zones, checking first to see if there is any mass in the zone (18). If the model has contracted out of a zone, the temperatures in the two zones closest to the surface are modified (30). Because they are needed for more than one zone, the new pressures (PF), internal energies (EE) and their derivatives with respect to the temperature at the new time (DPT , DET), the weighted opacities (see Sec. VIII) at $(i+1/2, j)$ (OP) and its derivatives with respect to the new temperature at (i, j) and at $(i+1, j)$ are stored. For example, $EE(1) = E_{i-1, j}^{n+1}, \dots, EE(N_\theta+1) = E_{1, j}^{n+1}, \dots$, and $EE(2N_\theta+1) = E_{i+1, j}^{n+1}$ are used in the generation of the matrix coefficients interior to the surface zones ($< N_1$), and the first set is generated (45).

The matrix solution subroutine (DP3), which also calls the subroutines to generate the matrix coefficients, is itself called. Upon return, various tests are performed concerning convergence. The results of these tests are returned from DP3: an integer indicating whether or not convergence has been obtained (IP), the maximum temperature correction ($SUMNN$), and the maximum ratio of the temperature correction to the temperature ($SUMNM$). The iteration procedure is stopped if the maximum temperature correction becomes too large, if the corrections become larger with each iteration after the first three, or if the number of iterations (ICT) is too large. Conversion to a solution has been obtained if $IP = NTHZ$. Once conversion has been obtained, the "local luminosities," defined by $L_{i+1/2, j} = 4\pi r_{i+1/2}^2 F_{r_{i+1/2, j}}$ are computed (120). The old temperature values are replaced by the new values (40,70) or by the value at half time steps if the difference is too large (100). This latter step is taken to remove oscil-

lating solutions, which were once a problem but have not been recently. If convergence has not been obtained, the new temperatures are taken as a weighted value of the old and new temperatures, with more weight being given to the old temperatures as the degree of nonconvergence increases (80,90). If the maximum value of the temperature correction is too large, execution is terminated (by IOU_T).

B. DP3

The matrix used in the Henyey method¹³ is generated and reduced. The major input parameters are the number of rows having both r and θ components ($L1 = N1 \times N_\theta$), $N_\theta(N)$, the width of the band in the matrix ($M = 2N_\theta + 1$), and the radial zone number through which the explicit integration of the energy equation in the interior has gone (IMINN). I shall first outline the mathematical problem and then discuss the solution in terms of the subroutine. Interior to the stellar surface, the difference equations of the second-order differential equation in two dimensions can be expressed as

$$f_k(T_{i-1,j}, T_{i,j-1}, T_{i,j}, T_{i,j+1}, T_{i+1,j}) = 0 \quad , \quad (46)$$

where k is a function of i and j and denotes the row of the matrix. The linearization of the difference equation leads to

$$C_1 \Delta T_{i-1,j} + C_2 \Delta T_{i,j-1} + C_3 \Delta T_{i,j} + C_4 \Delta T_{i,j+1} + C_5 \Delta T_{i+1,j} + D = 0 \quad (47)$$

for all (i,j) . If the zones are numbered by

$$k = N_\theta(i-1) + j - 1 \quad (48)$$

as $2 \leq j \leq N_\theta + 1$, Eq. (47) would become

$$C_{k,1} \Delta T_{k-n} + C_{k,n} \Delta T_{k-1} + C_{k,n+1} \Delta T_k + C_{k,n+2} \Delta T_{k+1} + C_{k,2n+1} \Delta T_{k+n} + F_k = 0 \quad (49)$$

where $INMIN + 1 \leq k \leq (N1 - 1) \times N_{\theta}$ and $n = N_{\theta}$. Obviously, if $j = 2$, $C_{k,n} = 0$ and $C_{k,n+2} = 0$ if $j = n$. In the zones at the surface of the model (radial zone number $\geq N1$) we assume there is no horizontal motion or energy flow; hence

$$C_{k,1} \Delta T_{k-n} + C_{k,n+1} \Delta T_k + C_{k,2n+1} \Delta T_{k+n} + F_k = 0. \quad (50)$$

For those zones on the surface [$i = NN(j)$],

$$C_{k,1} \Delta T_{k-n} + C_{k,n+1} \Delta T_k + F_k = 0. \quad (51)$$

To insure that the treatment at the surface is understandable, I shall present an example. Let $N_{\theta} = 2$, $NN(2) = N1$ and $NN(3) = N2 = N1 + 1$. The last few rows of the matrix would be

i, j	$N1-1, 2$	$N1-1, 3$	$N1, 2$	$N1, 3$	$N2, 2$	$N2, 3$	
$N1-1, 2$	x	0	X	x	x		x 0 X x x
$N1-1, 3$	x	x	X	0	x		x x X 0 x
$N1, 2$	x	0	X	0	0		x 0 1 0
$N1, 3$	x	0	X	0	x		x 0 1 0
$N2, 2$			0	0	0	0	0 0 0 0
$N2, 3$			x	0	X		x 0 1

$\xrightarrow{\text{reduce}}$

where X denotes diagonal elements and x other nonzero elements. The row with all zeros looks unpleasant from a theoretical viewpoint, but it merely represents the fact that $\Delta T_{N2,2} = 0$ and presents no problem in practice. The last four rows can be reduced to the second configuration shown. All other rows can now be row-reduced without referring to the last two rows. Also, the reduction for the last four rows has not involved the next to last row. The situation above also reveals that for the last four rows

$$\Delta T_k = \sum_{\ell=1}^n \alpha_{k,\ell} \Delta T_{k-\ell} + \beta_k \quad (52)$$

It is also clear that if each of the first two rows in the illustration above is reduced using the two (generally n) rows immediately above that row so that there are no terms to the right of the diagonal, the form of Eq. (52) will result.

In general, we eliminate $C_{k,2n+1}$ by

$$\left. \begin{aligned} C_{k,m}^1 &= C_{k,m} + C_{k,2n+1} \alpha_{k+n,m} & m &= n+1, \dots, 2n \\ C_{k,m}^1 &= C_{k,m} & m &= 1, \dots, n \\ F_k^1 &= F_k + C_{k,2n+1} \beta_{k+n} \end{aligned} \right\} \quad (53)$$

Similarly, $C_{k,2n}^1$ can be eliminated by

$$\left. \begin{aligned} C_{k,m}^2 &= C_{k,m}^1 + C_{k,2n}^1 \alpha_{k+n-1,m} & m &= n, \dots, 2n-1 \\ C_{k,m}^2 &= C_{k,m}^1 & m &= 1, \dots, n-1 \\ F_k^2 &= F_k^1 + C_{k,2n}^1 \beta_{k+n-1} \end{aligned} \right\} \quad (54)$$

Continuation of the process will eventually produce

$$C_1^n \Delta T_{k-n} + C_2^n \Delta T_{k-n+1} + \dots + C_{n+1}^n \Delta T_k + F^n = 0, \quad (55)$$

whence

$$\left. \begin{aligned} \alpha_{k,m} &= -\frac{C_{n+1-m}^n}{C_{n+1}^n} & m &= 1, \dots, n \\ \beta_k &= -\frac{F^n}{C_{n+1}^n} \end{aligned} \right\} \quad (56)$$

The process is performed down to radial zone number IMINN, where $\Delta T = 0$, since the values for the new temperature for this and all interior radial zones are known (from ESOL and ESOL1). This determines the value $\Delta T_{IMINN+1,2}$ from Eq. (52) and all other $\Delta T_{1,j}$ as well.

In DP3 the primary variables represent the present guess for the temperature at the advanced time (Y), the linearized temperature perturbation $\Delta T(DY)$, $\alpha_{k,m}[A(K,M), M = 1, N_\theta]$, and $\beta_k[A(K, N_\theta + 1)]$. First, all $\alpha_{k,m}$ and β_k are set to zero for all k exterior to N1 (640). For the surface zone for each angle, the first call to COEF is made. The coefficients of Eq. (51) -- $C_{k,1}$ (C2), $C_{k,n+1}$ (C3), and F_k (C4) -- are returned, and the definition of $\alpha_{k,m}$ and β_k from Eq. (52) implies

$$\begin{aligned}\alpha_{k,n} &= - C2/C3 , \\ \beta_k &= - C4/C3 .\end{aligned}\tag{57}$$

The other call to COEF is made to generate the coefficients of Eq. (50), if necessary: $C_{k,1}$ (C1), $C_{k,n+1}$ (C2), $C_{k,2n+1}$ (C3), and F_k (C4). There are no angular terms, so the reduction is simply

$$\begin{aligned}\alpha_{k,n} &= - C_{k,1} / (C_{k,n+1} + C_{k,2n+1} \alpha_{k+n,n}) , \\ \beta_k &= - (F_k + \beta_{k+n} C_{k,2n+1}) / (C_{k,n+1} + C_{k,2n+1} \alpha_{k+n,n}) .\end{aligned}\tag{58}$$

All rows of the matrix down to radial zone N1-1 are computed in this fashion (500).

Next, the general matrix reduction is carried out for all rows of the matrix between the radial zone where the explicit integration stops and radial zone N1 - 1 (matrix rows INDEX1 through K2 - Do 100 loop). The matrix coefficients for a row are computed by COEF1 and the reductions given in Eq. (53) are carried out (30). The process is repeated N_θ times (40), resulting in Eq. (55). Equation (56) generates $\alpha_{k,m}$ and β_k (50). A test is performed to see that $|\alpha_{k,m}|$ is not too small, thereby preventing underflows. When all rows have been found, Eq. (52) is solved for all ΔT_k (130), using the boundary conditions that $\Delta T_k = 0$ for all k where the energy equation has been explicitly integrated. The maximum values of $|\Delta T/T|$ and $|\Delta T|$ (SUM1 and SUM2) and the value of k for which each

occurs (ITRB and ITRB1) are found and printed (FORMAT 1). The values of DY are multiplied by a certain factor (FAC) so that the corrections are not too large ($\Delta T1 < FATT * T$) and are then added to the current temperature guesses (260). Convergence is considered obtained when the maximum value of $|\Delta T|$ is less than a certain amount (DTM).

C. COEF1

Before discussing the matrix coefficient generation subroutines, I shall linearize the energy equation with respect to the temperature. The energy equation is

$$\frac{\partial E}{\partial t} + (v - v_0) \frac{\partial E}{\partial r} + \frac{w}{r} \frac{\partial E}{\partial \theta} + \frac{P}{\rho} \nabla \cdot \nabla - P^t : \nabla \nabla = - \frac{1}{\rho} \nabla \cdot \left(\frac{1}{\kappa \rho} \nabla T^4 \right), \quad (59)$$

where the tensor, P^t , is the stress tensor with the thermal pressure removed and has coefficients as given in Ref. 14. The tensor expression (viscous work) may be written (Ref. 15, Eq. 9.14) for rectangular coordinates

$$P^t : \nabla \nabla = \sum_{i,j} P_{i,j} \frac{\partial v_i}{\partial x_j}. \quad (60)$$

In our case this becomes

$$P^t : \nabla \nabla = v \left\{ 2 \left[\frac{\partial (v - v_0)}{\partial r} \right]^2 + \left[\frac{\partial (v - v_0)}{r \partial \theta} + \frac{\partial w}{\partial r} \right]^2 + 2 \left(\frac{\partial w}{r \partial \theta} \right)^2 \right\}, \quad (61)$$

where the curvature terms have been neglected. As previously stated, this has proved a good approximation in all cases so far. The difference equation becomes

$$\begin{aligned} & \frac{E_{i,j}^{n+1} - E_{i,j}^n}{\Delta t} + \frac{1-\alpha}{2} \left[\left(v_{i+1/2,j} - v_{0,i+1/2} \right) \frac{E_{i+1,j} - E_{i,j}}{r_{i+1} - r_i} + \left(v_{i-1/2,j} - v_{0,i-1/2} \right) \right. \\ & \quad \times \frac{E_{i,j} - E_{i-1,j}}{r_i - r_{i-1}} + w_{i,j+1/2} \frac{E_{i,j+1} - E_{i,j}}{r_i \Delta \theta} + w_{i,j-1/2} \frac{E_{i,j} - E_{i,j-1}}{r_i \Delta \theta} \left. \right] \\ & + \frac{\alpha}{4} \left[\left(v_{i+1/2,j} + v_{i-1/2,j} - v_{0,i+1/2} - v_{0,i-1/2} \right) \left(v_{i+1/2,j} + v_{i-1/2,j} - v_{0,i+1/2} - v_{0,i-1/2} \right) \right] \end{aligned}$$

$$\begin{aligned}
& \times \frac{E_{i+1,j} - E_{i,j}}{r_{i+1} - r_i} + \left(v_{i+1/2,j} + v_{i-1/2,j} - v_{0,i+1/2} - v_{0,i-1/2} + |v_{i+1/2,j} + v_{i-1/2,j} - v_{0,i+1/2} - v_{0,i-1/2}| \right) \\
& \times \frac{E_{i,j} - E_{i-1,j}}{r_i - r_{i-1}} + \left(w_{i,j+1/2} + w_{i,j-1/2} - |w_{i,j+1/2} + w_{i,j-1/2}| \right) \frac{E_{i,j+1} - E_{i,j}}{r_i \Delta\theta} \\
& + \left(w_{i,j+1/2} + w_{i,j-1/2} + |w_{i,j+1/2} + w_{i,j-1/2}| \right) \frac{E_{i,j} - E_{i,j-1}}{r_i \Delta\theta} \Big] \\
& + \frac{P_{ij}}{\rho_{ij}} \left[\frac{1}{r_i^2} \frac{(r_{i+1/2}^{v_{i+1/2,j}} r_{i-1/2}^{v_{i-1/2,j}})}{r_{i+1/2} - r_{i-1/2}} + \frac{1}{r_i \sin\theta_j \Delta\theta} \left(\sin\theta_{j+1/2} w_{i,j+1/2} \right. \right. \\
& \left. \left. - \sin\theta_{j-1/2} w_{i,j-1/2} \right) \right] - v \left[2 \left(\frac{v_{i+1/2,j} - v_{0,i+1/2} - v_{i-1/2,j} + v_{0,i-1/2}}{r_{i+1/2} - r_{i-1/2}} \right)^2 \right. \\
& \left. + \left(\frac{v_{i+1/2,j} + v_{i-1/2,j} - v_{i+1/2,j-1} - v_{i-1/2,j-1}}{2 r_i \Delta\theta} + \frac{v_{i+1,j+1/2} - w_{i,j+1/2}}{2(r_{i+1} - r_i)} + \frac{v_{i,j-1/2} - w_{i-1,j-1/2}}{2(r_i - r_{i-1})} \right)^2 \right. \\
& \left. + 2 \left(\frac{w_{i,j+1/2} - w_{i,j-1/2}}{r_i \Delta\theta} \right)^2 \right] - \frac{4\sigma}{3 r_i^2 \rho_{i,j}} \\
& \times \left[\frac{r_{i+1/2}^2 \left(\frac{1}{\kappa\rho} \right)_{i+1/2,j}}{r_{i+1/2} - r_{i-1/2}} \frac{T_{i+1,j}^4 - T_{i,j}^4}{r_{i+1} - r_i} - r_{i-1/2}^2 \left(\frac{1}{\kappa\rho} \right)_{i-1/2,j} \frac{T_{i,j}^4 - T_{i-1,j}^4}{r_i - r_{i-1}} \right. \\
& \left. + \frac{1}{\sin\theta_j (\Delta\theta)^2} \left\{ \sin\theta_{j+1/2} \left(\frac{1}{\kappa\rho} \right)_{i,j+1/2} \left(T_{i,j+1}^4 - T_{i,j}^4 \right) - \sin\theta_{j-1/2} \left(T_{i,j}^4 - T_{i,j-1}^4 \right) \right\} \right] = 0
\end{aligned} \tag{62}$$

All quantities except those in the first term are at time $(n + 1/2)$. I feared that stability problems might arise if the exact half time step were used in the form of oscillating solutions,¹⁶ so I have offset the half time slightly, e.g.,

$$E_{i,j}^{n+1/2} = P45 \times E_{i,j}^n + P55 \times E_{i,j}^{n+1}, \tag{63}$$

and have done similarly for the other variables. The values of P45 and P55 are defined in DATA statements in subroutines RHOSOL, RADV, DPSET, COEF, and COEF1. Of course the values in all subroutines must be identical. Clearly, $P55 = 1 - P45$, but the exact values can be changed to suit individual taste. Current values are $P45 = 0.45$ and $P55 = 0.55$.

Equation (62) must now be linearized with respect to the temperature at the advanced time.

$$\begin{aligned}
C_{k,1} = & \frac{-P55}{2(r_i - r_{i-1})} \left(\frac{\partial E}{\partial T} \right)_{i-1,j} \left\{ (1-\alpha) (v_{i-1/2,j} - v_{0_{i-1/2}})^{\frac{\alpha}{2}} (v_{i+1/2,j} + v_{i-1/2,j} \right. \\
& \left. - v_{0_{i+1/2}} - v_{0_{i-1/2}} + |v_{i+1/2,j} + v_{i-1/2,j} - v_{0_{i+1/2}} - v_{0_{i-1/2}}|) \right\} \\
& + \frac{4\sigma P55}{3r_i^2 \rho_{i,j}} \left[-4 \left(\frac{1}{\kappa \rho} \right)_{i-1/2,j} T_{i-1,j}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i-1/2,j} \right\}_{i-1,j} \left(T_{i,j}^4 - T_{i-1,j}^4 \right) \right] \\
& \times \frac{r_{i-1/2}^2}{(r_{i+1/2} - r_{i-1/2})(r_i - r_{i-1})}, \quad (64)
\end{aligned}$$

$$\begin{aligned}
C_{k,2n+i} = & \frac{P55}{2(r_{i+1} - r_i)} \left(\frac{\partial E}{\partial T} \right)_{i+1,j} \left\{ (1-\alpha) (v_{i+1/2,j} - v_{0_{i+1/2}})^{\frac{\alpha}{2}} (v_{i+1/2,j} + v_{i-1/2,j} - v_{0_{i+1/2}} \right. \\
& \left. - v_{0_{i-1/2}} - |v_{i+1/2,j} + v_{i-1/2,j} - v_{0_{i+1/2}} + v_{0_{i-1/2}}|) \right\} - \frac{4\sigma P55}{3r_i^2 \rho_{i,j}} \\
& \times \left[4 \left(\frac{1}{\kappa \rho} \right)_{i+1/2,j} T_{i+1,j}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i+1/2,j} \right\}_{i+1,j} \left(T_{i+1,j}^4 - T_{i,j}^4 \right) \right], \quad (65)
\end{aligned}$$

$$\begin{aligned}
C_{k,n+2} = & \frac{P55}{2r_i \Delta \theta} \left(\frac{\partial E}{\partial T} \right)_{i,j+1} \left\{ (1-\alpha) w_{i,j+1/2}^{\frac{\alpha}{2}} (w_{i,j+1/2} + w_{i,j-1/2} - |w_{i,j+1/2} + w_{i,j-1/2}|) \right\} \\
& - \frac{4\sigma P55 \sin \theta_{j+1/2}}{3\rho_{i,j} r_i^2 \sin \theta_j (\Delta \theta)^2} \left[4 \left(\frac{1}{\kappa \rho} \right)_{i,j+1/2} T_{i,j+1}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i,j+1/2} \right\}_{i,j+1} \left(T_{i,j+1}^4 - T_{i,j}^4 \right) \right], \quad (66)
\end{aligned}$$

$$\begin{aligned}
C_{k,n} = & \frac{-P55}{2r_i \Delta \theta} \left(\frac{\partial E}{\partial T} \right)_{i,j-1} \left\{ (1-\alpha) w_{i,j-1/2}^{\frac{\alpha}{2}} (w_{i,j+1/2} + w_{i,j-1/2} - |w_{i,j+1/2} + w_{i,j-1/2}|) \right\} \\
& + \frac{4\sigma P55 \sin \theta_{j-1/2}}{3\rho_{i,j} r_i^2 \sin \theta_j (\Delta \theta)^2} \left[-4 \left(\frac{1}{\kappa \rho} \right)_{i,j-1/2} T_{i,j-1}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i,j-1/2} \right\}_{i,j-1} \left(T_{i,j}^4 - T_{i,j-1}^4 \right) \right], \quad (67)
\end{aligned}$$

$$\begin{aligned}
C_{k,n+1} = & \frac{1}{\Delta T} \left(\frac{\partial E}{\partial T} \right)_{i,j} + \frac{P55}{\rho_{i,j}} \left(\frac{\partial P}{\partial T} \right)_{i,j} \left[\frac{r_{i+1/2}^2 v_{i+1/2,j} - r_{i-1/2}^2 v_{i-1/2,j}}{r_i^2 (r_{i+1/2} - r_{i-1/2})} + \frac{1}{r_i \sin \theta_j \Delta \theta} \right. \\
& \times \left(\sin \theta_{j+1/2} w_{i,j+1/2} - \sin \theta_{j-1/2} w_{i,j-1/2} \right) \left. + \frac{P55}{2} \left(\frac{\partial E}{\partial T} \right)_{i,j} \left(1 - \alpha \right) \right. \\
& \times \left(\frac{v_{i+1/2,j} - v_{i-1/2,j}}{r_i - r_{i-1}} - \frac{v_{i+1/2,j} - v_{i-1/2,j}}{r_{i+1} - r_i} \right) + \frac{\alpha}{2} \\
& \times \left(\frac{v_{i+1/2,j} + v_{i-1/2,j} - v_{i+1/2,j} - v_{i-1/2,j}}{r_i - r_{i-1}} + \frac{v_{i+1/2,j} + v_{i-1/2,j} - v_{i+1/2,j} - v_{i-1/2,j}}{r_{i+1} - r_i} \right) \left. \right] \\
& - \frac{v_{i+1/2,j} + v_{i-1/2,j} - v_{i+1/2,j} - v_{i-1/2,j}}{r_{i+1} - r_i} \left. \right) \\
& + \frac{P55}{2 r_i \Delta \theta} \left(\frac{\partial E}{\partial T} \right)_{i,j} \left\{ \left(1 - \alpha \right) \left(w_{i,j+1/2} - w_{i,j-1/2} \right) + \frac{\alpha}{2} |w_{i,j+1/2} + w_{i,j-1/2}| \right\} \\
& + \frac{4 \sigma P55 \sin \theta_{j-1/2}}{3 r_i^2 \rho_{i,j} \sin \theta_j (\Delta \theta)^2} \left[4 \left(\frac{1}{\kappa \rho} \right)_{i,j-1/2} T_{i,j}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i,j-1/2} \right\}_{i,j} \left(T_{i,j}^4 - T_{i,j-1}^4 \right) \right] \\
& - \frac{4 \sigma P55 \sin \theta_{j-1/2}}{3 r_i^2 \rho_{i,j} \sin \theta_j (\Delta \theta)^2} \left[-4 \left(\frac{1}{\kappa \rho} \right)_{i,j+1/2} T_{i,j}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i,j+1/2} \right\}_{i,j} \left(T_{i,j+1}^4 - T_{i,j}^4 \right) \right] \\
& - \frac{4 \sigma P55}{3 r_i^2 \rho_{i,j}} \left[-4 \left(\frac{1}{\kappa \rho} \right)_{i+1/2,j} T_{i,j}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i+1/2,j} \right\}_{i,j} \left(T_{i+1,j}^4 - T_{i,j}^4 \right) \right] \frac{r_{i+1/2}^2}{(r_{i+1/2} - r_{i-1/2})(r_{i+1} - r_i)} \\
& + \frac{4 \sigma P55}{3 r_i^2 \rho_{i,j}} \left[4 \left(\frac{1}{\kappa \rho} \right)_{i-1/2,j} T_{i,j}^3 + \left\{ \frac{\partial}{\partial T} \left(\frac{1}{\kappa \rho} \right)_{i-1/2,j} \right\}_{i,j} \left(T_{i,j}^4 - T_{i-1,j}^4 \right) \right] \frac{r_{i-1/2}^2}{(r_{i+1/2} - r_{i-1/2})(r_i - r_{i-1})}.
\end{aligned} \tag{68}$$

In COEF1, Eqs. (62) and (64)-(68) are evaluated. As is usual with computer programs, factors disappear and reappear freely in different groupings. With the difference equations given above, the discussion of COEF1 will be limited to the order in which calculations are made rather than outlining the individual computations. First, the variables at the half time step [Eq. (63)] are expressed as simple variables. The radial index (J) and angular zone index (L)

are used for these simple variables, e.g., TJM1 is $T_{j-1,l}$. Various other often-used quantities are defined. The stored internal energies, pressures, opacities, and all derivatives are incremented for the new zone (10 and $E_{i-1,j}$ and $P_{i-1,j}$ are found. The viscous pressure is added. The matrix coefficients (C) are all set to zero (20). The term $\frac{\partial E}{\partial t}$ (TERM3), then the centered and donor cell terms of $(v - v_0) \frac{\partial E}{\partial r}$ (TERM4) and $w \frac{\partial E}{r \partial \theta}$ (TERM5), and the derivatives with respect to the new temperatures are computed. The possible coefficients are: $C_{k,1}[C(1)]$, $C_{k,n}[C(NTH)]$, $C_{k,n+1}[C(N1)]$, $C_{k,n+2}[C(N2)]$, and $C_{k,2n+1}[C(J1)]$. The viscous work (TERM10), for which there are no temperature derivatives, is calculated. Next come the radiative fluxes: $F_{r_{i+1/2,j}}$ (TERM6), $F_{r_{i-1/2,j}}$ (TERM7), $F_{\theta_{i,j+1/2}}$ (TERM8), and $F_{\theta_{i,j-1/2}}$ (TERM9), and the appropriate derivatives. Those terms which are zero on the angular boundaries are not computed. The term $\nabla \cdot \nabla$ (R1) is calculated and all terms added (F). Zero values are insured for the angular boundary conditions.

D. COEF

The matrix elements for radial zone numbers $\geq N1$ are evaluated, with those zones at the very surface being computed before statement (100). The determination of which set of calculations is carried out is made by an index (IAID), which is zero at the very surface. For this case the equations are relatively simple.

$$\text{TERM2} = \frac{P}{\rho r^2} \frac{\partial}{\partial r}(r^2 v) \rightarrow \frac{P_{ij}}{\rho_{ij} r_i^2} \frac{r_{i+1/2}^2 v_{i+1/2} - r_{i-1/2}^2 v_{i-1/2}}{r_{i+1/2} - r_{i-1/2}}, \quad (69)$$

$$\text{TERM3} = \frac{\partial E}{\partial t} = \frac{E_{i,j}^{n+1} - E_{i,j}^n}{\Delta t}, \quad (70)$$

$$\begin{aligned} \text{TERM4} = (v - v_0) \frac{\partial E}{\partial r} \rightarrow \frac{1}{2} \left\{ \left(v_{i+1/2,j} - v_{0_{i+1/2}} \right) \frac{E_{i+1,j} - E_{i,j}}{r_{i+1} - r_i} \right. \\ \left. + \left(v_{i-1/2,j} - v_{0_{i-1/2}} \right) \frac{E_{i,j} - E_{i-1,j}}{r_i - r_{i-1}} \right\}, \end{aligned} \quad (71)$$

$$\text{TERM5} = \frac{1}{\rho r^2} \frac{\partial}{\partial r}(r^2 F_r) \rightarrow \frac{\sigma}{\rho_{i,j} r_i^2} \left[\frac{2T_{i,j}^4 r_{i+1/2}^2 + \frac{4}{3} \left(\frac{1}{\kappa \rho} \right)_{i-1/2,j} \frac{T_{i,j}^4 - T_{i-1,j}^4}{r_i - r_{i-1}} r_{i-1/2}^2}{r_{i+1/2} - r_{i-1/2}} \right] \quad (72)$$

The variables $C_{k,1}$ (C2), $C_{k,n+1}$ (C3), and F_k (C4) of Eq. (51) are returned.

For the other case, which has not occurred in runs of the program in its present form, the terms are only slightly more complicated.

$$\begin{aligned} \text{TERM1} &= \frac{\partial E}{\partial t} \rightarrow \frac{E_{i,j}^{n+1} - E_{i,j}^n}{\Delta t} \\ \text{TERM3} &= \frac{P}{\rho r^2} \frac{\partial}{\partial r} (r^2 v) \rightarrow \frac{r_{i,j}}{\rho_{i,j} r_i} \left(\frac{r_{i+1/2}^2 v_{i+1/2,j} - r_{i-1/2}^2 v_{i-1/2,j}}{r_{i+1/2} - r_{i-1/2}} \right). \end{aligned} \quad (73)$$

The energy derivative has been omitted because at the time the program was written, it caused numerical difficulties. I would expect this to disappear with the present moving coordinate system form, but also would expect it to be small because the convective velocity does not penetrate into the top, optically thin zones. However, the energy derivative should be added for consistency. The radial flux term (TERM4) is the same as in Eq. (62). The variables returned are $C_{k,1}$ (C1), $C_{k,n+1}$ (C2), $C_{k,2n+1}$ (C3), and F_k (C4) of Eq. (50).

```

SUBROUTINE DPSET (IOUT,IMINN,NOTE)
COMMON / IPERT / IPERTM
COMMON / QUANT / RHO(50,6,2),CRAP(300),V(50,6),CRUD(300),w(50,6)
1  ,R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),DFLAD(50,6),ZL(50,-)
COMMON /TPARM / Y(200),DY(50,4)
COMMON /ZMAT / A(200,5)
COMMON/STF1/EL(20),PP(20),DET(20),DPT(20),DP(20),DOP(20),DOM(20)
COMMON /MUDPR/ NTH,NTHZ,MAXTH,MZONE,MAXZ
COMMON / SFACL / NX(10),N1,N2,NSAP
DIMENSION NOTE (6)
DATA ZO,ONH,R*OMH, P45,P55,OPF,TWO42,ON,ZNUM / 0.E+0,
1  5.E-1,1.E-11,4.5E-1,5.5E-1, 1.5E+0,2.E-2,1.E+0,0,40R40269E+0 /
N1=NN(2)
N2=NN(2)
DO 10 L=2,NTH
IF (NN(L),LT,N1) N1=NN(L)
IF (NN(L),GT,N2) N2=NN(L)
10 CONTINUE
WRITE (6,1) N1,N2
1  FORMAT (2X,3HN1=,15,2X,3HN2=,15)
SUMAP=1,0+6
1(LT=0
IWH0=IMINN+1
DO 20 I=IWH0,N1
DO 20 J=2,NTH
K=NTHZ*(I-1)+J-1
20 Y(K)=T(I,J)
C CASE OF EXPANSION OR CONTRACTION INTO A NEW ZONE
DO 18 J=2,NTH
DO 18 I=N1,N2
IF (RHO(I,J,2),LE,ZO) GO TO 18
IF (T(I,J),LE,ZO) T(I,J)=1(I=1,J)
K=NTHZ*(J-1)+J-1
Y(K)=T(I,J)
18 CONTINUE
C CONTRACTION
DO 30 L=2,NTH
IF (NOTE(L),NE,1) GO TO 30
K=NN(L)-2
K2=NN(L)
DO 25 K1=K,K2
25 T(K1,L)=ONH*(T(K1,L)+T(K1+1,L))
T(K2,L)=T(K2+1,L)
30 CONTINUE
35 CONTINUE
N1M=N1-1
DO 45 J=N1M,N1
DO 45 L=2,NTH
L2=NTHZ*(J-N1M)+L-1
L1=NTHZ*(J-1)+L-1
RHO1NE=RHO(J,L,1)
IF (RHO1NE,LE,ZO) RHO1NE=RHOMN
CALL THERM (RHO1NE,Y(L1),P1,EE(L2),DET(L2),DPT(L2),D1,D2,C3,2)
PP(L2)=P1+VISC(RHO(J,L,2),P1,V(J,L),V(J+1,L),w(J,L),w(J,L+1))
IF (J,EQ,N1) GO TO 45
L3=L1+NTHZ
TJ=P45*T(J+1,L)+P55*Y(L3)

```

```

      T1=P45*T(J,L)+P55*Y(L)
      RHOJ=RHO(J+1,L,2)
      RHOZ=RHO(J,L,2)
      CALL OPACIT (IJ,T0,RHOJ,RHOZ,OP(L2),DOP(L2),DOM(L2),1)
45  CONTINUE
      K=N1+NTNZ
      M=2+NTNZ+1
      IP=NTNZ
      CALL DP3 (K,NTNZ,M,IP,IMTNN,SUMNM,SUMNN)
      IF (SUMNN.GT.1.D+5) ICUT=1
      ICT=ICT+1
      IF (ICT.EQ.1) TPERTM=SUMNM
      IF (SUMNN.GE.SUMKP.AND.IC1.GE.3) GO TO 75
      SUMKP=SUMNN
      IF (ICT.GT. 8) GO TO 50
      IF (IP.NE.NTNZ) GO TO 35
50  CONTINUE
      IF (SUMNM.LT.1.D+2) ICUT=0
      DO 120 L=2,NTN
      K=NN(L)
      DO 110 J=3,K
      I=NTNZ*(J-1)+L-1
      I1=1-NTNZ
      T1=P45 *T(J,L)+P55 *Y(I)
      T2=P55 *Y(I1)+P45 *T(J-1,L)
      CALL OPACIT (T1,T2,RHO(J,L,2),RHO(J-1,L,2),FK,DUM1,DUM2,-1)
      RJ=ONH*(R(J)+R(J-1))
110  ZL(J-1,L-1)=(ZNUM*FK*(T1**4-T2**4)/(R(J)+R(J-1)))*RJ**2
      RJ=ONH*(R(K)+R(K+1))
120  ZL(K,L-1)= ZNUM*OPF*T1**4*RJ**2
      DO 100 L=2,NTN
      L2=NN(L)
      L1=L2-5
      DO 100 K=L1,L2
      K1=NTNZ*(K-1)+L-1
      QU= ABS(Y(K1)/T(K,L)-1.)
      IF (QU.GT.1.D+1) Y(K1)=P45*T(K,L)+P55*Y(K1)
100  CONTINUE
      DO 40 I=1,N1
      DO 40 J=2,NTN
      K=NTNZ*(I-1)+J-1
40  T(I,J)=Y(K)
      DO 70 L=2,NTN
      K=NN(L)
      IF (N1.EQ.K) GO TO 70
      K1=N1+1
      DO 60 J=K1,K
      I=NTNZ*(J-1)+L-1
60  T(J,L)=Y(I)
70  CONTINUE
      RETURN
75  S1=TWDM2/SUMNM
      IF (S1.GT.ON) S1=ON
      S2=ON-S1
      DO 80 I=1,N1
      DO 80 J=2,NTN
      K=NTNZ*(I-1)+J-1
80  T(I,J)=S2*T(I,J)+S1*Y(K)

```

```

DO 90 L=2,NTH
K=NN(L)
IF (N1,E0,K) GO TO 92
K1=N1+1
DO 85 J=K1,K
I=N1HZ*(J-1)+L-1
85 T(J,L)=S2*T(J,L)+S1*Y(I)
90 CONTINUE
RETURN
END

```

```

SUBROUTINE OPS (L1,N,M,IP,INTINN,SUM1,SUM2)
COMMON / MELT / FLUX
COMMON / SFACE / NN(10),N1,N2,MSKF
COMMON / QUANT / RH0(50,6,2),CRAP(1200),R(50),TH(6),
Z P(50,6),I(50,6),ENERG(50,6),KAST(500)
COMMON /TPARM/Y(200),DY(200)
COMMON /ZHAIT / A(200,5)
DIMENSION C(17)
DATA Z0,AMINA,DTM,FATT,ON,THINNA /0.E+0,1,E+20,1,E+1,8,E-2,
1 1.E+0,3.E+3 /
INDEX=INTINN*N
INDEX1=INDEX+1
NM1=N-1
NP1=N+1
DO 640 L=1,N
DO 640 I1=N1,N2
I1=N*(I1-1)+L
DY(I1)=Z0
DO 640 I2=1,NP1
640 A(I1,I2)=Z0
DO 500 L=1,N
K=NN(L+1)
DO 400 I1=N1,K
I1=N1+K-I1
I1=N*(I1-1)+L
IF (I,NE,K) GO TO 375
CALL COEF (C1,C2,C3,C4,I1,N,0,I,L+1)
A(I1,N)=-C2/C3
A(I1,NP1)=-C4/C3
GO TO 400
375 CONTINUE
CALL COEF (C1,C2,C3,C4,I1,N,1,I,L+1)
I1PN=I1+N
DN=C2+C3*A(I1PN,N)
A(I1,N)=-C1/DN
A(I1,NP1)=-C4+A(I1PN,NP1)*C3/DN
400 CONTINUE
500 CONTINUE
K2=L1=N
DO 100 I1=INDEX1,K2
I1=INDEX1+K2-I1
CALL COEF1 (C,M,F,I,IP)
DO 40 JJ=1,N
J=M+1-JJ
IJ=I+NP1-JJ
DO 30 J1=1,N
J2=J-J1
30 C(J2)=C(J2)+C(J)*A(IJ,J1)
F=F+C(J)*A(IJ,NP1)

```

```

40 CONTINUE
   CC=C(NP1)
   DO 50 J=1,N
     NP1MJ=NP1+J
     A(I,J)=-C(NP1MJ)/CC
     AABS=ABS(A(I,J))
     IF (AABS,LE,AMINA) A(I,J)=Z0
50 CONTINUE
   A(I,NP1)=-F/CC
120 CONTINUE
   DO 115 I=1,INDEX
115 DY(I)=Z0
     SUM2=Z0
     SUM1=Z0
     ITRB=0
     ITRB1=0
     L2=N*N2
     DO 150 J=INDEX1,L2
       DY(I)=A(I,NP1)
       DO 130 J=1,N
         IMJ=I-J
130 DY(I)=DY(I)+A(I,J)*DY(IMJ)
         IF (Y(I),LE,Z0) GO TO 150
         SUM=ABS(DY(I)/Y(I))
         SUM3=ABS(DY(I))
         IF (SUM,GT,SUM1) ITRB=I
         IF (SUM,GT,SUM1) SUM1=SUM
         IF (SUM3,GT,SUM2) ITRB1=I
         IF (SUM3,GT,SUM2) SUM2=SUM3
150 CONTINUE
       WRITE (6,1) SUM1,SUM2,ITRB,ITRB1
       1 FORMAT (1X,E14,6,1X,E14,6,2X,I4,2X,I4)
       FAC=ON
       IF (SUM2,LT,OTM) GO TO 250
       IP=NTHZ-1
       FAC=FATT/SUM1
       IF (FAC,GT,ON) FAC=ON
250 CONTINUE
     DO 260 J=INDEX1,L2
       IF (Y(J),LE,Z0) GO TO 260
       Y(J)=Y(J)+FAC*DY(J)
       IF (Y(J),LE,TMINNA) Y(J)=TMINNA
260 CONTINUE
   RETURN
   END

```

```

SUBROUTINE COEF1 (C,J1,F,ICT,IP)
DIMENSION C(17)
COMMON / RADIUS / RVEA(52),VRAR(52)
COMMON / SFACL / NM(12),NN3,NN4,NSKP1
COMMON / FSJA / SINE(12),COSI(12),DLH
COMMON / STF1 / FL(20),PP(20),DET(20),DPT(20),OP(20),DOP(20),DOM(20)
COMMON / MODPR / N1,NTH,N2,NZUNE,MAXZ
COMMON / QUANT / RHU(52,6,2),CRAP(300),V(52,6),CRUD(300),A(52,6),
1 R(52),TH(61),P(52,6),T(52,6),ENERG(52,6),DELD(52,6),ZLUM(2,2)
COMMON / IPARM / Y(200),DY(200)
COMMON / ZMOD / X,Z,ZNSTAR,Z-CORE,DT
COMMON / VISCOS / VSIP,ALPHA1,ALPHA,VCCNV,TVISCO
DATA P45,P55,Z0,RHOMN,SG43,TW,RATL,RATU,ONH,FR,IPT,ON / 0.45,
A 0.55, 0., 1, F-11, 7.55R66667E-5, 2., 0.5, 1.75, 0.5, 4., 2, E+4,
0 J, /
ICT1=ICT+IP
ICT2=ICT-IP
J=(ICT-1)/NTH+1
L=ICT-NTH*(J-1)+1
TF=P45 *T(J,L)+P55 *Y(ICT)
TJM1=P45 *T(J-1,L)+P55 *Y(ICT2)
TJP1=P45 *T(J+1,L)+P55 *Y(ICT1)
RHOU=RHU(J,L,2)
RHOUJ1=RHU(J+1,L,2)
RHOUJ1=RHU(J-1,L,2)
RHOLJ1=RHU(J,L+1,2)
RHOLM1=RHU(J,L-1,2)
T03=T0**3
T04=T0**4
T1=T04-TJM1**4
T2=TJP1**4-T04
DR=R(J+1)-R(J-1)
DR1=R(J+1)-R(J)
DR2=R(J)-R(J-1)
RB1=(R(J+1)+R(J))**2/DR1/DR
RB2=(R(J)+R(J-1))**2/DR2/DR
S1=SINE(L)+SINE(L+1)
S2=SINE(L)+SINE(L-1)
J2=J1-1
DO 10 K=1,J2
K1=J1-K
EE(K1+1)=EE(K1)
PP(K1+1)=PP(K1)
DET(K1+1)=DET(K1)
DPT(K1+1)=DPT(K1)
OP(K1+1)=OP(K1)
DOP(K1+1)=DOP(K1)
DOM(K1+1)=DOM(K1)
10 CONTINUE
RHONE=RHU(J-1,L,1)
IF (RHONE,LE,Z0) RHONE=RHOMN
CALL THERM (RHONE,Y(ICT2),PP(1),EE(1),DET(1),DPT(1),D1,D2,D3,2)
VV=Z0
IF ((J-2),LE,0) GO TO 15
VV=V(J-2,L)
15 CONTINUE
PP(1)=PP(1)+VISC(RHOUJ1,PP(1),V(J-1,L),VV,X(J-1,L),W(J-1,L-1))
F1=SG43/R(J)**2/RHOU
DTH=TW*DLH
DO 20 L1=1,J1

```



```

C(L1)=Z0
20 CONTINUE
TERM2=P55 *PP(N1)+P45 *P(J,L)
CALL OPACIT (TJM1,TJ,RHOJH1,RHOL1,UP(1),DOM(1),DOP(1),1)
TERM3=(EE(N1)-ENERG(J,L))/DT
VV=V(J,L)-VBAR(J)
WW=Λ(J,L)
E1=P45 *ENERG(J+1,L)+P55 *EE(J1)
E2=P45 *ENERG(J,L)+P55 *EE(N1)
E3=P45 *ENERG(J-1,L)+P55 *EE(1)
DE1=(E1-E2)/DR1
DE2=(E2-E3)/DR2
VV1=V(J-1,L)-VBAR(J-1)
VAVE=ONH*(VV+VV1)
VABS1=ABS(VAVE)
TERM4=(UN-ALPHA)*ONH*(VV*DE1+VV1*DE2)+ALPHA*ONH*(DE1*(VAVE-
A VABS1)+DE2*(VAVE+VABS1))
C(1)=ONH*P55*DET(1)/DR2*(UN-ALPHA)*VV1+ALPHA*(VAVE+VABS1)
C(J1)=ONH*P55*DET(J1)/DR1*(ON-ALPHA)*VV+ALPHA*(VAVE-VABS1)
C(N1)=ONH*P55*DET(N1)*((ON-ALPHA)*(VV1/DR2-VV/DR1)+
A ALPHA*((VAVE+VABS1)/DR2-(VAVE-VABS1)/DR1))
WW1=W(J,L-1)
VV=V(J,L)
VV1=V(J-1,L)
DG=R(J)*DLH
E4=P45*ENERG(J,L+1)+P55*EE(N2)
E5=P45*ENERG(J,L-1)+P55*EE(NTH)
VAVE=ONH*(WW+WW1)
VABS1=ABS(VAVE)
TERM5=(UN-ALPHA)*ONH/DG*(WW*(E4-E2)+WW1*(E2-E5))+ALPHA*ONH/DG*
A ((E4-E2)*(VAVE-VABS1)+(E2-E5)*(VAVE+VABS1))
C(NTH)=-ONH/DG*P55*DET(NTH)*((ON-ALPHA)*WW1+ALPHA*(VAVE+VABS1))
C(N2)=ONH/DG*P55*DET(N2)*((UN-ALPHA)*WW+ALPHA*(VAVE-VABS1))
C(N1)=C(N1)+ONH/DG*P55*DET(N1)*((ON-ALPHA)*(WW1-WW)+
A ALPHA*TW*VABS1)
CVISC=Z0
IF (J.LT.(NN3-4)) CVISC=VSIP*DG*ONH*(DR1+DR2)
S11=(V(J,L1)-VBAR(J)-V(J-1,L)+VBAR(J-1))/DR
S12=ONH*(V(J,L+1)+V(J-1,L+1)-V(J,L-1)-V(J-1,L-1))/DTH/R(J)
S13=ONH*((W(J+1,L)-W(J,L))/DR1+(W(J,L)-W(J-1,L))/DR2)
S11=S11*TW
S14=(WW-WW1)/DLH/R(J)
TERM10=-CVISC*(W*S11**2+(S12+S13)**2+1**S14**2)
TERM6=OP(N1)*RB1*T2*ONH
TERM7=OP(1)*RB2*T1*ONH
IF (L,NE,2) GO TO 150
TERM9=Z0
GO TO 160
150 CONTINUE
TLM1=P45 *T(J,L-1)+P55 *Y(1CT=1)
CALL OPACIT (T,TLM1,RHO2,RHOL1,FK3,DOPAC1,DOPAC2,1)
T4=T04-TLM1**4
TERM9=FK3*S2/DLH/DTH*T4/SINE(L)
F2=P55*F1*S2/SINE(L)/DLH/DTH
C(NTH)=C(NTH)+F2*(-FR*FK3*TLM1**3+DOPAC2*T4)
C(N1)=C(N1)+F2*(FR*FK3*T4+DOPAC1*T4)
160 CONTINUE
IF (L,NE,N1) GO TO 170
TERM8=Z0
GO TO 180

```

```

170 TLP1=P45 *I(J,L+1)+P55 *Y(ICT+1)
    T3=TLP1**4-TP4
    CALL OPACIT (TLP1,T3,RHOLP1,RHOB,FK3,DOPAC1,DOPAC2,1)
    TERM1=FK3*AS1/DLM/DTH*13/SINE(L)
    F2=P55*F1*AS1/SINE(L)/DLM/DTH
    C(N2)=C(N2)+F2*(FK+FK3*TLP1**3+DOPAC1*T3)
    C(N1)=C(N1)+F2*(-FR*FK3*TD3+DOPAC2*T3)
180 CONTINUE
    R1=ONH*(RB1+DN1)*V(J,L)-RB2*DR2*V(J-1,L))/R(J)**2+ONH*(N(J,L)*
Z    S1=N(J,L-1)*S2)/R(J)/SINE(L)/DLM
    F=TERM3+TERM4+TERM5+TERM2/RHOB**K1-F1*(TERM6-TERM7+TERM8-TERM9)
    ) +TERM10
    C(N1)=C(N1)+DPT(N1)*P55 *R1/RHOB*DET(N1)/DT
    F2=ONH*P55*F1
    C(J1)=C(J1)+F2*RB1*(FR*OP(N1)*TJP1**3+DOP(N1)*T2)
    C(N1)=C(N1)+F2*RB1*(-FR*OP(N1)*TD3+DOM(N1)*12)+F2*RB2*(FR*OP(1)
1    *TD3+DOP(1)*11)
    C(1)=C(1)+F2*RB2*(-FR*OP(1)*TJM1**3+DOM(1)*11)
    IF (L,NE,2) GO TO 40
    C(N1)=C(N1)+C(NTH)
    C(NTH)=Z0
    RETURN
40 CONTINUE
    IF (L,NE,N1) RETURN
    C(N1)=C(N1)+C(N2)
    C(N2)=Z0
    RETURN
END

```

```

SUBROUTINE COEF (C1,C2,C3,C4,ICT,N,IAID,J,L)
COMMON / RADIUS / PNEA(50),VBAR(50)
COMMON / QUANT / RHO(50,6,2),CRAP(300),V(50,6),CRUD(300),W(50,6),
1 R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),WASTE(500)
COMMON / TPARM / Y(200),DY(200)
COMMON / ZMOD / X,Z,ZMSTAR,ZMCORE,DT
DATA P45,P55,Z0,RHOMN,DNH,TH,SGMA2,FR,TE,ET / 4.5E-1, 5.5E-1,
1 0.E+0,1.E-11,5.E-1,2.E+0, 1,1338E+4,4.E+0,3.E+0,8.E+0 /
ICTMN=ICT+N
ICTPN=ICT+N
T0=P45*T(J,L)+P55*Y(ICT)
TH1=P45 *T(J-1,L)+P55 *Y(ICTMN)
RHO0=RHO(J,L,2)
RHO1=RHO(J-1,L,2)
RHO1=RHO(J,L,1)
IF (RHO1.LE.70) RHO1=RHOMN
CALL TTERM (RHO1,Y(ICT),P0,E0,DET0,DPT0,Dn1,Dn2,Dn3,2)
DPT0=P55*DPT0
RHO2=RHO(J-1,L,1)
IF (RHO2.LE.Z0) RHO2=RHOMN
CALL TTERM (RHO2,Y(ICTMN),P1,E1,DET1,DPT1,Dn1,Dn2,Dn3,2)
DPT1=P55*DPT1
RJP=DNH*(R(J)+R(J+1))
RJM=DNH*(R(J)+R(J-1))
RB1=(RJP/R(J))**2
RB2=(RJM/R(J))**2
IF (IAID.NE.3) GO TO 100
P0=P0+VISC(RHOP,P0,V(J,L),V(J-1,L),Z0,Z0)
P1=P1+VISC(RHOM1,P1,V(J,L),V(J-1,L),Z0,Z0)
CALL CPACTT (T0,TH1,RHO0,RHO1,FK,DFK0,DFK1,1)
TERM3=(E0-ENERG(J,1))/DT
DR=R(J+1)-R(J-1)
VV=V(J,L)
IF (VV.E0.Z0) VV=V(J-1,L)
VT1=RB1*(VV-VBAR(J))/DR
VT2=RB2*(V(J-1,L)-VBAR(J-1))/DR
TERM4=(P45 *ENERG(J-1,L)+P55 *E1)*VT2*(P45 *ENERG(J,L)+P55 *E2)
1 *(VT1+VT2)+P45 *ENERG(J+1,L)*VT1
RJP=TH*(VT1+VT2)/RHOP
TERM2=R10*(P55 *P0+P45 *P(J,L))
CC=SGMA2/RHO2/DR
DR=R(J)-R(J-1)
TERM5=CC*(T0**4-T1**4)*RB1+FR/TE*FK*(T0**4-T1**4)*RB2/DR)
C3=DET0/DT+DPT0*RJP-P55*DET0*(VT1+VT2)
C4=TERM2+TERM3+TERM4+TERM5
C2=P55 *DET1*VT2
CC=P55*CC
STS3=FR*FR/TE
C3=C3+CC*(ET*T0**3*RB1+STS3 *FK*T0**3*RB2/DR+FR/TE*DFK0*
1 (T0**4-T1**4)/DR*RB2)
C2=C2+CC*(FR/TE*DFK1*(T0**4-T1**4)/DR+STS3 *FK*TH**3/DR)*R32
RETURN
100 CONTINUE
TP1=P45 *T(J+1,L)+P55 *Y(1CTPN)
RHOP1=RHO(J+1,L,2)
RHO3=RHO(J+1,L,1)
IF (RHO3.LE.Z0) RHO3=RHOMN

```

```

CALL THERM (RH03,Y(ICTPN),P2,E2,DET2,DPT2,Dn1,Dn2,Dn3,2)
P0=P0+VISC(RH02,P0,V(J,L),V(J-1,L),Z0,Z0)
P1=P1+VISC(RH01,P1,V(J-1,L),V(J-2,L),Z0,Z0)
P2=P2+VISC(RH01,P2,V(J+1,L),V(J,L),Z0,Z0)
DR=R(J+1)-R(J-1)
TERM1=(E2-ENERG(J,L))/DT
C1=Z0
C2=Z0
C3=Z0
C TEST
  TERM2=Z0
C END OF TEST
  RR=R(J)**2
  R10=(RB1*V(J,L)-RB2*V(J-1,L))/DR/RH00
  TERM3=R10*(P45 *P(J,L)+P55 *P0)
  C2=DET0/D1+DPT0*R10
  CST=SGMA2/TE/RH00/DR
  CALL OPACIT (T0,TP1,RH00,RH01,FK1,DFK1,DFK11,1)
  CALL OPACIT (T0,TM1,RH00,RH01,FK2,DFK2,DFK22,1)
  R21=(R(J)+R(J+1))**2
  R22=(R(J)+R(J-1))**2
  DR1=R(J+1)-R(J)
  DR2=R(J)-R(J-1)
  T11=TP1**4-T0**4
  T12=TM1**4-T0**4
  TERM4=CST*((R21/RR)*FK1*(T11/DR1)-(R22/RR)*FK2*(T12/DR2))
  C4=TERM1+TERM2+TERM3+TERM4
  CST=P55*CST
  C3=C3+CST*(R21/RR)*(FK1*FR*(TP1**3/DR1)+DFK11*(T11/DR1))
  C1=C1+CST*(R22/RR)*(FK2*FR*(TM1**3/DR2)+DFK22*(T12/DR2))
  C5=CST*(R21/RR)*(DFK1*(T11/DR1)-FR*FK1*(T0**3/DR1))
  C6=CST*(R22/RR)*(FK2*FR*(T0**3/DR2)+DFK2*(T12/DR2))
  C2=C2+C5-C6
1 FORMAT (1X,2I3,1P7D13,5,/,10X,4D13,5)
RETURN
END

```

V. REZONING

The process of rezoning was considered piecemeal as various problems arose. Consequently, there is more than one way of rezoning. I shall first consider the various criteria that lead to rezoning and then discuss the subroutines in which these steps are taken.

Because the radial zoning is variable, one must worry that the ratio $f_i = \Delta r_i / \Delta r_{i+1}$, where $\Delta r_i = r_{i+1} - r_i$, not become too different from unity. Define the quantity R as

$$R = \max_i (\text{the larger of } f_i \text{ and } f_i^{-1}). \quad (74)$$

R tends to be largest in the optically thin zones, where zoning inaccuracies are not serious because the properties of this region are essentially otherwise determined. Therefore, I have computed two values of R , one for the entire mesh and one for the zones near the surface. I have rezoned if $R > 5$ for regions away from the hydrogen ionization zone (usually the optically thin zones) and for $R > 20$ anywhere. Rezoning in the hydrogen ionization region should be avoided if possible because interpolations therein must be inaccurate owing to the steepness of various functions there. These values of R may seem excessively large and are rarely encountered in practice. The rezoning procedure here is to adjust r_i (and the other variables in zone i) at the zone of maximum R so that $\Delta r_i = \Delta r_{i+1}$ or Δr_{i-1} , whichever is more convenient.

The other situation in which rezoning is required occurs when the value of Δr_i drops below a certain (model-dependent) limit. A too low value of Δr_i violates the viscosity restriction on the time step outlined in Sec. II. Such a violation has always occurred very close to the surface, where the zoning is smaller anyway. This requires rezoning of a different type. I have deleted a radial zone where Δr_i is too small and inserted a radial zone in a deeper layer. The decision on whether to rezone is made in RSTEP and REZONE. It is carried out for the first method in INTRP, and for the second in REVAMP.

A. RSTEP

This subroutine first solves for the horizontal average of the radial velocity (VBAR) and integrates Eq. (5) for the radial coordinate (RNEW) at the advanced time step (60). It then computes the maximum value of the ratio R , previously defined, over the entire mesh (DRRAT) and for the zones near the

surface (DRTAU). "Close to the surface" is defined as those zones with temperatures below a certain value (TTEST). The radial zone numbers of DRRAT and DRTAU (JKP and JTAU, respectively) are also kept. One other function of this subroutine is to check to see that the minimum allowed radial zoning step (DRMIN) has not been violated. This is checked using the index ICHK. If a violation has occurred, ICHK is the radial zone number of the violation; otherwise, ICHK = -1. When a violation has occurred (GC TO 80), I try to correct the problem by rearranging the zone locally (INTRP). This will succeed if R is large enough and is determined by that zone. If the difficulty has not been removed within the next IRZNKP time steps, the second rezoning method is applied (REVAMP).

B. REZONE

Subroutine rezone is to take the values of DRRAT and DRTAU computed in RSTEP and decide if rezoning is necessary. First, test to see if JKP and JKP1 are the same. If not, rezoning is allowed near the surface unless JKP1 is outside the surface for some angular zone (> N1) or unless DRTAU is too small (< 1.6 in the current version). For the maximum ratio in the mesh, rezoning is required if R is too large (> RATMAX) and disallowed if too small (< DRALLW). Between these two extremes, I try to determine if the zone is too close to the hydrogen ionization region (DO 20 loop) by checking the temperature variation. If the temperature variation is not too steep, rezoning is allowed.

The two methods of rezoning are demonstrated in Fig. 3. Figure 3a indicates how INTRP rezones, and 3b how REVAMP rezones. By the time either of these subroutines has been reached, the decision to rezone has been made.

C. INTRP

Rezoning of the radial zone JKP is to be carried out according to the following.

$$r_{JKP} = 1/2(r_{JKP+1} + r_{JKP-1})$$

$$q_{JKP} = 1/2 q_{JKP} + 1/4(q_{JKP+1} + q_{JKP-1}) ,$$
(75)

where q represents any of the variables ρ , T, P, E, v, or w.

Fig. 3

An indication of the two methods of rezoning. Figure 3a demonstrates the procedure when the zones are not near an ionization region. When one zone has moved too far from the average position of the two neighboring zones, this zone is repositioned at this average position. In Figure 3b a zone above (but near) the hydrogen zone is deleted and a new zone defined below the hydrogen ionization zone.

D. REVAMP

In this subroutine I delete radial zone IDELET and insert a new radial zone at radial zone number NSERT (IDELET). For zones NSERT + 1 to IDELET, set

$$q_{i+1,j} = q_{i,j} , \quad (76)$$

where q is any variable. This deletes zone IDELET and leaves a vacancy at zone NSERT. Conditions for this new zone are generated by

$$q_{\text{NSERT},j} = 1/2(q_{\text{NSERT}+1,j} + q_{\text{NSERT}-1,j}) \quad (77)$$

and the equation of state. The value of NSERT is determined by the temperature being between two limits ($T_{\text{MIN}} < T < T_{\text{MAX}}$), since one cannot insert a zone accurately into the hydrogen ionization region and the zoning does not alter significantly for high temperatures. The first value of NSERT is the first zone interior to T_{MIN} , and succeeding values occur every second zone until T_{MAX} is reached. The process is then repeated.


```

SUBROUTINE RSTEP (DT,NTH,N2)
COMMON / QUANT / RHO(50,6),RHON(50,6),V(50,6),VN(50,6),W(50,6),
1 WV(50,6),R(50),TW(6),P(50,6),T(50,6),ENERG(50,6),GARB(500)
COMMON / ZNCHK / DRRAT,JKP,JTAU,DRTAU
COMMON / DRST / DRSTD
COMMON / RADIUS / RNEW(50),VBAR(50)
DATA DRMIN / 0.E+7 /
DATA IREZN,IRZNP / 0, 13 /
DATA ZERO,DNM,ONE,TTEST / 0.E+0, 5.E-1, 1.E+0, 7.E+3 /
DNM=NTH-1
VBAR(1)=ZERO
RNEW(1)=R(1)
DO 50 J=2,N2
VBAR0=ZERO
DO 40 L=2,NTH
40 VBAR0=VBAR0+V(J,L)
VBAR(J)=VBAR0/DNM
50 RNEW(J)=R(J)+DNM*(VBAR(J)+VBAR(J-1))*DT
NP1=N2+1
DRSTD=R(NP1)-R(N2)
DO 60 J=NP1,50
60 RNEW(J)=RNEW(J-1)+DRSTD
DRRT=ZER0
DR1=RNEW(2)-RNEW(1)
N2M1=N2-1
DRTAU=ZERO
JKP1=1
ICLK=-1
DRKP=DRMIN
DO 75 J=2,N2M1
DR2=RNEW(J+1)-RNEW(J)
IF (J.GT.(N2-3)) GO TO 62
IF (DR2.LT.DRKP) ICLK=J
62 CONTINUE
DRRT=DR1/DR2
IF (DRRT.LT.ONE) DRRT=ONE/DRRT
DO 65 L=2,NTH
65 IF (T(J,L).GT.TTEST) JKP1=J
IF (JKP1.GE.(J-1)) GO TO 70
IF (DRRT.LT.DRTAU) GO TO 70
DRTAU=DRRT
JTAU=J
70 CONTINUE
IF (DRRT.LT.DRRAT) GO TO 75
DRRT=DRRT
JKP=J
75 DR1=DR2
WRITE (6,1) DRRT,JKP,DRTAU,JTAU,ICLK
1 FORMAT (10H DR RATIO=,F8.2, 6H AT J=, I2, 5X, 6HDRTAU=,F8.2,
A 6H AT J=,I2,5X,5HICLK=,I3)
IF (ICLK.GT.0) GO TO 80
IREZN=0
RETURN
80 CONTINUE
DR1=RNEW(ICLK)-RNEW(ICLK-1)
DR2=RNEW(ICLK+2)-RNEW(ICLK+1)
IF (DR2.GT.DR1) ICLK=ICLK+1

```

```

CALL INTRP (ICLK,NTH)
IREZN=IREZN+1
IF (IREZN.LT.IRZKNP) RETURN
IREZN=0
CALL REVAMP (ICLK,NTH)
RETURN
END

```

```

SUBROUTINE REZONE (NTH,N1)
COMMON / ZNCHK / DRRAT,JKP,JKP1,ORTAU
COMMON / JIANT / RHQ(50,6),RHON(50,6),V(50,6),VN(50,6),W(50,6),
1 W(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),YAST(500)
DATA DRALLW,TMAX,RATALW,ONH,ONG / 5.E+0, 0.E+3,1,30E+0,5.E-1,
1 2.5E-1 / ,RATMAX,TINST / 10.,2.5E+4 /
IF (JKP.GE.JKP1) GO TO 15
IF (JKP1.GE.N1) GO TO 15
IF (ORTAU.LT.1.E+0) GO TO 15
CALL INTRP (JKP,NTH)
15 CONTINUE
IF (DRRAT.GT.RATMAX) GO TO 25
IF (DRRAT.LT.DRALLW) RETURN
IF (JKP.GE.N1) RETURN
DO 20 L=2,NTH
T1=T(JKP-1,L)
T2=T(JKP+1,L)
IF (T2.GT.TINST) GO TO 17
IF (T1.GT.TMAX) RETURN
17 T0=T(JKP,L)
RAT=T1/T0
RAT1=T0/T2
IF (RAT1.GT.RAT) RAT=RAT1
IF (RAT.GT.RATALW) RETURN
20 CONTINUE
25 CONTINUE
CALL INTRP (JKP,NTH)
RETURN
END

```

```

SUBROUTINE INTRP (JKP,NTH)
COMMON / QUANT / RHQ(50,6),RHON(50,6),V(50,6),VN(50,6),W(50,6),
1 W(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),GARB(500)
DATA ONH,ONG / 5.E-1, 2.5E-1 /
DO 40 L=2,NTH
JKPP=JKP+1
JKPM=JKP-1
T(JKP,L)=ONH*T(JKP,L)+ONG*(T(JKPP,L)+T(JKPM,L))
P(JKP,L)=ONH*P(JKP,L)+ONG*(P(JKPP,L)+P(JKPM,L))
V(JKP,L)=ONH*V(JKP,L)+ONG*(V(JKPP,L)+V(JKPM,L))
W(JKP,L)=ONH*W(JKP,L)+ONG*(W(JKPP,L)+W(JKPM,L))
ENERG(JKP,L)=ONH*ENERG(JKP,L)+ONG*(ENERG(JKPP,L)+ENERG(JKPM,L))
RHQ(JKP,L)=ONH*RHQ(JKP,L)+ONG*(RHQ(JKPP,L)+RHQ(JKPM,L))
40 CONTINUE
P(JKP)=ONH*(P(JKPP)+P(JKPM))
RETURN
END

```

```

SUBROUTINE REVAMP (IDELET,NTH)
COMMON / QUANT / RHO(50,6),RHON(50,6),V(50,6),VN(50,6),W(50,6),
A WN(50,6),R(50),TH(6),P(50,6),I(50,6),E(50,6),GARB(500)
COMMON / RADIUS / RNEW(50),VBAR(50)
COMMON / SFACE / NN(10),N1,N2,NSKP
DATA ONH,TMIN,TMAX,NSERT / 0.5, 2.E+4, 6.E+4, 1 /
IF (NSERT.GT.0) GO TO 30
DU 10 JJ=2,N1
J=N1+2-JJ
IF (T(1,2).GT.TMIN) GO TO 20
10 CONTINUE
20 NSERT=J
30 IF (T(NSERT,2).LT.TMAX) GO TO 60
NSERT=-1
RETURN
60 IM=IDELET-1
MAXTH=NTH+1
DO 100 JJ=NSERT,IM
J=NSERT+IM-JJ
JP1=J+1
R(JP1)=R(J)
RNEW(JP1)=RNEW(J)
VBAR(JP1)=VBAR(J)
DO 100 L=1,MAXTH
T(JP1,L)=T(J,L)
RHO(JP1,L)=RHO(J,L)
V(JP1,L)=V(J,L)
W(JP1,L)=W(J,L)
P(JP1,L)=P(J,L)
100 E(JP1,L)=E(J,L)
NM1=NSERT-1
NP1=NSERT+1
R(NSERT)=ONH*(R(NM1)+R(NP1))
RNEW(NSERT)=ONH*(RNEW(NM1)+RNEW(NP1))
VBAR(NSERT)=ONH*(VBAR(NM1)+VBAR(NP1))
DO 120 L=2,NTH
T(NSERT,L)=ONH*(T(NM1,L)+T(NP1,L))
V(NSERT,L)=ONH*(V(NM1,L)+V(NP1,L))
W(NSERT,L)=ONH*(W(NM1,L)+W(NP1,L))
RHO(NSERT,L)=ONH*(RHO(NM1,L)+RHO(NP1,L))
120 CALL THERM (RHO(NSERT,L),T(NSERT,L),P(NSERT,L),E(NSERT,L),OUT,
A DPT,DPR,DELADI,VAD,1)
WRITE (6,1) IDELET,NSERT
1 FORMAT (25H IN REVAMP, DELETING ZONE, I3, 5X,12H INSERT ZONE, I3)
NSERT=VNSERT-2
RETURN
END

```

VI. EQUATION OF STATE

17

The solution of the equation of state is analytic and taken from Paczynski¹⁷. The solution is divided into two parts: the Saha equation is solved for the ionization stages of hydrogen and helium in certain ranges of density and temperature, and complete ionization is assumed otherwise. The selection is made at the beginning of THERM, which evaluates pressure and internal energy and computes the derivatives of each with respect to density and temperature. This is done analytically for complete ionization, but numerically in the solution of the Saha equation. The Saha equation is solved by EEERG, which assumes that only one type of ionization is going on (first $H \rightarrow H^+$, then $He \rightarrow He^+$, and finally $He^+ \rightarrow He^{++}$). Subroutine PIERW is a quadratic-equation solver ($X^2 + BX + C = 0$). Other properties, including $\left(\frac{\partial \ln T}{\partial \ln P}\right)_{ad}$ (DELAD) and the adiabatic sound speed (VAD), can also be computed by THERM. The parameter INDEX determines what is computed.

INDEX = 1 computes only P and E

INDEX = 2 computes P, E, $\left(\frac{\partial P}{\partial T}\right)_\rho$, $\left(\frac{\partial E}{\partial T}\right)_\rho$

INDEX = 3 computes P, E, $\left(\frac{\partial P}{\partial T}\right)_\rho$, $\left(\frac{\partial E}{\partial T}\right)_\rho$, $\left(\frac{\partial P}{\partial T}\right)_\rho$, ∇_{ad} , C_ρ , $\Gamma_3 - 1$, C.

As pointed out earlier, far too much time is spent in computing the Cougan condition, so the equation of state should be altered to make that computation faster. One can insert one's own equation-of-state solution by deleting these three subroutines and inserting others. However, because of the large number of places from which the equation of state is needed, it is advisable to keep the subroutine name THERM and its calling list. One further necessary modification would be the solution for the temperature in the interior zones, given the internal energy and density (ESOL1 - see Sec. III-D).

```

SUBROUTINE THERM (RO,T,P,U,DUT,DPT,DPRO,DELAD,VAD,INDEX)
COMMON/ EXTRA / P1,R1,R2,UTERM
COMMON / ZMOD / X,Y,ZHSTAR,ZMCORE,DT
DATA TYP,TB,A,B,C,OPF,ZK,THR,FR,TWELV,DEL,ONE,FOO / 1.E+0,
1 1.E+5,-6.30908E+5, 1.5E+0,2.65E+1, 1.5E+0,2.523E-15,3.E+0,
2 4.E+0,1.2E+1,1.E-3,1.E+0,5.E+2 /
IF (T,GT,TR) GO TO 1
IF (A/T+B*ALOG(T)-ALOG(RO))-C) 2,1,1
1 P2=ZK*T**3/RO
P=(P1+P2)*RO*T
U=(OPF*P1+THR*P2)*T+UTERM
IF (INDEX,EQ,1) RETURN
DPT=(P1+FR*P2)*RO
DUT=OPF*P1+TWELV*P2
IF (INDEX,EQ,2) RETURN
DPRO=P1*T
GO TO 3
2 CALL EEERG(RO,T,X,Y,P,U,TYP)
IF (INDEX,EQ,1) RETURN
TU=(ONE+DEL)*T
CALL EEERG (RO,TU,X,Y,PUT,UUT,TYP)
TL=(ONE-DEL)*T
CALL EEERG(RO,TL,X,Y,PLT,ULT,TYP)
DPT=(PUT-PLT)/T*FOO
DUT=(UUT-ULT)/T*FOO
IF (INDEX,EQ,2) RETURN
RHOU=(ONE+DEL)*RO
CALL EEERG(RHOU,T,X,Y,PUR,UUR,TYP)
RHOL=(ONE-DEL)*RO
CALL EEERG (RHOL,T,X,Y,PLR,ULR,TYP)
DPRO=(PUR-PLR)/RO*FOO
3 Q=T/RO*DPT/DPRO
CP=DUT+Q/RO*DPT
VAD= SORT(DPRO*CP/DUT)
G3M1=DPT/RO/DUT
DELAD=DUT/CP*G3M1*P/RO/DPRO
RETURN
END

```

```

SUBROUTINE EEERG (RO,T,X,Y,P,U,TYP)
REAL NT,NH2,NHI,NHEI,NHEII,NHEIII,NE,LOGT,NH,NHE,NMET,NHII
DATA TEMP,DM,H,ONQ,TW,ON,ZERO,THOT,COEF1,OPF,ZIH,COEF2,ZIHE1,
A TCOLD,ZIHE2,ZK,ZRAD,TR,C1,C2,C3,C4 / 5040., 2.302585, 1.6734E-7
B , 0.25, 2., 1., 0., 5.E+4, 15.3828, 1.5, 13.595, 15.9849, 20.58,
C 1.E+4, 54.403, 1.3805E-16, 2.523E-15, 3., 3.585E-17, 2.536E-11,
D 3.937E-11, 1.2652E-10 /
DATA EPS,TMID,C5,C6,C7,C8,TMID1,STEP,C9,C10,TLOW,C11,C12,C13,C14
11,E-3,2.E+4,28,0.925,4.92516,5.6191E-2,3.2688E-3,1.2E+4,1.E+3,2.1,
2 2.5E-4,3.000,,-1.89E+3,3.36,4.E-5,1.3805E-16 /
TETA=TEMP/T
LOGT=ALOG10(T)
NH=X*RO/H
NHFI=ONQ*Y*RO/H
NMET=TW*(ON-X*Y)*RO/H
NHEI=NHE
NHEII=ZERO
NHEIII=ZERO
IF (T.GT.THOT) GO TO 5
C H IONIZATION
FAC= EXP(DM*(COEF1+OPF*LOGT-ZIH*TETA))
NHII=PIERW (FAC,FAC*NH)
NE=NHII
NHI=NHII*NE/FAC
C HE I IONIZATION
FAC= EXP(DM*(COEF2+OPF*LOGT-ZIHE1*TETA))
NHEII=PIERW (FAC+NE,FAC*NHE)
NE=NE+NHEII
NHEI=NHEII*NE/FAC
GO TO 7
5 NHI=ZERO
NHEI=ZERO
NHII=NH
NHEII=NHE
NE=NH+NHE
7 CONTINUE
C HE II IONIZATION
IF (T.LT.TCOLD) GO TO 10
FAC= EXP (DM*(COEF1+OPF*LOGT-ZIHE2*TETA))
NHEIII=PIERW (FAC+NE,FAC*NHEII)
NE=NE+NHEIII
NHEII=NHEIII*NE/FAC
10 CONTINUE
NH2=ZERO
IF (NHI,LT,EPS*NH,OR,T.GT,TMID) GO TO 20
FAC=C5-TETA*(C6-TETA*(C7+TETA*C8))-LOGT
IF (T.GT,TMID1) FAC=FAC*(T-TMID1)/STEP
FAC=EXP(DM*FAC)
NHI=PIERW(FAC,FAC*NHI)
NH2=NH1**2/FAC/TW
20 NT=NH+NHE+NE=NH2
UH2=T*(C9+T*C10)
IF (T.GT,TLOW) UH2= C11+T*(C12+T*C13)
UU=C14*NH2*UH2/RO
PG=ZK*NT*T
PR=ZRAD*T**4
P=PG+PR
U=(OPF*PG+TR*PR+C1*NHI+C2*NHII+C3*NHEII+C4*NHEIII)/RO
U=U+UU
RETURN
END

```

```

FUNCTION PIERW (B,C)
DATA OM2,ONQ,ONH / 1.E-2, 2.5E-1, 5.E-1 /
IF (C/B/B-OM2 ) 11,11,10
10 PIERW=B*( SQRT(ONQ +C/B/B)-ONH)
RETURN
11 PIERW=C/(B+C/(B+C/(B+C/(B+C/B))))
RETURN
END

```

VII. TRANSITION BETWEEN RUNS

The subroutines RDMD, PNCHMD, UREAD, and STATIC are used to initiate a run or preserve data for the next run.

A. UREAD

This subroutine reads the opacity tables in the format used by Pacyznski for his program GOB. The composition (X,Y) is read first. The opacity table (FAP) is read, and the cards are numbered to detect any error in reading in the table. The table has dimensions 51 x 31, and is of the form

$$FAP(I,J) = \kappa(T_i, \rho_j) \quad (78)$$

where

$$\begin{aligned} -12 \leq \log \rho \leq 3 & \quad \Delta \log \rho = 0.5 \\ 3.3 \leq \log T \leq 8.2 & \quad \Delta \log T = .05 \quad \log T \leq 5 \\ & \quad \Delta \log T = 0.2 \quad \log T > 5 . \end{aligned} \quad (79)$$

Hence

$$\begin{aligned} i_0 &= 20 \log T - 65 \\ \text{if } i_0 > 35 & \quad i = 35 + (i_0 - 35)/4 \\ j &= 2 \log \rho + 25 . \end{aligned} \quad (80)$$

These relations are used in ZKAP. Finally, the quantities P1 and UTERM (used in the equation of state and ESOL1) and R1 and R2 (used in OPACIT) are computed.

B. STATIC

On the first run for a given model, a call to STATIC is made from the main program. The data read in by STATIC has been generated by the program STATPCS, whose function it is to produce a model of given M, T_{eff} , L, and composition in hydrostatic and radiative equilibrium (using the diffusion approximation) with the difference equations used in PCS. The initial model is taken to be spherically symmetric. First a data card is read in with the following:

N1 - number of radial zones in the initial model
 ISEQ(1) - indicator of opacity table (King Ia = 1)
 ISEQ(2) - L in 10^{33} ergs/s
 ISEQ(3) - T_{eff}
 IVTST - number of the radial zone which has a temperature as close to 10^5 K as possible (close to the node of 1H)
 ILAW - power law of imposed radial velocity distribution
 VFAC - surface velocity of imposed velocity distribution, i.e., $v_r(r) = \text{VFAC} (r/r_{\text{surf}})^{\text{ILAW}}$

The following cards have been punched by STATPCS and read in: r_i , T_i , ρ_i , and M_r (20). The power law velocity distribution is generated (60) and quantities are set up for all angular zones (70). To initiate convection, it is necessary to have some small horizontal velocity component somewhere. The zones exterior to the initial model are generated with radius increment DRSTD, and everything is set to zero outside the model (80). Results are printed and the constant value of the flux emerging from the inert core (FLUX) is computed.

C. RDMD AND PNCHMD

On all runs except the first, subroutine RDMD is called instead of STATIC. Of these two subprograms, the one to be called is determined by the parameter ISTRT, which is read by the main program from a punched data card. This card must be changed manually after the first run. The function of RDMD is to read (currently from punched cards) the results of the last model of the previous run. These results are punched by PNCHMD. Most of these two subroutines are self-explanatory. There is a "safety valve" in that each card (after the first four which are just identifiers) is numbered. The numbering is checked by RDMD to see that all cards are present and in order: if they are not, execution is terminated with the warning given in FORMAT 6.

```

SUBROUTINE UREAD
COMMON/ EXTRA / P1,R1,R2,UTERM
COMMON / ZMOD / X,Y,ZHSTAR,ZMCORE,DT
COMMON/KAPA/FAP(51,31)
READ (5,305) X,Y
305 FORMAT (1X,F7,5,1X,F7,5)
I=0
K2=0
300 CONTINUE
I=I+1
IF (I.GT.51) GO TO 304
301 FORMAT (1X,I5,14F5.2)
K2=K2+1
READ (5,301) K1,(FAP(I,J),J=1,14)
IF (K1.NE.K2) GO TO 302
K2=K2+1
READ (5,301) K1,(FAP(I,J),J=15,28)
IF (K1.NE.K2) GO TO 302
K2=K2+1
READ (5,301) K1,(FAP(I,J),J=29,31)
IF (K1.NE.K2) GO TO 302
GO TO 300
302 CONTINUE
303 FORMAT (1X,21HWHONG OPACITY CARD,K=,I3)
WRITE (6,303) K2
CALL EXIT
304 CONTINUE
P1=8.250+7*(5.0-1+1.50+0*X+0.250+0*Y)
R1=1.020+0**4
R2=0.980+0**4
UTERM=1.5160+13*X+1.8900+13*Y
RETURN
END

```

```

SUBROUTINE STATIC (NUMZN,DRSTD,VTEST,IVTST,ISEQ)
COMMON / GRAV / ZMSMR(50)
COMMON / MELT / FLUX
COMMON /QUANT / RHO(50,6,2), V(50,6,2),W(50,6,2),R(50),TH(6),
1 P(50,6),T(50,6),ENERG(50,6),STF(500)
COMMON/ MODPR / NTH,NTHZ,MAXTH,MAXZ,MZONE
COMMON / ZMOD / X,Y,ZMSTAR,ZMCORE,DT
COMMON / SFACE / NN(10),N1,N2,NSKP
COMMON / PRDY / PP
COMMON /POVDI/ WORK(50)
DIMENSION ISEQ(3)
DATA G,PI,ZERO,SIG43 / 6.668E-8,3.14159E+0,0.E+0,7.55866667E-5 /
DT=10.
READ(5,1) N1,ISEQ(1),ISEQ(2),ISEQ(3),IVTST,ILAW,VFAC
1 FORMAT(6I5,E14.6)
DO 10 L=1,MAXTH
10 NN(L)=N1
DO 20 I=1,N1
READ (5,2) R(I),T(I,2),RHO(I,2,1),ZMSMR(I)
2 FORMAT (2X,4E19,12)
20 CONTINUE
READ (5,2) ZMSTAR,ZMCORE
VMIN=VFAC*(R(1)/R(N1))**ILAW
DO 60 I=1,N1
RRE=(R(I)/R(N1))**ILAW
V(I,2,1)=-(VFAC*RRE-VMIN)
CALL THERM (RHO(I,2,1),T(I,2),P(I,2),ENERG(I,2),D,D1,D2,D3,D4,1)
60 CONTINUE
V(1,2,1)=ZERO
VTEST=ABS(V(IVTST,2,1))/1.2
DO 70 I=1,N1
WORK(I)=ZERO
DO 70 J=1,MAXTH
RHO(I,J,1)=RHO(I,2,1)
RHO(I,J,2)=RHO(I,2,1)
T(I,J)=T(I,2)
P(I,J)=P(I,2)
ENERG(I,J)=ENERG(I,2)
V(I,J,1)=V(I,2,1)
W(I,J,1)=ZERO
V(I,J,2)=ZERO
70 W(I,J,2)=ZERO
C PERTURBATION TO THETA VELOCITY
W(10,2,1)=1,D=3
C
C FIND DRSTD
DR1=R(N1)-R(N1-1)
DRSTD=DR1
C FILL OUT MESH
N2=N1+1
DO 80 I=N2,MZONE
ZMSMR(I)=ZMSTAR
WORK(I)=ZERO
R(I)=R(I-1)+DRSTD
DO 80 J=1,MAXTH
T(I,J)=ZERO
P(I,J)=ZERO

```

```

ENERG(I,J)=ZERO
DO 82 N=1,2
  RHO(I,J,N)=ZERO
  V(I,J,N)=ZERO
  W(I,J,N)=ZERO
80 CONTINUE
  WRITE (6,4)
4  FORMAT (15H INITIAL MODEL, //8X, 1HR, 14X, 1HT, 14X, 1HP, 12X, 3HRHO,
1  12X, 3HM R, 12X, 5HENERG, 5X, 8HRADIAL, V)
  DO 250 I=1,N2
    WRITE (6,6) R(I),T(I,2),P(I,2),RHO(I,2,1),ZMSHR(I),ENERG(I,2),
1  V(I,2,1)
6  FORMAT (7(1X,E14,6))
250 CONTINUE
  CALL OPACIT (T(1,2),T(2,2),RHO(1,2,1),RHO(2,2,1),FK,DU1,DU2,-1)
  FL1=-SIG43/(R(2)-R(1))
  FLUX=FL1*FK*(T(2,2)**4-T(1,2)**4)
  WRITE (6,3) FLUX,DRSTD,ILAN,VTEST,IYST
3  FORMAT (1X,5HFLUX=,E14,6,1X,6HDRSTD=,E14,6,2X,5HILAN=,I3,5X,
A  6HVTEST=,E14,6,9H IN ZONE ,I3)
  N2=N1
  RETURN
END

```

```

SUBROUTINE PNCMD (TIME,DRSTD,VTEST,IVTST,MODTOT,IWORK,ISEQ)
COMMON /QUANT / RHO(50,6,2), V(50,6,2), A(50,6,2), R(50), TM(6),
2   P(50,6), T(50,6), ENRG(50,6), NAST(500)
COMMON / MELT / FLUX
COMMON / SFACE / NN(10), N1, N2, NSKP
COMMON / ZMOD / X, Y, ZMSTAR, ZMCORE, DT
COMMON /MCDPR/ NTH, NTHZ, MAXTH, MAXZ, MZONE
COMMON /POVOT/ NORK(50)
DIMENSION ISEQ(3)
WRITE(7,7) MODTOT, ISEQ(1), ISEQ(2), ISEQ(3)
7   FORMAT(4I6)
WRITE (7,1) TIME, DT, N1, (NN(L), L=2, NTH), NSKP
1   FORMAT (   E14.6, 1X, F14.6, 9I4)
WRITE (7,3) ZMSTAR, ZMCORE, FLUX
3   FORMAT(3E24, 16)
WRITE (7,1) DRSTD, VTEST, IVTST
LCT=0
DO 20 L=2, NTH
K=NN(L)
DO 20 J=1, K
LCT=LCT+1
WRITE (7,4) T(J,L), RHO(J,L,1), LCT
LCT=LCT+1
WRITE (7,4) V(J,L,1), A(J,L,1), LCT
4   FORMAT(2E24, 16, 26X, I3)
20  CONTINUE
K1=MZONE+3
DO 40 J=1, K1, 3
K=J+2
IF (K,GT,MZONE) K=MZONE
LCT=LCT+1
WRITE (7,2) (R(L), L=J, K), LCT
2   FORMAT(3E24, 16, 2X, I3)
40  CONTINUE
IF( IWORK,LE,0) RETURN
DO 60 J=1, K1, 3
K=J+2
IF( K,GT,MZONE) K=MZONE
LCT=LCT+1
WRITE(7,2) (NORK(L), L=J, K), LCT
60  CONTINUE
RETURN
END

```

```

      SUBROUTINE ROND ( TIME,DRSTD,VTEST,IVTST,MODTOT,IWORK,ISEQ)
      COMMON /QUANT / RHO(50,6,2), V(50,6,2),M(50,6,2),H(50),IH(6),
2     P(50,6),T(50,6),ENERG(50,6),DELD(50,6),WAS(200)
      COMMON / MELT / FLUX
      COMMON / SFACE / NN(10),N1,N2,NSKP
      COMMON / ZMOD / X,Y,ZMSTAR,ZMCORE,DT
      COMMON /MOMPR/ NTH,NTHZ,MAXTH,MAXZ,MZONE
      COMMON /PDVDT / WORK(50)
      DIMENSION ISEQ(3)
      ZO=J,0+0
      READ(5,7) MODTOT,ISEQ(1),ISEQ(2),ISEQ(3)
7     FORMAT(4I6)
      READ (5,1) TIME,DT,N1,(NN(L),L=2,NTH),NSKP
1     FORMAT (E14,6,1X,E14,6,9I4)
      NN(1)=NN(2)
      NN(MAXTH)=NN(NTH)
      READ (5,3) ZMSTAR,ZMCORE,FLUX
3     FORMAT (3E24,16)
      READ (5,1) DRSTD,VTEST,IVTST
      LCT=0
      DO 20 L=2,NTH
      K=NN(L)
      DO 20 J=1,K
      LCT=LCT+1
      READ (5,4) T(J,L),RHO(J,L,1),L1
      IF (L1,NE,LCT) GO TO 5
      CALL THERM (RHO(J,L,1),T(J,L),P(J,L),ENERG(J,L),DU,D2,D3,
1     DELAD(J,L),VAD,3)
      LCT=LCT+1
      READ (5,4) V(J,L,1),M(J,L,1),L1
      IF (L1,NE,LCT) GO TO 5
4     FORMAT(2E24,16,26X,13)
20     CONTINUE
      K1=MZONE-3
      DO 40 J=1,K1,3
      K=J+2
      IF (K,GT,MZONE) K=MZONE
      LCT=LCT+1
      READ (5,2) (R(L),L=J,K),L1
2     FORMAT(3E24,16,2X,13)
      IF (L1,NE,LCT) GO TO 5
40     CONTINUE
      N2=J
      DO 50 L=2,NTH
      IF (NN(L),GT,N2) N2=NN(L)
      K=NN(L)+1
      DO 50 J=K,MZONE
      RHO(J,L,1)=ZO
      RHO(J,L,2)=ZO
      T(J,L)=ZO
      P(J,L)=ZO
      ENERG(J,L)=ZO
      V(J,L,1)=ZO
      M(J,L,1)=ZO
50     CONTINUE
      CALL FRAC (N2,NTH)
      IF(IWORK.LE,0) RETURN

```

```

00 55 J=1,K1,3
K=J+2
IF( K,GT,MZONE) K=MZONE
LCT=LCT+1
READ (5,2) (WORK(L),L=J,K),L1.
IF( L1.NE,LCT) GO TO 5
55 CONTINUE
GO TO 60
5 WRITE (6,6) LCT,L1
6 FORMAT (43H READING MODEL HAS WRONG CARD L SHOULD BE ,I3,
Z 8H AND IS ,I3)
CALL EXIT
68 CONTINUE
RETURN
END

```

VIII. OPACITY CALCULATION

The two subroutines governing the opacity are OPACIT and ZKAP. The opacity at interfaces is computed by the Christy¹ weighting scheme

$$1/\rho\kappa = \frac{2}{\rho_1 + \rho_2} \frac{T_1^4/\kappa_1 + T_2^4/\kappa_2}{T_1^4 + T_2^4} \quad (81)$$

where (ρ_1, T_1) and (ρ_2, T_2) are the density and temperature of adjacent zones. Derivatives of this function with respect to T_1 and T_2 are computed if the parameter (1) in OPACIT is greater than zero.

The opacities $\kappa_1 = \kappa(\rho_1, T_1)$ and $\kappa_2 = \kappa(\rho_2, T_2)$ are linearly interpolated in $\log \rho$ and $\log T$ through the opacity table read by UREAD. The table has uniform spacing, allowing an easy determination of the position in table (i,j) from knowledge of $\log \rho$ and $\log T$. This table⁵ slightly underestimates the bump in the second helium ionization zone. A different table can be used by deleting subroutines UREAD and ZKAP and inserting new subroutines to read the table and interpolate. However, the quantities in the common block EXTRA must be computed.

```
SUBROUTINE OPACIT (T1,T2,RHO1,RHO2,FKAP,D1,D2,I)
COMMON/ EXTRA / P1,R1,R2,UTERM
DATA TW,FIF / 2.E+0,5.E+1 /
T3=T1**4
T4=T2**4
FK1=T4/ZKAP(RHO2,T2)
FK0=T3/ZKAP(RHO1,T1)
FKAP=TW/(RHO1+RHO2)*(FK0+FK1)/(T3+T4)
IF (I.LE.0) RETURN
WGT1=1.02
WGT2=0.94
FK2=R1*T3/ZKAP(RHO1,WGT1*T1)
FK3=R2*T3/ZKAP(RHO1,WGT2*T1)
D1=FIF/(RHO1+RHO2)*((FK2+FK1)/(R1*T3+T4)-(FK3+FK1)/(R2*T3+T4))/T1
FK2=R1*T4/ZKAP(RHO2,WGT1*T2)
FK3=R2*T4/ZKAP(RHO2,WGT2*T2)
D2=FIF/(RHO1+RHO2)*((FK2+FK0)/(T3+R1*T4)-(FK3+FK0)/(T3+R2*T4))/T2
RETURN
END
```



```

FUNCTION ZKAP (R0,TE)
COMMON/KAPA/FAP(51,31)
DATA TW,TWF,TND,SNV,THRF,FR,ON / 2.E+0,2.5E+1,2.E+1,6.5E+1,3.5E+1
1 4.E+0, 1.E+0 /
INTEGER DI,TI
C INTERPOLATE KAPPA FROM CALCULATED VALUES GIVEN BY PACZYNSKI
D=TW*ALOG10(R0)+TWF
DI=INT(D)
D=D-DI
T=TND*ALOG10(TE)-SNV
IF (T.GT,THRF) T=THRF+(T-THRF)/FR
TI=INT(T)
T=T-TI
FAPP=((ON-T)*((ON-D)*FAP(TI,DI)+D*FAP(TI,DI+1)))+
Z T*((ON-D)*FAP(TI+1,DI)+D*FAP(TI+1,DI+1))
ZKAP=EXP(2.302585*FAPP)
RETURN
END

```

IX. SUBSIDIARY SUBROUTINES

A. PDV

The subroutine computes the work done per zone in a period.

$$w = \oint \frac{P}{\rho} \vec{v} \cdot \vec{v} \Delta m \, dt$$

where Δm is the mass in the zone. At each time step we compute the work for each radial zone by

$$w_i = \sum_{j=1}^{N_c} \frac{P_{ij}}{\rho_{ij}} (\vec{v} \cdot \vec{v})_{ij} \Delta m_{ij} \Delta t .$$

The work from the beginning of the period to the present (WORK) is stored for each radial zone. Note that

$$\Delta \text{vol}_{ij} = -\frac{\Delta m_{ij}}{\rho_{ij}} .$$

The period is defined by alternating zero crossings in the velocity of a zone located near 10^5 K in the static model. The temperature is selected because it is usually close to the position of the node of the 1F mode, hence ought to be relatively free of modal contamination and should determine the period rather well. However, the calculations do not yield realistic results unless the motion in the entire model is extremely periodic with the proper phase. If the parameter that determines the end of the period (IPRD) were defined for each radial zone, this calculation might give more realistic results. I do not use this subroutine, and have not corrected it.

```

SUBROUTINE PDV (DT,IPRD)
COMMON / QUANT / RHO(50,6),RHON(50,6),V(50,6),VN(50,6),A(50,6),
A  WN(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),WASTE(500)
COMMON / FSIN / SINE(10),COSI(10),DTH
COMMON / SFAC / NN(10),N1,N2,NSKP
COMMON / PDVDT / WORK(50)
DATA ZERO,ONEHF,PI2 / 0., 0.5, 6.2831853072 /
RJM=ONEHF*(R(1)+R(2))
DO 100 J=2,N1
RJP=ONEHF*(R(J)+R(J+1))
RDTH=R(J)*DTH
SINLM=ONEHF*(SINE(1)+SINE(2))
DO 50 L=2,NTH
T1=(RJP**2*V(J,L)-RJM**2*V(J-1,L))/R(J)**2/(RJP-RJM)
SINLP=ONEHF*(SINE(L)+SINE(L+1))
T2=(SINLP**2*(J,L)-SINLM**2*(J,L-1))/SINE(L)/RDTH
DVOL=PI2*DT*SINE(L)*R(J)**2*(RJP-RJM)
WORK(J)=WORK(J)+P(J,L)*(T1+T2)*DVOL
50 SINLM=SINLP
100 RJM=RJP
IF (IPRD,GT,0) RETURN
WKTOT=ZERO
DO 200 J=2,N1
WRITE (6,1) T(J,2),WORK(J)
1 FORMAT (1X,1PE13,5)
WKTOT=WKTOT+WORK(J)
200 WORK(J)=ZERO
WRITE (6,2) WKTOT
2 FORMAT (/12H TOTAL WORK=,1PE13,5,/)
RETURN
END

```

B. EKIN

The subroutine computes the kinetic energy of the model at the time step called. This is done in two ways:

$$\left. \begin{aligned} \text{EN} &= \frac{1}{2} \int \rho (v^2 + w^2) dV_0 \rightarrow \frac{1}{2} \sum_{i,j} \rho_{ij} (v_{1i}^2 + w_{1j}^2) dV_0|_{ij} \\ \text{ENPULS} &= \frac{1}{2} \int \rho v_0^2 dV_0 \rightarrow \frac{1}{2} \sum_{i,j} \rho_{ij} v_0^2 dV_0|_{ij} \end{aligned} \right\} \quad (82)$$

The first method (EN) is designed to measure both the pulsational and convective kinetic energies, whereas the second (ENPULS) should measure only the pulsational kinetic energy. Of course, the second calculation is accurate only to the extent that the horizontal average of the radial velocity (v_0) is a true measure of the pulsational velocity. Since velocities are measured at interfaces (see Fig. 1), we average to find the value at midpoints.

$$\left. \begin{aligned} v_{ij} &= \frac{1}{2}(v_{i+1/2,j} + v_{i-1/2,j}) \\ w_{ij} &= \frac{1}{2}(w_{i,j+1/2} + w_{i,j-1/2}) \end{aligned} \right\} \quad (83)$$

C. FRAC

The interior mass, M_r (ZMSMR), is computed for use in the momentum equation. First compute $\bar{\rho}$ (RHOBAR) by

$$\bar{\rho}_i = \frac{\int \rho(r, \theta) \sin \theta d\theta}{\int \sin \theta d\theta} \rightarrow \frac{\sum_j \rho_{ij} \sin \theta_j}{\sum_j \sin \theta_j} \quad (84)$$

From this

$$\Delta m_i = \bar{\rho}_i \Delta V_0|_i = \bar{\rho}_i 4\pi r_i^2 (r_{i+1/2} - r_{i-1/2}) \quad (85)$$

```

SUBROUTINE EKIN (NTH,N2)
COMMON / FGIN / SINE(10),COSI(10),DTH
COMMON / RADIUS / RNF*(50),VBAR(50)
COMMON / QUANT / RHC(50,6),RHON(50,6),V(50,6),VN(50,6),N(50,6),
1  WN(50,6),R(50),TH(6),P(50,6),I(50,6),ENERG(50,6),MAST(500)
DATA ZERN,ONH,RHOMN,PI2S3 / 0.E+0, 5,E-1, 3,E-10, 2.0943951E+3 /
EN=ZERN
ENPULS=ZERN
R1=ONH*(R(N2)+R(N2+1))
DO 170 J1=2,N2
J=N2+2-J1
R2=ONH*(R(J)+N(J-1))
DO 33 L=2,NTH
RHON=RHON(J,L)
IF (RHON.LT.RHOMN) GO TO 80
DVOL=PI2S3*(R1**3-R2**3)*SINE(L)*DTH
V1=ONH*(V(J,L)+V(J-1,L))
V2=ONH*(V(J,L)+N(J,L-1))
DKE=ONH*RHON*(V1**2+V2**2)*DVOL
FHE=EN+DKE
V1=ONH*(VBAR(J)+VBAR(J-1))
DKEPUL=ONH*RHON*V1**2*DVOL
ENPULS=ENPULS+DKEPUL
80 CONTINUE
R1=R2
100 CONTINUE
WRITE (0,1) EN,ENPULS
1 FORMAT (16H KINETIC ENERGY=1PE14,6,5X,14H PULSATION KE=,E14,6)
RETURN
END

```

and

$$M_{r_1} = M_* - \sum_{k=1}^{N_2} \Delta m_k \quad . \quad (86)$$

This form is used for accuracy.

D. VISC

The artificial viscosity is found in the usual way for astrophysical pulsation problems, and includes the modification suggested by Stellingwerf.¹⁸ First the square of the velocity difference (V) is computed. If this is smaller than a certain fraction (RATMIN) of the square of the adiabatic sound speed (approximated by P/ρ), no viscous pressure is computed. If it is larger, the viscous pressure may be composed of two terms, one for each direction of the form

$$\begin{aligned} P_{\text{visc}} &= \text{constant} \quad \rho \quad (\Delta v)^2 \quad \text{if } \Delta v < 0 \quad (\text{contraction}) \\ &= 0 \quad \quad \quad \text{otherwise.} \end{aligned} \quad (87)$$

The current value of the constant is 2.

```

SUBROUTINE FRAC (N2,NTH)
COMMON / QUANT / RHO(50,6),V(1500),R(50),TH(6),w(1400)
COMMON / ZMOD / X,Y,ZMSTAR,ZMCORE,DT
COMMON / FSIN/ SINE(10),COSI(10),DTH
COMMON / GRAV / ZMSMR(50)
DATA PIFTH,ZERO,ONH / 4,1887867E+0, 0,E+0, 5,E+1 /
C   ZMSMR(J) IS 4 SUB R AT INTERFACE J+ONE=HALF
      N2P1=N2+1
      DO 10 I=N2P1,50
10    ZMSMR(I)=ZMSTAR
      ZMASS=ZERO
      DO 20 I=2,N2
      K=N2+2-I
      R1=ONH*(R(K+1)+R(K))
      R2=ONH*(R(K+1)+R(K))
      S1=ZERO
      S2=ZERO
      DO 15 J=2,NTH
      S1=S1+SINE(J)
15    S2=S2+SINE(J)*RHO(K,J)
      RHOBAR=S2/S1
      ZM1=PIFTH*(R1**3-R2**3)*RHOBAR
      ZMASS=ZMASS+ZM1
20    ZMSMR(K)=ZMSTAR-ZMASS
      RETURN
      END

```

```

FUNCTION VISC (RHO,P,VR1,VR3,VA5,VA6)
DATA Z0,TW,RATMIN / 0., 2., 0.001 /
VISC=Z0
V=(VR1-VR3)**2+(VA5-VA6)**2
IF ((RHO*V/P),LT,RATMIN) RETURN
IF ((VR1-VR3),LT,Z0) VISC=RHO*(VR1-VR3)**2*TW
IF ((VA5-VA6),LT,Z0) VISC=VISC+RHO*(VA5-VA6)**2*TW
RETURN
END

```

X. PLOTTING

The plotting is one of the few features of the calculation that is machine-dependent. The subroutines used (GOODY and MNMX) were written by Stephen Hodson of LASL. For each plot, there are two calls to the Los Alamos computer library subprogram PLOJB (J562A). The first call sets up the grid, and the second plots the points. The machine subroutine LINCNT spaces to the line designated in the current film frame for writing the identifying notation. The boundaries for some of the graphs are taken to be some factor multiplying the maximum and minimum values of the arrays plotted, which are determined by subroutine MNMX (the maximum and minimum values of the arrays XS and YS, each containing KNT elements, are returned in XPN and YPN, respectively). Examples of each plot are given in Figs. 4-7.

In the 100 loop of GOODY, the convective flux (FLUX) is computed, consistent with our assumptions, by

$$F_i(r) = \langle \rho(r,\theta) v_{rc}(r,\theta) \delta E(r,\theta) \rangle \quad (88)$$

where $\langle \rangle$ denotes a horizontal average and

$$\left. \begin{aligned} v_{rc}(r,\theta) &= v_r(r,\theta) - v_0(r) \\ \delta E(r,\theta) &= E(r,\theta) - \langle E(r,\theta) \rangle \end{aligned} \right\} . \quad (89)$$

The quantities v_{rc} (VV) and $\langle E(r,\theta) \rangle$ (EBAR) are the convective radial velocity component and horizontal average of the internal energy, respectively.


```

SUBROUTINE GOODY (TIME,VS,N1,NTH,IRADIL)
COMMON/QUANT/ RHO(50,6),RHON(50,6),V(50,6),VN(50,6),W(50,6),
1NH(50,6),R(50),TH(6),P(50,6),T(50,6),ENERG(50,6),CELAD(50,6),
2 ZL(50,4)
COMMON /RADIUS /RNEW(50),VBAR(50)
DIMENSION TBAR(50),PBAR(50),ZLBAR(50)
C SOURCE SIES FOR ARRAYS.
DIMENSION PLMT(2),TLMT(2),PBLMT(2),VLMT(2),XPN(2),YPN(2)
DATA PLMT,TLMT,PBLMT,VLMT / 5,E3,1,E12.5,E3,1,E6,5,E-9,1,E-4,
1 -4.E06,4.E06 /
DATA ZERO/0.0/
ZNTN=NTH-1
DO 20 I=1,N1
ZLBAR(I)=ZERO
TBAR(I)=ZERO
PBAR(I)=ZERO
DO 10 J=2,NTH
TBAR(I)=TBAR(I)+T(I,J)/ZNTN
ZLBAR(I)=ZLBAR(I)+ZL(I,J-1)/ZNTN
10 PBAR(I)=PBAR(I)+P(I,J)/ZNTN
20 CONTINUE
C SET OF GRID.
C PLOT T.VS.P
CALL PLOJJB(PLMT,TLMT,-2,-1,-1,48,-1,,1,,1.,25HTEMPERATURE,VS,
1PRESSURE,-25,11HPRESSURE,11,11HTEMPERATURE,11)
CALL PLOJJB(PBAR,TBAR,-N1,-1,-1,-39,-1,,1,,1.,25HTEMPERATURE,VS
1PRESSURE,25,11HPRESSURE,11,11HTEMPERATURE,11)
CALL LINCNT(63)
WRITE(12,1110) TIME,VS
1110 FORMAT(30X,5HTIME=,1PE15.6,10X,3HVS=,1PE15.6)
C CALCULATE RHO.
DO 30 I=1,N1
PBAR(I)=0
DO 30 J=2,NTH
30 PBAR(I)=PBAR(I)+RHO(I,J)/ZNTN
C PLOT RHO VS T.
CALL PLOJJB(TLMT,PBLMT,-2,-1,-1,48,-1,,1,,1.,20HRHO,VS,TEMPER.
1TURE,-20,11HTEMPERATURE,11,3HRHO,3)
CALL PLOJJB(TBAR,PBAR,-N1,-1,-1,-39,-1,,1,,1.,20HRHO,VS,TEMPER.
1TURE,20,11HTEMPERATURE,11,3HRHO,3)
CALL LINCNT(63)
WRITE(12,1110) TIME,VS
C PLOT PULSATION V VS T.
CALL PLOJJB(TLMT,VLMT,-2,1,-1,48,-1,,1,,1.,46HRADIAL PULSATIO
1VELOCITY,VS,TEMPERATURE,-46,11HTEMPERATURE,11,
223HAVERAGE RADIAL VELOCITY,23)
CALL PLOJJB(TBAR,VBAR,-N1,1,-1,-39,-1,,1,,1.,46HRADIAL PULSATIO
1VELOCITY,VS,TEMPERATURE,46,11HTEMPERATURE,11,
223HAVERAGE RADIAL VELOCITY,23)
CALL LINCNT(63)
WRITE(12,1110) TIME,VS
C PLOT L VS T.
CALL JNNY(TBAR,ZLBAR,N1,XPN,YPN)
YPN(1)=YPN(1)/1.5
YPN(2)=YPN(2)*1.5
CALL PLOJJB(TLMT,YPN,-2,1,-1,48,-1,,1,,1.,37HRADIATIVE LUMIN
1SITY,VS,TEMPERATURE,-37,11HTEMPERATURE,11,20HRADIATIVE LUMINOSIT

```

```

2Y,2J)
  CALL PLUJHC(HBAR,ZLBAR,-N1,1,-1,-39,-1.,1.,1.,37HRADIATIVE LUMI-
  NISITY,VS,TEMPERATURE,37,11TEMPERATURE,11,20HRADIATIVE LUMINOSI-
  TY,2J)
  CALL LTHCNT(43)
  WRITE(12,1110) TIME,VS
  K=IRADIL+9
  DO 10 J=K,11
    EBAR=ZERO
    DO 50 L=2,NTH
      50 EBAR=EBAR+ENERG(J,L)/ZNTH
      FLUX=ZERO
      DO 60 L=2,NTH
        VV=J.5*(V(J,L)+V(J-1,L)-VBAR(J)-VBAR(J-1))
      60 FLUX=FLUX+RHO(J,L)*VV*(ENERG(J,L)-EBAR)/ZNTH
      WRITE(6,1) J,TBAR(J),PBAR(J),EBAR,VV,FLUX
    1 FORMAT (1X,I3,1P5E14,5)
100 CONTINUE
  RETURN
  END

```

```

      SUBROUTINE MNMX (XS,YS,KNT,XPN,YPN)
C
C   FINDS LARGEST POSITIVE AND NEGATIVE X AND
C   LARGEST POSITIVE AND NEGATIVE Y, OF THE ARRAYS XS AND YS.
C   VALUES ARE RETURNED THROUGH XPN AND YPN ARRAYS.
      DIMENSION X3(KNT),YS(KNT),XPN(2),YPN(2)
      XPN(1)=YPN(1)=1.E100
      XPN(2)=YPN(2)=-1.E100
      DO 1 I=1,KNT
C   LARGEST NEGATIVE X.
        IF(XS(I).LT.XPN(1)) XPN(1)=XS(I)
C   LARGEST POSITIVE X.
        IF(XS(I).GT.XPN(2)) XPN(2)=XS(I)
C   LARGEST NEGATIVE Y.
        IF(YS(I).LT.YPN(1)) YPN(1)=YS(I)
C   LARGEST POSITIVE Y.
        IF(YS(I).GT.YPN(2)) YPN(2)=YS(I)
      1 CONTINUE
      RETURN
      END

```

Fig. 4

Example of computer plot of horizontal average temperature versus horizontal average pressure. The time in seconds and the surface velocity in cm/s indicate the model number of the calculations.

Fig. 5

Example of computer plot of horizontal average density versus horizontal average temperature.

Fig. 6

Example of computer plot of horizontal average radial velocity component (assumed to be the pulsation velocity) versus horizontal average temperature.

Fig. 7

Example of computer plot of radiative luminosity versus horizontal average temperature.

XI. ASSORTED TOPICS

A. Time Step

The time step is restricted primarily by the Courant condition, which is determined in the main program. The time step is further restricted if the condition of Eq. (12) is not satisfied for the deepest zone in which the horizontal velocity is allowed to be nonzero. The time step may also be reduced if the initial temperature corrections were "too large" on the previous time step. This criterion is tested just after the Courant condition has been computed.

B. Static Model

The static model is generated in hydrostatic and thermal (radiative) equilibrium by a separate program (STATPCS). The conditions that must be met are

$$\frac{1}{\rho} \frac{dP}{dr} = - \frac{GM_r}{r^2} , \quad (90)$$

$$- \frac{4\sigma}{3\kappa\rho} \frac{dT^4}{dr} = L , \quad (91)$$

$$\frac{dM_r}{dr} = 4\pi r^2 \rho . \quad (92)$$

Of course, these equations must be integrated using the same difference expressions that are in the hydrodynamic code. Given the boundary conditions at the surface, the simultaneous inward integration of Eqs. (90)-(92) is relatively straightforward. The input variables are the composition, luminosity, effective temperature, and mass. The boundary conditions are

$$T_n = 4\sqrt{2} T_{\text{eff}} , \quad (93)$$

$$M_{n+1/2} = M , \quad (94)$$

$$\rho_n = \rho_0 \quad (95)$$

where ρ_0 is a small but otherwise arbitrary number and n is the surface zone number. The first two conditions are obvious, given the surface boundary conditions of the hydrodynamic integration, and the third replaces setting the ex-

terior zone in hydrostatic equilibrium. The reason hydrostatic equilibrium cannot be obtained in the last zone is discussed in Ref. 5 and lies in the fact that any reasonable difference equation approximation to Eq. (90) takes the form

$$\frac{2}{\rho_n} \frac{c \rho_n T_n}{r_{n+1} - r_n} = g, \quad (96)$$

where c is some constant. The temperature and gravity are effectively determined by the boundary conditions and the density cancels out, leaving the zoning to satisfy Eq. (93). The zoning that usually results from this condition is much larger than the zoning just inside the surface. Such a difference in zoning leads to numerical problems in the hydrodynamical calculation.

C. Stability

I would like to reemphasize the stability criterion imposed by the viscosity. This feature will bring out the importance of the parameter DRMIN, the minimum radial zoning allowed. This parameter is defined in a DATA statement in subroutine RSTEP. An equation analogous to Eq. (12) for the radial zoning restriction caused by the viscosity is

$$\Delta t < \text{constant} \frac{(\Delta r)^2}{\nu}, \quad (97)$$

where the constant may be a function of velocity and perhaps other quantities. Since Eq. (97) is not computed in the program [it is not entirely accurate, as can be seen from the derivation of Eq. (12)], numerical stability will depend on the smallest value of Δr allowed, i.e., of DRMIN. If one allows DRMIN to decrease beyond a certain point, one must also decrease the viscosity coefficient c_0 . However, by reducing Δr , ν has already been reduced [see Eq. (7)], so that to obtain the same physical viscosity, c_0 must be increased. Consequently, the value of DRMIN must be chosen so that reasonable spatial resolution, the required amount of eddy viscosity, and numerical stability can all be achieved. It is conceivable that there are models for which all three cannot be obtained.

D. Concluding Remarks

The version of the two-dimensional code presented here has been devised for the particular application of the study of convection in RR Lyrae models. Presumably, the code will work for other types of variable stars, although one can-

not guarantee that some feature not present in RR Lyrae models might not cause difficulties.

The program is being modified continually. A number of changes have been incorporated since the writing of this report began. The careful reader will also note a few technical inconsistencies and fossils of negligible importance, whose removal would result in a neater program.

These are, of course, remarks that apply to any active code. Ultimately, the user must insure his understanding of the program so that he can insert corrections for the specific problem he wishes to solve.

REFERENCES

1. R. F. Christy, "The Calculation of Stellar Pulsation," *Rev. Mod. Phys.* 36, 555-571 (1964).
2. R. G. Deupree, "On the Treatment of Convection as a Nonradial Stellar Pulsation," *Astrophys. J.* 198, 419-423 (1975).
3. A. N. Cox, R. R. Brownlee, and D. D. Eilers, "Time-Dependent Method for Computation of Radiation Diffusion and Hydrodynamics," *Astrophys. J.* 144, 1024-1037 (1966).
4. R. F. Christy, Methods of Computational Physics (Academic Press, New York, 1967), p. 191.
5. R. G. Deupree, "Nonlinear Convective Motion in Shallow Convective Envelopes," *Astrophys. J.* 205, 286-294 (1976).
6. C. W. Hirt, A. A. Amsden, and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *J. Comput. Phys.* 14, 227-253 (1974).
7. C. W. Hirt, E. D. Nichols, and N. C. Romero, "SOLA - A Numerical Solution Algorithm for Transient Fluid Flows," Los Alamos Scientific Laboratory report LA-5852 (April 1975).
8. H. M. Ruppel and L. D. Cloutman, "A General Numerical Fluid Dynamics Algorithm for Astrophysical Applications," Los Alamos Scientific Laboratory report LA-6149-MS (November 1975).
9. L. D. Cloutman, C. W. Hirt, and N. C. Romero, "SOLA-ICE: A Numerical Solution Algorithm for Transient Compressible Fluid Flows," Los Alamos Scientific Laboratory report LA-6236 (July 1976).
10. L. D. Cloutman, personal communication.
11. J. W. Deardorff, "A Numerical Study of Three-Dimensional Turbulent Channel Flow at Large Reynolds Numbers," *J. Fluid Mech.* 41, 453-480 (1970).

12. J. W. Deardorff, "The Use of Subgrid Transport Equations in a Three-Dimensional Model of Atmospheric Turbulence," J. Fluids Eng. 95, 429-438 (1973).
13. L. G. Henyey, J. E. Forbes, and N. L. Gould, "A New Method of Automatic Computation of Stellar Evolution," Astrophys. J. 139, 306-317 (1964).
14. L. D. Landau and E. M. Lifshitz, Fluid Mechanics (Pergamon Press, London, 1959), pp. 52-54.
15. J. P. Cox and R. T. Giuli, Principles of Stellar Structure (Gordon and Breach, New York, 1968).
16. K. E. Ziebarth, unpublished Ph.D. thesis, University of Colorado, Boulder, CO.
17. B. Paczynski, "Envelopes of Red Supergiants," Acta Astron. 19, 1-22 (1969).
18. R. F. Stellingwerf, "Modal Stability of RR Lyrae Stars," Astrophys. J. 195, 441-466 (1975).

```

PROGRAM STATPCS(INP,FSET5=INP,OUT,FSET6=OUT,FSET7)
COMMON/KAPA/FAP(51,31)
DIMENSION RHO(50),T(50),ZMR(50),R(50),P(50)
DATA TH,FUNT,ONT,UN,ZERO,ONEHF,G,FR,TYP,T1,T2,T3,ZMSTAR,ZLSUN,
1  RSUN / 2., 1.E-13, 0.1, 1., 0., 0.5, 6.668E-8, 4., 1., 1.05,
2  0.95, 10., 2.E+33, 4.E+33, 6.96E+10 /
RT42= SQRT( SQRT(T4))
FORTHMP=4./3.*3.1415926536
T4=3./16./3.1415926536/5.669E-5
I=0
READ (5,305) X,Y
305 FORMAT (2F8,5)
K2=0
300 CONTINUE
I=I+1
IF (I.GT.51) GO TO 304
301 FORMAT (1X,I5,14F5,2)
K2=K2+1
READ (5,301) K1,(FAP(I,J),J=1,14)
IF (K1.NE,K2) GO TO 302
K2=K2+1
READ (5,301) K1,(FAP(I,J),J=15,28)
IF (K1.NE,K2) GO TO 302
K2=K2+1
READ (5,301) K1,(FAP(I,J),J=29,31)
IF (K1.NE,K2) GO TO 302
GO TO 300
302 CONTINUE
303 FORMAT (1X,21MWRONG OPACITY CARD,K=,I3)
WRITE (6,303) K2
CALL EXIT
304 CONTINUE
READ 2,ZL,TE,ZMS
ZMS=ZMS*ZMSTAR
ZL=ZL*7LSUN
RSUN=1.
READ 1,M123
READ 1,N
1 FORMAT (I2)
DO 10 I=1,N
2 FORMAT (5X,E12,6,2X,E12,6,2X,E12,6)
READ 2,R(I),T(I),RHO(I)
R(I)=R(I)*RSUN
WRITE (6,2) R(I),T(I),RHO(I)
10 CONTINUE
NP1=N+1
DR=R(N)-R(N-1)
DO 15 I=NP1,50
ZMR(I)=ZMS
R(I)=R(I-1)+DR
RHO(I)=ZERO
T(I)=ZERO
15 P(I)=ZERO
I=N
T(N)=TE/RT42
C IF SURFACE CONDITION HAS TROUBLE, JUST SET RHO(N) EQUAL TO SOME VALUE
RHO(N)=5.E-4
CALL TERMO (RHO(N),T(N),X,Y,TYP,P(N),OPR,DPT)

```

```

      ISSET=0
      IF (ISSET.EQ.0) GO TO 500
C   SET SURFACE CONDITION
      ITER=0
      ITMAX=50
      RJP=ONEHF*(R(N)+R(NP1))
      DR=R(NP1)-R(N)
      RHO0=5.E-12
      RHOMIN=RH00
      FAC=ONEHF*DR*G*ZMS/RJP**2
117  CALL TERMO (RHO0,T(N),X,Y,TYP,P0,OPR,DPT)
      FZ=P0-RHO0*FAC
      FK=OPR=FAC
      DRHO=FZ/FK
      FK=ABS(DRHO/RHO0)
      IF (FK.LT.ON) GO TO 119
      ITER=ITER+1
      RHO0=RHOMIN+ITER*ONT*RHO0
      IF (ITER.GT.ITMAX) STOP
      GO TO 117
119  RHO0=RHO0+DRHO
      IF (FK.GT.FUNT) GO TO 117
      RHO(N)=RHO0
      P(N)=P0
      WRITE (6,3) R(N),RHO(N),T(N),P(N)
C   SURFACE CONDITION IS SET
500  CONTINUE
      ZM=ZMS
      ZM1=ZERO
20   RJP=ONEHF*(R(I)+R(I+1))
      RJM=ONEHF*(R(I)+R(I-1))
      FAC=T4*ZL/RJM**2
      ZM1=ZM1+FORTHP*(RJP**3-RJM**3)*RHO(I)
      ZMR(I)=ZMS-ZM1
      INDEX=1
      IF (I.LT.M123) GO TO 21
      RHO0=RHO(I-1)
      T0=T(I-1)
      GO TO 25
21   RHOMIN=ONT*ONT*RHO(I)
      TMIN=T(I)
      IF (TMIN.LT.1.2E+4) TMIN=1.5E+0*TMIN
      IF (T(I).LT.TE) TMIN=T(I)
      L1=1
      L2=1
      RHO0=RHOMIN
      T0=TMIN
22  CONTINUE
      L1=L1+1
      IF (L1.LT.20) GO TO 24
      L2=L2+1
      L1=1
      IF (L2.GT.c0) STOP
24  CONTINUE
      RHO0=RHOMIN+L1*ONT*RHO0
      T0=TMIN+L2*ONT*T0
25  CONTINUE
      FZ=G*(ZMS-ZM1)/RJM**2*(R(I)-R(I-1))

```

```

30 CALL TERMO (RH00,T0,X,Y,TYP,P0,DPR,DPT)
  B1=P(I)-P0+ONEHF*(RHO(I)+RH00)*FZ
  FK=FKAP(RH00,T0,RHO(I),T(I))
  FKR=T3/RH00*(FKAP(T1+RH00,T0,RHO(I),T(I))-FKAP(T2+RH00,T0,RHO(I),
1  T(I)))
  FKT=T3/T0*(FKAP(RH00,T1+T0,RHO(I),T(I))-FKAP(RH00,T2+T0,RHO(I),
1  T(I)))
  B2=FK*(T(I)**4-T0**4)/(R(I)-R(I-1))+FAC
  A11=-OPR+ONEHF*FZ
  A12=-DPT
  A21=FKR*(T(I)**4-T0**4)/(R(I)-R(I-1))
  A22=FKT*(T(I)**4-T0**4)/(R(I)-R(I-1))+FK*FR*T0**3/(R(I)-R(I-1))
  DET=A11*A22-A21*A12
  C11=A22/DET
  C22=A11/DET
  C21=-A21/DET
  C12=-A12/DET
  DRHO=-C11*B1-C12*B2
  DT=-C21*B1-C22*B2
  FACKP=ABS(DT/T0)
  FACK=ABS(DRHO/RH00)
  INDEX=INDEX+1
  IF (INDEX,LT,30) GO TO 35
  WRITE (6,3) RHO(I),T(I),P(I),FACKP,FACK,I
  IF (INDEX,GT,60) STOP
35 CONTINUE
  IF (FACK,GT,FACKP) FACKP=FACK
  IF (FACKP,LT,ON) GO TO 40
  GO TO 22
40 CONTINUE
  RAT=ONT/FACKP
  IF (RAT,GT,UN) RAT=ON
  RH00=RH00+RAT*DRHO
  T0=T0+RAT*DT
  IF (FACKP,GT,FUNT) GO TO 30
  I=I-1
  RHO(I)=RH00
  T(I)=T0
  CALL TERMO (RH00,T0,X,Y,TYP,P(I),DPR,DPT)
  WRITE (6,3) R(I),RHO(I),T(I),P(I),FACKP,I
3  FORMAT (5E14,6,2X,I3)
  IF (I,GT,1) GO TO 20
  RJP=ONEHF*(R(1)+R(2))
  RJM=R(1)-(RJP-R(1))
  ZM1=ZM1+FORTHF*(RJP**3-RJM**3)*RHO(1)
  ZMCORE=ZMS-ZM1
  ZMR(1)=ZMCORE
  DO 50 I=2,N
  RJ=ONEHF*(R(I)+R(I-1))
  DR=R(I)-R(I-1)
  FK=FKAP(RHO(I-1),T(I-1),RHO(I),T(I))
  ZLUM=FK/T4*(T(I)**4-T(I-1)**4)/DR*RJ**2
  WRITE (6,4) R(I-1),T(I-1),RHO(I-1),P(I-1),FK,ZMR(I-1),ZLUM
4  FORMAT (7(1X,1PE14,6))
50 CONTINUE
  WRITE (7,11) (I,R(I),T(I),RHO(I),ZMR(I),I=1,N)
11 FORMAT (I2,4E19,12)
  WRITE (7,11) N,ZMS,ZMCORE

```