W.W. Jones ES74

MATHEMATICAL MODEL FOR THE BRIDGMAN-STOCKBARGER CRYSTAL GROWING SYSTEM

Final Report
RAI-86-B-1

Period Covered: 8/20/84 - 9/19/86

Contract: NAS8-36190

Prepared for:
George C. Marshall Space Flight Center
MSFC, AL 35812

Prepared by:
Glyn O. Roberts

Date: 9/19/86

Roberts Associates, Incorporated
1726 Pine Valley Drive
Vienna, VA 22180

ROBERTS ASSOCIATES, INCORPORATED
1726 Pine Valley Drive
Vienna, VA 22180-2341

Phone (703)938-1757
FTS (202)938-1757

19 September 1986

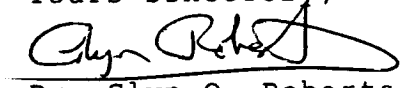
NASA/MSFC
Procurement Office

Subject: Final report on Contract NAS8-36190

Dear Sirs,

We are pleased to present the subject report. The attached distribution list includes that required by the contract. The report completes all tasks required on this contract.

Yours sincerely,



Dr. Glyn O. Roberts
President

Distribution:
AP29-F (0)
CN24D (5)
AT01
EM13A-27
ES74
NASA/STIF (1 + repro.)
W.W. Fowlis, ES73

ABSTRACT

In a major technical breakthrough, a computer model for Bridgman-Stockbarger crystal growth has been developed. The model includes melt convection, solute effects, thermal conduction in the ampoule, melt, and crystal, and the determination of the curved moving crystal-melt interface. The key to the numerical method is the use of a nonuniform computational mesh which moves with the interface, so that the interface is a mesh surface. In addition, implicit methods are used for advection and diffusion of heat, concentration, and vorticity, for interface movement, and for internal gravity waves. This allows large time-steps without loss of stability or accuracy.

No previous model has included the time-dependent curved interface or the effect of concentration changes on the melting-point. The best prior results excluded the ampoule temperature, and included the concentration only as a passive scalar in a calculation done after the flow and temperature solutions had been obtained.

Numerical results are presented for the interface shape, temperature distribution and concentration distribution, in steady-state crystal growth. Solutions are presented for two test cases using water, with two different salts in solution. The two diffusivities differ by a factor of ten, and the concentrations differ by a factor of twenty. The salts depress the freezing point, and the equations of state and state diagrams imply that at equilibrium, the concentration in the solid is one tenth that in the liquid. As a result, the melting temperatures in the two cases are approximately minus twenty and minus one centigrade, varying with the concentration along the interfaces. For the two cases considered, evolution to a steady state was reasonably rapid.

TABLE OF CONTENTS

CHAPTER 1

SUMMARY

1.1 THE BRIDGMAN-STOCKBARGER METHOD

In the Bridgman-Stockbarger method for crystal growth, a long cylindrical ampoule of melt is lowered from a relatively hot cylindrical furnace to a cooler cylinder, with the two cylinders separated by a short insulating cylinder. A crystal forms at the bottom of the ampoule, and grows upwards. Further details are given in Chapter 3.

Crystal quality is affected by the speed of growth, by the curvature of the interface, and by convection and other fluid motions in the melt.

Rapid crystal growth leads to poor quality even for single-component melts, due to dislocations. With multiple components, there is an increased concentration of an expelled component ahead of the advancing crystal interface. When the growth is too rapid and the temperature gradient is too small, there is an instability, and dendrites form at the interface.

The curvature of the interface at any low growth rate is largely determined by the heat flux and temperature distribution. This is in turn determined by the external boundary conditions, the thermal conductivities of the melt, crystal, and ampoule, and any convective heat flux in the melt. Also, the interface is not an isotherm if there are significant flow-induced variations in the concentration.

Convection is determined by horizontal gradients of temperature and concentration in the melt. Vertical variations of the concentration arise from the component expelled during crystallization, and may be gravitationally stabilizing or destabilizing. Convection is of course effectively eliminated in Spacelab operations, but is a dominant effect in many terrestrial applications.

Convection in the melt can be modified and reduced by rotating the ampoule. This introduces the Coriolis force and centrifugal buoyancy forces (Fowlis and Roberts, 1986, Roberts et al., 1984). Another option for electrically conducting melts is the use of magnetic damping. A third option to reduce convection in the melt, and eliminate turbulence, is the use of some kind of porous plug, kept ahead of the advancing crystal interface by a mechanical means.

1.2 OUTLINE

In Chapter 2 we describe the objectives of this modeling program, and the relationship of the present model with an earlier simplified model, with NASA needs and with prior and future experimental and theoretical work. The earlier model neglected convection and solute effects, but included thermal conduction in the crystal, melt and ampoule, and the determination of the curved moving crystal-melt interface. We have now upgraded the model to include convection and solute effects.

Chapter 3 extends the above background description of the Bridgman-Stockbarger method.

In Chapter 4, we describe the problem solved by the upgraded model. This chapter includes the full equations and boundary conditions.

Chapter 5 describes the spatial representation, and includes the transformed coordinates which allow us to use central differences efficiently on a nonuniform staggered moving mesh.

In Chapter 6 we describe numerical algorithms for the time representation of three simplified problems. The purpose of this chapter is to describe the different components of our time representation as they apply in simpler cases.

Chapter 7 presents our time representations of the equations. We use an implicit form of a leapfrog method, in which in alternate stages the motion is updated and the temperature, concentration and interface position are updated.

In Chapter 8, we describe the input data for the code. The input data page format is presented. The input is in four groups, defining in succession the problem parameters, the material properties, the method parameters, and the output control parameters.

Chapter 9 describes operation of the code. We use it routinely on our own microcomputer and on MSFC's VAX minicomputers. Operation is highly automated, using command procedures.

Chapter 10 presents steady-state numerical results for two test cases with solute but no flow. We have demonstrated excellent convergence with adequate resolution in VAX computer runs of between 10 minutes and an hour. This is because of our efficient algorithms and implicit methods.

References are listed in Chapter 11.

Appendix A is a listing of the code.

CHAPTER 2

OBJECTIVES

2.1 NASA NEEDS

The difficulty and expense of laboratory and Spacelab investigations, and the problems associated with diagnostic observations and measurements for certain Bridgman-Stockbarger systems, necessitate parallel modeling efforts. Analytic studies have been helpful, but are of limited scope.

Existing computer models are not adequate. Separately, they explain certain features of the method and of the experimental results, but they lack the generality required for a quantitative understanding of present and planned future crystal growth experiments. Further, the numerical methods of the more sophisticated models require excessive computer storage and time even for cases with limited resolution, no ampoule, and no concentration effects on the density or the interface temperature.

A good example is the work of Chang and Brown (1983a,b,c), who use a finite-element method which determines only steady states. The spatial resolution is limited because of the use of Newton's iteration, requiring the inversion of a large full matrix at every iteration. Doss et al. (1984) discuss the use of finite elements for this type of problem. Miller (1986) reports a study of vapor transport in a cylinder. He includes convection, but has no ampoule conduction or curved interface.

2.2 PREVIOUS PROGRAM

Under a previous contract, we developed a Bridgman-Stockbarger code which computes steady-state conditions for the crystal-melt interface and the temperature distribution in the melt, crystal, and ampoule (Roberts, 1984).

The previous program had three basic objectives. The numerical results demonstrate that these objectives were achieved.

First, the model provides a valid description for cases where solute effects and convection can be neglected. In particular, this applies to zero-gravity experiments with a single component or with a very small solute concentrations.

Secondly, the previous program was a natural first step towards the long-term objective of developing a complete model. Many of the model components were incorporated in the present code, including the non-orthogonal mesh transformation, and the input and output routines.

Thirdly, the previous model can be used as a diagnostic tool for interpreting and understanding the results of a full simulation. This is a significant benefit, since the time-step limit for a full dynamical computation is likely to be shorter than for the previous code.

2.3 LONG-TERM OBJECTIVES

Our first objective is to build a complete model, with the flexibility to describe a full range of Bridgman configurations and materials. This model would involve a simultaneous calculation of the coupled evolution in time of the interface, the flow and solute concentration in the melt, and the temperature distribution in the melt, crystal and ampoule. This objective has largely been achieved, although there are of course a number of code improvements and extensions required.

Our next long-term objective is to apply the code to a wide range of experimental conditions, with different geometries and thermal boundary conditions, different materials, and with and without gravity. This would allow us to validate the code against experimental measurements and to extend the usefulness of those experiments. It would also suggest further experiments, both in the laboratory and in space.

Our third objective is to apply the code usefully in the improvement of manufacturing processes. Once a code like this has been developed and validated, it can be applied operationally to a wide range of conditions as a part of process optimization.

Further details are given in the viewgraphs on the following pages, and in the rest of the report.

ORGANIZATION: RAI	MARSHALL SPACE FLIGHT CENTER	NAME: G. O. ROBERTS
CHART NO.: BACKGROUND - 1	CENTER DIRECTOR DISCRETIONARY FUND PROJECT NUMBER 84-22	DATE: AUGUST 1986

OBJECTIVES

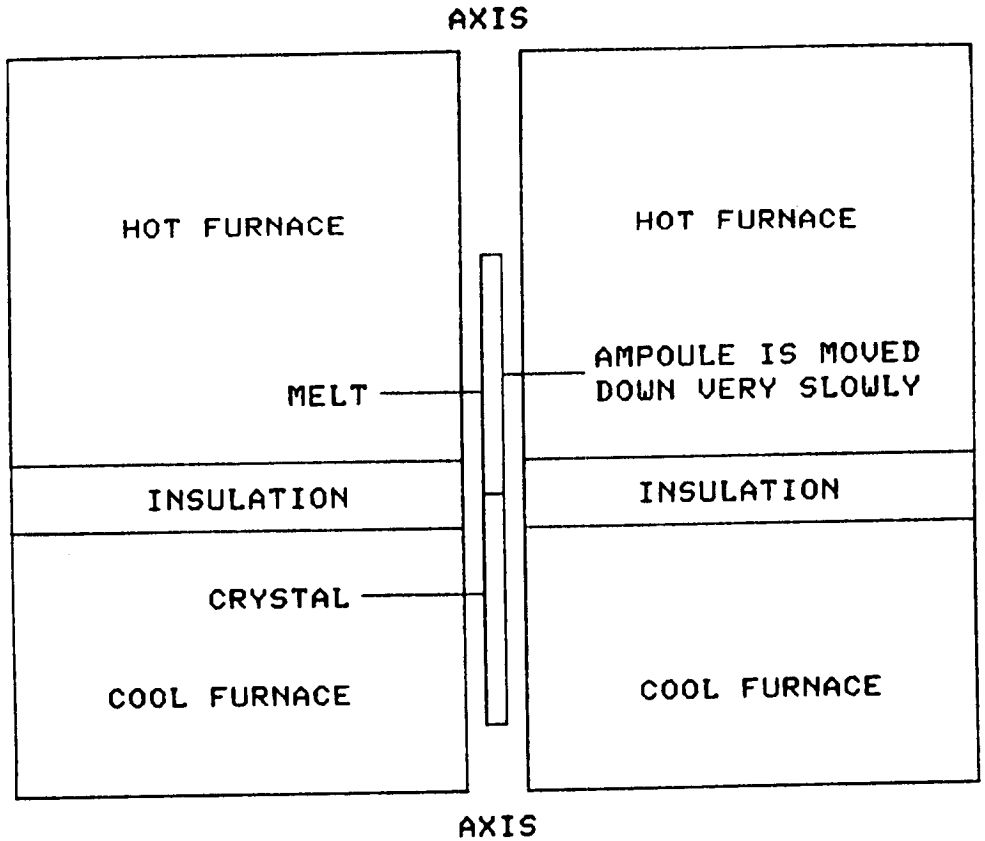
QUANTITATIVE MODEL OF BRIDGMAN CRYSTAL GROWTH

- THE BRIDGMAN SYSTEM IS WIDELY USED FOR MAKING HIGH QUALITY SINGLE CRYSTALS
- THERE ARE PROBLEMS WITH DISLOCATIONS AND INHOMOGENEITIES, PARTLY DUE TO CONVECTION
- THE EQUATIONS ARE NONLINEAR, AND QUANTITATIVE STUDY REQUIRES NUMERICAL MODELS
- EXISTING MODELS ARE HELPFUL, BUT LACK THE REQUIRED GENERALITY, POWER AND EFFICIENCY
- DETAILED OBJECTIVES
  - ★ INTERACTION WITH EXPERIMENTS
    - INTERPRETATION
    - SUPPORT OF PLANNING
    - VERIFICATION
  - ★ DESIGN OF SPACELAB AND SPACE STATION EXPERIMENTS
    - MODEL PROPOSED CONFIGURATIONS
    - DISCOVER POTENTIAL PROBLEMS
    - DEMONSTRATE PROBABILITY OF SUCCESS
    - DEMONSTRATE POTENTIAL FOR COMMERCIALIZATION

ORGANIZATION: RAI	MARSHALL SPACE FLIGHT CENTER	NAME: G. O. ROBERTS
CHART NO.: BACKGROUND - 2	CENTER DIRECTOR DISCRETIONARY FUND PROJECT NUMBER 84-22	DATE: AUGUST 1986

OBJECTIVES(continued)

MODEL THE BRIDGMAN APPARATUS



ORGANIZATION: RAI	MARSHALL SPACE FLIGHT CENTER	NAME: G. O. ROBERTS
CHART NO.: BACKGROUND - 3	CENTER DIRECTOR DISCRETIONARY FUND PROJECT NUMBER 84-22	DATE: AUGUST 1986

OBJECTIVES(continued)

MODEL BRIDGMAN PROBLEMS, CAUSES AND PROPOSED SOLUTIONS

- CRYSTAL DISLOCATIONS AND INHOMEGENEITIES
  - DISLOCATIONS SPREADING INWARD FROM AMPOULE
  - RADIAL SOLUTE SEGREGATION
  - STRIATIONS LIKE ROCK STRATA
- LIKELY CAUSES
  - CONCAVE INTERFACE, SO DISLOCATIONS PROPAGATE INWARD FROM AMPOULE
  - INTERFACE SLOPE CAUSES RADIAL SEGREGATION OF EXPELLED SOLUTE
  - CONVECTION NEAR FURNACE, DUE TO HEAT TRANSFER
  - CONVECTION NEAR MELT, DUE TO EXPELLED COMPONENT AND TO AMPOULE HEAT CONDUCTION AND INTERFACE SLOPE
  - TURBULENT CONVECTION CAUSES UNSTEADY CONDITIONS, REMELTING, AND STRIATIONS
- PROPOSED SOLUTIONS
  - THERMAL BOUNDARY CONDITIONS TO CONTROL CURVATURE
  - ZERO GRAVITY ELIMINATES CONVECTION
  - ROTATION REDUCES CONVECTION
  - MAGNETIC FIELD REDUCES CONVECTION

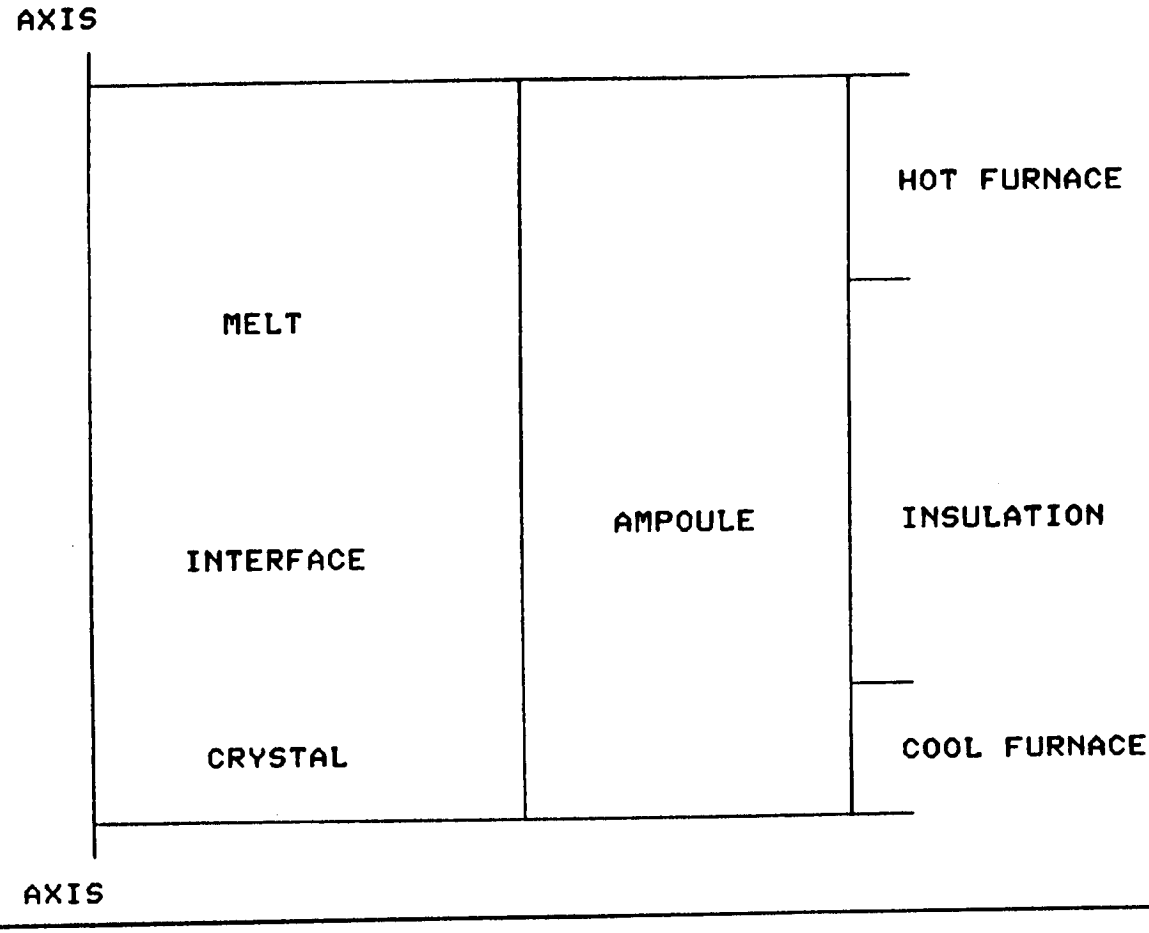
ORGANIZATION: RAI	MARSHALL SPACE FLIGHT CENTER	NAME: G. O. ROBERTS
CHART NO.: BACKGROUND - 4	CENTER DIRECTOR DISCRETIONARY FUND PROJECT NUMBER 84-22	DATE: AUGUST 1986

APPROACH SUMMARY

- A SINGLE CODE WITH GREAT FLEXIBILITY
- INPUT DATA DEFINES
  - PROBLEM
    - DOMAIN GEOMETRY
    - THERMAL FORCING
    - MATERIAL PROPERTIES
    - AMPOULE MOVEMENT
    - GRAVITY, ROTATION AND OTHER EXTERNAL PARAMETERS
    - INITIAL CONDITIONS
    - WHETHER DYNAMICS ARE REQUIRED OR JUST TEMPERATURE
  - METHOD
    - NUMBER OF MESH POINTS
    - NONUNIFORM MESH SPACING
    - TIME STEP
    - IMPLICIT STEPPING CONTROLS
    - METHOD FOR UPDATING THE INTERFACE POSITION
  - OUTPUT
    - NUMERICAL DIAGNOSTICS
    - PRINTER GRAPHICS
    - PLOTTER GRAPHICS

ORGANIZATION: RAI	MARSHALL SPACE FLIGHT CENTER	NAME: G. O. ROBERTS
CHART NO.: BACKGROUND - 5	CENTER DIRECTOR DISCRETIONARY FUND PROJECT NUMBER 84-22	DATE: AUGUST 1986

GEOMETRY OF MODELED REGION



CHAPTER 3

BACKGROUND

3.1 PHYSICAL PROCESSES

In the Bridgman-Stockbarger method for crystal growth from the melt, a cylindrical ampoule containing the well-mixed liquid is lowered steadily down a matched cylindrical opening. It passes in succession through a hotter isothermal region (above the melting point), through an adiabatic zone, and into a cooler isothermal region. Crystallization begins at the bottom of the ampoule, and proceeds upwards as the ampoule descends. For crystals with high melting points, both the isothermal regions are furnaces, separated by a thin insulating slab, and the ampoule is lowered in heat pipes with matching diameter; heat exchange is by conduction and radiation across the air gap.

For definiteness we will describe the mercury cadmium telluride ($Cd_xHg_{1-x}Te$) system, with the small x phase diagram shown in Figure 3.1.

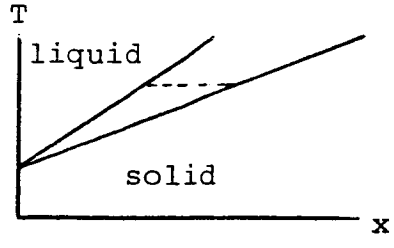


Figure 3.1 Simplified Phase Diagram

The liquid and solid are in equilibrium at the indicated temperature and molar fraction x. For this system, the liquid density decreases with molar fraction x, and for small fixed x it decreases with temperature only above a temperature of maximum density. For larger x, the density always decreases with increasing temperature.

Large-scale longitudinal crystal inhomogeneities arise from the phase diagram and from melt diffusion, and are well understood. Figure 3.2 shows the variation of the transverse average of x along the final crystal, and the variation in the melt at an intermediate stage. The initial molar fraction of the melt is x_o. The bottom of the crystal is cadmium rich, and a cadmium deficiency diffuses ahead of the interface, and finally solidifies at the top.

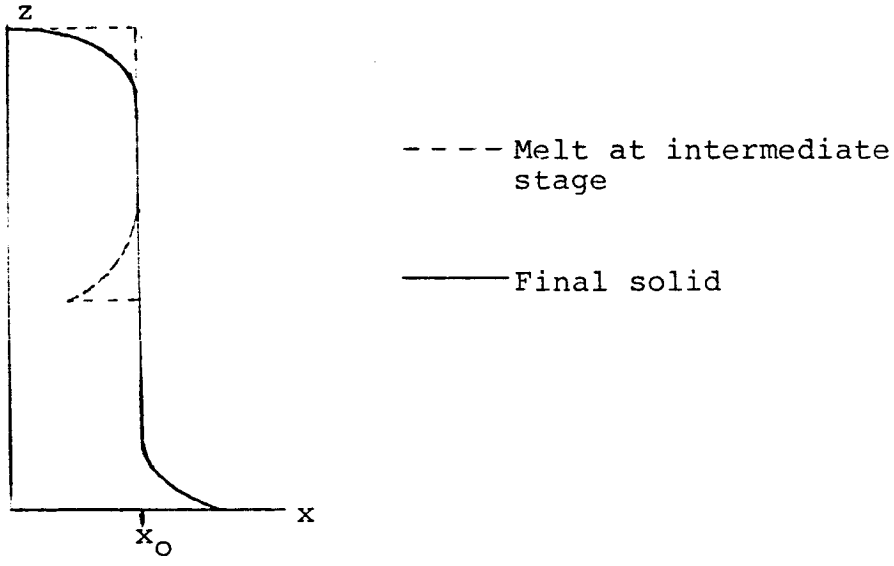


Figure 3.2 Longitudinal Variations in the Horizontally Averaged Molar Fraction

Other crystal inhomogeneities apparently arise from three main sources.

1. Unsteady movement of the interface causes remelting and microlayers in the crystal. This unsteady movement is produced by temperature fluctuations associated with turbulent convection in the melt.

2. Convection near the interface, due to radial temperature gradients, results in radial segregation of the molar fraction.

3. The interface is curved, due to the same radial temperature gradients, and radial diffusion of molar fraction results in radial segregation.

Many methods have been suggested for alleviating these problems and producing better quality crystals.

1. The first two effects involve gravity, and are effectively eliminated in the microgravity of space.

2. Alternatively, the convection can be reduced and the turbulence eliminated by
   - o Rotation, using the Coriolis force;
   - o Magnetic damping, requiring an electrically conducting melt and an external magnetic field; or
   - o A plug of porous material which is somehow kept ahead of the interface.

3. The turbulence in the first effect mainly involves the radial temperature gradients near the upper isothermal region. Radial temperature gradients at the upper furnace can be reduced by insulating the sides and letting the heat come in from the top; which would be a substantial modification from the conventional design.

4. The radial temperature gradients near the interface are inevitable; they are associated with the finite conductivity of the ampoule and the differing conductivities of the melt and crystal. The effect can be minimized by choices of the isothermal region temperatures relative to the melting point, and by careful control of the temperature on the ampoule boundary.

Even if inhomogeneities in the crystal are minimized, crystal dislocations can be a problem. For most materials, they tend to form at the ampoule, and to propagate normal to the interface as the crystal grows. They can therefore be minimized by ensuring that the growing crystal has a convex interface with the melt.

3.2 PREVIOUS MODELS

Axial segregation is readily modeled analytically, assuming uniformity in the transverse directions, and neglecting convection. The resulting ordinary differential equation predicts the profile shown in Figure 3.2, in agreement with measurements.

Analytic models have been developed for interface stability. These models confirm the observation that when freezing is too rapid, and the temperature gradient is too small, the molar fraction cannot

diffuse sufficiently fast ahead of the interface. When

$$dT/dx = (dT/dz) / (dx/dz)$$

is smaller than the upper slope in Figure 3.1, the interface can become unstable, and dendrites form.

Analytic and numerical models have been developed for the thermal conduction problem and the interface shape, in the absence of convection. These models either include ampoule conduction or parameterize it. Many use the isothermal-adiabatic-isothermal furnace structure which we have described.

Apparently no modelers have yet developed a full model, with ampoule conduction, concentration variations, an unknown curved interface determined by the equation of state and not necessarily an isotherm, and with the density a general function of concentration and temperature. This was our objective in the present program.

The most advanced previous model was that of Brown and his co-workers. This steady-state model neglects the ampoule, and neglects the effect of concentration variations on the melting point and on the density. Thus the concentration variations are passive, and are computed separately once the convection flow field is known. In this approximation, the interface is an isotherm. Their Newton algorithm gives only steady-state solutions, and the computer storage and time requirements are respectively proportional to the square and cube of the number of unknown mesh point or node values. This severely limits the resolution which can be used, with given computer resources.

CHAPTER 4

DEFINITION OF THE PROBLEM

4.1 DOMAIN

We confine attention to axisymmetric solutions. We also neglect transients at the ends of the ampoule. We therefore take (r,z) coordinates fixed to the furnace, with the ampoule and crystal moving down at velocity

$$\underline{v} = (0, -W_A) \ .$$

Here the z-axis is upward, and W_A is positive. We are interested both in steady and unsteady solutions, and we therefore use the full time-dependent equations.

The domain is illustrated in the following figure. The heat equation is solved simultaneously in the crystal, melt, and ampoule. The flow and concentration equations are applied only in the melt. The crystal-melt interface is

$$F(r,z,t) = z - f(r,t) = 0 \ ,$$

where the function f(r,t) is of course to be determined. The sample and ampoule radii r_S and r_A are input constants. The results should be effectively independent of the computational boundaries z_B and z_T; this can be tested by varying the position of these boundaries.

The normal to the interface is in the direction of ∇F. The speed of the interface normal to itself can be written as

$$-\dot{F}/|\nabla F| \ .$$

The normal velocity of the melt or crystal, relative to the interface, towards the melt, is

$$(DF/Dt)/|\nabla F| \ .$$

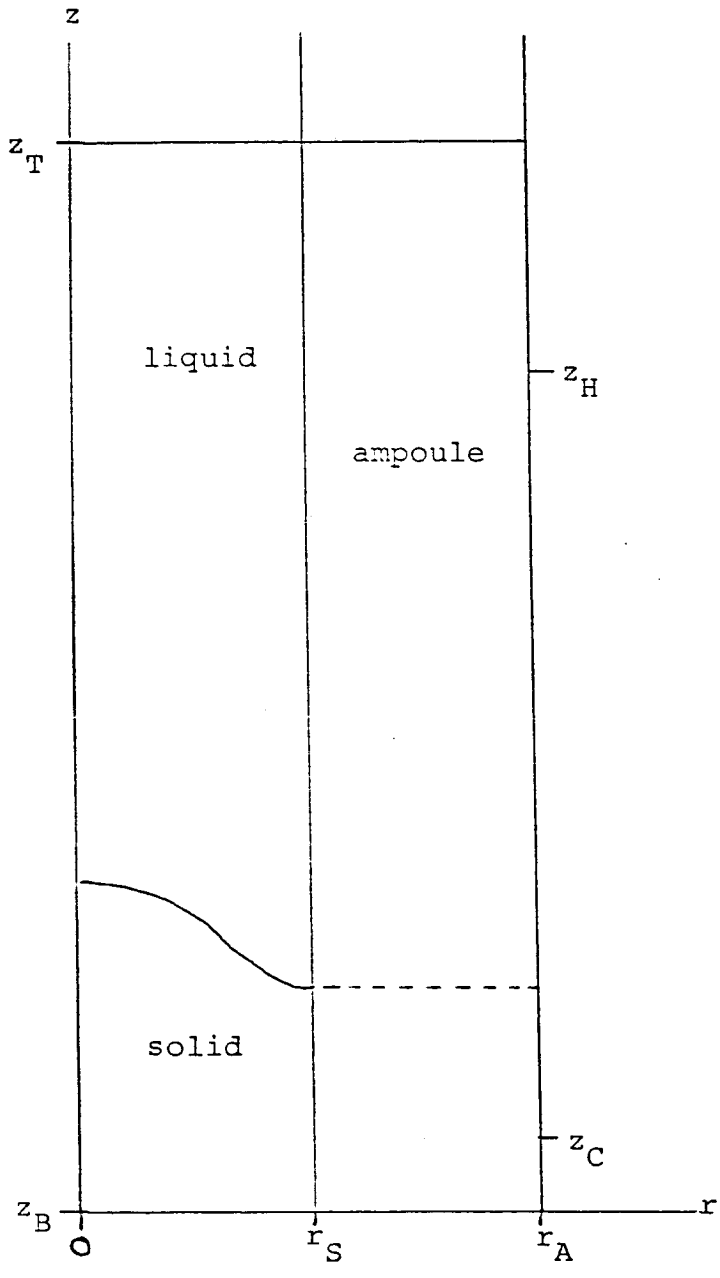


Figure 4.1 Computational Domain

Since the velocity is in general discontinuous at the boundary, the (DF/Dt) values are different. These results are used in obtaining several of the following boundary conditions.

4.2 EQUATIONS AND BOUNDARY CONDITIONS

The full thermodynamic and hydrodynamic equations for mixtures are quite complicated. Detailed discussions are given by Rosenberg (1979) and by Landau and Lifshitz (1960, p. 219ff). However, for a very wide range of practical Bridgman-Stockbarger cases, it is possible to use a much simpler formulation without significant error. In view of the complications associated with the phase change, the unknown interface, and the complicated geometry, we have adopted this approach.

4.2.1 Continuity

We use the Boussinesq approximation, in which density variations in the melt are neglected except in the buoyancy term. This simplifies the treatment of the continuity equation and pressure gradient terms by allowing us to use a vorticity and stream-function formulation. No significant physics is lost. We retain a constant imposed density density difference between the crystal and the melt, as this is frequently of order 10%, and may determine a significant mean flow modification in the melt.

The continuity equation in the melt becomes

$$\nabla . \underline{v} = 0 \ .$$

This allows us to write the velocity $\underline{v}$ at a point in terms of the stream-function ψ as

$$r\underline{v} = (-\partial_z \psi, +\partial_r \psi) \ ,$$

where the stream-function can be interpreted physically as the volume flux passing upward between the axis and a circle about the axis passing through the point, per radian in the azimuthal direction.

4.2.2 Equation of Motion and Vorticity Formulation

The equation of motion is

$$\rho_c D\underline{v}/Dt = - \nabla p + \rho_o \nu \nabla^2 \underline{v} + g\rho \ ,$$

where ρ_o is the constant mean density, ν is the kinematic viscosity, and ρ is the density as a function of temperature and concentration.

The vorticity equation in the conservative form

$$D(\omega/r)/Dt = \nabla.r^{-2}\nu\nabla(\omega r) + \nabla.r^{-1}(g/\rho_o)\rho\hat{r} ,$$

is obtained by taking the curl of the equation of motion; this eliminates the pressure gradient term. We have introduced the unit vector $\hat{r}$ to draw attention to the conservation properties of the vorticity equation. The vorticity is given by the equation

$$\omega = \partial_z u - \partial_r w .$$

In terms of u,

$$\omega/r = - \nabla.r^{-2}\nabla u .$$

A small additional term in these equations arising from viscosity variations has been neglected, although we allow the viscosity to vary with temperature and concentration.

The boundary conditions on the motion at the computational boundary at the top are artificial. We use conditions representing the uniform downward flow at W_A, together with a Poiseuille flow driven by the rate of solidification at the interface. Thus we impose the values of ω, u and w, as follows,

$$w = - W_A + A(1 - r^2/r_s^2) ,$$

$$u = (A-W_A)r^2/2 - Ar^4/4r_s^2 ,$$

$$\omega = 2Ar/r_s^2 .$$

The amplitude A of the Poiseuille flow is determined by the solidification rate mass flux and the continuity condition

$$r_s^2(A/4 - W_A/2) = (1-\rho_c)\int \dot{f} r dr - \rho_c W_A r_s^2/2 ,$$

and is

$$A = (1-\rho_c)[2W_A + (4/r_s^2)\int \dot{f} r dr] .$$

Here ρ_c is the ratio of the crystal density to the melt density, and is an input constant.

On the axis,

$$u = 0 ,$$

$$w = 0 .$$

This is just a choice of the arbitrary constant in u, together with the condition that the axis is not a singularity of the flow.

On the ampoule, we impose the value of u and of its radial derivative. The value is determined as on the top boundary,

$$u = r_s^2(A/4 - W_A/2) .$$

The radial derivative is

$$\partial_r u = -rW_A ,$$

to satisfy the no-slip condition.

The first boundary condition at the interface is the condition of zero slip tangential to the surface. This takes the form

$$[v] \times VF = 0 ,$$

which can be reduced to the equation

$$(rW_A + \partial_r u)\partial_r f - \partial_z u = 0 .$$

The second boundary condition is that the mass flux across the interface is continuous. If the densities are different, this implies a discontinuous normal flow. The boundary condition is

$$[\rho DF/Dt] = 0 ,$$

which reduces to the equation

$$-\rho_c(\dot{f} + W_A) = -\dot{f} + (\partial_r u + \partial_r f \partial_z u)/r .$$

Note that on multiplying by rdr and integrating along the interface, this gives a result consistent with the earlier expressions for A and for u on the ampoule.

4.2.3 The Temperature Equation

The temperature equation in the crystal, melt, and ampoule is

$$c_p DT/Dt = \nabla . K\nabla T ,$$

where c_p is the thermal capacity per unit volume at constant pressure. The thermal conductivity K naturally has separate formulations in the melt, crystal, and ampoule.

Our external thermal boundary conditions are

$$T = T_T \qquad \text{at } z = z_T ,$$

$$T = T_B \qquad \text{at } z = z_B ,$$

$$T = T_T \qquad \text{at } r = r_A , \; z > z_H ,$$

$$T = T_B \qquad \text{at } r = r_A , \; z < z_C ,$$

$$o\ T = 0 \qquad \text{at } r = r_A , \; z_C < z < z_H .$$

We also have the option, determined by an input parameter, of replacing the last boundary condition by the following one,

$$T = T_\ell(z) \qquad \text{at } r = r_A , \; z_C < z < z_H ,$$

where $T_\ell(z)$ is a linear interpolation between T_T at z_T and T_B at z_B. Other options for the external boundary conditions could be added, to increase the flexibility of the code.

The above conditions allow the option of taking $z_T < z_H$, so that the hot furnace is above the domain, and the upper boundary temperature is a calculated value. The same option is available at the bottom.

The interior temperature boundary conditions are the continuity of temperature and heat flux, modified by the latent heat release. In these equations, [T] denotes the difference between the melt or ampoule value and the value on the crystal or sample side. They take the form

$$[T] = 0 \qquad \text{at } r = r_S ,$$

$$[T] = 0 \qquad \text{at } z = f ,$$

$$[K\partial_r T] = 0 \qquad \text{at } r = r_S ,$$

$$[U\rho DF/Dt - K\nabla T.\nabla F] = 0 \qquad \text{at } z = f .$$

In the last equation, U is the internal energy per unit mass. From the mass continuity boundary condition, the factor $\rho DF/Dt$ is continuous. But [U] is the latent heat per unit mass. So

$$L_H DF/Dt - \nabla F.[K\nabla T] = 0 \qquad \text{at } z = f .$$

Here L_H is the latent heat per unit volume of melt, a function of the temperature at the interface, and DF/Dt is evaluated in the melt.

4.2.4 The Concentration Equation in the Melt

Our concentration variable x is strictly the mass fraction, or the ratio of the mass density of one species to the total density. It can also be interpreted as the molar fraction, with x molecules of one species to (1-x) of the other. With typical magnitudes, the error is not significant, provided the problem parameters and material properties are appropriately adjusted.

The concentration equation in the melt is

$$Dx/Dt = \nabla.\alpha\nabla x \ ,$$

where α is the diffusivity.

The concentration boundary conditions are

$$x = x_T \qquad \text{at } z = z_T \ ,$$

$$\partial_r x = 0 \qquad \text{at } r = r_S \ ,$$

where x_T is an imposed top inflow value. The interface boundary conditions arising from the equation of state are

$$x = x_m(T) \qquad \text{at } z = f+ \ ,$$

$$x = x_c(T) \qquad \text{at } z = f- \ .$$

Here the functions $x_m(T)$ and $x_c(T)$ are the equation of state functions, illustrated in Figure 3.1. The conservation of solute mass boundary condition is

$$[x\rho DF/Dt - \rho\alpha\nabla x.\nabla F] = 0 \qquad \text{at } z = f \ .$$

In this equation, the inside of the bracket is the normal mass flux of the solute. Note that α is zero in the crystal. From mass continuity, the term $\rho DF/Dt$ is the same on each side. Thus the boundary condition can be rewritten as

$$[x]DF/Dt - \alpha\nabla x.\nabla F = 0 \qquad \text{at } z = f \ ,$$

where DF/Dt and $\alpha\nabla x$ are evaluated in the melt. The solute deficiency (or excess) resulting from the production of crystal with a larger (or smaller) concentration than the melt, diffuses away into the melt.

Note that there is effectively one boundary condition too many at the interface. The extra boundary condition determines the interface position and its time evolution.

4.2.5 Crystal Concentration, Remelting, and Slush

The concentration equation in the crystal, with no diffusion, is

$$Dx/Dt = 0 \ .$$

The general solution is

$$x = X(r, z+W_A t) \ ,$$

where the function X is determined by the the crystal concentration $x_c(T)$ at the interface, provided there is no remelting.

If there is remelting, a single value of $(r, z+W_A t)$ will correspond to two connected points (r,z,t) on the interface, with in general distinct values of $x_c(T)$. Thus our system of equations is inconsistent for this case.

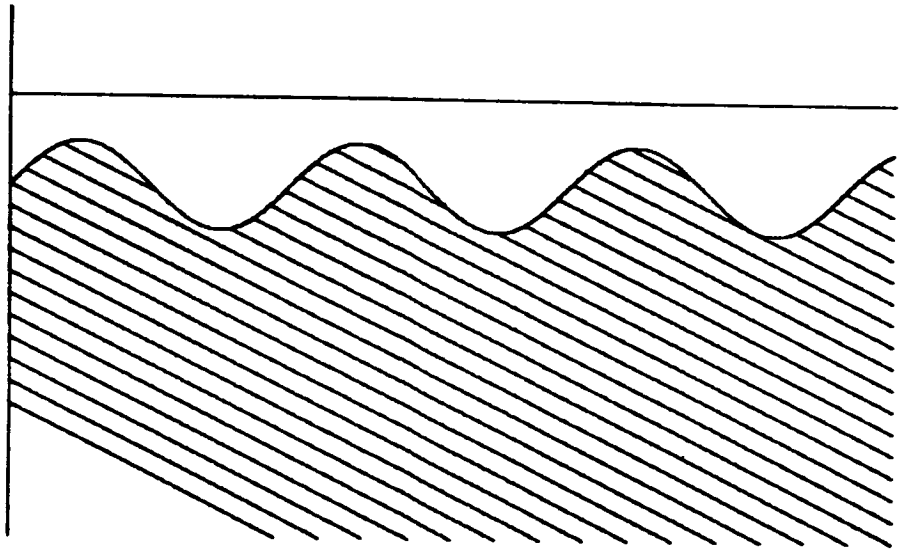


Figure 4.2 Periodic Remelting of the Interface

This is illustrated by the figure, which displays the interface position f as a function of time, for some fixed radius r. The lines with slope $-W_A$ are characteristics, with constant concentration x. This concentration is determined by the value at the first intersection of the characteristic with the interface. If the interface moves down with a speed greater than W_A, then there is

remelting. The interface then intersects characteristic lines on which the concentration x is already known, and is inconsistent with the local value of $x_e(T)$ determined by the conditions at the interface.

This inconsistency arises in our system of equations because of our neglect of the slush obtained when crystal is melted.

The following figure illustrates a typical phase diagram. The central region is excluded. Thus at the indicated fixed temperature the solid and liquid can exist in equilibrium at the two corresponding concentrations. For many binary mixtures, both curves have negative slopes, or one has a very steep or infinite slope, but the situation is still very similar.

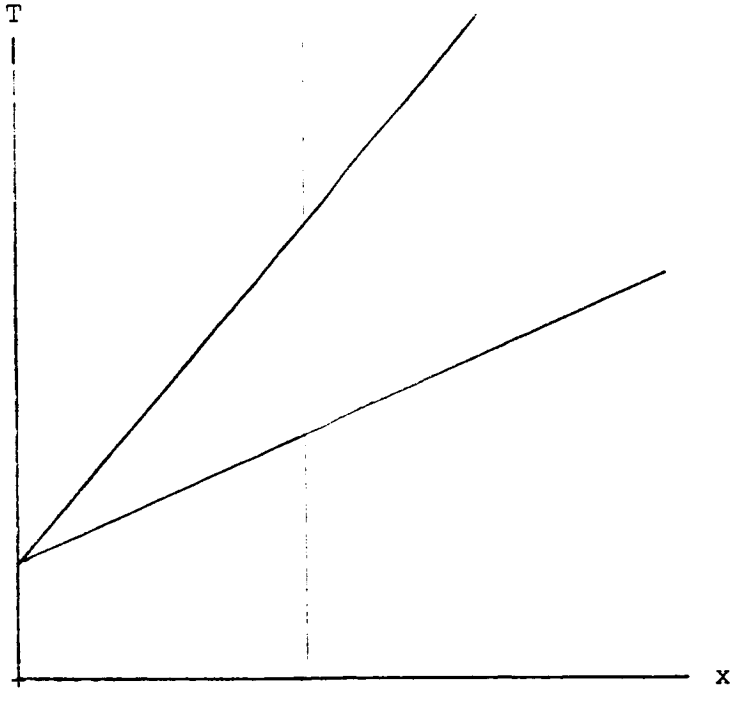


Figure 4.3 Typical Phase Diagram to Illustrate Remelting and Striations

When melt is cooled at the indicated fixed concentration, one of three things can happen, as follows.

1. A finely-dispersed suspension of amorphous solid particles or micro-crystals in the melt can form ("slush"), with the solid proportion gradually increasing as the temperature is lowered. The concentration in the resulting solid is highly nonuniform on a microscale.

2. A crystal can form at a higher concentration, with the corresponding concentration deficiency diffusing away into the melt. This, of course, is the objective in Bridgman-Stockbarger crystal growth.

3. When the growth rate is too large, the concentration deficit cannot diffuse away fast enough. There is then an instability at the advancing crystal interface, resulting in the formation of "dendrites", a forest of tiny crystals growing out into the melt. In practice, this alternative is very similar to the first.

When a crystal is warmed at the same fixed concentration, there is only one possible outcome, the formation of a finely-dispersed suspension of amorphous particles or micro-crystals. The proportion of solid in this slush gradually decreases until it is all liquid. The concentration in the liquid will then be nonuniform, but unless there has been significant gravitational settling in the slush, diffusion should reduce or eliminate this non-uniformity in a relatively short time.

If there is remelting in a model computation, the model will still predict the crystal concentration and its time evolution at any radius. The solution is correct, provided the crystal happens to have exactly the right concentration distribution as it advances into the melt. Clearly this is an unphysical requirement; the crystal concentration is determined by the conditions when it formed, and not by the conditions when it remelts.

In Bridgman-Stockbarger crystal growth, if remelting occurs, the resulting crystal will be striated, and will probably contain refrozen slush in thin microlayers.

Our objective in this program is to determine conditions for the growth of acceptable crystals, rather than to analyze in detail the crystal structure when there is remelting. We have not therefore included a parameterization of possible slush regions in our model. Our model therefore correctly predicts the onset of remelting, but not its outcome.

4.3 MATERIAL PROPERTIES

For many relevant materials, the properties vary significantly over the relevant ranges of temperature and concentration. We have therefore made all of our material properties general functions of temperature and/or concentration. They are flexibly-defined polynomials in the present formulation, with their form defined by input constants as described in Chapter 8. Additional options could

be added as required.

CHAPTER 5

SPATIAL REPRESENTATION

Our system of equations is a difficult one for numerical solution, for the following reasons.

1. The crystal-melt interface is curved, time-dependent, and unknown.

2. There are three separate coupled computational regions.

3. The temperature and concentration equations are coupled by the interface conditions.

4. The very low diffusivity for concentration, and the possible rapid motion driven by convection, imply thin boundary layers and associated computational problems.

5. The widely different diffusivities for heat and solute can lead to double diffusive flows. Particularly troublesome numerically are thin cold ascending plumes ("salt-fingers"), buoyant because of the solute, and not stabilized by the temperature gradient because heat diffuses in from the sides as they rise.

6. The large kinematic viscosities and thermal diffusivities of relevant materials can restrict the time step for most methods.

7. For most cases with convection, there are regions with strong stratification and large Vaisala (internal wave) frequency, further restricting the time step for most methods.

8. Experimental times are frequently very long, with low ampoule speeds.

Efficient numerical methods are therefore essential.

SPATIAL REPRESENTATION

Our spatial representation is described in this chapter. Some of our choices were influenced by the requirements of the time representation, as described in the following chapter.

5.1 CHANGE OF COORDINATES

We use a uniform mesh, with unit mesh spacing, applied to the transformed coordinates i and k. Here r is a function of i, chosen to resolve the possible thin boundary layers. Similarly, z is a function of k and of f(r,t). This function is chosen so that k is constant on the interface, and varies rapidly in boundary layers on the interface or possibly elsewhere. Since f(r,t) is changing, the mesh is not fixed in terms of the r and z coordinates.

The function f(r,t) is itself represented by its values at discrete times and half-odd-integer values of i.

5.1.1 Radial Mesh Transformation

At compilation time, the number of i intervals allocated to the sample (NR) and to the ampoule (NRA) are specified. In the r interval for the sample, we take the mesh spacing dr/di as proportional to the function

$$(r + a)(r_s + a - r) ,$$

where the a values are distinct input constants, estimates for the layer thicknesses. We use a similar expression for the ampoule r interval. These equations are readily integrated to give r and dr/di as functions of i, evaluated at the integers and at half-odd integers. The i values for the axis, sample boundary, and ampoule boundary are 1, 1+NR, and 1+NR+NRA.

Derivatives with respect to r can then be represented as derivatives with respect to i, using finite differences with unit mesh spacing. Thus for half-odd-integer i,

$$\begin{aligned} \delta_r f &= d_r i \delta_i f \\ &= d_r i [f(i+1) - f(i-1)]/2 , \\ &= d_r i \delta_i \bar{f}^i . \end{aligned}$$

Note that f is defined at the half-odd integers. For integer i,

$$\delta_r f = d_r i [f(i+1/2) - f(i-1/2)] ,$$

$$= d_r i \delta_i f \ .$$

The central differencing and averaging operators $\delta_i f$ and $\overline{f}^i$ are defined by the above equations, and will be used extensively.

5.1.2 Value Of f In The Ampoule

The value of f is required in the ampoule, in order to define the mesh. It is a function of time alone, equal to the value on the sample boundary. We obtain this value by extrapolation of the three sample f values adjacent to the boundary. The r derivative is obviously zero.

Second order formulations are used to obtain $\partial_r f$ in the sample, at the boundary and at the half point adjacent to the boundary. Clearly, it would be invalid to use a central difference formulation which involved the f values in the ampoule.

5.1.3 Axial Mesh Transformation

Our transformation uses two input parameters, z_I and L_I. Here z_I is a nominal mean position for the interface position f, and the mesh is approximately Cartesian for distances from the interface greater than L_I.

We take k as increasing downwards from the top, because some Fortran compilers only allow lower limits of unity for arrays, and many of our functions are defined only in the melt.

We first introduce a function ζ of k only, equal to z_T, z_I, and z_B, respectively, for k values of 1, 1+NZ, and 1+NZ+NZS. At compilation time, the number of k intervals allocated to the melt (NZ) and to the crystal (NZS) are specified. Our expressions for ζ as a function of k are then analogous to the calculation of r as a function of i, as described above. Thus in the ζ interval for the melt, from z_T to z_I, we take the mesh spacing $d\zeta/dk$ as proportional to the function

$$(\zeta - z_I + a)(z_T + a - \zeta) \ ,$$

where the a values are again distinct input constants. We use a similar expression for the crystal interval, from z_I to z_B. These equations are readily integrated to give ζ and $d\zeta/dk$ as functions of k, evaluated at the integers and at half-odd-integers.

We next write

$$z = \zeta(k) + \eta(k)\{f(r,t) - z_I\} .$$

For k values in the melt (1 to 1+NZ), we take

$$\eta = \{(z_T - \zeta)/(z_T - z_I)\}/\{1+(\zeta - z_I)/L_I\} ,$$

so that η is one at the interface and is zero at the top. The L_I term makes η small for $|\zeta - z_I| > L_I$. For k values in the crystal (1+NZ to 1+NZ+NZS), we take

$$\eta = \{(\zeta - z_B)/(z_I - z_B)\}/\{1+(z_I - \zeta)/L_I\} .$$

Then

$$z = f \qquad \text{for } \zeta = z_I ,$$

$$z = z_T \qquad \text{for } \zeta = z_T ,$$

$$z = z_B \qquad \text{for } \zeta = z_B ,$$

$$z \cong \zeta \qquad \text{for } |\zeta - z_I| >> L_I ,$$

$$z \cong f + \zeta - z_I \qquad \text{for } |\zeta - z_I| << L_I .$$

The inequalities demonstrate the purpose of L_I, which is to confine the mesh distortion effect of f to a limited region near the interface.

For bad input choices of z_I or L_I, $\partial_k z$ can become unacceptably small as f changes. The code avoids this by appropriate adjustments to z_I and L_I in the early stages of the computation, ensuring that L_I is greater than an imposed multiple of the maximum of $|f - z_I|$.

Note that z, f, and k form a functionally related triad. The key properties of this triad are

$$\partial_f z = \eta = (\partial_z f)^{-1} ,$$

$$\partial_k z = d_k \zeta + (f - z_I) d_k \eta = (\partial_z k)^{-1} ,$$

$$\partial_k f = -\partial_k z \partial_z f = -(d_k \zeta + (f - z_I) d_k \eta)/\eta = (\partial_f k)^{-1} ,$$

and these properties are used in evaluating the various required derivatives.

5.1.4 Transformation of Partial Derivatives

Care is required in the transformation of partial derivatives, because the z coordinate depends on both i and k. Thus from multivariate calculus,

$$\partial_r T = \partial_r i \partial_i T + \partial_r k \partial_k T \ ,$$

$$\partial_z T = \partial_z i \partial_i T + \partial_z k \partial_k T \ ,$$

Of the four mesh derivatives, only $\partial_z i$ is identically zero. Hence, using the relationships derived above,

$$\partial_r T = d_r i (\partial_i T - \eta \partial_i f \partial_z k \partial_k T) \ ,$$

$$\partial_z T = \partial_z k \partial_k T \ .$$

The partial derivatives with respect to i and k can now be represented directly as central differences.

5.2 STAGGERED COMPUTATIONAL MESH

Functions of i and k are represented by their values at either whole (W) or half-odd-integer (H) values of i and k. Our choices are shown in the following table.

Variable	i	k
ψ	W	W
ω	W	W
u	W	H
w	H	W
T	H	H
x	H	H
c_p	H	H
K	H	H
T_{wt}	H	-
x_m	H	-
x_c	H	-
L_H	H	-
ρ	H	H
α	H	H
ν	H	H
f	H	-

Quantities defined only at the interface are indicated with dashes in the k column. Note that the material properties depend on T and x, and are therefore most conveniently evaluated at those points.

SPATIAL REPRESENTATION

the following descriptions of the spatial representations, we the locations of the various quantities, where necessary, in (WH) for example. The representations then make use of the averaging and differencing operators described above. This he necessity of using subscripts or some other notation to ie mesh points for the representation.

NCENTRATION EQUATION AND BOUNDARY CONDITIONS

ie concentration equation in the melt can be written as

$$\dot{x} + \nabla.(\underline{u}x - \alpha\nabla x) = 0 \ .$$

presentation for the divergence is the flux formulation

$$(\delta_{\iota}C_{\iota} + \delta_{\kappa}C_{k}) \ / \ V \ .$$

is the physical volume per azimuthal radian of the unit cell ed at the concentration mesh point (HH), and C_i and C_k are the crossing corresponding sections of the cell boundaries ıding the part due to the fact that the mesh is moving), defined ctively at the (WH) and (HW) locations. The volume is

$$V = r d_{\iota} r \partial_{k} z \ .$$

actually negative for our mesh; this causes no problems since our are consistent.

The flux formulation has the following three advantages.

1. It reduces computation, because each flux C_i and C_k is applied to the cells on both sides of the cell boundary.

2. It conserves a representation of the integral of concentration. Material is neither created nor destroyed when the resolution is poor.

3. It avoids nonlinear computational instability by conserving a representation of the integral of the square of the concentration under the action of the advection terms.

Note that the time derivative of x in the above equation is taken fixed x and z. Transforming to (i,k,t) coordinates gives

$$\dot{x} - \dot{f}_{\eta}\partial_{z}k\partial_{k}x \ ,$$

and so replacing the k derivative of x by the finite difference,

$$V\dot{x} - rd_i rf\eta\delta_k\bar{x}^k + \delta_i C_i + \delta_k C_k = 0 .$$

5.3.1 Concentration Flux in the i Direction

For the cell boundaries in the i direction, the surface is defined by the equation

$$F(r,z,t) = r - r(i) = 0 ,$$

and the flux can be obtained (as in our discussion of the interface boundary conditions in the previous chapter) as

$$C_i = (r\partial_k z)(xDF/Dt - \alpha\nabla x.\nabla F)/|\nabla F| .$$

Here the first bracket is the area of the boundary segment (per azimuthal radian), and the remainder is the flux per unit area crossing the cell boundary in the direction of VF. Simplifying this expression,

$$C_i = r\partial_k z(xu - \alpha\partial_r x) .$$

The radial derivative is represented using the above equation. The horizontal velocity u is represented in terms of the stream function ψ as

$$r\partial_k zu = -\delta_k\psi .$$

The representation is therefore

$$C_i = -\bar{x}^i\delta_k\psi - \bar{\alpha}^i rd_r i(\partial_k z\delta_i x - \eta\delta_i f\delta_k\bar{x}^{ik}) .$$

5.3.2 Concentration Flux in the k Direction

For the cell boundaries in the k direction, the surface is defined by the equation

$$F(r,z,t) = z - z(k,f(r,t)) = 0 ,$$

and the flux can be written as

$$C_k = (rd_i r|\nabla F|)(xDF/Dt - \alpha\nabla x.\nabla F)/|\nabla F| .$$

Here the first bracket is the area of the boundary segment (per

azimuthal radian), and the remainder is the flux per unit area crossing the cell boundary in the direction of ∇F. Expanding this expression, and dropping the $\dot{f}$ term which comes from the moving mesh,

$$C_k = rd_i r[x(w - \eta u \partial_r f) - \alpha(\partial_z x - \eta \partial_r x \partial_r f)] .$$

The advection coefficient is

$$r(w - u\eta\partial_r f) = \partial_r \psi + \eta \partial_z u \partial_r f$$
$$= d_r i \partial_i \psi .$$

The diffusion bracket is

$$\partial_z k \partial_k x - \eta \partial_r f d_r i (\partial_r x - \eta \partial_i f \partial_z k \partial_k x)$$
$$= \partial_k x \partial_z k [1 + (\eta \partial_r f)^2] - \eta \partial_i x \partial_r f d_r i .$$

The representation is therefore

$$C_k = \bar{x}^k \delta_i u - \bar{\alpha}^k r[\delta_k x d_i r \partial_z k \{1 + (\eta \partial_r f)^2\} - \eta \delta_i \bar{x}^{ik} \partial_r f] .$$

5.3.3 Upwind Differencing Option

The diffusivity α is very small, and for flows of any significant magnitude the upwind diffusion length scale $\alpha/|v|$ is less than the mesh spacing. This can lead to odd-even mesh separation in certain circumstances. In this phenomenon, the concentration values at the odd mesh points appear to represent a different function from those at the even points.

To control such problems, we use an upwind differencing option, implemented in the following form. We can write the above representation of the flux C_k as

$$C_k = \bar{x}^k G_k + D_k \delta_k x + E_k \delta_i \bar{x}^{ik} .$$

We then replace D, which is positive, by the largest of D and $u_x|G_k|$, where u_x is an input parameter, the upwind differencing coefficient for concentration. We make the same substitution for C_i, G_i, and D_i.

Clearly, this change has has no effect if u_x is zero or smaller than the minimum of $D_k/|G_k|$ and of $D_i/|G_i|$. If D_k is small, G_k is positive (representing upward flow), and u_x is 0.5 (the largest useful value) then

$$C_k = G_k x(i,k+1/2) .$$

This explains the name of upwind differencing, since the point (i,k+1/2) is upwind from the mesh point where we are evaluating C_k.

If there is any sign of odd-even mesh separation, a u_x value of 0.1 or less is generally sufficient to remove it. This applies when the effect is a consequence of the spatial representation only, and is steady in time. Numerical instability of the time representation sometimes produces a similar odd-even mesh separation, but it applies to the time dependence as well.

The effect of upwind differencing is fairly easy to understand in one spatial dimension. We do not present a detailed analysis here.

The corresponding options are available for temperature and vorticity, with separate input constants u_T and u_μ. Since the diffusivities are so much larger than α, these options will rarely be needed.

5.3.4 Boundary Conditions on the Concentration

At z_T, we apply

$$\overline{x}^k = x_T \ ,$$

to get the concentration half a mesh point outside the boundary. This is used in getting C_k on the boundary, and C_i just inside the boundary, from the formulas above.

With sufficiently rapid melting or freezing (according to the sign of the factor $1-\rho_c$), the upward velocity w may be negative at the top, for r values sufficiently near the axis. If this occurs, we apply the passive boundary condition

$$\delta_k x = 0 \ ,$$

instead of imposing the concentration. This will avoid mesh separation. Our mesh near the top will normally be inadequate to resolve upwind diffusion of concentration.

At the axis and at r_S, we apply

$$C_i = 0 \ .$$

In addition, we use

$$\delta_i x = 0 \ ,$$

at the axis, to get the concentration half a mesh point outside the

boundary. This is used only in getting the last term of C_k just inside the boundary, from the formula above.

We do not use this symmetry condition to get the external x at the sample boundary r_s, because that would be inaccurate. Instead, for the C_k values just inside the sample boundary, we replace the term $\delta_i \bar{x}^{ik}$ by the one-sided difference $\delta_i \bar{x}^k$, evaluated half a mesh point to the left.

The boundary conditions at the interface are discussed below.

5.4 TEMPERATURE EQUATION AND BOUNDARY CONDITIONS

5.4.1 Relationship with the Concentration Equation

Our representation of the temperature equation is the same as our representation of the concentration equation, with the following exceptions.

1. The functions x and α are naturally replaced by T and K, and V is replaced by $c_p V$.

2. In the advection terms in the fluxes C_i and C_k, with x not differentiated or differenced, x is replaced by $c_p T$.

3. The temperature equation is applied in all three regions, the melt, the crystal, and the ampoule.

4. For the solid regions, the $\delta_k u$ horizontal advection term is dropped from C_i.

5. For the solid regions, the $\delta_i w$ vertical advection term in C_k is replaced by $- rd_i r W_A$.

6. For the top boundary, the same boundary condition is applied, using T_T.

7. For the bottom boundary, the corresponding boundary condition is applied, using T_B.

Additional conditions are required at the interface and at r_s and r_A, as described below.

This representation is not strictly accurate if c_p varies significantly from mesh point to mesh point. The point is minor, since any variation does not play an important role.

5.4.2 Temperature Boundary Conditions on the Ampoule

The boundary conditions can be conveniently represented by imposing $\bar{T}^i$ as T_T or T_B for the hot and cold furnace sections, and setting $\delta_i T$ to zero for the insulated segment in between. If the insulated segment is replaced by the imposed linear temperature gradient $T_\ell(z)$, then $\bar{T}^i$ is set to $T_\ell(z)$.

Because the boundary condition changes discontinuously for the insulating case, and in a z region where the mesh spacing is not necessarily very small, we have introduced an option of smoothing the boundary condition in the z direction. We therefore apply a weighted sum of the three boundary conditions, with weighting functions

$$W_H(z) = [1 + \tanh\{(z-z_H)/s_H\}]/2 ,$$

$$W_C(z) = [1 - \tanh\{(z-z_C)/s_C\}]/2 ,$$

which vary smoothly from 1 in the furnaces to zero elsewhere, on length scales s_H and s_C equal to an imposed multiple of the local mesh spacings $\delta_k z$ at z_H and z_C. The result is

$$(1-W_H-W_C)\delta_i T + W_H(2\bar{T}^i - 2T_T) + W_C(2\bar{T}^i - 2T_B) = 0 .$$

If the imposed multiple is zero, this reduces to the discontinuous boundary condition described above.

This relation is used to obtain T half a mesh point outside the boundary. This is then used in getting C_i on the boundary. Note that C_k just inside the boundary is not affected, since f is constant in the ampoule.

5.4.3 Temperature Boundary Conditions Between the Sample and the Ampoule

The objective here is to use representations of the two boundary conditions to obtain a representation for the common heat flux C_i at the boundary. Note that in general the quantities K, $d_i r$, and $\delta_r f$ are discontinuous at r_S.

We have four equations in four unknowns, at each half-odd-integer k value. The unknowns are the required flux C_i, the temperature T, and the two external temperatures which extrapolate smoothly from the sample and ampoule. The equations can be written as

$$C_i = D_S \delta_i T_S + E_S \delta_k \bar{T}_S^{ik} ,$$

$$C_i = D_A \delta_i T_A ,$$

$$T = \bar{T}_S^i ,$$

$$T = \bar{T}_A^i .$$

Note that E_A is zero in the ampoule, because the interface slope is zero. This system would be straightforward except that the k differencing and averaging in the E_S term couples the unknowns at different k values. So we approximate $\delta_k \bar{T}_S^{ik}$ as $\delta_k \bar{T}^{ik}$, the value obtained using the known temperatures half a point from the boundary on the two sides. Hence

$$C_i = D\delta_i T + E\delta_k \bar{T}^{ik} ,$$

$$D = 2D_S D_A / (D_S + D_A) ,$$

$$E = E_S D_A / (D_S + D_A) .$$

For the two C_i representations on the sample boundary and adjacent to the interface, we use the one-sided difference term $E\delta_k \bar{T}^i$, evaluated half a mesh interval further away from the interface. Also, as with the concentration, we use the one-sided difference $E\delta_i \bar{T}^k$ for C_k at points next to the sample boundary.

5.5 INTERFACE BOUNDARY CONDITIONS

Our representation of the temperature and concentration boundary conditions involves additional unknowns, the temperatures T_I on the interface at the half-odd-integer i values. These determine the corresponding melt and crystal concentrations (from the equations of state $x_m(T)$ and $x_c(T)$) and the interface values of c_p (both sides), K (both sides), L_{II}, and α.

There are special representations of the three interface C_k values (two for heat and one for concentration). For melt concentration,

$$C_k = x_m \delta_i \bar{\alpha}^k + D_k ,$$

$$D = -\alpha r [2(x_m - x_h) d_i r \partial_z k \{1 + (\partial_r f)^2\} - \delta_i \bar{x}_m^i \partial_r f\}] .$$

The diffusion component D_K is referred to below. Note that we have replaced $\bar{x}^k$ and $\bar{\alpha}^k$ by x_m and α, and $\delta_k x$ by $2(x_m - x_h)$, where x_h is the value at the mesh point half a mesh interval from the interface. This is directly analogous to the above formulation for the interface between the sample and the ampoule. An external x value, extrapolated from the melt to half a mesh point below the interface, has been introduced, and then eliminated using the condition,

$$x_m = \overline{x}^k .$$

This same external x value is used and then eliminated in the $\dot{x}$ equation at the HH points next to the interface, in the representation of the $\dot{f}$ term, which includes the expression $\delta_k \overline{x}^k$.

Next to the axis and to the sample boundary, the $\delta_i \overline{x}^i_m$ term is replaced by the one-sided differences $\delta_i x_m/2$ and $\delta_i x$, respectively. The 2 factor reflects the use of axis symmetry, as with C_k in the melt.

The two interface C_k representations for heat are obtained by replacing x by T, multiplying by c_p for advection, and using $2(T_I - T_h)$ and $2(T_k - T_I)$ for $\delta_k T$ on the two sides. The values of $\partial_z k$, K and c_p are of course different on the two sides. The advection coefficient in the crystal is also modified from that in the melt, with $\delta_i \dot{u}$ replaced by $-rd_i r W_A$.

The two additional boundary conditions are the equations for the conservation of material and heat, and can be regarded as determining the interface temperature and movement. For concentration and temperature respectively, they are

$$[x]^m_c \{-\rho_c rd_i r(W_A + \dot{f})\} + D_k = 0 ,$$

$$L_H \{-\rho_c rd_i r(W_A + \dot{f})\} + D_{km} - D_{kc} = 0 .$$

Here D_{km} and D_{kc} are the diffusive heat flux components in the melt and crystal, formulated as described above. The latent heat and solute deficiency or excess are released at the interface, and diffuse away.

Note that the melt volume flux crossing the interface, including the contribution from the interface movement, can be written in the alternative forms,

$$rd_i r(DF/Dt) = \delta_i \dot{u} - rd_i r\dot{f}$$

$$= -\rho_c rd_i r(\dot{f} + W_A) .$$

The representation at (WH) points of C_i for concentration includes a term which can be written as $E\delta_k \overline{x}^{ik}$. Adjacent to the interface, this must be replaced by

$$E(\overline{x}^i_m - \overline{x}^{ik}) ,$$

where $\overline{x}^{ik}$ is evaluated at k=NZ, one point from the interface. Corresponding representations must be used for temperature, just above and below the interface.

In the ampoule, the discontinuity in $\partial_k z$ at k=NZ+1 requires special treatment. In C_k, $\partial_z k$ must be replaced by the reciprocal of the average of the two $\partial_K z$ values.

5.6 VORTICITY EQUATION

The vorticity is represented by its values at the (WW) points in the melt and on its boundary.

Our representation of the vorticity equation is analogous to that for the concentration equation, with a flux formulation which conserves representations of the volume integral of (ω/r) and (under the action of advection alone) of its square. The representation is

$$d_i r \partial_K z \dot{\omega} - d_\iota r \eta \bar{f}^i \delta_K \bar{\omega}^K + \delta_\iota C_\iota + \delta_K C_k = 0 \ .$$

Here the unit cell is centered at the vorticity mesh point (WW), and C_ι and C_K are the fluxes of (ω/r) crossing corresponding sections of the cell boundaries. The fluxes are defined respectively at the (HW) and (WH) locations, and include advection terms, diffusion terms and buoyancy terms.

5.6.1 Advective Fluxes for the Vorticity Equation

Our representations of the advection components of C_i and C_K are analogs of those for concentration, as described above. The mesh locations are changed. We take

$$C_i = - \overline{(\omega/r)}^i \delta_k \bar{u}^{ik} \ ,$$

$$C_K = + \overline{(\omega/r)}^k \delta_i \bar{u}^{ik} \ .$$

Note that averages of $\dot{u}$ must be used in obtaining the volume flux across these cell boundaries.

5.6.2 Buoyancy Term for the Vorticity Equation

The buoyancy contribution to the fluxes C_i and C_K arises from our representation of the radial fluxes across the interface areas, as with concentration, and is

$$C_i = + (g/\rho_0) \bar{\rho}^{iK} \partial_K z \ ,$$

$$C_k = - (g/\rho_0) \bar{\rho}^i \eta \delta_i f \ .$$

5.6.3 Diffusion Term for the Vorticity Equation

Our representation of the diffusion term contributions to the fluxes of (ω/r) across the cell boundaries is again analogous to that used for the concentration equation. We take

$$C_i = -(\overline{v}^k/r)d_{ri}[\delta_{kz}\delta_{i}(\omega r) - \eta\delta_{i}\overline{f}^{i}\delta_{k}(\overline{\omega r})^{ik}] ,$$

$$C_K = -(\overline{v}^{i}/r)[\delta_{K}(\omega r)d_{ir}\delta_{zk}\{1+(\eta\delta_{r}f)^2\} - \eta\delta_{r}f\delta_{i}(\overline{\omega r})^{ik}] .$$

5.7 STREAM-FUNCTION EQUATION

The stream-function is represented by its values at the (WW) points in the melt and on its boundary.

The definition of the vorticity in terms of the stream-function as a divergence allows us to use the representation

$$-V\omega/r = \delta_{i}C_{i} + \delta_{k}C_{k} ,$$

where C_i and C_k are fluxes across the boundaries of the unit cell with volume V, centered on the ψ point. Our representations for C_i and C_k are

$$C_i = r^{-1}d_{ri}[\delta_{kz}\delta_{i}u - \eta\delta_{i}\overline{f}^{i}\delta_{k}\overline{u}^{ik}] ,$$

$$C_k = r^{-1}[\delta_{K}ud_{ir}\delta_{zk}\{1+(\eta\delta_{r}f)^2\} - \eta\delta_{r}f\delta_{i}\overline{u}^{ik}] .$$

5.8 BOUNDARY CONDITIONS ON THE MOTION

There are two boundary conditions on the motion, on the whole of the melt boundary, as presented earlier. The functions ψ and ω are represented by their values at (WW) points, which includes points on the four boundaries.

The top is straightforward, as we have expressions for ψ and ω there,

$$u = (A-W_A)r^2/2 - Ar^4/4r_S^2 ,$$

$$w = 2Ar/r_S^2 .$$

The axis is straightforward, since ψ and ω are zero.

On the boundary of the melt with the ampoule, we impose the values of ψ and of $\partial_r u$ (both constants). Because ψ is constant, there is no k derivative when $\partial_r u$ is transformed to i,k coordinates. Thus our representations are

$$u = r_s^2(A/4 - W_A/2) ,$$
$$\delta_i \bar{\psi}^i = -r_s W_A d_i r .$$

This latter boundary condition determines the value of $\psi(NR+2)$, one point outside the boundary, in terms of the value one point inside. This allows us to evaluate ω on the boundary, using the same formula as in the interior.

The interface conditions must be transformed to i,k coordinates and represented using finite differences applied to ψ. The mass conservation condition becomes

$$\delta_i \psi = r d_i r[(1-\rho_c)\dot{f} - \rho_c W_A] .$$

Thus ψ is obtained on the interface as

$$\psi = (1-\rho_c)\sum \dot{f} r d_i r - \rho_c W_A r^2/2 ,$$

where the sum is over the half points on the interface, from the axis out to the whole point where ψ is being found.

We can now obtain the numerical representation of A, by matching the above expression with u on the ampoule. The result is

$$A = (1-\rho_c)[2W_A + (4/r_s^2)\sum \dot{f} r d_i r] ,$$

where the sum is over the whole interface. This expression corresponds exactly with the analytic expression in the previous chapter.

The no-slip condition then reduces to the form

$$\delta_k \bar{\psi}^k = \partial_k z \partial_r f(rW_A + d_r i \partial_i \bar{\psi}^i) / [1 + (\partial_r f)^2] .$$

This determines a value for $\psi(NZ+2)$, one point below the interface. This in turn allows us to evaluate ω on the boundary, using the representation presented earlier for the stream-function equation.

5.9 SUMMARY OF THE SPATIAL REPRESENTATION

Depending on the resolution, we have a very large set of

equations, determining the rate of change of the temperature, concentration, and vorticity, at each appropriate mesh point, and the position of the interface and its rate of change, at each radial mesh point. The rates of change for the individual temperature, concentration and vorticity values only enter one of the equations. On the other hand, practically every equation involves one or more of the rates of change of the interface positions. In addition, we have equations determining the stream function mesh-point values in terms of the other variables. The total number of equations is of course equal to the number of unknowns.

The equations can be conveniently divided into two groups. The first is concerned with temperature and concentration, with the corresponding boundary conditions, and with the interface. The second group is concerned only with the motion in the melt. Although the interface position and its rate of movement enters into both groups, it is the first group that should be regarded as determining the interface.

The T and x equations involve the neighboring T, x, and ψ values, together with the $\dot{f}$ values at the same radial mesh point and the three neighboring f values. The boundary conditions on the top, bottom, and ampoule boundaries, and on the ampoule-sample interface, are required to close this system. In addition, the boundary conditions at the crystal-melt interface involve the interface values of T and x, the values half a mesh interval above and below, the local f and $\dot{f}$ values, and the f values at the two neighboring mesh points.

The ω equations involve the neighboring ω, ψ, and ρ values, together with the f and $\dot{f}$ values at the two neighboring radial mesh points. The ψ equations only involve the ω value, the eight neighboring ψ values, and the f values at the two neighboring mesh points. The boundary conditions do not introduce additional complications, except that due to the no-slip condition the boundary conditions cannot be used to express ω on the boundary in terms of the neighboring values. Instead, the definition of ω in terms of ψ must be used, in conjunction with the two boundary conditions on ψ.

CHAPTER 6

TIME REPRESENTATIONS FOR SIMPLIFIED PROBLEMS

The main features of our time representation, described in the following chapter, are as follows.

1. Implicit treatment of advection and diffusion in the temperature, concentration, and vorticity equations, using an alternating-direction implicit (ADI) formulation.

2. Solution of simultaneous equations determining the changes in the temperature, concentration, and interface.

3. Implicit treatment of internal gravity waves.

4. ADI iterative solution for the change in the stream-function, with a correct inclusion of the no-slip conditions.

To clarify the methods involved, we consider three simplified problems in this chapter. We then present the temporal staggering of the variables, outline the algorithms, and describe the details for our problem, in the following chapter.

6.1 IMPLICIT ONE-DIMENSIONAL TEMPERATURE AND INTERFACE ALGORITHM

In order to present our method for the simultaneous updating of the temperature, concentration, and interface, we first discuss a simplified problem, in one dimension.

6.1.1 Definition of the Problem

We simplify the problem by assuming no motion or concentration or ampoule, only one dimension (no radial variation), zero melting point and latent heat, and other melt and crystal properties unity.

The equation is simply the one-dimensional diffusion equation

$$\dot{T} = \partial_z^2 T \ ,$$

with the boundary conditions of impcsing the temperature at the top and bottom. The melting interface is passive.

The initial temperature profile should be monotonic, at least near the melting point, so that there is no uncertainty about its location. The numerical solution of this problem is of course trivial.

6.1.2 Structure of the Spatial Representation

We will seek to solve this problem using an unnecessarily complicated mesh which follows the melting interface, as in the previous chapter. The spatial representation becomes

$$\partial_k z \dot{T} - \dot{f}_\eta \delta_k \bar{T}^k - \delta_k (\partial_z k \delta_k T) = 0 \ ,$$

The boundary conditions at the top and bottom are

$$\bar{T}^k = T_T \ ,$$

$$\bar{T}^k = T_B \ ,$$

The interface conditions are similar to those at the sample-ampoule boundary and at the interface in the previous chapter. Note that $\partial_k z$ is discontinuous. We extend the definitions of T_u and T_ℓ to points half a mesh interval the other side of the boundary, and define the interface temperature as the mean,

$$\bar{T}^k_u = \bar{T}^k_\ell = 0 \ .$$

The interface heat flux is the same using the upper and lower variables,

$$(\partial_z k \delta_k T)_u = (\partial_z k \delta_k T)_\ell \ .$$

The following points should be noted concerning this spatial representation.

1. There are NZ+NZS interior temperature points and equations.

2. There are two exterior temperature points, and two temperature boundary conditions.

3. There are two additional unknowns at the interface, and three equations.

4. The temperature equation at each interior mesh point involves $\dot{f}$.

5. The interface temperature condition determines $\dot{f}$.

6. The number of equations matches the number of unknowns.

7. The linear heat equation has been transformed into a nonlinear system by using a mesh which moves with the solid-liquid interface.

6.1.3 Requirements for the Time Representation

We will impose the following two requirements.

1. The method should be unconditionally stable and second order accurate in time, so far as practicable, in order to allow the large time steps required for efficient modeling of prolonged experiments, without significant loss of accuracy. This effectively requires an implicit representation of the Crank-Nicolson type.

2. The method should be readily and efficiently generalized to the full problem. This excludes an iterative method for solving a nonlinear system in the new variables or the changes.

6.1.4 Time Staggering of the Variables and Algorithm Outline

The variables T and f are represented by their values at discrete time steps, with a time interval Δt. A single time step starts with T and f at an initial time, together with the previous change Δf, and replaces them by the new values and the new change Δf.

The following table shows the relative positions of the different variables in time, with time increasing down the page. The primary variables T, f and Δf are specified at the initial level, and half a step back for f.

Each advance of two rows then corresponds to a single time step. The evaluated quantities shown in the table are obtained in order and used to update the primary variables, as outlined below.

PRIMARY VARIABLES	EVALUATED VARIABLES
Δf	
f T	
Δf	f_α $\dot{f}_\alpha$ T
f T	

For each successive time step, the algorithm consists of the following stages.

1. Get the temporary second-order approximation $f_\alpha = f + \Delta f/2$, at the intermediate time level.
2. Get the temporary first-order approximation $\dot{f}_\alpha = \Delta f/\Delta t$, at the intermediate time level.
3. Set up and solve a set of simultaneous linear equations for the change Δf and the changes ΔT.
4. Update f, by adding Δf to the old value.
5. Update T, by adding ΔT to the old value.

Details are given in the following subsection.

6.1.5 Implicit Time Representation

Our representation is

$$\partial_k z \Delta T/\Delta t - \Delta f/\Delta t \eta \delta_k \overline{T}^k - \dot{f}_\alpha \eta \delta_k \overline{\Delta T}^k \beta - \delta_k \{\partial_z k \delta_k (T + \beta \Delta T)\} = 0 .$$

where $\partial_z k$ is evaluated using the approximation f_α at the intermediate time. The parameter β is 0.5 for Crank-Nicolson second-order accuracy.

The representation of the top and bottom boundary conditions and of the interface conditions is straightforward.

This representation is fully second order accurate. The linear system can be solved for the unknowns ΔT and Δf, using the familiar tridiagonal algorithm, appropriately generalized to include the single unknown Δf.

6.1.6 Discussion

In both the illustrative problem and the full problem, the position of the interface is determined by the instantaneous distributions of T (and x for the full problem) near the interface. The motion of the interface and the rate of change of the temperature are directly related.

For very small time steps, this method is essentially equivalent to an explicit method. The time derivatives are determined from the temperatures, concentrations, and interface position at the beginning of the step, and are influenced only by their neighbors.

On the other hand, for very large time steps, the time derivatives play a smaller role in the representation, and we are determining an approximate steady state solution in one step, within the limits of the linearization. Naturally, for very large steps, accuracy in time is limited, but convergence to a steady state is more rapid. Time-stepping with a very large step is a possible method for obtaining steady-state solutions. A more powerful method is related to time stepping, but with time steps which vary from mesh point to mesh point and which are different for different variables.

In the above one-dimensional algorithm, distinct time steps could be used for T and for f, and they could be allowed to vary with position. In particular, larger time steps are appropriate in regions where the mesh is coarse. It is therefore natural to make the time step proportional to imposed powers of $d_i r$ and $d_k \zeta$, with the powers specified by input to the computer code.

6.2 IMPLICIT INTERNAL GRAVITY WAVE ALGORITHM

To illustrate our method for the stabilized treatment of internal gravity waves, we consider a simplified problem, in Cartesian two-dimensional coordinates.

6.2.1 Application to the Linearized Equations

We first consider the linearized internal gravity wave equations, with no diffusion or interface. The equations are

$$\dot{\omega} = \partial_y \rho \ ,$$

$$\dot{\rho} = N^2 \partial_y \psi \ ,$$

$$\nabla^2 \psi + \omega = 0 \ ,$$

where N is the Vaisala frequency of internal gravity waves. Hence

$$\nabla^2 \ddot{\psi} + N^2 \partial_y^2 u = 0 \ .$$

An explicit leapfrog staggered representation of the equations, with time step Δt, leads after elimination of variables to the equation

$$\nabla^2 \delta_t^2 \psi + (N\Delta t)^2 \partial_y^2 \psi = 0 \ ,$$

where δ is the usual central difference operator with respect to time. In this and the following equations, the spatial derivatives are notations for their finite-difference representations. This is only stable if $(N\Delta t)^2 < 4$, which is an unduly restrictive limitation. However, if we use the second-order accurate stream-function equation formulation,

$$(\nabla^2 + aN^2\Delta t^2 \partial_y^2)\psi + \omega = 0 \ ,$$

then on elimination,

$$\delta_t^2(\nabla^2 + aN^2\Delta t^2 \partial_y^2)\psi + N^2\Delta t^2 \partial_y^2 \psi = 0 \ ,$$

which is stable provided

$$N^2\Delta t^2 < 4(1 + aN^2\Delta t^2) \ .$$

This requires

$$a > 1/4 - 1/(N\Delta t)^2 \ .$$

It can be shown that this representation of the eliminated ψ equation is fourth order accurate in time if a is 1/12. It is therefore natural to choose a as the largest of 1/12 and the above value.

If $N\Delta t$ is large compared with unity, this representation is of course poor for the high-frequency waves with the wave fronts almost vertical. However, if there is no significant forcing of these high-frequency modes, the representation can be highly satisfactory.

6.2.2 Generalization to the Nonlinear Equations

Since the above method only involves a slight modification of the Poisson equation for the stream-function, it is readily generalized to the nonlinear problem, and to cases with diffusion. Provided an implicit representation of the Crank-Nicolson type is used for the advection and diffusion terms, the stability criterion derived above remains sufficient.

In our generalization to the full problem, we retain this formulation, and use the Vaisala frequency given by

$$N^2 = - (g/\rho_o) \partial_z k \delta_k \rho \ .$$

The numbers 1/4 and 1/12 are replaced by input parameters, for greater flexibility. Modifications are required if the flow is strong. Further, there may be local regions with a density inversion (N negative); so we use the absolute value in applying the algorithm.

This method was successfully applied by Roberts and Rubenstein (1981) to a two-dimensional problem with nonlinear advection but no diffusion, and with N t values as large as 20. Even larger values may be required with the present code.

6.3 IMPLICIT VORTICITY ALGORITHM FOR A NO-SLIP CARTESIAN PROBLEM

Our system includes the vorticity equation in two dimensions, with a no-slip condition on two boundary segments. Since we wish to use time steps with $\nu\Delta t$ much greater than the square of the smallest mesh interval, we must use implicit methods. To illustrate our method of solution, we consider first a fixed Cartesian geometry and a uniform mesh.

The domain is $|x|<a$, $|y|<b$. The equations are

$$\dot{\omega} + \nabla.(\underline{u}\omega - \nu\nabla\omega) = 0 ,$$

$$\nabla^2\psi + \omega = 0 .$$

The boundary conditions are

$$u = 0 ,$$

$$\partial_n\psi = 0 ,$$

on all four boundaries. There is no direct boundary condition on ω; the boundary values required for the derivatives near the boundary are determined from ψ and the additional boundary condition on ψ.

The spatial representation is straightforward, using a uniform mesh such that the boundaries are mesh surfaces. For ψ, the mesh extends one point beyond the boundaries, to allow a centered difference representation of the boundary condition, and thus to allow the centered evaluation of ω on the boundary.

Our Crank-Nicolson representation of these equations uses an extrapolation from previous steps to obtain the stream function and hence the flow $\underline{u}$ at the middle of the step. We then have

$$[1 + \beta\Delta t(\underline{u}.\nabla - \nu\nabla^2)]\Delta\omega = \Delta t(- \underline{u}.\nabla\omega + \nu\nabla^2\omega) ,$$

$$[-\nabla^2]\Delta\psi - \Delta\omega = 0 ,$$

with the boundary conditions

$$\psi = 0 ,$$

$$\delta_{\hat{}}\psi = 0 \ .$$

The derivatives are now a notation for the natural central difference representations. The parameter β is 0.5 for second order accuracy in time. The right hand side is the known value from the previous step; the changes on the left are the unknowns.

This linear system can be solved iteratively or by a direct method. We have considered the use of an alternating-direction implicit (ADI) iteration, but have adopted a direct method instead.

CHAPTER 7

TIME STEPPING ALGORITHM

In this chapter we present our algorithms for time stepping the equations for the spatial representation presented earlier. These algorithms combine the features illustrated in the previous chapter.

7.1 TIME STAGGERING OF THE VARIABLES AND ALGORITHM SUMMARY

The variables are represented by their values at discrete time steps, with a constant time interval Δt. The motion variables u and ω are represented at times half a step away from those for the variables f, T, and x. Note that these sets of variables are similarly staggered in space, in both coordinate directions.

In addition to initial conditions for f, T, and x, and for u half a step behind, the previous value for Δf and the new value for Δu must be provided, so that second order accuracy can be maintained. It is reasonable to set the changes to zero for initial conditions.

For a restart, this group of distributions must be saved, and reinitialized.

To complete the initialization, it is necessary to compute the material properties, such as the conductivities K.

A complete time step then advances all these variables by Δt, to the values at the next time step.

The following table shows the relative positions of the different variables in time, with time increasing down the page. Each advance of two rows corresponds to a single time step. The first two rows must be specified in the initialization, or from the previous step. The successive stages of a time step are shown in the table and outlined below.

TIME STEPPING ALGORITHM

STEP	PRIMARY VARIABLES	EVALUATED VARIABLES
0	ψ Δf	
0.5	Δu f T x	K
1	ψ Δf	f_α z $\dot{f}_\alpha$ ΔT Δx ω
1.5	$\Delta\psi$ f T x	K z $\dot{f}_\alpha$ ψ_α

Table 7.1 Time Placement of the Primary and Evaluated Variables

For each successive time step, the calculation proceeds as follows.

1. Update u to the new time step, by adding Δu.

2. Get the temporary second-order approximation $f_\alpha = f + \Delta f/2$, at the intermediate time level.

3. Set up the vertical mesh, using f_α. This is indicated in the table by a z on row 1.

4. Get the temporary first-order approximation $\dot{f}_\alpha = \Delta f/\Delta t$, at the intermediate time level.

5. Set up and solve a set of simultaneous linear equations for the changes Δf, ΔT, and Δx. This is easier said than done.

6. Get ω from u.

7. Update f, by adding Δf to the old values. Simultaneously get the second order approximation $\dot{f}_\alpha$, at the new time level, by extrapolating the old and new Δf values.

8. Update T and x by adding ΔT and Δx to the old values.

9. Update the temperature and concentrations at the interface, maintaining consistency with the equations of state.

10. Get the material properties (signified in the table by the single variable K) from T and x. There is an option not to do this every time step, since often they are constants or change very slowly. The buoyancy and the interface concentrations must normally be updated every time step.

11. Set up the new vertical mesh, using f.

12. Extrapolate the ψ values to get ψ_α, for advection of momentum.

13. Get Δu from the vorticity and stream function equations and boundary conditions. Again, this is easier said than done.

Details are given in the following sections.

7.2 TEMPERATURE, CONCENTRATION, AND INTERFACE

7.2.1 Equations

Our time representation of the system of equations for the temperature, concentration, and interface is analogous to the representation of the one-dimensional temperature equation described earlier. We solve a set of simultaneous linear equations for the changes Δf, ΔT, and Δx, and use them to update the interface and the concentration and temperature distributions.

Our spatial representation for the concentration equation is

$$V\dot{x} - rd_i r\dot{f}\eta\delta_k\bar{x}^k + \delta_i C_i + \delta_k C_k = 0 ,$$

$$C_i = -\bar{x}^i\delta_k u - \bar{\alpha}^i rd_r i(\partial_k z\delta_i x - \eta\delta_i f\delta_k\bar{x}^{ik}) ,$$

$$C_k = \bar{x}^k\delta_i u - \bar{\alpha}^k r[\delta_k x d_i r\partial_z k\{1 + (\eta\partial_r f)^2\} - \eta\delta_i\bar{x}^{ik}\partial_r f] .$$

To represent this equation with second order accuracy, at the mid point of the time step, we use the following representations

x	$\Delta x/\Delta t$
f	$\Delta f/\Delta t$
$\partial_k z$	Use f_α to get vertical mesh
$\dot{f}\delta_k\bar{x}^k$	$(\Delta f/\Delta t)\delta_k\bar{x}^k + \dot{f}_\alpha\delta_k(\beta\Delta x)$
ψ	Latest value is at the right time
x	$x + \beta\Delta x$

$f \qquad f + \beta\Delta f$

The resulting nonlinear system is linearized by dropping the nonlinear terms, which are all second order. The resulting equation links nine Δx values, in a 3x3 square surrounding the point, and three Δf values, surrounding the same i point.

The temperature equation representation in the melt, crystal, and ampoule, is similar, with the same differences as were listed in Section 5.4.1. In the ampoule, there is a single unknown Δf_a, which is a linear combination of three sample Δf values.

7.2.2 Boundary and Interface Conditions

The boundary conditions on the concentration at the top, axis, and ampoule are represented using the same replacement for x as given above. The boundary conditions and ampoule-sample interface conditions for the temperature are represented similarly.

In the interface conditions, we replace T_I, x_m, and x_c by $T_I + \beta\Delta T_I$, $x_m + \beta(dx_m/dT)\Delta T_I$, and $x_c + \beta(dx_c/dT)\Delta T_I$.

Thus we obtain a closed linear system, with the same number of equations as unknowns. The system is analogous to the system described in Chapter 6, but of course it is much more complex.

7.2.3 Solution of the System

This system of linear equations is solved by Gaussian elimination. By making use of the sparse character of the matrix, the number of arithmetic operations per mesh point is reduced to approximately the square of (NR + NRA). However, this is still high, for the anticipated resolution requirements. We have obtained a further reduction by decomposing the matrices used to eliminate the unknowns x and T away from the interface. This stage need only be repeated perhaps once every 30 steps.

Iterative methods could be used to solve the system. Since for cases with gravity, the solution of the equation of motion requires a substantial number of iterations at each time step, the use of an iterative method here would not be unreasonable.

We tried to develop a successful iterative scheme for a long period, without success. The failure does not indicate that it is impossible, and we hope to return to this if resolution becomes limited by the demands the present algorithm puts on our computer resources.

7.3 EQUATION OF MOTION

7.3.1 Objectives of the Time Representation

Our overall objectives are unconditional stability and second order accuracy in time. These are ambitious objectives, in view of the following special difficulties associated with the equations of motion:

1. The incompressibility condition, requiring us to solve a Poisson equation for the stream-function;

2. Very strong maximum density stratification in most cases with gravity, requiring implicit treatment of internal gravity waves.

3. The cylindrical geometry, the sloping and moving bottom, and the non-orthogonal time-dependent coordinates;

4. No-slip boundary sections, so that the boundary condition cannot be expressed in terms of vorticity alone.

These objectives cannot be achieved without a fairly sophisticated iterative motion algorithm, as outlined in the following subsections.

7.3.2 The Vorticity Equation

Our time representation of the vorticity equation is partially analogous to the representation of the one-dimensional temperature equation described earlier. We would like to solve the system

$$d_i r \partial_k z \Delta\omega/\Delta t - d_i r \eta \bar{f}^i \delta_k \overline{(\omega+\beta\Delta\omega)}^k + \delta_i C_i + \delta_k C_k = 0 .$$

where C_i and C_k are representations of the fluxes of (ω/r) crossing the respective boundary segments, formulated to be second order accurate in time.

The spatial representations of C_i and C_k were described in three parts, corresponding to the advection terms, the buoyancy terms, and the diffusion terms.

The buoyancy fluxes are

$$C_i = + (g/\rho_o) \bar{\rho}^k \partial_k z ,$$

$$C_k = - (g/\rho_o) \bar{o}^i \eta \delta_i f .$$

The quantities f, z, and ρ (a material property) are already available half way through the step, as shown in the table and algorithm outline above.

The advection fluxes are written as

$$C_i = - \overline{[(\omega+\beta\Delta\omega)/r]}^{i} \delta_k \psi_a^{-ik} ,$$

$$C_k = + \overline{[(\omega+\beta\Delta\omega)/r]}^{k} \delta_i \psi_a^{-ik} .$$

In these equations, ψ_a is obtained by extrapolation from the earlier primary variables, as shown in the table.

The vorticity diffusion flux terms are

$$C_i = - (\bar{\nu}^{k}/r) d_r i [\partial_k z \delta_i (\omega r) - \eta \delta_i \bar{f}^{i} \delta_k \overline{(\omega r)}^{ik}] ,$$

$$C_k = - (\bar{\nu}^{i}/r) [\delta_k (\omega r) d_i r \partial_z k \{1+(\eta \partial_r f)^2\} - \eta \partial_r f \delta_i \overline{(\omega r)}^{ik}] .$$

All terms in these expressions are already available with second order accuracy at the appropriate time, except for (ωr). We replace ω by

$$\omega+\beta\Delta\omega .$$

This representation couples nine $\Delta\omega$ unknowns, in a 3x3 square surrounding the point. Boundary conditions are required to determine $\Delta\omega$ on the four boundaries of the melt. Since boundary conditions on ω directly are not available on the ampoule or the crystal, the boundary conditions have to be considered together with the stream function ψ, as described in the following subsection.

7.3.3 The Stream Function Equation

We take the difference in time of the stream-function equation in Section 5.7, and linearize the right hand side, to obtain an equation for $\Delta\omega$ in terms of the surrounding $\Delta\psi$, plus a nonhomogeneous term. We similarly take the difference of the boundary conditions. This gives a closed system in $\Delta\omega$ and $\Delta\psi$, with the same number of equations as unknowns.

7.3.4 Solution Of The Equations

We have investigated two methods for solving this linear system, iteration (using an alternating direction implicit algorithm) and direct Gaussian elimination (using a banded matrix algorithm to take advantage of the sparse nature of the system). Our attempts at an iterative scheme have not succeeded, although this method is known to work in a Cartesian geometry. We are therefore implementing a direct

solver. We are using an L-U matrix decomposition, which is updated perhaps every 30 steps, for computational economy. The computational effort per mesh point is thus reduced; it would otherwise be of order the square of NR.

CHAPTER 8

INPUT DATA FOR THE CODE

The default data page is shown on the following page. The code is used to create this file, which contains default values. The default values represent a fairly easy model problem, minor variations of which are studied in the following chapter on our numerical results. The default file is edited to specify the required parameter values. The parameters are divided into four groups: problem; material properties; method; and output.

The text at the top left (defaulting to SETUP DATA) is read in as a label, and later printed or plotted with the output, as in the plots shown later.

8.1 THE PROBLEM PARAMETERS

The problem parameters are the first group on the data page, and define the geometry and mesh and the external forcing. The DR and DZ parameters control the non-uniformity of the mesh in each direction and near each boundary and interface; they are the distinct parameters called a in Section 5.1. ZINTI and ZINTL are initial values for z_I and L_I, for the z coordinate transformation.

The top and bottom furnace temperatures apply above and below the corresponding z values, while the ampoule is insulated between them. The temperature boundary condition on the ampoule is smoothed over ZTBSC mesh intervals, as described in Chapter 5. If ZTBSC is zero, the unsmoothed boundary condition is applied. If ZTBSC is -1, this is a flag to impose the linearly interpolated temperature, between the two furnaces, as described in Chapter 4.

WAMP is the downward speed of the crystal and ampoule relative to the furnace. GTERR is the downward gravity; it is zero for Spacelab simulations, and can be negative to simulate cases with the melt below the crystal. CTOP is the inflow concentration x_T at the top of the furnace. DENCBM is the ratio ρ_c of the crystal density to the melt density.

SETUP DATA

PROBLEM PARAMETERS

R MESH

RINS	=	0.000000	DRINS	=	0.100000						
RSAM	=	0.100000	DRSAM	=	0.030000						
RAMP	=	0.200000	DRAMPI	=	0.020000						
			DRAMPO	=	0.060000						

Z MESH

ZT	=	0.200000	DZT	=	0.500000						
ZB	=	-0.200000	DZB	=	0.500000						
ZINTI	=	0.000000	DZINTM	=	0.030000						
ZINTO	=	0.500000	DZINTS	=	0.030000						

EXTERNAL PARAMETERS

TTOP	=	50.000000	ZTTOP	=	0.100000						
TBOT	=	-100.000000	ZTBOT	=	-0.100000						
GTERR	=	980.000000	ZTBSCL	=	0.500000						
WAMP	=	0.001000	CTOP	=	0.020000						
DENCBM	=	0.900000									

MATERIAL PROPERTIES

PROPERTY	UNITS	CONTROL	CONSTANTS							
CPMELT	cal/cc/deg	1	1.000000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CPCRYS	cal/cc/deg	1	0.500000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CPAMP	cal/cc/deg	1	0.100000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CDMELT	cal/cm/sec/deg	1	0.001400	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CDCRYS	cal/cm/sec/deg	1	0.001000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CDAMP	cal/cm/sec/deg	1	0.000800	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
LATHEAT	cal/cc	1	80.000000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CINTM	gm/gm	2	0.000000	-1.00E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CINTC	gm/gm	2	0.000000	-1.00E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
BUOYANCY	gm/gm	12	0.000000	-1.00E-04	1.00E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
DIFFUSIVITY	cm2/sec	1	0.000100	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
VISCOSITY	cm2/sec	1	0.010000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

METHOD PARAMETERS

NSTEP	=	50	LC	=	F	LU	=	F			
TAUS	=	0.500	TAUA	=	0.500						
TAUT	=	0.500	PRT	=	0.000	PZT	=	0.000			
TAUC	=	0.500	PRC	=	0.000	PZC	=	0.000			
TAUU	=	0.500	PRU	=	0.000	PZU	=	0.000			
TAUF	=	0.500	PRF	=	0.000	BEF	=	0.700			

ADVECTION

TUPWND	=	0.000									
CUPWND	=	0.050									
UUPWND	=	0.000									
NUF	=	10									

FIXING CONTROLS

FIXTR	=	T	FIXTZ	=	T						
FIXCR	=	T	FIXCZ	=	T						
FIXUR	=	T	FIXUZ	=	T						
NTF	=	10	NCF	=	30						

FIXING AMPLITUDES

BETTA	=	0.700	BETTD	=	0.700						
BETCA	=	0.700	BETCD	=	0.700						
BETUA	=	0.700	BETUD	=	0.700						
AQURTN	=	0.250	ATWLV	=	0.083						

OUTPUT PARAMETERS*

DIRECT ACCESS READ & WRITE SEGMENTS

ISEGR = 0 ISEGW = 1

(0: ANALYTIC, -1: AFTER, PRIOR CASE)

WRITE STEP BEGINNING & INCREMENT

IBEGDA = -1 IINCDA = 9900

(-1 MEANS NSTEP)

DIAGNOSTIC FREQUENCY

NDIAG = 0.

CONTOUR LINES

NCLP = 0. NCLI = 0.

RSTRPL = 1.0 ZBPL = 0.00 ZTPL = 0.00 NCOPIES = 0

ISTEP BEGIN & INCREMENT FOR PRINTER DIAGNOSTICS

RANGES FOR I AND K

PLOTTER IDADUT BEGIN & INCREMENT

VARIABLE/I		PHYSICAL	CONTOUR	INTEGER	CONTOUR	PRINT	NUMBERS	IB	IE	II	KB	KE	KI	PHYSICAL		INTEGER		INCR
MESH	1	9901	9902	9903	9904	9905	9906	2	25	1	2	51	1	9913	9914	9915	9916	0.0
TEMP	2	9901	9902	9903	9904	9905	9906	2	25	1	2	51	1	9913	9914	9915	9916	0.0
DT	3	9901	9902	9903	9904	9905	9906	2	25	1	2	51	1	9913	9914	9915	9916	0.0
CP	4	9901	9902	9903	9904	9905	9906	2	25	1	2	51	1	9913	9914	9915	9916	0.0
COND	5	9901	9902	9903	9904	9905	9906	2	25	1	2	51	1	9913	9914	9915	9916	0.0
FINT	6	9901	9902	9903	9904	9905	9906	2	16	1	1	1	1	9913	9914	9915	9916	0.0
DFINT	7	9901	9902	9903	9904	9905	9906	2	16	1	1	1	1	9913	9914	9915	9916	0.0
FINT	8	9901	9902	9903	9904	9905	9906	2	16	1	1	1	1	9913	9914	9915	9916	0.0
DFINT	9	9901	9902	9903	9904	9905	9906	2	16	1	1	1	1	9913	9914	9915	9916	0.0
CINTM	10	9901	9902	9903	9904	9905	9906	2	16	1	1	1	1	9913	9914	9915	9916	0.0
CINTC	11	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
CONC	12	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
DCONC	13	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
VORT	14	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
DVORT	15	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
PSI	16	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
DPSI	17	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
BUOY	18	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
DIFF	19	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0
VISC	20	9901	9902	9903	9904	9905	9906	2	16	1	2	31	1	9913	9914	9915	9916	0.0

INPUT DATA FOR THE CODE

8.2 THE MATERIAL PROPERTIES PARAMETERS

The material properties are as follows:

- the specific heat CP for the three regions;
- the conductivity COND for the three regions;
- the interface latent heat as a function of the temperature;
- the interface melt concentration as a function of the temperature;
- the interface crystal concentration as a function of the temperature;
- the relative density ρ/ρ_0, which is multiplied by GTERR to give the downward acceleration;
- the diffusivity for concentration, α; and
- the kinematic viscosity ν.

These functions are constants if the control is unity, otherwise they are calculated from the temperature and/or the concentration based on the integer control value and the parameters. Thus the decimal digits of the control parameter 124 indicate that the property is a four-term power series in T, plus the product of the concentration with a 2-term power series, plus the product of the square of the concentration with a 1-term power series in T (i.e. a constant).

The interface properties can also be expressed as functions of the concentration, with the flag being a negative control parameter. The properties are then as follows:

- latent heat as a function of the melt concentration;
- temperature as a function of the melt concentration;
- temperature as a function of the crystal concentration.

The constant in the relative density has no effect on the flow. It should normally be set to zero, to reduce rounding errors in the finite differences.

8.3 THE METHOD PARAMETERS

The first method parameters are the number of time-steps or iterations and the logical variables to control whether the

concentration and flow are to be updated.

The next row has time steps for temperature in the solid part of the sample (crystal) and in the ampoule. The objective in allowing distinct time steps is to get rapid iterative convergence to a steady state.

The next three rows of parameters are for the melt temperature, concentration, and velocity. The first three parameters control the time step, for iterating to a steady state. In this case, more rapid convergence can be obtained by taking larger steps for concentration, with smaller diffusivity, and by taking smaller steps where the mesh spacing is smaller. For concentration, TAUC is the maximum of the time step, and it varies with the powers PCR and PCZ of the mesh spacings dr/di and $d\zeta/dk$.

The next parameter in each row controls the extent to which upwind differencing may be used if the flow is strong. The logical variables control whether the implicit tridiagonal division ("fixing") is applied to each variable in each direction, and are not used at present, since we only have the Gaussian elimination method. The BET parameters control the amplitude of the advection and diffusion terms in the implicit algorithm. The value 0.5 corresponds to the Crank-Nicolson algorithm, with second order accuracy in time and good stability properties.

The first three method parameters on the last row are the time step and radial power for the interface function f, and the beta value for the implicit terms. The following three integers are the step intervals for updating the matrix decompositions, for flow, temperature, and concentration, respectively. It is computationally efficient not to do this every step. The frequency is automatically increased at the beginning of a computation. The last two method parameters are the factors of a quarter and a twelfth, used in the internal gravity wave algorithm.

8.4 THE OUTPUT PARAMETERS

The output parameters control different types of output. The direct access write and read option is used both for restarts of the calculation and for post-processing to obtain plots. The remaining rows of parameters control the printer and plotter output options and other diagnostics. Many are self-explanatory, but we will not give details of all the available options in this report.

CHAPTER 9

COMPUTER USAGE

9.1 COMPUTER SYSTEMS

The code development was mostly done on our microcomputer system. The code is one of a family of computer codes which we routinely run both on this system and on three VAX minicomputers at MSFC (MIPS1 and MIPS4 in HOSC, and the SSL VAX in Space Sciences Lab). These VAX computers are made available to us for NASA-sponsored work. We anticipate implementing some of these codes on the new Cray XMP computer in the future, for the convenience of our customers.

The microcomputer calculations take about 10 times as long as those done on an empty VAX. Naturally, we do not do intensive computations on the VAXes during the working day; if we did, the speed would be comparable with that of the micro.

Our VAX plots use Tektronix 4014 terminals and the TCS plot package, with our own interfaces. On the micro, we have implemented a TCS simulation on the dot matrix printer.

9.2 CODE USAGE AND INPUT DATA

Code usage on the two systems is very similar. The command

```
C BS
```

sets the environment. The command

```
CUP B 15 40 10 25
```

sets up the dimensions NR, NZ, NRA, and NZS in the common file of FORTRAN statements which are included automatically in every subroutine. The command

```
CALL
```

compiles all the subroutines. The command

```
CG PD
```

can be used to recompile the subroutines in the particular file PD, for example. The command

```
LGO B 15X40 NG
```

links the compiled subroutines into an executable program. The parameter NG stands for NO GO. To execute the program with null data, use the command

```
GO B 15X40 NULL
```

This produces the output file SETP.DAT printed in the previous chapter. We refer to this file as the data page. Copies of this file, with different names such as SALT.DAT, are edited and then used as input by the command

```
GO B 15X40 SALT
```

This command produces both printed output and computer files for later executions of the program to read. The command

```
LGO B 15X40 SALT CON
```

is used to link and execute the contouring program (CON) which produces plots from the output of the previous run.

CHAPTER 10

NUMERICAL RESULTS

In its new form, the code can be run in one of four modes, according to the input data:

1. Temperature alone;

2. Temperature and convection;

3. Temperature and concentration; and

4. Temperature, convection and concentration.

We have not presented any convection cases, as we are not yet satisfied with our algorithms; the flow routines presented in the appendix are an initialization to satisfy the boundary conditions, and dummy routines.

In this chapter, we present numerical results for two test problems, with temperature and concentration included. The two problems were the default case, with data listed in the previous chapter, and a minor modification, with smaller concentration and solute diffusivity.

The sample and ampoule diameters are 2 and 4 mm, respectively. The furnace ends are 2 mm apart, and the computational boundaries at the top and bottom are 4 mm apart. The top and bottom temperatures are 50 and -100 degrees. The ampoule speed is 0.001 cm/sec, or 3.6 cm/hour.

The material is nominally water, with zero melting point and latent heat 80 calories per cc of melt. The volume specific heats are 1.0, 0.5, and 0.1, respectively, for the melt, crystal, and ampoule, in units of cal/cc/deg. The respective conductivities are 14, 10, and 8, in units of 0.0001 cal/cm/sec/deg. The ratio of the crystal density to the melt is 0.9, so the material expands on solidifying.

The phase diagrams shown in Chapters 3 and 4 were for solutes that increased the melting point. Our test solute, however, decreases the melting point, and is rejected as the crystal grows, leading to a

solute excess ahead of the interface, cf. Figure 3.2. Melt with a 20% concentration is in equilibrium with crystal with a 2% concentration, at a temperature of -20 degrees.

For our model material, the buoyancy is linear in the temperature and concentration. However, we have not computed the convection flow, as stated above.

10.1 CASE 1: HIGH INPUT CONCENTRATION AND DIFFUSIVITY

For our first test case, we used a solute concentration at the top of 2%. With our state diagram, this implies a crystal concentration of 2%, and a melt concentration at the interface of 20%, at a temperature of -20 degrees.

The diffusivity was 0.0001 cm /sec. With the ampoule speed of 0.001 cm/sec, the length scale of the diffusion layer in the melt (neglecting the flow modification due to the expansion on solidification) is α/W_A, or 1mm. An estimate for the parameter $(dT/dz)/(dx/dz)$ at the interface is -250, which is greater than the slope of the melt line in the equation of state (-100). Thus dendrites should not form.

The following three figures show our steady state solution.

Figure 10.1 shows the computational mesh. We used

NR = 8,

NZ = 25,

NRA = 6,

NZS = 15.

Our choices of the DR and DZ input parameters gave a fine mesh near the interface and near the sample boundary, in order to resolve fine structure there withiut wasting mesh points elsewhere.

Figure 10.2 shows the temperature solution. The interface temperature varies between -18.1 and -19.5 degrees, at the ampoule and at the axis respectively. This corresponds to variations in the concentration. The melt concentration is one hundredth of the temperature magnitude; the crystal concentration is one thousandth of the temperature magnitude. The crystal surface is concave, due to latent heat release (a significant factor in this case), the lower conductivity of the crystal, and the fact that the temperature is approximately half way between the top and bottom tmperatures.

Figure 10.3 shows the steady-state solution for the concentration. The balance is between downward advection and upward diffusion. The flow is the ampoule flow, modified by the expansion of the melt as it freezes, which leads to a small superposed Poiseuille flow up the cylinder. The thickness of the diffusion layer agrees with the predicted value of 1mm. This thickness is too large for dendrites to form, even at these large concentrations. The maximum of the concentration is 19.5%, on the interface and at the axis. The minimum is of course the imposed value of 2% at the top.

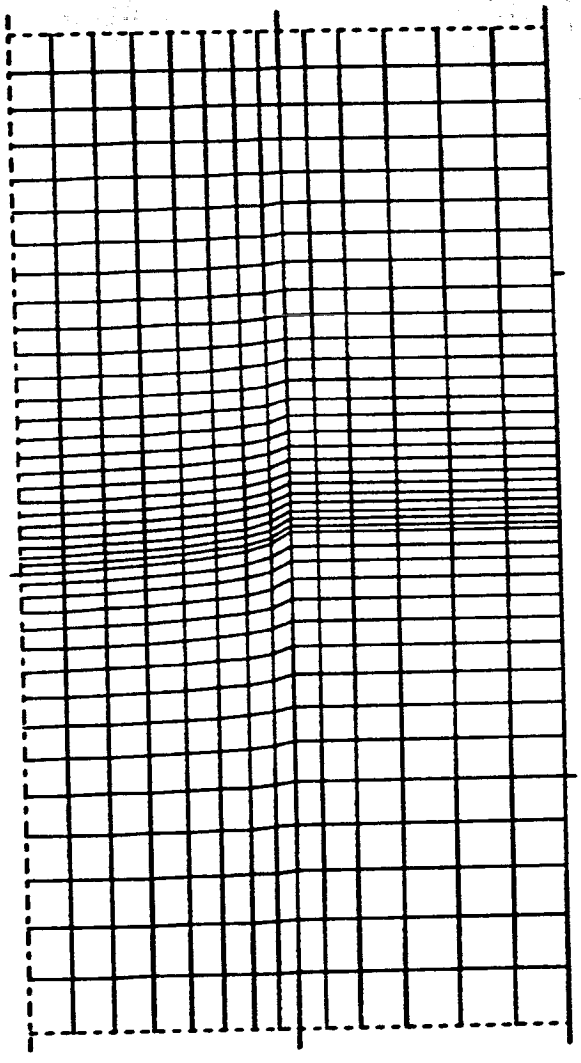


Figure 10.1 Computational Mesh for High Concentration and Diffusivity

TEMPERATURE IN THE WHOLE SYSTEM
SALT AND WATER TEST/.02/.0001
MAXIMUM = 50.000
MINIMUM = -100.00
INCREMENT = 8.0000
TIME = 80.491

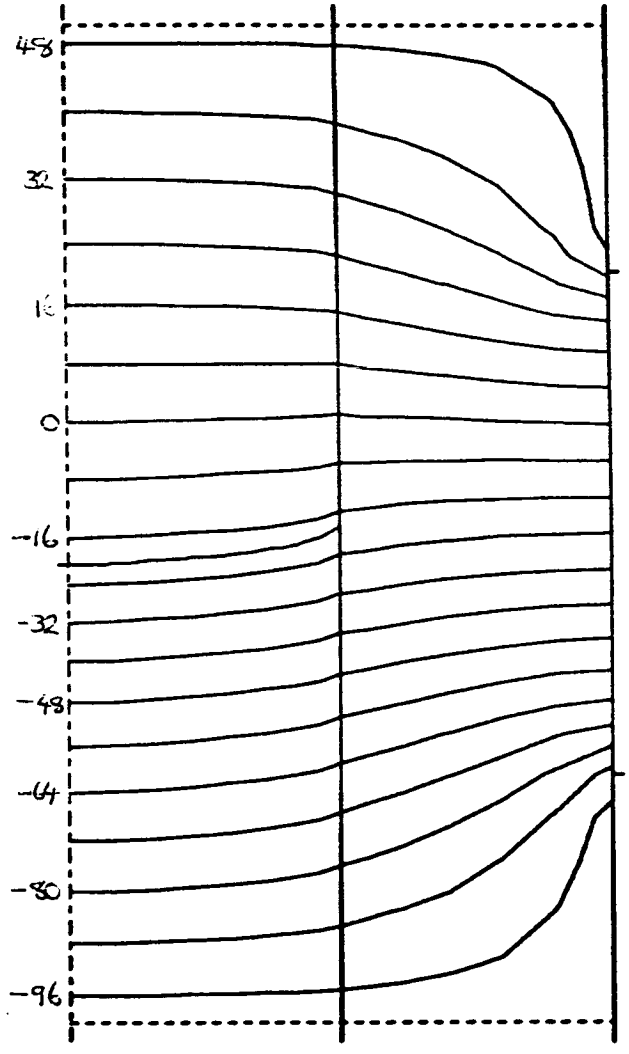


Figure 10.2 Temperature for High Input Concentration and Diffusivity

CONCENTRATION IN THE MELT
SALT AND WATER TEST/.02/.0001
MAXIMUM = 0.19504
MINIMUM = 2.00000E-02
INCREMENT = 2.00000E-02
TIME = 80.491

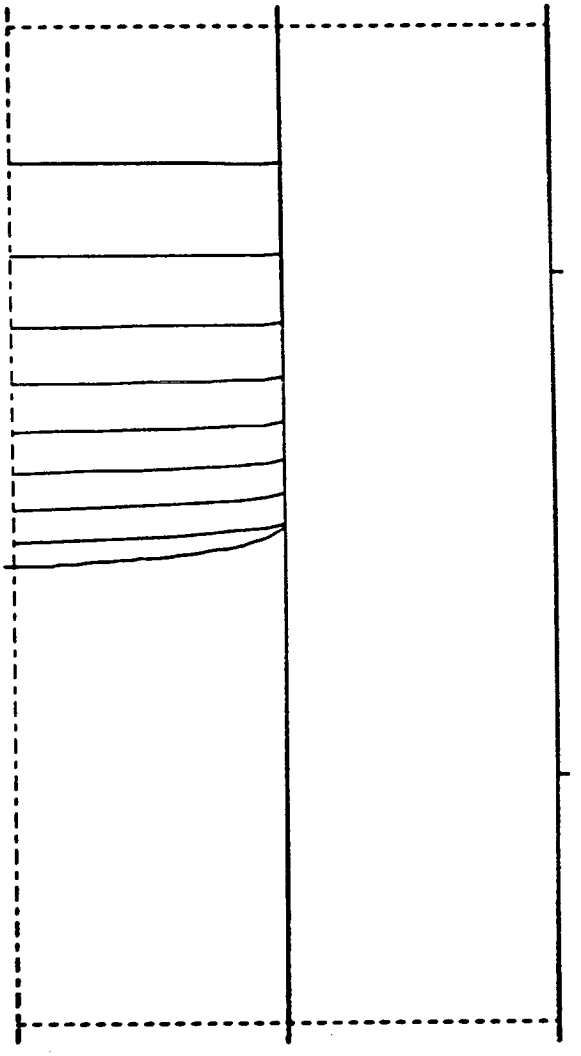


Figure 10.3 Concentration for High Concentration and Diffusivity

10.2 CASE 2: LOW INPUT CONCENTRATION AND DIFFUSIVITY

For our second test case, we used a solute concentration at the top of 0.1%. With our state diagram, this implies a crystal concentration of 0.1%, and a melt concentration at the interface of 1%, at a temperature of -1 degrees.

The diffusivity was 0.00001 cm /sec. With the ampoule speed of 0.001 cm/sec, the length scale of the diffusion layer in the melt (neglecting the flow modification due to the expansion on solidification) is α/W_A, or 0.1mm. An estimate for the parameter (dT/dz)/(dx/dz) at the interface is -500, which is greater than the slope of the melt line in the equation of state (-100). Thus dendrites should not form.

The following three figures show our steady state solution.

Figure 10.4 shows the computational mesh. We again used

NR = 8,

NZ = 25,

NRA = 6,

NZS = 15.

Our choices of the DR and DZ input parameters gave a fine mesh near the interface and near the sample boundary, in order to resolve fine structure there withiut wasting mesh points elsewhere. The mesh is different from the previous case, of course, because the interface is different.

Figure 10.5 shows the temperature solution. The interface temperature was -1 degrees, with very little variation. The crystal surface is less concave. The latent heat release and the lower conductivity of the crystal are the same as before, but the interface temperature is much closer to the upper imposed temperature, and this tends to make the crystal convex.

Figure 10.6 shows the steady-state solution for the concentration. The balance is between downward advection and upward diffusion. The flow is the ampoule flow, modified by the expansion of the melt as it freezes, which leads to a small superposed Poiseuille flow up the cylinder. The thickness of the diffusion layer agrees with the predicted value of 0.1mm. This small thickness is still too large for dendrites to form, due to the low concentration. The maximum of the concentration is 1%, on the whole interface. The minimum is of course the imposed value of 0.1% at the top.

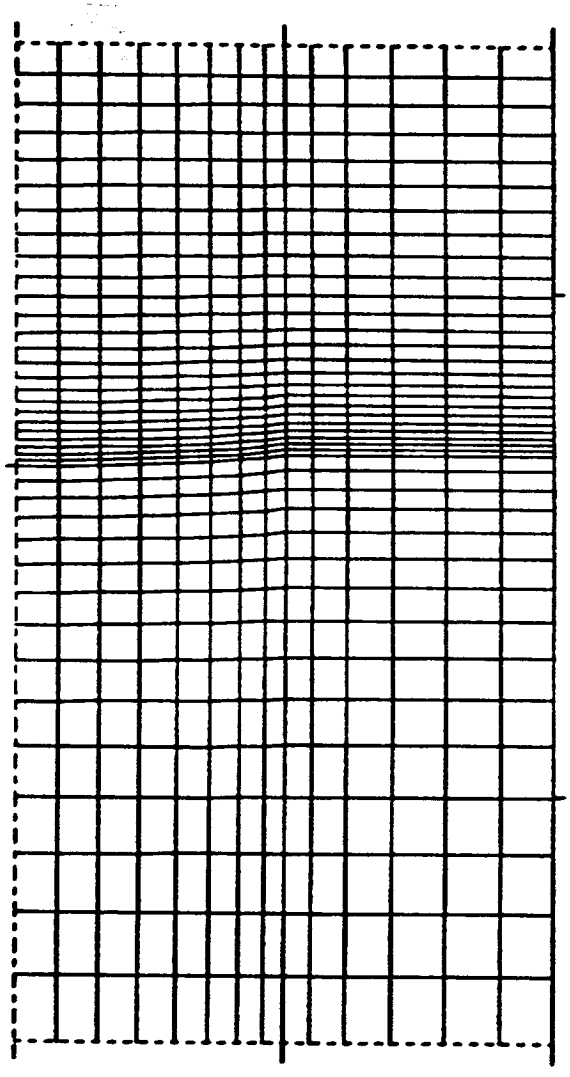


Figure 10.4 Computational Mesh for Low Concentration and Diffusivity

TEMPERATURE IN THE WHOLE SYSTEM
SALT AND WATER TEST/.001/.00001
MAXIMUM = 50.000
MINIMUM = -100.00
INCREMENT = 8.0000
TIME = 50.000

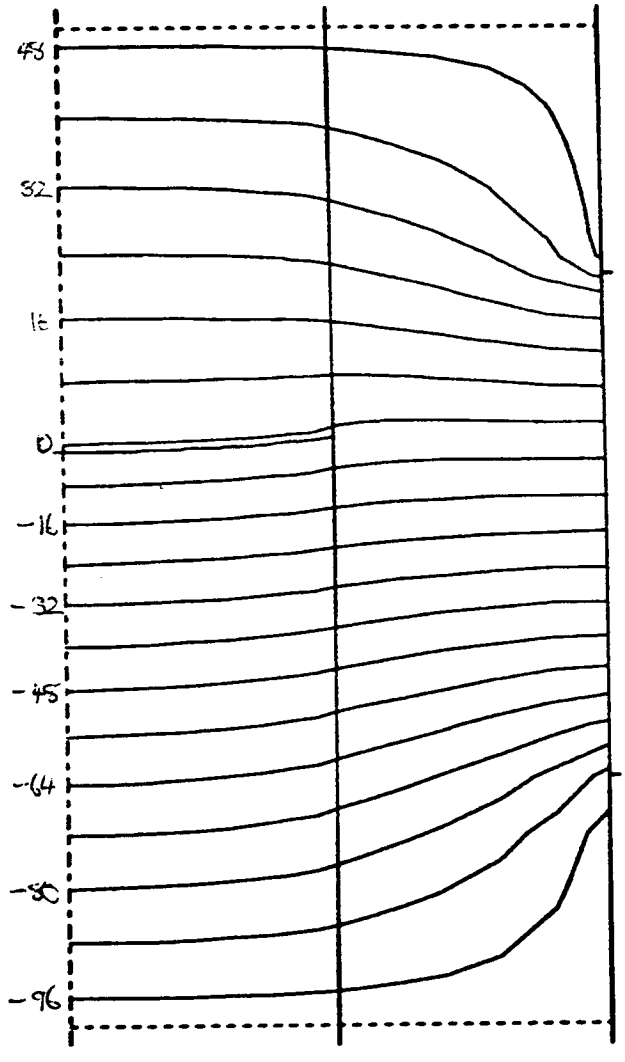


Figure 10.5 Temperature for Low Input Concentration and Diffusivity

CONCENTRATION IN THE MELT
SALT AND WATER TEST/.001/.00001
MAXIMUM = 1.00000E-02
MINIMUM = 1.00000E-03
INCREMENT = 1.00000E-03
TIME = 50.000

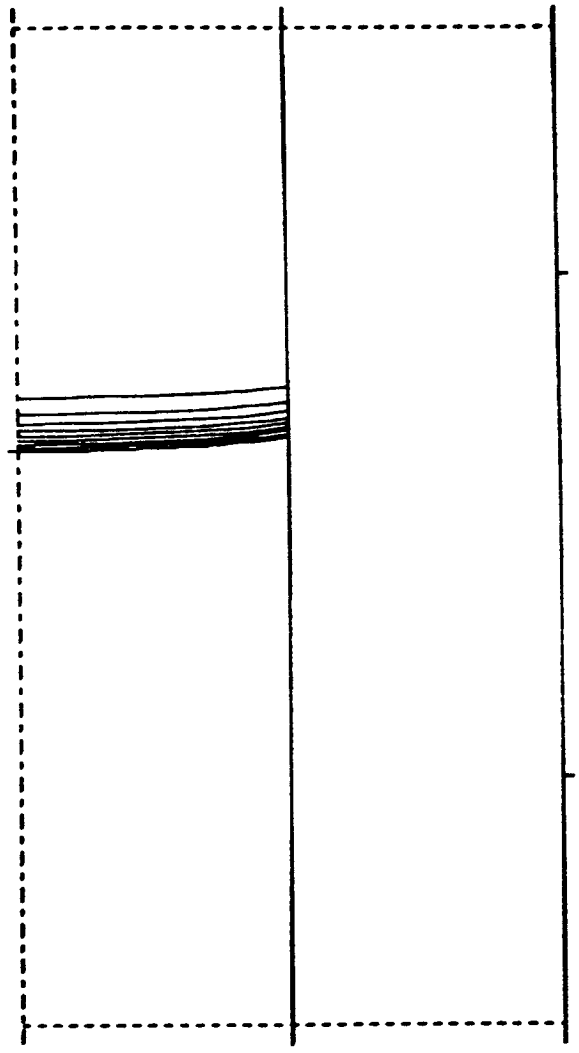


Figure 10.6 Concentration for Low Input Concentration and Diffusivity

CHAPTER 11

REFERENCES

C.J. Chang and R.A. Brown, 1983a, "Multiple Steady Solutions to a Buoyancy-Driven Convection Problem Including a Melt-Solid Phase Boundary", Int. J. Num. Methods Fluids.

C.J. Chang and R.A. Brown, 1983b, "Natural Convection in Steady Solidification. Finite-Element Analysis of a Two-Phase Rayleigh-Benard Problem", J. Comp. Phys.

C.J. Chang and R.A. Brown, 1983c, "Radial Segregation Induced by Natural Convection and Melt-Solid Interface Shape in Vertical Bridgman Growth", J. Crystal Growth 63, 22.

S.K. Doss, N.N. Carlson, and K. Miller, 1984, "Moving Finite Elements in 2D, Technical Progress Report: Year 3", Report SAI-84/1572, Science Applications, Inc., Pleasanton, CA.

W.W. Fowlis and G.O. Roberts, 1986, "Confinement of Thermocapillary Floating Zone Flow by Uniform Rotation", J. Crys. Growth 74, 301-320.

L.D. Landau and E.M. Lifshitz, 1959, "Fluid Dynamics", translated by J.B. Sykes and W.H. Reid, Addison Wesley.

T.L. Miller, "Numerical Modeling of Physical Vapor Transport in a Vertical Cylindrical Ampoule, With and Without Gravity", NASA Technical Paper 2620, July, 1986.

G.O. Roberts, 1984, "Mathematical Model for a Simplified Bridgman-Stockbarger Crystal Growing System", Report RAI-84-B-4.

G.O. Roberts, W.W. Fowlis, and T.L. Miller, 1984, "Finite-Difference Fluid Dynamics Computer Mathematical Models for the Design and Interpretation of Experiments for Space Flight", NASA Technical Paper 2323, 12pp..

G.O. Roberts and D. M. Rubenstein, 1981, "The Computer Code GORWAK for Stratified Wake Computations (U)". Classified reports.

REFERENCES

G.O. Roberts and J.L. Seftor, 1980, "A Sigma Coordinate Ocean Forecasting Computer Code, Parts 1 through V", Reports SAI-80-956-WA through SAI-80-958-WA, SAI-81-262-WA, and SAI-81-299-WA.

F. Rosenberger, 1979, "Fundamentals of Crystal Growth. I. Macroscopic Equilibrium and Transport Concepts", Springer Series in Solid State Science, Vol 5, Springer-Verlag.

APPENDIX A

CODE LISTING

The FORTRAN code is listed in the following pages. There are sixteen files, P., P1., P2., P3., PCON., PD., PG., PI., PP., PP1., PT., PT1., PT2., PT3., PT4., and PU., containing subroutines specific to this code. The file COM. contains the common blocks and other FORTRAN statements which are automatically included in every subroutine. The program is self-documenting.

The file P. contains the main program. P1. through P3. contain init=ial=iz=ation routines, and P3. includes the routine calls for direct access input and output. PCON. contains a main program and subroutines for the various plot options. PD. routines provide numerical diagnostic lines for each time step. PG. contains the routines which control most output options. The PI. subroutine reads and writes the data page file. PP. and PP1. routines handle the material properties. Subroutines in the PT series of files handle updating the temperature, concentration, and interface. The subroutines in the PU series do the motion.

The seven files, QB., QC., QCZ., QE., QM., QVAX., and QZ., contain general-purpose utility subroutines, used in a whole family of codes, and are not included. Nor is the PC substitute for QVAX.

We use the TCS plot graphics software package on VAX computers and Tektronix 4014 terminals with hard-copy units. We use our own graphics software (also not included in the listing) to emulate the TCS package on a dot-matrix printer.

```
      PARAMETER (NR = 15 , NZ = 30 , NRA = 9 , NZS = 20)
C
      PARAMETER (NRP1=NR+1,NRP2=NR+2)
      PARAMETER (NRAP1=NRA+1,NRAP2=NRA+2)
      PARAMETER (NRC=NR+NRA,NRCP1=NRP1+NRA,NRCP2=NRP2+NRA)
C
      PARAMETER (NZP1=NZ+1,NZP2=NZ+2)
      PARAMETER (NZSP1=NZS+1,NZSP2=NZS+2)
      PARAMETER (NZC=NZ+NZS,NZCP1=NZP1+NZS,NZCP2=NZP2+NZS)
C
      PARAMETER (NMATPR=12,NPRCON=8,NAPC=20)
C
C CUP B 15 40 10 20
C ADDS A PARAMETER STATEMENT DIMENSIONING NR, NZ, NRA, NZS
C AND REPLACES THESE LINES PR1, etc., BY THE FILES:
C PROB1, PROB2, METH1, METH2, AND OUTP.BC
C WHICH CONTAIN CONTINUATION LINES FOR I/O STATEMENTS IN INIT0
C AND FOR COMMON BLOCKS IN COMO TO MAKE THE INCLUDED FILE COM.
C
      COMMON /PROB/
     1 RINS  ,DRINS  ,ZT     ,DZT    ,TTOP   ,ZTTOP
     1 ,RSAM ,DRSAM  ,ZB     ,DZB    ,TBOT   ,ZTBOT
     1 ,RAMP ,DRAMPI ,ZINTI  ,DZINTM ,GTERR  ,ZTBSC
     1       ,DRAMPO ,ZINTL  ,DZINTS ,WAMP   ,CTOP
     1                                ,DENCBM
      COMMON /METH/
     1 NSTEP ,LC     ,LU
     1 ,TAUS ,TAUA
     1 ,TAUT ,PRT    ,PZT    ,TUPWND ,FIXTR  ,FIXTZ  ,BETTA  ,BETTD
     1 ,TAUC ,PRC    ,PZC    ,CUPWND ,FIXCR  ,FIXCZ  ,BETCA  ,BETCD
     1 ,TAUU ,PRU    ,PZU    ,UUPWND ,FIXUR  ,FIXUZ  ,BETUA  ,BETUD
     1 ,TAUF ,PRF    ,BEF    ,NUF    ,NTF    ,NCF    ,AQURTN ,ATWLV
      COMMON /OUTP/
     1 ISEGR ,ISEGW  ,IBEGDA ,IINCDA ,NDIAG  ,NCLP   ,NCLI
     1 ,RSTRPL       ,ZBPL   ,ZTPL   ,NCOPY
C
      LOGICAL FIXCZ,FIXCR,FIXTZ,FIXTR,FIXUZ,FIXUR,PRTEST,LC,LU,
     1 DIAG,TLOG,PPCHAN,LCF,LTF,LLF,LUF
      COMMON /LOGIC/
     1 DIAG,TLOG,PPCHAN,LCF,LTF,LLF,LUF
      CHARACTER *40 TEXPRG(4)
      CHARACTER *8 STITLE(NAPC)
      CHARACTER *42 LTITLE(NAPC)
      CHARACTER *44 DESCR
      CHARACTER *15 CMATPR(NMATPR)
      CHARACTER *20 CUNITS(NMATPR)
      CHARACTER *28 JTIME
      CHARACTER *40 CGROW
      COMMON /CHAR/ TEXPRG,LTITLE,STITLE,DESCR,CMATPR,CUNITS,JTIME,CGR
C
      INTEGER *4 MATPRT(NMATPR)
      COMMON /MATPRP/ PRPMAT(NPRCON,NMATPR),MATPRT
C
      COMMON /CPRNT/ ACONT(NAPC),IPC(16,NAPC)
C
      COMMON /SCAL/
     1  PI    ,ICASE  ,ISTEP  ,NTYPE
     1 ,IDAOUT       ,IDAIO  ,TTIME
```

```
590           1 ,IREC1R       ,IREC1W ,NREC1  ,IREC1
600           1 ,IREC2R       ,IREC2W ,NREC2  ,IREC2
610           1 ,ISEGRN       ,ISEGWN ,ISEGRU ,ISEGWU
620           1 ,FINT ,FRITE  ,FBOT   ,FTOP   ,FSUM
630           1 ,XINT ,XTOP   ,XSUM
640           1 ,DIDRAB       ,DKDZTC ,DEDZTC
650           1 ,DKDZMM       ,DIDRMS
660           1 ,DKDZMC       ,DIDRMA
670           1 ,TUPM ,CUPM   ,UUPM
680           1 ,TMELT        ,CMELT
690           1 ,ZL   ,ZINT
700           1 ,FAMP ,DFAMP  ,NFABC  ,FABC(3)
710           1 ,AWALL
720     C
730           COMMON /ARAYZ/
740           1  ZETW(NZCP1)  ,DKDZTW(NZCP1)  ,DZTDKW(NZCP1)
750           1 ,ZETH(NZCP2)  ,DKDZTH(NZCP2)  ,DZTDKH(NZCP2)
760           1 ,ETAW(NZCP1)  ,DEDZTW(NZCP1)  ,DETDKW(NZCP1)  ,ETA2W(NZCP1)
770           1 ,ETAH(NZCP2)  ,DEDZTH(NZCP2)  ,DETDKH(NZCP2)  ,ETA2H(NZCP2)
780           1 ,TRBC(NZCP2)  ,TRBM(NZCP2)
790           1 ,TVBTZH(NZCP2)        ,CVBTZH(NZCP2)  ,UVBTZW(NZCP2)
800           1 ,DKDZAH(NZCP2)        ,DKDZAW(NZCP1)  ,DZDKAH(NZCP2)  ,DZDKAW(
810           1 ,ZAH(NZCP2)   ,ZAW(NZCP1)
820     C
830           COMMON /ARAYR1/
840           1  RW(NRCP1)    ,DIDRW(NRCP1)   ,RH(NRCP2)      ,DIDRH(NRCP2)
850           1 ,HBRDRH(NRCP2)        ,DRB2RW(NRCP1)  ,RDRWQH(NRCP2)
860           1 ,RDRB2H(NRCP2)        ,RDRB2W(NRCP2)
870           1 ,RB2DRH(NRCP2)        ,RB2DRW(NRCP2)
880           1 ,R2W(NRP1)    ,R4W(NRP1)
890           1 ,DFDRW(NRP2)  ,DFDRH(NRP2)
900           1 ,RDRF8H(NRP2) ,RDRF8W(NRP2)
910           1 ,DRF8RH(NRP2) ,DRF8RW(NRP2)
920           1 ,ZDRFH(NRP2)  ,ZDRFW(NRP2)
930           1 ,WK1(NRCP1)
940           COMMON /ARAYR2/
950           1  FINTH(NRP2)  ,DFINTH(NRCP2)  ,RFINTH(NRCP2)  ,RFINTW(NRCP1)
960           1  ,DFINTW(NRCP2)
970           1 ,TINTH(NRP2)  ,CINTMH(NRP2)   ,CINTCH(NRP2)   ,HEATLAT(NRP2)
980           1 ,DTINTH(NRP2)
990           1 ,CPINTM(NRP2) ,CDINTM(NRP2)   ,DINTH(NRP2)
1000          1 ,CPINTC(NRP2) ,CDINTC(NRP2)
1010          1 ,DTFINT(NRP2) ,DCFINT(NRP2)   ,TFZC(NRP2)
1020          1 ,DFTINT(NRP2) ,DFCINT(NRP2)   ,DTCINT(NRP2)
1030          1 ,DCMDT(NRP2)  ,DTDCM(NRP2)    ,DCCDT(NRP2)    ,DTDCC(NRP2)
1040          1 ,DHLDT(NRP2)  ,DHLDC(NRP2)
1050          1 ,TZICD(NRP2)  ,TZICE(NRP2)
1060          1 ,TZICF(NRP2)  ,TZICG(NRP2)
1070          1 ,CTMULT(NRP2)
1080          1 ,TVBTRH(NRCP2)        ,CVBTRH(NRP2)   ,UVBTRW(NRP2)   ,BTF(NRP
1090          COMMON /ARAYR3/
1100          1  TDRWH(NRCP1) ,TGRWH(NRCP1)
1110          1 ,TFRWH(NRCP1) ,TERWH(NRCP1)
1120          1 ,TDZHW(NRCP1) ,TGZHW(NRCP1)
1130          1 ,TFZHW(NRCP1) ,TEZHW(NRCP1)
1140          1 ,CDRWH(NRP1)  ,CGRWH(NRP1)
1150          1 ,CFRWH(NRP1)  ,CERWH(NRP1)
1160          1 ,CDZHW(NRP1)  ,CGZHW(NRP1)
```

```
1170            1 ,CFZHW(NRP1)   ,CEZHW(NRP1)
1180            1 ,D2T(NRCP2)    ,D2C(NRP2)
1190    C
1200            COMMON /ARAY2/T(NRCP2,NZCP2)     ,DT(NRCP2,NZCP2)
1210            COMMON /AR2A/ CP(NRCP2,NZCP2)    ,COND(NRCP2,NZCP2)
1220            COMMON /AR2B/ C(NRP2,NZP2)       ,DC(NRP2,NZP2)
1230            COMMON /AR2C/ PSI(NRP2,NZP2)     ,DPSI(NRP2,NZP2)          ,PSIO(NR
1240            COMMON /AR2D/ DVORT(NRP1,NZP1)   ,VORT(NRP1,NZP1)
1250            COMMON /AR2E/ BUOY(NRP2,NZP2)    ,DIFF(NRP2,NZP2)
1260            1 ,VISC(NRP2,NZP2)
1270    C
1280            PARAMETER (NCB=2*NR+3,NTB=2*NRC+3,NAI=5*NR+2*NRA,NAIS=3*NR+2*NRA
1290            COMMON /ARG/
1300            1  CRD(NRP2,NZP2)        ,CRG(NRP2,NZP2)
1310            1 ,CRE(NRP2,NZP2)        ,CRF(NRP2,NZP2)
1320            1 ,CZD(NRP2,NZP2)        ,CZG(NRP2,NZP2)
1330            1 ,CZE(NRP2,NZP2)        ,CZF(NRP2,NZP2)
1340            1 ,CA(NCB,NR,NZ)         ,CF(NR,NR,NZ)
1350            1 ,DCNG(NR,NZ)
1360            1 ,TRD(NRCP2,NZCP2)      ,TRG(NRP2,NZP2)
1370            1 ,TRE(NRP2,NZCP2)       ,TRF(NRP2,NZCP2)
1380            1 ,TZD(NRCP2,NZCP2)      ,TZG(NRCP2,NZCP2)
1390            1 ,TZE(NRP2,NZCP2)       ,TZF(NRP2,NZCP2)
1400            1 ,TA(NTB,NRC,NZC)       ,TF(NR,NRC,NZC)
1410            1 ,DTNG(NRC,NZC)
1420            1 ,AI(NAI,NAI)   ,BI(NAI)
1430    C
1440            INCLUDE 'LUTIL:MACH.INC'
```

```
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C:::                                                                       :::
C:::          BRIDGMAN-STOCKBARGER CRYSTAL GROWTH CODE                     :::
C:::                                                                       :::
C:::     AUTHOR:              GLYN ROBERTS                                 :::
C:::                  ROBERTS ASSOCIATES, INCORPORATED                     :::
C:::                       1726 PINE VALLEY DRIVE                          :::
C:::                          VIENNA, VA 22180                             :::
C:::                            703/938-1757                               :::
C:::                                                                       :::
C:::     MONITOR:             WILLIAM W. FOWLIS                            :::
C:::                       NASA/ES73, MSFC, AL 35812                       :::
C:::                            205/544-7813                               :::
C:::                                                                       :::
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C
C
      PROGRAM MAIN
C
      INCLUDE 'COM./NOLIST'
C
      EXTERNAL IWRITE
C
C INITIALIZE
C
      LCONT = .FALSE.
      CALL SETUP
C
C CASE DO LOOP TERMINATES IN INIT0 WHEN DATA RUNS OUT
C
      DO 20 ICASE = 1, 100
C
C INITIALIZE TIMING AND CGROW
C
      CALL LIB$INIT_TIMER(IHAND)
      CGROW = ' '
C
C INPUT SEGMENT
C
      CALL INIT0
C
C INIT2 INITIALIZES THE ARRAYS AND MESH,
C       CALLS PRNT FOR THE INITIAL ARRAYS, AND
C       INITIALIZES THE CALCULATION
C
      CALL INIT2
C
C TIME STEP LOOP
C
      DO 10 ISTEP = 1, NSTEP
C
C UPDATE TIME AND T, C, AND F.
C
      TTIME = TTIME + TAUT
      CALL TFSTEP
```

```
590   C
600   C UPDATE MATERIAL PROPERTIES
610   C
620           ISTEPM = MAX(1,NSTEP/5)
630           IF (ISTEP/ISTEPM*ISTEPM .EQ. ISTEP) CALL SMATPR
640   C
650   C UPDATE FLOW
660   C
670   C       CALL USTEP
680   C
690   C STEPS COMPLETED FOR COPY, IN CASE OF GROWTH TERMINATION
700   C
710           NSTEPC = ISTEP
720   C
730   C GROWTH TERMINATION PROCESSING
740   C
750           IF (CGROW .NE. ' ') THEN
760                   WRITE (*,7) ICASE, ISTEP, CGROW
770                   WRITE (6,7) ICASE, ISTEP, CGROW
780   7               FORMAT (/ ' GROWTH TERMINATION ON CASE ',I4,
790           1                 '     AT STEP',I7,5X,'CGROW = ',A/)
800                   GO TO 15
810           ENDIF
820   C
830   10      CONTINUE
840   C
850   15      CONTINUE
860   C
870   C TIMER STATISTICS TO STREAM 6 AND SYS$OUTPUT
880   C
890           CALL LIB$SHOW_TIMER(IHAND)
900           CALL LIB$SHOW_TIMER(IHAND,,IWRITE)
910   C
920   C COPY DIAGNOSTICS TO RESULT AND SUMMARY FILES
930   C
940           DO 18 IS = 30, 34
950   18      CALL COPY (IS,6,16,NSTEPC+1,NDIAG,NSTEP+1,(NLPP-NLMA)/5-4)
960   C
970   C END CASE LOOP
980   C
990   20       CONTINUE
1000  C
1010          CALL STOPP ('TOO MANY CASES%')
1020          END
```

```
      SUBROUTINE SETUP
C
C OPEN UNITS AND INITIALIZE DATA PAGE DATA
C
      INCLUDE 'COM./NOLIST'
C     DATA BLANK /' '/
C
C PROBLEM, METHOD, AND OUTPUT PARAMETERS ARE SET TO DEFAULTS,
C AND CHANGED USING INPUT IN INIT0.
C IF THE INPUT IS NULL, THESE DEFAULTS ARE USED TO WRITE SETP.DAT.
C
C SET MACHINE PARAMETERS IN COMMON BLOCK /MACHIN/
C USE A DIFFERENT VERSION ON EACH COMPUTER
C
      CALL SMACH
C
C OPEN FILES FOR DATA, WORK, AND SUMMARY.
C LCONT IS .TRUE. FOR CONTOUR.
C
      OPEN (UNIT=5,FILE='FOR005',STATUS='OLD')
      CALL SQOPN(5,0)
      CALL SQOPN(16,1)
C
      DO 1 J = 30, 34
1     CALL SQOPN(J,1)
C
C GET TEXT FOR DATE AND TIME
C
      CALL DATTIM (JTIME)
C
C COPY COMPLETE INPUT TO SUMMARY FILE.
C
      WRITE (16,10) NR,NZ,NRA,NZS, JTIME
10    FORMAT ('1BRIDGMAN-STOCKBARGER CODE  COMPLETE INPUT DATA FILE'
     1       ,T75,'MESH',I3,3(',',I3),T100,A/)
      IF (.NOT. LCONT) CALL COPYP (5,16,137)
C
C                                          PROBLEM PARAMETERS
C
      DESCR = 'SETUP DATA'
C
C R DOMAIN
C
      RINS = 0.
      RSAM = .1
      RAMP = .2
C
C R MESH BOUNDARY RESOLUTION REQUIREMENTS
C
      DRINS = .1
      DRSAM = .03
      DRAMPI = .02
      DRAMPO = .06
C
C Z DOMAIN
C
      ZT = .2
      ZB = -.2
```

```
C
C Z MESH
C
        ZINTI = 0
        ZINTL = 0.5
C
        DZT = .5
        DZB = .5
        DZINTM = .03
        DZINTS = .03
C
C EXTERNAL PARAMETERS
C
        TTOP = 50
        TBOT = -100
        GTERR = 980
        WAMP = 0.001
        DENCBM = .9
C
        ZTTOP = .1
        ZTBOT = -.1
        ZTBSC = 0.5
        CTOP  = .02
C
C                                          MATERIAL PROPERTIES
C
        DO 20 IMAT = 1, NMATPR
        DO 20 K = 1, NPRCON
        MATPRT(IMAT) = 1
        PRPMAT(K,IMAT) = 0
20      CONTINUE
C
C PROPERTY NAMES, UNITS, AND INITIAL DEFAULTS
C
        CMATPR(1) = 'CPMELT'
        CMATPR(2) = 'CPCRYS'
        CMATPR(3) = 'CPAMP'
        CMATPR(4) = 'CDMELT'
        CMATPR(5) = 'CDCRYS'
        CMATPR(6) = 'CDAMP'
        CMATPR(7) = 'LATHEAT'
        CMATPR(8) = 'CINTM'
        CMATPR(9) = 'CINTC'
        CMATPR(10) = 'BUOYANCY'
        CMATPR(11) = 'DIFFUSIVITY'
        CMATPR(12) = 'VISCOSITY'
C
        CUNITS(1) = 'cal/cc/deg'
        CUNITS(2) = 'cal/cc/deg'
        CUNITS(3) = 'cal/cc/deg'
        CUNITS(4) = 'cal/cm/sec/deg'
        CUNITS(5) = 'cal/cm/sec/deg'
        CUNITS(6) = 'cal/cm/sec/deg'
        CUNITS(7) = 'cal/cc'
        CUNITS(8) = 'gm/gm'
        CUNITS(9) = 'gm/gm'
        CUNITS(10) = 'gm/gm'
        CUNITS(11) = 'cm2/sec'
```

```
1170            CUNITS(12) = 'cm2/sec'
1180    C
1190            MATPRT(8) = 2
1200            MATPRT(9) = 2
1210            MATPRT(10) = 12
1220    C
1230            PRPMAT(1,1) = 1.0
1240            PRPMAT(1,2) = 0.5
1250            PRPMAT(1,3) = 0.1
1260            PRPMAT(1,4) = 0.0014
1270            PRPMAT(1,5) = 0.0010
1280            PRPMAT(1,6) = 0.0008
1290            PRPMAT(1,7) =80.0
1300            PRPMAT(1,8) = 0.0
1310            PRPMAT(1,9) = 0.0
1320            PRPMAT(2,8) =-0.01
1330            PRPMAT(2,9) =-0.001
1340            PRPMAT(1,10)= 0.0
1350            PRPMAT(2,10)=-1.E-4
1360            PRPMAT(3,10)= 0.01
1370            PRPMAT(1,11)= 1.E-4
1380            PRPMAT(1,12)= 1.E-2
1390    C
1400    C                                                  METHOD PARAMETERS
1410    C
1420    C NUMBER OF TIME STEPS OR ITERATIONS
1430    C
1440            NSTEP = 50
1450    C
1460    C LOGICAL VARIABLES TO UPDATE CONCENTRATION OR FLOW
1470    C
1480            LC = .FALSE.
1490            LU = .FALSE.
1500    C
1510    C TIME STEPS AND POWERS OF DR AND DZETA
1520    C FOR CONCENTRATION, TEMPERATURE, AND MOTION.
1530    C
1540            TAUT = 0.5
1550            TAUS = 0.5
1560            TAUA = 0.5
1570            PRT = .0
1580            PZT = .0
1590    C
1600            TAUC = 0.5
1610            PRC = .0
1620            PZC = .0
1630    C
1640            TAUU = 0.5
1650            PRU = .0
1660            PZU = .0
1670    C
1680    C UPWIND DIFFERENCING COEFFICIENTS
1690    C
1700            CUPWND = 0.05
1710            TUPWND = 0.0
1720            UUPWND = 0.0
1730    C
1740    C IMPLICIT LOGICAL CONTROLS.
```

```
1750    C
1760            FIXCR = .TRUE.
1770            FIXCZ = .TRUE.
1780            FIXTR = .TRUE.
1790            FIXTZ = .TRUE.
1800            FIXUR = .TRUE.
1810            FIXUZ = .TRUE.
1820    C
1830    C IMPLICIT FIXING AMPLITUDES.
1840    C
1850            BETCA  = .7
1860            BETCD  = .7
1870            BETTA  = .7
1880            BETTD  = .7
1890            BETUA  = .7
1900            BETUD  = .7
1910    C
1920    C F STEPS
1930    C
1940            TAUF = 0.5
1950            PRF  = 0.
1960            BEF  = .7
1970    C
1980    C INTERNAL WAVE METHOD
1990    C
2000            AQURTN = .25
2010            ATWLV  = 1./12.
2020            NPI = 10
2030    C
2040    C FREQUENCIES FOR UPDATING L-U DECOMPOSITIONS IN GAUSS ELIMINATION
2050    C METHOD FOR DT,DC AND DF.
2060    C
2070            NUF = 10
2080            NTF = 10
2090            NCF = 30
2100    C
2110    C                                              OUTPUT CONTROL PARAMETERS
2120    C
2130    C DIRECT ACCESS OUTPUT AND INPUT.
2140    C ISEGR -  READ SEGMENT FOR INITIAL CONDITIONS. ZERO FOR ANALYTIC.
2150    C ISEGW -  FIRST WRITE SEGMENT.
2160    C
2170    C -1 MEANS USE NEXT SEGMENT OF SAME TYPE
2180    C
2190    C ISEGR NEGATIVE MEANS USE NEXT AVAILABLE SEGMENT
2200    C ISEGW NEGATIVE MEANS USE NEXT AVAILABLE SEGMENT
2210    C  (AFTER THE LAST ONE READ OR WRITTEN IN PREVIOUS CASE)
2220    C INITIAL DEFAULTS FOR NULL DATA ARE 0, 1.
2230    C INITIAL NEXT SEGMENTS ARE 0, 1.
2240    C
2250             ISEGR  = 0
2260             ISEGW  = 1
2270    C
2280             ISEGRN = 0
2290             ISEGWN = 1
2300    C
2310    C IBEGDA - FIRST DIRECT ACCESS OUTPUT STEP  (-1 MEANS NSTEP)
2320    C IINCDA - INCREMENT FOR DIRECT ACCESS OUTPUT STEP
```

```
2330    C
2340            IBEGDA = -1
2350            IINCDA = 9900
2360    C
2370    C DIAGNOSTIC PRINTING INTERVAL. ZERO MEANS NSTEP/55
2380    C
2390            NDIAG = 0
2400    C
2410    C PRINT AND CONTOUR CONTROL USES  IPC(J,IAPC)
2420    C
2430    C NUMBER OF PRINTER LINES FOR CONTOUR PLOTS, PHYSICAL AND INTEGER.
2440    C ZERO MEANS USE CODE DEFAULTS
2450    C
2460            NCLP = 0
2470            NCLI = 0
2480    C
2490    C PLOT CONTROLS
2500    C R STRETCH FACTOR FOR PLOTS
2510    C Z BOTTOM AND TOP FOR PLOTS
2520    C   (ZERO IMPLIES CALCULATION OF A DEFAULT)
2530    C
2540            RSTRPL = 1
2550            ZBPL = 0
2560            ZTPL = 0
2570    C
2580    C NUMBER OF PRINTED COPIES OF MAIN RESULT FILE.
2590    C NOT USED FOR THE LAST CASE.
2600    C
2610            NCOPIES = 1
2620    C
2630    C IPC(16,IAPC) AND ACONT(IAPC) CONTROL ARRAY OUTPUT.
2640    C
2650    C J = 1  ISTEP START     FOR PRINTER CONTOUR AGAINST PHYSICAL VARIABLES
2660    C J = 2  ISTEP INCREMENT FOR PRINTER CONTOUR AGAINST PHYSICAL VARIABLES
2670    C
2680    C J = 3  ISTEP START     FOR PRINTER CONTOUR AGAINST INTEGER VARIABLES
2690    C J = 4  ISTEP INCREMENT FOR PRINTER CONTOUR AGAINST INTEGER VARIABLES
2700    C
2710    C J = 5  ISTEP START     FOR PRINTING NUMBERS
2720    C J = 6  ISTEP INCREMENT FOR PRINTING NUMBERS.
2730    C                NEGATIVE MEANS WRITE THEM IN A FILE FOR LATER PLOTTING.
2740    C                J=7 SIGN NEGATIVE TO WRITE NEW PLOT FILE WITH HEADER.
2750    C                J=7 SIGN CORRECTED ONCE HEADER IS WRITTEN.
2760    C                J=7 SIGN POSITIVE - OPEN FILE AS 'OLD'.
2770    C
2780    C J = 7, 8, 9    PRINT DO LOOP CONTROLS FOR I
2790    C J = 10, 11, 12 PRINT DO LOOP CONTROLS FOR K
2800    C
2810    C J = 13  IDAIO START     FOR PLOTTER CONTOUR AGAINST PHYSICAL VARIABLES
2820    C J = 14  IDAIO INCREMENT FOR PLOTTER CONTOUR AGAINST PHYSICAL VARIABLES
2830    C
2840    C J = 15  IDAIO START     FOR PLOTTER CONTOUR AGAINST INTEGER VARIABLES
2850    C J = 16  IDAIO INCREMENT FOR PLOTTER CONTOUR AGAINST INTEGER VARIABLES
2860    C
2870    C ACONT .NE. 0 MEANS USE ACONT FOR PLOTTER CONTOUR INCREMENT
2880    C ACONT .EQ. 0 MEANS USE NCONT FOR PRINTER AND 2*NCONT FOR PLOTS.
2890    C NCONT IS NOT CHANGED BY INIT0.
2900    C
```

```
2910   C THIS LOOP TO NAPC COVERS ALL THE PLOT TYPES.
2920   C
2930           DO 100 IAPC = 1, NAPC
2940   C
2950           DO 110 J = 1, 16
2960   110     IPC(J,IAPC) = J + 9900
2970   C
2980   C
2990           IPC( 7,IAPC) = 2
3000           IPC( 8,IAPC) = NRP1
3010           IPC( 9,IAPC) = 1
3020           IPC(10,IAPC) = 2
3030           IPC(11,IAPC) = NZP1
3040           IPC(12,IAPC) = 1
3050   C
3060           IF (IAPC .LE. 5) THEN
3070                   IPC( 8,IAPC) = NRCP1
3080                   IPC(11,IAPC) = NZCP1
3090           ELSE IF (IAPC .LE. 11) THEN
3100                   IPC(10,IAPC) = 1
3110                   IPC(11,IAPC) = 1
3120           ELSE IF (IAPC .GE. 14  .AND.  IAPC .LE. 17) THEN
3130                   IPC( 8,IAPC) = NRP1
3140                   IPC(11,IAPC) = NZP1
3150           ENDIF
3160   C
3170           ACONT(IAPC) = 0.
3180   C
3190   100     CONTINUE
3200   C
3210   C                                          LONG AND SHORT CHARACTER TITLES
3220   C
3230           LTITLE(1) = 'MESH'
3240           LTITLE(2) = 'TEMPERATURE IN THE WHOLE SYSTEM'
3250           LTITLE(3) = 'CHANGE IN TEMPERATURE'
3260           LTITLE(4) = 'SPECIFIC HEAT (cal/cc/deg)'
3270           LTITLE(5) = 'CONDUCTIVITY (cal/cm sec deg)'
3280           LTITLE(6) = 'INTERFACE HEIGHT'
3290           LTITLE(7) = 'INTERFACE CHANGE'
3300           LTITLE(8) = 'INTERFACE TEMPERATURE'
3310           LTITLE(9) = 'INTERFACE TEMPERATURE CHANGE'
3320           LTITLE(10) = 'MELT CONCENTRATION AT THE INTERFACE'
3330           LTITLE(11) = 'CRYSTAL CONCENTRATION AT THE INTERFACE'
3340           LTITLE(12) = 'CONCENTRATION IN THE MELT'
3350           LTITLE(13) = 'CHANGE IN THE CONCENTRATION'
3360           LTITLE(14) = 'VORTICITY'
3370           LTITLE(15) = 'CHANGE IN THE VORTICITY'
3380           LTITLE(16) = 'STREAM FUNCTION'
3390           LTITLE(17) = 'CHANGE IN THE STREAM FUNCTION'
3400           LTITLE(18) = 'RELATIVE DENSITY EXCESS'
3410           LTITLE(19) = 'DIFFUSIVITY'
3420           LTITLE(20) = 'KINEMATIC VISCOSITY'
3430    C
3440           STITLE(1) = 'MESH'
3450           STITLE(2) = 'TEMP'
3460           STITLE(3) = 'DT'
3470           STITLE(4) = 'CP'
3480           STITLE(5) = 'COND'
```

```
3490            STITLE(6) = 'FINT'
3500            STITLE(7) = 'DFINT'
3510            STITLE(8) = 'TINT'
3520            STITLE(9) = 'DTINT'
3530            STITLE(10) = 'CINTM'
3540            STITLE(11) = 'CINTC'
3550            STITLE(12) = 'CONC'
3560            STITLE(13) = 'DCONC'
3570            STITLE(14) = 'VORT'
3580            STITLE(15) = 'DVORT'
3590            STITLE(16) = 'PSI'
3600            STITLE(17) = 'DPSI'
3610            STITLE(18) = 'BUOY'
3620            STITLE(19) = 'DIFF'
3630            STITLE(20) = 'VISC'
3640    C
3650            RETURN
3660            END
```

```
      SUBROUTINE INIT1
C
C INITIALIZE RADIAL ARRAYS AND CALL MESH
C
      INCLUDE 'COM./NOLIST'
C
C                                   ARRAYS FOR R DIRECTION
C
C
      CALL SCMBLP (NRP2,NRP1,RINS,RSAM,DRINS,DRSAM,
     1       RW,RH,DIDRW,DIDRH,DIDRMS)
C
C FLAG TO AVOID WRITING FIRST VALUES, AND TO SAVE WHOLE DERIVATIVE IN DX
C
      DIDRMA = 1757
      DIDRAB = DRAMPI
      IF (NRA .GT. 0)
     1       CALL SCMBLP (NRAP2,NRAP1,RSAM,RAMP,DIDRAB,DRAMPO,
     1       RW(NRP1),RH(NRP1),DIDRW(NRP1),DIDRH(NRP1),DIDRMA)
C
C WRITE MESH DIAGNOSTICS
C
      WRITE (6,5) 'RW',RW
      WRITE (6,5) 'RH',RH
      WRITE (6,5) 'DIDRW',DIDRW
      WRITE (6,5) 'DIDRH',DIDRH
5     FORMAT (//1X,A,T12,1P10E11.3,:/(T12,1P10E11.3,:))
C
C HALF POINT ARRAYS
C FOR THE CRYSTAL, TVBTRH MUST BE MULTIPLIED BY TAUT/TAUS WHEN USED.
C
      DO 3 I = 1, NRCP2
C
      RDRB2H(I) = RH(I) / DIDRH(I) / 2
      RB2DRH(I) = RH(I) * DIDRH(I) / 2
      RDRWQH(I) = RH(I) / DIDRH(I) * WAMP / 4.
C
      IF (I .GT. NRP1) THEN
            TVBTRH(I) = RH(I)/DIDRH(I) / TAUA / (DIDRMA/DIDRH(I))**P
      ELSE
            TVBTRH(I) = RH(I)/DIDRH(I) / TAUT / (DIDRMS/DIDRH(I))**P
            CVBTRH(I) = RH(I)/DIDRH(I) / TAUC / (DIDRMS/DIDRH(I))**P
            BTF(I) = 1. / TAUF / (DIDRMS/DIDRH(I))**PRF
            HBRDRH(I)  = .5 * DIDRH(I) / RH(I)
      ENDIF
3     CONTINUE
C
C WHOLE POINT ARRAYS
C
      DO 4 I = 1, NRCP1
      RDRB2W(I) = RW(I)/DIDRW(I)/2
      RB2DRW(I) = RW(I)*DIDRW(I)/2
4     CONTINUE
C
      DO 6 I = 1, NRP1
      R2W(I) = RW(I)**2
      R4W(I) = RW(I)**4
      DRB2RW(I) = .5 / DIDRW(I) / RW(I)
      UVBTRW(I) = RW(I)/DIDRW(I) / TAUU / (DIDRMS/DIDRW(I))**PRU
```

```
  590    6        CONTINUE
  600    C
  610    C SET UP Z-MESH AND ARRAYS
  620    C
  630             CALL MESH
  640    C
  650             RETURN
65688             END
```

```
      SUBROUTINE MESH
C
C GET Z MESH AND INTERFACE ARRAYS AT NEW TIME. CALLED FROM INIT1 AND TFS
C
      INCLUDE 'COM./NOLIST'
C
C EQUIVALENCE FOR SCMBLP CALL
C
C     DIMENSION ZESW(NZP1),DKDZSW(NZP1)
C     DIMENSION ZESH(NZP1),DKDZSH(NZP1)
C     EQUIVALENCE (ZESW,ZETW(NZP1)),(DKDZSW,DKDZTW(NZP1))
C     EQUIVALENCE (ZESH,ZETH(NZP1)),(DKDZSH,DKDZTH(NZP1))
C
C GET FMAX AND FMIN
C
      FMAX = FINTH(2)
      FMIN = FINTH(2)
      DO 10 I = 3, NRP1
      FMAX = MAX(FMAX,FINTH(I))
10    FMIN = MIN(FMIN,FINTH(I))
C
C SET ZL AND ZINT FOR THE COORDINATE TRANSFORMATION.
C IF THEY CHANGE, THEN THE INPUT ZINTL AND ZINTI WERE WRONG.
C IF THEY DON'T CHANGE, THEN ZETA AND ETA STAY FIXED.
C THESE STATEMENTS CAN BE CHANGED TO REFORMULATE ZINT AND ZL CHOICE
C
C THESE THREE CONSTANTS MUST BE IN DECREASING ORDER
C
      ZL1 = 16
      ZL2 = 10
      ZL3 = 6
C
      DFM = FMAX - FMIN
      ZLI = (RSAM-RINS) * ZINTL
C
      ZL = MAX( ZLI, DFM*ZL2, MIN(ZL,DFM*ZL1) )
C
      ZINT = MIN(ZINT,FMIN+ZL/ZL3)
      ZINT = MAX(ZINT,FMAX-ZL/ZL3)
C
C SKIP ZETA AND ETA SET-UP IF ZL AND ZINT WERE NOT CHANGED
C
      IF (ISTEP .GT. 0 .AND. ZL .EQ. ZLP  .AND.  ZINT .EQ. ZINTP)
     1       GO TO 80
C
C WRITE ONLY AT STEPS 0, 1, 2, 4, 8, ...
C
      IF (ISTEP .LE. 2) ISTEPN = ISTEP
      IF (ISTEP .EQ. ISTEPN) THEN
            ISTEPN = 2 * ISTEP
            WRITE (6,99) ISTEP, ZL, ZINT, FMAX, FMIN
99          FORMAT(/' MESH. ISTEP =',I4,9X,'ZL, ZINT, FMAX, FMIN ='
     1      ,9F8.4)
      ENDIF
C
C SAVE PREVIOUS VALUES
C
      ZLP = ZL
```

```
590           ZINTP = ZINT
600   C
610   C SET UP ZETA MESHES
620   C
630           CALL SCMBLP (NZP2,NZP1,ZT,ZINT,-DZT,-DZINTM,
640           1       ZETW,ZETH,DKDZTW,DKDZTH,DKDZMM)
650   C
660   C FLAG TO AVOID WRITING FIRST VALUES, AND TO SAVE WHOLE DERIVATIVE IN DX
670   C
680           DKDZMC = 1757
690           DKDZTC = -DZINTS
700           CALL SCMBLP (NZSP2,NZSP1,ZINT,ZB,DKDZTC,-DZB
710           1       ,ZETW(NZP1),ZETH(NZP1),DKDZTW(NZP1),DKDZTH(NZP1),DKDZMC)
720   C
730   C SET UP ETA ARRAYS FOR Z MESH
740   C Z = ZET + RF * ETA(ZET)
750   C ETA = 1 AT INTERFACE
760   C       0 AT TOP AND BOTTOM
770   C       SMALL EXCEPT NEAR INTERFACE
780   C
790   C HALF POINTS BELOW INTERFACE
800   C
810           DO 60 K = NZP2, NZCP2
820           ETAH(K) = (ZETH(K) - ZB) / (ZINT - ZB)
830           1       / (1. + ((ZINT - ZETH(K))/ZL))
840   60      DEDZTH(K) = (1./(ZINT - ZB) + ETAH(K)/ZL)
850           1       / (1. + ((ZINT - ZETH(K))/ZL))
860   C
870   C HALF POINTS ABOVE INTERFACE
880   C
890           DO 40 K = 1, NZP1
900           ETAH(K) = (ZT - ZETH(K)) / (ZT - ZINT)
910           1       / (1. + ((ZETH(K) - ZINT)/ZL))
920   40      DEDZTH(K) = - (1./(ZT-ZINT) + ETAH(K)/ZL)
930           1       / (1. + ((ZETH(K) - ZINT)/ZL))
940   C
950   C WHOLE POINTS BELOW INTERFACE
960   C
970           DO 70 K = NZP1, NZCP1
980           ETAW(K) = (ZETW(K) - ZB) / (ZINT - ZB)
990           1       / (1. + ((ZINT - ZETW(K))/ZL))
1000  70      DEDZTW(K) = (1./(ZINT - ZB) + ETAW(K)/ZL)
1010          1       / (1. + ((ZINT - ZETW(K))/ZL))
1020  C
1030  C WHOLE POINTS ABOVE AND ON THE INTERFACE
1040  C
1050          DO 50 K = 1, NZP1
1060          ETAW(K) = (ZT - ZETW(K)) / (ZT - ZINT)
1070          1       / (1. + ((ZETW(K) - ZINT)/ZL))
1080  50      DEDZTW(K) = - (1./(ZT-ZINT) + ETAW(K)/ZL)
1090          1       / (1. + ((ZETW(K) - ZINT)/ZL))
1100  C
1110  C CRYSTAL INTERFACE VALUE
1120  C
1130          DEDZTC = 1./(ZINT-ZB) + 1./ZL
1140  C
1150  C ALL HALF POINTS
1160  C
```

C-2

```
1170            DO 72 K = 1, NZCP2
1180            ETA2H(K) = ETAH(K)**2
1190            DZTDKH(K) = 1. / DKDZTH(K)
1200    72      DETDKH(K) = DEDZTH(K) * DZTDKH(K)
1210    C
1220    C ALL WHOLE POINTS
1230    C
1240            DO 74 K = 1, NZCP1
1250            ETA2W(K) = ETAW(K)**2
1260            DZTDKW(K) = 1. / DKDZTW(K)
1270    74      DETDKW(K) = DEDZTW(K) * DZTDKW(K)
1280    C
1290    C SET UP ARRAYS FOR THE VARIABLE TIME STEPS
1300    C
1310    C MELT
1320    C
1330            DO 15 K = 2, NZP1
1340            TVBTZH(K) = DZTDKH(K) * (DKDZMM/DKDZTH(K))**PZT
1350            UVBTZW(K) = DZTDKW(K) * (DKDZMM/DKDZTW(K))**PZU
1360            CVBTZH(K) = DZTDKH(K) * (DKDZMM/DKDZTH(K))**PZC
1370    15      CONTINUE
1380    C
1390    C CRYSTAL
1400    C
1410            DO 25 K = NZP2, NZCP1
1420            TVBTZH(K) = DZTDKH(K) * (DKDZMC/DKDZTH(K))**PZT
1430    25      CONTINUE
1440    C
1450    C END ZETA AND ETA SETUP
1460    C
1470    80      CONTINUE
1480    C
1490    C
1500    C SET INTERFACE ARRAYS, STARTING FROM FINTH AND DFINTH FROM 2 TO NRP1
1510    C
1520    C SET AXIS VALUES USING SYMMETRY
1530    C
1540             FINTH(1) =  FINTH(2)
1550            DFINTH(1) = DFINTH(2)
1560    C
1570    C SET FAMP AND DFAMP USING EXTRAPOLATION, SEE BLKDAT.
1580    C
1590             FAMP = 0
1600            DFAMP = 0
1610            DO 19 J = 1, 3
1620             FAMP =  FAMP +  FINTH(NRP2-J) * FABC(J)
1630    19      DFAMP = DFAMP + DFINTH(NRP2-J) * FABC(J)
1640    C
1650    C SET RFINT ARRAYS AT HALF AND WHOLE POINTS IN SAMPLE
1660    C
1670            DO 20 I = 1, NRP1
1680    20      RFINTH(I) = FINTH(I) - ZINT
1690    C
1700            DO 30 I = 1, NR
1710            DFINTW(I) = (DFINTH(I) + DFINTH(I+1) ) / 2.
1720    30      RFINTW(I) = (RFINTH(I) + RFINTH(I+1) ) / 2.
1730    C
1740    C SET RFINT AND DFINT ARRAYS AT HALF AND WHOLE POINTS IN AMPOULE
```

```
1750    C
1760            DO 22 I = NRP2, NRCP2
1770            DFINTH(I) = DFAMP
1780    22      RFINTH(I) = FAMP - ZINT
1790    C
1800            DO 32 I = NRP1, NRCP1
1810            DFINTW(I) = DFAMP
1820    32      RFINTW(I) = FAMP - ZINT
1830    C
1840    C SET ARRAYS FOR DFDR AND MULTIPLES
1850    C SINCE DFDR MAY BE SMOOTHER THAN DFDI (DUE TO THE NON-UNIFORM MESH),
1860    C WE EXTRAPOLATE DFDR NEAR THE BOUNDARY.
1870    C
1880            DO 33 I = 1, NR
1890    33      DFDRW(I) = DIDRW(I) * (FINTH(I+1) - FINTH(I))
1900            DFDRW(NRP1) = 2 * DFDRW(NR) - DFDRW(NR-1)
1910    C
1920            DO 34 I = 1, NRP1
1930            DFDR = DFDRW(I)
1940            RDRF8W(I) = RW(I) * DFDR / 8.
1950            DRF8RW(I) = DFDR / 8. / RW(I)
1960    34      ZDRFW(I) = DFDR*DFDR * DRB2RW(I)
1970    C
1980            DO 35 I = 2, NRP1
1990            DFDR = (DFDRW(I) + DFDRW(I-1)) * .5
2000            RDRF8H(I) = RH(I) * DFDR / 8
2010            DRF8RH(I) = DFDR / 8. / RH(I)
2020    35      ZDRFH(I) = DFDR*DFDR * RDRB2H(I)
2030    C
2040    C Z ARRAYS IN AMPOULE
2050    C
2060            DO 81 K = 1, NZCP2
2070            ZAH(K) = ZETH(K) + (FAMP-ZINT)*ETAH(K)
2080            DKDZAH(K) = DKDZTH(K)/(1. + (FAMP-ZINT)*DEDZTH(K))
2090    81      DZDKAH(K) = 1./DKDZAH(K)
2100    C
2110            DO 82 K = 1, NZCP1
2120            ZAW(K) = ZETW(K) + (FAMP-ZINT)*ETAW(K)
2130            DKDZAW(K) = DKDZTW(K)/(1. + (FAMP-ZINT)*DEDZTW(K))
2140    82      DZDKAW(K) = 1./DKDZAW(K)
2150    C
2160    C CORRECT MIDDLE TERM
2170    C
2180            DZDKAW(NZP1) = 0.5 * ( DZDKAH(NZP1)
2190            1       + (1 + (FAMP-ZINT)*DEDZTC) / DKDZTC )
2200            DKDZAW(NZP1) = 1./DZDKAW(NZP1)
2210    C
2220    C WRITE ONE-DIMENSIONAL MESH DIAGNOSTICS
2230    C
2240            IF (ISTEP .EQ. 0) THEN
2250                    WRITE (6,5) 'ZETW',ZETW
2260                    WRITE (6,5) 'ZETH',ZETH
2270                    WRITE (6,5) 'DKDZTW',DKDZTW
2280                    WRITE (6,5) 'DKDZTH',DKDZTH
2290                    WRITE (6,5) 'ETAW',ETAW
2300                    WRITE (6,5) 'ETAH',ETAH
2310                    WRITE (6,5) 'DEDZTW',DEDZTW
2320                    WRITE (6,5) 'DEDZTH',DEDZTH
```

```
2330    5               FORMAT (//1X,A,T12,1P10E11.3,:/(T12,1P10E11.3,:))
2340            ENDIF
2350    C
2360    C MESH DIAGNOSTICS
2370    C
2380            CALL MDIAG
2390            CALL PRNT (ZHH,NRCP2,1)
2400            CALL PRNT (FINTH,NRP2,6)
2410            IF (ISTEP .GT. 0) CALL PRNT (DFINTH,NRP2,7)
2420    C
2430    C ZHH ABOVE IS A DUMMY ARGUMENT, NOT USED BY PRNT.
2440    C
2450            RETURN
2460            END
```

```
      SUBROUTINE INIT2
C
C INIT2 INITIALIZES THE ARRAYS AND MESH,
C         USING ANALYTIC FORMS OR DIRECT ACCESS INPUT,
C       CALLS PRNT FOR THE INITIAL ARRAYS, AND
C       CALLS SMATPR, SINTPR, MESH, WHICH CALL PRNT.
C
C CALLED FROM MAIN, FOR INITIALIZATION
C ALSO CALLED FROM CMAIN, VIA CONT, TO DO PLOTS.
C
C SNREC SETS UP THE DIRECT ACCESS READ AND WRITE
C DDAOUT DOES THE DIRECT ACCESS OUTPUT
C
      INCLUDE 'COM./NOLIST'
C
      CHARACTER JTDES(72)
C
C ZHH FROM STATEMENT FUNCTION
C
      ZHH(I,K) = ZETH(K) + ETAH(K) * RFINTH(I)
C
C SET ISTEP AND LOGICAL DIAG.
C
      ISTEP = 0
      DIAG = .TRUE.
C
      IF ( ISEGR .EQ. 0 ) THEN
            TTIME = 0.
C
C                       GET TMELT AND CMELT FROM CCRYS=CTOP
C
            CALL GTMCM (TMELT,CMELT,CTOP)
            CINSC = PRPMAT(1,11) / WAMP
            WRITE (6,'(/'' INIT2.  '',
     1              ''TMELT, CMELT, CINSC ='',2F8.2,F10.5/)')
     1              TMELT, CMELT, CINSC
            FF = ZTBOT + (ZTTOP - ZTBOT)
     1      / (1.+PRPMAT(1,4)*(TTOP-TMELT)/PRPMAT(1,5)/(TMELT-TBOT))
C
            ZL = (RSAM-RINS) * ZINTL
            IF (ZINTI .EQ. 0) THEN
                  ZINT = FF
            ELSE
                  ZINT = ZINTI
            ENDIF
C
            DO 2 I = 1, NRP2
            TINTH(I) = TMELT
            CINTMH(I) = CMELT
            CINTCH(I) = CTOP
            DFINTH(I) = 0
            QR = 1 - .1*(1 - ((I-1.5)/NR)**2)
     2      FINTH(I) = FF * QR
C
            CALL INIT1
C
C MELT TEMPERATURE
            DO 11 K = 1, NZP1
```

```
590                     DO 11 I = 1, NRP1
600                     TT = TTOP - (ZTTOP-ZHH(I,K))/(ZTTOP-FINTH(I))
610           1                   * (TTOP-TINTH(I))
620                     TT = MAX (TT,TBOT)
630                     TT = MIN (TT,TTOP)
640     11              T(I,K) = TT
650     C CRYSTAL TEMPERATURE
660                     DO 12 K = NZP2, NZCP2
670                     DO 12 I = 1, NRP1
680                     TT = TBOT + (ZHH(I,K)-ZTBOT)/(FINTH(I)-ZTBOT)
690           1                   * (TINTH(I)-TBOT)
700                     TT = MAX (TT,TBOT)
710                     TT = MIN (TT,TTOP)
720     12              T(I,K) = TT
730     C AMPOULE TEMPERATURE
740                     DO 13 K = 1, NZCP2
750                     DO 13 I = NRP2, NRCP2
760                     TT = TBOT + (ZHH(I,K)-ZTBOT)/(ZTTOP-ZTBOT) * (TTOP-TBOT)
770                     TT = MAX (TT,TBOT)
780                     TT = MIN (TT,TTOP)
790     13              T(I,K) = TT + (T(NRP1,K) - TT) * (RAMP - RH(I)) / (RAMP
800     C CONCENTRATION
810                     DO 3 I = 1, NRP1
820                     DO 3 K = 1, NZP2
830     3               C(I,K) = CTOP + (CMELT-CTOP)
840             1       *(EXP((FINTH(I)-ZHH(I,K))/CINSC)
850             1       - EXP((FINTH(I)-ZT)/CINSC))
860             1       /(1 - EXP((FINTH(I)-ZT)/CINSC))
870     C STREAM FUNCTION
880                     CALL SETPSI
890             ELSE
900                     CALL DAOPN(NR,21)
910                     CALL DAOPN(NRC,22)
920     C
930                     IREC1 = IREC1R
940                     IREC2 = IREC2R
950                     CALL DAINP(FINTH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
960                     CALL DAINP(DFINTH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
970                     CALL DAINP(TINTH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
980                     CALL DAINP(CINTMH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
990                     CALL DAINP(CINTCH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
1000                    CALL DAINP(C,1,1,NRP2,NZP2,1,21,IREC1)
1010                    CALL DAINP(PSI,1,1,NRP2,NZP2,1,21,IREC1)
1020    C
1030                    CALL DAINP(T,1,1,NRCP2,NZCP2,1,22,IREC2)
1040    C
1050    C READ SEGMENT DESCRIPTOR FROM SINGLE RECORD
1060    C
1070                    LJT = MIN (72,NCPRN*(NRC-3))
1080                    READ (22,REC=IREC2) TTIME,ZL,ZINT, (JTDES(I),I=1,LJT)
1090                    WRITE (6,10) ISEGR, TTIME,ZL,ZINT, (JTDES(I),I=1,LJT)
1100    10              FORMAT (/' INIT2. ISEGR, TTIME =',I6,F14.3/
1110          1                       T8,'ZL, ZINT =',2F10.5/10X,72A1)
1120    C
1130                    CLOSE (21,DISP='KEEP')
1140                    CLOSE (22,DISP='KEEP')
1150    C
1160                    CALL INIT1
```

```
1170            ENDIF
1180    C
1190    C APPLY BOUNDARY CONDITIONS
1200    C
1210            CALL TBC
1220            CALL CBC
1230            CALL PSIBC
1240    C
1250    C DIAGNOSTICS FOR INITIAL CONDITIONS AND FOR WHEN INIT2 IS CALLED FROM C
1260    C
1270            CALL PRNT (T,NRCP2,2)
1280            CALL TDIAG
1290    C
1300            IF (LC) THEN
1310                    CALL PRNT (C,NRP2,12)
1320                    CALL CDIAG
1330            ENDIF
1340    C
1350            IF (LU) THEN
1360                    CALL PRNT (PSI,NRP2,16)
1370                    CALL PDIAG
1380                    CALL GVORT
1390                    CALL PRNT (VORT,NRP1,14)
1400                    CALL VDIAG
1410            ELSE
1420                    CALL PRNT (PSI,NRP2,16)
1430            ENDIF
1440    C
1450            CALL SINTPR
1460            CALL SMATPR
1470    C
1480    C DO A SINGLE STEP TO PLOT CHANGES IF REQUIRED
1490    C
1500            IF (LCONT) THEN
1510                    IF (PRTEST(3)  .OR.  (LC  .AND.  PRTEST(13)) ) THEN
1520                            IDAIOS = IDAIO
1530                            IDAIO = -1
1540                            IPC(13,3) = -1
1550                            IPC(13,13) = -1
1560                            ISTEP = 1
1570                            CALL TFSTEP
1580                            IDAIO = IDAIOS
1590                    ENDIF
1600    C                       IF (LU .AND. (PRTEST(15)  .OR.  PRTEST(17)) CALL USTEP
1610            ENDIF
1620    C
1630            RETURN
1640            END
```

```
  10            SUBROUTINE SNREC
  20    C
  30    C DIRECT ACCESS READ AND WRITE SEGMENTS
  40    C FOR FIRST CASE THE 'NEW' VALUES ARE SET IN SETUP AS 0, 1.
  50    C LATER THEY ARE SET BY DAREC.
  60    C ISEGRU AND ISEGWU ARE USED. ISEGR AND ISEGW ARE LEFT FOR LATER PRINTIN
  70    C
  80    C PRINT DIRECT ACCESS RECORD RANGE INFORMATION
  90    C GET ISEGRN, ISEGWN.
 100    C GET ISEGRU, ISEGWU.
 110    C GET IRECR , IRECW , FOR BOTH DIRECT ACCESS FILES.
 120    C
 130            INCLUDE 'COM./NOLIST'
 140    C
 150            CALL DAREC ('READ ',ISEGR,ISEGRN,1,1,ISEGRU)
 160            NOUT = MAX(0,(NSTEP-IDAOUT+IINCDA)/IINCDA)
 170            CALL DAREC ('WRITE',ISEGW,ISEGWN,NOUT,1,ISEGWU)
 180    C
 190    C SET NREC1 AND NREC2 TO MATCH THE DAINP AND DAOUT CALLS
 200    C
 210            NREC1 = 3
 220            NREC1 = NREC1 + 2 + NZ
 230            NREC1 = NREC1 + NZ
 240            NREC2 = NZC + 1
 250    C
 260            IREC1R = 1 + (ISEGRU - 1) * NREC1
 270            IREC1W = 1 + (ISEGWU - 1) * NREC1
 280            IREC2R = 1 + (ISEGRU - 1) * NREC2
 290            IREC2W = 1 + (ISEGWU - 1) * NREC2
 300    C
 310            RETURN
65538           END
```

```
 10            SUBROUTINE DDAOUT
 20    C
 30    C DO DIRECT ACCESS OUTPUT
 40    C CALLED FROM TFSTEP
 50    C PLACED HERE TO SIMPLIFY CHANGES IN THE STRUCTURE
 60    C
 70            INCLUDE 'COM./NOLIST'
 80    C
 90            I = ISTEP - IDAOUT
100            J = I/IINCDA
110            IF (I .LT. 0  .OR.  J*IINCDA .NE. I) RETURN
120            ISEG = ISEGW + J
130    C
140            WRITE (*,311) ISTEP,ICASE, ISEG
150            WRITE (6,311) ISTEP,ICASE, ISEG
160    311     FORMAT (/' STARTING DAOUT AT STEP',I6,'  OF CASE',I2
170            1       ,'.  SEGMENT',I3/)
180    C
190            IREC1 = IREC1W + J * NREC1
200            IREC2 = IREC2W + J * NREC2
210    C
220            CALL DAOPN(NR,21)
230            CALL DAOPN(NRC,22)
240    C
250            CALL DAOUT( FINTH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
260            CALL DAOUT(DFINTH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
270            CALL DAOUT( TINTH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
280            CALL DAOUT(CINTMH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
290            CALL DAOUT(CINTCH(1-NRP2),1,1,NRP2,3,1,21,IREC1)
300            CALL DAOUT(C,1,1,NRP2,NZP2,1,21,IREC1)
310            CALL DAOUT( PSI,1,1,NRP2,NZP2,1,21,IREC1)
320    C
330            CALL DAOUT(T,1,1,NRCP2,NZCP2,1,22,IREC2)
340    C
350            LJTIM = MIN(28,NCPRN*(NRC-4))
360            LDESC = MIN(44,NCPRN*(NRC-4)-28)
370    C
380    C WRITE SEGMENT DESCRIPTOR TO SINGLE RECORD
390    C
400            LJTIM = MAX(MIN(28,NCPRN*(NRC-4)),1)
410            LDESC = MAX(MIN(44,NCPRN*(NRC-4)-28),1)
420            WRITE (22,REC=IREC2) TTIME,ZL,ZINT,JTIME(1:LJTIM),DESCR(1:LDESC)
430    C
440            CLOSE (21,DISP='KEEP')
450            CLOSE (22,DISP='KEEP')
460    C
470            RETURN
65488          END
```

```
10            BLOCK DATA BLKDT
20    C
30            INCLUDE 'COM./NOLIST'
40    C
50            DATA PI /3.141592654/
60            DATA FABC /1.875,-1.25,0.375/
70    C       DATA FABC /1.5,-0.5,0./
80    C       DATA FABC /1.,0.,0./
90            END
```

```
   10
   20            PROGRAM CMAIN
   30    C
   40    C MAIN PROGRAM TO CALL CONT FOR POST-PROCESSING.
   50    C THE DATA FILE SHOULD BE CHANGED TO GET DIFFERENT PLOTS
   60    C
   70            INCLUDE 'COM./NOLIST'
   80    C
   90    C LCONT IS TRUE FOR THIS PROGRAM.
  100    C
  110            LCONT = .TRUE.
  120    C
  130            CALL SETUP
  140    C
  150    C CASE DO LOOP TERMINATES IN INIT0 WHEN DATA RUNS OUT
  160    C
  170            DO 20 ICASE = 1, 100
  180    C
  190    C INPUT SEGMENT
  200    C
  210            CALL INIT0
  220    C
  230            CALL LINEPL
  240            CALL CONT
  250    C
  260    20      CONTINUE
  270    C
  280            CALL STOPP ('CMAIN. TOO MANY CASES%')
65298            END
```

```
   10            SUBROUTINE CONT
   20    C
   30    C CONTOUR PLOT OUTPUT
   40    C
   50            INCLUDE 'COM./NOLIST'
   60    C
   70    C IDAIO LOOP
   80    C
   90            NOUT = (NSTEP + IINCDA - IDAOUT) / IINCDA
  100    C
  110    C LOOP FOR IDAIO
  120    C
  130            DO 100 IDAIO = 0, NOUT
  140    C
  150    C FIND WHETHER DIRECT ACCESS READ IS REQUIRED
  160    C
  170            DO 10 IAPC = 1, NAPC
  180            L = 13
  190            J = IDAIO - IPC(L,IAPC)
  200            K = IPC(L+1,IAPC)
  210            IF ((J/K)*K .EQ. J .AND. J .GE. 0) GO TO 11
  220    10      CONTINUE
  230            GO TO 100
  240    11      CONTINUE
  250    C
  260    C GET RECORD FOR INIT2 AND MAKE CALL AS WITH RESTART.
  270    C IT CALLS PRNT, WHICH CALLS CPR.
  280    C
  290            IREC1R = IREC1W + (IDAIO-1) * NREC1
  300            IREC2R = IREC2W + (IDAIO-1) * NREC2
  310            ISEGR = ISEGW + (IDAIO-1) * 1
  320            IF (IDAIO .EQ. 0) ISEGR = 0
  330    C
  340            CALL INIT2
  350    C
  360    100     CONTINUE
  370    C
  380            RETURN
65598            END
```

```
      SUBROUTINE CPR (VAR,ND,NCZ,IAPC,NCONT
     1               ,NHHI,RPLL,RPLR,ZPLB,ZPLT)
C
C CONTOUR PLOT OUTPUT. CALLED FROM PRNT.
C
      INCLUDE 'COM./NOLIST'
C
      PARAMETER (MXMYMAX = 4 * NRC * NZCP2)
      PARAMETER (MXMAX = 4 * NRC)
      COMMON W2(MXMYMAX),W3(4,MXMAX)
C
      DIMENSION VAR(1)
      DIMENSION RINT(1)
      EQUIVALENCE (RINT,A)
C
C PLOT MESH FOR IAPC = 1, UNDER PHYSICAL CONTOUR PARAMETERS CONTROL
C
      IF (IAPC .EQ. 1) THEN
            J = IDAIO - IPC(13,IAPC)
            K = IPC(14,IAPC)
            IF ((J/K)*K .EQ. J .AND. J .GE. 0)
     1            CALL MESHPL(RPLL,RPLR,ZPLB,ZPLT)
            RETURN
      ENDIF
C
C SPECIAL TREATMENT OF ONE-DIMENSIONAL PLOTS
C
      IF (NCZ .LE. 1) RETURN
C
C IS ZERO CONTOUR TO BE PLOTTED? NOT IF AC = -1.
C
      AC = ACONT(IAPC)
C
C CONTOUR AGAINST PHYSICAL VARIABLES.
C
      J = IDAIO - IPC(13,IAPC)
      K = IPC(14,IAPC)
      IF ( ((J/K)*K .NE. J) .OR. (J .LT. 0) ) GO TO 20
      CALL CPL (VAR,ND,NCZ,IAPC,NCONT,AC
     1               ,NHHI,RPLL,RPLR,ZPLB,ZPLT)
20    CONTINUE
C
C CONTOUR AGAINST INTEGERS
C
      IPHYSICAL = 2
      J = IDAIO - IPC(15,IAPC)
      K = IPC(16,IAPC)
      IF ( ((J/K)*K .NE. J) .OR. (J .LT. 0) ) GO TO 40
C
      N = MAX0(NCZ,ND)
      DO 30 I = 1, N
30    RINT(I) = I
C
      XT = ND
      YT = NCZ
C
      MX = ND
      MY = NCZ
```

```
  590    C
  600            IF (MX*MY.GT.MXMYMAX .OR. MX.GT.MXMAX)
  610           1                CALL STOPP ('CPR DIMENSION%')
  620            CALL CONPL (VAR,RINT,RINT,ND,NCZ
  630           1                ,IPHYSICAL,NCONT,AC,IAPC,TTIME
  640           1                ,1. ,XT ,1.,YT
  650           1                ,W2,MX,MY,W3
  660           1                ,LTITLE(IAPC),DESCR,MZW,1,1)
  670    40      CONTINUE
  680    C
  690            RETURN
65300            END
```

```
      SUBROUTINE CPL (VAR,MX,MY,IAPC,NCONT,AC
     1        ,NHHI,RPLL,RPLR,ZPLB,ZPLT)
C
C CONTOUR PLOT ON THE TEKTRONIX IN WORLD SPACE
C
      INCLUDE 'COM./NOLIST'
C
C    CONTVL IS AN ARRAY OF VALUES AT WHICH THE CONTOURS WILL BE PLOTTED.
C    NCONT IS APPROXIMATE NUMBER OF CONTOURS DESIRED, IF ACONT IS ZERO.
C
      DIMENSION VAR(MX,MY)
      PARAMETER (MCONTN = 30)
      DIMENSION CONTVL(MCONTN)
      CHARACTER *40  BLANK
      DATA BLANK /' '/
C
C STATEMENT FUNCTION TO GET V AT WW POINTS FROM V AT HH POINTS
C
      V(I,K)=(VAR(I,K)+VAR(I,K+NHHI)
     1        +VAR(I+NHHI,K)+VAR(I+NHHI,K+NHHI)) * 0.25
C
C ZWW AND DZDKWH FROM STATEMENT FUNCTIONS
C
      ZWW(I,K) = ZETW(K) + ETAW(K) * RFINTW(I)
      DZDKWH(I,K) = DZTDKH(K) + DETDKH(K) * RFINTW(I)
C
C GET MAXIMUM AND MINIMUM
C
      VMAX = V(1,1)
      VMIN = V(1,1)
      DO 3 K = 1, MY - NHHI
      DO 3 I = 1, MX - NHHI
      VMAX = MAX(V(I,K),VMAX)
3     VMIN = MIN(V(I,K),VMIN)
      IF (VMAX .EQ. VMIN) RETURN
C
C GET CONTOUR LEVELS
C
      IF (AC .EQ. 0  .OR.  AC .EQ. -1) THEN
            VINC = ROUND((VMAX-VMIN)/NCONT)
      ELSE
            VINC = ABS(AC)
      ENDIF
C
C GET MIN AND MAX CONTOUR LEVELS
C DON'T PLOT ZERO CONTOUR IF EXTREMUM IS CLOSE, SINCE
C   THE VALUES ARE PROBABLY SPURIOUS. THE CLOSENESS
C   CRITERION IS EPS, WHICH CAN BE CHANGED AT WILL.
C   VALUES BETWEEN ZERO AND 0.2 ARE APPROPRIATE.
C
      EPS = 0.1
C
      VR = VMIN/VINC
      ICMIN = VR + 4000.
      ICMIN = ICMIN - 4000 + 1
      IF (VR .LE. 0  .AND.  VR .GT. -EPS) ICMIN = 1
C
      VR = VMAX/VINC
```

```
590             ICMAX = VR + 4000.
600             ICMAX = ICMAX - 4000
610             IF (VR .GE. 0  .AND.  VR .LT.  EPS) ICMAX = -1
620     C
630     C DON'T PLOT FUNCTIONS WHICH ARE APPROXIMATELY CONSTANT,
640     C  BUT WRITE THE TEXT.
650     C
660             ICMIN = MAX (ICMIN, -3000)
670             ICMAX = MIN (ICMAX, +3000)
680     C
690     C GET THE CONTOUR LEVELS ARRAY
700     C
710             NCONTN = ICMAX - ICMIN + 1
720             IF (NCONTN .GT. MCONTN) THEN
730                     WRITE(6,*)' CPL. NCONTN REDUCED FROM'
740             1                ,NCONTN,'TO MCONTN =',MCONTN
750                     NCONTN = MCONTN
760             ENDIF
770             DO 2 I = 1, NCONTN
780     2       CONTVL(I) = VINC * (I - 1 + ICMIN)
790     C
800     C CLEAR SCREEN
810     C GET TERMINAL WINDOW AND RATIO
820     C
830             CALL INTERM
840             CALL SEETW (MINX,MAXX,MINY,MAXY)
850             MB = MAXX / 50
860             MINX = MINX + MB
870             MINY = MINY + MB
880             MAXX = MAXX - MB
890             MAXY = MAXY - MAX(MB,MAXX/20)
900     C
910             XR = MAXX - MINX
920             YR = MAXY - MINY
930     C
940     C GET NUMBER OF LINES OF TEXT AND RASTER EQUIVALENT
950     C GET MAXIMUM NUMBER OF CHARACTERS AND RASTER EQUIVALENT
960     C
970             NLT = 6
980             IF (TTIME .NE. 0) NLT = NLT + 1
990             RNLT = NLT * YR / NLPP
1000    C
1010            LD = LENGTH(DESCR,44)
1020            LT = LENGTH(LTITLE(IAPC),42)
1030            MNC = MAX(24,LD,LT) + 6
1040            RMNC = MNC * XR / NCPL
1050    C
1060    C GET RECTANGLE HEIGHT, WIDTH, AND RATIO
1070    C
1080            HR = ZPLT - ZPLB
1090            WR = (RPLR - RPLL) * RSTRPL
1100            HOW = HR / WR
1110    C
1120    C GET SCALE WITH TEXT AT LEFT
1130    C
1140            SCALEL = MAX(HR/YR,WR/(XR-RMNC) )
1150    C
1160    C GET SCALE WITH TEXT AT BOTTOM
```

```
1170    C
1180            SCALEB = MAX(HR/(YR-RNLT),WR/XR)
1190            SCALE = MIN (SCALEB,SCALEL)
1200    C
1210    C PLACE TEXT AND TERMINAL WINDOW FOR LEFT
1220    C
1230            IF (SCALEL .LT. SCALEB) THEN
1240    C
1250                    NBL = (NLPP - NLT) / 2
1260    C
1270                    NBC = 1 + NBCPL
1280    C
1290                    MINX = MINX + RMNC
1300                    MINY = MINY + (YR/2) * MAX(0.,1.-HOW/(YR/(XR-RMNC)))
1310    C
1320    C PLACE TEXT AND TERMINAL WINDOW FOR BOTTOM
1330    C
1340            ELSE
1350    C
1360                    NBL = NLPP - NLT
1370                    NBC = 1 + NBCPL + (NCPL - MNC) / 2
1380    C
1390                    MINY = MINY + RNLT
1400                    MINX = MINX + (RX/2) * MAX(0.,1.-((YR-RNLT)/XR)/HOW)
1410    C
1420            ENDIF
1430    C
1440    C GET THE OPPOSITE CORNER OF THE WINDOW
1450    C
1460            MAXX = MINX + WR / SCALE
1470            MAXY = MINY + HR / SCALE
1480    C
1490    C TEXT ON PLOT
1500    C
1510            DO 120 IBL = 1, NBL-NLPA
1520    120     WRITE (1,130)
1530    130     FORMAT (1X)
1540            WRITE (1,140)BLANK(1:NBC),LTITLE(IAPC)(1:LT)
1550            WRITE (1,140)BLANK(1:NBC),DESCR(1:LD)
1560    140     FORMAT (A,A)
1570            WRITE (1,150)BLANK(1:NBC),VMAX
1580            WRITE (1,160)BLANK(1:NBC),VMIN
1590            WRITE (1,170)BLANK(1:NBC),VINC
1600            IF (TTIME .NE. 0) WRITE (1,180)BLANK(1:NBC),TTIME
1610    150     FORMAT (A,'MAXIMUM   =',1PG13.5)
1620    160     FORMAT (A,'MINIMUM   =',1PG13.5)
1630    170     FORMAT (A,'INCREMENT =',1PG13.5)
1640    180     FORMAT (A,'TIME      =',1PG13.5)
1650    C
1660    C SET PLOT WINDOW AND TRANSFORM
1670    C
1680            EPSW = .02
1690    C
1700            RPWL = RPLL - EPSW * WR / RSTRPL
1710            RPWR = RPLR + EPSW * WR / RSTRPL
1720            ZPWB = ZPLB - EPSW * HR
1730            ZPWT = ZPLT + EPSW * HR
1740    C
```

```
1750            CALL DWINDO(RPWL,RPWR,ZPWB,ZPWT)
1760    C
1770            CALL TWINDO (MINX,MAXX,MINY,MAXY)
1780    C
1790    C
1800    C DRAW BOUNDARY, INTERFACE, AND AMPOULE.
1810    C
1820            IF (IMACH .NE. 2) CALL CZAXIS(1)
1830            RTICK = .05 * (RAMP - RINS)
1840    C
1850            DO 190 K = 1, NZCP1, NZC
1860            CALL MOVEA(RW(1),ZWW(1,K))
1870            I = NRCP1
1880    190     CALL DASHA(RW(I),ZWW(I,K),12)
1890    C
1900            CALL MOVEA(RW(1),ZWW(1,NZP1))
1910            DO 192 I = 2, NRP1
1920    192     CALL DASHA(RW(I),ZWW(I,NZP1),31)
1930    C
1940            DO 195 I = NRP1, NRCP1, NRA
1950            CALL MOVEA(RW(I),ZWW(I,1)+RTICK)
1960    195     CALL DRAWA(RW(I),ZWW(I,NZCP1)-RTICK)
1970    C
1980    C
1990            CALL MOVEA(RW(1),ZWW(1,1)+RTICK)
2000    198     CALL DASHA(RW(1),ZWW(1,NZCP1)-RTICK,3212)
2010    C
2020    C DRAW TICKS AT FURNACE ENDS AND MELT
2030    C
2040            RTICK = .05 * (RAMP - RINS)
2050    C
2060            CALL MOVEA(RAMP,ZTTOP)
2070            CALL DRAWR(RTICK,0.)
2080    C
2090            CALL MOVEA(RAMP,ZTBOT)
2100            CALL DRAWR(RTICK,0.)
2110    C
2120            CALL MOVEA(RINS,FINTH(2))
2130            CALL DRAWR(-RTICK,0.)
2140    C
2150            CALL CZAXIS(0)
2160    C
2170    C                                       MAIN LOOP OVER CELLS
2180    C
2190            DO 80 I = 1, MX - 1 - NHHI
2200            DO 70 K = 1, MY - 1 - NHHI
2210    C
2220    C GET 4 CORNER VALUES ON WW MESH USING STATEMENT FUNCTION V
2230    C NHHI IS 0 FOR WW FUNCTIONS
2240    C
2250            V1 = V(I,K)
2260            V2 = V(I+1,K)
2270            V3 = V(I+1,K+1)
2280            V4 = V(I,K+1)
2290    C
2300    C       K+1.... V4    S3    V3
2310    C
2320    C                S4         S2
```

```
2330    C
2340    C       K...... V1    S1    V2
2350    C                |           |
2360    C                |           |
2370    C                I           I+1
2380    C
2390    C
2400              VMN = AMIN1(V1,V2,V3,V4)
2410              VMX = AMAX1(V1,V2,V3,V4)
2420    C
2430    C CALCULATE INNER LOOP OVER CONTOURS
2440    C
2450              IF ((CONTVL(1) .GT. VMX).OR.(VMN .GT. CONTVL(NCONTN))) GO TO 70
2460              LB = 1
2470              LT = NCONTN
2480              IF (NCONTN .EQ. 1.) GO TO 110
2490              DO 90 L = 1,NCONTN
2500              IF (VMN .GT. CONTVL(L)) LB = L + 1
2510              IF (VMX .GT. CONTVL(L)) LT = L
2520    90        CONTINUE
2530    C
2540    C DO INNER LOOP OVER CONTOURS
2550    C
2560    110       DO 60 L = LB,LT
2570              CONTPV = CONTVL(L)
2580              IF (CONTPV .EQ. 0.  .AND.  AC .LT. 0.) GO TO 60
2590              Q1 = V1-CONTPV
2600              Q2 = V2-CONTPV
2610              Q3 = V3-CONTPV
2620              Q4 = V4-CONTPV
2630    C
2640              IF (Q1 .EQ. 0.) Q1 = VINC * 1.E-6
2650              IF (Q2 .EQ. 0.) Q2 = VINC * 1.E-6
2660              IF (Q3 .EQ. 0.) Q3 = VINC * 1.E-6
2670              IF (Q4 .EQ. 0.) Q4 = VINC * 1.E-6
2680    C
2690              ONE = 1
2700              T1 = SIGN (ONE,Q1)
2710              T2 = SIGN (ONE,Q2)
2720              T3 = SIGN (ONE,Q3)
2730              T4 = SIGN (ONE,Q4)
2740    C
2750              S1 = T1 * T2
2760              S2 = T2 * T3
2770              S3 = T3 * T4
2780              S4 = T4 * T1
2790    C
2800              IF (S1 .GE. 0) GO TO 10
2810              CALL MOVEA(RW(I)+Q1/(Q1-Q2)/DIDRH(I+1),
2820              1        ZWW(I,K  )+(ZWW(I+1,K  )-ZWW(I,K  ))*Q1/(Q1-Q2))
2830              IF (S2 .GE. 0) GO TO 20
2840              CALL DRAWA(RW(I+1),ZWW(I+1,K)+Q2/(Q2-Q3)*DZDKWH(I+1,K+1))
2850              GO TO 30
2860    10        IF (S2 .GE. 0) GO TO 30
2870              CALL MOVEA(RW(I+1),ZWW(I+1,K)+Q2/(Q2-Q3)*DZDKWH(I+1,K+1))
2880    20        IF (S3 .GE. 0) GO TO 40
2890              CALL DRAWA(RW(I)+Q4/(Q4-Q3)/DIDRH(I+1),
2900              1        ZWW(I,K+1)+(ZWW(I+1,K+1)-ZWW(I,K+1))*Q4/(Q4-Q3))
```

```
 2910             GO TO 50
 2920    30       IF (S3 .GE. 0) GO TO 50
 2930             CALL MOVEA(RW(I)+Q4/(Q4-Q3)/DIDRH(I+1),
 2940             1        ZWW(I,K+1)+(ZWW(I+1,K+1)-ZWW(I,K+1))*Q4/(Q4-Q3))
 2950    40       IF (S4 .LT. 0)
 2960             1        CALL DRAWA(RW(I),ZWW(I,K)+Q1/(Q1-Q4)*DZDKWH(I,K+1))
 2970    50       CONTINUE
 2980    60       CONTINUE
 2990    70       CONTINUE
 3000    80       CONTINUE
 3010    C
 3020             CALL HDCOPY
 3030             CALL CLTERM
 3040    C
 3050             RETURN
65888             END
```

```
      SUBROUTINE MESHPL(RPLL,RPLR,ZPLB,ZPLT)
C
C PLOT MESH CURVES ON THE TEKTRONIX IN WORLD SPACE
C
      INCLUDE 'COM./NOLIST'
C
      CHARACTER *40  BLANK
      DATA  BLANK /' '/
C
C ZWW FROM STATEMENT FUNCTION
C
      ZWW(I,K) = ZETW(K) + ETAW(K) * RFINTW(I)
C
C
C CLEAR SCREEN
C GET TERMINAL WINDOW AND RATIO
C
      CALL INTERM
      CALL SEETW (MINX,MAXX,MINY,MAXY)
      MB = MAXX / 50
      MINX = MINX + MB
      MINY = MINY + MB
      MAXX = MAXX - MB
      MAXY = MAXY - MAX(MB,MAXX/20)
C
      XR = MAXX - MINX
      YR = MAXY - MINY
C
C GET NUMBER OF LINES OF TEXT AND RASTER EQUIVALENT
C GET MAXIMUM NUMBER OF CHARACTERS AND RASTER EQUIVALENT
C
      NLT = 2
      RNLT = NLT * YR / NLPP
C
      LD = LENGTH(DESCR,44)
      MNC = MAX(24,LD) + 6
      RMNC = MNC * XR / NCPL
C
C GET RECTANGLE HEIGHT, WIDTH, AND RATIO
C
      HR = ZPLT - ZPLB
      WR = (RPLR - RPLL) * RSTRPL
      HOW = HR / WR
C
C GET SCALE WITH TEXT AT LEFT
C
      SCALEL = MAX(HR/YR,WR/(XR-RMNC) )
C
C GET SCALE WITH TEXT AT BOTTOM
C
      SCALEB = MAX(HR/(YR-RNLT),WR/XR)
      SCALE = MIN (SCALEL,SCALEB)
C
C PLACE TEXT AND TERMINAL WINDOW FOR LEFT
C
      IF (SCALEL .LT. SCALEB) THEN
C
            NBL = (NLPP - NLT) / 2
```

```
590   C
600                   NBC = 1 + NBCPL
610   C
620                   MINX = MINX + RMNC
630                   MINY = MINY + (YR/2) * MAX(0.,1.-HOW/(YR/(XR-RMNC)))
640   C
650   C PLACE TEXT AND TERMINAL WINDOW FOR BOTTOM
660   C
670           ELSE
680   C
690                   NBL = NLPP - NLT
700                   NBC = 1 + NBCPL + (NCPL - MNC) / 2
710   C
720                   MINY = MINY + RNLT
730                   MINX = MINX + (RX/2) * MAX(0.,1.-((YR-RNLT)/XR)/HOW)
740   C
750           ENDIF
760   C
770   C GET THE OPPOSITE CORNER OF THE WINDOW
780   C
790           MAXX = MINX + WR / SCALE
800           MAXY = MINY + HR / SCALE
810   C
820   C TEXT ON PLOT
830   C
840           DO 120 IBL = 1, NBL-NLPA
850   120     WRITE (1,130)
860   130     FORMAT (1X)
870           WRITE (1,140)BLANK(1:NBC),DESCR(1:LD)
880           WRITE (1,140)BLANK(1:NBC),'COMPUTATIONAL MESH'
890   140     FORMAT (A,A)
900   C
910   C SET PLOT WINDOW AND TRANSFORM
920   C
930           EPSW = .02
940   C
950           RPWL = RPLL - EPSW * WR / RSTRPL
960           RPWR = RPLR + EPSW * WR / RSTRPL
970           ZPWB = ZPLB - EPSW * HR
980           ZPWT = ZPLT + EPSW * HR
990   C
1000          CALL DWINDO(RPWL,RPWR,ZPWB,ZPWT)
1010  C
1020          CALL TWINDO (MINX,MAXX,MINY,MAXY)
1030  C
1040  C
1050  C DRAW MESH, EXCLUDING BOUNDARIES, INCLUDING INTERFACE
1060  C
1070          DO 192 K = 2, NZC
1080          CALL MOVEA(RW(1),ZWW(1,K))
1090          DO 190 I = 2, NRCP1
1100  190     CALL DRAWA(RW(I),ZWW(I,K))
1110  192     CONTINUE
1120  C
1130          DO 197 I = 2, NRC
1140          IF (I .EQ. NRP1) GO TO 197
1150          CALL MOVEA(RW(I),ZWW(I,1))
1160  195     CALL DRAWA(RW(I),ZWW(I,NZCP1))
```

```
1170    197     CONTINUE
1180    C
1190    C DRAW BOUNDARIES AND INTERFACES
1200    C
1210            IF (IMACH .NE. 2) CALL CZAXIS(1)
1220            RTICK = .05 * (RAMP - RINS)
1230    C
1240    C DASH TOP AND BOTTOM
1250    C
1260            DO 290 K = 1, NZCP1, NZC
1270            CALL MOVEA(RW(1),ZWW(1,K))
1280            I = NRCP1
1290    290     CALL DASHA(RW(I),ZWW(I,K),12)
1300    C
1310    C DASH INTERFACE SEGMENT
1320    C
1330    C       CALL MOVEA(RW(1),ZWW(1,NZP1))
1340    C       DO 292 I = 2, NRP1
1350    C292    CALL DASHA(RW(I),ZWW(I,NZP1),31)
1360    C
1370    C INSIDE AND OUTSIDE OF AMPOULE
1380    C
1390            DO 295 I = NRP1, NRCP1, NRA
1400            CALL MOVEA(RW(I),ZWW(I,1)+RTICK)
1410    295     CALL DRAWA(RW(I),ZWW(I,NZCP1)-RTICK)
1420    C
1430    C DASH AXIS
1440    C
1450            CALL MOVEA(RW(1),ZWW(1,1)+RTICK)
1460    298     CALL DASHA(RW(1),ZWW(1,NZCP1)-RTICK,3212)
1470    C
1480    C DRAW TICKS AT FURNACE ENDS AND MELT
1490    C
1500            CALL MOVEA(RAMP,ZTTOP)
1510            CALL DRAWR(RTICK,0.)
1520    C
1530            CALL MOVEA(RAMP,ZTBOT)
1540            CALL DRAWR(RTICK,0.)
1550    C
1560            CALL MOVEA(RINS,FINTH(2))
1570            CALL DRAWR(-RTICK,0.)
1580    C
1590            CALL CZAXIS(0)
1600    C
1610    C
1620            CALL HDCOPY
1630            CALL CLTERM
1640    C
1650            RETURN
65600           END
```

```
      SUBROUTINE LINEPL
C
C LINE PLOT OUTPUT
C
      INCLUDE 'COM./NOLIST'
C
C EQUIVALENCE STATEMENTS TO ALLOCATE SPACE FOR THE ARRAYS
C
      PARAMETER (NKMAX = 2, NTMAX = 21)
      DIMENSION VAR(NZP2,NKMAX,NTMAX),RAR(NZP2),TAR(NTMAX)
      EQUIVALENCE (DT,VAR)
      CHARACTER *120  BLANK
      DATA BLANK /' '/
C
C LOOP OVER PLOT TYPES IN .Ynn FILE PRODUCED BY PRNT AND MOVIE IN PRIOR
C
      DO 200 IAPC = 2, NAPC
      IF (NSTEP .LT. IPC(5,IAPC)  .OR.  IPC(6,IAPC) .GT. 0) GO TO 200
C
C GET NUMBER OF TIMES
C IF THE RUN FAILED, THE NUMBER WRITTEN MAY BE SMALLER,
C    AND NT IS RESET WHEN THE READ FAILS.
C
      NT = (NSTEP - IPC(5,IAPC)) / ABS(IPC(6,IAPC))   + 1
      NT = MIN(NT,NTMAX)
C
C NT IS WRONG IF IPC(7,IAPC) IS NOT NEGATIVE
C
      IF (IPC(7,IAPC) .GE. 0) NT = NTMAX
C
C GET RANGES FOR I AND K
C
      I1 = ABS(IPC(7,IAPC))
      I2 = IPC(8,IAPC)
      I3 = IPC(9,IAPC)
      NI = 1 + (I2-I1)/I3
C
      K1 = IPC(10,IAPC)
      K2 = IPC(11,IAPC)
      K3 = IPC(12,IAPC)
      NK = 1 + (K2-K1)/K3
      NKS = NK
C
C GET STREAM NUMBER
C OPEN OLD FILE
C
      IS = 40 + IAPC
      CALL SQOPN(IS,2)
C
C READ HEADER
C
      READ (IS) DESCR
      READ (IS) LTITLE(IAPC)
     1                ,IB,IE,NI
     1                ,KB,KE,NKI
C
C
      DO 40 KT = 1, NT
```

```
590   C
600           READ (IS,ERR=41) ISTEP,TTIMR
610   30      FORMAT('ISTEP =',I8,F14.7)
620           TAR(KT) = TTIMR
630           READ (IS) ((VAR(I,K,KT),I=I1,I2,I3),K=1,NKS)
640   C
650           DO 60 I = I1, I2, I3
660           DO 60 K = 1, NK
670           VMAX = MAX(VMAX,VAR(I,K,KT))
680           VMIN = MIN(VMIN,VAR(I,K,KT))
690   60      CONTINUE
700   C
710   40      CONTINUE
720           GO TO 42
730   41      NT = KT-1
740   42      CONTINUE
750   C
760   C CLOSE FILE AT END.
770   C
780           CLOSE (UNIT=IS)
790   C
800   C
810   C
820   C NO PLOT IF MAX = MIN
830   C
840           IF (VMAX .EQ. VMIN) GO TO 200
850   C
860   C GET ROUNDED V PLOT RANGE
870   C
880           VINC = ROUND((VMAX-VMIN)/4.3)
890           I = VMAX/VINC + 9999
900           IMAX = I - 9998
910           I = VMIN/VINC + 9999
920           IMIN = I - 9999
930           VPMAX = VINC * IMAX
940           VPMIN = VINC * IMIN
950   C
960   C INITIALIZE TERMINAL
970   C
980           CALL INTERM
990   C
1000  C GET TERMINAL WINDOW AND RATIO
1010  C
1020          CALL SEETW (MINX,MAXX,MINY,MAXY)
1030          MB = MAXX / 40
1040          MINX = MINX + MB * 3
1050          MINY = MINY + MB * 3
1060          MAXX = MAXX - MB
1070          MAXY = MAXY - MB
1080  C
1090          XR = MAXX - MINX
1100          YR = MAXY - MINY
1110  C
1120  C GET NUMBER OF LINES OF TEXT AND RASTER EQUIVALENT
1130  C GET MAXIMUM NUMBER OF CHARACTERS AND RASTER EQUIVALENT
1140  C
1150          NLT = 7
1160          IF (NY .NE. 1) NLT = NLT + 1
```

```
1170            RNLT = NLT * YR / NLPP
1180    C
1190            MNC = MAX(24,LENGTH(DESCR,44) ) + 6
1200            RMNC = MNC * XR / NCPL
1210    C
1220    C PLACE TEXT AND TERMINAL WINDOW FOR BOTTOM
1230    C
1240            NBL = NLPP - NLT - NLPA
1250            NBC = 1 + (NCPL - MNC) / 2
1260            MINY = MINY + RNLT
1270    C
1280    C TEXT ON PLOT
1290    C
1300            DO 120 IBL = 1, NBL
1310    120     WRITE (IUPLOT,130)
1320    130     FORMAT (1X)
1330            WRITE (IUPLOT,140)BLANK(1:NBC),LTITLE(IAPC)
1340            WRITE (IUPLOT,140)BLANK(1:NBC),DESCR
1350    140     FORMAT (A,A)
1360            WRITE (IUPLOT,150)BLANK(1:NBC),VMAX
1370            WRITE (IUPLOT,160)BLANK(1:NBC),VMIN
1380            IF (NT .GT. 1) WRITE (IUPLOT,180)BLANK(1:NBC),TAR(1),TAR(NT)
1390            IF (NT .EQ. 1  .AND.  TAR(1) .NE. 0)
1400            1       WRITE (IUPLOT,181)BLANK(1:NBC),TAR(1)
1410            WRITE (IUPLOT,195)BLANK(1:NBC),VINC*IMIN,VINC*IMAX,VINC
1420    150     FORMAT (A,'MAXIMUM    =',1PG13.5)
1430    160     FORMAT (A,'MINIMUM    =',1PG13.5)
1440    180     FORMAT (A,'TIME RANGE',F7.0,' TO',F9.0)
1450    181     FORMAT (A,'TIME ',F8.0)
1460    195     FORMAT (A,'V RANGE',1P2E10.2,'  BY',E10.2)
1470    C
1480    C SET PLOT WINDOWS
1490    C
1500            CALL TWINDO (MINX,MAXX,MINY,MAXY)
1510    C
1520            DR = .02*(RSAM-RINS)
1530            CALL DWINDO(RINS-DR,RSAM+DR
1540            1       ,VINC*(-.1+IMIN),VINC*(+.1+IMAX))
1550    C
1560    C DRAW AXES
1570    C
1580            CALL MOVEA(RINS,VPMAX)
1590            CALL DRAWA(RINS,VPMIN)
1600            CALL DRAWA(RSAM,VPMIN)
1610    C
1620    C DRAW V MARKERS
1630    C
1640            DO 90 I = IMIN, IMAX
1650            V = I * VINC
1660            CALL MOVEA (RINS,V)
1670    90      CALL DRWREL (-15,0)
1680    C
1690    C DRAW PLOTS
1700    C
1710            DO 70 KT = 1, NT
1720            TTIMR = TAR(KT)
1730            DO 65 I = I1, I2, I3
1740    65      RAR(I) = RH(I)
```

```
1750    C
1760    C CLEAN UP ENDS
1770    C
1780            IF (I1 .EQ. 1) THEN
1790                    RAR(1) = RINS
1800                    IF (I3 .EQ. 1) THEN
1810                            DO 66 K = 1, NK
1820    66                      VAR(1,K,KT) = .5*(VAR(1,K,KT)+VAR(2,K,KT))
1830                    ENDIF
1840            ENDIF
1850    C
1860            IF (I2 .EQ. NRP2) THEN
1870                    RAR(NRP2) = RSAM
1880                    IF (I3 .EQ. 1) THEN
1890                            DO 68 K = 1, NK
1900    68                      VAR(NRP2,K,KT) = .5*(VAR(NRP2,K,KT)+VAR(NRP1,K,K
1910                    ENDIF
1920            ENDIF
1930    C
1940            DO 70 K = 1, NK
1950            CALL MOVEA(RAR(I1),VAR(I1,K,KT))
1960            DO 70 I = I1+I3 , I2, I3
1970            CALL DRAWA(RAR(I),VAR(I,K,KT))
1980    70      CONTINUE
1990    C
2000            CALL HDCOPY
2010    C
2020    200     CONTINUE
2030    C
2040            CALL CLTERM
2050    C
2060            RETURN
2070            END
```

```
      SUBROUTINE MDIAG
C
C MESH DIAGNOSTIC LINE EVERY NDIAG STEPS.
C
      INCLUDE 'COM./NOLIST'
C
      IF (LCONT .OR. .NOT. DIAG) RETURN
C
C WRITE HEADING
C
      IF (ISTEP .EQ. 0) WRITE (30,10)
10    FORMAT (1H1,T40,'INTERFACE DIAGNOSTICS'//
     1        ' ISTEP    FMAX(IMAX)    FMIN(IMIN)  '
     1        ,'     DFMAX(IMAX)     DFMIN(IMIN)'
     1        ,'    TINTMAX(IMAX)  TINTMIN(IMIN)'
     1        ,'   DTINTMAX(IMAX) DTINTMIN(IMIN)'/)
C
C GET DIAGNOSTICS
C MXMN LOOKS ONLY AT THE INTERIOR OF A 2D ARRAY (ASSUMING THE BOUNDARY
C IS DETERMINED BY BOUNDARY CONDITIONS).  HENCE THESE WEIRD ARGUMENTS.
C
      CALL MXMN( FINTH(1-NRP2),NRP2,3
     1        , FMAX,JMAX,MAXD, FMIN,JMIN,MIND,3,2)
      CALL MXMN(DFINTH(1-NRP2),NRP2,3
     1        ,DFMAX,IMAX,MAXD,DFMIN,IMIN,MIND,3,2)
      CALL MXMN( TINTH(1-NRP2),NRP2,3
     1        , TMAX,KMAX,MAXD, TMIN,KMIN,MIND,3,2)
      CALL MXMN(DTINTH(1-NRP2),NRP2,3
     1        ,DTMAX,LMAX,MAXD,DTMIN,LMIN,MIND,3,2)
C
C WRITE DIAGNOSTICS
C
      WRITE (30,30) ISTEP,
     1                FMAX,JMAX, FMIN,JMIN
     1               ,DFMAX,IMAX,DFMIN,IMIN
     1               , TMAX,KMAX, TMIN,KMIN
     1               ,DTMAX,LMAX,DTMIN,LMIN
30    FORMAT (1X,I4,1X,2(F9.5,'(',I3,')'),
     1              2X,2(F11.7,'(',I3,')') ,
     1              2X,2(F10.4,'(',I3,')') ,
     1              2X,2(F10.6,'(',I3,')') )
C
C CLEAN TERMINATION ON INSTABILITY
C
       FRNGE = 0.9 * (RSAM - RINS)
      DFRNGE = .07 * (RSAM - RINS) * (1. + NSTEP/(1.+ISTEP) )
C
      IF (  FMAX -  FMIN .GT.  FRNGE ) CALL CEXT (CGROW,40,2,'FR')
      IF ( DFMAX - DFMIN .GT. DFRNGE ) CALL CEXT (CGROW,40,2,'DF')
C
      RETURN
      END
```

```
      SUBROUTINE TDIAG
C
C T DIAGNOSTIC LINE EVERY NDIAG STEPS.
C
      INCLUDE 'COM./NOLIST'
C
      IF (LCONT .OR. .NOT. DIAG) RETURN
C
C WRITE HEADING
C
      IF (ISTEP .EQ. 0) WRITE (*,10)
      IF (ISTEP .EQ. 0) WRITE (31,10)
10    FORMAT (1H1,T40,'T DIAGNOSTICS'//
     1        ' ISTEP  TMAX(IMAX,KMAX) TMIN(IMIN,KMIN)  '
     1        ,'  DTMAX(IMAX,KMAX)  DTMIN(IMIN,KMIN)'
     1        ,'     FBOT        FTOP       FRITE       FINT       RSUM
C
C GET DIAGNOSTICS
C
      RSUM = ( FRITE + FBOT + FTOP + FINT )
     1       / (1.E-20 + ABS(FRITE) + ABS(FBOT) + ABS(FTOP) + ABS(FIN
C
      CALL MXMN(T,NRCP2,NZCP2
     1       ,TMAX,JMAX,LMAX,TMIN,JMIN,LMIN,3,3)
      CALL MXMN(DT,NRCP2,NZCP2
     1       ,DTMAX,IMAX,KMAX,DTMIN,IMIN,KMIN,3,3)
C
C WRITE DIAGNOSTICS
C
      WRITE (*,30) ISTEP,
     1               TMAX,JMAX,LMAX,TMIN,JMIN,LMIN
     1               ,DTMAX,IMAX,KMAX,DTMIN,IMIN,KMIN
     1               ,FBOT,FTOP,FRITE,FINT,RSUM
      WRITE (31,30) ISTEP,
     1               TMAX,JMAX,LMAX,TMIN,JMIN,LMIN
     1               ,DTMAX,IMAX,KMAX,DTMIN,IMIN,KMIN
     1               ,FBOT,FTOP,FRITE,FINT,RSUM
30    FORMAT (1X,I4,2X,2(F7.2,'(',I3,',',I3,')'),
     1               2X,2(F9.4,'(',I3,',',I3,')')
     1               ,1P4E12.3,0PF8.4)
C
C CLEAN TERMINATION ON INSTABILITY
C
       TRNGE = (TTOP - TBOT) * 1.5
      DTRNGE = (TTOP - TBOT) * 0.3 * (1. + NSTEP/(1.+ISTEP) )
C
      IF (  TMAX -  TMIN .GT.  TRNGE ) CALL CEXT (CGROW,40,2,'TR')
      IF ( DTMAX - DTMIN .GT. DTRNGE ) CALL CEXT (CGROW,40,2,'DT')
C
      RETURN
      END
```

```
  10            SUBROUTINE CDIAG
  20    C
  30    C C DIAGNOSTIC LINE EVERY NDIAG STEPS.
  40    C
  50            INCLUDE 'COM./NOLIST'
  60    C
  70            IF (LCONT .OR. .NOT. DIAG) RETURN
  80    C
  90    C WRITE HEADING
 100    C
 110            IF (ISTEP .EQ. 0) WRITE (32,10)
 120    10      FORMAT (1H1,T40,'C DIAGNOSTICS'//
 130           1       ' ISTEP     CMAX(IMAX,KMAX)    CMIN(IMIN,KMIN)  '
 140           1       ,'    RDCMAX(IMAX,KMAX)    RDCMIN(IMIN,KMIN)'
 150           1       ,'       XINT       XTOP       RSUM'/)
 160    C
 170    C GET DIAGNOSTICS
 180    C
 190            RSUM = ( XINT + XTOP )
 200           1       / (1.E-20 + ABS(XINT) + ABS(XTOP) )
 210    C
 220            CALL MXMN(C,NRP2,NZP2
 230           1        ,CMAX,JMAX,LMAX,CMIN,JMIN,LMIN,3,3)
 240            CALL MXMN(DC,NRP2,NZP2
 250           1        ,DCMAX,IMAX,KMAX,DCMIN,IMIN,KMIN,3,3)
 260    C
 270    C GET RELATIVE CHANGE EXTREMA
 280    C
 290            CDIF = CMAX - CMIN
 300            IF (CDIF .EQ. 0.) CDIF = 1
 310            DCMAX = DCMAX / CDIF
 320            DCMIN = DCMIN / CDIF
 330    C
 340    C WRITE DIAGNOSTICS
 350    C
 360            WRITE (32,30) ISTEP,
 370           1               CMAX,JMAX,LMAX,CMIN,JMIN,LMIN
 380           1               ,DCMAX,IMAX,KMAX,DCMIN,IMIN,KMIN
 390           1               ,XINT,XTOP,RSUM
 400    30      FORMAT (1X,I4,2X,2(F10.5,'(',I3,',',I3,')'),
 410           1               2X,2(F12.7,'(',I3,',',I3,')')
 420           1               ,1P2E12.3,0PF8.4)
 430    C
 440    C CLEAN TERMINATION ON INSTABILITY
 450    C
 460             CRNGE = 3*MAX(CTOP,CMELT)
 470            DCRNGE = CRNGE*.3 * (1. + NSTEP/(1.+ISTEP) )
 480    C
 490            IF (  CMAX -  CMIN .GT.  CRNGE ) CALL CEXT (CGROW,40,2,'CR')
 500            IF ( DCMAX - DCMIN .GT. DCRNGE ) CALL CEXT (CGROW,40,2,'DC')
 510    C
 520            RETURN
65538           END
```

```
      SUBROUTINE PDIAG
C
C PSI DIAGNOSTIC LINE EVERY NDIAG STEPS.
C
      INCLUDE 'COM./NOLIST'
C
      IF (LCONT .OR. .NOT. DIAG) RETURN
C
C WRITE HEADING
C
      IF (ISTEP .EQ. 0) WRITE (33,10)
10    FORMAT (1H1,T40,'PSI DIAGNOSTICS'//
     1       ' ISTEP   PSIMAX(IMAX,KMAX)  PSIMIN(IMIN,KMIN)  '
     1       ,'   DPSIMAX(IMAX,KMAX)   DPSIMIN(IMIN,KMIN)'/)
C
C GET DIAGNOSTICS
C
      CALL MXMN(PSI,NRP2,NZP2
     1       ,PMAX,JMAX,LMAX,PMIN,JMIN,LMIN,3,3)
      CALL MXMN(DPSI,NRP2,NZP2
     1       ,DPMAX,IMAX,KMAX,DPMIN,IMIN,KMIN,3,3)
C
C WRITE DIAGNOSTICS
C
      WRITE (33,30) ISTEP,
     1               PMAX,JMAX,LMAX,PMIN,JMIN,LMIN
     1               ,DPMAX,IMAX,KMAX,DPMIN,IMIN,KMIN
30    FORMAT (1X,I4,2X,2(F10.5,'(',I3,',',I3,')'),
     1               2X,2(F12.7,'(',I3,',',I3,')') )
C
C CLEAN TERMINATION ON INSTABILITY
C
       PRNGE = 9999
      DPRNGE = 9999
C
      IF (  PMAX -  PMIN .GT.  PRNGE ) CALL CEXT (CGROW,40,2,'PR')
      IF ( DPMAX - DPMIN .GT. DPRNGE ) CALL CEXT (CGROW,40,2,'DP')
C
      RETURN
      END
```

```
 10           SUBROUTINE VDIAG
 20    C
 30    C VORT DIAGNOSTIC LINE EVERY NDIAG STEPS.
 40    C
 50           INCLUDE 'COM./NOLIST'
 60    C
 70           IF (LCONT .OR. .NOT. DIAG) RETURN
 80    C
 90    C WRITE HEADING
100    C
110           IF (ISTEP .EQ. 0) WRITE (34,10)
120    10     FORMAT (1H1,T40,'VORTICITY DIAGNOSTICS'//
130           1       ' ISTEP  VORTMAX(IMAX,KMAX) VORTMIN(IMIN,KMIN)  '
140           1       ,'  DVORTMAX(IMAX,KMAX)  DVORTMIN(IMIN,KMIN)'/)
150    C
160    C GET DIAGNOSTICS
170    C
180           CALL MXMN(VORT,NRP1,NZP1
190           1        ,VMAX,JMAX,LMAX,VMIN,JMIN,LMIN,3,3)
200           CALL MXMN(DVORT,NRP1,NZP1
210           1        ,DVMAX,IMAX,KMAX,DVMIN,IMIN,KMIN,3,3)
220    C
230    C WRITE DIAGNOSTICS
240    C
250           WRITE (34,30) ISTEP,
260           1               VMAX,JMAX,LMAX,VMIN,JMIN,LMIN
270           1               ,DVMAX,IMAX,KMAX,DVMIN,IMIN,KMIN
280    30     FORMAT (1X,I4,2X,2(F10.5,'(',I3,',',I3,')'),
290           1               2X,2(F12.7,'(',I3,',',I3,')') )
300    C
310    C CLEAN TERMINATION ON INSTABILITY
320    C
330            VRNGE = 9999
340           DVRNGE = 9999
350    C
360           IF (  VMAX -  VMIN .GT.  VRNGE ) CALL CEXT (CGROW,40,2,'VR')
370           IF ( DVMAX - DVMIN .GT. DVRNGE ) CALL CEXT (CGROW,40,2,'DV')
380    C
390           RETURN
400           END
```

```
      SUBROUTINE PRNT (VAR,ND,IAPC)
C
C PRINTER ARRAY OUTPUT
C
      INCLUDE 'COM./NOLIST'
C
      DIMENSION VAR(ND,1)
C
      DIMENSION RIN(NRCP2)
      DIMENSION ZIN(NZCP2)
      CHARACTER *8 CSHOWZERO
      CHARACTER *6 TIMECC
      DATA TIMECC /'TIME  '/
C
C GET PLOT DOMAIN AND I AND K RANGES FROM IAPC
C
      RPLL = RINS
      RPLR = RAMP
      ZPLB = ZBPL
      ZPLT = ZTPL
C
      IF (IAPC .LE. 5) THEN
            NCZ = NZCP2
            NINTP = 2
            ZCPB = ZPLB
            RCPR = RAMP
      ELSE IF (IAPC .LE. 11) THEN
            ZPLB = 0
            ZCPB = 0
            ZPLT = 0
            NCZ  = 1
      ELSE IF (IAPC .GE. 14  .AND.  IAPC .LE. 17) THEN
            NCZ  = NZP1
            NINTP = 4
            ZCPB = ZINT
            RCPR = RSAM
      ELSE
            NCZ  = NZP2
            NINTP = 4
            ZCPB = ZINT
            RCPR = RSAM
      ENDIF
C
C CONTOL TO INTERPOLATE HH POINTS TO WW FOR PLOTTING
C
      IF (IAPC .GE. 14  .AND.  IAPC .LE. 17) THEN
            NHHI = 0
      ELSE
            NHHI = 1
      ENDIF
C
C NUMBER OF CONTOURS
C
      NCONT = 10
      IF (IAPC .EQ. 2) NCONT = 20
C
C PLOTTER EXITS AND MESH CALL RETURN
C
```

```
590             IF (LCONT) THEN
600                     CALL CPR (VAR,ND,NCZ,IAPC,NCONT
610             1                       ,NHHI,RPLL,RPLR,ZPLB,ZPLT)
620                     RETURN
630             ELSE
640                     IF (IAPC .EQ. 1) RETURN
650             ENDIF
660     C
670     C NO CONTOUR PLOTS FOR ONE-DIMENSIONAL ARRAYS
680     C
690             IF (NCZ .EQ. 1) THEN
700     C               IF (LCONT) CALL PLOT1(VAR,ND,IAPC)
710                     GO TO 40
720             ENDIF
730     C
740     C IS ZERO CONTOUR REQUIRED?
750     C
760             CSHOWZERO = 'SHOWZERO'
770             IF (IAPC .EQ. 3 ) CSHOWZERO = 'NOZERO'
780             IF (IAPC .EQ. 13) CSHOWZERO = 'NOZERO'
790             IF (IAPC .EQ. 15) CSHOWZERO = 'NOZERO'
800             IF (IAPC .EQ. 17) CSHOWZERO = 'NOZERO'
810     C
820     C
830     C                       CONTOUR AGAINST PHYSICAL VARIABLES.
840     C
850             J = ISTEP - IPC(1,IAPC)
860             K = IPC(2,IAPC)
870             IF ( ((J/K) * K .NE. J) .OR. (J .LT. 0) ) GO TO 20
880     C
890     C CALL PLOTTER
900     C
910             IF (IAPC .GE. 14  .AND.  IAPC .LE. 17) THEN
920                     CALL CONTR (VAR,RW,ZETW,ND,NCZ
930             1                       ,RPLL,RCPR,ZCPB,ZPLT
940             1                       ,1,NCLP,1.,7
950             1                       ,ACONT(IAPC),NCONT,1
960             1                       ,'NOSHADOW',CSHOWZERO,'OVERPRINT',NINTP
970             1                       ,LTITLE(IAPC),42,TIMECC,TTIME)
980             ELSE
990                     CALL CONTR (VAR,RH,ZETH,ND,NCZ
1000            1                       ,RPLL,RCPR,ZCPB,ZPLT
1010            1                       ,1,NCLP,1.,7
1020            1                       ,ACONT(IAPC),NCONT,1
1030            1                       ,'NOSHADOW',CSHOWZERO,'OVERPRINT',NINTP
1040            1                       ,LTITLE(IAPC),42,TIMECC,TTIME)
1050            ENDIF
1060    20      CONTINUE
1070    C
1080    C                       CONTOUR AGAINST INTEGERS
1090    C
1100            J = ISTEP - IPC(3,IAPC)
1110            K = IPC(4,IAPC)
1120            IF ( ((J/K) * K .NE. J) .OR. (J .LT. 0) ) GO TO 40
1130    C
1140            DO 30 I = 1, ND
1150    30      RIN(I) = I
1160            DO 31 I = 1, NCZ
```

```
1170   31       ZIN(I) = - I
1180   C
1190            XT = ND+.5
1200            YT = NCZ+.5
1210   C
1220            NLINC = NCLI
1230            IF (NLINC .EQ. 0) NLINC = NCZ
1240   C
1250            CALL CONTR (VAR,RIN,ZIN,ND,NCZ
1260            1                  ,.5,XT,-YT,-.5
1270            1                  ,1,NLINC,1.,-4
1280            1                  ,ACONT(IAPC),NCONT,1
1290            1                  ,'NOSHADOW',CSHOWZERO,'OVERPRINT',NINTP
1300            1                  ,LTITLE(IAPC),42,TIMECC,TTIME)
1310   C
1320   40       CONTINUE
1330   C
1340   C PRINT NUMBERS OR WRITE ARRAYS FOR MOVIE OR 1D PLOTS.
1350   C CORRECT IPC8 AND 11 IF NECESSARY
1360   C
1370            ISTB = IPC(5,IAPC)
1380            J = ISTEP - ISTB
1390            ISTI = IPC(6,IAPC)
1400            IF ( ((J/ISTI) * ISTI .NE. J) .OR. (J .LT. 0) ) GO TO 100
1410   C
1420            I1 = IPC(7,IAPC)
1430            I2 = MIN(IPC(8,IAPC),ND)
1440            I3 = IPC(9,IAPC)
1450            K1 = IPC(10,IAPC)
1460            K2 = MIN(IPC(11,IAPC),NCZ)
1470            K3 = IPC(12,IAPC)
1480   C
1490   C MOVIE JUMP CONDITION
1500   C
1510            IF (ISTI .LT. 0) GO TO 70
1520   C
1530   C HEADING
1540   C
1550            ICH = 1
1560            IF (((K2+K3-K1)/K3) * ((I2+I3-I1)/I3) .LT. 24) ICH = 0
1570            WRITE (6,45) ICH,LTITLE(IAPC),ISTEP,(I,I = I1,I2,I3)
1580   45       FORMAT (I1/T18,A,' AT ISTEP =',I4// (3X,12I10) )
1590   C
1600   C VALUES
1610   C
1620            DO 50 K = K1, K2, K3
1630   50       WRITE (6,60) K,(VAR(I,K),I = I1,I2,I3)
1640   60       FORMAT (/1X,I3,3X,1P12E10.3/ (7X,1P12E10.3) )
1650   C
1660            GO TO 100
1670   C
1680   C WRITE ARRAY FOR MOVIE
1690   C
1700   70       CONTINUE
1710            CALL MOVIE (VAR,ND,IAPC,I1,I2,I3,K1,K2,K3,ISTB,ISTI,J)
1720   C
1730   100      CONTINUE
1740   C
```

```
 1750            RETURN
65588            END
```

```
   10            SUBROUTINE PRNTK (K,VAR,IAPC)
   20    C
   30    C PRINTER OUTPUT, LINE BY LINE.
   40    C
   50            INCLUDE 'COM./NOLIST'
   60    C
   70            DIMENSION VAR(1)
   80    C
   90    C PRINT
  100    C
  110            J = ISTEP - IPC(5,IAPC)
  120            L = IPC(6,IAPC)
  130            IF ( ((J/L) * L .NE. J) .OR. (J .LT. 0) ) GO TO 100
  140    C
  150            I1 = IPC(7,IAPC)
  160            I2 = IPC(8,IAPC)
  170            I3 = IPC(9,IAPC)
  180            K1 = IPC(10,IAPC)
  190            K2 = IPC(11,IAPC)
  200            K3 = IPC(12,IAPC)
  210    C
  220    C VALUES
  230    C
  240    48      KK = K - K1
  250            IF (( (K-K1)*(K2-K) .GE. 0 ) .AND. (KK/K3*K3 .EQ. KK) ) THEN
  260    50              WRITE (6,60) LTITLE(IAPC),K,(VAR(I),I = I1,I2,I3)
  270    60              FORMAT (/1X,A8,2X,I3,3X,1P12E10.3/ (13X,1P12E10.3) )
  280            ENDIF
  290    C
  300    100     CONTINUE
  310    C
  320            RETURN
65538            END
```

```
   10            FUNCTION PRTEST (IAPC)
   20    C
   30    C TEST IF FUNCTION IS REQUIRED
   40    C
   50            INCLUDE 'COM./NOLIST'
   60    C
   70            PRTEST = .FALSE.
   80    C
   90    C GET APPROPRIATE LIMITS FOR REGULAR OR PLOT CASE
  100    C
  110            IF (LCONT) THEN
  120                    LB = 13
  130                    LE = 15
  140                    I = IDAIO
  150            ELSE
  160                    LB = 1
  170                    LE = 5
  180                    I = ISTEP
  190            ENDIF
  200    C
  210            DO 10 L = LB, LE, 2
  220            J = I - IPC(L,IAPC)
  230            K = IPC(L+1,IAPC)
  240            IF ((J/K)*K .EQ. J  .AND.  J .GE. 0) PRTEST = .TRUE.
  250    10      CONTINUE
  260    C
  270            RETURN
65288          END
```

```
      SUBROUTINE MOVIE (VAR,NX,IAPC,I1,I2,I3,K1,K2,K3,ISTB,ISTI,ISTD)
C
C WRITES FILE FOR LATER LINEPL OR MOVIE INPUT.
C
      INCLUDE 'COM./NOLIST'
C
      DIMENSION VAR(NX,1)
C
C OUTPUT STREAM
C
      IS = 40 + IAPC
C
C IF ISTD IS ZERO AND I1 IS NEGATIVE OR MORE THAN 999
C OPEN NEW FILE AND WRITE HEADER
C
      IF (ISTD .NE. 0) GO TO 90
      IF (I1 .GT. 999) GO TO 89
      IF (I1 .GT. 0  ) GO TO 90
C
C CORRECT SIGN OF I1 = IPC(7,IAPC)
C
      I1 = - I1
      IPC(7,IAPC) = I1
89    CONTINUE
C
C I RANGE FOR DATA
C
      IB = 1
      IE = IB + (I2-I1)/I3
      NI = IE
C
C K RANGE FOR DATA
C
      KB = 1
      KE = KB + (K2-K1)/K3
      NK = KE
C
C OPEN NEW FILE
C
      CALL SQOPN(IS,3)
C
C WRITE HEADER
C
      WRITE (IS) DESCR
      WRITE (IS) LTITLE(IAPC)
     1                ,IB,IE,NI
     1                ,KB,KE,NK
     1                ,ZL,ZR,YB,YT
C
      GO TO 100
C
C OPEN EXISTING FILE FOR WRITING AT THE END
C
90    CALL SQOPN(IS,2)
      DO 91 I=1,9999
91    READ (IS,END=92)
92    CONTINUE
      IF (IMACH .EQ. 2) BACKSPACE IS
```

```
590   C
600   C WRITE ISTEP, TIME AND DATA
610   C
620   100      CONTINUE
630   C
640            WRITE (IS) ISTEP,TTIME
650            WRITE (IS) ((VAR(I,K),I=I1,I2,I3),K=K1,K2,K3)
660   C
670   C CLOSE FILE
680   C
690            CLOSE (UNIT=IS)
700   C
710            RETURN
720            END
```

```
      SUBROUTINE INIT0
C
C OPEN RESULT FILES, FOR EACH CASE.
C READ INPUT AND WRITE HEADINGS, FOR EACH CASE. STOP AT END.
C
      INCLUDE 'COM./NOLIST'
C
C                          READ INPUT
C
      IOSTAT = 0
C
C THIS READ SKIPS THE SOS PAGE MARKER BETWEEN PAGES OF INPUT
C
      IF ( ICASE .GT. 1 ) READ (5,230,END=201,IOSTAT=IOSTAT)
230   FORMAT ()
231   FORMAT (5X,A///
     1        T7,7X,F10.6,T27,8X,F10.6
     1        ,T47,7X,F10.6,T67,8X,F10.6
     1        ,T87,8X,F13.6,T110,8X,F10.6/
     1        T7,7X,F10.6,T27,8X,F10.6
     1        ,T47,7X,F10.6,T67,8X,F10.6
     1        ,T87,8X,F13.6,T110,8X,F10.6/
     1        T7,7X,F10.6,T27,8X,F10.6
     1        ,T47,7X,F10.6,T67,8X,F10.6
     1        ,T87,8X,F13.6,T110,8X,F10.6/
     1                ,T27,8X,F10.6
     1        ,T47,7X,F10.6,T67,8X,F10.6
     1        ,T87,8X,F13.6,T110,8X,F10.6/
     1        ,T87,8X,F13.6/)
232   FORMAT (///
     1        (T3,A12,T18,A20,T38,I4,T42,0P,F12.6,1P,7E10.2))
233   FORMAT (//
     1        T2,7X,I5,T20,6X,L2,T36,5X,L2/
     1        T2,7X,F8.3,T20,6X,F7.3/
     1        T2,7X,F8.3,T20,6X,F7.3,T36,5X,F6.3
     1        ,T51,8X,F6.3,T71,7X,L2,T84,7X,L2
     1        ,T97,7X,F6.3,T114,7X,F6.3/
     1        T2,7X,F8.3,T20,6X,F7.3,T36,5X,F6.3
     1        ,T51,8X,F6.3,T71,7X,L2,T84,7X,L2
     1        ,T97,7X,F6.3,T114,7X,F6.3/
     1        T2,7X,F8.3,T20,6X,F7.3,T36,5X,F6.3
     1        ,T51,8X,F6.3,T71,7X,L2,T84,7X,L2
     1        ,T97,7X,F6.3,T114,7X,F6.3/
     1        T2,7X,F8.3,T20,6X,F7.3,T36,5X,F6.3
     1        ,T51,8X,I6,T71,7X,I3,T84,7X,I3
     1        ,T97,7X,F6.3,T114,7X,F6.3)
235   FORMAT (////
     1        T4,7X,I5,T18,7X,I5
     1        ,T50,8X,I4,T66,8X,I5
     1                ,T90,7X,I3,T105,6X,I3,8X,I3/
     1        ,T66,8X,F4.1,T80,6X,F6.2,T93,6X,F6.2
     1        ,T118,I3///
     1        (2X,A8,I2,I8,I6,2(I10,I6),2X,2(3X,3I4),2(I8,I6),F6.1 ) )
C
C CIN REPLACES THESE LINES PR1, etc., BY THE FILES:
C PROB1, PROB2, METH1, METH2, AND OUTP.BC
C WHICH CONTAIN CONTINUATION LINES FOR I/O STATEMENTS IN INIT0
C AND FOR COMMON BLOCKS IN COMO TO MAKE THE INCLUDED FILE COM.
```

```
590   C
600           READ(5,231,END=200,ERR=140,IOSTAT=IOSTAT) DESCR,
610   *PR1
620           READ(5,232,END=140,ERR=140,IOSTAT=IOSTAT)
630           1       (CMATPR(IMAT),CUNITS(IMAT),MATPRT(IMAT),
640           1       (PRPMAT(K,IMAT),K=1,NPRCON)
650           1       ,IMAT=1,NMATPR)
660   *PR2
670           READ(5,233,END=140,ERR=140,IOSTAT=IOSTAT)
680   *ME1
690   *ME2
700           READ(5,235,END=140,ERR=140,IOSTAT=IOSTAT)
710   *OUT
720           1       ,( STITLE(I), IDUM, (IPC(J,I), J = 1,16), ACONT(I), I =
730   C
740   C PRINT PREVIOUS RESULT FILE, UNDER NCOPY CONTROL.
750   C KEEP RESULT FILE, FOR CONTOUR RUN.
760   C LCONT IS .TRUE. FOR CONTOUR.
770   C
780   140     CONTINUE
790           IF (ICASE .GT. 1) THEN
800                   IF ( .NOT. LCONT ) THEN
810                           IF (NCOPY .GE. 2) THEN
820                                   CLOSE (6,STATUS='PRINT')
830                           ELSE IF (NCOPY .EQ. 1) THEN
840                                   CLOSE (6,STATUS='PRINT/DELETE')
850                           ELSE
860                                   CLOSE (6,STATUS='KEEP')
870                           ENDIF
880                   ELSE
890                           CLOSE (6,STATUS='KEEP')
900                   ENDIF
910           ENDIF
920   C
930   C OPEN NEW RESULT FILE.
940   C
950           IF (LCONT .AND. IMACH .EQ. 2) THEN
960                   OPEN (6,FILE='NUL',STATUS='NEW')
970           ELSE
980                   OPEN (6,FILE='PRN'
990           1       ,STATUS= 'NEW',RECL=160,SHARED,FORM='FORMATTED')
1000          ENDIF
1010  C
1020  C WRITE HEADING PAGE, EVEN IF THERE WAS AN ERROR.
1030  C
1040  C GET TEXT FOR DATE AND TIME
1050  C
1060          CALL DATTIM (JTIME)
1070  C
1080  C GET NTYPE
1090  C
1100          TEXPRG(1) = 'TEMPERATURE SOLUTION'
1110          TEXPRG(2) = 'TEMPERATURE AND CONCENTRATION'
1120          TEXPRG(3) = 'TEMPERATURE AND FLOW'
1130          TEXPRG(4) = 'TEMPERATURE, CONCENTRATION AND FLOW'
1140  C
1150          NTYPE = 1
1160          IF (LU) NTYPE = 3
```

```
1170            IF (LC) NTYPE = NTYPE + 1
1180    C
1190    C WRITE HEADER FOR SUMMARY PAGE
1200    C
1210            IF (LCONT .AND. IMACH .EQ. 2) GO TO 160
1220            IPAGE = MIN(ICASE-1,1)
1230            WRITE (16,162) IPAGE, TEXPRG(NTYPE),ICASE,DESCR
1240            1                ,NR,NZ,NRA,NZS,JTIME
1250    162     FORMAT (I1,A40,'CASE',I3,4X,A30,4X
1260            1       ,I3,3(',',I3),4X,A)
1270    C
1280    C WRITE HEADER PAGE ON STREAM 6
1290    C
1300            WRITE (6,139) TEXPRG(NTYPE)
1310            1       , ICASE, NR,NZ,NRA,NZS, JTIME
1320    139     FORMAT ('1',A,12X,'CASE ',I3
1330            1       ,12X,'MESH',I3,3(',',I3),12X,A/)
1340    C
1350    C
1360    C CIN REPLACES THESE LINES PR1, etc., BY THE FILES:
1370    C PROB1, PROB2, METH1, METH2, AND OUTP.BC
1380    C WHICH CONTAIN CONTINUATION LINES FOR I/O STATEMENTS IN INIT0
1390    C AND FOR COMMON BLOCKS IN COMO TO MAKE THE INCLUDED FILE COM.
1400    C
1410    C
1420    141     WRITE (6,131) DESCR,
1430    *PR1
1440            WRITE (6,132)
1450            1       (CMATPR(IMAT),CUNITS(IMAT),MATPRT(IMAT),
1460            1       (PRPMAT(K,IMAT),K=1,NPRCON)
1470            1       ,IMAT=1,NMATPR)
1480    *PR2
1490            WRITE (6,133)
1500    *ME1
1510    *ME2
1520            WRITE (6,135)
1530    *OUT
1540            1       ,( STITLE(I), I, (IPC(J,I), J = 1,16), ACONT(I), I = 1,N
1550    C
1560            WRITE (6,136)
1570    131     FORMAT (5X,A,T50,'***PROBLEM PARAMETERS***'//
1580            1       T7,'R MESH'       ,T47,'Z MESH'
1590            1       ,T87,'EXTERNAL PARAMETERS'/
1600            1       T7,'RINS  =',F10.6,T27,'DRINS  =',F10.6
1610            1       ,T47,'ZT    =',F10.6,T67,'DZT    =',F10.6
1620            1       ,T87,'TTOP   =',F13.6,T110,'ZTTOP  =',F10.6/
1630            1       T7,'RSAM  =',F10.6,T27,'DRSAM  =',F10.6
1640            1       ,T47,'ZB    =',F10.6,T67,'DZB    =',F10.6
1650            1       ,T87,'TBOT   =',F13.6,T110,'ZTBOT  =',F10.6/
1660            1       T7,'RAMP  =',F10.6,T27,'DRAMPI =',F10.6
1670            1       ,T47,'ZINTI =',F10.6,T67,'DZINTM =',F10.6
1680            1       ,T87,'GTERR  =',F13.6,T110,'ZTBSC  =',F10.6/
1690            1               ,T27,'DRAMPO =',F10.6
1700            1       ,T47,'ZINTL =',F10.6,T67,'DZINTS =',F10.6
1710            1       ,T87,'WAMP   =',F13.6,T110,'CTOP   =',F10.6/
1720            1       ,T87,'DENCBM =',F13.6/)
1730     132    FORMAT (T50,'***MATERIAL PROPERTIES***'//
1740            1       T3,'PROPERTY',T18,'UNITS',T35,'CONTROL',T46,'CONSTANTS'/
```

```
1750            1       (T3,A12,T18,A20,T38,I4,T42,0P,F12.6,1P,7E10.2))
1760    133     FORMAT (/T50,'***METHOD PARAMETERS***'/
1770            1       T2,'NSTEP =',I5,T20,'LC   =',L2,T36,'LU  =',L2/
1780            1       T2,'TAUS  =',F8.3,T20,'TAUA =',F7.3
1790            1               ,T51,'ADVECTION'
1800            1               ,T71,'FIXING CONTROLS'
1810            1                       ,T97,'FIXING AMPLITUDES'/
1820            1       T2,'TAUT  =',F8.3,T20,'PRT  =',F7.3,T36,'PZT =',F6.3
1830            1       ,T51,'TUPWND =',F6.3,T71,'FIXTR =',L2,T84,'FIXTZ =',L2
1840            1       ,T97,'BETTA =',F6.3,T114,'BETTD =',F6.3/
1850            1       T2,'TAUC  =',F8.3,T20,'PRC  =',F7.3,T36,'PZC =',F6.3
1860            1       ,T51,'CUPWND =',F6.3,T71,'FIXCR =',L2,T84,'FIXCZ =',L2
1870            1       ,T97,'BETCA =',F6.3,T114,'BETCD =',F6.3/
1880            1       T2,'TAUU  =',F8.3,T20,'PRU  =',F7.3,T36,'PZU =',F6.3
1890            1       ,T51,'UUPWND =',F6.3,T71,'FIXUR =',L2,T84,'FIXUZ =',L2
1900            1       ,T97,'BETUA =',F6.3,T114,'BETUD =',F6.3/
1910            1       T2,'TAUF  =',F8.3,T20,'PRF  =',F7.3,T36,'BEF =',F6.3
1920            1       ,T51,'NUF    =',I6,T71,'NTF   =',I3,T84,'NCF   =',I3
1930            1       ,T97,'AQURTN=',F6.3,T114,'ATWLV =',F6.3)
1940    135     FORMAT (/T50,'**OUTPUT PARAMETERS***'//
1950            1       T4,'DIRECT ACCESS READ & WRITE SEGMENTS'
1960            1       ,T48,'WRITE STEP BEGINNING & INCREMENT'
1970            1               ,T85,'DIAGNOSTIC FREQUENCY   CONTOUR LINES'/
1980            1       T4,'ISEGR =',I5,T18,'ISEGW =',I5
1990            1       ,T50,'IBEGDA =',I4,T66,'IINCDA =',I5
2000            1               ,T90,'NDIAG =',I3,T105,'NCLP =',I3,'  NCLI =',I3
2010            1       T4,'(0: ANALYTIC. -1: AFTER PRIOR CASE)',
2020            1       T48,'(-1 MEANS NSTEP)'
2030            1       ,T66,'RSTRPL =',F4.1,T80,'ZBPL =',F6.2,T93,'ZTPL =',F6.2
2040            1       ,T109,'NCOPIES =',I3
2050            1       / T14,'ISTEP BEGIN & INCREMENT FOR PRINTER DIAGNOSTICS'
2060            1       ,T69,'RANGES FOR I AND K'
2070            1       ,T93,'PLOTTER IDAOUT BEGIN & INCREMENT'
2080            1       /T3,'VARIABLE/I PHYSICAL CONTOUR'
2090            1       ,T32,'INTEGER CONTOUR',T49,'PRINT NUMBERS'
2100            1               ,T66,'IB  IE  II     KB  KE  KI'
2110            1       ,T96,'PHYSICAL',T110,'INTEGER    INCR'/
2120            1       (2X,A8,I2,I8,I6,2(I10,I6),2X,2(3X,3I4),2(I8,I6),F6.1 ) )
2130    136     FORMAT (1H1)
2140    C
2150    C STOP ON ERROR OR NULL DATA
2160    C
2170            IF (IOSTAT .EQ. 0) GO TO 160
2180            WRITE (6,150) IOSTAT
2190            WRITE (*,150) IOSTAT
2200    150     FORMAT (/' INPUT ERROR',I6/)
2210            CALL STOPP ('INPUT ERROR OR NULL DATA%')
2220    C
2230    C CONTINUE AFTER SUCESSFUL READ AND WRITE
2240    C
2250    160     CONTINUE
2260    C
2270    C SPECIAL HANDLING FOR DEFAULTS
2280    C   IDAOUT   - ZERO MEANS NSTEP
2290    C   NDIAG    - ZERO MEANS UP TO 6 + 54 LINES
2300    C
2310            IDAOUT = IBEGDA
2320            IF (IDAOUT .EQ. -1) IDAOUT = NSTEP
```

```
2330            IF (NDIAG .EQ. 0) NDIAG = (NSTEP-2)/(NLPP-NLMA-6-4) + 1
2340            IF (NDIAG .LE. 0) NDIAG = 1
2350    C
2360    C DIRECT ACCESS I/O SET UP HANDLED AS ENTRY TO INIT2, FOR CONVENIENT CHA
2370    C
2380            CALL SNREC
2390    C
2400    C Z PLOT LIMITS AND DEFAULT CALCULATIONS
2410    C
2420            IF (ZBPL .EQ. 0) ZBPL = ZTBOT - RAMP
2430            ZBPL = MAX(ZBPL,ZB)
2440            IF (ZTPL .EQ. 0) ZTPL = ZTTOP + RAMP
2450            ZTPL = MIN(ZTPL,ZT)
2460    C
2470    C CORRECT THE RANGES FOR PRINTING NUMBERS.
2480    C
2490            DO 100 IAPC = 1, NAPC
2500    C
2510            IF (IAPC .LE. 5) THEN
2520                    IPC( 8,IAPC) = MIN(IPC( 8,IAPC),NRCP2)
2530                    IPC(11,IAPC) = MIN(IPC(11,IAPC),NZCP2)
2540            ELSE IF (IAPC .LE. 11) THEN
2550                    IPC( 8,IAPC) = MIN(IPC( 8,IAPC),NRP2)
2560                    IPC(11,IAPC) = MIN(IPC(11,IAPC),1)
2570            ELSE IF (IAPC .GE. 14  .AND.  IAPC .LE. 15) THEN
2580                    IPC( 8,IAPC) = MIN(IPC( 8,IAPC),NRP1)
2590                    IPC(11,IAPC) = MIN(IPC(11,IAPC),NZP1)
2600            ELSE
2610                    IPC( 8,IAPC) = MIN(IPC( 8,IAPC),NRP2)
2620                    IPC(11,IAPC) = MIN(IPC(11,IAPC),NZP2)
2630            ENDIF
2640    C
2650    100     CONTINUE
2660    C
2670    C PRODUCE MATERIAL PROPERTY DIAGNOSTICS
2680    C
2690            CALL PROPD
2700            IF (LC  .AND.  CTOP .NE. 0) CALL PROPDI
2710    C
2720            RETURN
2730    C
2740    C NO MORE DATA. USE SETUP FOR CASE 1, TO WRITE SETP.DAT
2750    C
2760    200     IF ( ICASE .EQ. 1 ) THEN
2770                    IF (IMACH .EQ. 2) THEN
2780                            OPEN(6,FILE='..\SETP.DAT',STATUS='NEW')
2790                            GO TO 141
2800                    ELSE
2810                            GO TO 140
2820                    ENDIF
2830            ENDIF
2840    C
2850    201     CALL STOPP ('END OF DATA%')
2860    C
2870            END
```

```
      SUBROUTINE SMATPR
C
C SET MATERIAL PROPERTIES
C
      INCLUDE 'COM./NOLIST'
C
C SET MATERIAL PROPERTIES AS TEMPERATURE AND CONCENTRATION POWER SERIES
C
      CALL PROPT(CP,DT,T,C,NRCP2,1,NRP1,1,NZP1,PRPMAT(1,1)
     1     ,MATPRT(1))
      CALL PROPT(CP,DT,T,C,NRCP2,1,NRP1,NZP2,NZCP2,PRPMAT(1,2)
     1     ,MATPRT(2))
      CALL PROPT(CP,DT,T,C,NRCP2,NRP2,NRCP2,1,NZCP2,PRPMAT(1,3)
     1     ,MATPRT(3))
C
      CALL PROPT(COND,DT,T,C,NRCP2,1,NRP1,1,NZP1,PRPMAT(1,4)
     1     ,MATPRT(4))
      CALL PROPT(COND,DT,T,C,NRCP2,1,NRP1,NZP2,NZCP2,PRPMAT(1,5)
     1     ,MATPRT(5))
      CALL PROPT(COND,DT,T,C,NRCP2,NRP2,NRCP2,1,NZCP2,PRPMAT(1,6)
     1     ,MATPRT(6))
C
C LATENT HEAT AT INTERFACE
C
      IF (MATPRT(7) .GT. 0) THEN
            CALL PROPIN(HEATLAT,DHLDT,PRPMAT(1,7),MATPRT(7),TINTH,NR
      ELSE
            CALL PROPIN(HEATLAT,DHLDC,PRPMAT(1,7),MATPRT(7),CINTMH,N
            DO 90 I = 1, NRP2
90          DHLDT(I) = DHLDC(I) * DCMDT(I)
      ENDIF
C
C ADDITIONAL INTERFACE PROPERTIES, FUNCTIONS OF T AND C
C
      CALL PROPT(CDINTM,DT,TINTH,CINTMH,NRP2,1,NRP1,1,1,PRPMAT(1,4)
     1     ,MATPRT(4))
C
      CALL PROPT(CPINTM,DT,TINTH,CINTMH,NRP2,1,NRP1,1,1,PRPMAT(1,1)
     1     ,MATPRT(1))
C
      CALL PROPT(CDINTC,DT,TINTH,CINTCH,NRP2,1,NRP1,1,1,PRPMAT(1,5)
     1     ,MATPRT(5))
C
      CALL PROPT(CPINTC,DT,TINTH,CINTCH,NRP2,1,NRP1,1,1,PRPMAT(1,2)
     1     ,MATPRT(2))
C
      CALL PROPT(DINTH,DT,TINTH,CINTMH,NRP2,1,NRP1,1,1,PRPMAT(1,11)
     1     ,MATPRT(11))
C
C FINAL GROUP OF MELT PROPERTIES
C
      IF (.NOT.LU)
     1     CALL PROPT(BUOY,DT,T,C,NRP2,1,NRP2,1,NZP2,PRPMAT(1,10)
     1     ,MATPRT(10))
      IF (LC) CALL PROPT(DIFF,DT,T,C,NRP2,1,NRP2,1,NZP2,PRPMAT(1,11)
     1     ,MATPRT(11))
      IF (LU) CALL PROPT(VISC,DT,T,C,NRP2,1,NRP2,1,NZP2,PRPMAT(1,12)
     1     ,MATPRT(12))
```

```
  590   C
  600           CALL PRNT(CP,NRCP2,4)
  610           CALL PRNT(COND,NRCP2,5)
  620   C
  630           IF (.NOT.LU) CALL PRNT(BUOY,NRP2,18)
  640           IF (LC) CALL PRNT(DIFF,NRP2,19)
  650           IF (LU) CALL PRNT(VISC,NRP2,20)
  660   C
  670           RETURN
  680   C
65598           END
```

```
      SUBROUTINE GTMCM(TMELTA,CMELTA,CCRYSA)
C
C GET TMELTA FROM CCRYSA, AND CMELTA FROM TMELTA, FOR INIT2 AND PROPDI.
C
      INCLUDE 'COM./NOLIST'
C
      NNEWT = 10
C
C TMELTA FROM CCRYSA
C
      IF (MATPRT(9) .GT. 0) THEN
C
C                         INITIAL APPROXIMATION
C
            TMELTA = (CCRYSA-PRPMAT(1,9))/PRPMAT(2,9)
C
C                         NEWTON ITERATION TO CORRECT TMELTA
C
            DO 1 INEWT = 1, NNEWT
            CALL PROPIN(CX,DCDT,PRPMAT(1,9),MATPRT(9),TMELTA,1)
1           TMELTA = TMELTA - (CX-CCRYSA)/DCDT
      ELSE
            CALL PROPIN(TMELTA,DTDC,PRPMAT(1,9),MATPRT(9),CCRYSA,1)
      ENDIF
C
C CMELTA FROM TMELTA
C
      IF (MATPRT(8) .GT. 0) THEN
            CALL PROPIN(CMELTA,DMDT,PRPMAT(1,8),MATPRT(8),TMELTA,1)
      ELSE
C
C                         INITIAL APPROXIMATION
C
            CMELTA = (TMELTA-PRPMAT(1,8))/PRPMAT(2,8)
C
C                         NEWTON ITERATION TO CORRECT CMELTA
C
            DO 2 INEWT = 1, NNEWT
            CALL PROPIN(TT,DTDC,PRPMAT(1,8),MATPRT(8),CMELTA,1)
2           CMELTA = CMELTA - (TT-TMELTA)/DTDC
      ENDIF
C
      RETURN
C
      END
```

```
      SUBROUTINE PROPT(PR,WK,TT,CX,ID,IB,IE,KB,KE,PM,MP)
C
C SET MATERIAL PROPERTY AS A TEMPERATURE AND CONCENTRATION POWER SERIES
C
      PARAMETER (MCS = 7)
      INCLUDE 'CON./NOLIST'
C
      DIMENSION PR(ID,1),PM(NPRCON)
      DIMENSION WK(ID,1),TT(NRCP2,1),CX(NRP2,1)
      DIMENSION NTS(MCS)
      INTEGER *4 MP, MPT, MTEN
C
C USE ONLY LAST DIGIT IN NO CONCENTRATION CASE
C
      IF (.NOT. LC) MP = MOD(MP,10)
C
C ONLY SET CONSTANT FUNCTIONS ONCE
C
      IF (ISTEP .GT. 0  .AND.  MP .EQ. 1 ) RETURN
C
C PROCESS DECIMAL DIGITS OF MP TO GET RANGE OF POLYNOMIAL TERMS
C 124 MEANS THERE ARE 4 POWERS OF T (i.e. a + bT + cTT + dTTT)
C                     2 POWERS OF T MULTIPLYING C (i.e. C(e + fT) )
C                     1 POWERS OF T MULTIPLYING C * C (i.e. CCg)
C   - IN THAT ORDER
C     SUM OF DIGITS MUST BE LESS THAN NPRCON
C
      MPT = MP
      NPC = 0
      NCS = MCS
      DO 1 M = MCS, 1, -1
      MTEN = 10 ** (M-1)
      NTS(M) = MPT / MTEN
      IF (NTS(M) .EQ. 0) NCS = M - 1
      NPC = NPC + NTS(M)
1     MPT = MPT - NTS(M) * MTEN
C
      IF (NPC .GT. NPRCON) CALL STOPP ('PROPT. NPC .GT. NPRCON%')
      IF (MPT .NE. 0     ) CALL STOPP ('PROPT. MP WRONG%')
C
C INITIALIZE TEMPERATURE POLYNOMIAL FOR HIGHEST C POWER
C
      DO 10 K = KB, KE
      DO 10 I = IB, IE
10    PR(I,K) = PM(NPC)
C
C DOWNWARD LOOP FOR HIGHEST TEMPERATURE POLYNOMIAL EVALUATION
C
      DO 20 IP = NTS(NCS)-1, 1, -1
      NPC = NPC - 1
      DO 20 K = KB, KE
      DO 20 I = IB, IE
20    PR(I,K) = PM(NPC) + TT(I,K) * PR(I,K)
C
C LOOP DOWN THROUGH LOWER POWERS OF C
C
      DO 60 M = NCS-1, 1, -1
      NPC = NPC - 1
```

```
590   C
600   C INITIALIZE TEMPERATURE POLYNOMIAL
610   C
620           DO 30 K = KB, KE
630           DO 30 I = IB, IE
640   30      WK(I,K) = PM(NPC)
650   C
660   C DOWNWARD LOOP FOR TEMPERATURE POLYNOMIAL EVALUATION
670   C
680           DO 40 IP = NTS(M)-1, 1, -1
690           NPC = NPC - 1
700           DO 40 K = KB, KE
710           DO 40 I = IB, IE
720   40      WK(I,K) = PM(NPC) + TT(I,K) * WK(I,K)
730   C
740   C UPDATE THE PROPERTY FOR THIS C POWER
750   C
760           DO 60 K = KB, KE
770           DO 60 I = IB, IE
780   60      PR(I,K) = WK(I,K) + CX(I,K) * PR(I,K)
790   C
800           RETURN
810   C
820           END
```

```
   10            SUBROUTINE PROPIN(PR,PRD,PM,MP,X,NC)
   20    C
   30    C SET MATERIAL PROPERTY PR AT INTERFACE AS A POWER SERIES IN X
   40    C SET MATERIAL PROPERTY DERIVATIVE PRD AT INTERFACE
   50    C
   60            INCLUDE 'COM./NOLIST'
   70    C
   80            DIMENSION PR(NC),PM(NPRCON),X(NC),PRD(NC)
   90            INTEGER *4 MP
  100    C
  110            MPA = ABS(MP)
  120            IF (MPA .GT. NPRCON) CALL STOPP ('PROPIN. MPA WRONG%')
  130    C
  140    C ONLY SET CONSTANT FUNCTIONS ONCE
  150    C
  160            IF (ISTEP .GT. 0  .AND.  MPA .EQ. 1 ) RETURN
  170    C
  180    C INITIALIZE POLYNOMIAL
  190    C
  200            DO 10 I = 1, NC
  210            PRD(I) = PM(MPA) * (MPA-1)
  220    10      PR(I) = PM(MPA)
  230    C
  240    C DOWNWARD LOOP FOR POLYNOMIAL EVALUATION
  250    C
  260            DO 20 IP = MPA-1, 1, -1
  270            DO 20 I = 1, NC
  280            IF (IP .GT. 1) PRD(I) = PM(IP) * (IP-1) + X(I) * PRD(I)
  290    20      PR(I) = PM(IP) + X(I) * PR(I)
  300    C
  310            RETURN
  320    C
65538            END
```

```
   10            SUBROUTINE PROPD
   20    C
   30    C MATERIAL PROPERTIES DIAGNOSTICS OVER TEMPERATURE RANGE, FOR FIXED C, E
   40    C
   50            INCLUDE 'COM./NOLIST'
   60    C
   70            PARAMETER (M=11)
   80            COMMON PD(M), WK(M)
   90    C
  100    C INITIALIZE TEMPERATURES
  110    C
  120            DO 1 I = 1, M
  130            C(I,1) = CTOP
  140    1       T(I,1) = TBOT + (TTOP - TBOT) * (I-1.) / (M-1.)
  150    C
  160            WRITE (6,5)
  170    5       FORMAT(//T30,'MATERIAL PROPERTIES OVER THE TEMPERATURE RANGE,'
  180            1                ,' FOR FIXED C = CTOP')
  190    C
  200            WRITE (6,3)'TEMPERATURE   ',' deg C',(T(I,1),I=1,M)
  210    C
  220    C LOOP OVER PROPERTIES
  230    C
  240            DO 2 IP = 1, NMATPR
  250            IF (IP .GE. 7  .AND.  IP .LE. 9) GO TO 2
  260            CALL PROPT(PD,WK,T,C,M,1,M,1,1,PRPMAT(1,IP),MATPRT(IP))
  270            WRITE (6,3)CMATPR(IP),CUNITS(IP),PD
  280    2       CONTINUE
  290    C
  300    3       FORMAT(/1X,A,A/12X,1P,11E11.3)
  310    C
  320            RETURN
65538            END
```

```
      SUBROUTINE PROPDI
C
C INTERFACE PROPERTIES DIAGNOSTICS OVER C RANGE FROM 0 TO 2*CTOP
C       OR TO 2*CMELT IF IT IS LARGER.
C
        INCLUDE 'COM./NOLIST'
C
        PARAMETER (M=11)
        COMMON PD(M), WK(M), CD(M), TD(M)
C
C NOT IF CTOP IS ZERO
C
        IF (CTOP .EQ. 0) RETURN
C
C GET TMELT AND CMELT
C
        CALL GTMCM(TMELT,CMELT,CTOP)
C
C ROUND
C
        CTP = ROUND(MAX(CTOP,CMELT))
C
C INITIALIZE C
C
        DO 1 I = 1, M
1       CD(I) = CTP * 2. * (I-1.) / (M-1.)
C
        WRITE (6,5)
5       FORMAT(//T30,'INTERFACE PROPERTIES OVER THE '
        1       ,'CONCENTRATION RANGE')
C
        WRITE (6,3)'CONCENTRATION',' ',(CD(I),I=1,M)
C
        NNEWT = 10
C
C LOOP OVER PROPERTIES
C
        DO 2 IP = 9, 8, -1
        IF (MATPRT(IP) .GT. 0) THEN
C
C                       INITIAL APPROXIMATION
C
                DO 6 I = 1, M
6               TD(I) = (CD(I)-PRPMAT(1,IP))/PRPMAT(2,IP)
C
C                       NEWTON ITERATION
C
                DO 4 INEWT = 1, NNEWT
                CALL PROPIN(PD,WK,PRPMAT(1,IP),MATPRT(IP),TD,M)
                DO 4 I = 1, M
4               TD(I) = TD(I) - (PD(I)-CD(I))/WK(I)
        ELSE
                CALL PROPIN(TD,WK,PRPMAT(1,IP),MATPRT(IP),CD,M)
        ENDIF
        WRITE (6,3)CMATPR(IP),' deg C' ,TD
2       CONTINUE
C
3       FORMAT(/1X,A,A/12X,1P,11E11.3)
```

```
  590   C
  600             IF (MATPRT(7) .GT. 0) THEN
  610                     CALL PROPIN(PD,WK,PRPMAT(1,7),MATPRT(7),TD,M)
  620             ELSE
  630                     CALL PROPIN(PD,WK,PRPMAT(1,7),MATPRT(7),CD,M)
  640             ENDIF
  650             WRITE (6,3)CMATPR(7),CUNITS(7),PD
  660   C
  670             RETURN
65588             END
```

```
      SUBROUTINE SINTPR
C
C SET INTERFACE PROPERTIES
C
      INCLUDE 'COM./NOLIST'
C
C EQUATION OF STATE FOR INTERFACE. FOUR DISTINCT CASES.
C T, CM, CC ARE ALL INITIALIZED IN INIT0 AND UPDATED IN TFSTEP
C THIS SECTION RECOMPUTES 2 OF THEM TO BE CONSISTENT WITH THE THIRD
C THE DERIVATIVES DCMDT AND DCCDT ARE REQUIRED FOR TFSTEP
C
      IF (MATPRT(8) .GT. 0) THEN
            IF (MATPRT(9) .GT. 0) THEN
                  CALL PROPIN(CINTMH,DCMDT,PRPMAT(1,8),MATPRT(8),T
                  CALL PROPIN(CINTCH,DCCDT,PRPMAT(1,9),MATPRT(9),T
            ELSE
                  CALL PROPIN(TINTH,DTDCC,PRPMAT(1,9),MATPRT(9),CI
                  CALL PROPIN(CINTMH,DCMDT,PRPMAT(1,8),MATPRT(8),T
                  DO 10 I = 1, NRP2
10                DCCDT(I) = 1./DTDCC(I)
            ENDIF
      ELSE
            IF (MATPRT(9) .GT. 0) THEN
                  CALL PROPIN(TINTH,DTDCM,PRPMAT(1,8),MATPRT(8),CI
                  CALL PROPIN(CINTCH,DCCDT,PRPMAT(1,9),MATPRT(9),T
                  DO 20 I = 1, NRP2
20                DCMDT(I) = 1./DTDCM(I)
            ELSE
                  CALL PROPIN(TINTH,DTDCC,PRPMAT(1,9),MATPRT(9),CI
C
C                       NEWTON ITERATION TO CORRECT CM
C
                  DO 30 INEWT = 1, 2
                  CALL PROPIN(WK1,DTDCM,PRPMAT(1,8),MATPRT(8),CINT
                  DO 30 I = 1, NRP2
30                CINTMH(I) = CINTMH(I) - (WK1(I)-TINTH(I)) / DTDC
                  DO 40 I = 1, NRP2
                  DCMDT(I) = 1./DTDCM(I)
40                DCCDT(I) = 1./DTDCC(I)
            ENDIF
      ENDIF
C
C APPLY SYMMETRY AT AXIS, AND EXTRAPOLATE AT AMPOULE.
C
      TINTH(1) = TINTH(2)
      TINTH(NRP2) = TINTH(NRP1)*2 - TINTH(NR)
      CINTMH(1) = CINTMH(2)
      CINTMH(NRP2) = CINTMH(NRP1)*2 - CINTMH(NR)
      CINTCH(1) = CINTCH(2)
      CINTCH(NRP2) = CINTCH(NRP1)*2 - CINTCH(NR)
      DCMDT(1) = DCMDT(2)
      DCMDT(NRP2) = DCMDT(NRP1)*2 - DCMDT(NR)
      DCCDT(1) = DCCDT(2)
      DCCDT(NRP2) = DCCDT(NRP1)*2 - DCCDT(NR)
      CALL PRNT(TINTH,NRP2,8)
      IF (ISTEP .GT. 0) CALL PRNT(DTINTH,NRP2,9)
      CALL PRNT(CINTMH,NRP2,10)
      CALL PRNT(CINTCH,NRP2,11)
```

```
590             RETURN
600             END
```

```
      SUBROUTINE TFSTEP
C
C UPDATES TEMPERATURE, CONCENTRATION, INTERFACE AND MESH.
C
      INCLUDE 'COM./NOLIST'
C
C
C SET DIAG
C INITIALIZE BOUNDARY OUTWARD FLUX DIAGNOSTICS
C INITIALIZE MEANS AND INTEGRAL
C
      DIAG = ( ISTEP/NDIAG*NDIAG .EQ. ISTEP )
     1       .OR.  ( ISTEP .EQ. NSTEP )
C
      IF (DIAG) THEN
              FRITE = 0.
              FTOP = 0
              FBOT = 0
              FINT = 0
C
              XINT = 0
              XTOP = 0
C
              TUPM = 0
              CUPM = 0
      ENDIF
C
C NUF LOGIC TO DO UA SETUP AND DECOMPOSITION
C
      J = ISTEP - 1
      NF4 = MAX(1,NUF/4)
      NF2 = NF4 * 2
      LUF = LU  .AND.  ( (J/NUF*NUF .EQ. J) .OR.
     1       (ISEGR .EQ. 0  .AND.
     1       ( (J/NF4*NF4 .EQ. J .AND. J .LT. NUF-NF4) .OR.
     1         (J/NF2*NF2 .EQ. J .AND. J .LT. 3*NUF-NF2)
     1       ))      )
C
C NCF LOGIC TO DO CA SETUP AND DECOMPOSITION
C
      J = ISTEP - 1
      NF4 = MAX(1,NCF/4)
      NF2 = NF4 * 2
      LCF = LC  .AND.  ( (J/NCF*NCF .EQ. J) .OR.
     1       (ISEGR .EQ. 0  .AND.
     1       ( (J/NF4*NF4 .EQ. J .AND. J .LT. NCF-NF4) .OR.
     1         (J/NF2*NF2 .EQ. J .AND. J .LT. 3*NCF-NF2)
     1       ))      )
C
C NTF LOGIC TO DO TA SETUP AND DECOMPOSITION
C
      J = ISTEP - 1
      NF4 = MAX(1,NTF/4)
      NF2 = NF4 * 2
      LTF = (J/NTF*NTF .EQ. J) .OR.
     1       (ISEGR .EQ. 0  .AND.
     1       ( (J/NF4*NF4 .EQ. J .AND. J .LT. NTF-NF4) .OR.
     1         (J/NF2*NF2 .EQ. J .AND. J .LT. 3*NTF-NF2)
```

```
590           1       ))
600     C
610     C LLF MEANS DO LEQ SETUP AND DECOMPOSITION
620     C
630           LLF = LTF .OR. LCF
640     C
650     C GET EXPLICIT RIGHT HAND SIDES
660     C GET COEFFICIENTS FOR CRANK-NICOLSON INVERSION
670     C
680     C
690     C     FLUXES AT TOP BOUNDARY
700     C
710           CALL TFZ(1)
720     C
730     C                              MAIN SWEEP DOWN THRU LAYERS
740     C
750           DO 100 K = 2, NZCP1
760     C
770     C INITIALIZE USING PRECALCULATED FLUXES ABOVE
780     C
790           DO 5 I = 2, NRCP1
800     5     DT(I,K) = TFZHW(I)
810     C
820           IF (LC .AND. K .LE. NZP1) THEN
830                   DO 6 I = 2, NRP1
840     6             DC(I,K) = CFZHW(I)
850           ENDIF
860     C
870           CALL TFZ(K)
880           CALL TFR(K)
890     C
900     C GET DT, THE EXPLICIT RIGHT HAND SIDE TIMES THE CELL VOLUME
910     C         I.E.  - DI.FR - DK.FZ
920     C
930     1     FORMAT (/1X,A,(T11,1P,12E10.2))
940     C     WRITE(6,1) 'TFRWH', (TFRWH(I),I=1,NRP1)
950     C     WRITE(6,1) 'TFZHW', (TFZHW(I),I=1,NRP1)
960     C
970           DO 40 I = 2, NRCP1
980     40    DT(I,K) = DT(I,K) - TFRWH(I) + TFRWH(I-1) - TFZHW(I)
990     C
1000          IF (LC .AND. K .LE. NZP1) THEN
1010                  DO 50 I = 2, NRP1
1020    50            DC(I,K) = DC(I,K) - CFRWH(I) + CFRWH(I-1) - CFZHW(I)
1030          ENDIF
1040    C
1050    C CORRECT TFZHW ON INTERFACE TO CRYSTAL VALUE, FOR NEXT K VALUE.
1060    C TFZ PREVIOUSLY CALLED TCINT
1070    C
1080          IF (K .EQ. NZP1) THEN
1090                  DO 90 I = 2, NRP1
1100    90            TFZHW(I) = TFZC(I)
1110          ENDIF
1120    C
1130    100   CONTINUE
1140    C
1150    C
1160    C DO CRANK-NICOLSON INVERSION
```

```
1170    C
1180            CALL CNINV
1190    C
1200    C
1210    C UPDATE TINTH(I), CINTMH(I), AND CINTCH(I), MAKE THEM CONSISTENT, AND P
1220    C
1230            IF (LC) THEN
1240                    DO 130 I = 2, NRP1
1250                     TINTH(I) =  TINTH(I) + DTINTH(I)
1260                    CINTMH(I) = CINTMH(I) + DTINTH(I) * DCMDT(I)
1270    130             CINTCH(I) = CINTCH(I) + DTINTH(I) * DCCDT(I)
1280    C
1290                    CALL SINTPR
1300    C
1310            ENDIF
1320    C
1330    C
1340    C UPDATE MESH AND CALL PRNT
1350    C
1360            DO 115 I = 2, NRP1
1370    115     FINTH(I) = FINTH(I) + DFINTH(I)
1380    C
1390            CALL MESH
1400    C
1410    C UPDATE TEMPERATURE AND APPLY TEMPERATURE BOUNDARY CONDITIONS.
1420    C
1430            DO 110 K = 2, NZCP1
1440            DO 110 I = 2, NRCP1
1450            T(I,K) = T(I,K) + DT(I,K)
1460    110     CONTINUE
1470    C
1480            CALL TBC
1490    C
1500            CALL TDIAG
1510    C
1520    C TEMPERATURE OUTPUT
1530    C
1540            CALL PRNT (T,NRCP2,2)
1550            CALL PRNT (DT,NRCP2,3)
1560    C
1570    C UPDATE CONCENTRATION AND APPLY CONCENTRATION BOUNDARY CONDITIONS.
1580    C
1590            IF (LC) THEN
1600    C
1610                    DO 120 K = 2, NZP1
1620                    DO 120 I = 2, NRP1
1630    120             C(I,K) = C(I,K) + DC(I,K)
1640    C
1650                    CALL CBC
1660    C
1670                    CALL CDIAG
1680    C
1690                    CALL PRNT (C,NRP2,12)
1700                    CALL PRNT (DC,NRP2,13)
1710    C
1720            ENDIF
1730    C
1740    C
```

```
 1750    C DIRECT ACCESS OUTPUT HANDLED AS ENTRY TO INIT2 FOR CONVENIENT CHANGE
 1760    C
 1770              CALL DDAOUT
 1780    C
 1790              RETURN
65888             END
```

```
      SUBROUTINE CNINV
C
C SOLVES SIMULTANEOUS LINEAR SYSTEM FOR dT, dC IN INTERIORS
C AND dF, dT, dC, dCC ON INTERFACE
C
      INCLUDE 'COM./NOLIST'
C
C FOR dT AND dC IN THE MELT:
C Cp(VdT/dt - rdrBkDkT.eta.df/dt) + DiCi + DkCk = - (EXPLICIT TERMS) (st
C Ci = (-RDDi + RGBi + REBikDk)dT + RFDrdF
C Ck = (-ZDDk + ZGBk + ZEBikDi)]dT + ZFDrdF
C
C Di IS THE CENTRAL DIFFERENCE OPERATOR IN THE I DIRECTION
C Bi IS THE DOUBLED AVERAGING OPERATOR IN THE I DIRECTION
C
C DrdF USES THE REPRESENTATION OF DFDR IN MESH, WITH DIDRW(I).DidF
C AND APPROPRIATE AVERAGING OR EXTRAPOLATION.
C
C THERE ARE SPECIAL REPRESENTATIONS NEAR THE EXTERIOR AND INTERIOR
C BOUNDARIES, ESPECIALLY FOR THE SLOPING MESH TERMS E AND F.
C
C FOR dT IN CRYSTAL, RG IS ZERO
C FOR dT IN AMPOULE, RG = RE = RF = ZE = ZF = 0
C
C INTERFACE CONDITIONS
C EXPRESSIONS FOR 2 T FLUXES AND ONE C FLUX IN THE k DIRECTION
C dC AND dCC ARE MULTIPLES OF dT
C 2 CONSERVATION EQUATIONS, CORRESPONDING TO THE 2 UNKNOWNS dT AND dF
C EXPRESSION FOR CONSTANT dF IN AMPOULE
C
C THERMAL FLUX DOUNDARY CONDITION
C HEAT INCREASE AT INTERFACE (LATENT HEAT RELEASE AND DIFFUSION) = 0
C = r.dr.Latheat.(WAMP+df/dt).DENCBM + Diffusion from melt and crystal
C Diffusion has explicit terms, and implicit terms like those above.
C TCINT SETS THE EXPLICIT TERMS AND THE ARRAYS TZD, TZE, TZF, TZG, FOR
C THE MELT AND TZICD, TZICE, TZICF, TZICG FOR THE CRYSTAL.
C
C SOLUTE FLUX DOUNDARY CONDITION
C SOLUTE INCREASE AT INTERFACE (DIFFERENCE OF FLUXES ON TWO SIDES) = 0
C = r.dr.(WAMP+df/dt).DENCBM.(Cm-Cc) + Diffusion from melt
C Diffusion has explicit terms, and implicit terms like those above.
C TCINT SETS THE EXPLICIT TERMS AND THE ARRAYS CZD, CZE, CZF, CZG.
C
C WE HAVE TRIED 3 METHODS:
C
C 1. APPROXIMATE SOLUTION AS 2 GENERALIZED TRIDIAGONAL DIVISIONS, IN THE
C                TWO COORDINATE DIRECTIONS, WITH TREATMENT FOR dF
C
C 2. ITERATIVE SOLUTION WITH 2 GENERALIZED TRIDIAGONAL DIVISIONS AT EACH
C
C 3. DIRECT GAUSSIAN ELIMINATION, USING THE BANDED MATRIX STRUCTURE
C
      CALL CNINVG
C
      RETURN
65588 END
```

```
      SUBROUTINE TBC
C
C SET UP AND APPLY BOUNDARY CONDITIONS TO TEMPERATURE AND DT.
C CALLED EVERY STEP. NOTE THAT THE SET UP MUST BE REPEATED SINCE THE MES
C
      INCLUDE 'COM./NOLIST'
C
C GET INVERSE MESH SPACING AT ZTBOT AND ZTTOP
C
      DKDZB = 1.E5
      DO 10 K = NZCP1, 1, -1
      IF (ZAW(K) .LE. ZTBOT) DKDZT = DKDZAH(K+1)
      IF (ZAW(K) .LE. ZTTOP) DKDZB = DKDZAH(K+1)
10    CONTINUE
C
C SET CONSTANT AND MULTIPLIER ARRAYS FOR RIGHT
C   ZTBSC =  0 : ADIABATIC SECTION, UNSMOOTHED
C   ZTBSC  POS : ADIABATIC SECTION, SMOOTHED OVER ZTBSC TIMES MESH SPACI
C   ZTBSC = -1 : LINEAR INTERPOLATION
C
      IF (ZTBSC .EQ. 0) THEN
            DO 20 K = 1, NZCP2
            IF (ZAH(K) .LT. ZTBOT) THEN
                  TRBC(K) = 2 * TBOT
                  TRBM(K) = -1
            ELSE IF (ZAH(K) .GT. ZTTOP) THEN
                  TRBC(K) = 2 * TTOP
                  TRBM(K) = -1
            ELSE
                  TRBC(K) = 0
                  TRBM(K) = 1
            ENDIF
20          CONTINUE
      ELSE IF (ZTBSC .EQ. -1) THEN
            DO 25 K = 1, NZCP2
            IF (ZAH(K) .GT. ZTBOT) THEN
                  TRBC(K) = 2 * TBOT
                  TRBM(K) = -1
            ELSE IF (ZAH(K) .LT. ZTTOP) THEN
                  TRBC(K) = 2 * TTOP
                  TRBM(K) = -1
            ELSE
                  TRBC(K) = 2. * (TTOP + (TBOT - TTOP)
     1            * (ZAH(K) - ZTTOP) / (ZTBOT - ZTTOP) )
                  TRBM(K) = -1
            ENDIF
25          CONTINUE
      ELSE IF (ZTBSC .GT. 0) THEN
            DO 30 K = 1, NZCP2
            WBOT = (1 + TANH((ZAH(K)-ZTBOT)/ZTBSC*DKDZT))
            WTOP = (1 - TANH((ZAH(K)-ZTTOP)/ZTBSC*DKDZB))
            TRBC(K) = WBOT * TBOT + WTOP * TTOP
            TRBM(K) = (1. - WBOT - WTOP)
30          CONTINUE
      ELSE
            CALL STOPP ('TBC. ZTBSC ILLEGAL%')
      ENDIF
C
```

```
  590   C
  600   C WRITE DIAGNOSTIC ARRAYS
  610   C
  620           IF (ISTEP .EQ. 0) THEN
  630                   WRITE (6,5) 'TRBC',TRBC
  640                   WRITE (6,5) 'TRBM',TRBM
  650                   WRITE (6,'(/'' TBC. '',
  660           1       ''ZT, ZTTOP, ZTBOT, ZB, ZTBSC, DKDZT, DKDZB ='',9F9.3)')
  670           1        ZT, ZTTOP, ZTBOT, ZB, ZTBSC, DKDZT, DKDZB
  680   5               FORMAT (//1X,A,T12,1P10E11.3,:/(T12,1P10E11.3,:))
  690           ENDIF
  700   C
  710   C
  720   C DO SIDES
  730   C
  740           DO 50 K = 1, NZCP2
  750   C
  760           T(1     ,K) = T(2     ,K)
  770           T(NRCP2,K) = T(NRCP1,K) * TRBM(K) + TRBC(K)
  780   C
  790           DT(1     ,K) = DT(2     ,K)
  800           DT(NRCP2,K) = DT(NRCP1,K) * TRBM(K)
  810   50      CONTINUE
  820   C
  830   C DO ENDS
  840   C
  850           DO 60 I = 1, NRCP2
  860   C
  870           T(I,     1) = 2 * TTOP - T(I,     2)
  880           T(I,NZCP2) = 2 * TBOT - T(I,NZCP1)
  890   C
  900           DT(I,     1) = - DT(I,     2)
  910           DT(I,NZCP2) = - DT(I,NZCP1)
  920   60      CONTINUE
  930   C
  940           RETURN
65988           END
```

```
      SUBROUTINE CBC
C
C APPLY CONCENTRATION BOUNDARY CONDITIONS
C
      INCLUDE 'COM./NOLIST'
C
C IMPOSED AT THE TOP AND BOTTOM
C
      DO 2 I = 2, NRP1
      DC(I,NZP2) = - DC(I,NZP1)
2      C(I,NZP2) = -  C(I,NZP1) + 2 * CINTMH(I)
C
      CTMULT(I) = -1.
      DC(I,1) = - DC(I,2)
2      C(I,1) = -  C(I,2) + 2 * CTOP
C
C SYMMETRY AT THE LEFT AND EXTRAPOLATE AT RIGHT
C RIGHT VALUE IS USED ONLY FOR GRAPHICS
C
      DO 1 K = 1, NZP2
      DC(NRP2,K) = DC(NRP1,K)*2 - DC(NR,K)
       C(NRP2,K) =  C(NRP1,K)*2 -  C(NR,K)
      DC(1,K) = DC(2,K)
1      C(1,K) =  C(2,K)
C
      RETURN
      END
```

```
      SUBROUTINE TFR(K)
C
C GET R TEMPERATURE AND CONCENTRATION FLUXES AT LEVEL K.
C SAVE ARRAYS FOR CRANK-NICOLSON INVERSION
C
      INCLUDE 'COM./NOLIST'
C
C
C DO AMPOULE
C
      DO 100 I = NRP2, NRCP1
      TDRWH(I) = (COND(I,K) + COND(I+1,K)) * RB2DRW(I) * DZDKAH(K)
100   TFRWH(I) = - TDRWH(I) * (T(I+1,K) - T(I,K) )
C
C
C DO SAMPLE
C
C
C ADVECTION FLUXES
C
      IF (K .LE. NZP1) THEN
            DO 4 I = 2, NR
            IF (LC) CGRWH(I) = ( - PSI(I,K) + PSI(I,K-1)) * 0.5
4           TGRWH(I) = ( - PSI(I,K) + PSI(I,K-1))
     1                 * (CP(I,K)+CP(I+1,K)) * 0.25
            IB = NRP1
      ELSE
            IB = 2
      ENDIF
C
      DO 8 I = IB, NRP1
8     TGRWH(I) = 0.
C
C
C
C DO SAMPLE SEPARATELY FOR TEMPERATURE AND CONCENTRATION COMBINED
C
C
      IF (K .LE. NZP1  .AND.  LC) THEN
C
C
C
      DO 20 I = 2, NR
      E = COND(I,K) + COND(I+1,K)
      F = DIFF(I,K) + DIFF(I+1,K)
      TERWH(I) = E * ETAH(K)
      CERWH(I) = F * ETAH(K)
      DF = RB2DRW(I) * (DZTDKH(K) + DETDKH(K)*RFINTW(I))
      TDRWH(I) = E * DF
      CDRWH(I) = F * DF
20    CONTINUE
C
C CORRECT FOR UPWIND DIFFERENCE
C
      IF (LU  .AND.  TUPWND .NE. 0) THEN
            DO 40 I = 2, NR
            RAT = ABS(TGRWH(I)*TUPWND/TDRWH(I))
            TUPM = MAX(TUPM,RAT)
```

```
590   40                TDRWH(I) = TDRWH(I) * MAX(1.,RAT)
600             ENDIF
610   C
620             IF (LU  .AND.  CUPWND .NE. 0) THEN
630                     DO 50 I = 2, NR
640                     RAC = ABS(CGRWH(I)*CUPWND/CDRWH(I))
650                     CUPM = MAX(CUPM,RAC)
660   50                CDRWH(I) = CDRWH(I) * MAX(1.,RAC)
670             ENDIF
680   C
690   C SLOPING MESH TERM D2T = 4DKBIKT MODIFIED NEXT TO INTERFACE
700   C
710             IF (K .EQ. NZP1) THEN
720                     DO 51 I = 2, NR
730                     D2T(I) = 2. * (TINTH(I+1) + TINTH(I))
740             1                       - T(I+1,NZ) - T(I,NZ)
750             1                       - T(I+1,NZP1) - T(I,NZP1)
760   51                D2C(I) = 2. * (CINTMH(I+1) + CINTMH(I))
770             1                       - C(I+1,NZ) - C(I,NZ)
780             1                       - C(I+1,NZP1) - C(I,NZP1)
790             ELSE
800                     DO 60 I = 2, NR
810                     D2T(I) = T(I+1,K+1) + T(I,K+1)
820             1                 - T(I+1,K-1) - T(I,K-1)
830   60                D2C(I) = C(I+1,K+1) + C(I,K+1)
840             1                 - C(I+1,K-1) - C(I,K-1)
850             ENDIF
860   C
870   C
880   C
890             ELSE IF (K .GE. NZP2  .OR.  (.NOT. LC) ) THEN
900   C
910   C
920   C
930   C SAMPLE SECTION WITH TEMPERATURE BUT NO CONCENTRATION
940   C
950             DO 25 I = 2, NR
960             E = COND(I,K) + COND(I+1,K)
970             TERWH(I) = E * ETAH(K)
980             DF = RB2DRW(I) * (DZTDKH(K) + DETDKH(K)*RFINTW(I))
990             TDRWH(I) = E * DF
1000  25        CONTINUE
1010  C
1020  C CORRECT FOR UPWIND DIFFERENCE
1030  C
1040            IF (LU  .AND.  TUPWND .NE. 0  .AND.  K .LE. NZP1) THEN
1050                    DO 47 I = 2, NR
1060                    RAT = ABS(TGRWH(I)*TUPWND/TDRWH(I))
1070                    TUPM = MAX(TUPM,RAT)
1080  47                TDRWH(I) = TDRWH(I) * MAX(1.,RAT)
1090            ENDIF
1100  C
1110  C
1120  C SLOPING MESH TERM D2T = 4DKBIKT MODIFIED NEXT TO INTERFACE
1130  C
1140            IF (K .EQ. NZP1) THEN
1150                    DO 55 I = 2, NR
1160  55                D2T(I) = 2. * (TINTH(I+1) + TINTH(I))
```

```
1170            1                - T(I+1,NZ) - T(I,NZ)
1180            1                - T(I+1,NZP1) - T(I,NZP1)
1190            ELSE IF (K .EQ. NZP2) THEN
1200                    DO 65 I = 2, NR
1210    65              D2T(I) = T(I+1,NZP2+1) + T(I,NZP2+1)
1220            1               + T(I+1,NZP2) + T(I,NZP2)
1230            1               - 2. * (TINTH(I+1) + TINTH(I))
1240            ELSE
1250                    DO 75 I = 2, NR
1260    75              D2T(I) = T(I+1,K+1) + T(I,K+1)
1270            1               - T(I+1,K-1) - T(I,K-1)
1280            ENDIF
1290    C
1300    C
1310    C
1320            ENDIF
1330    C
1340    C END IF K BLOCKS FOR INTERFACE AND MELT CONCENTRATION
1350    C
1360    C
1370    C
1380    C
1390    C DO SAMPLE BOUNDARY, FOR TEMPERATURE
1400    C
1410            TDRWH(NRP1) = RSAM * DZDKAH(K) * 2
1420            1       * (COND(NRP1,K)*DIDRW(NRP1) * COND(NRP2,K)*DIDRAB)
1430            1       / (COND(NRP1,K)*DIDRW(NRP1) + COND(NRP2,K)*DIDRAB)
1440            TERWH(NRP1) = 2 * (COND(NRP1,K)*ETAH(K) * COND(NRP2,K)*DIDRAB)
1450            1       / (COND(NRP1,K)*DIDRW(NRP1) + COND(NRP2,K)*DIDRAB)
1460    C
1470    C SLOPING MESH TERM D2T = 4DKBIKT MODIFIED NEXT TO INTERFACE
1480    C
1490            IF (K .EQ. NZP1) THEN
1500                    D2T(NRP1) = 2 * (- T(NRP2,NZ) - T(NRP1,NZ)
1510            1                       + T(NRP2,NZP1) + T(NRP1,NZP1) )
1520            ELSE IF (K .EQ. NZP2) THEN
1530                    D2T(NRP1) = 2 * (T(NRP2,NZ+3) + T(NRP1,NZ+3)
1540            1                       - T(NRP2,NZP2) - T(NRP1,NZP2) )
1550            ELSE
1560                    D2T(NRP1) = T(NRP2,K+1) + T(NRP1,K+1)
1570            1               - T(NRP2,K-1) - T(NRP1,K-1)
1580            ENDIF
1590    C
1600    C
1610    C GET CONCENTRATION FLUX
1620    C
1630            IF (LC .AND.  K .LE. NZP1) THEN
1640                    DO 80 I = 2, NR
1650    80              CFRWH(I) = CGRWH(I) * (C(I+1,K) + C(I,K) )
1660            1               - CDRWH(I) * (C(I+1,K) - C(I,K) )
1670            1               + CERWH(I) * RDRF8H(I) * D2C(I)
1680            ENDIF
1690    C
1700    C
1710    C GET HEAT FLUX
1720    C
1730            DO 90 I = 2, NRP1
1740    90      TFRWH(I) = TGRWH(I) * (T(I+1,K) + T(I,K) )
```

```
1750            1       - TDRWH(I) * (T(I+1,K) - T(I,K) )
1760            1       + TERWH(I) * RDRF8H(I) * D2T(I)
1770    C
1780    C ZERO CONCENTRATION FLUX AT AXIS AND SAMPLE BOUNDARY
1790    C
1800            IF (LC) THEN
1810                    CFRWH(1) = 0.
1820                    CFRWH(NRP1) = 0.
1830            ENDIF
1840    C
1850    C SAVE THE COEFFICIENTS IF NECESSARY
1860    C
1870            BEFB8 = BEF/8
1880    C
1890            IF (LTF) THEN
1900                    DO 110 I = 2, NRCP1
1910    110             TRD(I,K) = BETTD * TDRWH(I)
1920    C
1930                    IF (K .LE. NZP1) THEN
1940                            DO 113 I = 2, NRP1
1950    113                     TRG(I,K) = BETTA * TGRWH(I)
1960                    ENDIF
1970    C
1980                    DO 115 I = 2, NRP1
1990                    TRF(I,K) = BEFB8 * TERWH(I) * RW(I) * D2T(I)
2000    115             TRE(I,K) = BETTD * TERWH(I) * RDRF8H(I)
2010            ENDIF
2020    C
2030            IF (LCF  .AND.  K .LE. NZP1) THEN
2040                    DO 70 I = 2, NR
2050                    CRD(I,K) = BETCD * CDRWH(I)
2060                    CRE(I,K) = BETCD * CERWH(I) * RDRF8H(I)
2070                    CRF(I,K) = BEFB8 * CERWH(I) * RW(I) * D2C(I)
2080    70              CRG(I,K) = BETCA * CGRWH(I)
2090            ENDIF
2100    C
2110    C SET AXIS FLUX
2120    C
2130            TFRWH(1) = 0.
2140    C
2150    C UPDATE RIGHT HEAT FLUX
2160    C
2170    C HEAT FLUXES (cal/cm2/sec) AND SOLUTE FLUXES (cc/cm2/sec = cm/sec)
2180    C ARE OUTWARD FROM THE COMPUTATIONAL DOMAIN (NEGATIVE AT THE TOP),
2190    C AND INTO THE INTERFACE, FOR HEAT, OR TO THE CRYSTAL, FOR SOLUTE.
2200    C
2210    C ADVECTIVE HEAT FLUX IS DEFINED AS CpT.U
2220    C
2230            FRITE = FRITE - TFRWH(NRCP1)
2240    C
2250            RETURN
2260            END
```

```
      SUBROUTINE TFZ(K)
C
C GET Z TEMPERATURE FLUXES AT LEVEL K, IN SAMPLE AND AMPOULE.
C GET Z CONCENTRATION FLUXES AT LEVEL K, IN MELT.
C GET COEFFICIENT ARRAYS FOR CNINV.
C
      INCLUDE 'COM./NOLIST'
C
      BEFB8 = BEF/8
C
C DO AMPOULE
C
      DO 70 I = NRP2, NRCP1
      TGZ = - RDRWQH(I) * (CP(I,K)+CP(I,K+1))
      TDZHW(I) = (COND(I,K) + COND(I,K+1)) * RDRB2H(I) * DKDZAW(K)
      TFZHW(I) = - TDZHW(I) * (T(I,K+1) - T(I,K) )
     1       + TGZ * (T(I,K+1) + T(I,K) )
      TZD(I,K) = BETTD * TDZHW(I)
70    TZG(I,K) = BETTA * TGZ
C
C GET ADVECTION COEFFICIENTS IN THE SAMPLE
C
      IF (K .LE. NZP1) THEN
            IF (LC) THEN
                  DO 3 I = 2, NRP1
                  CGZHW(I) = (PSI(I,K) - PSI(I-1,K)) * 0.5
                  TGZHW(I) = (PSI(I,K) - PSI(I-1,K))
3                        * (CP(I,K)+CP(I,K+1)) * 0.25
      1       ELSE
                  DO 5 I = 2, NRP1
                  TGZHW(I) = (PSI(I,K) - PSI(I-1,K))
5                        * (CP(I,K)+CP(I,K+1)) * 0.25
      1       ENDIF
      ELSE
            DO 10 I = 2, NRP1
            TGZHW(I) = - RDRWQH(I) * (CP(I,K)+CP(I,K+1))
10    ENDIF
C
C
C DO SAMPLE
C
C SAMPLE SPLIT INTO INTERFACE, CONCENTRATION AND TEMPERATURE IN MELT,
C AND TEMPERATURE IN BOTH OR JUST IN CRYSTAL.
C
C
      IF (K .EQ. NZP1) THEN
C
C
            CALL TCINT
            RETURN
C
C
      ELSE IF (K .LE. NZ  .AND.  LC) THEN
C
C
C
      DO 20 I = 2, NRP1
      E = COND(I,K) + COND(I,K+1)
```

```
590             F = DIFF(I,K) + DIFF(I,K+1)
600             TEZHW(I) = E * ETAW(K)
610             CEZHW(I) = F * ETAW(K)
620             DF = (RDRB2H(I) + ZDRFH(I)*ETA2W(K))
630            1        / (DZTDKW(K) + DETDKW(K)*RFINTH(I))
640             TDZHW(I) = E * DF
650             CDZHW(I) = F * DF
660     20      CONTINUE
670     C
680     C CORRECT FOR UPWIND DIFFERENCE
690     C
700             IF (LU  .AND.  TUPWND .NE. 0) THEN
710                     DO 40 I = 2, NRP1
720                     RAT = ABS(TGZHW(I)*TUPWND/TDZHW(I))
730                     TUPM = MAX(TUPM,RAT)
740     40              TDZHW(I) = TDZHW(I) * MAX(1.,RAT)
750             ENDIF
760     C
770             IF (LU  .AND.  CUPWND .NE. 0) THEN
780                     DO 50 I = 2, NRP1
790                     RAC = ABS(CGZHW(I)*CUPWND/CDZHW(I))
800                     CUPM = MAX(CUPM,RAC)
810     50              CDZHW(I) = CDZHW(I) * MAX(1.,RAC)
820             ENDIF
830     C
840     C SLOPING MESH TERM D2T = 4DIBIKT MODIFIED NEXT TO INTERFACE
850     C
860             DO 60 I = 2, NR
870             D2T(I) = T(I+1,K+1) + T(I+1,K)
880            1        - T(I-1,K+1) - T(I-1,K)
890             D2C(I) = C(I+1,K+1) + C(I+1,K)
900            1        - C(I-1,K+1) - C(I-1,K)
910     60      CONTINUE
920             D2T(NRP1) = ( T(NRP1,K+1) + T(NRP1,K)
930            1                - T(NR,K+1) - T(NR,K) ) * 2
940             D2C(NRP1) = ( C(NRP1,K+1) + C(NRP1,K)
950            1                - C(NR,K+1) - C(NR,K) ) * 2
960     C
970     C SAVE THE C COEFFICIENTS IF NECESSARY
980     C
990             IF (LCF) THEN
1000                    DO 62 I = 2, NRP1
1010                    CZD(I,K) = BETCD * CDZHW(I)
1020                    CZE(I,K) = BETCD * CEZHW(I) * RDRF8H(I)
1030                    CZF(I,K) = BEFB8 * CEZHW(I) * RH(I) * D2C(I)
1040    62              CZG(I,K) = BETCA * CGZHW(I)
1050            ENDIF
1060    C
1070    C
1080    C
1090            ELSE IF (K .GE. NZP2  .OR.  (.NOT. LC) ) THEN
1100    C
1110    C T IN CRYSTAL, AND ALSO IN MELT FOR THE NO-CONCENTRATION CASE
1120    C
1130            DO 25 I = 2, NRP1
1140            E = COND(I,K) + COND(I,K+1)
1150            TEZHW(I) = E * ETAW(K)
1160            DF = (RDRB2H(I) + ZDRFH(I)*ETA2W(K))
```

```
1170            1       / (DZTDKW(K) + DETDKW(K)*RFINTH(I))
1180            TDZHW(I) = E * DF
1190    25      CONTINUE
1200    C
1210    C CORRECT FOR UPWIND DIFFERENCE
1220    C
1230            IF (LU  .AND.  TUPWND .NE. 0  .AND.  K .LE. NZ) THEN
1240                    DO 45 I = 2, NRP1
1250                    RAT = ABS(TGZHW(I)*TUPWND/TDZHW(I))
1260                    TUPM = MAX(TUPM,RAT)
1270    45              TDZHW(I) = TDZHW(I) * MAX(1.,RAT)
1280            ENDIF
1290    C
1300    C
1310    C SLOPING MESH TERM D2T = 4DIBIKT MODIFIED NEXT TO INTERFACE
1320    C
1330            DO 65 I = 2, NR
1340            D2T(I) = T(I+1,K+1) + T(I+1,K)
1350            1       - T(I-1,K+1) - T(I-1,K)
1360    65      CONTINUE
1370            D2T(NRP1) = ( T(NRP1,K+1) + T(NRP1,K)
1380            1               - T(NR,K+1) - T(NR,K) ) * 2
1390    C
1400    C
1410    C
1420            ENDIF
1430    C
1440    C END IF K BLOCKS FOR SAMPLE
1450    C SAMPLE SPLIT INTO INTERFACE, CONCENTRATION AND TEMPERATURE IN MELT,
1460    C AND TEMPERATURE IN BOTH OR JUST IN CRYSTAL.
1470    C
1480    C
1490    C
1500    C
1510    C GET VERTICAL HEAT FLUXES IN SAMPLE
1520    C
1530            DO 75 I = 2, NRP1
1540    75      TFZHW(I) = TGZHW(I) * (T(I,K+1) + T(I,K) )
1550            1       - TDZHW(I) * (T(I,K+1) - T(I,K) )
1560            1       + TEZHW(I) * RDRF8H(I) * D2T(I)
1570    C
1580    C
1590    C GET CONCENTRATION FLUXES
1600    C
1610            IF (LC .AND.  K .LE. NZ) THEN
1620            DO 77 I = 2, NRP1
1630    77              CFZHW(I) = CGZHW(I) * (C(I,K+1) + C(I,K) )
1640            1               - CDZHW(I) * (C(I,K+1) - C(I,K) )
1650            1               + CEZHW(I) * RDRF8H(I) * D2C(I)
1660            ENDIF
1670    C
1680    C
1690    C SAVE THE TEMPERATURE COEFFICIENTS IF NECESSARY
1700    C
1710            IF (LTF) THEN
1720                    DO 79 I = 2, NRP1
1730                    TZD(I,K) = BETTD * TDZHW(I)
1740                    TZE(I,K) = BETTD * TEZHW(I) * RDRF8H(I)
```

```
1750                    TZF(I,K) = BEFB8 * TEZHW(I) * RH(I) * D2T(I)
1760    79              TZG(I,K) = BETTA * TGZHW(I)
1770            ENDIF
1780    C
1790    C GET TOP AND BOTTOM FLUX DIAGNOSTICS
1800    C
1810    C HEAT FLUXES (cal/cm2/sec) AND SOLUTE FLUXES (cc/cm2/sec = cm/sec)
1820    C ARE OUTWARD FROM THE COMPUTATIONAL DOMAIN (NEGATIVE AT THE TOP).
1830    C
1840    C ADVECTIVE HEAT FLUX IS DEFINED AS CpT.U
1850    C
1860    C SOLUTE UNIT IS PRODUCT OF CONCENTRATION AND FLUID VOLUME.
1870    C
1880    C GET TOP FLUX DIAGNOSTICS
1890    C
1900            IF (K .EQ. 1) THEN
1910                    DO 80 I = 2, NRCP1
1920    80              FTOP = FTOP + TFZHW(I)
1930                    IF (LC) THEN
1940                            DO 85 I = 2, NRP1
1950    85                      XTOP = XTOP + CFZHW(I)
1960                    ENDIF
1970            ENDIF
1980    C
1990    C GET BOTTOM FLUX DIAGNOSTIC
2000    C
2010            IF (K .EQ. NZCP1) THEN
2020                    DO 90 I = 2, NRCP1
2030    90              FBOT = FBOT - TFZHW(I)
2040            ENDIF
2050    C
2060            RETURN
62838           END
```

```
      SUBROUTINE TCINT
C
C INTERFACE CONDITIONS FOR TEMPERATURE AND CONCENTRATION
C
C GET FLUXES TFZHW AND CFZHW ON MELT SIDE
C       AND FLUX TFZC ON CRYSTAL SIDE, FOR TFSTEP.
C GET COEFFICIENT ARRAYS TZ%(I,NZP1) AND TZIC%(I)
C       FOR CNINV DTSU AND DTSL EQUATIONS
C GET COEFFICIENT ARRAYS CZ%(I,NZP1)
C
C GET FLUX IMBALANCES DFTINT AND DFCINT FOR CORRECTION BY CNINV
C SET CORRESPONDING COEEFFICIENT ARRAYS FOR CNINV
C
      INCLUDE 'COM./NOLIST'
C
      DO 10 I = 2, NRP1
C
      CGZ = (PSI(I,NZP1) - PSI(I-1,NZP1)) * 0.5
      TGZ = (PSI(I,NZP1) - PSI(I-1,NZP1)) * 0.5 * CPINTM(I)
      TGC = - CPINTC(I) * 2. * RDRWQH(I)
C
      DINT2 =  DINTH(I) * 2.
      CDIM2 = CDINTM(I) * 2.
      CDIC2 = CDINTC(I) * 2.
C
C
      DFM = (RDRB2H(I) + ZDRFH(I))
     1       / (DZTDKW(NZP1) + DETDKW(NZP1)*RFINTH(I))
      DFC = (RDRB2H(I) + ZDRFH(I))
     1       * DKDZTC / (1. + DEDZTC * RFINTH(I))
C
      CDZ = DINT2 * DFM
      TDZ = CDIM2 * DFM
      TDC = CDIC2 * DFC
C
      IF (I .EQ. NRP1) THEN
              DT2 = 4 * (TINTH(I) - TINTH(I-1) )
              DC2 = 4 * (CINTMH(I) - CINTMH(I-1) )
      ELSE
              DT2 = 2 * (TINTH(I+1) - TINTH(I-1) )
              DC2 = 2 * (CINTMH(I+1) - CINTMH(I-1) )
      ENDIF
C
      CFZ = DINT2 *DC2*RDRF8H(I) - 2.*CDZ * (CINTMH(I) - C(I,NZP1) )
      TFZ = CDIM2 * DT2*RDRF8H(I) - 2.*TDZ * (TINTH(I)  - T(I,NZP1) )
      TFC = CDIC2 * DT2*RDRF8H(I) - 2.*TDC * (T(I,NZP2) - TINTH(I) )
C
      CFZHW(I) = 2. * CGZ * CINTMH(I) + CFZ
      TFZHW(I) = 2. * TGZ * TINTH(I) + TFZ
      TFZC(I) = 2. * TGC * TINTH(I) + TFC
C
      BEFB8 = BEF/8
C
      CZD(I,NZP1) = BETCD * CDZ
      CZE(I,NZP1) = BETCD * DINT2 * RDRF8H(I)
      CZF(I,NZP1) = BEFB8 * DINT2 * RH(I) * DC2
      CZG(I,NZP1) = BETCA * CGZ
C
```

```
590             TZD(I,NZP1) = BETTD * TDZ
600             TZE(I,NZP1) = BETTD * CDIM2 * RDRF8H(I)
610             TZF(I,NZP1) = BEFB8 * CDIM2 * RH(I) * DT2
620             TZG(I,NZP1) = BETTA * TGZ
630     C
640             TZICD(I) = BETTD * TDC
650             TZICE(I) = BETTD * CDIC2 * RDRF8H(I)
660             TZICF(I) = BEFB8 * CDIC2 * RH(I) * DT2
670             TZICG(I) = BETTA * TGC
680     C
690     C THERMAL FLUX DOUNDARY CONDITION
700     C DTFINT = EXPLICIT HEAT INCREASE AT INTERFACE (LATENT HEAT RELEASE AND
710     C        = A.L.WAMP.DENCBM + "TM - TI" + "TC - TI"
720     C        = A.L.WAMP.DENCBM     - TFZ       + TFC            (NOTE THAT D's A
730     C        = - A.L.FDOT.DENCBM + IMPLICIT TEMPERATURE TERMS
740     C        = - A.L.FDOT.DENCBM + DM(DTM-DTI) + DC(DTC-DTI)
750     C
760             RDRDEN = 2.*RDRB2H(I)*DENCBM
770             H = RDRDEN * HEATLAT(I)
780             DTFINT(I) = H * WAMP - TFZ + TFC
790             DFTINT(I) = - H * BTF(I)
800     C
810     C SOLUTE MASS FLUX BOUNDARY CONDITION
820     C DCFINT = EXPLICIT RATE OF INCREASE IN SOLUTE AT INTERFACE BY ADVECTION
830     C        = WAMP.RDR.DENCBM.(CMI - CCI) + "CM - CMI"
840     C        = WAMP.RDR.DENCBM.(CMI - CCI)     - CFZ          (NOTE THAT D IS
850     C        = - FDOT.RDR.DENCBM.(CMI - CCI) - IMPLICIT DT TERMS WITH WAMP +
860     C        = - FDOT.RDR.DENCBM.(CMI - CCI) - WAMP.RDR.DENCBM.(DCMDT-DCCCT).
870     C
880             DCFINT(I) = RDRDEN * (CINTMH(I) - CINTCH(I)) * WAMP - CFZ
890             DFCINT(I) = - RDRDEN * (CINTMH(I) - CINTCH(I)) * BTF(I)
900             DTCINT(I) = - RDRDEN * (DCMDT(I) - DCCDT(I)) * WAMP
910     C
920     C INTERFACE FLUX DIAGNOSTICS
930     C
940     C HEAT FLUXES (cal/cm2/sec) AND SOLUTE FLUXES (cc/cm2/sec = cm/sec)
950     C ARE OUTWARD FROM THE COMPUTATIONAL DOMAIN (NEGATIVE AT THE TOP),
960     C AND INTO THE INTERFACE, FOR HEAT, OR TO THE CRYSTAL, FOR SOLUTE.
970     C
980     C ADVECTIVE HEAT FLUX IS DEFINED AS CpT.U
990     C
1000            FINT = FINT - TFZHW(I) + TFZC(I)
1010            XINT = XINT - CFZHW(I)
1020    C
1030    10      CONTINUE
1040    C
1050    C DIAGNOSTICS
1060    C
1070            IF (ISTEP/1*1 .EQ. ISTEP  .AND.  ISTEP .LE. 20) THEN
1080    C       WRITE (6,'(//A,I4/)') ' INTERFACE DIAGNOSTICS AT STEP', ISTEP
1090    20      FORMAT(/1X,A6,4X,1P,12E10.2/(11X,12E10.2))
1100            WRITE (6,20) 'CFZHW',(CFZHW(I),I=2,NRP1)
1110    C       WRITE (6,20) 'TFZHW',(TFZHW(I),I=2,NRP1)
1120    C       WRITE (6,20) 'TFZC',(TFZC(I),I=2,NRP1)
1130    C       WRITE (6,20) 'CZD',(CZD(I,NZP1),I=2,NRP1)
1140    C       WRITE (6,20) 'CZE',(CZE(I,NZP1),I=2,NRP1)
1150    C       WRITE (6,20) 'CZF',(CZF(I,NZP1),I=2,NRP1)
1160    C       WRITE (6,20) 'CZG',(CZG(I,NZP1),I=2,NRP1)
```

```
1170    C       WRITE (6,20) 'TZD',(TZD(I,NZPl),I=2,NRPl)
1180    C       WRITE (6,20) 'TZE',(TZE(I,NZPl),I=2,NRPl)
1190    C       WRITE (6,20) 'TZF',(TZF(I,NZPl),I=2,NRPl)
1200    C       WRITE (6,20) 'TZG',(TZG(I,NZPl),I=2,NRPl)
1210    C       WRITE (6,20) 'TZICD',(TZICD(I),I=2,NRPl)
1220    C       WRITE (6,20) 'TZICE',(TZICE(I),I=2,NRPl)
1230    C       WRITE (6,20) 'TZICF',(TZICF(I),I=2,NRPl)
1240    C       WRITE (6,20) 'TZICG',(TZICG(I),I=2,NRPl)
1250    C       WRITE (6,20) 'DTFINT',(DTFINT(I),I=2,NRPl)
1260    C       WRITE (6,20) 'DFTINT',(DFTINT(I),I=2,NRPl)
1270    C       WRITE (6,20) 'DCFINT',(DCFINT(I),I=2,NRPl)
1280    C       WRITE (6,20) 'DFCINT',(DFCINT(I),I=2,NRPl)
1290    C       WRITE (6,20) 'DTCINT',(DTCINT(I),I=2,NRPl)
1300            ENDIF
1310    C
1320            RETURN
1330            END
```

```
   10            SUBROUTINE CNINVG
   20    C
   30    C SOLVES SIMULTANEOUS LINEAR SYSTEM FOR dT, dC IN INTERIORS
   40    C AND dF, dT, dC, dCC ON INTERFACE
   50    C USES BANDED STRUCTURE GAUSSIAN ELIMINATION
   60    C
   70            INCLUDE 'COM./NOLIST'
   80    C
   90    C CN%1 SETS UP %A AND %F, AND DOES A BANDED DECOMPOSITION DOWN TO THE IN
  100    C IT IS NOT REPEATED EVERY STEP
  110    C
  120    C CN%2 ELIMINATES D% EXCEPT NEXT TO THE INTERFACE, USING THE DECOMPOSED
  130    C
  140    C LEQ SETS UP AND SOLVES THE INTERFACE PROBLEM FOR THE CHANGES IN INTERF
  150    C HEIGHT AND TEMPERATURE, AND IN T AND C NEXT TO THE INTERFACE
  160    C
  170    C CN%3 SUBSTITUTES BACK TO OBTAIN D% AWAY FROM THE INTERFACE
  180    C
  190            IF (LTF) CALL CNT1
  200            CALL CNT2
  210            IF (LC) THEN
  220                    IF (LCF) CALL CNC1
  230                    CALL CNC2
  240                    CALL LEQ(NAI)
  250                    CALL CNC3
  260            ELSE
  270                    CALL LEQ(NAIS)
  280            ENDIF
  290            CALL CNT3
  300    C
  310            RETURN
65538            END
```

```
      SUBROUTINE CNC1
C
C SET UP AND L-U DECOMPOSE THE BANDED MATRIX FOR DC
C SET UP ROWS OF INTERFACE MATRIX
C ELIMINATE DC EXCEPT ADJACENT TO INTERFACE
C SET RIGHT HAND SIDE TERMS FOR INTERFACE MATRIX
C
      INCLUDE 'COM./NOLIST'
C
C
C WRITE ARRAY CHARACTER DIAGNOSTICS
C
      IF (ISTEP .EQ. 1) THEN
C             CALL WAC(CRD,NRP2,NZP1,2,'CRD%')
C             CALL WAC(CRE,NRP2,NZP1,2,'CRE%')
C             CALL WAC(CRF,NRP2,NZP1,2,'CRF%')
C             CALL WAC(CRG,NRP2,NZP1,2,'CRG%')
C             CALL WAC(CZD,NRP2,NZP1,2,'CZD%')
C             CALL WAC(CZE,NRP2,NZP1,2,'CZE%')
C             CALL WAC(CZF,NRP2,NZP1,2,'CZF%')
C             CALL WAC(CZG,NRP2,NZP1,2,'CZG%')
      ENDIF
C
C
C CLEAR CF AND CA, THEN SET CA-DIAGONAL TO VBT
C
      DO 17 K = 1, NZ
      DO 17 I = 1, NR
C
      DO 5 J = 1, NR
5     CF(J,I,K) = 0
C
      DO 10 J = 1, NCB
10    CA(J,I,K) = 0.
C
      CA(NRP2,I,K) = CVBTRH(I+1)*CVBTZH(K+1)
     1       * (1+RFINTH(I+1)*DEDZTH(K+1))
C
C DO MOVING-MESH TERMS, BOTH DC AND DF, IN CA AND CF.
C
      CFF = - RDRB2H(I+1) * ETAH(K+1) * BTF(I+1)
C
C SPECIAL REPRESENTATIONS FOR DKC2 AND ITS CHANGE
C THE REPRESENTATION NEXT TO THE INTERFACE INCLUDES CINTMH
C
      DKC2 = C(I+1,K+2) - C(I+1,K)
C
      DO 16 KE = -1, 1, 2
C
      IF (K .EQ. 1  .AND.  KE .EQ. -1) THEN
            RKE = - CTMULT(I+1)
            KEE = 0
      ELSE IF (K .EQ. NZ) THEN
            KEE = MIN(KE,0)
            DKC2 = 2*CINTMH(I+1) - (C(I+1,K+1) + C(I+1,K))
            CA(NRP2+NR,I,K) =
     1      CA(NRP2+NR,I,K) + CFF * BETCA * 2 * DFINTH(I+1)
            RKE = -1
```

```
590           ELSE
600                   RKE = KE
610                   KEE = KE
620           ENDIF
630   C
640   16       CA(NRP2+KEE*NR,I,K) =
650           1CA(NRP2+KEE*NR,I,K) + CFF * BETCA * RKE * DFINTH(I+1)
660   C
670   17      CF(I,I,K) = CFF * DKC2
680   C
690   C DO I-FLUX TERMS BY LOOPING THROUGH INTERMEDIATE WHOLE POINTS,
700   C INCREMENTING 6 TERMS EACH, FOR APPLICATION POINTS TO LEFT AND RIGHT.
710   C THERE ARE NO FLUXES ON THE SIDES
720   C
730           DO 25 K = 1, NZ
740           DO 25 I = 2, NR
750   C
760   C LEFT AND RIGHT FROM W TO APPLY FLUX
770   C
780           DO 25 ID = 1, 0, -1
790           IDM = ID + ID -1
800   C
810   C LEFT AND RIGHT FROM W TO GET FLUX
820   C
830           DO 25 IE = -1, 0
840           IEM = IE+IE+1
850   C
860            CA(NRP2+IE+ID,I-ID,K) =
870           1CA(NRP2+IE+ID,I-ID,K) - CRD(I,K+1)*IDM*IEM + CRG(I,K+1)*IDM
880   C
890   C UP AND DOWN FOR SLOPING MESH DIFFUSION TERM
900   C
910           DO 20 KE = -1, 1, 2
920   C
930   C SPECIAL TREATMENT FOR SLOPING-MESH TERMS AT TOP AND BOTTOM
940   C INTERFACE TREATMENT IS COMPLETED BELOW
950   C
960           IF (K .EQ. 1  .AND.  KE .EQ. -1) THEN
970                   RKE = - CTMULT(I+1+IE)
980                   KEE = 0
990           ELSE IF (K .EQ. NZ  .AND.  KE .EQ. 1) THEN
1000                  RKE = -1
1010                  KEE = 0
1020          ELSE
1030                  RKE = KE
1040                  KEE = KE
1050          ENDIF
1060   20      CA(NRP2+IE+ID+KEE*NR,I-ID,K) =
1070          1CA(NRP2+IE+ID+KEE*NR,I-ID,K) + CRE(I,K+1)*IDM*RKE
1080  C
1090  C DO THE INTERFACE CHANGE TERMS IN THE SAME I, K, ID, AND IE LOOPS
1100  C NOW GO LEFT AND RIGHT TO H NEIGHBORS
1110  C
1120  25       CF(I+IE,I-ID,K) =
1130          1CF(I+IE,I-ID,K) + CRF(I,K+1) * IDM * IEM * DIDRW(I)
1140  C
1150  C DO K-FLUX TERMS BY LOOPING THROUGH INTERMEDIATE WHOLE POINTS,
1160  C INCREMENTING 6 TERMS EACH, FOR POINTS TO DOWN AND UP.
```

```
1170    C
1180            DO 35 K = 2, NZ
1190            DO 35 I = 1, NR
1200    C
1210    C DOWN AND UP FROM W TO APPLY FLUX
1220    C
1230            DO 35 KD = 1, 0, -1
1240            KDM = KD + KD -1
1250    C
1260    C DOWN AND UP FROM W TO GET FLUX
1270    C
1280            DO 30 KE = -1, 0
1290            KEM = KE+KE+1
1300    C
1310             CA(NRP2+(KE+KD)*NR,I,K-KD) =
1320            1CA(NRP2+(KE+KD)*NR,I,K-KD) - CZD(I+1,K)*KDM*KEM+CZG(I+1,K)*KDM
1330    C
1340    C LEFT AND RIGHT FOR SLOPING MESH DIFFUSION TERM
1350    C
1360            DO 30 IE = -1, 1, 2
1370    C
1380    C AXIS SYMMETRY AND ONE-SIDED DIFFERENCE AT RIGHT EDGE USED IN GETTING C
1390    C
1400            IF (I .EQ. NR) THEN
1410                    IEE = MIN(0,IE)
1420                    RIE = 2*IE
1430            ELSE IF (I .EQ. 1) THEN
1440                    IEE = MAX(0,IE)
1450                    RIE = IE
1460            ELSE
1470                    IEE = IE
1480                    RIE = IE
1490            ENDIF
1500    C
1510    30       CA(NRP2+(KE+KD)*NR+IEE,I,K-KD) =
1520            1CA(NRP2+(KE+KD)*NR+IEE,I,K-KD) + CZE(I+1,K)*KDM*RIE
1530    C
1540    C DO THE INTERFACE CHANGE TERMS IN THE SAME I, K, AND KD LOOPS
1550    C
1560            IF (I .EQ. NR) THEN
1570                    IWS = 1
1580            ELSE
1590                    IWS = 0
1600            ENDIF
1610    C
1620    C LOOP OVER TWO W POINTS (NORMALLY JUST THE NEIGHBORS)
1630    C ONLY DO IW = 0 NEAR AXIS, SINCE F IS SYMMETRIC.
1640    C
1650            DO 35 IW = IWS, MIN(IWS+1,I-1)
1660    C
1670            WM = 0.5 + IWS - 2*(IW-IWS)*IWS
1680    C
1690    C IWS = 0 GIVES (IW,WM) = (0,0.5) AND (1,0.5)
1700    C IWS = 1 GIVES (IW,WM) = (1,1.5) AND (2,-0.5)
1710    C
1720    C NOW GO LEFT AND RIGHT TO H NEIGHBORS
1730    C
1740            DO 35 IE = -1, 0
```

```
1750            IEM = IE+IE+1
1760    C
1770    35       CF(I+1-IW+IE,I,K-KD) =
1780            1CF(I+1-IW+IE,I,K-KD) + CZF(I+1,K) * KDM * IEM
1790            1                      * WM * DIDRW(I+1-IW)
1800    C
1810    C
1820    C DO DOMAIN TOP K FLUX.  NO SLOPING-MESH TERM.
1830    C
1840            DO 40 I = 1, NR
1850    40       CA(NRP2,I,1) =
1860            1CA(NRP2,I,1)
1870            1       + CZD(I+1,1)*(1.-CTMULT(I+1))
1880            1       - CZG(I+1,1)*(1.+CTMULT(I+1))
1890    C
1900    C DO INTERFACE
1910    C PUT dT COEFFICIENTS IN THE UNUSED RIGHT OF THE DIAGONAL
1920    C
1930    C EXTRA Ci TERM
1940    C
1950            DO 50 I = 2, NR
1960            DO 50 ID = 1, 0, -1
1970            RID = (ID + ID -1) * 2.
1980            DO 50 IE = -1, 0
1990    50       CA(NRP2+IE+ID+NR,I-ID,NZ) =
2000            1CA(NRP2+IE+ID+NR,I-ID,NZ) + CRE(I,NZP1) * RID
2010    C
2020    C DO Ck TERMS
2030    C
2040            DO 70 I = 1, NR
2050    C
2060             CA(NRP2,I,NZ) =
2070            1CA(NRP2,I,NZ) + 2.*CZD(I+1,NZP1)
2080             CA(NRP2+NR,I,NZ) =
2090            1CA(NRP2+NR,I,NZ) + (CZG(I+1,NZP1)-CZD(I+1,NZP1)) * 2
2100    C
2110    C LEFT AND RIGHT FOR SLOPING MESH DIFFUSION TERM
2120    C
2130            DO 60 IE = -1, 1, 2
2140    C
2150    C AXIS SYMMETRY AND ONE-SIDED DIFFERENCE AT RIGHT EDGE USED IN GETTING D
2160    C
2170            IF (I .EQ. NR) THEN
2180                    IEE = MIN(IE,0)
2190                    RIE = 2*IE
2200            ELSE IF (I .EQ. 1) THEN
2210                    IEE = MAX(IE,0)
2220                    RIE = IE
2230            ELSE
2240                    IEE = IE
2250                    RIE = IE
2260            ENDIF
2270    60       CA(NRP2+NR+IEE,I,NZ) =
2280            1CA(NRP2+NR+IEE,I,NZ) + CZE(I+1,NZP1)*2.*RIE
2290    C
2300    C DO THE INTERFACE DF TERMS IN THE SAME I LOOP
2310    C
2320            IF (I .EQ. NR) THEN
```

```
2330                    IWS = 1
2340            ELSE
2350                    IWS = 0
2360            ENDIF
2370    C
2380    C LOOP OVER TWO W POINTS (NORMALLY JUST THE NEIGHBORS)
2390    C
2400            DO 70 IW = IWS, MIN(IWS+1,I-1)
2410    C
2420            WM = 0.5 + IWS - 2*(IW-IWS)*IWS
2430    C
2440    C IWS = 0 GIVES (IW,WM) = (0,0.5) AND (1,0.5)
2450    C IWS = 1 GIVES (IW,WM) = (1,1.5) AND (2,-0.5)
2460    C
2470    C NOW GO LEFT AND RIGHT TO H NEIGHBORS
2480    C
2490            DO 70 IE = -1, 0
2500            IEM = IE+IE+1
2510    C
2520    70       CF(I+1-IW+IE,I,NZ) =
2530           1CF(I+1-IW+IE,I,NZ) + CZF(I+1,NZP1) * IEM * WM * DIDRW(I+1-IW)
2540    C
2550    C WRITE ARRAY DIAGNOSTIC
2560    C
2570            IF (ISTEP .LE.  1   .AND.  NCB .LE. 67) THEN
2580    C               CALL WAC(CA,NCB,NR*NZ,2,'CA BEFORE%')
2590    C               CALL WAC(CF,NR,NR*NZ,2,'CF BEFORE%')
2600            ENDIF
2610    C
2620    C DOWNWARD L-U SWEEP (DECOMPOSITION)
2630    C
2640            DO 80 I = 1, NR*(NZ-1)
2650            DO 80 J = I+1,MIN(NR*NZ,NRP1+I)
2660            L = NRP2 - J + I
2670            AP = CA(L,J,1)/CA(NRP2,I,1)
2680            CA(L,J,1) = AP
2690            IF (AP .EQ. 0.) GO TO 80
2700            DO 75 K = 1, NR
2710    75      CF(K,J,1) = CF(K,J,1) - AP * CF(K,I,1)
2720            DO 79 K = 1, NRP1
2730            CA(L+K,J,1) = CA(L+K,J,1) - AP * CA(NRP2+K,I,1)
2740    79      CONTINUE
2750    80      CONTINUE
2760    C
2770    C WRITE ARRAY DIAGNOSTIC
2780    C
2790            IF (ISTEP .LE.  1   .AND.  NCB .LE. 67) THEN
2800    C               CALL WAC(CA,NCB,NR*NZ,2,'CA AFTER%')
2810    C               CALL WAC(CF,NR,NR*NZ,2,'CF AFTER%')
2820            ENDIF
2830    C
2840            RETURN
68886           END
```

```
      SUBROUTINE CNC2
C
C ELIMINATE DC EXCEPT ADJACENT TO INTERFACE
C SET RIGHT HAND SIDE TERMS FOR INTERFACE MATRIX
C
      INCLUDE 'COM./NOLIST'
C
C COPY DC INTO DCNG (NO GAPS)
C
      DO 110 K = 1, NZ
      DO 110 I = 1, NR
110   DCNG(I,K) = DC(I+1,K+1)
C
C ELIMINATE DC FROM THE TOP DOWN TO THE INTERFACE
C
      DO 120 I = 1, NR*(NZ-1)
      DO 120 J = I+1,MIN(NR*NZ,NRP1+I)
      L = NRP2 - J + I
      DCNG(J,1) = DCNG(J,1) - CA(L,J,1) * DCNG(I,1)
120   CONTINUE
C
C SET DCU ROWS OF INTERFACE MATRIX
C
      IF (LLF) THEN
C
C MATRIX ORDER:          DTUS    DTUA    DTLS    DTLA    DF      DT
C BASE (xNR,xNRA):       0,0     1,0     1,1     2,1     2,2     3,2
C BASE (xNR,xNRC):       0,0     1,0     0,1     1,1     0,2     1,2
C
            DO 150 I = 1, NR
            J = I+NR*4+NRA*2
C
            DO 130 K = 1, NAI
130         AI(J,K) = 0.
C
C SET DCU AND DF TERMS
C
            DO 140 K = 1, NR
            AI(J,NR*4+NRA*2+K) = CA(NRP2+K-I,I,NZ)
140         AI(J,NRC*2+K) = CF(K,I,NZ)
C
C SET DT TERMS (TRIDIAGONAL)
C
            DO 150 K = MAX(I-1,1), MIN(I+1,NR), 1
150         AI(J,NR*3+NRA*2+K) = CA(NRP2+NR+K-I,I,NZ) * DCMDT(K+1)
      ENDIF
C
C SET RIGHT HAND SIDES FOR INTERFACE MATRIX
C
      DO 160 I = 1, NR
160   BI(NR*4+NRA*2+I) = DCNG(I,NZ)
C
      RETURN
      END
```

```
 10            SUBROUTINE CNC3
 20    C
 30    C BAND BACK SUBSTITUTION
 40    C
 50            INCLUDE 'COM./NOLIST'
 60    C
 70    C DC IS ALREADY CORRECT ON THE BOTTOM ROW (NZ)
 80    C
 90            DO 20 I = NR*(NZ-1), 1, -1
100            DO 10 J = 1, MIN(NRP1, NR*NZ-I)
110    10      DCNG(I,1) = DCNG(I,1) - CA(NRP2+J,I,1) * DCNG(I+J,1)
120            DO 15 J = 1, NR
130    15      DCNG(I,1) = DCNG(I,1) - CF(J,I,1) * DFINTH(J+1)
140    20      DCNG(I,1) = DCNG(I,1) / CA(NRP2,I,1)
150    C
160    C RESTORE DCNG TO DC
170    C
180            DO 30 K = 2, NZP1
190            DO 30 I = 2, NRP1
200    30      DC(I,K) = DCNG(I-1,K-1)
210    C
220            RETURN
230            END
```

```
      SUBROUTINE CNT1
C
C SET UP AND L-U DECOMPOSE THE BANDED MATRIX TA
C SET UP INTERFACE MATRIX TF AND INCLUDE IN THE DECOMPOSITION
C
      INCLUDE 'COM./NOLIST'
C
C
C WRITE ARRAY CHARACTER DIAGNOSTICS
C
      IF (ISTEP .EQ. 1) THEN
C             CALL WAC(TRD,NRCP2,NZCP1,2,'TRD%')
C             CALL WAC(TRE,NRP2,NZCP1,2,'TRE%')
C             CALL WAC(TRF,NRP2,NZCP1,2,'TRF%')
C             CALL WAC(TRG,NRP2,NZP1,2,'TRG%')
C             CALL WAC(TZD,NRCP2,NZCP1,2,'TZD%')
C             CALL WAC(TZE,NRP2,NZCP1,2,'TZE%')
C             CALL WAC(TZF,NRP2,NZCP1,2,'TZF%')
C             CALL WAC(TZG,NRCP2,NZCP1,2,'TZG%')
      ENDIF
C
C SET TA(J,I,K) AND TF(J,I,K) FOR HH POINTS (I+1,K+1)
C
C
C CLEAR TF AND TA
C DO MOVING-MESH TERMS, BOTH TA AND TF
C SET TA-DIAGONAL TO TVBT
C
      DO 90 K = 1, NZC
      DO 90 I = 1, NRC
C
      DO 50 J = 1, NR
50    TF(J,I,K) = 0
C
      DO 60 J = 1, NTB
60    TA(J,I,K) = 0.
C
C
C SPECIAL REPRESENTATIONS FOR DKT2 AND ITS CHANGE
C THE REPRESENTATION NEXT TO THE INTERFACE INCLUDES TINTH
C
      DKT2 = T(I+1,K+2) - T(I+1,K)
      TFF = - RDRB2H(I+1) * ETAH(K+1) * CP(I+1,K+1)
      IF (I .LE. NR) THEN
            TFF = TFF * BTF(I+1)
      ELSE
            TFF = TFF * BTF(NRP1)
      ENDIF
C
      DO 70 KE = -1, 1, 2
C
      IF (K .EQ. 1) THEN
            RKE = 1
            KEE = MAX(KE,0)
      ELSE IF (K .EQ. NZC) THEN
            RKE = -1
            KEE = MIN(KE,0)
      ELSE IF (K .EQ. NZ) THEN
```

```
590                     KEE = MIN(KE,0)
600                     IF (I .GT. NR) THEN
610                             DKT2 = (T(I+1,K+1) - T(I+1,K)) *2
620                             RKE = KE*2
630                     ELSE
640                             DKT2 = 2*TINTH(I+1) - (T(I+1,K+1) + T(I+1,K))
650                             TA(NRCP2+NRC,I,K) =
660             1               TA(NRCP2+NRC,I,K) + TFF * BETTA * DFINTH(I+1)
670     C
680     C TA IS INCREMENTED TWICE, BOTH HERE AND BELOW, SO A FACTOR 2 IS OMITTED
690     C
700                             RKE = -1
710                     ENDIF
720             ELSE IF (K .EQ. NZP1) THEN
730                     KEE = MAX(KE,0)
740                     IF (I .GT. NR) THEN
750                             DKT2 = (T(I+1,K+2) - T(I+1,K+1)) *2
760                             RKE = KE*2
770                     ELSE
780                             DKT2 = (T(I+1,K+2) + T(I+1,K+1))- 2*TINTH(I+1)
790                             TA(NRCP2-NRC,I,K) =
800             1               TA(NRCP2-NRC,I,K) - TFF * BETTA * DFINTH(I+1)
810                             RKE = 1
820                     ENDIF
830             ELSE
840                     RKE = KE
850                     KEE = KE
860             ENDIF
870     C
880     70       TA(NRCP2+KEE*NRC,I,K) =
890             1TA(NRCP2+KEE*NRC,I,K) + TFF * BETTA * RKE * DFINTH(I+1)
900     C
910             TFT = TFF * DKT2
920     C
930             IF (I .LE. NR) THEN
940                     TF(I,I,K) = TFT
950             ELSE
960                     DO 80 J = 1, 3
970     80              TF(NRP1-J,I,K) = TFT * FABC(J)
980             ENDIF
990     C
1000    C VOLUME OVER TIME STEP ON THE DIAGONAL
1010    C
1020            VBT = TVBTRH(I+1)*TVBTZH(K+1)
1030            1       * (1+RFINTH(I+1)*DEDZTH(K+1)) * CP(I+1,K+1)
1040    C
1050    C CORRECT VBT IN CRYSTAL
1060    C
1070            IF (K .GE. NZP1  .AND.  I .LE. NR) VBT = VBT * TAUT / TAUS
1080    C
1090    90       TA(NRCP2,I,K) =
1100            1TA(NRCP2,I,K) + VBT
1110    C
1120    C TF WRONG?
1130    C
1140            IF (ISTEP .EQ. 1) THEN
1150    C                CALL WAC(TF,NR,NRC*NZC,2,'TF BEFORE SLOPE TERMS%')
1160            ENDIF
```

```
1170    C
1180    C
1190    C
1200    C                          DO I-FLUX TERMS
1210    C
1220    C
1230    C LOOP THROUGH INTERMEDIATE WHOLE POINTS,
1240    C INCREMENTING 6 TERMS EACH, FOR POINTS TO LEFT AND RIGHT.
1250    C THERE IS NO FLUX ON THE LEFT
1260    C
1270            DO 130 K = 1, NZC
1280            DO 130 I = 2, NRC
1290    C
1300    C LEFT AND RIGHT FROM W TO APPLY FLUX
1310    C
1320            DO 130 ID = 1, 0, -1
1330            IDM = ID + ID -1
1340    C
1350    C DO THE TEMPERATURE CHANGE TERMS
1360    C
1370    C
1380    C LEFT AND RIGHT FROM W TO GET FLUX
1390    C
1400            DO 110 IE = -1, 0
1410            IEM = IE+IE+1
1420    C
1430             TA(NRCP2+IE+ID,I-ID,K) =
1440            1TA(NRCP2+IE+ID,I-ID,K)
1450            1       - TRD(I,K+1)*IDM*IEM
1460            IF (LU  .AND.  K .LE. NZ  .AND.  I .LE. NR)
1470            1       TA(NRCP2+IE+ID,I-ID,K) =
1480            1       TA(NRCP2+IE+ID,I-ID,K)
1490            1                + TRG(I,K+1)*IDM
1500    C
1510    C NO SLOPING-MESH DIFFUSION TERMS WITHIN THE AMPOULE
1520    C
1530            IF (I .GT. NRP1) GO TO 110
1540    C
1550    C UP AND DOWN FOR SLOPING MESH DIFFUSION TERM
1560    C
1570            DO 100 KE = -1, 1, 2
1580    C
1590    C SPECIAL TREATMENT FOR SLOPING-MESH TERMS AT TOP AND BOTTOM (T IMPOSED)
1600    C SPECIAL INTERFACE TREATMENT IS COMPLETED BELOW
1610    C
1620            IF (K .EQ. 1) THEN
1630                    RKE = 1
1640                    KEE = MAX(KE,0)
1650            ELSE IF (K .EQ. NZC) THEN
1660                    RKE = -1
1670                    KEE = MIN(KE,0)
1680            ELSE IF (K .EQ. NZ) THEN
1690                    IF (I .EQ. NRP1) THEN
1700                            RKE = KE*2
1710                            KEE = MIN(KE,0)
1720                    ELSE
1730                            RKE = -1
1740                            KEE = MIN(KE,0)
```

```
1750                    ENDIF
1760            ELSE IF (K .EQ. NZP1) THEN
1770                    IF (I .EQ. NRP1) THEN
1780                            RKE = KE*2
1790                            KEE = MAX(KE,0)
1800                    ELSE
1810                            RKE = 1
1820                            KEE = MAX(KE,0)
1830                    ENDIF
1840            ELSE
1850                    RKE = KE
1860                    KEE = KE
1870            ENDIF
1880    C
1890             TA(NRCP2+IE+ID+KEE*NRC,I-ID,K) =
1900            1TA(NRCP2+IE+ID+KEE*NRC,I-ID,K) + TRE(I,K+1)*IDM*RKE
1910    100     CONTINUE
1920    110     CONTINUE
1930    C
1940    C DO THE INTERFACE CHANGE TERMS IN THE SAME I, K, AND ID LOOPS
1950    C
1960            IF (I .GT. NRP1) THEN
1970                    GO TO 130
1980            ELSE IF (I .EQ. NRP1) THEN
1990                    NW = 1
2000            ELSE
2010                    NW = 0
2020            ENDIF
2030    C
2040    C LOOP OVER W POINTS (NORMALLY JUST THE SAME POINT)
2050    C
2060            DO 120 JW = 0, NW
2070    C
2080            IW = JW + NW
2090            WM = NW + 1 - 3*JW
2100    C
2110    C NW = 0 GIVES (JW,IW,WM) = (0,0,1)
2120    C NW = 1 GIVES (JW,IW,WM) = (0,1,2) AND (1,2,-1)
2130    C
2140    C NOW GO LEFT AND RIGHT TO H NEIGHBORS
2150    C
2160            DO 120 IE = -1, 0
2170            IEM = IE+IE+1
2180    C
2190    120      TF(I-IW+IE,I-ID,K) =
2200            1TF(I-IW+IE,I-ID,K) + TRF(I,K+1) * IDM * IEM * WM * DIDRW(I-IW)
2210    130     CONTINUE
2220    C
2230    C
2240    C                               DIFFUSION AT AMPOULE BOUNDARY
2250    C
2260            DO 140 K = 1, NZC
2270    140     TA(NRCP2,NRC,K) = TA(NRCP2,NRC,K)
2280            1                       + TRD(NRCP1,K+1) * (1. - TRBM(K+1))
2290    C
2300    C
2310    C                               DO K-FLUX TERMS
2320    C
```

```
2330    C LOOP THROUGH INTERMEDIATE WHOLE POINTS,
2340    C INCREMENTING 6 TERMS EACH, FOR POINTS TO DOWN AND UP.
2350    C INTERFACE IS DONE SEPARATELY
2360    C
2370            DO 180 K = 2, NZC
2380            IF (K .EQ. NZP1) THEN
2390                    IB = NRP1
2400            ELSE
2410                    IB = 1
2420            ENDIF
2430            DO 180 I = IB, NRC
2440    C
2450    C DOWN AND UP FROM W TO APPLY FLUX
2460    C
2470            DO 180 KD = 1, 0, -1
2480            KDM = KD + KD -1
2490    C
2500    C DOWN AND UP FROM W TO GET FLUX
2510    C
2520            DO 160 KE = -1, 0
2530            KEM = KE+KE+1
2540    C
2550             TA(NRCP2+(KE+KD)*NRC,I,K-KD) =
2560            1TA(NRCP2+(KE+KD)*NRC,I,K-KD) - TZD(I+1,K)*KDM*KEM
2570            1                               + TZG(I+1,K)*KDM
2580    C
2590    C NO SLOPING-MESH DIFFUSION TERMS IN THE AMPOULE
2600    C
2610            IF (I .GT. NR) GO TO 160
2620    C
2630    C LEFT AND RIGHT FOR SLOPING MESH DIFFUSION TERM
2640    C
2650            DO 150 IE = -1, 1, 2
2660    C
2670    C AXIS SYMMETRY USED IN GETTING Ck
2680    C ONE-SIDED DIFFERENCE NEXT TO THE AMPOULE
2690    C
2700            IF (I .EQ. 1  .AND.  IE .EQ. -1) THEN
2710                    IEE = 0
2720                    RIE = IE
2730            ELSE IF (I .EQ. NR) THEN
2740                    IEE = MIN(0,IE)
2750                    RIE = 2*IE
2760            ELSE
2770                    IEE = IE
2780                    RIE = IE
2790            ENDIF
2800     150     TA(NRCP2+(KE+KD)*NRC+IEE,I,K-KD) =
2810            1TA(NRCP2+(KE+KD)*NRC+IEE,I,K-KD) + TZE(I+1,K)*KDM*RIE
2820     160    CONTINUE
2830    C
2840    C DO THE INTERFACE CHANGE TERMS IN THE SAME I, K, AND KD LOOPS
2850    C
2860            IF (I .GT. NR) THEN
2870                    GO TO 180
2880            ELSE IF (I .EQ. NR) THEN
2890                    IWS = 1
2900            ELSE
```

```
2910                    IWS = 0
2920            ENDIF
2930    C
2940    C LOOP OVER TWO W POINTS (NORMALLY JUST THE NEIGHBORS)
2950    C
2960            DO 170 IW = IWS, MIN(IWS+1,I-1)
2970    C
2980            WM = 0.5 + IWS - 2*(IW-IWS)*IWS
2990    C
3000    C IWS = 0 GIVES (IW,WM) = (0,0.5) AND (1,0.5)
3010    C IWS = 1 GIVES (IW,WM) = (1,1.5) AND (2,-0.5)
3020    C
3030    C NOW GO LEFT AND RIGHT TO H NEIGHBORS
3040    C
3050            DO 170 IE = -1, 0
3060            IEM = IE+IE+1
3070    C
3080    170      TF(I+1-IW+IE,I,K-KD) =
3090            1TF(I+1-IW+IE,I,K-KD) + TZF(I+1,K) * KDM * IEM *
3100            1                       WM * DIDRW(I+1-IW)
3110    180     CONTINUE
3120    C
3130    C
3140    C                       DO DOMAIN TOP AND BOTTOM
3150    C
3160            DO 190 I = 1, NRC
3170            TA(NRCP2,I,NZC) = TA(NRCP2,I,NZC) + 2*TZD(I+1,NZCP1)
3180    190     TA(NRCP2,I,  1) = TA(NRCP2,I,  1) + 2*TZD(I+1,1)
3190    C
3200    C
3210    C                       DO INTERFACE
3220    C
3230    C PUT dTINT COEFFICIENTS TO THE RIGHT OF THE DIAGONAL FOR DTSU EQUATIONS
3240    C PUT dTINT COEFFICIENTS TO THE LEFT  OF THE DIAGONAL FOR DTSL EQUATIONS
3250    C
3260    C EXTRA Ci TERM
3270    C
3280            DO 200 I = 2, NR
3290            DO 200 ID = 1, 0, -1
3300            RID = (ID + ID -1) * 2.
3310            DO 200 IE = -1, 0
3320             TA(NRCP2+IE+ID+NRC,I-ID,NZ  ) =
3330            1TA(NRCP2+IE+ID+NRC,I-ID,NZ  ) + TRE(I,NZP1) * RID
3340    200      TA(NRCP2+IE+ID-NRC,I-ID,NZP1) =
3350            1TA(NRCP2+IE+ID-NRC,I-ID,NZP1) - TRE(I,NZP2) * RID
3360    C
3370    C DO Ck TERMS
3380    C TZ%(I,NZP1) REFERS TO THE MELT. THE CRYSTAL TERMS ARE IN TZIC%(I).
3390    C
3400            DO 218 I = 1, NR
3410    C
3420             TA(NRCP2,I,NZ) =
3430            1TA(NRCP2,I,NZ) + TZD(I+1,NZP1)*2.
3440    C
3450             TA(NRCP2+NRC,I,NZ) =
3460            1TA(NRCP2+NRC,I,NZ) + (TZG(I+1,NZP1) - TZD(I+1,NZP1))*2.
3470    C
3480             TA(NRCP2,I,NZP1) =
```

```
3490            1TA(NRCP2,I,NZP1) + TZICD(I+1)*2.
3500    C
3510             TA(NRCP2-NRC,I,NZP1) =
3520            1TA(NRCP2-NRC,I,NZP1) - (TZICG(I+1) + TZICD(I+1))*2.
3530    C
3540    C LEFT AND RIGHT FOR SLOPING MESH DIFFUSION TERM
3550    C
3560            DO 210 IE = -1, 1, 2
3570    C
3580    C AXIS SYMMETRY AND ONE-SIDED DIFFERENCE AT RIGHT EDGE USED IN GETTING D
3590    C
3600            IF (I .EQ. NR) THEN
3610                    RIE = 2*IE
3620                    IEE = MIN(0,IE)
3630            ELSE IF (I .EQ. 1  .AND.  IE .EQ. -1) THEN
3640                    IEE = 0
3650                    RIE = IE
3660            ELSE
3670                    IEE = IE
3680                    RIE = IE
3690            ENDIF
3700    C
3710             TA(NRCP2+NRC+IEE,I,NZ) =
3720            1TA(NRCP2+NRC+IEE,I,NZ) + TZE(I+1,NZP1)*2.*RIE
3730    210      TA(NRCP2-NRC+IEE,I,NZP1) =
3740           - 1TA(NRCP2-NRC+IEE,I,NZP1) + TZICE(I+1)*2.*RIE
3750    C
3760    C DO THE INTERFACE DF TERMS IN THE SAME I LOOP
3770    C
3780            IF (I .EQ. NR) THEN
3790                    IWS = 1
3800            ELSE
3810                    IWS = 0
3820            ENDIF
3830    C
3840    C LOOP OVER TWO W POINTS (NORMALLY JUST THE NEIGHBORS)
3850    C
3860            DO 218 IW = IWS, MIN(IWS+1,I-1)
3870    C
3880            WM = 0.5 + IWS - 2*(IW-IWS)*IWS
3890    C
3900    C IWS = 0 GIVES (IW,WM) = (0,0.5) AND (1,0.5)
3910    C IWS = 1 GIVES (IW,WM) = (1,1.5) AND (2,-0.5)
3920    C
3930    C NOW GO LEFT AND RIGHT TO H NEIGHBORS
3940    C
3950            DO 218 IE = -1, 0
3960            IEM = IE+IE+1
3970    C
3980             TF(I+1-IW+IE,I,NZ) =
3990            1TF(I+1-IW+IE,I,NZ) + TZF(I+1,NZP1) * IEM * WM * DIDRW(I+1-IW)
4000    218      TF(I+1-IW+IE,I,NZP1) =
4010            1TF(I+1-IW+IE,I,NZP1) - TZICF(I+1) * IEM * WM * DIDRW(I+1-IW)
4020    C WRITE ARRAY DIAGNOSTIC
4030    C
4040            IF (ISTEP .LE.  1   .AND.  NTB .LE. 67) THEN
4050    C               CALL WAC(TA,NTB,NRC*NZC,2,'TA BEFORE%')
4060    C               CALL WAC(TF,NR,NRC*NZC,2,'TF BEFORE%')
```

```
4070            ENDIF
4080    C
4090    C DOWNWARD L-U SWEEP (DECOMPOSITION)
4100    C
4110            DO 240 I = 1, NRC*(NZ-1)
4120            DO 240 J = I+1, MIN(NRC*NZ,NRCP1+I)
4130            L = NRCP2 - J + I
4140            AP = TA(L,J,1)/TA(NRCP2,I,1)
4150            TA(L,J,1) = AP
4160            IF (AP .EQ. 0.) GO TO 240
4170            DO 220 K = 1, NR
4180    220     TF(K,J,1) = TF(K,J,1) - AP * TF(K,I,1)
4190            DO 230 K = 1, NRCP1
4200            TA(L+K,J,1) = TA(L+K,J,1) - AP * TA(NRCP2+K,I,1)
4210    230     CONTINUE
4220    240     CONTINUE
4230    C
4240    C UPWARD L-U SWEEP (DECOMPOSITION)
4250    C
4260            DO 270 I = NRC*NZC, NRC*NZP1+1, -1
4270            DO 270 J = I-1, MAX(NRC*NZ+1,I-NRCP1), -1
4280            L = NRCP2 + I - J
4290            AP = TA(L,J,1)/TA(NRCP2,I,1)
4300            TA(L,J,1) = AP
4310            IF (AP .EQ. 0.) GO TO 270
4320            DO 250 K = 1, NR
4330    250     TF(K,J,1) = TF(K,J,1) - AP * TF(K,I,1)
4340            DO 260 K = 1, NRCP1
4350            TA(L-K,J,1) = TA(L-K,J,1) - AP * TA(NRCP2-K,I,1)
4360    260     CONTINUE
4370    270     CONTINUE
4380    C
4390    C WRITE ARRAY DIAGNOSTIC
4400    C
4410            IF (ISTEP .LE.  1   .AND.  NTB .LE. 67) THEN
4420    C               CALL WAC(TA,NTB,NRC*NZC,2,'TA AFTER%')
4430    C               CALL WAC(TF,NR,NRC*NZC,2,'TF AFTER%')
4440            ENDIF
4450    C
4460            RETURN
65536           END
```

```
      SUBROUTINE CNT2
C
C ELIMINATE DT EXCEPT ADJACENT TO INTERFACE
C SET RIGHT HAND SIDE TERMS FOR INTERFACE MATRIX
C
      INCLUDE 'COM./NOLIST'
C
C COPY DT INTO DTNG (NO GAPS)
C
      DO 280 K = 1, NZC
      DO 280 I = 1, NRC
280   DTNG(I,K) = DT(I+1,K+1)
C
C WRITE DT ARRAY DIAGNOSTIC
C
      IF (ISTEP .LE.  1   .AND.  NRC .LE. 67) THEN
C             CALL WAC(DTNG,NRC,NZC,2,'DT BEFORE%')
      ENDIF
C
C ELIMINATE DT FROM THE TOP DOWN TO THE INTERFACE
C
      DO 290 I = 1, NRC*(NZ-1)
      DO 290 J = I+1, MIN(NRC*NZ,NRCP1+I)
      L = NRCP2 - J + I
      DTNG(J,1) = DTNG(J,1) - TA(L,J,1) * DTNG(I,1)
290   CONTINUE
C
C ELIMINATE DT FROM THE BOTTOM UP TO THE INTERFACE
C
      DO 300 I = NRC*NZC, NRC*NZP1+1, -1
      DO 300 J = I-1, MAX(NRC*NZ+1,I-NRCP1), -1
      L = NRCP2 + I - J
      DTNG(J,1) = DTNG(J,1) - TA(L,J,1) * DTNG(I,1)
300   CONTINUE
C
C WRITE DT ARRAY DIAGNOSTIC
C
      IF (ISTEP .LE.  1   .AND.  NRC .LE. 67) THEN
C             CALL WAC(DTNG,NRC,NZC,2,'DT AFTER ELIMINATION%')
      ENDIF
C
C SET DTU ROWS OF INTERFACE MATRIX
C
      IF (LLF) THEN
C
C MATRIX ORDER:          DTUS    DTUA    DTLS    DTLA    DF      DT
C BASE (xNR,xNRA):       0,0     1,0     1,1     2,1     2,2     3,2
C BASE (xNR,xNRC):       0,0     1,0     0,1     1,1     0,2     1,2
C
            DO 310 I = 1, NRC*2
            DO 310 K = 1, NAI
310         AI(I,K) = 0.
C
C SET DTU AND DF TERMS
C
            DO 330 I = 1, NRC
            DO 320 K = 1, NR
320         AI(I,NRC*2+K) = TF(K,I,NZ)
```

```
590                     DO 330 K = 1, NRC
600    330              AI(I,K) = TA(NRCP2+K-I,I,NZ)
610    C
620    C SET DT TERMS (TRIDIAGONAL) FOR DTUS
630    C
640                     DO 340 I = 1, NR
650                     DO 340 K = MAX(1,I-1), MIN(NR,I+1)
660    340              AI(I,NR+NRC*2+K) = TA(NRCP2+NRC+K-I,I,NZ)
670    C
680    C SET DTLA TERMS (DIAGONAL) FOR DTUA
690    C
700                     DO 350 I = NRP1, NRC
710    350              AI(I,NRC+I) = TA(NRCP2+NRC,I,NZ)
720    C
730    C SET DTL AND DF TERMS
740    C
750                     DO 370 I = 1, NRC
760                     DO 360 K = 1, NR
770    360              AI(NRC+I,NRC*2+K) = TF(K,I,NZP1)
780                     DO 370 K = 1, NRC
790    370              AI(NRC+I,NRC+K) = TA(NRCP2+K-I,I,NZP1)
800    C
810    C SET DT TERMS (TRIDIAGONAL) FOR DTAS
820    C
830                     DO 380 I = 1, NR
840                     DO 380 K = MAX(1,I-1), MIN(NR,I+1)
850    380              AI(NRC+I,NR+NRC*2+K) = TA(NRCP2-NRC+K-I,I,NZP1)
860    C
870    C SET DTUA TERMS (DIAGONAL) FOR DTLA
880    C
890                     DO 390 I = NRP1, NRC
900    390              AI(NRC+I,I) = TA(NRCP2-NRC,I,NZP1)
910    C
920             ENDIF
930    C
940    C SET RIGHT HAND SIDES FOR INTERFACE MATRIX
950    C
960             DO 400 I = 1, NRC*2
970    400      BI(I) = DTNG(I,NZ)
980    C
990             RETURN
65000           END
```

```
   10            SUBROUTINE CNT3
   20
   30    C
   40    C BAND BACK SUBSTITUTION
   50    C
   60            INCLUDE 'COM./NOLIST'
   70    C
   80    C DTNG IS ALREADY CORRECT ON THE INTERFACE ROWS
   90    C
  100            DO 20 K = NZ-1, 1, -1
  110            DO 20 I = NRC, 1, -1
  120            DO 10 J = 1, NRCP1
  130            IPJ = I+J
  140            L = (IPJ-1) / NRC
  150            IPJ = IPJ - L*NRC
  160            KM = K + L
  170    10      DTNG(I,K) = DTNG(I,K) - TA(NRCP2+J,I,K) * DTNG(IPJ,KM)
  180            DO 30 J = 1, NR
  190    30      DTNG(I,K) = DTNG(I,K) - TF(J,I,K) * DFINTH(J+1)
  200    20      DTNG(I,K) = DTNG(I,K) / TA(NRCP2,I,K)
  210    C
  220            DO 25 K = NZP2, NZC
  230            DO 25 I = 1, NRC
  240            DO 15 J = 1, NRCP1
  250    15      DTNG(I,K) = DTNG(I,K) - TA(NRCP2-J,I,K) * DTNG(I-J,K)
  260            DO 35 J = 1, NR
  270    35      DTNG(I,K) = DTNG(I,K) - TF(J,I,K) * DFINTH(J+1)
  280    25      DTNG(I,K) = DTNG(I,K) / TA(NRCP2,I,K)
  290    C
  300    C RESTORE DTNG TO DT
  310    C
  320            DO 95 K = 2, NZCP1
  330            DO 95 I = 2, NRCP1
  340    95      DT(I,K) = DTNG(I-1,K-1)
  350    C
  360            RETURN
65538            END
```

```
C            D              .D.              D          ...
C            D              .D.              D          ...
C  DTLA      D              .D.              D          ...
C            D              .D.              D          ...
C            D              .D               D          ...
C
C            D.              D.              D.      ..
C           .D.             .D.             .D.      ...
C           .D.             .D.             .D.       ...
C  DTC      .D.             .D.             .D.        ...
C           .D.             .D.             .D.         ...
C           .D.             .D.             .D.         ...
C
C           .D              .D.             .D          ...
C            D              .D.              D          ...
C            D              .D.              D          ...
C  DTLA      D              .D.              D          ...
C            D              .D.              D          ...
C            D              .D               D          ...
C
C            D.              D.                      ..
C           .D.             .D.                      ...
C           .D.             .D.                       ...
C  DTC      .D.             .D.                        ...
C           .D.             .D.                         ...
C           .D.             .D.                         ...
C
C           .D              .D.                         ...
C            D              .D.                         ...
C            D              .D.                         ...
C  DTLA      D              .D.                         ...
C            D              .D.                         ...
C            D              .D                          ...
C
```


PRECEDING PAGE BLANK NOT FILMED

```
      SUBROUTINE LEQ(NAIU)
C
C SOLVES SIMULTANEOUS LINEAR SYSTEM (SQUARE) NEAR INTERFACE
C SETS RESULTS IN RESULT ARRAYS
C
      INCLUDE 'COM./NOLIST'
C
      DIMENSION WK(NAI,3)
C
C
C MATRIX ORDER:          DTUS    DTUA    DTLS    DTLA    DF      DT
C BASE (xNR,xNRA):       0,0     1,0     1,1     2,1     2,2     3,2
C BASE (xNR,xNRC):       0,0     1,0     0,1     1,1     0,2     1,2
C
C SET INTERFACE RHS                                      DTFI    DCFI
C
C THE REST OF THE RHS WAS SET EARLIER BY CNT2, AND POSSIBLY CNC2.
C
      DO 10 I = 1, NR
      BI(2*NRC+I) = DTFINT(I+1)
10    BI(2*NRC+NR+I) = DCFINT(I+1)
C
C WRITE ARRAY DIAGNOSTIC BEFORE
C
      IF (ISTEP .LE.  1   .AND.  NAIU .LE. 67) THEN
            WRITE(6,'(/'' LEQ.  RIGHT HAND SIDE'')')
70          FORMAT (/1X,A,(T10,1P,10E12.3))
            WRITE(6,70) 'DTU',(BI(I),I=1,NRC)
            WRITE(6,70) 'DTL',(BI(I+NRC),I=1,NRC)
            WRITE(6,70) 'DTFI',(BI(I+2*NRC),I=1,NR)
            IF (LC) WRITE(6,70) 'DCFI',(BI(I+2*NRC+NR),I=1,NR)
            IF (LC) WRITE(6,70) 'DCU',(BI(I+2*NRC+NR*2),I=1,NR)
      ENDIF
C
C
C
C SET UP MATRIX, IF REQUIRED.
C
      IF (LLF) THEN
            CALL LEQSET(NAIU)
            IJOB = 0
      ELSE
            IJOB = 1
      ENDIF
C
C CALL SOLVER
C
      CALL LEQIF(AI,NAI,NAIU,NAIU,BI,NAI,1,IJOB,WK,IER)
C
      IF (IER .NE. 0) WRITE (6,'(/'' LEQ. IER ='',I4)') IER
      IF (IER .NE. 0) WRITE (*,'(/'' LEQ. IER ='',I4)') IER
C
C WRITE ARRAY DIAGNOSTIC AFTER
C
      IF (ISTEP .LE.  1   .AND.  NAIU .LE. 67) THEN
            IF (IJOB .EQ. 0) THEN
C           CALL WAC(AI,NAI,NAIU,1,'AI AFTER%')
            WRITE(6,'(/'' LEQ.  PIVOTS''/(1X,10F12.0))')
```

```
590           1                (WK(I,1),I=1,NAIU)
600                   WRITE(6,'(/'' LEQ.  PIVOTS''/(5X,1P,10E12.3))')
610           1                (WK(I,2),I=1,NAIU)
620                   ENDIF
630                   WRITE(6,'(/'' LEQ.  SOLUTION'')')
640                   WRITE(6,70) 'DTU',(BI(I),I=1,NRC)
650                   WRITE(6,70) 'DTL',(BI(I+NRC),I=1,NRC)
660                   WRITE(6,70) 'DF',(BI(I+2*NRC),I=1,NR)
670                   IF (LC) WRITE(6,70) 'DTINT',(BI(I+2*NRC+NR),I=1,NR)
680                   IF (LC) WRITE(6,70) 'DCU',(BI(I+2*NRC+NR*2),I=1,NR)
690           ENDIF
700   C
710   C SET RESULT ARRAYS
720   C
730           DO 200 I = 1, NRC
740           DTNG(I,NZ) = BI(I)
750   200     DTNG(I,NZP1) = BI(I+NRC)
760   C
770           DO 300 I = 1, NR
780   300     DFINTH(I+1) = BI(I+2*NRC)
790   C
800           IF (LC) THEN
810                   DO 400 I = 1, NR
820                   DTINTH(I+1) = BI(I+2*NRC+NR)
830   400             DCNG(I,NZ) = BI(I+2*NRC+2*NR)
840           ENDIF
850   C
860           RETURN
65838         END
```

```
      SUBROUTINE LEQSET(NAIU)
C
C SETS UP INTERFACE FLUX ROWS OF INTERFACE MATRIX
C
C IF NOT (LC) THEN ROWS AND COLUMNS FOR DT AND DCU ARE SET BUT NOT USED,
C       UNDER CONTROL OF THE ARGUMENT NAIU PASSED TO LEQIF.
C
      INCLUDE 'COM./NOLIST'
C
C
C MATRIX ORDER:          DTUS    DTUA    DTLS    DTLA    DF      DT
C BASE (xNR,xNRA):       0,0     1,0     1,1     2,1     2,2     3,2
C BASE (xNR,xNRC):       0,0     1,0     0,1     1,1     0,2     1,2
C
C INTERFACE RHS                                          DTFI    DCFI
C
      DO 70 I = 1, NR
C
C SET THE TFI (DF) AND CFI (DT) EQUATION ROWS TO ZERO
C
      IDF = I + 2*NRC
      IDT = I + NRC*2 + NR
C
      DO 10 K = 1, NIA
      AI(IDT,K) = 0.
10    AI(IDF,K) = 0.
C
C SET COLUMNS FOR TERMS WITH SAME I VALUE
C
      IDC = I + 2*NRC + 2*NR
      IDU = I
      IDL = I + NRC
C
      AI(IDF,IDF) = DFTINT(I+1)
      AI(IDF,IDU) = AI(IDF,IDU) + TZD(I+1,NZP1) * 2.
      AI(IDF,IDL) = AI(IDF,IDL) + TZICD(I+1) * 2.
      AI(IDF,IDT) = AI(IDF,IDT) - ( TZD(I+1,NZP1)  + TZICD(I+1) ) * 2.
      AI(IDT,IDF) = DFCINT(I+1)
      AI(IDT,IDC) = AI(IDT,IDC) + CZD(I+1,NZP1) * 2.
      AI(IDT,IDT) = AI(IDT,IDT) - 2 * CZD(I+1,NZP1) * DCMDT(I+1)
      1                         + DTCINT(I+1)
C
C
C LEFT AND RIGHT FOR SLOPING MESH DIFFUSION TERMS
C
      DO 20 IE = -1, 1, 2
C
C AXIS SYMMETRY AND ONE-SIDED DIFFERENCE AT RIGHT EDGE USED IN GETTING D
C
      IF (I .EQ. NR) THEN
            RIE = 2*IE
            IEE = MIN(0,IE)
      ELSE IF (I .EQ. 1  .AND.  IE .EQ. -1) THEN
            IEE = 0
            RIE = IE
      ELSE
            IEE = IE
            RIE = IE
```

C-3

```
590             ENDIF
600     C
610             AI(IDF,IDT+IEE) = AI(IDF,IDT+IEE)
620             1       + (TZE(I+1,NZP1) - TZICE(I+1) ) * 2.*RIE
630     20      AI(IDT,IDT+IEE) = AI(IDT,IDT+IEE)
640             1       + CZE(I+1,NZP1) * 2.*RIE * DCMDT(I+1+IEE)
650     C
660     C DO THE INTERFACE DF TERMS IN THE SAME I LOOP
670     C
680             IF (I .EQ. NR) THEN
690                     IWS = 1
700             ELSE
710                     IWS = 0
720             ENDIF
730     C
740     C LOOP OVER TWO W POINTS (NORMALLY JUST THE NEIGHBORS)
750     C
760             DO 70 IW = IWS, MIN(IWS+1,I-1)
770     C
780             WM = 0.5 + IWS - 2*(IW-IWS)*IWS
790     C
800     C IWS = 0 GIVES (IW,WM) = (0,0.5) AND (1,0.5)
810     C IWS = 1 GIVES (IW,WM) = (1,1.5) AND (2,-0.5)
820     C
830     C NOW GO LEFT AND RIGHT TO H NEIGHBORS
840     C
850             DO 70 IE = -1, 0
860             IEM = IE+IE+1
870     C
880              AI(IDT,IDF+1-IW+IE) =
890             1AI(IDT,IDF+1-IW+IE) + CZF(I+1,NZP1) * IEM * WM * DIDRW(I+1-IW)
900     70       AI(IDF,IDF+1-IW+IE) =
910             1AI(IDF,IDF+1-IW+IE) + (TZF(I+1,NZP1) - TZICF(I+1))
920             1                          * IEM * WM * DIDRW(I+1-IW)
930     C
940             IF (ISTEP .LE. 1    .AND.  NAIU .LE. 67) THEN
950     C               CALL WAC(AI,NAI,NAIU,1,'AI BEFORE%')
960             ENDIF
970     C
980             RETURN
65998           END
```

```
C
C STRUCTURE OF AI MATRIX (NR = NRA = 6)
C
C
C          DTUS     DTUA     DTLS     DTLA      DF       DT       DCU
C
C         ......   ......                      ......   ..
C         ......   ......                      ......   ...
C DTUS    ......   ......                      ......    ...
C         ......   ......                      ......     ...
C         ......   ......                      ......      ...
C         ......   ......                      ......       ..
C
C         ......   ......          .           ......
C         ......   ......            .         ......
C DTUA    ......   ......              .       ......
C         ......   ......                .     ......
C         ......   ......                  .   ......
C         ......   ......                    . ......
C
C                           ......   ......   ......   ..
C                           ......   ......   ......   ...
C DTLS                      ......   ......   ......    ...
C                           ......   ......   ......     ...
C                           ......   ......   ......      ...
C                           ......   ......   ......       ..
C
C                    .      ......   ......   ......
C                     .     ......   ......   ......
C DTLA                 .    ......   ......   ......
C                       .   ......   ......   ......
C                        .  ......   ......   ......
C                         . ......   ......   ......
C
C          .                .                 ..       ..
C           .               .                 ...      ...
C TFI        .              .                  ...      ...
C             .             .                   ...      ...
C              .            .                   ...       ..
C               .           .                   ...
C
C                                              ..       ..        .
C                                              ...      ...       .
C CFI                                           ...      ...       .
C                                               ...      ...       .
C                                                ...      ...       .
C                                                ...       ..       .
C
C                                              ......   ..       ......
C                                              ......   ...      ......
C DCU                                          ......    ...     ......
C                                              ......     ...    ......
C                                              ......      ...   ......
C                                              ......       ..   ......
```

```
   10            SUBROUTINE GVORT
   20    C
   30    C
   40            INCLUDE 'COM./NOLIST'
   50    C
   60    C
   70            RETURN
65588            END
```

```
   10            SUBROUTINE SETPSI
   20    C
   30    C SET STREAM FUNCTION USING ANALYTIC APPROXIMATE FORMULA ALLOWING FOR DE
   40    C
   50            INCLUDE 'COM./NOLIST'
   60    C
   70            DO 10 K = 1, NZP1
   80            IF (K .EQ. NZP1) THEN
   90                    PF = 1.E5
  100            ELSE
  110                    PF = RSAM/(ZETW(K)-ZINT)/(PI*PI/4-1)
  120            ENDIF
  130            PFP1 = PF+1
  140            P1 = .5*WAMP*((1-DENCBM)*(1+1/PFP1) - 1)
  150            P2 = .5*WAMP*(1-DENCBM)*(1/PFP1)
  160            DO 10 I = 1, NRP1
  170            PP = R2W(I) * ( P1 - P2 * EXP(LOG(R2W(I)/(RSAM*RSAM) ) *PFP1 ) )
  180            PSIO(I,K) = PP
  190    10      PSI (I,K) = PP
  200    C
  210            WRITE (6,'(//'' SETPSI.'',
  220           1        '' WAMP, DENCBM, RSAM, PF, P1, P2, PFP1, PP ='' 
  230           1        /(1P10E12.3))')
  240           1        WAMP, DENCBM, RSAM, PF, P1, P2, PFP1, PP
  250    C
  260            RETURN
65238              END
```

```
      SUBROUTINE PSIBC
C
C APPLY PSI BOUNDARY CONDITIONS, INCLUDING NO-SLIP.
C
      INCLUDE 'COM./NOLIST'
C
C GET AWALL
C
      IF (LU) THEN
              AWALL = 0
              DO 3 I = 2, NRP1
3             AWALL = AWALL + (1-DENCBM)*(4/RSAM/RSAM)
     1        *(WAMP+DFINTH(I)*BTF(I)) *RDRB2H(I)*2
      ELSE
              AWALL = (1-DENCBM) * 2 * WAMP
      ENDIF
C
C SYMMETRY AT THE LEFT AND RIGHT
C
      DO 1 K = 1, NZP2
      DPSI(NRP2,K) = DPSI(NR,K)
       PSI(NRP2,K) =  PSI(NR,K) - 2 * WAMP / DIDRW(NRP1)
      DPSI(NRP1,K) = 0
       PSI(NRP1,K) = (AWALL/4 - WAMP/2) * RSAM * RSAM
      DPSI(1,K) = 0
1      PSI(1,K) = 0
C
C IMPOSED AT THE TOP AND BOTTOM
C
      DO 2 I = 2, NRP1
      DPSI(I,NZP2) = DPSI(I,NZP1)
       PSI(I,NZP2) =  PSI(I,NZ)
      DPSI(I,1) = DPSI(I,2)
2      PSI(I,1) = (AWALL-WAMP) * 0.5 * R2W(I)
     1                - AWALL * .25 / (RSAM*RSAM) * R4W(I)
C
      RETURN
      END
```