LBL-20457 c.2

Center for Advanced Materials

# CAM REPORT

RECEIVED
LAWRENCE
BERKELEY LABORATORY

JUN 26 1987

LIBRARY AND
DOCUMENTS SECTION

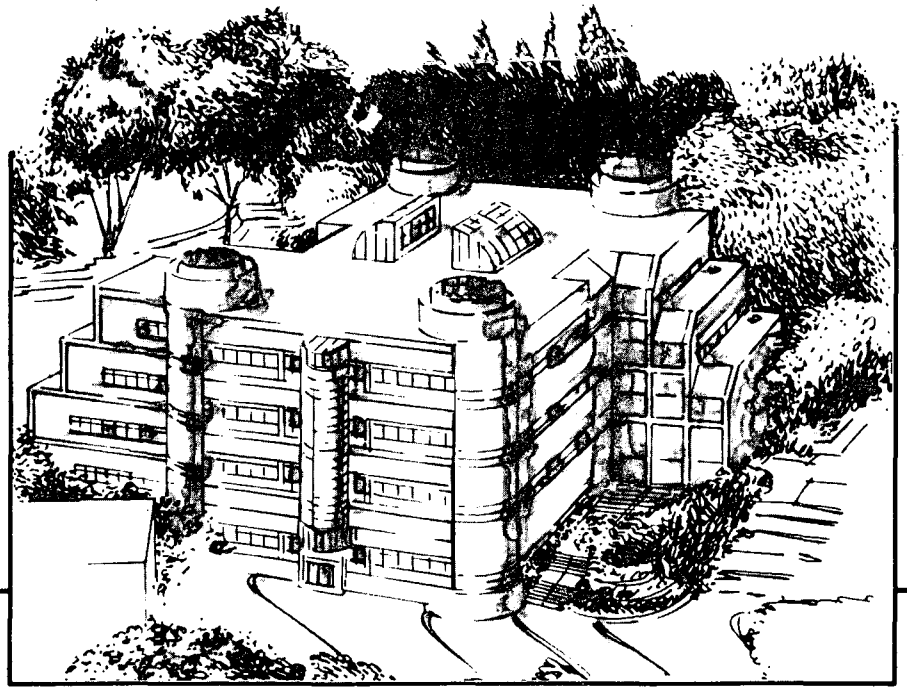
## A FINITE ELEMENT ANALYSIS OF THE SEEDING PROCESS FOR CRYSTAL GROWTH BY THE HORIZONTAL BRIDGEMAN METHOD

M. Smidebush
(M.S. Thesis)

TWO-WEEK LOAN COPY

This is a Library Circulating Copy
which may be borrowed for two weeks.

September 1985



**Materials and Chemical Sciences Division**

**Lawrence Berkeley Laboratory • University of California**

ONE CYCLOTRON ROAD, BERKELEY, CA 94720 • (415) 486-4755

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

LBL-20457 c.2

## DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

# A FINITE ELEMENT ANALYSIS OF THE SEEDING PROCESS FOR CRYSTAL GROWTH BY THE HORIZONTAL BRIDGEMAN METHOD

Michael Smidebush

Department of Mechanical Engineering &
Center for Advanced Materials
Lawrence Berkeley Laboratory
University of California
Berkeley, California

September 30, 1985

Master's Project Report

Project Advisor: Prof. Boris Rubinsky

ABSTRACT

A Front Tracking finite element code was used to study movements of the solid-liquid interface which occur during the seeding process of semiconductor crystal growth. The analysis focused on improving control of the interface through use of seed crystal " blankets ". Thermal properties of the blanket materials ranged from isotropic to strongly anisotropic. Results showed that a seed blanket can be effective in control of both the interface position and shape during the seeding process.

This work was supported by the U.S. Department of Energy under Contract Number DE-AC03-76SF00098.

TABLE OF CONTENTS


| | page |
|---|---|
| Nomenclature | i |
| List of Figures | iii |
| List of Tables | v |
| Introduction | 1 |
| Crystal Growth Process | 3 |
| Horizontal Bridgeman Method | 7 |
| Objective | 11 |
| Mathematical Model | 11 |
| Solution Algorithm | 26 |
| Results and Discussion | 27 |
| Conclusions and Recommendations | 34 |
| Appendix | 59 |

# NOMENCLATURE

$\underset{\sim}{a}$ - partial derivative wrt x, of the field variable interpolation function

$\underset{\sim}{b}$ - partial derivative wrt y, of the field variable interpolation function

$\underline{B}$ - temperature gradient matrix

$c$ - specific heat

$\underline{C}$ - capacity matrix

$ds$ - element of arc length on the solid-liquid interface

$ds_x$ - displacement of an interface node in the x-direxion

$ds_y$ - displacement of an interface node in the y-direction

$\underset{\sim}{ds}$ - vector displacement of an interface node

$\underset{\sim}{dS}$ - nodal displacement vector, in terms of field variable values

$\underset{\sim}{dx}$ - nodal point value for x-displacement of an interface node

$\underset{\sim}{dy}$ - nodal point value for y-displacement of an interface node

$\underset{\sim}{H}$ - interpolation function for nodal displacements

$\underline{J}$ - transformation matrix

$k$ - thermal conductivity

$\underline{k}$ - directional thermal conductivity

$\underline{K}$ - stiffness matrix

$L$ - latent heat of fusion

$n$ - normal vector (to the interface)

$\bar{q}$ - imposed heat flux on the boundary

$\vec{r}$ - position vector

$s$ - position of the interface

$t$ - time

$T$ - temperature

# NOMENCLATURE

$T_{PH}$ - phase change temperature

$\underset{\sim}{T}$ - nodal temperature vector

$\delta T$ - variation in the exact temperature, used as a weight function in the Galerkin method weak form

$\underline{U}$ - interpolation matrix for nodal displacements of the interface

$V_n$ - velocity of a point on the interface along the normal direction

$V_{n_x}$ - x-component of the normal velocity

$V_{n_y}$ - y-component of the normal velocity

$\underset{\sim}{x}$ - x-coordinate transformation vector

$\underset{\sim}{y}$ - y-coordinate transformation vector

$\alpha$ - angle between interface normal and the positive x-direction

$\rho$ - density

$\underset{\sim}{\phi}$ - interpolation vector

$\eta$ - weight function

$\sigma$ - domain boundary

$\underset{\sim}{\Psi}$ - coordinate interpolation vector

$\omega_i, \omega_j$ - Gaussian Quadrature weight functions

subscripts

1 - solid
2 - liquid

# LIST OF FIGURES


| | | page |
|---|---|---|
| 1. | Schematic of the Czochralski crystal growth method | 2 |
| 2. | Schematic of the Floating Zone crystal growth method | 2 |
| 3. | Crystal structures of various compounds and elements | 5 |
| 4. | Schematic of the Horizontal Bridgeman crystal growth method | 5 |
| 5. | Axial temperature profiles as a function of time for a typical Horizontal Bridgeman system | 9 |
| 6. | Physical model for the seeding process | 13 |
| 7. | Coordinate transformation from x,y space to , space | 17 |
| 8. | Two node model of the solid-liquid interface | 24 |
| 9a. | Interface shape as a function of time, with an isotropic graphite blanket, y-orientation, tbvel .5mm/hr | 36 |
| 9b. | Interface shape as a function of time, with a Boron Nitride blanket, y-orientation, tbvel .5mm/hr | 37 |
| 9c. | Interface shape as a function of time, with a pyrolytic graphite blanket, y-orientation, tbvel | 38 |
| 9d. | Interface shape as a function of time, with an ideal anisotropic blanket, y-orientation, tbvel .5mm/hr | 39 |
| 10a. | Interface shape as a function of time, with an isotropic graphite blanket, y-orientation, tbvel 5mm/hr | 40 |
| 10b. | Interface shape as a function of time, with a Boron Nitride blanket, y-orientation, tbvel 5mm/hr | 41 |
| 10c. | Interface shape as a function of time, with a pyrolytic graphite blanket, y-orientation, tbvel 5mm/hr | 42 |
| 11a. | Interface shape as a function of time, with a Boron Nitride blanket, x-orientation, tbvel .5mm/hr | 43 |
| 11b. | Interface shape as a function of time, with a pytolytic graphite blanket, x-orientation, tbvel .5mm/hr | 44 |
| 11c. | Interface shape as a function of time, with an ideal anisotropic blanket, x-orientation, tbvel .5mm/hr | 45 |
| 12a. | Interface location as a function of time, isotropic graphite blanket, tbvel .5mm/hr | 46 |

# LIST OF FIGURES


| | | page |
|---|---|---|
| 12b. | Interface locations as a function of time, Boron Nitride blanket, y-orientation, tbvel .5mm/hr | 47 |
| 12c. | Interface location as a function of time, Pyrolytic graphite blanket, y-orientation, tbvel .5mm/hr | 48 |
| 12d. | Interface location as a function of time, ideal anisotropic blanket, y-orientation, tbvel .5mm/hr | 49 |
| 13a. | Interface location as a function of time, isotropic graphite blanket, y-orientation, tbvel 5mm/hr | 50 |
| 13b. | Interface location as a function of time, Boron Nitride blanket, y-orientation, tbvel 5mm/hr | 51 |
| 13c. | Interface location as a function of time, Pyrolytic graphite blanket, y-orientation, tbvel 5mm/hr | 52 |
| 14a. | Interface location as a function of time, Boron Nitride blanket, x-orientation, tbvel .5mm/hr | 53 |
| 14b. | Interface location as a function of time, Pyrolytic graphite blanket, x-orientation, tbvel .5mm/hr | 54 |
| 14c. | Interface location as a function of time, ideal anisotropic blanket, x-orientation, tbvel .5mm/hr | 55 |
| 15. | Experimental interface shape for a seed crystal used in Horizontal Bridgeman growth of Gallium Arsenide | 56 |
| 16. | Interface shape as a function of time, Boron Nitride blanket, x-orientation, tbvel 2.5mm/hr | 57 |
| 17. | Interface location as a function of time, Boron Nitride blanket, x-orientation, tbvel 2.5mm/hr | 58 |

# LIST OF TABLES

page

1. Thermal properties of blanket materials and Gallium Arsenide 29

2. Temperature history imposed on the seed crystal during the Horizontal Bridgeman process 30

Integrated circuit technology today can put hundreds of thousands of transistors on less than a square centimeter of area. Even so the level of sophistication could be increased. However, the performance of the most carefully prepared circuit can not exceed the capability of its base material, or substrate. Progress of the semiconductor industry thus relies on improving the quality of substrates, or more fundamentally, the quality of bulk crystals the substrates come from.

Since the "silicon chip's" rise to fame nearly twenty years ago, three main methods have been used to grow bulk crystals or ingots. The methods are Czochralski (CZ), Floating Zone (FZ), and Horizontal Bridgeman (HB). The CZ method is a vertical growth technique. The crystal is formed by slowly withdrawing a seed from a molten pool of polycrystalline material. The crystal grows at the solid liquid interface with the same structure as the seed. Floating Zone is also a vertical technique. Here the crystal is formed by melting a small region of a polycrystalline rod next to a seed crystal. The molten zone is moved up the length of the rod and solidification occurs behind it. In doing so the crystal structure of the rod changes to the seed structure (see Figures 1 and 2 for schematics of these processes). Crystals formed by the HB method are grown in a boat shaped container. Solidification begins at the seed which is located in bow. More on this later.

Virtually all production grade silicon is grown using the CZ method. Standard sizes exceed ten centimeters in diameter and

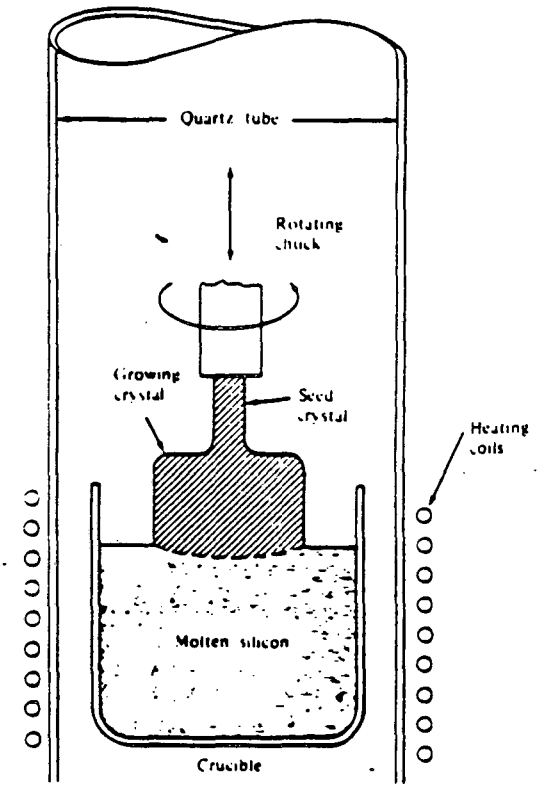



Figure 1. Schematic of the Czochralski crystal growth method

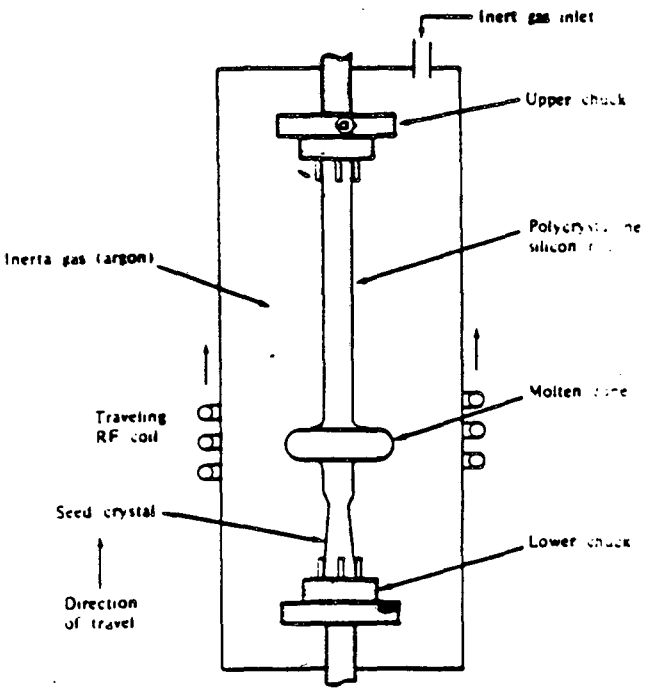



Figure 2. Schematic of the Floating Zone crystal growth method

fifty centimeters in length. In foreign markets silicon has been produced with diameters up to twenty centimeters. Gallium Arsenide (GaAs), a semiconductor material which can outperform silicon, does not have a standard growth method as yet. However, the HB method does seem to be a favorite choice.

Ideally, semiconductor crystals are grown with a minimum of defects and a uniform structure. The technical difficulties associated with these aims depend both on the crystal and the growth process. Silicon crystals for example, are prepared from a single major component and thus are easier to grow than Gallium Arsenide crystals. Similarly, a Horizontal Bridgeman furnace has relatively few moving parts compared to a Czochralski furnace. Thus the process control algorithm for the Bridgeman furnace would be simplier.

Since this work focuses on a specific aspect of the HB method it will be useful to describe it in more detail. To appreciate the process though, it is first necessary to understand more about the crystal growth process itself.

## CRYSTAL GROWTH PROCESS

A crystal is defined as an ordered periodic structure. Component atoms hold specific positions in that structure or lattice. Different lattice arrangements are described with terms such as "body centered cubic", "face centered cubic" and "hexagonal closepack". Different structures and/or lattice orientations can be present in a single material. For example, surface har-

dening of some steels is produced by rapid quenching from a high temperature to change the normal face centered cubic structure to body centered cubic. A structure typical of martensite. On the other hand, polycrystalline silicon is diamond structured throughout but has nonuniform orientation. This is evidenced on a microscale by the presence of grain boundaries. An ideal crystal of Gallium Arsenide has a zincblende structure, which is similar to diamond, with Gallium and Arsenic atoms occupying alternate positions in the lattice. Figure 3 shows representative crystal structures.

The electronic properties of a crystal (electron mobility, resistivity) are dependent on both structure and orientation. Thus crystals used in the semiconductor industry must be "single" crystals. That is the crystal must have one structure with a uniform orientation throughout. This emphasizes the importance of the seed crystal since it provides the template for both.

The seed is a highly pure single crystal with a desirable orientation as determined by x-ray diffraction. Orientations are described in terms of the Miller indices of a unit cell. Most silicon is grown with a <1,1,1> or a <1,0,0> orientation. Gallium arsenide is also grown with <1,0,0> orientation.

Mechanical properties of crystals also are influenced by the lattice structure. Of most importance is the tendency for a crystal to cleave along preferred crystallographic planes. This property must be exploited to efficiently separate individual circuits from the finished wafer. A process which is called

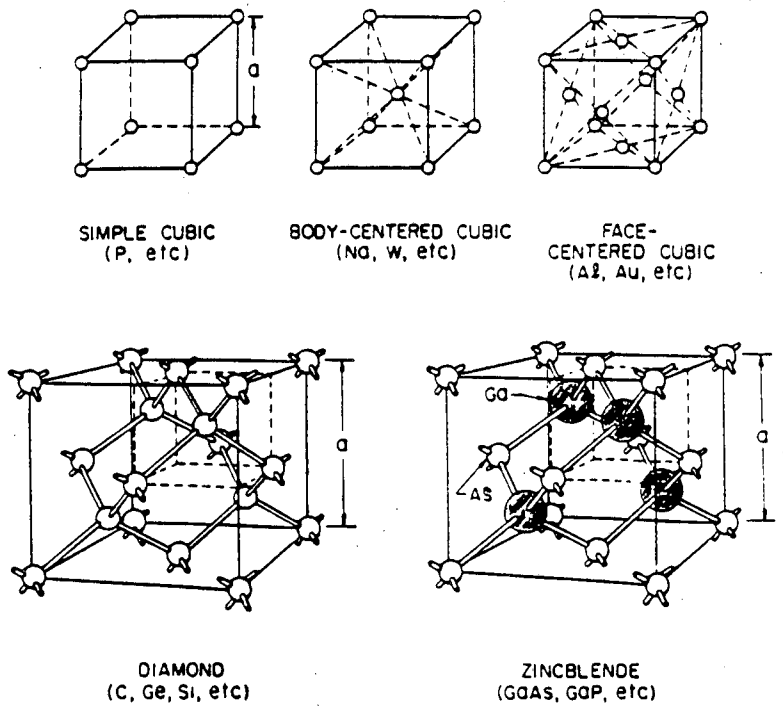



Figure 3. Crystal structures of various compounds and elements

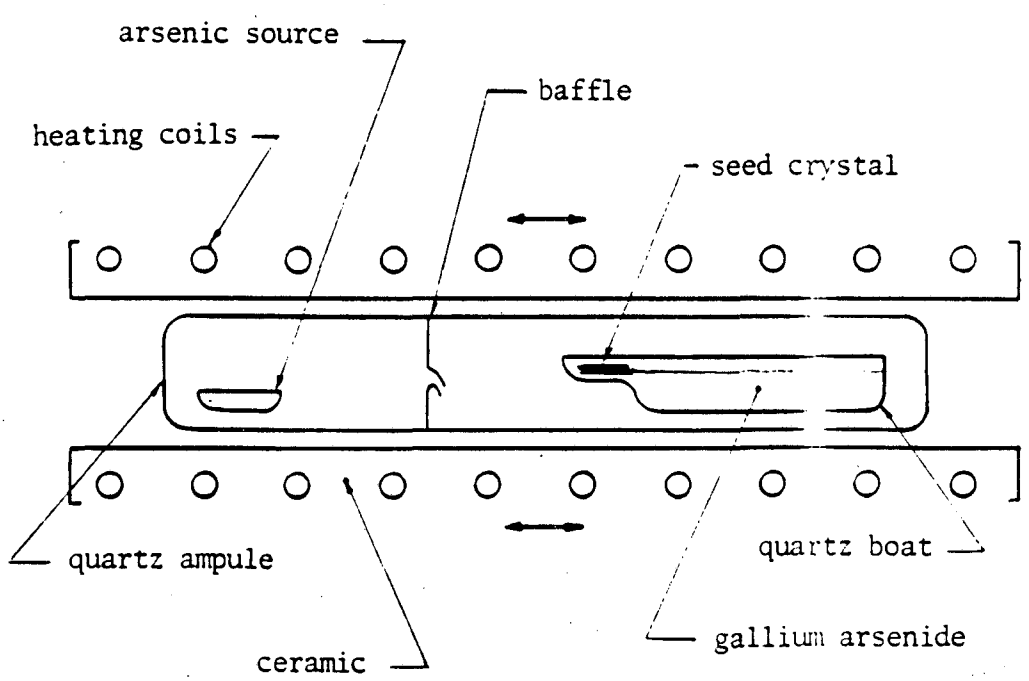



Figure 4. Schematic of the Horizontal Bridgeman crystal growth method

dicing.

The actual growth process begins by melting a charge of polycrystalline material in a special furnace. A seed is then introduced to the melt and by extracting energy the crystal begins to grow. Unfortunately, this is also when the problems begin. The rate at which energy can be extracted, which determines the crystal growth rate, depends mainly on temperature gradients in the solid and losses to surrounding media. Both of these factors are strongly influenced by the radiation environment within the furnace. Complex geometric shape factors and unknown surface properties make accurate temperature control extremely difficult. The lack of control can in turn produce non uniform growth rates with corresponding variations in crystal properties (an undesirable side effect from a circuit production standpoint).

Temperature gradients in the solid also produce thermal stress. It has been proposed that thermal stresses contribute to the formation of defects in the crystal lattice (5). Defects are known to impair the electrical performance of semiconductors. Gallium Arsenide appears to be more sensitive to this than does silicon.

Great care is taken to produce crystals which are free of defects and unknown impurities. It seems ironic then that for many applications, minute levels of known impurities (called dopants) are introduced intentionally. Dopants actually play a vital role in giving semiconductors their electrical properties.

Typical dopant concentrations are on the order of $(10)^{15}$ atoms per cubic centimeter. It is, of course, important to incorporate the dopants uniformly as the crystal solidifies.

Single crystal growth of GaAs also is faced with unique difficulties since it is a compound semiconductor. Gallium and Arsenic atoms must be present in the crystal structure in a one-to-one ratio which is accurate to better than one part in ten million. This requirement is especially difficult to achieve since Gallium and Arsenic have different vapor pressures at the melting point. Arsenic pressure is the greater, thus it volatilizes and disturbs the stoichiometry. Elaborate growth schemes have been used to prevent this from happening. Liquid Encapsulated Czochralski (LEC) is one such method.

In comparison to Silicon, Gallium Arsenide crystals are more difficult to grow. Commensurate with these difficulties, todays average Gallium Arsenide crystal is around two to five centimeters in diameter. However, the performance of Gallium Arsenide provides more than enough motivation to tackle the technical problems. A Gallium Arsenide circuit is almost ten times faster than a comparable silicon one. It consumes less power and thus requires less cooling. It also is insensitive to pulses of electromagnetic radiation, a desirable property for military applications.

## THE HORIZONTAL BRIDGEMAN METHOD

The Horizontal Bridgeman method is currently used to produce

low defect Gallium Arsenide crystals. A sketch of one typical system is shown in Figure 4. The figure shows two quartz boats inside of a quartz ampoule. One boat contains the polycrystalline charge. The other holds an Arsenic source that will provide an Arsenic rich atmosphere to control the stoichiometry in the melt. Before placement in the furnace, the ampoule is evacuated to approximately $(10)^{-7}$ torr and then sealed. At operating temperatures the vapor pressure of the Arsenic is around one atmosphere. The seed crystal is located on the furnace centerline in the neck of the boat.

The furnace supplies energy via resistance heating elements that are embedded in a ceramic material. Temperature sensing is done with shielded thermocouples. Axial temperature distributions versus time for one typical growth cycle (heat up and growth stages) are shown in Figure 5. The charge of Gallium Arsenide is melted in the first six hours of the heat-up stage. During this time the seed partially backmelts and contacts the melted charge in the body of the boat. Temperatures are held fixed for approximately five hours to let the system equilibrate. This time also allows the solid-liquid interface (in the seed region) to reach a steady shape. Optimally the shape would be planar at this stage and it would remain so throughout the process.

In the growth stage the temperature boundary profile is moved towards the liquid phase at a fixed velocity, pushing the interface with it. Both the slope of the boundary profile and the

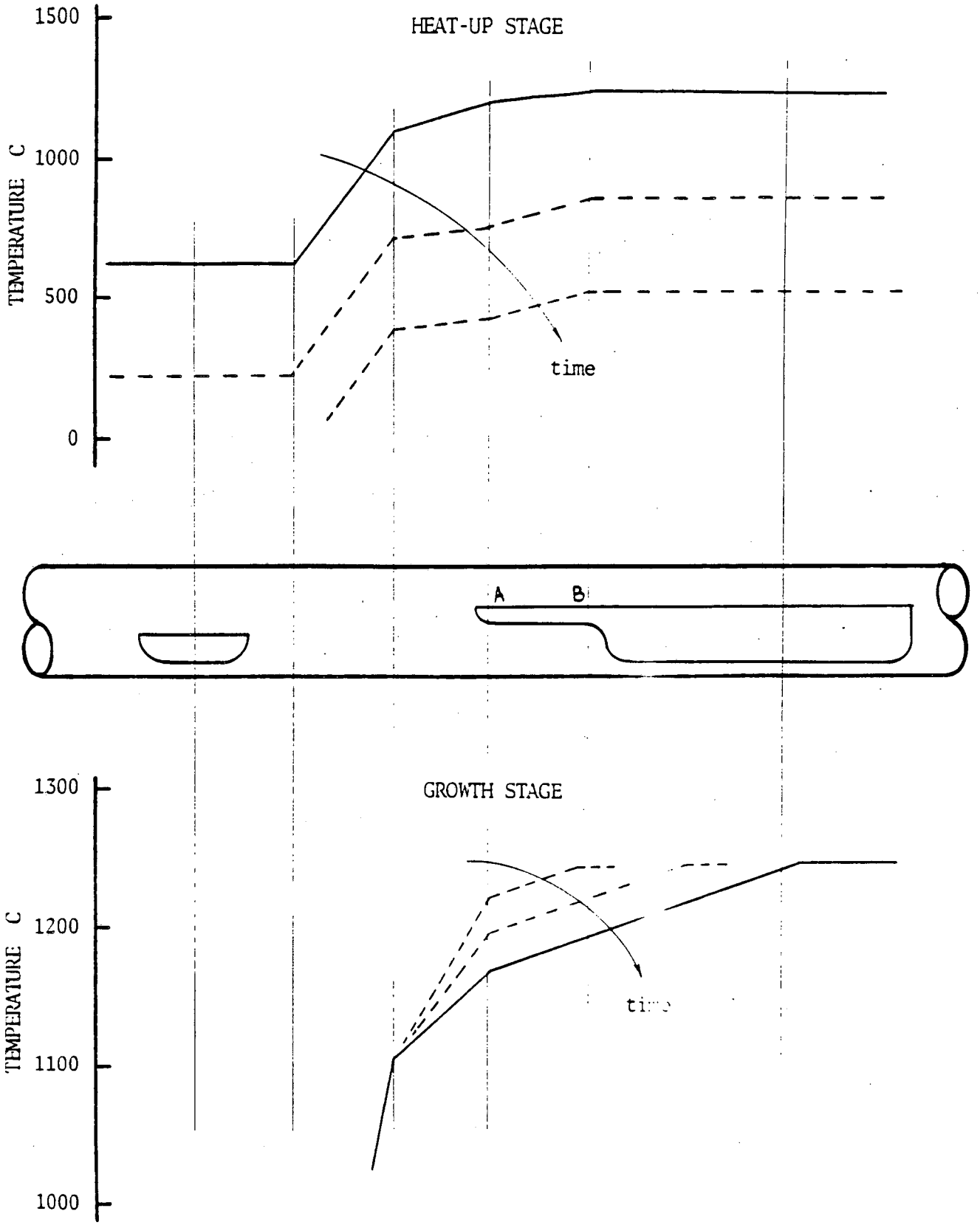



Figure 5. Axial temperature profiles as a function of time, during heat up and growth stages, for a typical Horizontal Bridgeman system (3)

speed it moves at must be constant to obtain uniform growth rates. In the time frame of the process, with temperature boundary velocities around three millimeters per hour, it typically takes seven to ten days to solidify all of the melt.

The furnace arrangement of the HB method requires few moving parts. The boundary temperature does move but this can be accomplished mechanically or electronically. Control requirements are therefore simplified. Also, for Gallium Arsenide growth, the HB method provides tighter control on the process stoichiometry because of the Arsenic rich atmosphere above the melt. Thus crystals can be grown with a minimum of defects.

The method does have its short commings. The furnace itself introduces asymmetry that can produce nonuniformities in the crystal's growth. Crystals take the shape of the quartz boat and thus require more processing to get to the wafer stage. The dominant heat transfer mechanism is radiation, so that accurate (axial) temperature profiles are difficult to achieve. The real risk here is the potential for completely melting the seed during the heat up stage. Significant losses of time and energy would result from this since the crystal would unknowingly be grown without a design orientation. For this reason development work on the HB process is aimed at controlling the location, and shape, of the solid liquid interface during both the seeding process and subsequent growth phase.

## OBJECTIVE

The objective of this work is to investigate a new scheme to control the position and shape of the solid-liquid interface during the seeding process, as it applies to the HB growth method. The scheme involves the use of various seed "blanket" materials as a means to obtain accurate temperature control. Assessment of each material's performance will be made using a "front tracking" finite element code developed by Yoo and Rubinsky (6).

## MATHEMATICAL MODEL

The model used to analyze the seeding process is shown in Figure 6. Regions $R_1$ and $R_2$ represent the solid and liquid phases of the seed respectively. Region $R_3$ is the seed blanket material. The solid-liquid interface is denoted by a surface s. The entire solution space is given as $\tau$ with boundary $\sigma$. An outward unit normal vector on the interface is given by $\vec{n}$. Note that regions $R_1$ and $R_2$ as well as the surface s can be time varying whereas the boundary $\sigma$ is fixed. The target of this analysis is to determine the temperature distributions throughout the domain and the position of the solid-liquid interface as functions of time.

The following assumptions are made:

1. The seed material has a definite phase transition temperature

2. The change of phase interface is a definite surface in space

3. Maximum temperature differences and physical dimensions in the liquid region make natural convection effects neglegable

4. Material properties are not functions of temperature

5. Volume change from solid to liquid is small

6. Mass diffusion is not considered

7. The position of the interface is known at time zero

Considering these assumptions, the governing equations of the problem are:

$$\nabla \cdot (k \nabla T) = \rho c \frac{\partial T}{\partial t} \qquad \text{in } \tau \tag{1}$$

subject to

$$T = f(\vec{r}, t) \qquad \text{on } \sigma_1 \tag{2}$$

$$k \frac{\partial T}{\partial n} = \bar{q} \qquad \text{on } \sigma_2 \tag{3}$$

with the interface conditions

$$T = T_{PH} \qquad \text{on } s(t) \tag{4}$$

$$k_1 \frac{\partial T_1}{\partial n} - k_2 \frac{\partial T_2}{\partial n} = \rho L V_n \qquad \text{on } s(t) \tag{5}$$

and arbitrary initial conditions

$$T = T_i(\vec{r}) \qquad \text{in } \tau \tag{6}$$

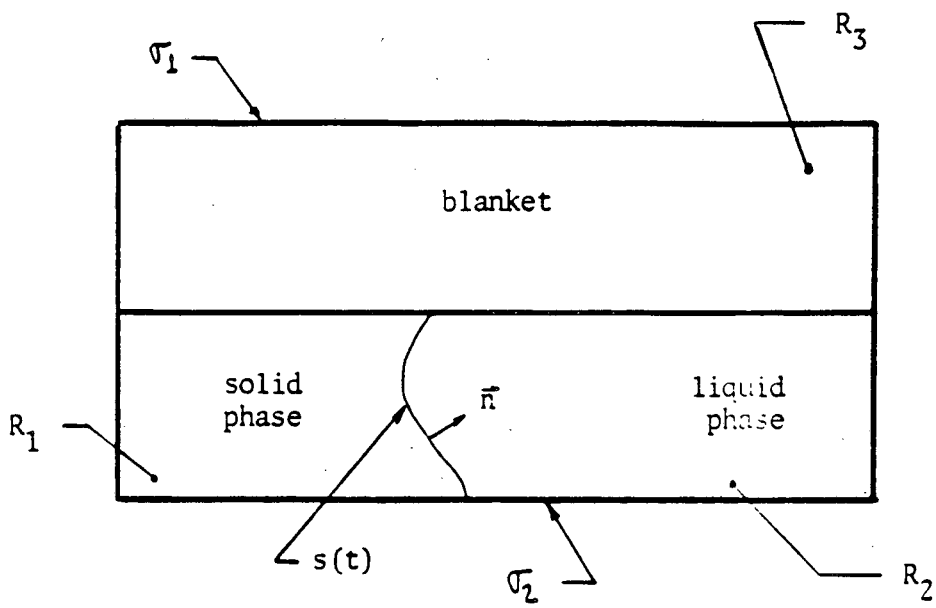



Figure 6. Physical model for the seeding process

Note that material properties in the blanket may be anisotropic but are not temperature dependent.

The finite element formulation is limited to two-dimensional or three-dimensional axisymmetric problems. Proceeding with the Galerkin Method of Weighted Residuals we construct the following weak form from the governing equation and boundary conditions:

$$\int_{\tau} \left\{ \nabla \cdot (k \nabla T) - \rho c \frac{\partial T}{\partial t} \right\} \eta \, d\tau + \int_{\sigma_1} (T - f) \eta \, d\sigma_1 + \int_{\sigma_2} \left\{ k \frac{\partial T}{\partial n} - \bar{q} \right\} \eta \, d\sigma_2 = 0 \tag{7}$$

The interior and boundary errors are to be minimized with respect to an arbitrary weight function $\eta$. In keeping with the basis of the Galerkin method this weight function is chosen to be a small variation of the exact solution. That is

$$\eta = \delta T$$

Note that on the $\sigma_1$ boundary $\delta T$ is zero since the value of the field variable is specified. Thus the boundary integral vanishes on $\sigma_1$. Integrating Eq.1 once by parts and then applying the divergence theorem gives

$$\int_{\tau} \left\{ k T_{,i} \, \delta T_{,i} - \rho c \frac{\partial T}{\partial t} \delta T \right\} d\tau - \int_{\sigma_2} \bar{q} \, \delta T \, d\sigma_2 = 0 \tag{8}$$

And choosing Cartesian coordinates gives the final weak form.

$$\int_{\tau} \left\{ k \left[ \frac{\partial T}{\partial x} \frac{\partial \delta T}{\partial x} + \frac{\partial T}{\partial y} \frac{\partial \delta T}{\partial y} \right] - \rho c \frac{\partial T}{\partial t} \delta T \right\} dA - \int_{\sigma_2} \bar{q} \, \delta T \, d\sigma_2 = 0 \tag{9}$$

The problem domain is now divided into N distinct subdomains. The field variable distribution is approximated as a finite sum over all N subdomains, using known interpolation functions (shape functions) and unknown coefficients or nodal point values. Specifically

$$T = \sum_{i=1}^{N} \underset{\sim}{\phi}_i^t \underset{\sim}{T}_i \tag{10}$$

$$\delta T = \sum_{i=1}^{N} \underset{\sim}{\phi}_i^t \delta \underset{\sim}{T}_i \tag{11}$$

$$\frac{\partial T}{\partial x} = \sum_{i=1}^{N} \frac{\partial}{\partial x} \underset{\sim}{\phi}_i^t \underset{\sim}{T}_i = \sum_{i=1}^{N} \underset{\sim}{a}_i^t \underset{\sim}{T}_i \tag{12}$$

$$\frac{\partial T}{\partial y} = \sum_{i=1}^{N} \frac{\partial}{\partial y} \underset{\sim}{\phi}_i^t \underset{\sim}{T}_i = \sum_{i=1}^{N} \underset{\sim}{b}_i^t \underset{\sim}{T}_i \tag{13}$$

These distributions are represented in a global context by dropping the subscripts. For instance

$$T = \underset{\sim}{\phi}^t \underset{\sim}{T}$$

Here the local summation is implied. Substituting the global forms of Eqs.10-13 into the weak form gives

$$\delta \underset{\sim}{T} \left[ \int_{\tau} \left\{ k(\underset{\sim}{a}\underset{\sim}{a}^t + \underset{\sim}{b}\underset{\sim}{b}^t)\underset{\sim}{T} - \rho c \underset{\sim}{\phi}\underset{\sim}{\phi}^t \frac{\partial T}{\partial t} \right\} dA - \int_{\sigma_2} \bar{q} \underset{\sim}{\phi} \, d\sigma_2 \right] = 0 \tag{14}$$

or

$$\delta \underset{\sim}{T} \left[ \int_{\tau} \left\{ k \underline{B}\underline{B}^t \underset{\sim}{T} - \rho c \underset{\sim}{\phi}\underset{\sim}{\phi}^t \frac{\partial T}{\partial t} \right\} dA - \int_{\sigma_2} \bar{q} \underset{\sim}{\phi} \, d\sigma_2 \right] = 0 \tag{15}$$

where

$$\underline{B} = \begin{bmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \end{bmatrix} \tag{16}$$

For arbitrary dt this equation reduces to the form

$$\underline{C}\,\frac{d\underline{T}}{dt} + \underline{K}\,\underline{T} + \underline{F} = 0 \tag{17}$$

with

$$\underline{C} = \int_{\tau} \rho c\, \underline{\phi}\, \underline{\phi}^{t}\, dA \tag{18}$$

$$\underline{K} = \int_{\tau} \underline{B}^{t}\, \underline{k}\, \underline{B}\, dA \tag{19}$$

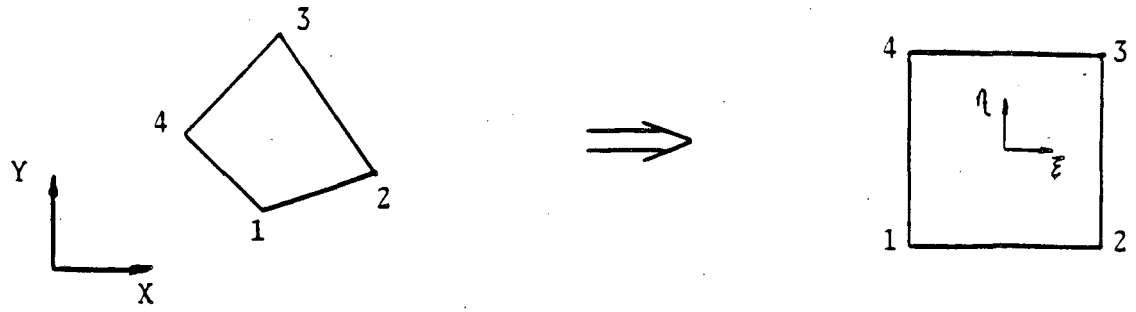
for anisotropic materials

$$\underline{k} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

$$\underline{F} = \int_{\sigma_2} \bar{q}\, \underline{\phi}\, d\sigma_2 \tag{20}$$

In the interest of computing efficiency it is desirable to evaluate the integrals in Eqs.17-20 over regular shapes. Thus we introduce an invertible coordinate transformation as shown in Figure 7. such that

$$X = x(\xi,\eta) = \underline{\psi}^{t}\underline{x} \quad \text{and} \quad Y = Y(\xi,\eta) = \underline{\psi}^{t}\underline{Y} \tag{21}$$




MAPPING

$$X = \underset{\sim}{\psi}^t \underset{\sim}{X} = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & \psi_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

INTERPOLATION

$$T = \underset{\sim}{\phi}^t \underset{\sim}{T} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$$

$$\phi_1 = \tfrac{1}{4}(1 - \xi)(1 - \eta)$$

$$\phi_2 = \tfrac{1}{4}(1 + \xi)(1 - \eta)$$

$$\phi_3 = \tfrac{1}{4}(1 + \xi)(1 + \eta)$$

$$\phi_4 = \tfrac{1}{4}(1 - \xi)(1 + \eta)$$

TRANSFORMATION MATRIX

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -\tfrac{1}{4}(1 - \eta) & \tfrac{1}{4}(1 - \eta) & \tfrac{1}{4}(1 + \eta) & -\tfrac{1}{4}(1 + \eta) \\ -\tfrac{1}{4}(1 - \xi) & -\tfrac{1}{4}(1 + \xi) & \tfrac{1}{4}(1 + \xi) & \tfrac{1}{4}(1 - \xi) \end{bmatrix} \begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ X_3 & Y_3 \\ X_4 & Y_4 \end{bmatrix}$$

Figure 7. Coordinate transformation from x,y space to $\xi,\eta$ space

Where $\underset{\sim}{\psi}$ is the known interpolation function and $\underset{\sim}{x}, \underset{\sim}{y}$ are the coordinate points to be mapped into $\xi, \eta$ space. We chose an isoparametric mapping, thus the transformation $\underset{\sim}{\psi}$ is identical to the field variable shape function $\underset{\sim}{\phi}$. Then

$$x = \underset{\sim}{\phi}^t \underset{\sim}{x} \qquad \text{and} \qquad Y = \underset{\sim}{\phi}^t \underset{\sim}{Y} \tag{22}$$

Temperature gradients in $\xi, \eta$ space are calculated using the chain rule of differentiation. In matrix form we have

$$\begin{bmatrix} \frac{\partial T}{\partial \xi} \\ \frac{\partial T}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = \underline{J} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} \tag{23}$$

which defines the transformation matrix $\underline{J}$. Thus we can write

$$\underline{B}_{(x,y)} \underset{\sim}{T} = \underline{J}^{-1} \underline{B}_{(\xi,\eta)} \underset{\sim}{T} \tag{24}$$

It can also be shown that an increment of area in x,y space transforms to $\xi, \eta$ space weighted by the Jacobian of transformation. That is

$$\int dx\, dy = \int \det \underline{J}\, d\xi\, d\eta \tag{25}$$

Hence the integrations in Eqs.17-20 become

$$\underline{C} = \int_{-1}^{+1} \int_{-1}^{+1} \rho c\, \underset{\sim}{\phi}\, \underset{\sim}{\phi}^t \det \underline{J}\, d\xi\, d\eta \tag{26}$$

$$\underline{K} = \int_{-1}^{+1} \int_{-1}^{+1} \underline{B}^t \underline{J}^{-t} \underline{K}\, \underline{J}^{-1} \underline{B} \det \underline{J}\, d\xi\, d\eta \tag{27}$$

$$\underset{\sim}{F} = \int_{-1}^{+1} \bar{q}\, \underset{\sim}{\phi} \det \underline{J}\, d\eta \tag{28}$$

These are evaluated numerically using a Gaussian Quadrature method. The form of the integrals becomes

$$\underline{C} = \sum_{i=1}^{m} \sum_{j=1}^{m} w_i w_j \rho c \, \underset{\sim}{\phi} \underset{\sim}{\phi}^t \det \underline{J} \tag{29}$$

$$\underline{K} = \sum_{i=1}^{m} \sum_{j=1}^{m} w_i w_j \, \underline{B}^t \underline{J}^{-t} \underline{K} \underline{J}^{-1} \underline{B} \det \underline{J} \tag{30}$$

$$\underset{\sim}{F} = \sum_{i=1}^{m} w_i \bar{q} \underset{\sim}{\phi} \det \underline{J} \tag{31}$$

where m is the number of function evaluations in each coordinate direction and $w_{i,j}$ are the weight factors. Note that an m point Gaussian Quadrature routine will integrate polynomials of degree 2m-1 exactly.

The heart of the front tracking finite element method is the use of the energy balance on the interface as an independent equation. The solution of which gives the position of the interface as a function of time. As before we begin by constructing the following weak form from Eq.5

$$\int_{S(t)} \left\{ k_1 \frac{\partial T_1}{\partial n} - k_2 \frac{\partial T_2}{\partial n} - \rho L V_n \right\} \omega \, ds = 0 \tag{32}$$

Where $\omega$ is the weight function. Note that this form does not yield the interface position directly. Also, from a finite element standpoint the normal derivatives are inconvenient to work with. Rubinsky and Yoo (4) decomposed Eq.5 using a new statement of the interface's isothermal nature. Their method is repeated here. The solid-liquid interface is described at each point by a

normal vector n, and the surface s. Derivatives in the n and s directions can be related to x and y directions using the following transformation

$$\begin{bmatrix} \frac{\partial}{\partial n} \\ \frac{\partial}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial n} & \frac{\partial y}{\partial n} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \tag{33}$$

where '$\alpha$' is the angle between the normal vector and the positive x-direction. Using Eq.33 we can rewrite Eq.5 as

$$k_1\left(\cos\alpha\frac{\partial T_1}{\partial x} + \sin\alpha\frac{\partial T_1}{\partial y}\right) - k_2\left(\cos\alpha\frac{\partial T_2}{\partial x} + \sin\alpha\frac{\partial T_2}{\partial y}\right) = \rho L V_n \tag{34}$$

The velocity of any point on the interface is a vector quantity with components

$$V_{n_x} = V_n\cos\alpha \qquad \text{and} \qquad V_{n_y} = V_n\sin\alpha \tag{35}$$

Multiplying Eq.34 by cos($\alpha$) and rearranging gives

$$\cos\alpha\left\{\cos\alpha\left(k_1\frac{\partial T_1}{\partial x} - k_2\frac{\partial T_2}{\partial x}\right) + \sin\alpha\left(k_1\frac{\partial T_1}{\partial y} - k_2\frac{\partial T_2}{\partial y}\right)\right\} = \rho L V_{n_x} \tag{36}$$

Eq.4 is now used to construct an energy balance on the interface for the s-direction. Thus

$$k_1\frac{\partial T_1}{\partial s} - k_2\frac{\partial T_2}{\partial s} = 0 \tag{37}$$

or, using Eq.33 gives

$$k_1\left(-\sin\alpha\frac{\partial T_1}{\partial x} + \cos\alpha\frac{\partial T_1}{\partial y}\right) - k_2\left(-\sin\alpha\frac{\partial T_2}{\partial x} + \cos\alpha\frac{\partial T_2}{\partial y}\right) = 0 \tag{38}$$

Rearranging

$$\sin\alpha\left(k_1\frac{\partial T_1}{\partial x} - k_2\frac{\partial T_2}{\partial x}\right) = \cos\alpha\left(k_1\frac{\partial T_1}{\partial y} - k_2\frac{\partial T_2}{\partial y}\right) \tag{39}$$

Eliminating the y-derivatives between Eqs.34 and 39, and using a common trigonometric identity leads to

$$k_1\frac{\partial T_1}{\partial x} - k_2\frac{\partial T_2}{\partial x} = \rho L V_{n_x} \tag{40}$$

A similar decomposition can be used for the y-direction to obtain

$$k_1\frac{\partial T_1}{\partial y} - k_2\frac{\partial T_2}{\partial y} = \rho L V_{n_y} \tag{41}$$

Then using the definition of velocity for a point on the interface in Eqs.40 and 41, gives a form that can be readily discretized for solution

$$\begin{aligned} k_1\frac{\partial T_1}{\partial x} - k_2\frac{\partial T_2}{\partial x} &= \rho L\frac{ds_x}{dt} \\ k_1\frac{\partial T_1}{\partial y} - k_2\frac{\partial T_2}{\partial y} &= \rho L\frac{ds_y}{dt} \end{aligned} \tag{42}$$

where $ds_x$ and $ds_y$ are the x and y displacements of the point. Thus the final weak form is written as

$$\int_{s(t)}\begin{bmatrix} k_1\frac{\partial T_1}{\partial x} - k_2\frac{\partial T_2}{\partial x} \\ k_1\frac{\partial T_1}{\partial y} - k_2\frac{\partial T_2}{\partial y}\end{bmatrix}\omega\, ds = \int_{s(t)}\rho L\begin{bmatrix}\frac{ds_x}{dt} \\ \frac{ds_y}{dt}\end{bmatrix}\omega\, ds \tag{43}$$

Since at this point the temperature field is assumed known, Eq.43 can be solved for the interface displacements as a function of time.

In the finite element formulation of this weak form the interface displacements are approximated in a manner similar to the temperature distributions. That is

$$ds_x = \sum_{i=1}^{M} \underset{\sim}{H}_i^t \, d\underset{\sim}{x}_i \qquad \text{and} \qquad ds_y = \sum_{i=1}^{M} \underset{\sim}{H}_i^t \, d\underset{\sim}{y}_i \tag{44}$$

where the $\underset{\sim}{H}i$ are known interpolation functions and $d\underset{\sim}{x}_i, d\underset{\sim}{y}_i$ are unknown nodal point values for the ith element. Combining these we have, in a global sense

$$d\underset{\sim}{s} = \begin{Bmatrix} ds_x \\ ds_y \end{Bmatrix} = \underline{U} \, d\underset{\sim}{S} \tag{45}$$

where $\underline{U}$ is a combination of the interpolation functions and $d\underset{\sim}{S}$ is the nodal displacement vector. The velocity is likewise given as

$$\frac{d\underset{\sim}{s}}{dt} = \underline{U} \, \frac{d\underset{\sim}{S}}{dt} \tag{46}$$

Denoting the temperature gradients as before and choosing the weight function in the same space as the interpolation Eq.43 becomes

$$\int_{s(t)} \underline{U}^t \left\{ k_1 \underline{B}_1 \underset{\sim}{T}_1 - k_2 \underline{B}_2 \underset{\sim}{T}_2 \right\} ds = \int_{s(t)} \rho L \, \underline{U} \, \underline{U}^t \frac{d\underset{\sim}{S}}{dt} \, ds \tag{47}$$

We again want to perform the integration over regular shapes and thus the interface elements are mapped into $\xi, \eta$ space as in Figure 7. This gives an equation of the form

$$\underline{P}\,\frac{d\underset{\sim}{S}}{dt} = \underset{\sim}{Q} \tag{48}$$

where

$$\underline{P} = \int_{-1}^{+1} \rho L\, \underline{U}\,\underline{U}^t \det \underline{J}\, d\eta \tag{49}$$

$$\underset{\sim}{Q} = \int_{-1}^{+1} \underline{U}^t \left\{ k_1 \underline{J}^{-1} \underline{B}_1 \underset{\sim}{T}_1 - k_2 \underline{J}^{-1} \underline{B}_2 \underset{\sim}{T}_2 \right\} \det \underline{J}\, d\eta \tag{50}$$

Gaussian Quadrature is used to evaluate the integrals resulting in the form

$$\underline{P} = \sum_{i=1}^{m} w_i\, \rho L\, \underline{U}\,\underline{U}^t \det \underline{J} \tag{51}$$

$$\underset{\sim}{Q} = \sum_{i=1}^{m} w_i\, \underline{U}^t \left\{ k_1 \underline{J}^{-1} \underline{B}_1 \underset{\sim}{T}_1 - k_2 \underline{J}^{-1} \underline{B}_2 \underset{\sim}{T}_2 \right\} \det \underline{J} \tag{52}$$

To illustrate the formulation of Eq.48 consider a portion of the interface modeled as a two node element in Figure 8. The interface displacement is approximated in this element by

$$ds_x = \left\lfloor \tfrac{1}{2}(1+\eta),\ \tfrac{1}{2}(1-\eta) \right\rfloor \begin{Bmatrix} dx_1 \\ dx_2 \end{Bmatrix} \tag{53}$$

$$ds_y = \left\lfloor \tfrac{1}{2}(1+\eta),\ \tfrac{1}{2}(1-\eta) \right\rfloor \begin{Bmatrix} dy_1 \\ dy_2 \end{Bmatrix} \tag{54}$$

Note that in $\xi, \eta$ space the interface movement is a function of $\eta$ alone. Combining these we get

$$\underset{\sim}{ds} = \underline{U}\,\underset{\sim}{dS} = \begin{bmatrix} \tfrac{1}{2}(1+\eta) & 0 & \tfrac{1}{2}(1-\eta) & 0 \\ 0 & \tfrac{1}{2}(1+\eta) & 0 & \tfrac{1}{2}(1-\eta) \end{bmatrix} \begin{Bmatrix} dx_1 \\ dy_1 \\ dx_2 \\ dy_2 \end{Bmatrix} \tag{55}$$

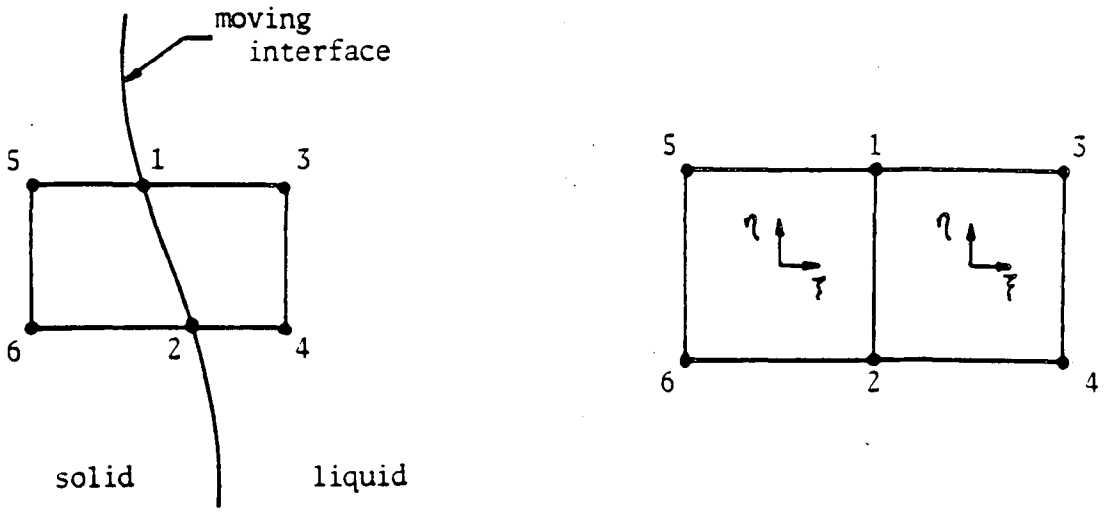



Figure 8. Two node representation of the solid-liquid interface, mapped onto a regular shape

Temperature gradient matrices are derived from the temperature distributions in elements adjacent to the interface. For instance

$$\underline{B}_1(\xi,\eta) = \begin{bmatrix} \frac{\partial \phi}{\partial \xi} \\ \frac{\partial \phi}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4}(1-\eta) & \frac{1}{4}(1-\eta) & \frac{1}{4}(1+\eta) & -\frac{1}{4}(1+\eta) \\ -\frac{1}{4}(1-\xi) & -\frac{1}{4}(1+\xi) & \frac{1}{4}(1+\xi) & \frac{1}{4}(1-\xi) \end{bmatrix} \tag{56}$$

Both $J(\xi,\eta)$ and $B(\xi,\eta)$ are evaluated at the Gauss points on the interface, so the B matrix becomes

$$\underline{B}_1(1,\eta_G) = \frac{1}{4}\begin{bmatrix} -(1-\eta_G) & (1-\eta_G) & (1+\eta_G) & -(1+\eta_G) \\ 0 & -2 & 2 & 0 \end{bmatrix} \tag{57}$$

The transformation for an element of arc length $ds$ is determined as follows:

$$ds = \left[ dx^2 + dy^2 \right]^{1/2} \tag{58}$$

where

$$dx = \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta \tag{59}$$

$$dy = \frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta \tag{60}$$

But on the interface

$$x = x(\xi,\eta) = x(1,\eta) \quad \text{and} \quad y = y(\xi,\eta) = y(1,\eta) \tag{61}$$

so that

$$\frac{\partial x}{\partial \xi} = \frac{\partial x}{\partial \xi} = 0$$

and Eq.58 becomes

$$d\Delta = \left[ \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 \right]^{1/2} d\eta = \det \underline{J}\, d\eta \qquad (61)$$

Thus for the two node element

$$\det \underline{J} = \frac{1}{2}\left[ (x_1 - x_2)^2 + (y_1 - y_2)^2 \right]^{1/2} \qquad (62)$$

## SOLUTION ALGORITHM

The simulation of the seeding process begins at a time when both the temperature field and the interface position are known. The basic steps for solution are:

1. Generate the finite element mesh for the solution space. Individual elements do not cross the boundaries between solid, liquid or blanket. Initial conditions are applied to all regions.

2. Solve Eq.17 for the current temperature field (ie. after a time interval dt) using a forward difference scheme for the time derivative. Note that boundary conditions are a function of time (see Figure 5) and hence are different for each time step.

3. Use the current temperature distribution to solve Eq.48 for the displacement of the interface nodes in this time interval.

4. Generate a new mesh for the solution space. Note that the size and shape of regions R1 and R2 will reflect the change in the interface location.

5. Calculate nodal point temperatures for the new mesh from the current temperature field (step 2). Essentially this is an interpolation routine.

6. Repeat steps 2 - 5 for each time step in the process.

A listing of the computer code and flow chart are given in the Appendix.

## RESULTS AND DISCUSSION

Recall the objective of this study was to investigate the use of various seed blanket materials to control position and shape of the solid/liquid interface during the seeding process. Originally, research was focused on ways to control the axial temperature distribution imposed on the seed inside the furnace. To do this required controlling the axial heat flow. Thus we envisioned filling the space between the seed and the furnace wall with an anisotropic media. The preferred direction of the media would be perpendicular to the seed's long axis. Thereby limiting axial heat flow, in favor of radial heat flow, and obtaining better temperature control at the seed.

We compared four material that were compatible with the melting temperature of Gallium Arsenide. These were a standard (isotropic) graphite, hot-pressed Boron Nitride (mildly anisotropic), pyrolytic graphite (strongly anisotropic) and an ideal anisotropic material with a 1000 to 1 difference in thermal conductivities. Thermal properties are listed in Table 1.

In addition, we varied both the growth rate and blanket orientation. Growth rate is determined by the speed at which the boundary temperature profile moves. Typical values are on the order of millimeters per hour. Orientation indicates the alignment of the preferred heat transfer direction (in the blanket) with the coordinate axes. We have assumed symmetry and thus model only half of the seed/blanket geometry. Actual physical dimensions are given in the Appendix. Temperature boundary conditions were imposed on all sides except the symmetry line which was insulated.

The growth, or temperature history the seed is subjected to is listed in Table 2. In the interest of computational expense we model the seeding process as three 250 minute stages: back-melting, stablization and solidification. Essentially we begin the analysis after Segment 2.

Interface shapes for a boundary velocity of 0.5 mm/hr and blanket orientation in the y-direction are shown in Figures 9a-d. Notice that once solidification begins the interface is concave. The most pronounced curvature results with the ideal anisotropic blanket. Physically the shape is consistent. The long edge of

| | Gallium Arsenide[1] solid | Gallium Arsenide[1] liquid | Graphite[2] | Hot-Pressed[3] Boron Nitride | Pyrolytic[4] Graphite | Ideal Anisotropic |
|---|---|---|---|---|---|---|
| Density ($kg/m^3$) | 5170 | 5700 | 1760 | 1900 | 2200 | 2200 |
| Specific Heat (J/kg°C) | 424 | 531 | 2009 | 2008 | 2009 | 2009 |
| Thermal Conductivity (W/m°C) | 7.1 | 14.0 | 110 | "A" 28.9<br>"C" 18.0 | "A" 125<br>"C" 1.11 | "A" 100<br>"C" 0.10 |
| Latent Heat of Fusion (kJ/kg) | 790 | | — | — | — | — |
| Melting Temperature (°C) | 1237 | | ≈ 4200 | >2000 | 4200 | — |

1. Taken from references (1) and (2)
2. Union Carbide technical data, UCAR Grade ECL
3. Union Carbide technical data, UCAR Grade HBC
4. Union Carbide technical data, UCAR Grade PG

Table 1. Thermal properties of blanket materials and Gallium Arsenide

| Segment | Location[a] | Temperature initial (°C) | Temperature final (°C) | Duration (min) |
|---|---|---|---|---|
| 1 | A | 25 | 1080 | 180 |
| (heat up) | B | 25 | 1244 | |
| 2 | A | 1080 | 1150 | 180 |
| (heat up) | B | 1244 | 1244 | |
| 3 | A | 1150 | 1224 | 300 |
| (back melt) | B | 1244 | 1244 | |
| 4 | A | 1224 | 1224 | 360 |
| (stablize) | B | 1244 | 1244 | |
| 5 | A | 1224 | 1204 | 1000 |
| (solidify) | B | 1244 | 1224 | |

a. Location refers to positions on the seed as shown in Figure 5.

Table 2. Example of the temperature history imposed on the seed crystal in the Horizontal Bridgeman process, taken from (3).

the seed provides good heat transfer area. So, with the blanket oriented in this direction it is possible to extract more energy perpendicular to the seed than along its length. As the degree of anisotropy increases so does the curvature. Interface curvature is more exaggerated for a growth rate of 5.0 mm/hr, as seen in Figures 10a-c. Note here that more solidification occurs with the pyrolytic graphite blanket for the given time.

We also oriented the blankets in the x-direction to approach the one dimensional heat transfer case. The inteface shapes for this orientation with 0.5 mm/hr growth rates, are shown in Figures 11a-c. The inteface is essentially flat for the pyrolytic graphite and ideal anisotropic blankets. These results are expected However, no solidification occurred even though the boundary profile was moved to begin that stage. These results are expected since heat transfer is essentially one dimensional, along the length of the seed. The Boron Nitride blanket being only mildly anisotropic, does not produce a flat interface. However the amount of curvature is reduced from the y-direction orientation (Figure 9b).

We had an opportunity to compare the predicted interface shape with actual experiments. Figure 15 shows the seed crystal used in that test. A power failure during the seeding process froze the interface at its steady shape. The system had been allowed to stablize. The seed material was Gallium Arsenide and it was contained in a blanket of Boron Nitride oriented in the y-direction. The growth rate was 3 mm/hr. The interface shape

compared qualitatively with the predicted shape for similar growth conditions (Figure 10b). Both are concave but the actual curvature is greater. The difference may be due to gravity effects since this was not in our model. In Figure 15 gravity acts into the page. Thus gravitational forces would draw liquid away from the interface and increase the curvature.

We turn now to the idea of interface position control. During the seeding process the boundary temperature profile follows a prescribed path in time (listed in Table 2). Ideally we want the interface to track the profile exactly. That is, the interface should follow the position of the phase change temperature (on the boundary) as it moves during each stage of the process. Precise control would eliminate the risk of losing the seed during the backmelting stage. Each blanket material was evaluated in this regard. The results are presented in Figures 12a-d to 14a-c. In these figures the location of the phase change temperature, imposed as a boundary condition on the blanket, is plotted as a function of time. The curve clearly shows three stages in the process. The positions of two interface nodes are also shown. Node 1 is on the symmetry line and Node 20 is on the seed blanket boundary.

Figures 12a-d are for a growth rate of 0.5 mm/hr with blanket orientation in the y-direction. The interface position tracks the boundary profile most closely for the strongly anisotropic materials. Again, this behavior is consistent since a high degree of anisotropy will transfer boundary information to

the seed more quickly. The isotropic graphite and Boron Nitride blankets let the interface gradually fall behind the boundary profile during solidification. However, the solidification rate is uniform.

Response to the 5.0 mm/hr growth rate is shown in figures 13a-c. The pyrolytic graphite blanket still provides the best overall response. Note that the Boron Nitride and graphite blankets now produce non-uniform solidification rates. The interface first lags then gains on the boundary profile. It is not clear whether a uniform rate will eventually take over. The actual amount of solidification is greatest for the strongly anisotropic blanket. The benefit here is crystals can be grown quicker and thus more economically.

Figures 14a-c show the response of the interface to blankets oriented in the x-direction. Since the Boron Nitride is only mildly anisotropic, there is still some measure of position control. The other blankets gave little if any control. It is apparent that a trade off is surfacing. To get a flat interface (for radial uniformity in the crystal) requires a strong anisotropic blanket oriented in the x-direction. In doing so we sacrifice control over the interface position.

A good compromise can be obtained using a Boron Nitride blanket, with x-orientation, and moving the boundary temperature profile at an intermediate speed. Figures 16 and 17 show the interface shape and response curves for these growth conditions. Note that interface curvature is reduced and the solidification

rate settles on a steady value. The blanket orientation makes this compromise somewhat counter-intuitive. However, the results suggest that optimal behavior might be obtained by just fine adjustments of that orientation.

The effect of initial conditions was also studied. Figures 9-14 were generated with linear initial conditions. That is, before the start of backmelting the temperature distribution in the seed is linear in 'x' and uniform in 'y'. The profile is that which exists on the boundary at the end of Segment 2 (see Table 2). We felt the time scale of the problem would dilute any variations in initial conditions. Test cases were run with uniform initial conditions throughout and results were unchanged. The same behavior was true of time step size. The bulk of our data was generated using a step size of thirty seconds. A factor of two increase or decrease had no effect on the interface shape or position.

## CONCLUSIONS AND RECOMMENDATIONS

A major conclusion from this study is anisotropic blankets can be used effectively, to control the position of the solid-liquid interface during the seeding process. The best response was obtained using the pyrolytic graphite material with its preferred direction oriented perpendicular to the seed's long axis. The risk of losing the seed during the backmelting stage is eliminated with this material. The subsequent solidification rate is uniform. Therefore, axial variation in crystal properties will be a minimum. Boron Nitride and isotropic graphite blankets also

provided good position control. However, the solidification rate was non-uniform during the first 250 minutes of growth.

The trade off for good position control is with the interface shape. All blanket materials oriented in the y-direction produced concave interface shapes upon solidification. The amount of curvature was not distinctly affected by the degree of anisotropy. Orienting in the x-direction eliminated interface curvature for the strong anisotropic materials. The Boron Nitride blanket did retain some curvature but it was reduced from the y-orientation. Qualitatively, the predicted interface shape agreed with the limited experimental evidence we had access to. experiment.

A good compromise between interface shape and position control was found using a Boron Nitride blanket, oriented in the x-direction, and moving the boundary temperature profile at 2.5 mm/hr. This result also suggested that optimum behavior might be obtained with just fine adjustments of the blanket's orientation.

Tests were not run for an unblanketed seed. An adequate treatment in this case requires modeling the radiation environment inside the furnace and was beyond the scope of this work. To include radiation in the Gallium Arsenide model would require information on surface properties (emissivity, reflectivity and transmissivity) for both the solid and liquid phases. Information which is currently unknown.

FIGURE 9a : INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH AN ISOTROPIC GRAPHITE BLANKET

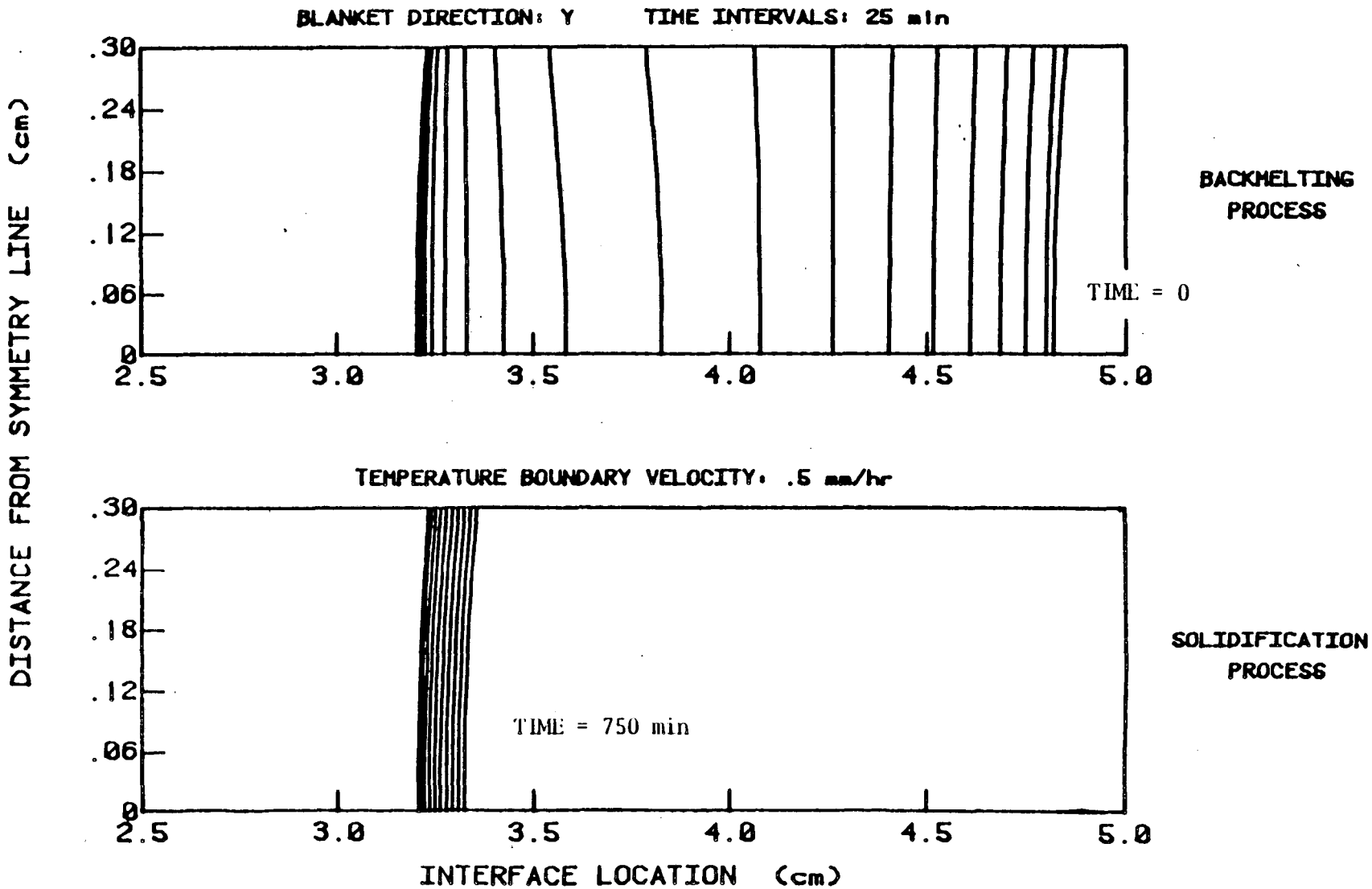


FIGURE 96 : INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A BORON NITRIDE BLANKET

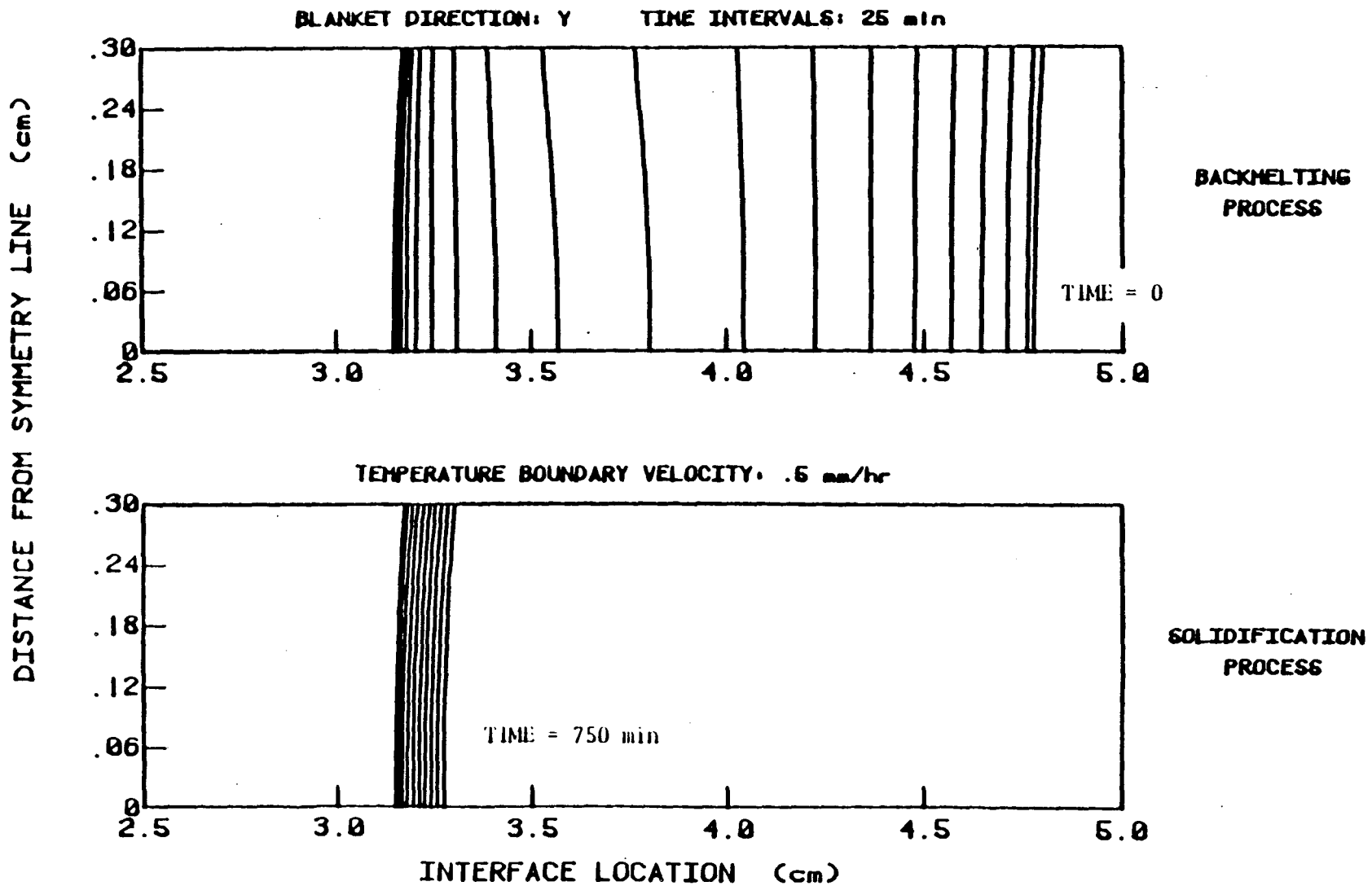


FIGURE 9c : INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A PYROLYTIC GRAPHITE BLANKET

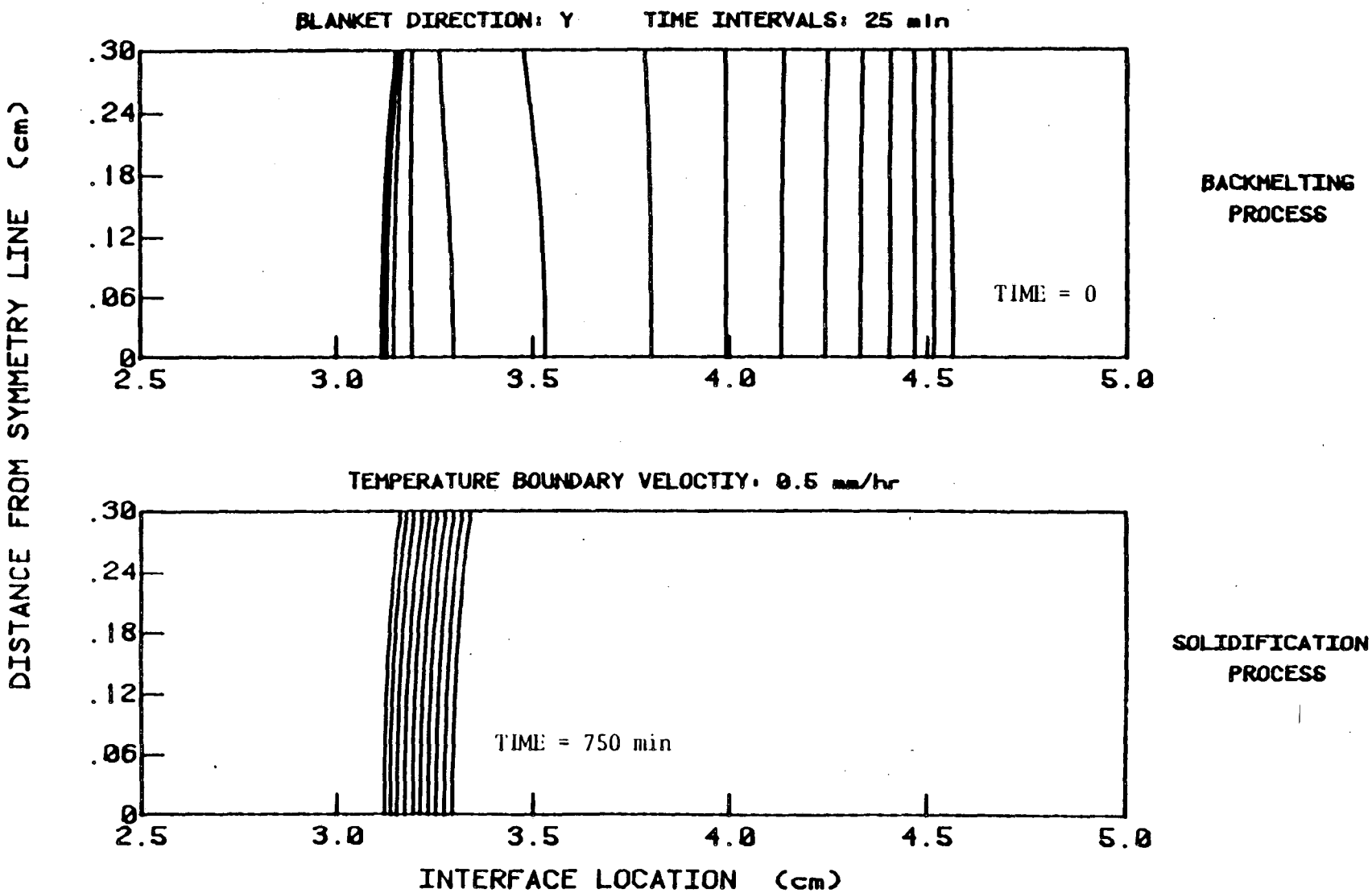


FIGURE 9d : INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH AN IDEAL ANISOTROPIC BLANKET

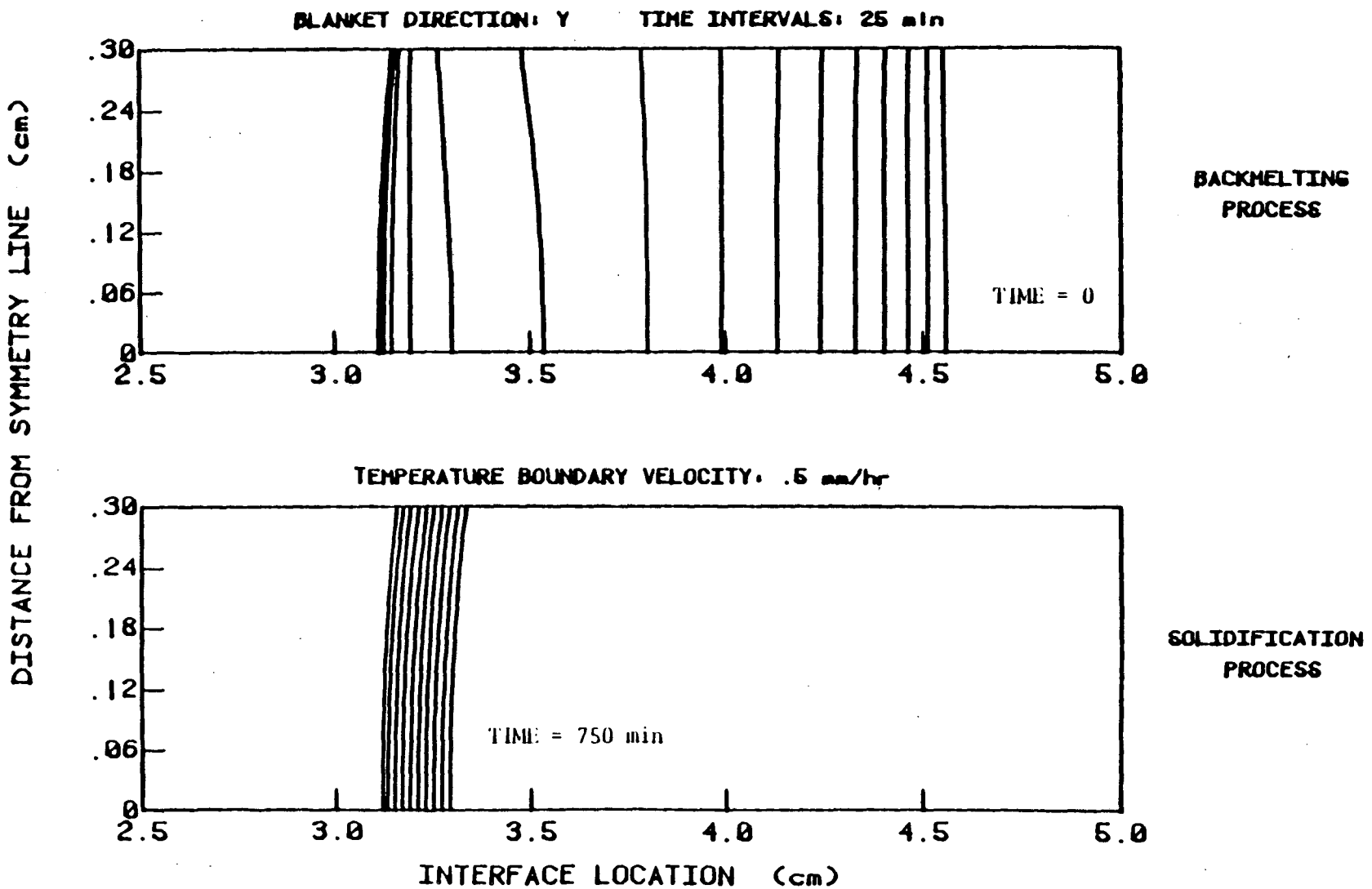


FIGURE 10a: INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH AN ISOTROPIC GRAPHITE BLANKET

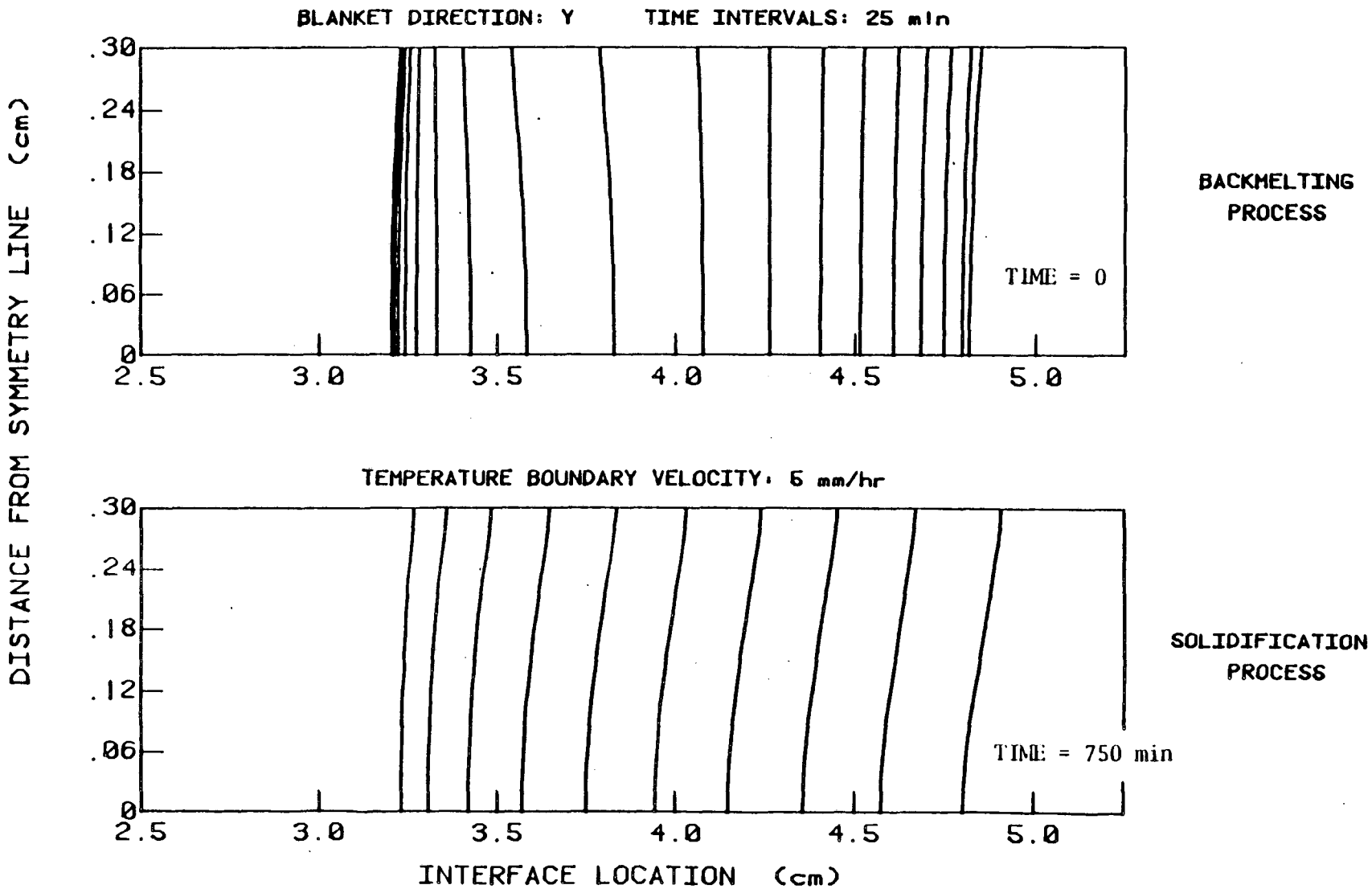


FIGURE 10b: INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A BORON NITRIDE BLANKET

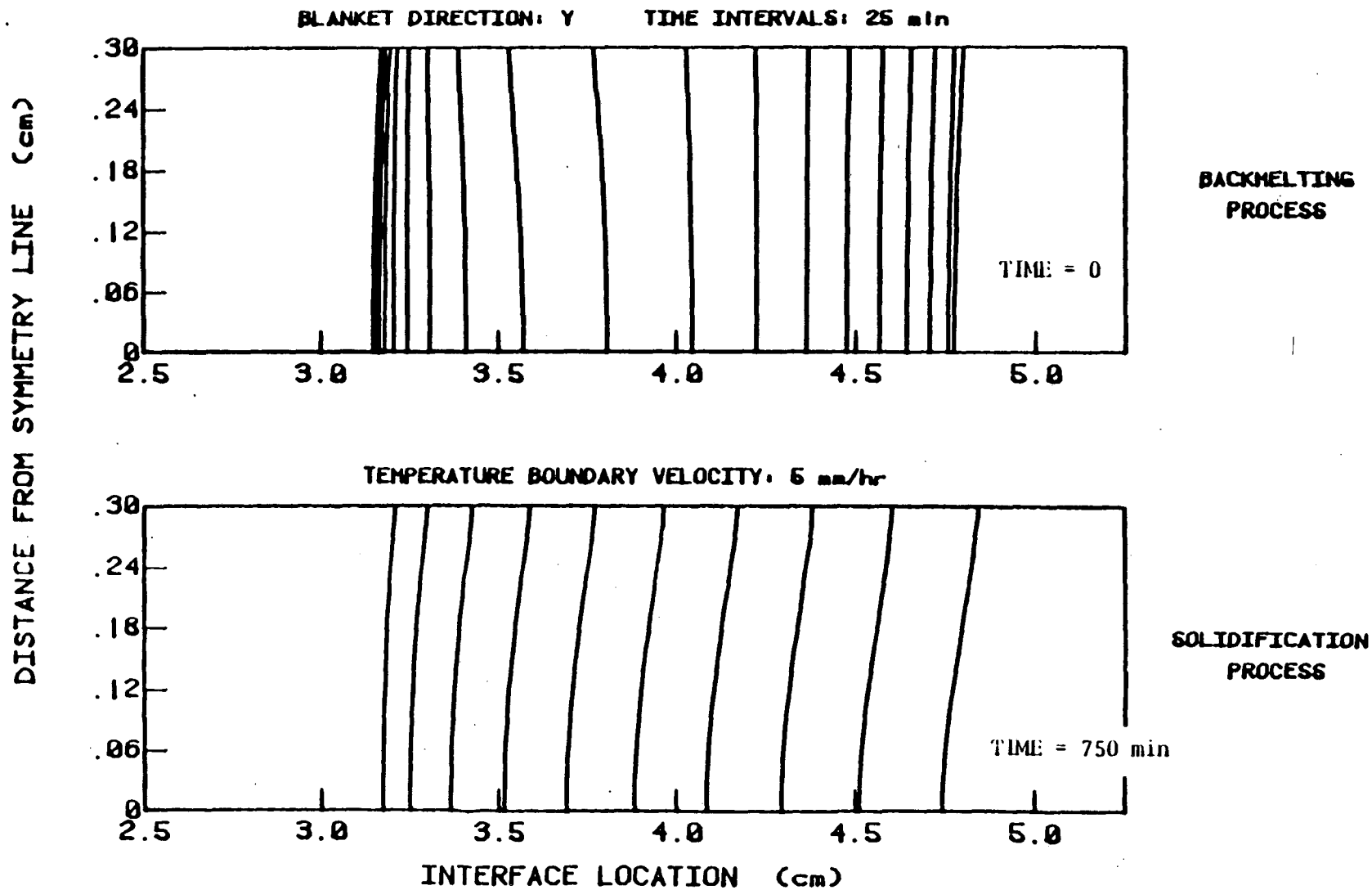


FIGURE 10c: INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A PYROLYTIC GRAPHITE BLANKET

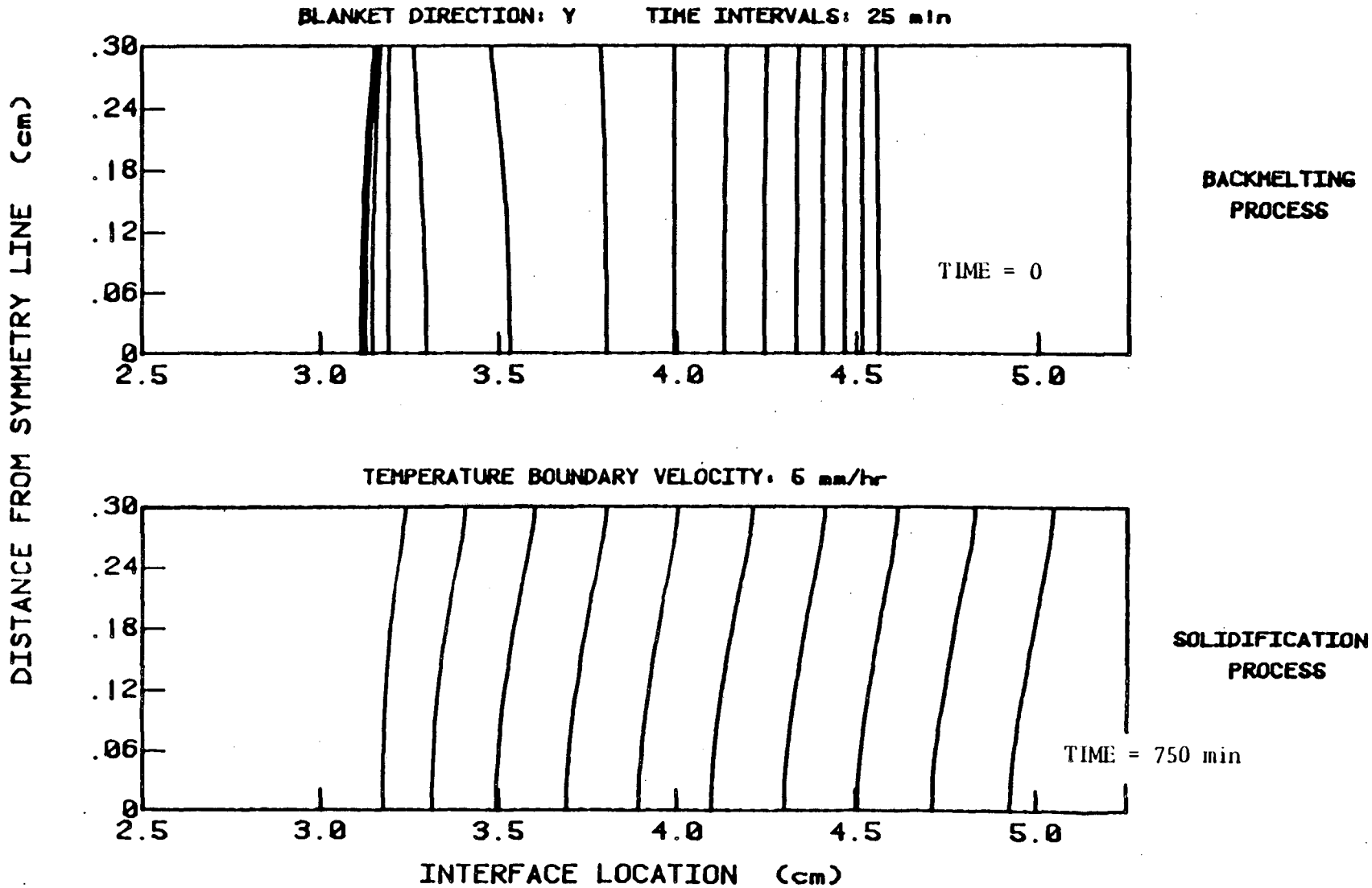


FIGURE 11a: INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A BORON NITRIDE BLANKET

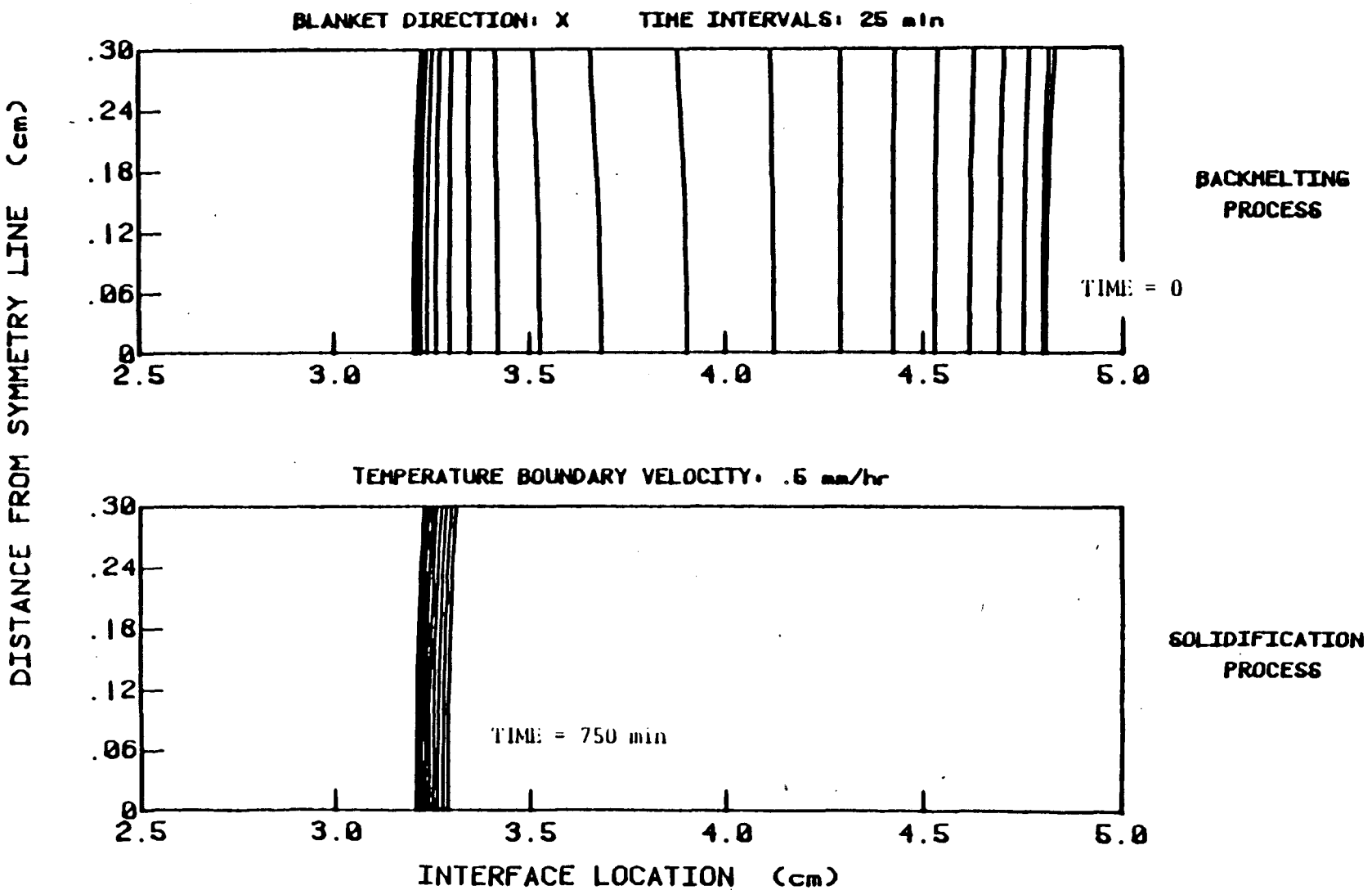


FIGURE 116: INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A PYROLYTIC GRAPHITE BLANKET

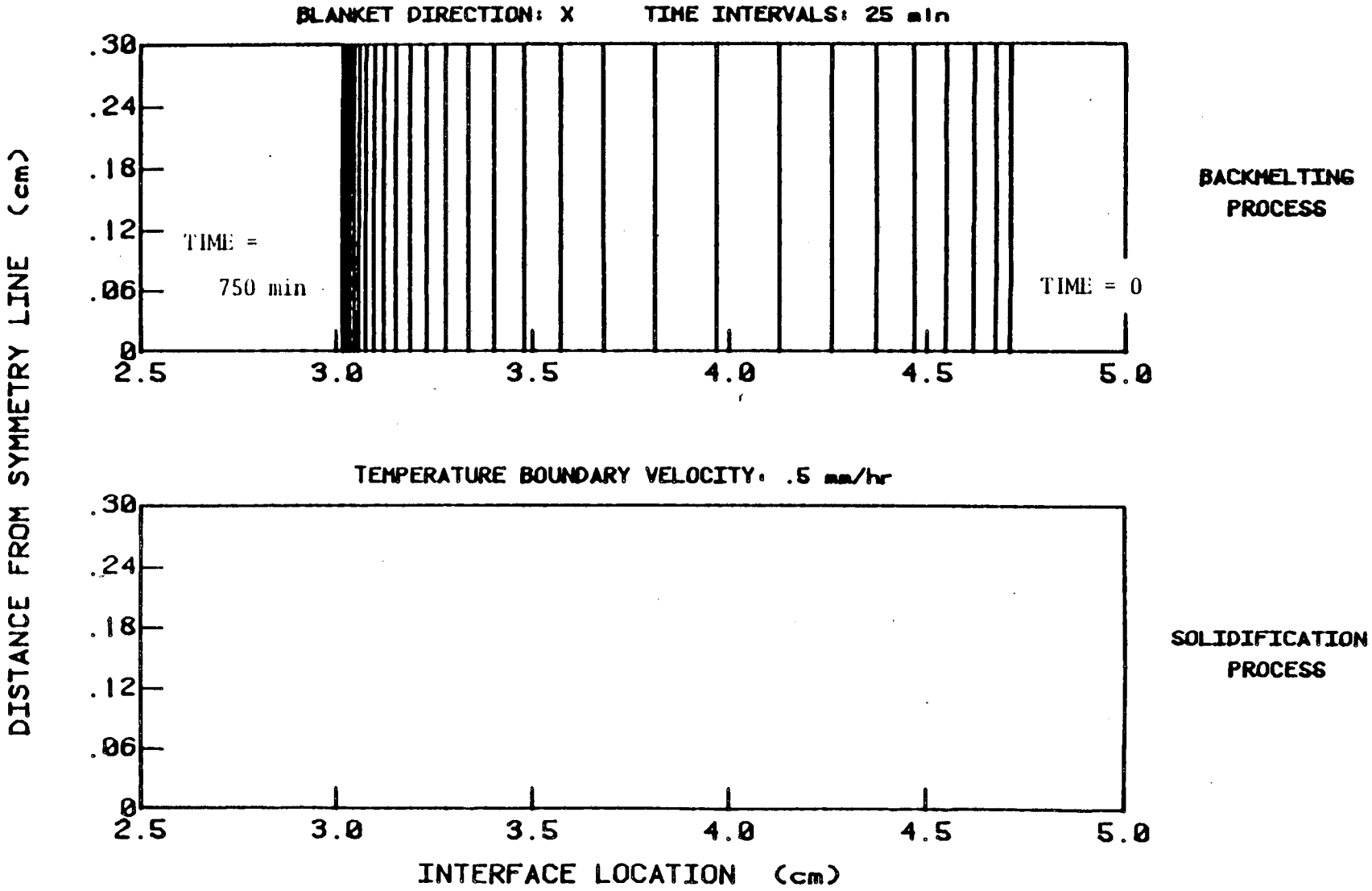


FIGURE IIc : INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH AN IDEAL ANISOTROPIC BLANKET

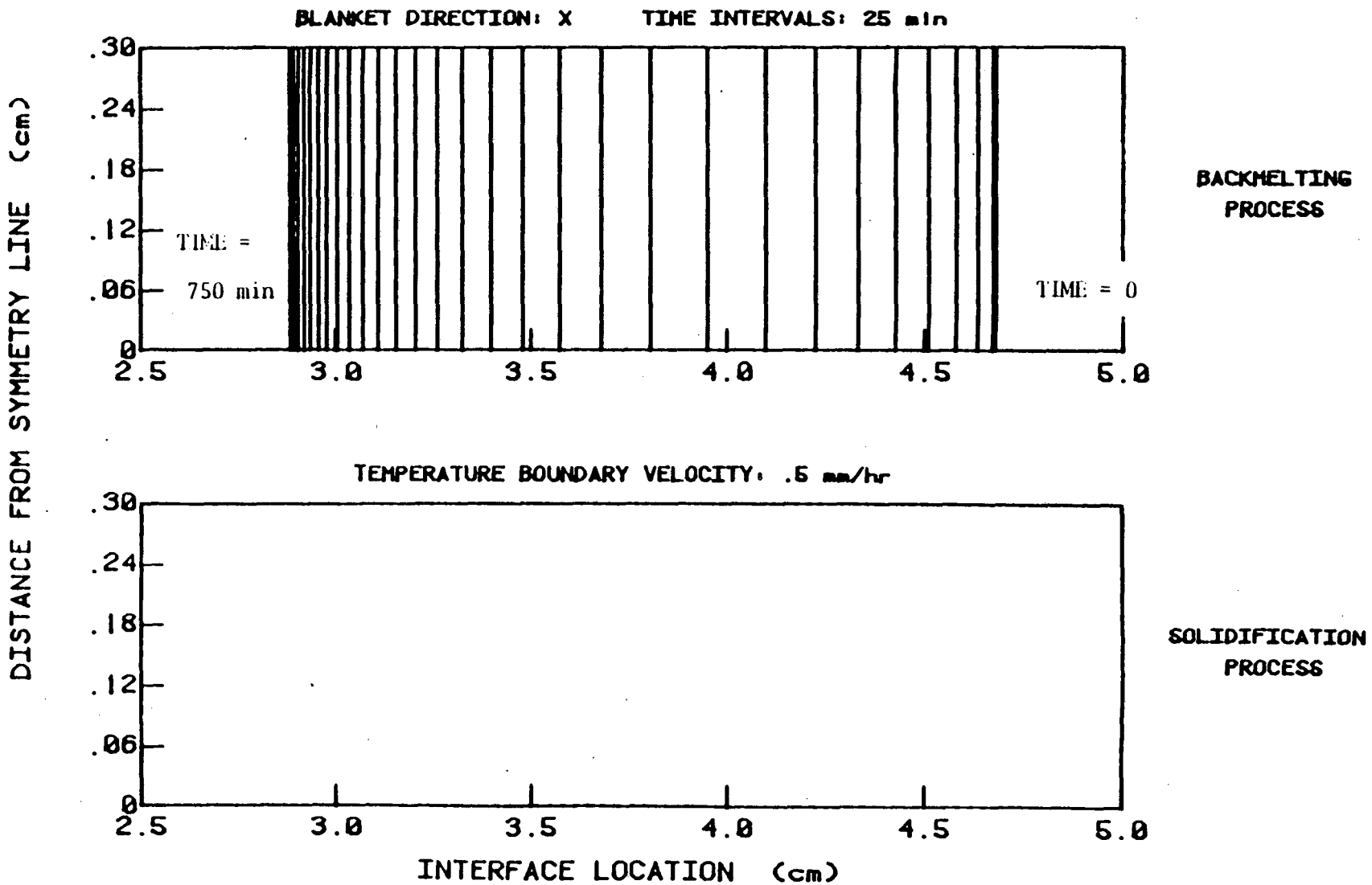


FIGURE 12a. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR AN ISOTROPIC GRAPHITE BLANKET

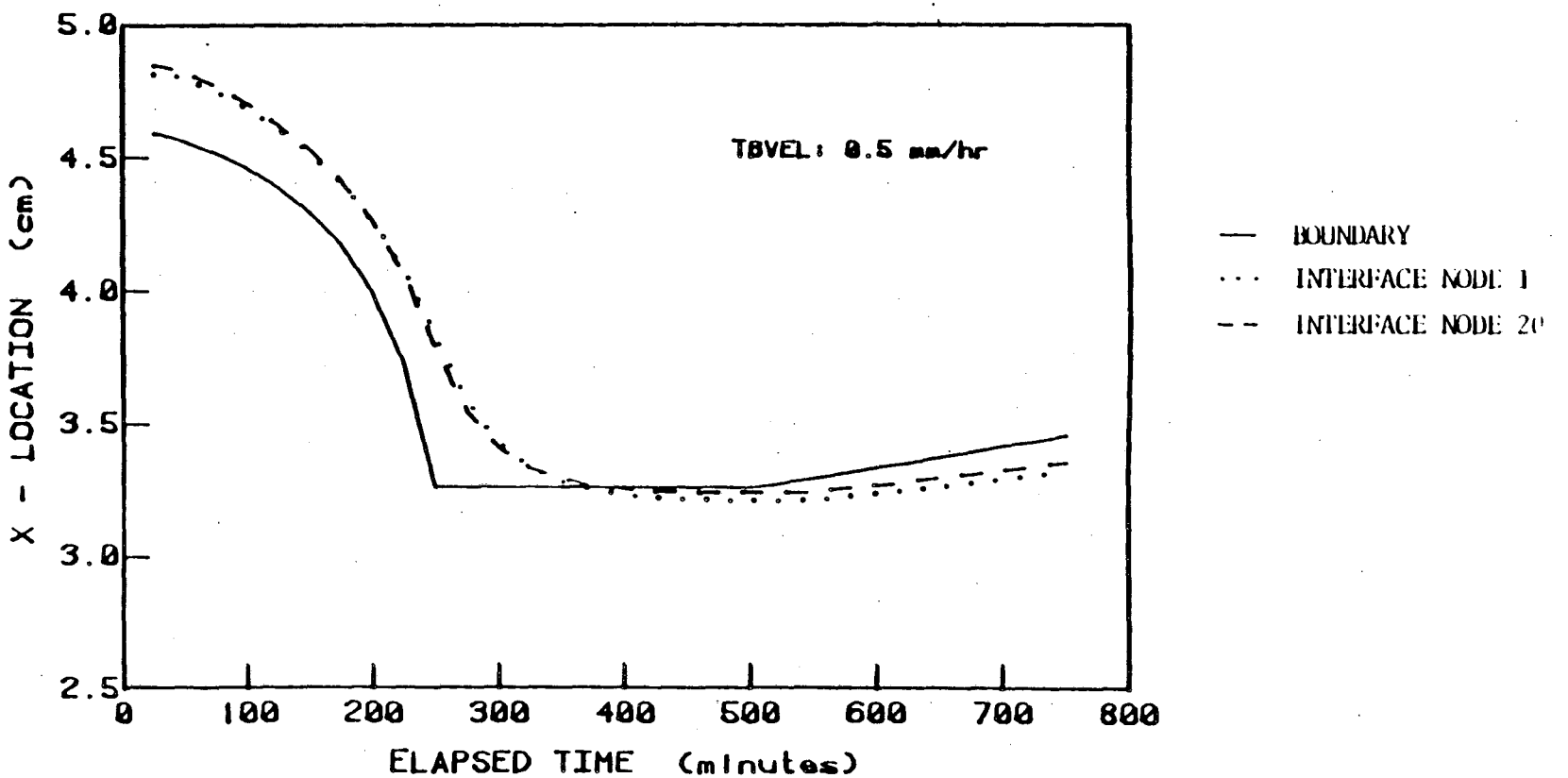


FIGURE 12b. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR A BORON NITRIDE BLANKET ORIENTED IN THE Y-DIRECTION

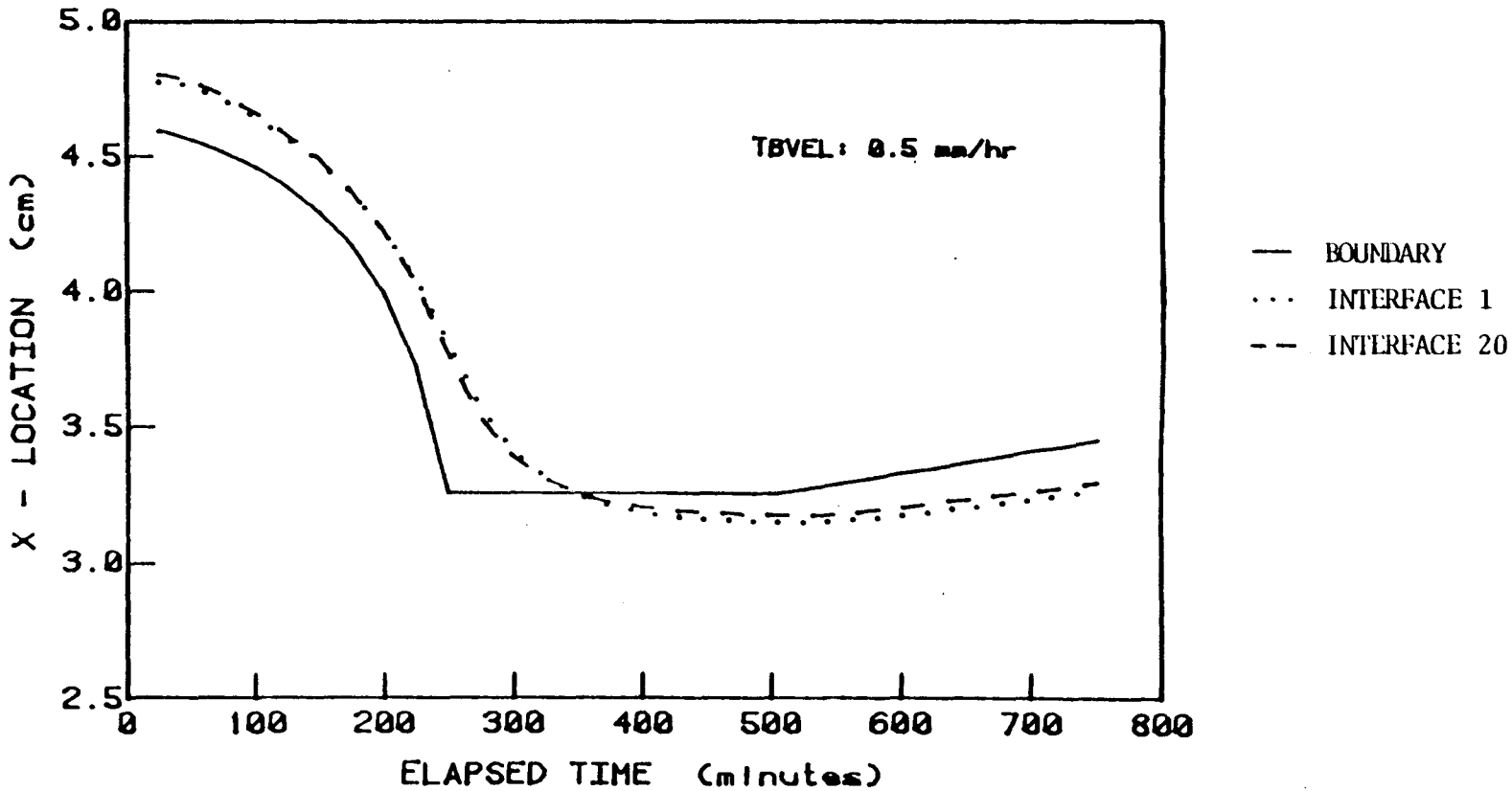


FIGURE 12c. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR A PYROLYTIC GRAPHITE BLANKET ORIENTED IN THE Y-DIRECTION

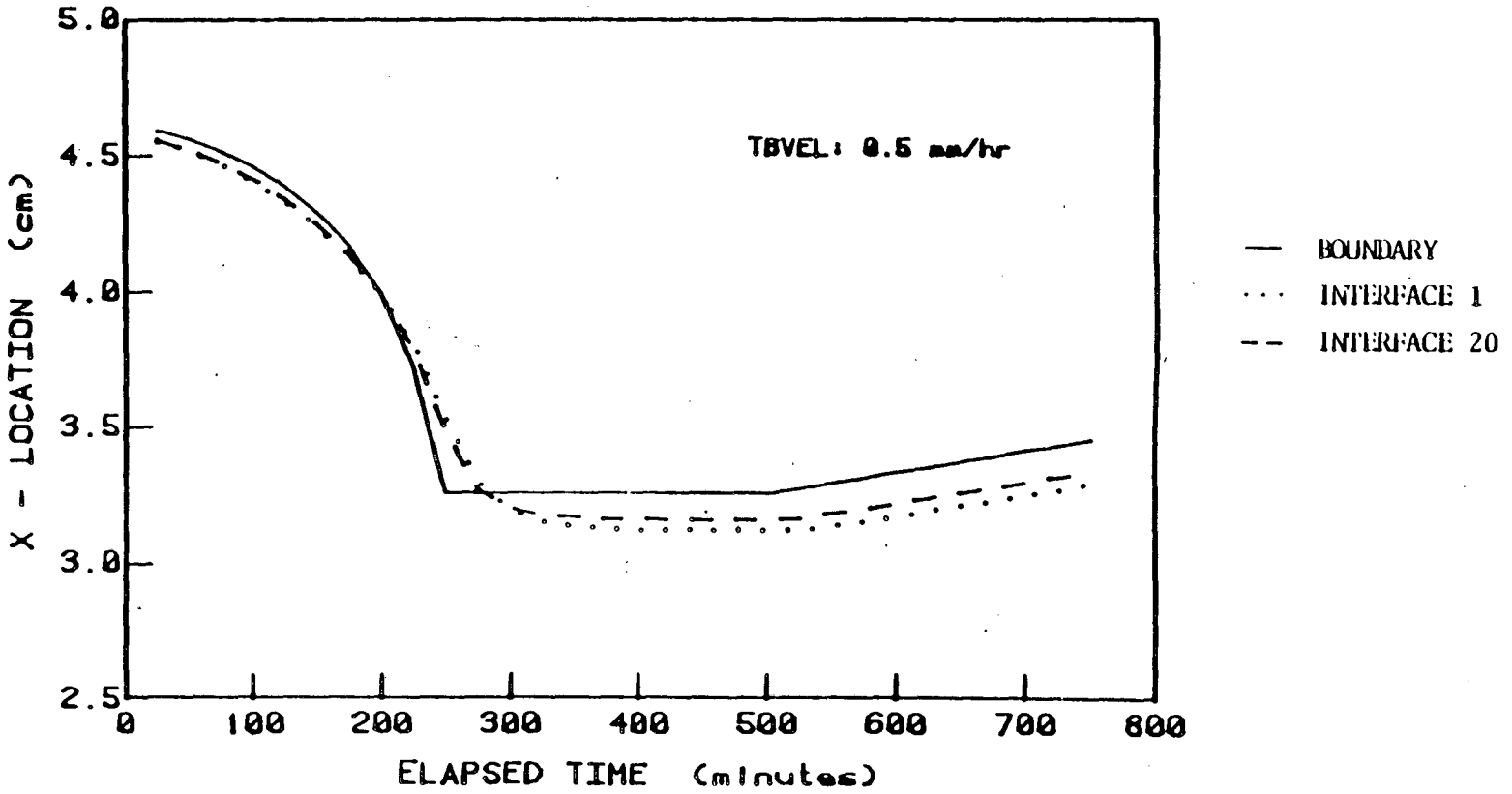


FIGURE 12d. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR AN IDEAL ANISOTROPIC BLANKET ORIENTED IN THE Y-DIRECTION

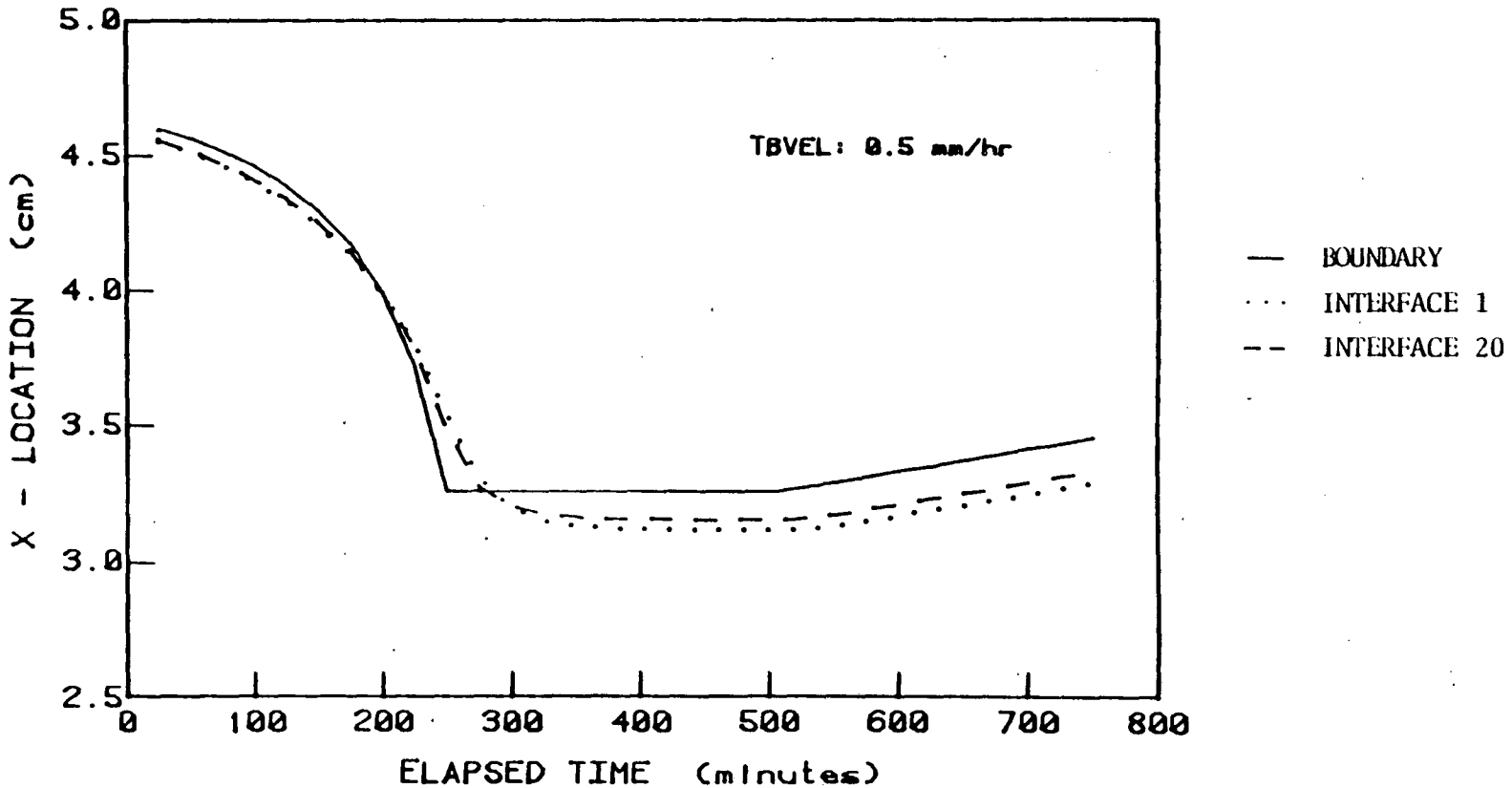


FIGURE 13a. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR AN ISOTROPIC (GRAPHITE) BLANKET

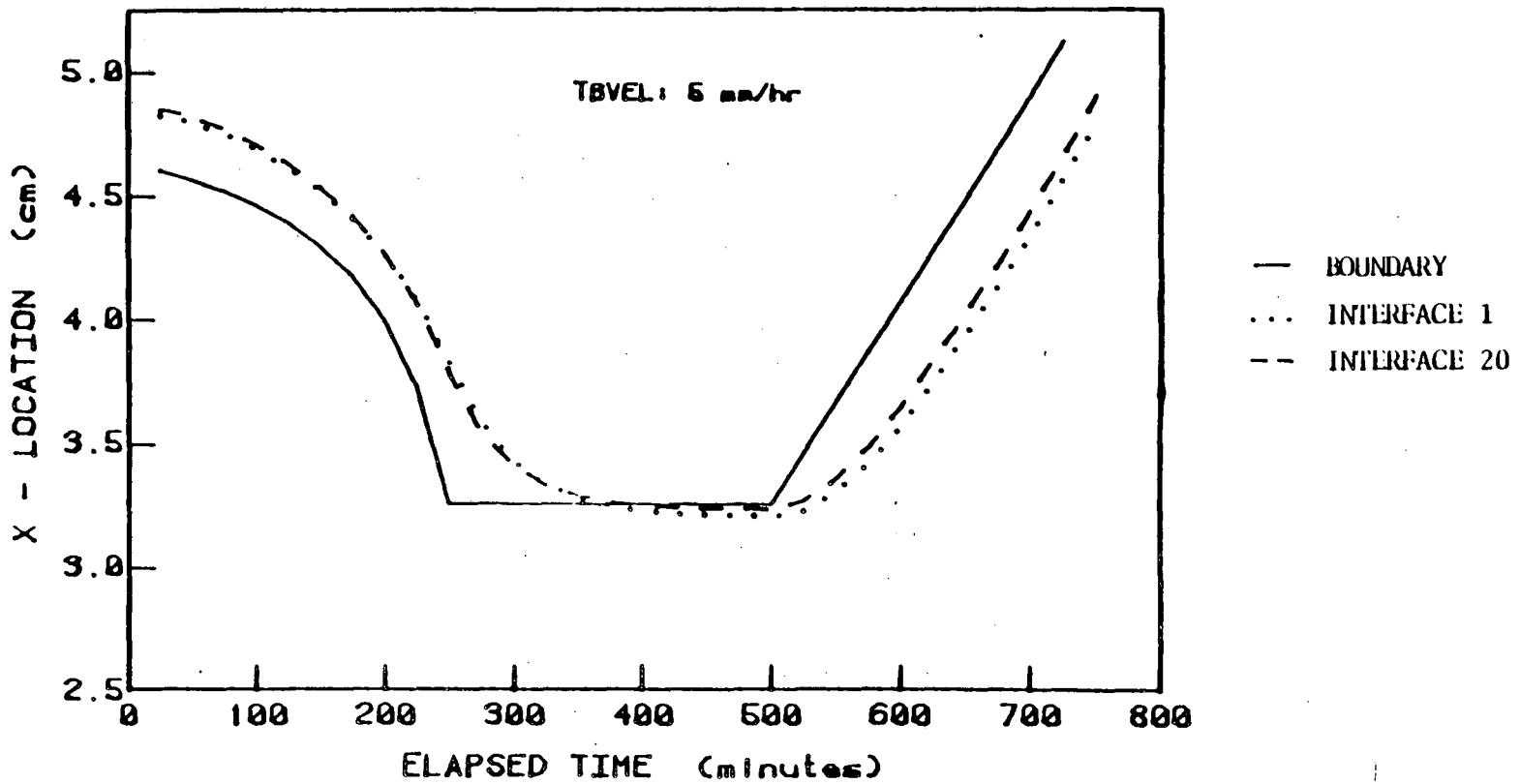


FIGURE 136. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR A BORON NITRIDE BLANKET ORIENTED IN THE Y-DIRECTION

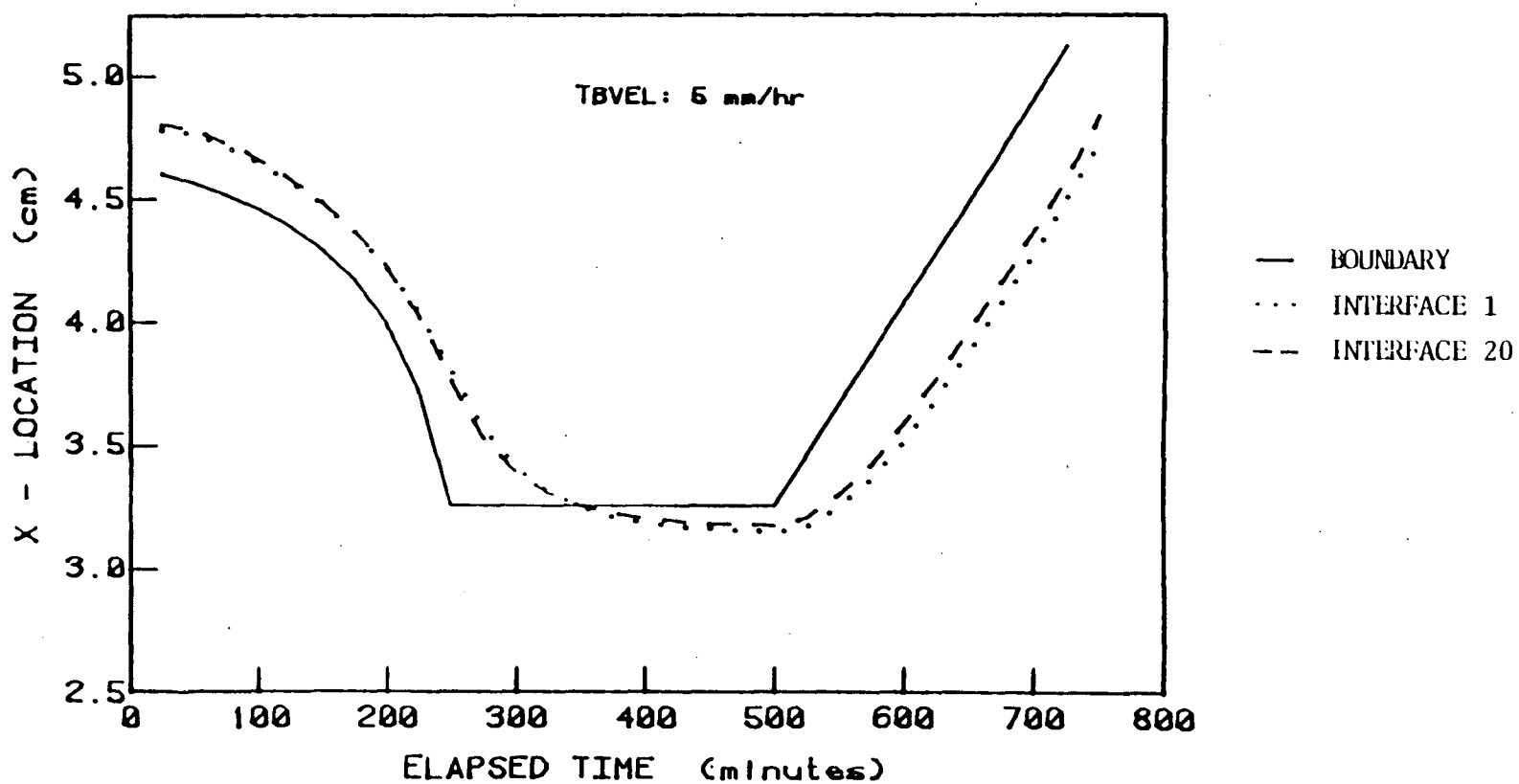


FIGURE 13c. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR A PYROLYTIC GRAPHITE BLANKET ORIENTED IN THE Y-DIRECTION

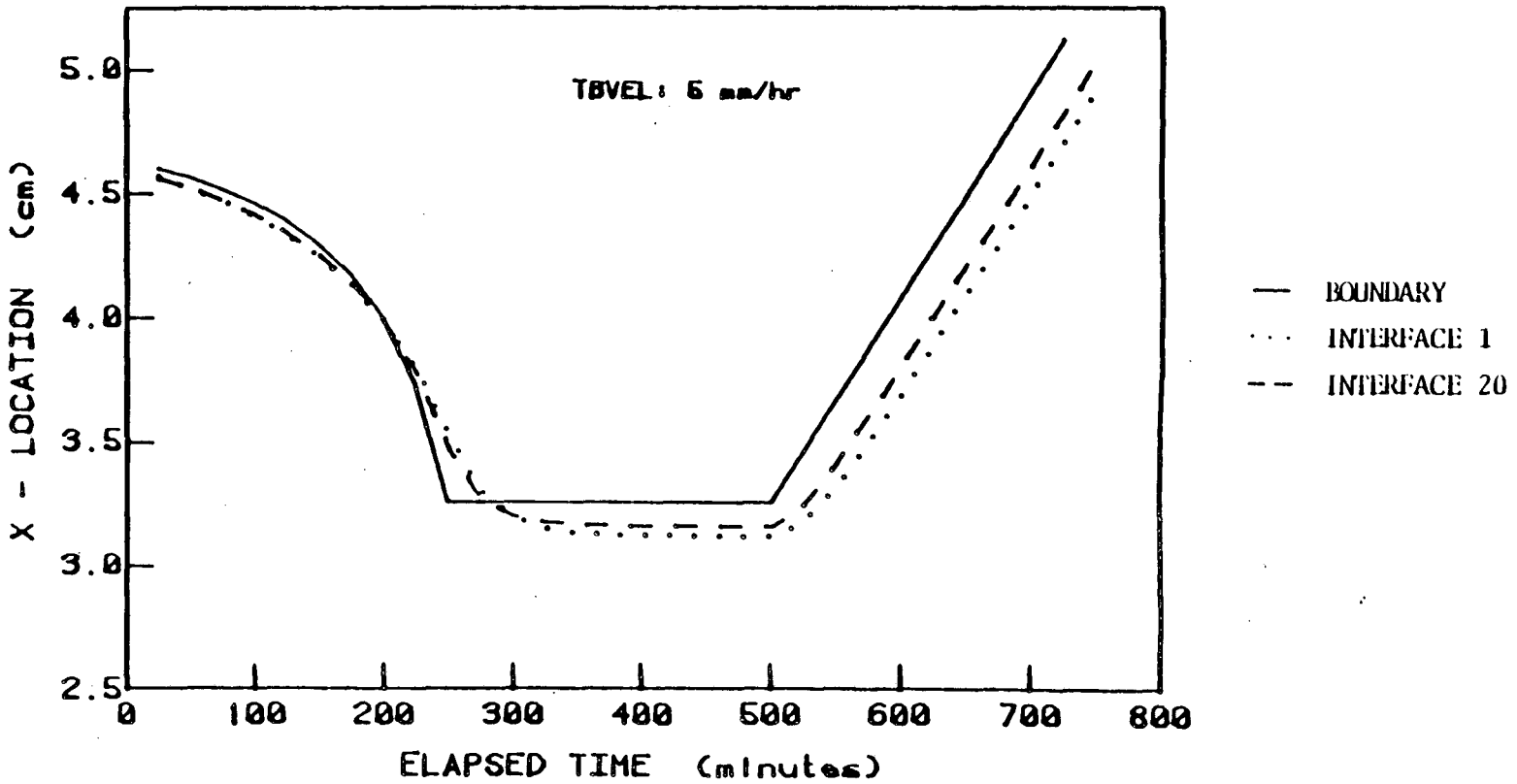


FIGURE 14a. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR A BORON NITRIDE BLANKET ORIENTED IN THE X-DIRECTION

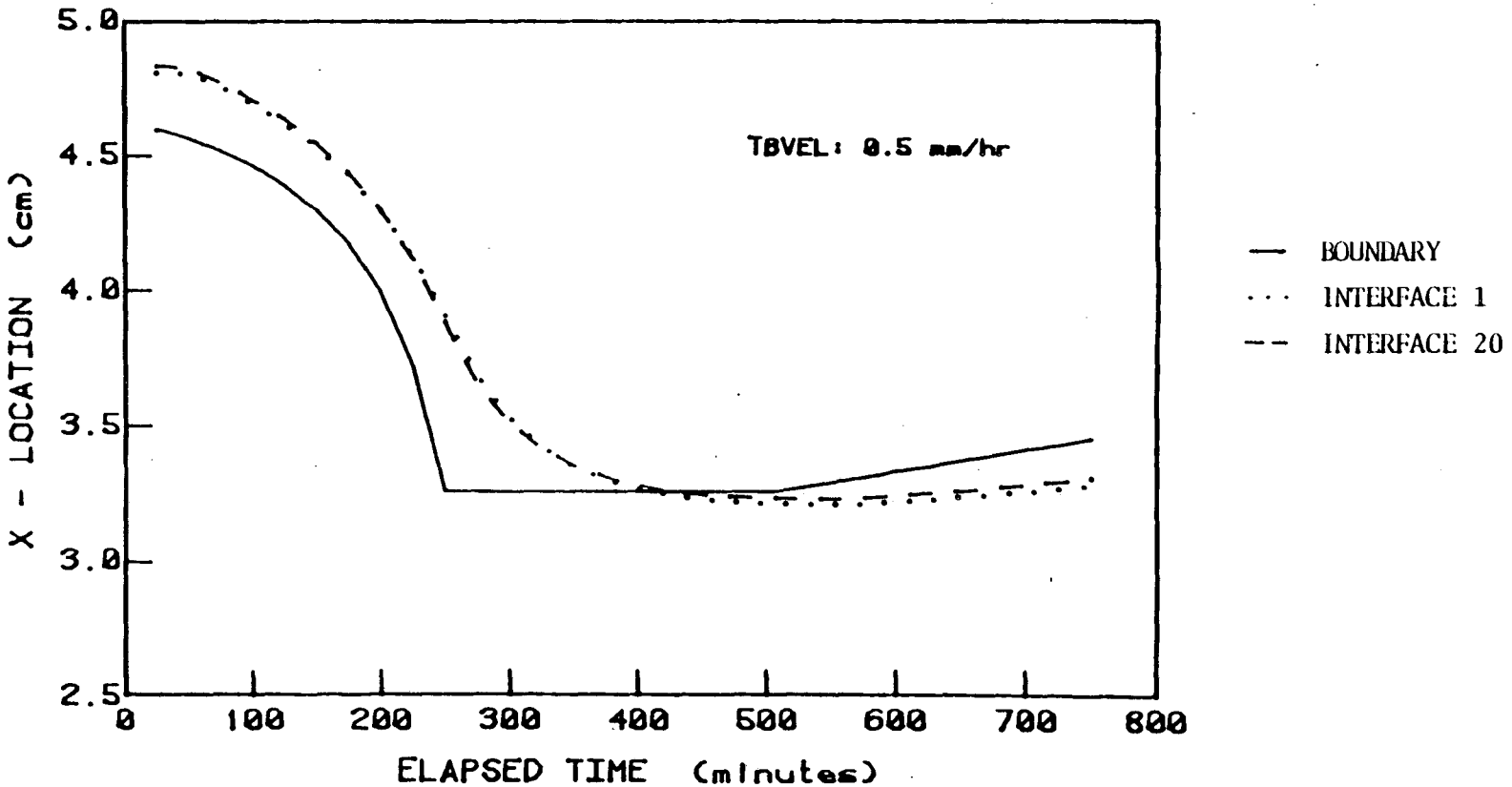


FIGURE 14b. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR A PYROLYTIC GRAPHITE BLANKET ORIENTED IN THE X-DIRECTION

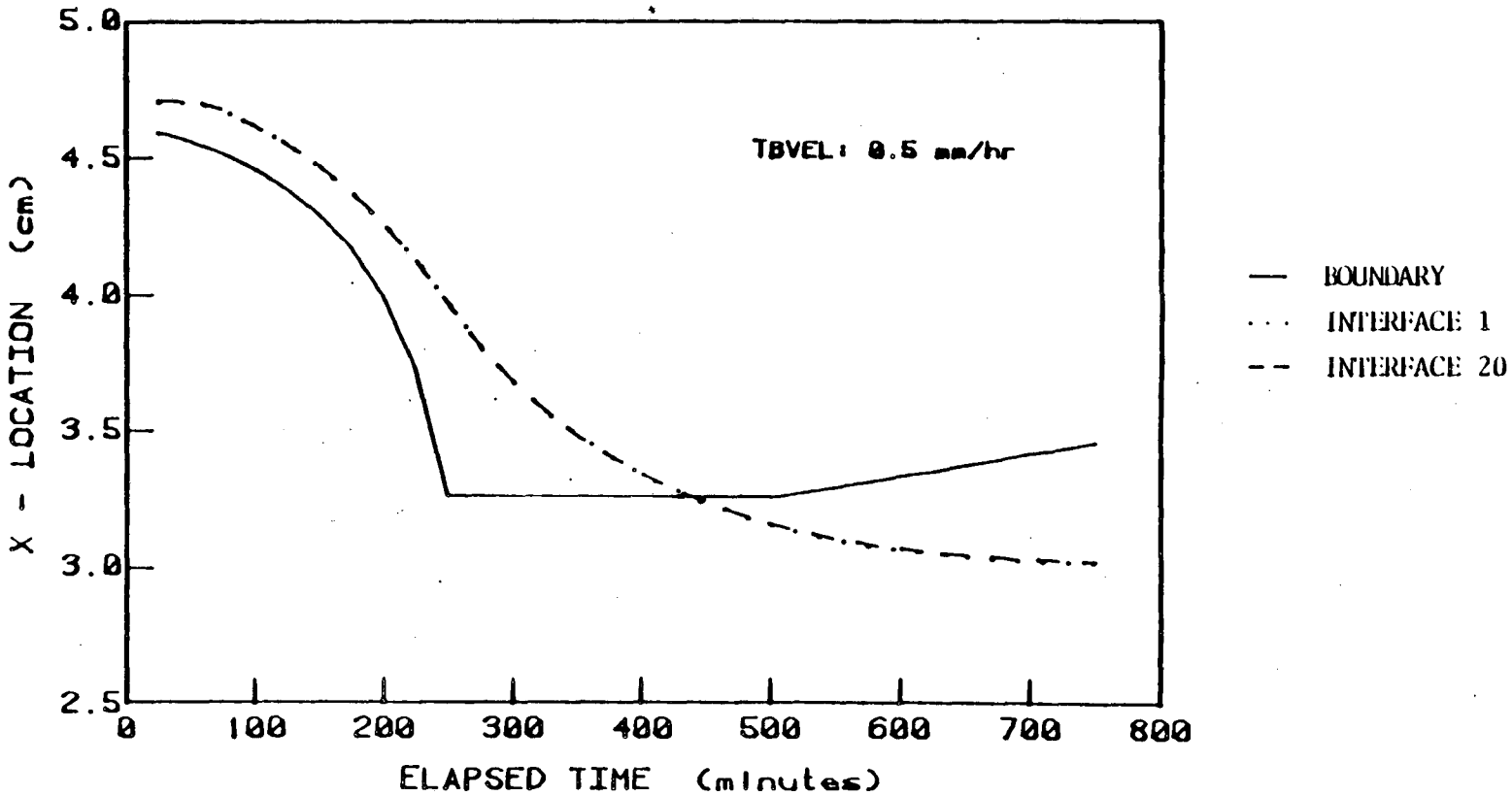


FIGURE 14c. LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR AN IDEAL ANISOTROPIC BLANKET ORIENTED IN THE X-DIRECTION

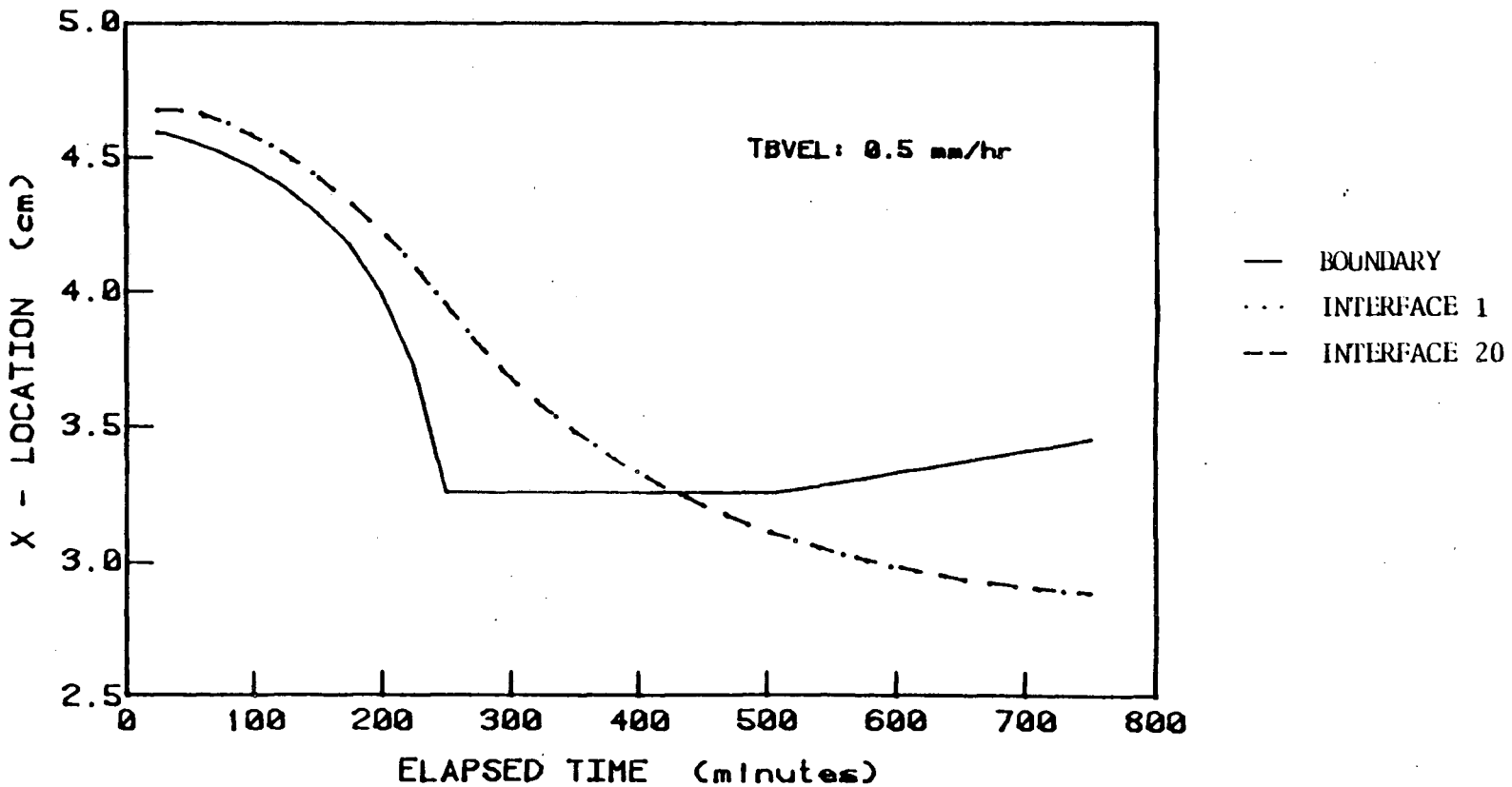


Figure 15. Experimental interface shape for a seed crystal used in Horizontal Bridgeman growth of Gallium Arsenide, courtesy of the Center for Advanced Materials, Lawrance Berkeley Laboratory.

FIGURE 16 : INTERFACE LOCATIONS DURING THE SEEDING PROCESS, WITH A BORON NITRIDE BLANKET

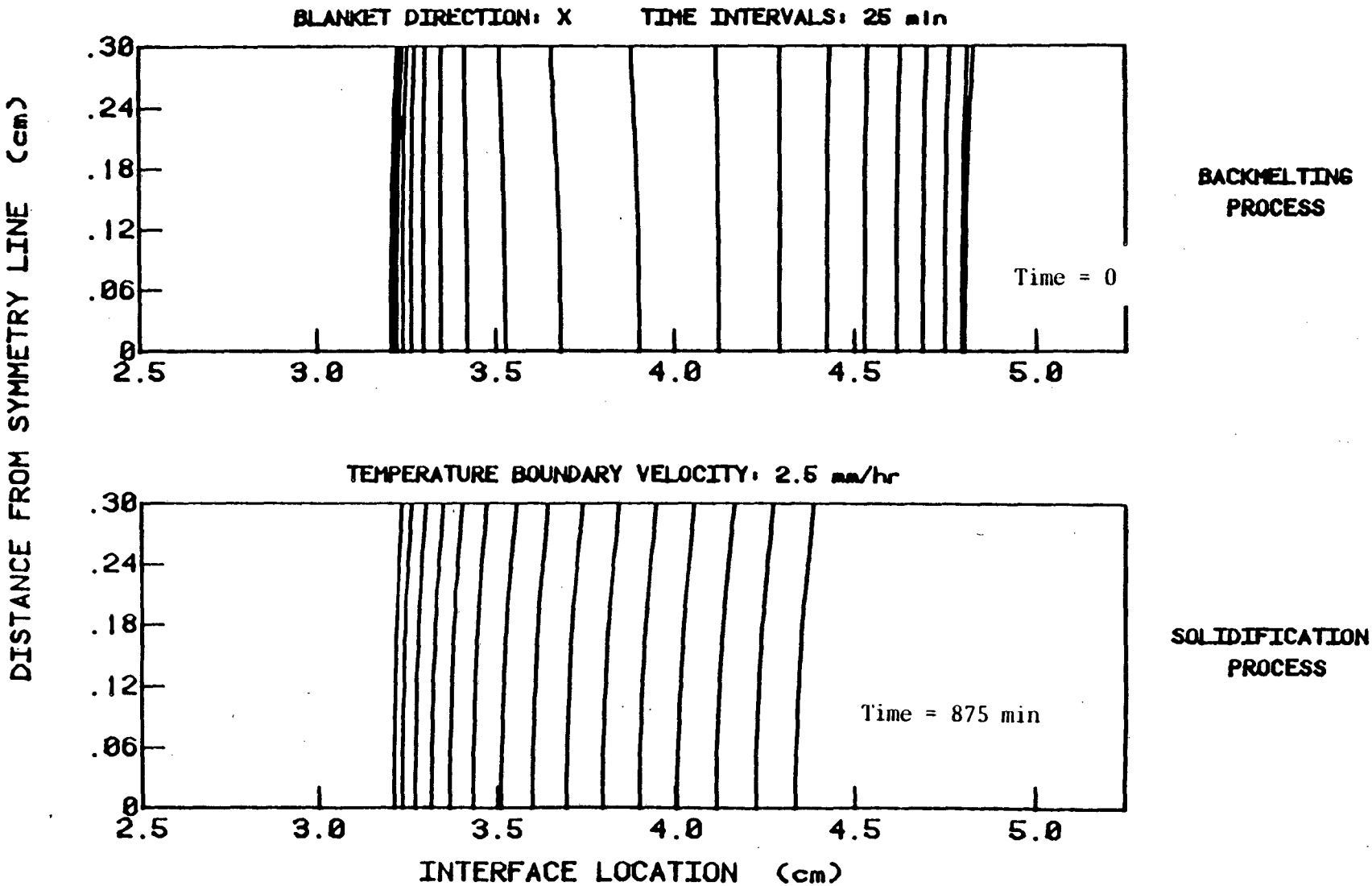
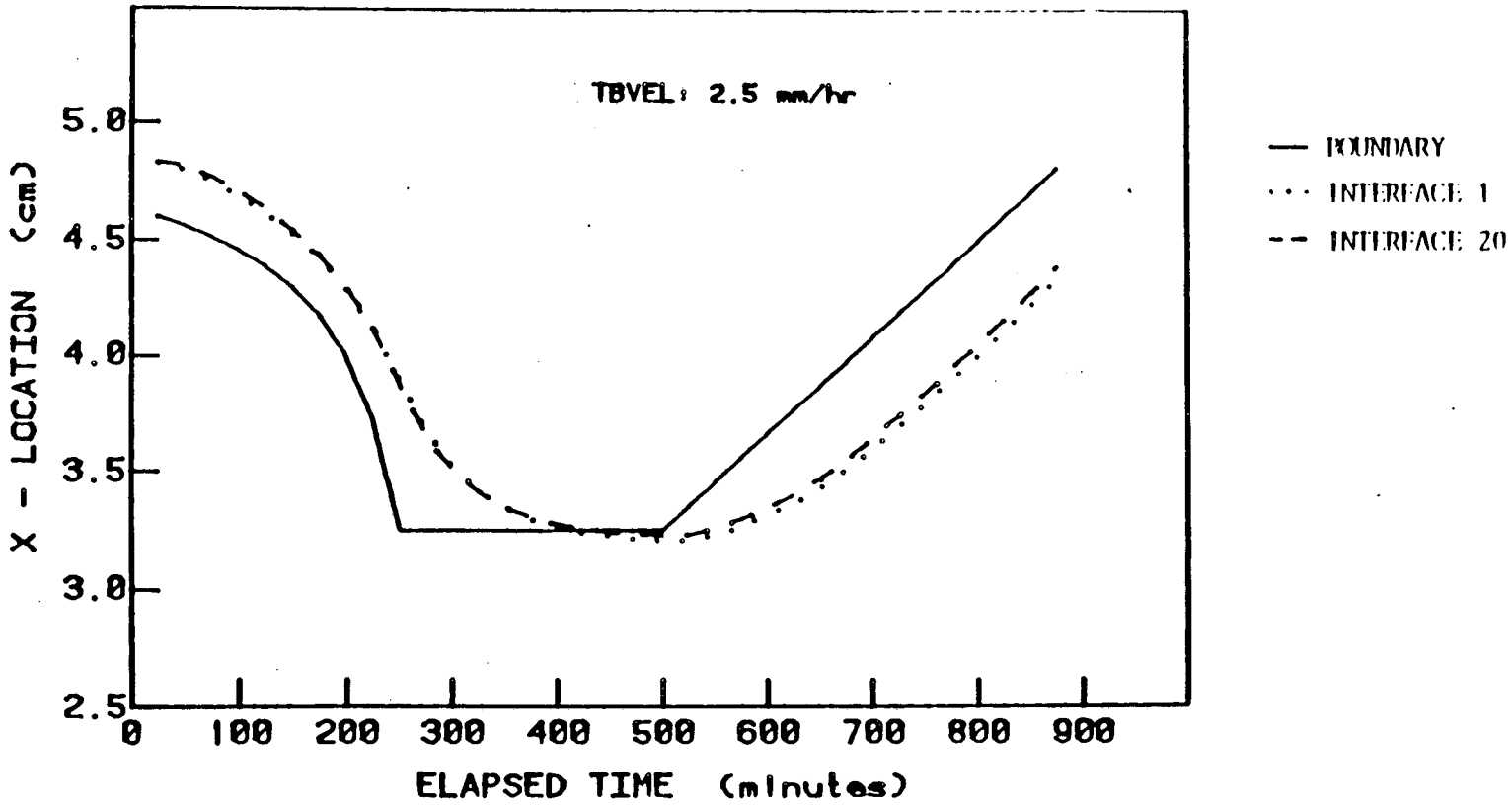


FIGURE 17: LOCATION OF THE CHANGE OF PHASE TEMPERATURE VERSUS TIME, ON THE BOUNDARY AND INTERFACE, FOR AN BORON NITRIDE BLANKET ORIENTED IN THE X-DIRECTION



## REFERENCES

1. A.S. Jordan, Journal of Crystal Growth, V49. (1980) 631

2. L. Hollan, The Preparation of Gallium Arsenide

3. Test data from the Center for Advanced Materials at Lawrance Berkeley Laboratory (1985)

4. J. Yoo, B. Rubinsky, Lawrance Berkeley Laboratory report under DOE contract DE-AC03-76SF00098 (1984)

5. A.S. Jordan, R. Caruso, A.R. VonNeida, J.W. Nielsen, Journal of Applied Physics, V52 N5 (1981)3331

6. J. Yoo, B. Rubinsky, Numerical Heat Transfer, V6 (1983)

7. S.M.Sze, Physics of Semiconductor Devices, 2nd Edition John Wiley and Sons

8. A. Bar-Lev, Semiconductors and Electronic Devices, 2nd Edition Prentice Hall

9. K.H. Huebner, The Finite Element Method for Engineers, John Wiley and Sons

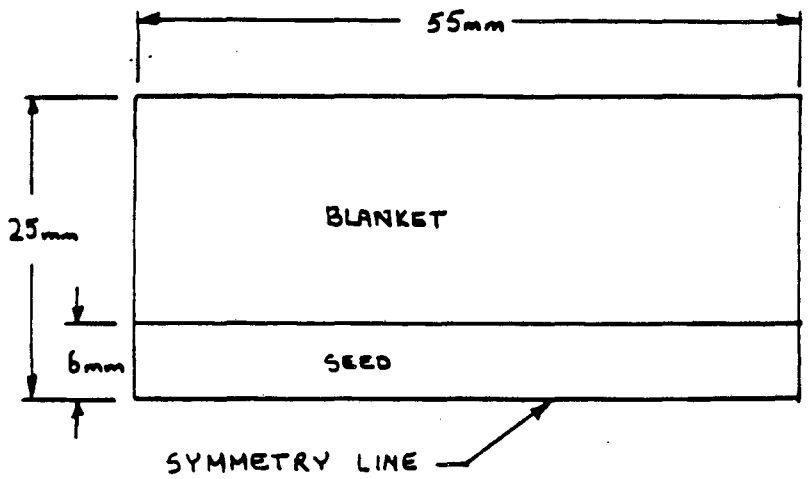
## ACKNOWLEDGMENT

This work was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Materials Sciences Division of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

# APPENDIX

A. Physical dimensions of the seed crystal used in this study

B. Listing of the computer code




PHYSICAL DIMENSIONS USED IN ANALYSIS

FILE: SOLID6  FORTRAN  A  VM/SP CMS RELEASE 3.1, CFD - U.C. BERKELEY

```
C         PROGRAM SOLID6 IS A MODIFIED VERSION OF DR. J. YOO'S          SOL
C         TWO DIMENSIONAL FINITE ELEMENT CODE TO STUDY SOLIDIFICATION   SOL
C         DURING THE BRIDGEMAN CRYSTAL PROCESS. THE GEOMETRY IN THIS    SOL
C         VERSION MODELS AN ANISOTROPIC MEDIA WHICH SURROUNDS THE       SOL
C         QUARTZ AMPULE WITHIN A FURNACE. AN ANALYSIS IS MADE ON        SOL
C         THE EFFECT OF THIS 'BLANKET' ON BOTH THE INTERFACE SHAPE      SOL
C         AND THE TEMPERATURE DISTRIBUTION IN THE CRYSTAL.              SOL
C         IN ADDITION, THIS VERSION MODELS THE INITIAL MELTING          SOL
C         AND SUBSEQUENT FREEZING OF THE SEED CRYSTAL MORE ACCURATELY   SOL
C         THAN VERSION 5.                                               SOL
C                                                                       SOL
C*********************************************************************** SOL
C                                                                       SOL
C         VARIABLE LIST                                                 SOL
C                                                                       SOL
C         UNLESS OTHERWISE NOTED, VARIABLE SUBSCRIPTS 1,2,3 REFER TO    SOL
C         SOLID PHASE, LIQUID PHASE AND BLANKET RESPECTIVELY.  VARIABLES SOL
C         FLAGED WITH AN ASTERIS (*) ARE INPUT VARIABLES.  UNITS OF THE SOL
C         INPUT VARIABLES MUST BE CONSISTENT (FOR INSTANCE ALL IN SI).  SOL
C                                                                       SOL
C         A ..........MAIN STORAGE ARRAY FOR PROGRAM VARIABLES.  POINTER SOL
C                     POSITIONS ARE CALCULATED IN MAIN.  THIS STORAGE   SOL
C                     METHOD SIMPLIFIES CHANGING THE PROBLEM SIZE SINCE SOL
                      ONLY ONE ARRAY DIMENSION NEED BE ALTERED.         SOL
C                                                                       SOL
C       * CAPA ......SPECIFIC HEAT                                      SOL
C       * CONDK .....THERMAL CONDUCTIVITY                               SOL
C       * COND3 .....THERMAL CONDUCTIVITY OF THE BLANKET MATERIAL.  ELEMESOL
C                     OF THIS ARRAY REPRESENT THE BLANKET CONDUCTIVITY MATSOL
C       * DTIME .....TIME STEP SIZE                                     SOL
C         GRAD ......GRADIENT OF THE BOUNDARY TEMPERATURE PROFILE.  THIS SOL
C                     IS CALCULATED IN THE FUNCTION "TBOUND"            SOL
C         HASPEC ....ASPECT RATIO, DEFINED AS THE RATIO OF Y-DIMENSION TOSOL
C                     X-DIMENSION OF THE SEED CRYSTAL                   SOL
C       * HEATSL ....LATENT HEAT OF FUSION FOR THE SEED MATERIAL        SOL
C       * INIFLG ....INDICATES THE TYPE OF INITIAL CONDITIONS           SOL
C                     THAT WILL BE APPLIED. OPTIONS ARE:                SOL
C                            0 - LINEAR IN THE X-DIRECTION WITH         SOL
C                                HIGH LOW LIMITS GIVEN BY THETAH AND    SOL
C                                THETAI. THE Y DIRECTION IS UNIFORM     SOL
C                            1 - UNIFORM THROUGHOUT THE PHASES.  ON THE SOL
C                                SOLID SIDE OF THE INTERFACE THE VALUE  SOL
C                                IS SET AT THETAI.  ON THE LIQUID SIDE  SOL
C                                THE VALUE IS SET TO THETAH             SOL
C         IPHBC .....INDICATES THE DIRECTION OF CONSTRAINED NODAL       SOL
C                     DISPLACEMENT FOR NODES OF THE INTERFACE.  IT IS   SOL
C                     GIVEN AS A ROW LOCATION IN THE GLOBAL STIFFNESS   SOL
C                     MATRIX.  IN THIS PROBLEM, INTERFACE NODES ON THE  SOL
C                     SYMMETRY LINE AND BLANKET BOUNDARY ARE CONSTRAINED SOL
C                     TO MOVE ONLY IN THE X-DIRECTION.                  SOL
C       * KBOUN .....FLAG TO DETERMINE THE METHOD FOR ADVANCING THE    SOL
C                     BOUNDARY TEMPERATURE PROFILE.  ORIGINAL OPTIONS   SOL
C                     WERE   0 - MOVE PROFILE AT A SPECIFIED VELOCITY   SOL
C                            1 - MOVE PROFILE AT THE SOLIDIFICATION RATE SOL
C                                OF THE INTERFACE NODE CLOSEST TO BLANKET SOL
```

```
C                          2 - MOVE PROFILE AT THE SOLIDIFICATION RATE  SOL
C                              OF THE INTERFACE NODE ON SYMMETRY LINE   SOL
C                    NOTE THAT IN VERSION SOLID6 ONLY OPTION 0 IS AVAILABSOL
C      * LXMAX .....MAXIMUM NUMBER OF NODES IN THE X DIRECTION THAT THE SOL
C                    "MESH" SUBROUTINE WILL GENERATE                     SOL
C        NA ........MAIN STORAGE ARRAY FOR INTEGER NUMBERS.  POINTER     SOL
C                    POSITIONS ARE CALCULATED IN MAIN.                   SOL
C        NBOUND ....NUMBER OF NODES ON THE SOLID-LIQUID INTERFACE DURINGSOL
C                    TIMESTEP PRECEEDING THE CURRENT TIME STEP           SOL
C      * NCNTUR ....THE INTERVAL IN NUMBER OF TIME STEPS FOR OUTPUTING  SOL
C                    TEMPERATURE DATA FOR CONTOUR PLOTTING               SOL
C      * NCBOUN ....NUMBER OF NODES ON THE INTERFACE IN THE CURRENT     SOL
C                    TIME STEP                                           SOL
C        NOD .......TOTAL NUMBER OF NODES LESS THOSE ON THE INTERFACE, TSOL
C                    EXIST ON A GIVEN SIDE OF THE INTERFACE DURING THE   SOL
C                    CURRENT TIME STEP.  THIS INCLUDES BOTH BLANKET AND  SOL
C                    SEED NODES ON EITHER SIDE OF THE INTERFACE          SOL
C        NODEMX ....TOTAL NUMBER OF NODES THAT EXISTS IN THE ENTIRE     SOL
C                    SOLUTION SPACE IN THE CURRENT TIME STEP             SOL
C        NODMXO ....THIS IS NODEMX OF THE PREVIOUS TIME STEP             SOL
C        NODO ......THIS IS NOD OF THE PREVIOUS TIME STEP                SOL
C        NOMAX .....MAXIMUM NUMBER OF NODES THAT "MESH" WILL GENERATE FOSOL
C                    EITHER SIDE OF THE INTERFACE                        SOL
C        NPHBC .....NUMBER OF INTERFACE NODES THAT ARE CONSTRAINED       SOL
C                    TO MOVE IN A PARTICULAR DIRECTION                   SOL
C      * NPLOT .....INTERVAL IN NUMBER OF TIME STEPS FOR OUTPUTING      SOL
C                    INTERFACE LOCATIONS FOR PLOTTING ROUTINES           SOL
C      * NPRINT ....INTERVAL IN NUMBER OF TIME STEPS FOR OUTPUTING      SOL
C                    ALL RESULTS TO A PRINT FILE                         SOL
C      * NTHC ......IDENTIFIES LOCATIONS ON THE BOUNDARY OF THE         SOL
C                    DOMAIN AND THE TYPE OF BOUNDARY CONDITIONS THAT     SOL
C                    WILL BE APPLIED THERE. ELEMENT POSITION IN THE      SOL
C                    MATRIX IDENTIFIES LOCATION.  THE CODE IS:           SOL
C                              1,1 - SOLID SIDE OF INTERFACE, ON YBOUND(3)SOL
C                              1,2 - SOLID SIDE OF INTERFACE, ON XBOUND(1)SOL
C                              1,3 - SOLID SIDE OF INTERFACE, ON YBOUND(1)SOL
C                              2,1 - LIQUID SIDE, ON YBOUND(3)          SOL
C                              2,2 - LIQUID SIDE, ON XBOUND(2)          SOL
C                              2,3 - LIQUID SIDE, ON XBOUND(1)          SOL
C                    ELEMENT VALUE IDENTIFIES BOUNDARY CONDITION TYPE.  SOL
C                    THE CODE IS:                                       SOL
C                              1 -   INSULATED                          SOL
C                              2 -   TEMPERATURE                        SOL
C                              3 -   NATURAL                            SOL
C                    NOTE, IN SOLID6 NATURAL BC'S ARE NOT AN OPTION     SOL
C      * NTEND .....NUMBER OF TIME STEPS IN THE PROCESS. INDEX 1 REFERS SOL
C                    TO THE BACKMELTING STAGE, INDEX 2 REFERS TO THE    SOL
C                    STABILIZING STAGE AND INDEX 3 REFERS TO THE        SOL
C                    SOLIDIFICATION STAGE                               SOL
C        NTOTL1 ....DIMENSION OF THE 'A' ARRAY                          SOL
C        NTOTL2 ....DIMENSION OF THE 'NA' ARRAY                         SOL
C        NX ........NUMBER OF NODES IN THE X-DIRECTION ON A GIVEN SIDE  SOL
C                    OF THE INTERFACE, INCLUDING THE INTERFACE NODE,    SOL
C                    AT THE CURRENT TIME STEP.                          SOL
C        NXO .......THIS IS 'NX' OF THE PREVIOUS TIME STEP              SOL
```

```
C         NY ........NUMBER OF NODES IN THE Y-DIRECTION ON A GIVEN SIDE  SOL
C                    OF THE INTERFACE, AT THE CURRENT TIME STEP          SOL
C       * NYMAX .....MAXIMUM NUMBER OF NODES IN THE Y-DIRECTION          SOL
C         NYO .......THIS IS 'NY' OF THE PREVIOUS TIME STEP               SOL
C         PHVAL .....VALUE OF DISPLACEMENT FOR CONSTRAINED INTERFACE     SOL
C                    NODES.  IN THIS PROBLEM THE Y-DISPLACEMENTS         SOL
C                    OF INTERFACE NODES ON THE SYMMETRY LINE AND         SOL
C                    BLANKET INTERFACE ARE SET TO ZERO.                  SOL
C       * RHO .......DENSITY                                             SOL
C         RLENTH ....REFERENCE LENGTH FOR THE Y-DIRECTION (NOT USED)     SOL
C         RINCRE ....DISPLACEMENT OF THE INTERFACE NODE AT YBOUND(2)     SOL
C                    THAT RESULTS IN THE CURRENT TIME STEP               SOL
C       * SINI ......LOCATION OF THE MAXIMUM IMPOSED TEMPERATURE ON THE  SOL
C                    BLANKET BOUNDARY, AT TIME ZERO                      SOL
C       * TPVEL .....VELOCITY OF THE BOUNDARY TEMPERATURE PROFILE.       SOL
C                    IN AN EARLIER VERSION THE BOUNDARY PROFILE          SOL
C                    WAS MOVED IN THE BACKMELTING STAGE WITH A           SOL
C                    FIXED GRADIENT, AT A DIFFERENT VELOCITY             SOL
C                    THAN THE SOLIDIFICATION STAGE. THUS THE             SOL
C                    INDEXES 1 (BACKMELTING), 2 (STABLIZE) AND           SOL
C                    3 (SOLIDIFICATION). IN VERSION SOLID6 ONLY          SOL
C                    TPVEL(3) IS USED.                                   SOL
C       * THETAI ....INITIAL TEMPERATURE (IF UNIFORM CONDITIONS)         SOL
C       * THETAC ....COLD WALL TEMPERATURE (NOT USED)                    SOL
C       * THETAH ....MAXIMUM TEMPERATURE IMPOSED ON THE BOUNDARY         SOL
C       * THETA1 ....INITIAL TEMPERATURE (IF UNIFORM )                   SOL
C       * THET5F ....TEMPERATURE AT LOCATION XBOUND(1), YBOUND(3) AT THE SOL
C                    END OF THE BACKMELTING STAGE                        SOL
C       * UNITS .....INDICATES THE UNIT CONVENTION, SUCH AS 'SI'         SOL
C       * XBOUND ....DEFINES THE X BOUNDARIES OF THE DOMAIN.  THE        SOL
C                    SEED IS ASSUMMED TO LIE BETWEEN THE LINES   X=0     SOL
C                    AND  X= X2 . THEREFORE, XBOUND(1) ALWAYS EQUALS 0.  SOL
C         XREFER ....REFERENCE LENGTH USED BY 'MESH' TO ADD MORE NODES   SOL
C                    IN THE X-DIRECTION                                  SOL
C       * XINI ......INITIAL POSITION OF THE INTERFACE.  INDEXES 1,2,3   SOL
C                    REFER TO BOTTOM ( YBOUND(1) ), MIDDLE AND           SOL
C                    TOP ( YBOUND(2) ).  NOTE THE INITIAL INTERFACE      SOL
C                    SHAPE CAN BE STRAIGHT, LINEAR OR QUADRATIC AS       SOL
C                    DEFINED BY XINI                                     SOL
C       * YBOUND ....DEFINES THE Y BOUNDARIES OF THE DOMAIN.  THE SEED'S SOL
C                    SYMMETRY LINE IS ASSUMED TO LIE ON THE LINE         SOL
C                    Y = 0.  THUS YBOUND(1) IS ALWAYS ZERO.  INDEX 2     SOL
C                    IS THE UPPERMOST Y-BOUNDARY OF THE SEED.  THIS IS   SOL
C                    ALSO THE LOWER Y-BOUNDARY OF THE BLANKET.  INDEX 3  SOL
C                    IS THE UPPER Y-BOUNDARY OF THE BLANKET              SOL
C                    SHAPE CAN BE STRAIGHT, LINEAR OR QUADRATIC AS       SOL
C                    DEFINED BY XINI                                     SOL
C       * YBOUND ....DEFINES THE Y BOUNDARIES OF THE DOMAIN.  THE SEED'S SOL
C                    SYMMETRY LINE IS ASSUMED TO LIE ON THE LINE         SOL
C                    Y = 0.  THUS YBOUND(1) IS ALWAYS ZERO.  INDEX 2     SOL
C                    IS THE UPPERMOST Y-BOUNDARY OF THE SEED.  THIS IS   SOL
C                    ALSO THE LOWER Y-BOUNDARY OF THE BLANKET.  INDEX 3  SOL
C                    IS THE UPPER Y-BOUNDARY OF THE BLANKET.             SOL
C                                                                        SOL
C*************************************************************************SOL
```

```
C                                                                       SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE1/NOD(2), NX(2), NY(2),NODEMX,NYMAX                    SO
      COMMON/CODE2/NOBOUN                                               SO
      COMMON/CODE3/XBOUND(2),YBOUND(3),XREFER                           SO
      COMMON/CODE4/NOND(2),NXO(2),NYO(2),NBOUND,NODMXO                  SO
      COMMON/CODE6/HASPEC,THETAI,THETAC,TSAT,THETAH,THETSF              SO
      COMMON/CODE7/DTIME                                                SO
      COMMON/COD11/NTBC(2,3)                                            SO
      COMMON/COD12/CONDK(3),CAPA(3),RHO(3),HEATSL,COND3(2,2)            SO
      COMMON/COD14/PHVAL(2),NPHBC,IPHBC(2)                              SO
      COMMON/COD16/RLENTH                                               SO
      COMMON/COD19/NTEND(3),NPRINT,NPLOT,NCNTUR                         SO
      COMMON/COD20/LXMAX,UNITS                                          SO
      COMMON/RAD1/SINI,GRAD,TBVEL(3),INIFLG                             SO
      COMMON/POSI/XINI(3)                                               SO
      COMMON/BOUN/KBOUN                                                 SO
      COMMON A(80000)                                                   SO
      COMMON/DIREC/RINCRE                                               SO
      COMMON/JYOO/NA(15000)                                             SO
      CHARACTER*3 UNITS                                                 SO
C                                                                       SO
      NTOTL1= 80000                                                     SO
      NTOTL2= 15000                                                     SO
C                                                                       SO
      READ(5,*)UNITS                                                    SO
      READ(5,*)KBOUN                                                    SO
      READ(5,*)TBVEL(1),TBVEL(2),TBVEL(3)                               SO
      READ(5,*)NTEND(1),NTEND(2),NTEND(3),NPRINT,NPLOT,NCNTUR,INIFLG SO
      READ(5,*)LXMAX,NYMAX,NOBOUN                                       SO
      READ(5,*)THETAI,THETAC,TSAT,THETAH,THETSF                         SO
      READ(5,*)CONDK(1),CONDK(2),CAPA(1),CAPA(2),RHO(1),RHO(2),HEATSLSO
      READ(5,*)COND3(1,1),COND3(1,2),COND3(2,1),COND3(2,2)              SO
      READ(5,*)CAPA(3),RHO(3)                                           SO
      READ(5,*)DTIME                                                    SO
      READ(5,*)XBOUND(1),XBOUND(2),YBOUND(1),YBOUND(2),YBOUND(3)        SO
      READ(5,*)NTBC(1,1),NTBC(1,2),NTBC(1,3),                           SO
     1         NTBC(2,1),NTBC(2,2),NTBC(2,3)                            SO
      READ(5,*)XINI(1),XINI(2),XINI(3)                                  SO
      READ(5,*)SINI                                                     SO
C                                                                       SO
      NOMAX= LXMAX * NYMAX                                              SO
      NOMAX2= NOMAX * 2                                                 SO
      NPHBC= 2                                                          SO
      IPHBC(1)= 2                                                       SO
      IPHBC(2)= NOBOUN *2                                               SO
      PHVAL(1)= 0.000                                                   SO
      PHVAL(2)= 0.000                                                   SO
      HASPEC= (YBOUND(2)-YBOUND(1))/(XBOUND(2)-XBOUND(1))               SO
      XREFER= (XBOUND(2)-XBOUND(1)) /(LXMAX-3)                          SO
      RLENTH= HASPEC /(NOBOUN-3)/8.00                                   SO
      NYO(1)= NOBOUN                                                    SO
      NYO(2)= NOBOUN                                                    SO
C                                                                       SO
C     CALCULATE POINTER POSITIONS FOR MAIN STORAGE ARRAY                SO
```

```
      N1= 1                                                             SO
      N2= N1+ NOMAX2 * (NYMAX+2) *2                                     SO
      N3= N2+ NOMAX2 *2                                                 SO
      N4= N3+ NOMAX2 *2                                                 SO
      N5= N4+ NOMAX2 *2                                                 SO
      N6= N5+ NOMAX2 *2                                                 SO
      N7= N6+ NOMAX                                                     SO
      N8= N7+ NOMAX                                                     SO
      N9= N8+ NYMAX                                                     SO
      N10= N9+ NYMAX                                                    SO
      N11= N10+ NYMAX                                                   SO
      N12= N11+ NYMAX                                                   SO
      N13= N12+ NOBOUN *2                                               SO
      N14= N13+ NOMAX2                                                  SO
      N15= N14+ NOMAX2                                                  SO
      N16= N15+ NOMAX2                                                  SO
      N17= N16+ NOMAX                                                   SO
      N18= N17+ NOMAX                                                   SO
      N19= N18+ NYMAX                                                   SO
      N20= N19+ NYMAX                                                   SO
      N21= N20+ NOMAX                                                   SO
      N22= N21+ NOMAX                                                   SO
      N23= N22+ NOMAX                                                   SO
      N24= N23+ NOMAX2                                                  SO
      N25= N24+ NYMAX *2                                                SO
      N26= N25+ NOMAX                                                   SO
      N27= N26+ NOMAX
      N28= N27+ NOMAX                                                   SO
      IF(NTOTL1 .LT. N28)GO TO 8899                                     SO
C                                                                       SO
      NYMM1= NYMAX -1                                                   SO
      NXMM1= LXMAX -1                                                   SO
      NN1= 1                                                            SO
      NN2= NN1+ NYMM1 *NXMM1 *4                                         SO
      NN3= NN2+ NYMM1 *NXMM1 *4                                         SO
      NN4= NN3+ NYMM1 *NXMM1 *4                                         SO
      NN5= NN4+ NYMM1 *NXMM1 *4                                         SO
      IF(NTOTL2 .LT. NN5)GO TO 8889                                     SO
C                                                                       SO
      CALL INIXYP(A(N8),A(N9))                                          SO
      ICOUN= 1                                                          SO
      JOUN= 1                                                           SO
      KOUN= 1                                                           SO
      NTIME= 0                                                          SO
      DO 120 IPART=1,3                                                  SO
      DO 110 ICOUNT= 1,NTEND(IPART)                                     SO
         NTIME= NTIME +1                                                SO
         CALL PREP(A(N3),A(N4),A(N5),A(N8),A(N9),                       SO
     1             NA(NN1),NA(NN2),A(N13),A(N14),A(N15),                SO
     2             A(N18),A(N19),NA(NN3),NA(NN4),NYMM1,NXMM1,           SO
     3             NTIME)                                               SO
         CALL SOLVER(A(N1),A(N2),A(N3),A(N4),A(N5),A(N8),               SO
     1               A(N9),A(N10),A(N11),NA(NN1),NA(NN2),               SO
     2               A(N18),A(N19),A(N23),A(N24),                       SO
     3               NOMAX2,NYMM1,NXMM1,NTIME)                          SO
```

```
C                                                                       SOI
C         OUTPUT RESULTS AT SPECIFIED INTERVALS                         SOI
          IF(IDUM .EQ. NPRINT)THEN                                      SOI
              IDUM=  1                                                  SOI
              GO TO 77                                                  SOI
          ELSEIF(NTIME .LE. 3)THEN                                      SOI
              IDUM= IDUM +1                                             SO
              GO TO 77                                                  SO
          ELSE                                                          SO
              IDUM= IDUM + 1                                            SO
              GO TO 100                                                 SO
          ENDIF                                                         SO
 77       CALL POST(A(N3),A(N4),A(N5),A(N8),A(N9),A(N10),A(N11),NTIME)  SO
 100      IF(JDUM .EQ. NPLOT)THEN                                       SO
              CALL PLOTDT(A(NP),A(N9),NTIME)                            SO
              JDUM= 1                                                   SO
          ELSE                                                          SO
              JDUM= JDUM + 1                                            SO
          ENDIF                                                         SO
          IF(KDUM .EQ. NCNTUR)THEN                                      SO
              CALL CONTUR(A(N3),A(N4),A(N5),NTIME)                      SO
              KDUM= 1                                                   SO
          ELSE                                                          SO
              KDUM= KDUM + 1                                            SO
          ENDIF                                                         SO
 110      CONTINUE                                                      SO
 120      CONTINUE                                                      SO
          STOP                                                          SO
 8899     WRITE(6,9999)N2H,NTOTL1,NN5,NTOTL2                            SO
 9999     FORMAT('TOTAL DIMENSION IS SMALL'/T5,2(I10,3X)/T5,2(I10,3X))  SO
          STOP                                                          SO
          END                                                           SO
C                                                                       SO
C*********************************************************************** SO
          SUBROUTINE APLBC(GK,GF,NOMAX2)                                SO
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                           SO
          COMMON/CODE2/NOBOUN                                           SO
          COMMON/COD14/PHVAL(2),NPHBC,IPHBC(2)                          SO
          DIMENSION GK(NOMAX2,1),GF(1)                                  SO
          BIG= 1.0D30                                                   SO
          DO 10 I=1, NPHBC                                              SO
          DO 20 J=1,4                                                   SO
          GK(IPHBC(I),J) = 0.0D0                                        SO
          JI= IPHBC(I) -J +1                                            SO
          IF(JI .LE. 0)GO TO 20                                         SO
          GK(JI,J)= 0.0D0                                               SO
 20       CONTINUE                                                      SO
          GK(IPHBC(I),1)= BIG                                           SO
 10       GF(IPHBC(I))= BIG *PHVAL(I)                                   SO
          RETURN                                                        SO
          END                                                           SO
C                                                                       SO
C*********************************************************************** SO
          SUBROUTINE ASEMBL(GK,GF,NEL1,NEL2,NOMAX2,NYMM1,NXMM1,         SO
     1                      IK,EF,IN,JN,NP)                             SO
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                SOI
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                       SOI
      DIMENSION GK(NOMAX2,1), GF(1),NEL1(NYMM1,NXMM1,1),                 SOI
     1          EK(4,4),FF(4),NEL2(NYMM1,NXMM1,1)                        SOI
C                                                                        SOI
      JNG= NX(1) + NX(2) -1 -JN                                          SOI
      IDUM= NYMAX * (JNG-1) + IN                                         SOI
      DO 10 M=1,4                                                        SOI
      IF(NP .EQ. 1)THEN                                                  SO
          IG= NEL1(IN,JN,M)                                              SO
      ELSE                                                               SO
          IF(M.EQ.1)THEN                                                 SO
              IG= IDUM                                                   SO
          ELSEIF(M.EQ.2)THEN                                             SO
              IG= IDUM + NYMAX                                           SO
          ELSEIF(M.EQ.3)THEN                                             SO
              IG= IDUM + NYMAX +1                                        SO
          ELSEIF(M.EQ.4)THEN                                             SO
              IG= IDUM +1                                                SO
          ENDIF                                                          SO
      ENDIF                                                              SO
      GF(IG)= GF(IG) + FF(M)                                             SO
      DO 11 N=1,4                                                        SO
      IF(NP .EQ. 1)THEN                                                  SO
          J= NEL1(IN,JN,N)                                               SO
      ELSE                                                               SO
          IF(N.EQ.1)THEN                                                 SO
              J= IDUM                                                    SO
          ELSEIF(N.EQ.2)THEN                                             SO
              J= IDUM + NYMAX                                            SO
          ELSEIF(N.EQ.3)THEN                                             SO
              J= IDUM + NYMAX +1                                         SO
          ELSEIF(N.EQ.4)THEN                                             SO
              J= IDUM +1                                                 SO
          ENDIF                                                          SO
      ENDIF                                                              SOI
      JG= J - IG + 1                                                     SOI
      IF(JG .LE. 0)GO TO 11                                              SOI
      GK(IG,JG)= GK(IG,JG) + EK(M,N)                                     SOI
 11   CONTINUE                                                           SOI
 10   CONTINUE                                                           SO
      RETURN                                                             SO
      END                                                                SO
C                                                                        SO
C*********************************************************************** SO
      SUBROUTINE ASEMPF(GF,EF,IP,JP)                                     SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                SO
      DIMENSION GF(1), EF(4)                                             SO
      DO 10 M=1,4                                                        SO
      IF(M .GE. 3)THEN                                                   SO
            IG= 2*(JP -1) + M - 2                                        SO
         ELSE                                                            SO
            IG= 2*(IP -1) + M                                            SO
      ENDIF                                                              SO
 10   GF(IG)= GF(IG) + EF(M)                                             SO
```

```
      RETURN                                                            SC
      END                                                               SC
C                                                                       SC
C*********************************************************************** SC
      SUBROUTINE ASEMPK(GK,NOMAX2,HSH,IP,JP)                            SC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SC
      DIMENSION GK(NOMAX2,1), HSH(4,4)                                  SC
      DO 10 M=1,4                                                       SC
      IF(M .GE. 3) THEN                                                 SC
            IG= 2*(JP -1) +M -2                                         SC
         ELSE                                                           SC
            IG= 2*(IP -1) + M                                           SC
      ENDIF                                                             SC
      DO 10 N=1,4                                                       SC
      IF(N .GE. 3)THEN                                                  SC
            J= 2*(JP-1) +N -2                                           SC
         ELSE                                                           SC
            J= 2*(IP-1) + N                                             SC
      ENDIF                                                             SC
      JG= J - IG + 1                                                    SC
      IF(JG .LE. 0)GO TO 10                                             SC
      GK(IG,JG)= GK(IG,JG) + HSH(M,N)                                   SC
 10   CONTINUE                                                          SC
      RETURN                                                            SC
      END                                                               SC
C                                                                       SC
C*********************************************************************** SC
      SUBROUTINE CLEAR(GK,GF,NOMAX2,M,N)                                SC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SC
      DIMENSION GK(NOMAX2,1), GF(1)                                     SC
      DO 10 I=1,M                                                       SC
      GF(I)= 0.0D0                                                      SC
      DO 10 J=1,N                                                       SC
 10   GK(I,J)= 0.0D0                                                    SC
      RETURN                                                            SC
      END                                                               SC
C                                                                       SC
C*********************************************************************** SC
      SUBROUTINE CONTEM(GK,GF,NEL1,NEL2,NOMAX2,NYMM1,NXMM1,             SC
     1                  IN,JN,XX,TTM1,NP,NTIME)                         SC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SC
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SC
      COMMON/CODE7/DTIME                                                SC
      COMMON/CODE12/CONDK(3),CAPA(3),RHO(3),HEATSL,COND3(2,2)           SC
      DIMENSION GK(NOMAX2,1), GF(1), NEL1(NYMM1,NXMM1,1),               SC
     1          XG(2), WG(2), SMOMEN(4,4), SDIF(4,4),                   SC
     2          H(4), B(2,4), XX(2,4), TTM1(4), EK(4,4), EF(4),         SC
     3          NEL2(NYMM1,NXMM1,1),RKB(2,4)                            SC
      DATA XG/-.57735026918962600,.57735026918962600/                   SC
      DATA WG/1.00000000000,1.00000000000/                              SC
C                                                                       SC
C                                                                       SC
      IF(NP.EQ.1)THEN                                                   SC
         IF(IN.GE.NY(NP))THEN                                           SC
            CAP= CAPA(3)*RHO(3)                                         SC
```

```
                ELSE                                                    SO
                     RAT= CONDK(1)                                      SO
                     CAP= CAPA(1)*RHO(1)                                SO
                ENDIF                                                   SO
           ELSE                                                         SO
                IF(IN.GE.NY(NP))THEN                                    SO
                     CAP= CAPA(3)*RHO(3)                                SO
                ELSE                                                    SO
                     RAT= CONDK(2)                                      SO
                     CAP= CAPA(2)*RHO(2)                                SO
                ENDIF                                                   SO
           ENDIF                                                        SO
C                                                                       SO
C          INITIALIZE                                                   SO
           DO 10 I=1,4                                                  SO
           DO 10 J=1,4                                                  SO
           SMOMEN(I,J)= 0.0D0                                           SO
 10        SDIF(I,J)= 0.0D0                                             SO
C                                                                       SO
C          BEGIN NUMERICAL INTEGRATIONS                                 SO
           DO 60 LX= 1,2                                                SO
           RI= XG(LX)                                                   SO
           DO 60 LY= 1,2                                                SO
           SI= XG(LY)                                                   SO
           CALL STDM(RI,SI,H,B,DET,XX)                                  SO
           WT= WG(LX) * WG(LY) * DET                                    SO
C                                                                       SO
C          CALCULATE THE MATRIX PRODUCT  'COND3 * B'                    SO
           IF(IN .GE. NY(NP))THEN                                       SO
              DO 20 I=1,2                                               SO
              DO 20 J=1,4                                               SO
 20           RKB(I,J)= 0.D0                                            SO
              DO 44 N=1,2                                               SO
              DO 33 J=1,4                                               SO
              DO 22 M=1,2                                               SO
 22           RKB(N,J)= RKB(N,J) + COND3(N,M)*B(M,J)                    SO
 33           CONTINUE                                                  SO
 44           CONTINUE                                                  SO
           ENDIF                                                        SO
C                                                                       SO
C          CALCULATE THE MATRIX PRODUCTS  'H * HT' AND  'BT * K * B'    SO
           DO 50 I= 1,4                                                 SO
           DO 50 J= 1,4                                                 SO
           SMOMEN(I,J)= SMOMEN(I,J) + WT * H(I) * H(J) *CAP             SO
           DO 50 M= 1,2                                                 SO
           IF(IN .GE. NY(NP))THEN                                       SO
              SDIF(I,J)= SDIF(I,J) + B(M,I)*RKB(M,J) *WT                SO
           ELSE                                                         SO
              SDIF(I,J)= SDIF(I,J) + B(M,I)*B(M,J) *WT *RAT             SO
           ENDIF                                                        SO
 50        CONTINUE                                                     SO
 60        CONTINUE                                                     SO
C          END OF NUMERICAL INTEGRATION                                 SO
C                                                                       SO
C          ELEMENT K AND F MATRICES                                     SO
```

```
      DO 70 I= 1,4                                                      SOI
      DO 70 J= 1,4                                                      SOI
 70   EK(I,J)= SMOMEN(I,J) /DTIME + SDIF(I,J)                           SOI
      DO 71 I=1,4                                                       SOI
      STIFF= 0.0D0                                                      SOI
      DO 72 J= 1,4                                                      SOI
 72   STIFF= STIFF + SMOMEN(I,J) * ITM1(J) /DTIME                       SOI
 71   EF(I)= STIFF                                                      SOI
C                                                                       SOI
      CALL ASEMBL(GK,GF,NEL1,NEL2,NCMAX2,NYMM1,NXMM1,                   SOI
     1            EK,EF,IN,JN,NP)                                       SOI
      RETURN                                                            SOI
      END                                                               SOI
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE CONTUR(X,Y,THETA,NTIME)                                SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SO
      COMMON/CODE7/DTIME                                                SO
      DIMENSION X(2,1), Y(2,1), THETA(2,1)                              SO
C                                                                       SO
C     THIS ROUTINE OUTPUTS TEMPERATURE DATA TO A FILE FOR LATER         SO
C     PLOTTING OF TEMPERATURE CONTOURS.  NOTE THAT TIME HAS BEEN        SO
C     CHANGED TO MINUTES FOR OUTPUT TO THIS FILE                        SO
      TIME= NTIME * DTIME /60.D0                                        SO
      WRITE(8,100)TIME,NODEMX                                           SO
      DO 10 I=1,NOD(1)*NYMAX                                            SO
 10   WRITE(8,101)X(1,I),Y(1,I),THETA(1,I)                              SO
      DO 20 I=1,NOD(2)                                                  SO
 20   WRITE(8,101)X(2,I),Y(2,I),THETA(2,I)                              SO
 100  FORMAT(E14.8,5X,I4)                                               SO
 101  FORMAT(3(E12.5,5X))                                               SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE FPHASE(GF,XX,XXP,IP,JP,COND,T,RI,NP)                   SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      DIMENSION GF(1),XG(2),WG(2),HSB(4,4),H(4),                        SO
     1          B(2,4),XX(2,4),XXP(2,2),T(4),HS(2,4),EF(4)              SO
      DATA XG/-.5773502691896260D0,.5773502691896260D0/                 SO
      DATA WG/1.0000000000D0,1.0000000000D0/                            SO
C                                                                       SO
C                                                                       SO
      DO 10 I=1,4                                                       SO
      DO 10 J=1,4                                                       SO
 10   HSB(I,J)= 0.0D0                                                   SO
C                                                                       SO
C                                                                       SO
      DO 40 LI= 1,2                                                     SO
      SI= XG(LI)                                                        SO
      CALL STDM(RI,SI,H,B,DET,XX)                                       SO
      CALL STDB(SI,HS,DEL,XXP)                                          SO
      WT= WG(LI)*DEL                                                    SO
C                                                                       SO
```

```
C                                                                       SOL
      DO 30 I= 1,4                                                      SOL
      DO 30 J= 1,4                                                      SOL
      STIFF= 0.0D0                                                      SOL
      DO 20 M= 1,2                                                      SOL
 20   STIFF= STIFF + COND * HS(M,I) * B(M,J)                            SOL
 30   HSP(I,J)= HSP(I,J) + STIFF * WT                                   SOL
 40   CONTINUE                                                          SOL
      DO 50 I= 1,4                                                      SOL
      EF(I)= 0.0D0                                                      SOL
      DO 50 J= 1,4                                                      SOL
 50   EF(I)= EF(I) + HSP(I,J) * T(J)                                    SOL
C                                                                       SOL
C                                                                       SOL
      CALL ASEMPF(GF,EF,IP,JP)                                          SOL
      RETURN                                                            SOL
      END                                                               SOL
C                                                                       SOL
C***********************************************************************SOL
      SUBROUTINE HTCOND(GK,GF,X,Y,THETA,NEL1,NEL2,                      SOL
     1                  NOMAX2,NYMM1,NXMM1,NP,NTIME)                    SOL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SOL
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SOL
      COMMON/CODE2/NOBOUN                                               SOL
      DIMENSION GK(NOMAX2,1),GF(1),NEL1(NYMM1,NXMM1,1),                 SOL
     1          X(2,1),Y(2,1),THETA(2,1),XX(2,4),T(4),                  SOL
     2          NEL2(NYMM1,NXMM1,1)                                     SOL
C                                                                       SOL
      IB= NYMAX +2                                                      SOL
      IB2= IB *2                                                        SOL
      IF(NP .EQ.1)THEN                                                  SOL
      CALL CLEAR(GK,GF,NOMAX2,NOMAX2,IB)                                SOL
      ENDIF                                                             SOL
C                                                                       SOL
C     GENERATE ELEMENT MATRICES                                         SOL
      DO 10 I= 1, NYMAX -1                                              SOL
      DO 10 J= 1, NX(NP) -1                                             SOL
      DO 20 K= 1,4                                                      SOL
      IF(NP .EQ. 1)THEN                                                 SOL
          NOUM= NEL1(I,J,K)                                             SOL
      ELSE                                                              SOL
          NOUM= NEL2(I,J,K)                                             SOL
      ENDIF                                                             SOL
      XX(1,K)= X(NP,NOUM)                                               SOL
      XX(2,K)= Y(NP,NOUM)                                               SOL
 20   T(K)= THETA(NP,NOUM)                                              SOL
 10   CALL CONTEM(GK,GF,NEL1,NEL2,NOMAX2,NYMM1,NXMM1,I,J,XX,T,          SOL
     1            NP,NTIME)                                             SOL
C                                                                       SOL
      RETURN                                                            SOL
      END                                                               SOL
C                                                                       SOL
C***********************************************************************SOL
      SUBROUTINE INICON(X,Y,THETA)                                      SOL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SOL
```

```
      EXTERNAL TBOUND                                                   SO
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SO
      COMMON/CODE2/NOBOUN                                               SO
      COMMON/CODE3/XBOUND(2),YBOUND(3),XREFER                           SO
      COMMON/CODE6/HASPEC,THETAI,THETAC,TSAT,THETAH,THETSF              SO
      COMMON/CODE11/NTBC(2,3)                                           SO
      COMMON/RAD1/SINI,GRAD,TBVEL(3),INIFLG                             SO
      DIMENSION X(2,1),Y(2,1),THETA(2,1)                                SO
C                                                                       SO
C     CALCULATE INITIAL CONDITIONS                                      SO
C     THE PROFILE IS ASSUMED LINEAR IN THE 'X' DIRECTION                SO
C     AND UNIFORM IN THE 'Y' DIRECTION                                  SO
      IF(INIFLG .EQ. 0)THEN                                             SO
          DO 10 NP=1,2                                                  SO
          DO 20 IN=1,NX(NP)-1                                           SO
              I= 1 + (IN-1)*NYMAX                                       SO
              IF(X(NP,I) .GE. SINI)THEN                                 SO
                  TEMPI= THETAH                                         SO
              ELSE                                                      SO
                  PINTRP= (X(NP,I)-XBOUND(1))/(SINI-XBOUND(1))          SO
                  TEMPI= THETAI + (THETAH-THETAI)*PINTRP                SO
              ENDIF                                                     SO
          DO 30 J=1,NYMAX                                               SO
              L= J + I-1                                                SO
 30           THETA(NP,L)= TEMPI                                        SO
 20       CONTINUE                                                      SO
          DO 40 M= NOD(NP)+1,NOD(NP)+NYMAX                              SO
 40       THETA(NP,M)= TSAT                                             SO
 10       CONTINUE                                                      SO
      ELSEIF(INIFLG .EQ.1)THEN                                          SO
              DO 50 NP= 1,2                                             SO
              DO 60 IN= 1,NOD(NP)                                       SO
                 IF(NP .EQ. 1)THEN                                      SO
                     THETA(NP,IN)= THETAI                               SO
                 ELSE                                                   SO
                     THETA(NP,IN)= THETAH                               SO
                 ENDIF                                                  SO
 60           CONTINUE                                                 SO
              DO 70 JN= NOD(NP),NOD(NP)+NYMAX                           SO
 70           THETA(NP,JN)= TSAT                                       SO
 50           CONTINUE                                                 SO
      ENDIF                                                             SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE INIXYP(XP,YP)                                          SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SO
      COMMON/CODE2/NOBOUN                                               SO
      COMMON/CODE3/XBOUND(2),YBOUND(3),XREFER                           SO
      COMMON/POSI/XINI(3)                                               SO
      DIMENSION XP(1),YP(1)                                             SO
C                                                                       SO
C                                                                       SO
```

```
      YP(1)= YBOUND(1)                                                  SC
      YP(NOBOUN)= YBOUND(2)                                             SC
      Y1= YP(1)                                                         SC
      Y2= (YP(1) + YP(NOBOUN))/2.DO                                     SC
      Y3= YP(NOBOUN)                                                    SC
C                                                                       SC
      DETA= Y1**2.DO *(Y2-Y3) - Y1*(Y2**2.DO - Y3**2.DO)                SC
     1        + Y2**2.DO * Y3 - Y3**2.DO * Y2                           SC
      A= (XINI(1)*(Y2-Y3)- Y1*(XINI(2)-XINI(3))+ XINI(2)*Y3             SC
     1     - XINI(3)*Y2)/DETA                                           SC
      B= (Y1**2.DO *(XINI(2)-XINI(3))- XINI(1)*(Y2**2.DO-Y3**2.DO)      SC
     1     + Y2**2.DO * XINI(3) - Y3**2.DO * XINI(2))/DETA              SC
      C= (Y1**2.DO *(Y2*XINI(3)-Y3*XINI(2)) - Y1*(Y2**2.DO*XINI(3)      SC
     1    -Y3**2.DO*XINI(2))+ XINI(1)*(Y2**2.DO*Y3-Y3**2.DO*Y2))/DETA   SC
C                                                                       SC
      DYP= ( YBOUND(2)-YBOUND(1) )/( NOBOUN - 3)                        SC
      YP(2)= YP(1) + DYP/2.DO                                           SC
      YP(3)= YP(2) + DYP/2.DO                                           SC
      YP(NOBOUN -1)= YP(NOBOUN) - DYP/2.DO                              SC
      YP(NOBOUN -2)= YP(NOBOUN-1) - DYP/2.DO                            SC
C                                                                       SC
      DO 10 I=4, NOBOUN -3                                              SC
 10   YP(I)= YP(I-1) + DYP                                              SC
      DO 20 I= 1,NOBOUN                                                 SC
 20   XP(I)= A* YP(I) * YP(I) + B* YP(I) + C                            SC
      RETURN                                                            SC
      END                                                               SC
C                                                                       SC
C*********************************************************************** SC
      SUBROUTINE KPHASE(GK,NOMAX2,XXP,IP,JP)                            SC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SC
      COMMON/CODE7/DTIME                                                SC
      COMMON/COD12/CONDK(3),CAPA(3),RHO(3),HEATSL,COND3(2,2)            SC
      DIMENSION GK(NOMAX2,1),XG(2),WG(2),HSH(4,4),                      SC
     1          HS(2,4),XXP(2,2)                                        SC
      DATA XG/-.5773502691896260D0,.5773502691896260D0/                 SC
      DATA WG/1.0000000000D0,1.0000000000D0/                            SC
C                                                                       SC
      DO 10 I= 1,4                                                      SC
      DO 10 J= 1,4                                                      SC
 10   HSH(I,J)= 0.0D0                                                   SC
C                                                                       SC
      DO 40 LI= 1,2                                                     SC
      SI= XG(LI)                                                        SC
      CALL STDR(SI,HS,DEL,XXP)                                          SC
      WT= WG(LI)*DEL                                                    SC
C                                                                       SC
      DO 30 I= 1,4                                                      SC
      DO 30 J= 1,4                                                      SC
      STIFF= 0.0D0                                                      SC
      DO 20 M= 1,2                                                      SC
 20   STIFF= STIFF + HS(M,I)*HS(M,J)                                    SC
 30   HSH(I,J)= HSH(I,J) + STIFF * WT * RHO(1) * HEATSL /DTIME          SC
 40   CONTINUE                                                          SC
C                                                                       SC
```

```
      CALL ASEMPK(GK,NOMAX2,HSH,IP,JP)                                  SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE MERGE(Y,XP,YP)                                         SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE2/NOBOUN                                               SO
      COMMON/CODE16/RLENTH                                              SO
      DIMENSION XP(1), YP(1), Y(2,1),XTEM(200), YTEM(200)               SO
C                                                                       SO
      DO 110 N= 2, NOBOUN -1                                            SO
      DO 10 I= 1, NOBOUN -1                                             SO
      DEL= (Y(1,N) - YP(I))*(Y(1,N) - YP(I+1))                          SO
      IF(DEL.GT.0.0D0)GOTO 10                                           SO
      YTEM(N)= Y(1,N)                                                   SO
      XTEM(N) = ( XP(I+1) - XP(I) )*( YTEM(N) -YP(I))                   SO
     1          / ( YP(I+1) - YP(I) )+ XP(I)                            SO
      GO TO 110                                                         SO
 10   CONTINUE                                                          SO
 110  CONTINUE                                                          SO
C                                                                       SO
      DO 20 I=2, NOBOUN-1                                               SO
      XP(I)= XTEM(I)                                                    SO
 20   YP(I)= YTEM(I)                                                    SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE MESH(X,Y,XP,YP,NEL1,NEL2,NYMM1,NXMM1)                  SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SO
      COMMON/CODE2/NOBOUN                                               SO
      COMMON/CODE3/XBOUND(2), YBOUND(3), XXLEN                          SO
      COMMON/CODE4/NODO(2), NXO(2), NYO(2),NBOUND,NODMXO                SO
      DIMENSION X(2,1), Y(2,1), XP(1), YP(1),                           SO
     1          NEL1(NYMM1,NXMM1,1),XX(2,4),                            SO
     2          NEL2(NYMM1,NXMM1,1)                                     SO
C                                                                       SO
C     CALCULATE REFERENCE LENGTHS FOR GRID SPACING                      SO
      NXMAX= NXMM1 +1                                                   SO
      DO 110 NP= 1,2                                                    SO
      DELTA= XP(1) - XBOUND(NP)                                         SO
      XMAX= ABS(DELTA)                                                  SO
      DO 10 I= 2,NOBOUN                                                 SO
      DUM= XP(I) - XBOUND(NP)                                           SO
      DUM1= ABS(DUM)                                                    SO
      IF(DUM1.LT.XMAX) XMAX= DUM1                                       SO
 10   CONTINUE                                                          SO
C                                                                       SO
      IF(NP.EQ.1) THEN                                                  SO
          XLEN= XXLEN/5.D0                                              SO
      ELSE                                                              SO
          XLEN= XXLEN                                                   SO
      ENDIF                                                             SO
```

```
C                                                                       S
      I    MAX.LT.XLEN) THEN                                            S
           DUM= XMAX/(XLEN/2.D0)                                        S
           NX(NP)= INT(DUM) + 2                                         S
      EL.  IF(XMAX.LT.(2.D0*XLEN))THEN                                  S
           DUM= XMAX/(XLEN/2.D0)                                        S
           NX(NP)= INT(DUM) + 1                                         S
      ELSE                                                              S
           DUM= XMAX/XLEN                                               S
           NX(NP)= INT(DUM) +3                                          S
      ENDIF                                                             S
C                                                                       S
      IF(NX(NP).GT.NXMAX) NX(NP)= NXMAX                                 S
      IF(NX(NP).EQ.1) THEN                                              S
           DXU= XP(1)- XBOUND(NP)                                       S
           DXL= XP(NDBOUN) - XBOUND(NP)                                 S
           DXUD2= DXU                                                   S
           DXLD2= DXL                                                   S
       ELSEIF(NX(NP).LE.5) THEN                                         S
           DXU= (XP(1) - XBOUND(NP))/(NX(NP)-1)                         S
           DXL= (XP(NDBOUN)-XBOUND(NP))/(NX(NP)-1)                      S
           DXUD2= DXU                                                   S
           DXLD2= DXL                                                   S
       ELSE                                                             S
           DXU= (XP(1)-XBOUND(NP))/(NX(NP)-3)                           S
           DXL= (XP(NDBOUN)-XBOUND(NP))/(NX(NP)-3)                      S
           DXUD2= DXU/2.D0                                              S
           DXLD2= DXL/2.D0                                              S
       ENDIF                                                            S
C                                                                       S
      NY(NP)= NYD(NP)                                                   S
      NDD(NP)= NYMAX  * ( NX(NP)-1 )                                    S
      YLEN= (YBOUND(2)-YBOUND(1))/(NY(NP)-3)                            S
      X(NP,1)= XBOUND(NP)                                               S
      Y(NP,1)= YBOUND(1)                                                S
C                                                                       S
C     CALCULATE MESH POINTS ON THE 'X' BOUNDARIES OF THE CRYSTAL        S
      DO 20 I= 2, NY(NP)-1                                              S
      IF((I.LE.3).OR.(I.GE.(NY(NP)-1))) THEN                            S
           RLEN = YLEN/2.D0                                             S
      ELSE                                                              S
           RLEN= YLEN                                                   S
      ENDIF                                                             S
      X(NP,I)= XBOUND(NP)                                               S
 20   Y(NP,I)= Y(NP,I-1) + RLEN                                         S
      X(NP,NY(NP))= XBOUND(NP)                                          S
      Y(NP,NY(NP))= YBOUND(2)                                           S
C                                                                       S
C     CALCULATE MESH POINTS ON THE 'X' BOUNDARIES OF THE BLANKET        S
      RLEN3= (YBOUND(3)-YBOUND(2))/(NYMAX-NY(NP))                       S
      DO 11 I= NY(NP)+1,NYMAX                                           S
      X(NP,I)= XBOUND(NP)                                               S
 11   Y(NP,I)= Y(NP,I-1)+RLEN3                                          S
C                                                                       S
C     CALCULATE MESH POINTS ON THE INTERFACE                            S
```

```
      DO 30 I= 1,NOBOUN                                                 SOI
      M= NOD(NP) + I                                                    SOI
      X(NP,M)= XP(I)                                                    SOI
 30   Y(NP,M)= YP(I)                                                    SOI
C                                                                       SOI
C     CALCULATE INTEFACE MESH POINTS IN THE BLANKET                     SOI
      DO 22 I= NY(NP)+1,NYMAX                                           SOI
      M=NOD(NP)+I                                                       SOI
      X(NP,M)=XP(NOBOUN)                                                SOI
 22   Y(NP,M)= YP(NOBOUN) +(I-NY(NP))*RLEN3                             SOI
      DO 77 K=NOBOUN+1,NYMAX                                            SOI
 77   XP(K)= XP(NOBOUN)                                                 SOI
C                                                                       SOI
      IF(NX(NP).LE.2) GO TO 123                                         SOI
C                                                                       SOI
C     CALCULATE MESH POINTS ON THE 'Y' BOUNDARIES                       SO
      DO 40 M= 2, NX(NP)-1                                              SO
      IF(M.LE.3.OR.M.GE.(NX(NP)-1)) THEN                                SOI
          RLEN= DXUD2                                                   SO
      ELSE                                                              SO
          RLEN= DXU                                                     SO
      ENDIF                                                             SO
      I= NYMAX *(M-1) + 1                                               SO
      X(NP,I) = X(NP,I-NYMAX ) + RLEN                                   SO
 40   Y(NP,I) = YBOUND(I)                                               SO
C                                                                       SO
      DO 50 M= 2, NX(NP)-1                                              SO
      IF(M.LE.3.OR.M.GE.(NX(NP)-1)) THEN                                SO
          RLEN= DXLD2                                                   SO
      ELSE                                                              SO
          RLEN= DXL                                                     SO
      ENDIF                                                             SO
      I= NYMAX *M                                                       SO
      X(NP,I) = X(NP,I-NYMAX ) + RLEN                                   SO
 50   Y(NP,I) = YBOUND(3)                                               SOI
C                                                                       SOI
C     CALCULATE MESH POINTS IN THE BODY OF THE DOMAIN                   SOI
      DO 70 N= 2,NYMAX -1                                               SOI
      DO 70 M= 2,NX(NP)-1                                               SOI
      I= NYMAX *(M-1) + N                                               SO
      IF(NX(NP).LE.5) THEN                                              SO
          DUM = (XP(N)- XPOUND(NP))/(NX(NP)-1)                          SO
          DUM2= DUM                                                     SO
      ELSE                                                              SO
          DUM= (XP(N)-XBOUND(NP))/(NX(NP)-3)                            SO
          DUM2= DUM/2.D0                                                SO
      ENDIF                                                             SO
C                                                                       SO
      IF(M.LE.3.OR.M.GE.(NX(NP)-1)) THEN                                SO
          RLEN= DUM2                                                    SO
      ELSE                                                              SO
          RLEN= DUM                                                     SO
      ENDIF                                                             SO
      X(NP,I) = X(NP,I-NYMAX ) + RLEN                                   SO
 70   Y(NP,I) = Y(NP,N)                                                 SO
```

```
C                                                                       SC
C         CALCULATE ELEMENT NODE NUMBERS                                SC
          DO 80 N=1, NX(NP)-2                                           SC
          DO 80 M=1, NYMAX -1                                           SC
          IDUM= (N-1)*(NYMAX) +M                                        SC
          N1= IDUM                                                      SC
          N2= N1 + NYMAX                                                SC
          N3= N2 +1                                                     SC
          N4= N1 +1                                                     SC
          IF(NP.EQ.1) THEN                                              SC
               NEL1(M,N,1) = N1                                         SC
               NEL1(M,N,2) = N2                                         SC
               NEL1(M,N,3) = N3                                         SC
               NEL1(M,N,4) = N4                                         SC
          ELSE                                                          SC
               NEL2(M,N,1) = N2                                         SC
               NEL2(M,N,2) = N1                                         SC
               NEL2(M,N,3) = N4                                         SC
               NEL2(M,N,4) = N3                                         SC
          ENDIF                                                         SC
 80       CONTINUE                                                      SC
C                                                                       SC
 123      CONTINUE                                                      SC
C                                                                       SC
          N= NX(NP) -1                                                  SC
          DO 90 M=1, NYMAX -1                                           SC
          IDUM= (N-1)*(NYMAX -1) + M                                    SC
          N1= IDUM + N -1                                               SC
          N2= NOD(NP) +M                                                SC
          N3= N2 +1                                                     SC
          N4= N1 +1                                                     SC
          IF(NP.EQ.1) THEN                                              SC
               NEL1(M,N,1) = N1                                         SC
               NEL1(M,N,2) = N2                                         SC
               NEL1(M,N,3) = N3                                         SC
               NEL1(M,N,4) = N4                                         SC
          ELSE                                                          SC
               NEL2(M,N,1) = N2                                         SC
               NEL2(M,N,2) = N1                                         SC
               NEL2(M,N,3) = N4                                         SC
               NEL2(M,N,4) = N3                                         SC
          ENDIF                                                         SC
 90       CONTINUE                                                      SC
 110      CONTINUE                                                      SC
C                                                                       SC
          NODEMX= NOD(1) + NOD(2) + NYMAX                               SC
          RETURN                                                        SC
          END                                                           SC
C                                                                       SC
C*********************************************************************** SC
          SUBROUTINE MIXBC(GF,X,Y,NSYM,NP,NK)                           SC
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                           SC
          COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                  SC
          COMMON/COD12/C1(3),C2(3),C3(3),C4,COND3(2,2)                  SC
          DIMENSION GF(1), X(2,1), Y(2,1),                              SC
```

```
     1              XG(2), HP(2), EF(2)                                  SOL
      DATA XG/-.5773502691896260D0,.5773502691896260D0/                  SOL
C                                                                        SOL
      DO 110 II= 1, NX(NP) +NY(NP) -2                                    SOL
      IP=II                                                              SOL
      JP= II +1                                                          SOL
      DO 10 I=1,2                                                        SOL
 10   EF(I)= 0.0D0                                                       SOL
      SQL= ( X(NP,IP) - X(NP,JP) )                                       SOL
     1     * ( X(NP,IP) - X(NP,JP) )                                     SOL
     2     + ( Y(NP,IP) - Y(NP,JP) )                                     SOL
     3     * ( Y(NP,IP) - Y(NP,JP) )                                     SOL
      DEL= SQRT(SQL)/2.D0                                                SOL
      DO 30 LI= 1,2                                                      SOL
      RI= XG(LI)                                                         SOL
      HP(1)= .5D0*(1.0D0 - RI)                                           SOL
      HP(2)= .5D0*(1.0D0 + RI)                                           SOL
      DO 20 I= 1,2                                                       SOL
 20   EF(I)= EF(I) + HP(I)*DEL                                           SOL
 30   CONTINUE                                                           SOL
C                                                                        SOL
      DO 40 I=1,2                                                        SOL
      IF(I.EQ.1)THEN                                                     SOL
          K= IP                                                          SOL
        ELSE                                                             SOL
          K= JP                                                          SOL
      ENDIF                                                              SOL
 40   GF(K)= GF(K) + EF(I)                                               SOL
C                                                                        SOL
C                                                                        SOL
 110  CONTINUE                                                           SOL
      RETURN                                                             SOL
      END                                                                SOL
C                                                                        SOL
C*****************************************************************      SOL
      SUBROUTINE OLDNEW(X,Y,T,NEL1,NEL2,XO,YO,TO,                        SOL
     1                  XPO,YPO,NELO1,NELO2,NYMM1,NXMM1)                 SOL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                SOL
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                       SOL
      COMMON/CODE2/NOBOUN                                                SOL
      COMMON/CODE4/NODO(2),NXO(2),NYO(2),NBOUNO,NODMXO                   SOL
      DIMENSION X(2,1),Y(2,1),T(2,1),XO(2,1),YO(2,1),TO(2,1),            SOL
     1          NEL1(NYMM1,NXMM1,1),                                     SOL
     2          NEL2(NYMM1,NXMM1,1),XPO(1),YPO(1),                       SOL
     3          NELO1(NYMM1,NXMM1,1),NELO2(NYMM1,NXMM1,1)                SOL
C                                                                        SOL
C     SAVE NODAL COORDINATES AND TEMPERATURES FROM CURRENT TIME STEP     SOL
      DO 110 NPHASE= 1,2                                                 SO
      NODO(NPHASE)= NOD(NPHASE)                                          SO
      DO 10 M=1, NODO(NPHASE)                                            SO
      XO(NPHASE,M)= X(NPHASE,M)                                          SO
      YO(NPHASE,M)= Y(NPHASE,M)                                          SO
 10   TO(NPHASE,M)= T(NPHASE,M)                                          SO
C                                                                        SO
      DO 20 M=1,NYMAX                                                    SO
```

```
      I= NODO(NPHASE) + M                                               SOL
      XO(NPHASE,I)= XPO(M)                                              SOL
      YO(NPHASE,I)= YPO(M)                                              SOL
 20   TO(NPHASE,I)= T(NPHASE,I)                                         SOL
C                                                                       SOL
      DO 30 M= 1, NYMAX -1                                              SOL
      DO 30 N= 1, NX(NPHASE) -1                                         SOL
      DO 30 K=1,4                                                       SOL
      NELO1(M,N,K)= NEL1(M,N,K)                                         SOL
 30   NELO2(M,N,K)= NEL2(M,N,K)                                         SOL
C                                                                       SOL
      NXO(NPHASE)= NX(NPHASE)                                           SOL
      NYO(NPHASE)= NY(NPHASE)                                           SOL
 110  CONTINUE                                                          SOL
      NBOUNO= NOBOUN                                                    SOL
      NODMXO= NODEMX                                                    SOL
      RETURN                                                            SOL
      END                                                               SOL
C                                                                       SOL
C*********************************************************************SOL
      SUBROUTINE PLOTDT(XP,YP,NTIME)                                    SOI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SOI
      COMMON/CODE2/NOBOUN                                               SOI
      COMMON/CODE7/DTIME                                                SOI
      DIMENSION XP(1),YP(1)                                             SOI
C                                                                       SOI
C     OUTPUT INTERFACE LOCATIONS AND ELAPSED TIME TO A PLOTTING FILE    SOI
C     CHANGE DIMENSIONS FROM METERS TO CENTIMETERS AND                  SOI
C     TIME FROM SECONDS TO MINUTES                                      SOI
      TIME= NTIME*DTIME/60.D0                                           SOI
      WRITE(7,100)TIME                                                  SOI
      DO 10 I=1,NOBOUN                                                  SOI
      PXP= XP(I)*100.D0                                                 SOI
      PYP= YP(I)*100.D0                                                 SOI
 10   WRITE(7,101)PXP,PYP                                               SOI
 100  FORMAT(E14.9)                                                     SOI
 101  FORMAT(T10,2(E12.5,5X))                                           SOI
      RETURN                                                            SOI
      END                                                               SOI
C                                                                       SOI
C*********************************************************************SOI
      SUBROUTINE POST(X,Y,THETA,XP,YP,DSX,DSY,NTIME)                    SOI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SOI
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SOI
      COMMON/CODE2/NOBOUN                                               SOI
      COMMON/CODE6/HASPEC,THETAI,THETAC,TSAT,THETAH,THET5F              SOI
      COMMON/CODE7/DTIME                                                SOI
      COMMON/CODE11/NTMC(2,3)                                           SOI
      COMMON/CODE12/CONDK(3),CAPA(3),RHO(3),HEATSL,COND3(2,2)           SOI
      COMMON/CODE19/NTEND(3),NPRINT,NPLOT,NCNTUR                        SOI
      COMMON/CODE20/LXMAX,UNITS                                         SOI
      COMMON/CODE3/XBOUND(2),YBOUND(3),XREFER                           SOI
      COMMON/RAD1/START,GRAD,TRVEL(3),INIFLG                            SOI
      COMMON/BOUN/KBOUN                                                 SOI
      COMMON/POSI/XINI(3)                                               SOI
```

```
      CHARACTER *3 UNITS                                                SC
      DIMENSION X(2,1),Y(2,1),THETA(2,1),XP(1),YP(1),DSX(1),DSY(1)      SC
C                                                                       SC
      TIME= DTIME * NTIME                                               SC
      IF(NTIME .NE. 1)GO TO 10                                          SC
C                                                                       SC
C     PRINT INPUT PARAMETERS                                            SC
C     NOTE THAT 'GRAD' IS CALCULATED IN TROUND                          SC
      WRITE(6,901)UNITS                                                 SC
      WRITE(6,902)XBOUND(1),XBOUND(2),YBOUND(1),YBOUND(2),YBOUND(3)     SC
      WRITE(6,903)TSAT,THETAI,THETAC,THETAM,THETSF,TBVEL(1),            SC
     1            TBVEL(2),TBVEL(3),GRAD                                SC
      WRITE(6,904)CONDK(1),CONDK(2),COND3(1,1),COND3(1,2),              SC
     1                                COND3(2,1),COND3(2,2)             SC
      WRITE(6,905)CAPA(1),CAPA(2),CAPA(3)                               SC
      WRITE(6,906)RHO(1),RHO(2),RHO(3)                                  SC
      WRITE(6,907)HEATSL                                                SC
      WRITE(6,908)NTBC(1,1),NTBC(1,2),NTBC(1,3)                         SC
      WRITE(6,909)NTBC(2,1),NTBC(2,2),NTBC(2,3)                         SC
      WRITE(6,910)KBOUN,INIFLG,NTEND(1),NTEND(2),NTEND(3),DTIME,        SC
     1            NPRINT,NPLOT,NCNTUR,NYMAX,LXMAX,NOBOUN                SC
      WRITE(6,911)XINI(1),XINI(2),XINI(3)                               SC
      WRITE (6,912)START                                                SC
 901  FORMAT(T10,'ANALYSIS OF THE BRIDGEMAN'/                           SC
     1       T10,'CRYSTAL GROWTH PROCESS'/                              SC
     2       T10,'FROM PROGRAM SOLID5'///                               SC
     3       T10,'INPUT PARAMETERS IN ',A3,' UNITS'/                    SC
     4       T10,'UNLESS NOTED OTHERWISE'/)                             SC
 902  FORMAT(T10,'X BOUNDARIES :',2(F10.5)/                             SC
     1       T10,'Y BOUNDARIES :',3(F10.5)/)                            SC
 903  FORMAT(T10,'PHASE CHANGE TEMPERATURE :',F8.1/                     SOI
     1       T10,'INITIAL TEMPERATURE :',F8.1/                          SOI
     2       T10,'COLD WALL TEMPERATURE :',F8.1/                        SOI
     3       T10,'HOT WALL TEMPERATURE :',F8.1/                         SOI
     4       T10,'TCS TEMP AT END OF SEG3 :',F8.1/                      SOL
     5       T10,'SPEED OF THE TEMPERATURE BNDRY'/                      SOL
     6       T15,'PROCESS 1 : ',E11.5/                                  SOL
     7       T15,'PROCESS 2 : ',E11.5/                                  SOL
     8       T15,'PROCESS 3 : ',E11.5/                                  SOL
     9       T10,'GRADIENT OF THE TEMPERATURE BNDRY (DEGC/M):',F9.5/)SOL
 904  FORMAT(T10,'THERMAL CONDUCTIVITY'/                                SOL
     1  T15,'SOLID :',F6.1/T15,'LIQUID :',F6.1/                         SOL
     2  T15,'BLANKET  1,1 :',F7.2/                                      SOL
     3  T24,'1,2 :',F7.2/                                               SOL
     4  T24,'2,1 :',F7.2/                                               SOL
     5  T24,'2,2 :',F7.2)                                               SOL
 905  FORMAT(T10,'SPECIFIC HEAT'/                                       SOL
     1  T15,'SOLID :',F8.1/T15,'LIQUID :',F8.1/T15,'BLANKET :',F8.1/)   SOL
 906  FORMAT(T10,'DENSITY'/                                             SOL
     1  T15,'SOLID :',F8.1/T15,'LIQUID :',F9.1/T15,'BLANKET :',F8.1/)   SOL
 907  FORMAT(T10,'LATENT HEAT OF FUSION :',E11.4/)                      SOL
 908  FORMAT(T10,'BOUNDARY CONDITIONS: 1= INSULATED, 2= TEMPERATURE'/SOL
     1  T15,'NTBC 1,1 :',I2/T15,'NTBC 1,2 :',I2/T15,'NTBC 1,3 :',I2)    SOL
 909  FORMAT(T15,'NTBC 2,1 :',I2/T15,'NTBC 2,2 :',I2/                  SOL
     1       T15,'NTBC 2,3 :',I2/)                                      SOL
```

```
 910    FORMAT(T10,'TEMPERATURE BNDRY CONTROL, KBOUN :',I2/                S
     1          T10,'INITIAL CONDITION TYPE, INIFLG :',I2//                S
     2          T10,'NUMBER OF TIME STEPS'/                                S
     3          T15,'PROCESS 1: ',I6/                                      S
     4          T15,'PROCESS 2: ',I6/                                      S
     5          T15,'PROCESS 3: ',I6/                                      S
     6          T10,'TIME STEP SIZE :',F8.1/                               S
     7          T10,'TIME STEPS PER PRINT :',I4/                           S
     8          T10,'TIME STEPS PER PLOT DATA :',I4/                       S
     9          T10,'TIME STEPS PER CONTOUR DATA :',I4//                   S
     1          T10,'NODES IN THE Y-DIRECTION :',I4/                       S
     2          T10,'NODES IN THE X-DIRECTION :',I4/                       S
     3          T10,'NUMBER OF NODES ON THE INTERFACE :',I4/)              S
 911    FORMAT(T10,'INITIAL INTERFACE POSITION, XINI :',3(F7.4,2X))        S
 912    FORMAT(T10,'INITIAL TEMPERATURE BNDRY POSITION, SINI: ',F7.4//)S
C                                                                          S
 10     CONTINUE                                                           S
C                                                                          S
C       PRINT RESULTS AT SPECIFIED TIME STEPS                              S
        IF(KBOUN .EQ. 0)THEN                                               S
            IF(NTIME .LE. NTEND(1))THEN                                    S
                SDUMY= START + TRVEL(1) *NTIME *DTIME                      S
            ELSEIF(NTIME .LE. NTEND(1)+NTEND(2))THEN                       S
                SDUMY= START + TRVEL(1) *NTEND(1) *DTIME                   S
            ELSE                                                           S
                SDUMY= START + TRVEL(1)*NTEND(1)*DTIME                     S
     1                + TRVEL(3)*(NTIME-(NTEND(1)+NTEND(2)))*DTIME         S
            ENDIF                                                          S
        ELSEIF(KBOUN .EQ. 1)THEN                                           S
            SDUMY= XP(1) - DSX(1)                                          S
        ELSE                                                               S
            LDUM= NINT((NCPOUN +1)/2.00)                                   S
            SDUMY= XP(LDUM) -DSX(LDUM)                                     S
        ENDIF                                                              S
        WRITE(6,913) TIME,SDUMY                                            S
 913    FORMAT(///,5X,'ELAPSED TIME =',E14.8,                              S
     1          /,5X,'POSITION OF THETAH AT BOUNDARY =',E14.8)             S
C                                                                          S
        DO 110 NPHASE= 1,2                                                 S
           IF(NPHASE .EQ. 1)THEN                                           S
               WRITE(6,914)                                                S
           ELSE                                                            S
               WRITE(6,915)                                                S
           ENDIF                                                           S
           DO 120 I= 1,NOD(NPHASE)+NYMAX                                   S
              IF(NPHASE .EQ. 1)THEN                                        S
                  WRITE(6,916) I,X(1,I),Y(1,I),THETA(1,I)                  S
              ELSE                                                         S
                  WRITE(6,916) I,X(2,I),Y(2,I),THETA(2,I)                  S
              ENDIF                                                        S
 120       CONTINUE                                                        S
 110    CONTINUE                                                           S
 914    FORMAT(/,2X,'NODE',9X,'X',17X,'Y',14X,'T SOLID',/)                 S
 915    FORMAT(/,2X,'NODE',9X,'X',17X,'Y',14X,'T LIQUID'/)                 S
 916    FORMAT(2X,I4,3(2X,E16.8))                                          S
```

```
      WRITE(6,917)                                                      SO
      DO 210 I=1, NORDUN                                                SO
 210  WRITE(6,918)I,XP(I),YP(I),DSX(I),DSY(I)                           SO
 917  FORMAT(//T34,'NEW COORDINATES OF PHASE INTERFACE '//              SO
     1        6X,' NO ',10X,' XP ',16X,' YP ',16X,' DSX ',               SO
     2        14X,' DSY ')                                               SO
 918  FORMAT(/,5X,I3,4(4X,F16.8))                                       SO
C                                                                       SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************  SO
      SUBROUTINE PREP(X,Y,THETA,XP,YP,NEL1,NEL2,                        SO
     1                XO,YO,TO,XPO,YPO,NELO1,NELO2,                     SO
     2                NYMM1,NXMM1,NTIME)                                SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE4/NODO(2),NXO(2),NYO(2),NROUND,NODMXO                  SO
      DIMENSION X(2,1),Y(2,1),THETA(2,1),XP(1),YP(1),                   SO
     1          NEL1(NYMM1,NXMM1,1),NEL2(NYMM1,NXMM1,1),                SO
     2          XO(2,1),YO(2,1),TO(2,1),XPO(1),YPO(1),                  SO
     3          NELO1(NYMM1,NXMM1,1),NELO2(NYMM1,NXMM1,1)               SO
C                                                                       SO
      IF(NTIME.EQ.1) GO TO 100                                          SO
      CALL OLDNEW(X,Y,THETA,NEL1,NEL2,XO,YO,TO,                         SO
     1            XPO,YPO,NELO1,NELO2,NYMM1,NXMM1)                      SO
 100  CONTINUE                                                          SO
      CALL MESH(X,Y,XP,YP,NEL1,NEL2,NYMM1,NXMM1)                        SO
      IF(NTIME.EQ.1)GO TO 200                                           SO
      CALL WHERE(X,Y,THETA,XO,YO,TO,NELO1,NELO2,NYMM1,NXMM1)            SO
      RETURN                                                            SO
 200  CALL INICON(X,Y,THETA)                                            SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************  SO
      SUBROUTINE RHSB(AA,X,B,L1,N,IB)                                   SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      DIMENSION AA(L1,1),X(1),B(1)                                      SO
      NPI= N +1                                                         SO
      DO 110 I= 2, N                                                    SO
      SUM = 0.0D0                                                       SO
      K1= MINO(IB-1,I-1)                                                SO
      DO 100 K= 1,K1                                                    SO
 100  SUM= SUM + AA(I-K,K+1) / AA(I-K,1) * B(I-K)                       SO
 110  B(I)= B(I) - SUM                                                  SO
      X(N)= B(N)/AA(N,1)                                                SO
      DO 130 K= 2,N                                                     SO
      I= NPI - K                                                        SO
      J1= I +1                                                          S
      J2= MINO(N,I+IB-1)                                                S
      SUM= 0.0D0                                                        S
      DO 120 J= J1,J2                                                   S
      MM = J - J1 +2                                                    S
 120  SUM= SUM + X(J) * AA(I,MM)                                        S
 130  X(I)= ( B(I) - SUM) / AA(I,1)                                     S
```

```
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE SOLVER(GK,GF,X,Y,THETA,XP,YP,DSX,DSY,NEL1,NEL2,        SO
     1                  XPO,YPO,TX,PX,NOMAX2,NYMM1,NXMM1,NTIME)         SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SO
      DIMENSION GK(NOMAX2,1),GF(1),X(2,1),Y(2,1),THETA(2,1),            SO
     1          XP(1),YP(1),DSX(1),DSY(1),XPO(1),YPO(1),TX(1),PX(1),    SO
     2          NEL1(NYMM1,NXMM1,1),NEL2(NYMM1,NXMM1,1)                 SO
C                                                                       SO
      IB= NYMAX +2                                                      SO
      IB2= IB*2                                                         SO
      CALL HTCOND(GK,GF,X,Y,THETA,NEL1,NEL2,                            SO
     1            NOMAX2,NYMM1,NXMM1,1,NTIME)                           SO
      CALL HTCOND(GK,GF,X,Y,THETA,NEL1,NEL2,                            SO
     1            NOMAX2,NYMM1,NXMM1,2,NTIME)                           SO
      CALL TEMBC(GK,GF,X,NOMAX2,0,1,IB,NTIME)                           SO
      CALL TRIB(GK,NOMAX2,NODEMX,IB)                                    SO
      CALL RHSB(GK,TX,GF,NOMAX2,NODEMX,IB)                              SO
C                                                                       SO
C     TRANSFER TEMPERATURES TO LOCAL NODE NUMBERS                       SO
      DO 10 I=1,NOD(1)+NYMAX                                            SO
 10   THETA(1,I)= TX(I)                                                 SO
      DO 20 J=1,NX(2)                                                   SO
         N= NODEMX - J*NYMAX                                            SO
         M= (J-1) *NYMAX                                                SO
         DO 30 K=1,NYMAX                                                SO
 30      THETA(2,M+K)= TX(N+K)                                          SO
 20   CONTINUE                                                          SO
C                                                                       SO
      CALL VPHASE(GK,GF,X,Y,THETA,XP,YP,DSX,DSY,NEL1,NEL2,XPO,YPO,      SO
     1            PX,NOMAX2,NYMM1,NXMM1,NTIME)                          SO
      RETURN                                                            SO
      END                                                               SO
C                                                                       SO
C*********************************************************************** SO
      SUBROUTINE STDR(R,HS,DEL,XX)                                      SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      DIMENSION HS(2,4),XX(2,4)                                         SO
C                                                                       SO
      SQL= ( XX(1,1) - XX(1,2) )*( XX(1,1) - XX(1,2) )                  SO
     1   + ( XX(2,1) - XX(2,2) )*( XX(2,1) - XX(2,2) )                  SO
      DEL = SQRT(SQL)/2.D0                                              SO
C                                                                       SO
      DO 10 I= 1,2                                                      SO
      DO 10 J= 1,4                                                      SO
 10   HS(I,J)= 0.0D0                                                    SO
      HS(1,1)= .5D0*(1.D0-R)                                            SO
      HS(1,3)= .5D0*(1.D0+R)                                            SO
      HS(2,2)= HS(1,1)                                                  SO
      HS(2,4)= HS(1,3)                                                  SO
      RETURN                                                            SO
      END                                                               SO
```

```
C                                                                       SO
C***********************************************************************SO
      SUBROUTINE STDM(R,S,H,B,DET,XX)                                   SO
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SO
      DIMENSION H(4),P(2,4),P(2,4),XX(2,4),                             SO
     1          XJ(2,2),XJI(2,2)                                        SO
C                                                                       SO
      RP = 1.000+R                                                      SO
      RM= 1.000-R                                                       SO
      SP= 1.000+S                                                       SO
      SM= 1.000-S                                                       SO
C                                                                       SO
      H(1) = .2500*RM * SM                                              SO
      H(2) = .2500*RP * SM                                              SO
      H(3) = .2500*RP * SP                                              SO
      H(4) = .2500*RM * SP                                              SO
C                                                                       SO
      P(1,1) =-.2500*SM                                                 SO
      P(1,2) = .2500*SM                                                 SO
      P(1,3) = .2500*SP                                                 SO
      P(1,4) =-.2500*SP                                                 SO
      P(2,1) =-.2500*RM                                                 SO
      P(2,2) =-.2500*RP                                                 SO
      P(2,3) = .2500*RP                                                 SO
      P(2,4) = .2500*RM                                                 SO
C                                                                       SO
C     CALCULATE THE TRANSFORMATION (J) MATRIX                           SO
      DO 30 I= 1,2                                                      SO
      DO 30 J= 1,2                                                      SO
      DUM= 0.000                                                        SO
      DO 20 K= 1,4                                                      SO
 20   DUM= DUM + P(I,K) * XX(J,K)                                       SO
 30   XJ(I,J)= DUM                                                      SO
C                                                                       SO
C     CHECK FOR ZERO OR NEGATIVE DETERMINANT                            SO
      DET= XJ(1,1) * XJ(2,2) - XJ(1,2) * XJ(2,1)                        SO
      IF(DET.GT.0.0D0)THEN                                              SO
          GO TO 40                                                      SO
        ELSE                                                            SO
          WRITE(6,*) (XX(1,I),I=1,4)                                    SO
          WRITE(6,*) (XX(2,I),I=1,4)                                    SO
          WRITE(6,999)                                                  SO
 999      FORMAT('ZERO OR NEGATIVE DETERMINENT')                        SO
          STOP                                                          S
      ENDIF                                                             S
C                                                                       S
 40   DUM= 1.000/DET                                                    S
      XJI(1,1)= XJ(2,2) * DUM                                           S
      XJI(1,2)= - XJ(1,2) * DUM                                         S
      XJI(2,1)= -XJ(2,1) * DUM                                          S
      XJI(2,2)= XJ(1,1) * DUM                                           S
C                                                                       S
      DO 60 I= 1,2                                                      S
      DO 60 J= 1,4                                                      S
      DUM= 0.000                                                        S
```

```
      DO 50 M= 1,2                                                      SC
 50   DUM= DUM + XJI(I,M) * P(M,J)                                      SC
 60   B(I,J)= DUM                                                       SC
      RETURN                                                            SC
      END                                                               SC
C                                                                       SC
C*********************************************************************** SC
      FUNCTION TBOUND(NP,X,ILOCAL,NTIME)                                SC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SC
      COMMON/CODE6/HASPEC,THETAI,THETAC,TSAT,THETAH,THET5F              SC
      COMMON/CODE7/DTIME                                                SC
      COMMON/RAD1/SINI,GRAD,VEL(3),INIFLG                               SC
      COMMON/BOUN/KBOUN                                                 SC
      COMMON/CODE19/NTEND(3),NPRINT,NPLOT,NCNTUR                        SC
      DIMENSION X(2,1)                                                  SC
C                                                                       SC
      GRAD= (THETAH - THET5F)/SINI                                      SC
      IF(KBOUN.EQ.0) THEN                                               SC
          IF(NTIME .LE. NTEND(1))THEN                                   SC
              THETA5= (THET5F-THETAI) * NTIME/NTEND(1) + THETAI         SC
              IF(X(NP,ILOCAL) .GE. SINI)THEN                            SC
                  TBOUND= THETAH                                        SC
              ELSE                                                      SC
                  TBOUND= THETA5 + (THETAH-THETA5)*X(NP,ILOCAL)/SINI    SC
              ENDIF                                                     SC
          ELSEIF(NTIME .LE. NTEND(1)+NTEND(2))THEN                      SC
              IF(X(NP,ILOCAL) .GE. SINI)THEN                            SC
                  TBOUND= THETAH                                        SC
              ELSE                                                      SC
                  TBOUND= THET5F + (THETAH-THET5F)*X(NP,ILOCAL)/SINI    SC
              ENDIF                                                     SC
          ELSE                                                          SC
              RUN=  VEL(3) * (NTIME-(NTEND(1)+NTEND(2))) * DTIME        SC
              THETA5= THET5F - RUN*GRAD                                 SC
              IF(X(NP,ILOCAL) .GE. RUN+SINI)THEN                        SC
                  TBOUND= THETAH                                        SC
              ELSE                                                      SC
                  TBOUND= THETA5 + X(NP,ILOCAL)*GRAD                    SC
              ENDIF                                                     SC
          ENDIF                                                         SC
      ELSE                                                              SC
          START= SINI - GRAD                                            SC
      ENDIF                                                             SC
C                                                                       SC
      RETURN                                                            SC
      END                                                               SC
C                                                                       SC
C*********************************************************************** SC
      SUBROUTINE TEMBC(GK,GF,X,NOMAX2,NSTM,NP,IB,NTIME)                 SC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                               SC
      EXTERNAL TBOUND                                                   SC
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                      SC
      COMMON/CODE2/NCBOUN                                               SC
      COMMON/CODE6/H1,H2,H3,TSAT,THETAH,THET5F                          SC
      COMMON/CODE11/NTBC(2,3)                                           SC
```

```
      DIMENSION GK(NDMAX2,1),CF(1),X(2,1)                               SC
      BIG= 1.00                                                         SC
C                                                                       SC
C     ADJUST STIFFNESS MATRIX FOR IMPOSED INTERFACE TEMPERATURES        SC
      IF(NSYM .EQ. 0)THEN                                               SC
C     FOR A SYMMETRIC STIFFNESS MATRIX                                  SC
          DO 401 J=1, NODEMX                                            SC
          DO 402 K=1, NCPOUN                                            SC
          I= NOD(NP) + K                                                SC
          JG= J -1 +1                                                   SC
          IF((JG.LE. 0).OR.(JG .GT. IB))GO TO 403                       SC
          CF(J)= CF(J) - GK(I,JG)*TSAT                                  SC
          GK(I,JG)= 0.000                                               SC
          GO TO 402                                                     SC
 403      KG= 1 - J + 1                                                 SC
          IF(KG .LE. 0 .OR. KG .GT. IB)GO TO 402                        SC
          CF(J)= CF(J) - GK(J,KG)*TSAT                                  SC
          GK(J,KG)= 0.000                                               SC
 402      CONTINUE                                                      SC
 401      CONTINUE                                                      SC
          DO 501 K=1,NCPOUN                                             SC
          I= NOD(NP) + K                                                SC
          GK(I,1)= BIG                                                  SC
 501      CF(I)= BIG*TSAT                                               SC
      ELSE                                                              SC
C     FOR A NON-SYMMETRIC STIFFNESS MATRIX                              SC
          DO 411 K=1,NCPOUN                                             SC
          I= NOD(NP) + K                                                SC
          DO 412 J=1, NOD(NP)+NCBOUN                                    SC
          CF(J)= CF(J) - GK(J,I)*TSAT                                   SC
          GK(I,J)= 0.000                                                SC
 412      GK(J,I)=0.000                                                 SC
 411      CONTINUE                                                      SC
          DO 511 K=1,NCBOUN                                             SC
          I= NOD(NP) + K                                                SC
          GK(I,I)= BIG                                                  SC
 511      CF(I)= BIG*TSAT                                               SC
      ENDIF                                                             SC
C                                                                       SC
      DO 2000 NB=1,2                                                    SC
C                                                                       SC
      DO 701 NK=1,3                                                     SC
      IF(NTBC(NB,NK) .EQ.3)THEN                                         SC
      CALL MIXBC(CF,X,Y,NSYM,NB,NK)                                     SC
      ENDIF                                                             SC
 701  CONTINUE                                                          SC
C                                                                       SC
      DO 1001 NK=1,3                                                    SC
      IF(NTBC(NB,NK) .EQ.1)GO TO 1001                                   SC
      IF(NTBC(NB,NK) .EQ.3)GO TO 1001                                   SC
      IF(NSYM .EQ. 0)THEN                                               SC
C     FOR A SYMMETRIC STIFFNESS MATRIX                                  SC
C                                                                       SC
          IF(NK .EQ. 1)THEN                                             SC
C         ADJUST STIFFNESS MATRIX FOR TEMPERATURES IMPOSED              SC
```

```
C               ON THE (NB,1) BOUNDARIES                                SC
                     DO 200 J=1, NODEMX                                 SC
                     DO 201 K=1,NYMAX                                   SC
                     I= K + (NP-1)*(NODEMX - NYMAX)                     SC
                     JG= J -I +1                                        SC
                     IF((JG.LE.0).OR.(JG.GT.IB)) GO TO 291              SC
                     GF(J)= GF(J) - GK(I,JG)*TBOUND(NB,X,1,NTIME)       SC
                     GK(I,JG)= 0.000                                    SC
                     GO TO 201                                          SC
 291                 KG= I -J +1                                        SC
                     IF((KG.LE.0).OR.(KG.GT.IB))GO TO 201               SC
                     GF(J)= GF(J) - GK(J,KG)*TBOUND(NB,X,1,NTIME)       SC
                     GK(J,KG)= 0.000                                    SC
 201                 CONTINUE                                           SC
 200                 CONTINUE                                           SC
C                                                                       SC
               ELSEIF(NK.EQ.2)THEN                                      SC
C              ADJUST STIFFNESS MATRIX FOR TEMPERATURES IMPOSED         SC
C              ON THE (NB,2) BOUNDARIES                                 SC
                     DO 210 J=1, NODEMX                                 SC
                     DO 211 K=1, NX(NB)                                 SC
                          IF(NB.EQ.1)THEN                               SC
                               I= K*NYMAX                               SC
                               ILOCAL= I                                SC
                          ELSE                                          SC
                               I= NODEMX - (K-1)*NYMAX                  SC
                               ILOCAL= K*NYMAX                          SC
                          ENDIF                                         SC
                     JG= J-I+1                                          SC
                     IF((JG.LE.0).OR.(JG.GT.IB))GO TO 212               SC
                     GF(J)= GF(J) - GK(I,JG)*TBOUND(NB,X,ILOCAL,NTIME)  SC
                     GK(I,JG)= 0.000                                    SC
                     GO TO 211                                          SC
 212                 KG= I -J +1                                        SC
                     IF((KG.LE.0).OR.(KG.GT.IB))GO TO 211               SC
                     GF(J)= GF(J) - GK(J,KG)*TBOUND(NB,X,ILOCAL,NTIME)  SC
                     GK(J,KG)= 0.000                                    SC
 211                 CONTINUE                                           SC
 210                 CONTINUE                                           SC
               ELSE                                                     SC
C              ADJUST STIFFNESS MATRIX FOR TEMPERATURES IMPOSED         SC
C              ON THE (NB,3) BOUNDARIES                                 SC
                     DO 220 J=1, NODEMX                                 SC
                     DO 221 K=1, NX(NB)                                 SC
                          IF(NB.EQ.1)THEN                               SC
                               I= 1 + (K-1)*NYMAX                       SC
                               ILOCAL= I                                SC
                          ELSE                                          SC
                               I= 1 + NODEMX - K*NYMAX                  SC
                               ILOCAL= 1 + (K-1)*NYMAX                  SC
                          ENDIF                                         SC
                     JG= J -I +1                                        SC
                     IF((JG.LE.0).OR.(JG.GT.IB))GO TO 222               SC
                     GF(J)= GF(J) - GK(I,JG)*TBOUND(NB,X,ILOCAL,NTIME)  SC
                     GK(I,JG)= 0.000                                    SC
```

```
                GO TO 221                                          SC
 222            KG= I - J +1                                       SC
                IF((KG.LE.0).OR.(KG.GT.IB))GO TO 221               SC
                GF(J)= GF(J) - GK(J,KG)*TBOUND(NB,X,ILOCAL,NTIME)  SC
                GK(J,KG)= 0.0D0                                    SC
 221            CONTINUE                                           SC
 220            CONTINUE                                           SC
 909        ENDIF                                                  SC
        ELSE                                                       SC
C       FOR A NON-SYMMETRIC STIFFNESS MATRIX                       SC
C       MODIFICATIONS ARE NOT COMPLETE                             SC
           IF(NK.EQ.1)THEN                                         SC
              DO 300 J=1, NOD(NB)+NOBOUN                           SC
              DO 301 K=1, NY(NB)                                   SC
              I=K                                                  SC
              GF(J)= GF(J) - GK(J,I) * TBOUND(NB,X,I,NTIME)        SC
              GK(I,J)= 0.0D0                                       SC
 301          GK(J,I)= 0.0D0                                       SC
 300          CONTINUE                                             SC
           ELSEIF(NK.EQ.2)THEN                                     SC
              DO 310 J= 1, NOD(NB)+NOBOUN                          SC
              DO 311 K= 1, NX(NB) -1                               SC
              I= 1 + NY(NB)*(K-1)                                  SC
              GF(J)= GF(J) - GK(J,I) * TBOUND(NB,X,I,NTIME)        SC
              GK(I,J)= 0.0D0                                       SC
 311          GK(J,I)= 0.0D0                                       SC
 310          CONTINUE                                             SC
           ELSE                                                    SC
              DO 320 J= 1, NOD(NB)+NOBOUN                          SC
              DO 321 K= 1, NX(NB) -1                               SC
              I= NY(NB)*K                                          SC
              GF(J)= GF(J) - GK(J,I) * TBOUND(NB,X,I,NTIME)        SC
              GK(I,J)= 0.0D0                                       SC
 321          GK(J,I)= 0.0D0                                       SC
 320          CONTINUE                                             SC
           ENDIF                                                   SC
        ENDIF                                                      SC
 1001   CONTINUE                                                   SC
 2000   CONTINUE                                                   SC
C                                                                  SC
        DO 3000 NB=1,2                                             SC
C                                                                  SC
C       IMPOSE TEMPERATURES                                        SC
        IF(NTBC(NB,1).EQ.2)THEN                                    SC
C       ON THE (NB,1) BOUNDARY                                     SC
              DO 203 K=1, NYMAX                                    S
              I= K + (NB-1)*(NODEMX - NYMAX)                       S
              DO 202 J=1,IB                                        S
 202          GK(I,J)= 0.0D0                                       S
 203          CONTINUE                                             S
              DO 20 K=1, NYMAX                                     S
              I= K + (NB-1)*(NODEMX - NYMAX)                       S
            IF(NSYM.EQ.0) THEN                                     S
              J=1                                                  S
            ELSE                                                   S
```

```
                J=I                                                     SC
              ENDIF                                                     SC
                GK(I,J)= 1.0D0                                          SC
 20             GF(I)= TBOUND(NB,X,1,NTIME)                             SC
        ENDIF                                                           SC
C                                                                       SC
        IF(NTBC(NB,2).EQ.2)THEN                                         SC
C       ON THE (NB,2) BOUNDARY                                          SC
                DO 213 K=2, NX(NB)                                      SC
                     IF(NB.EQ.1)THEN                                    SC
                          I= K*NYMAX                                    SC
                     ELSE                                               SC
                          I= NODEMX - (K-1)*NYMAX                       SC
                     ENDIF                                              SC
                DO 242 J=1,IB                                           SC
 242            GK(I,J)= 0.0D0                                          SC
 213            CONTINUE                                                SC
                DO 21 K=2, NX(NB)                                       SC
                     IF(NB.EQ.1)THEN                                    SC
                          I= K*NYMAX                                    SC
                          ILOCAL= I                                     SC
                     ELSE                                               SC
                          I= NODEMX - (K-1)*NYMAX                       SC
                          ILOCAL= K*NYMAX                               SC
                     ENDIF                                              SC
              IF(NSYM.EQ.0)THEN                                         SC
                J= 1                                                    SC
              ELSE                                                      SC
                J= 1                                                    SC
              ENDIF                                                     SC
                GK(I,J)= 1.0D0                                          SC
 21             GF(I)= TBOUND(NB,X,ILOCAL,NTIME)                        SC
        ENDIF                                                           SC
C                                                                       SC
        IF(NTBC(NB,3).EQ.2)THEN                                         SC
C       ON THE (NB,3) BOUNDARY                                          SC
                DO 223 K=2, NX(NB) -1                                   SC
                        IF(NB.EQ.1)THEN                                 SC
                             I= 1 + (K-1)*NYMAX                         SC
                        ELSE                                            SC
                             I= 1 + NODEMX - K*NYMAX                    SC
                        ENDIF                                           SC
                DO 252 J=1,IB                                           SC
 252            GK(I,J)= 0.0D0                                          SC
 223            CONTINUE                                                SC
                DO 22 K=2, NX(NB) -1                                    SC
                        IF(NB.EQ.1)THEN                                 SC
                             I= 1 + (K-1)*NYMAX                         SC
                             ILOCAL= I                                  SC
                        ELSE                                            SC
                             I= 1 + NODEMX - K*NYMAX                    SC
                             ILOCAL= 1 + (K-1)*NYMAX                    SC
                        ENDIF                                           SC
              IF(NSYM.EQ.0) THEN                                        SC
                J= 1                                                    SC
```

```
                ELSE                                                    SO
                  J= I                                                  SO
                ENDIF                                                   SO
                  GK(I,J)= 1.000                                        SO
 22               GF(I)= TRCUND(NR,X,ILOCAL,NTIME)                      SO
          ENDIF                                                         SO
 3000     CONTINUE                                                      SO
          RETURN                                                        SO
          END                                                           SO
C                                                                       SO
C*********************************************************************** SO
          SUBROUTINE TRIB(B,L1,N,IB)                                    SO
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                           SO
          DIMENSION B(L1,1)                                             SO
          DO 120 I= 2,N                                                 SO
          M1 = MINO(IB-1,N-I+1)                                         SO
          DO 120 J= 1,M1                                                SO
          SUM = 0.0D0                                                   SO
          K1= MINO(I-1,IB-J)                                            SO
          DO 100 K= 1,K1                                                SO
 100      SUM= SUM + B(I-K,K+1) * B(I-K,J+K) / B(I-K,1)                 SO
 120      B(I,J)= B(I,J) - SUM                                          SO
          RETURN                                                        SO
          END                                                           SO
C                                                                       SO
C*********************************************************************** SO
          SUBROUTINE VPHASE(GK,GF,X,Y,THETA,XP,YP,                      SO
     1                      DSX,DSY,NEL1,NEL2,XPO,YPO,PX,NOMAX2,        SO
     2                      NYMM1,NXMM1,NTIME)                          SO
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)                           SO
          COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                  SO
          COMMON/CODE2/NOBOUN                                           SO
          COMMON/CODE6/H1,H2,H3,TSAT,THETAH,THETSF                      SO
          COMMON/CODE7/DTIME                                            SO
          COMMON/CODE12/CONDK(3),CAPA(3),RHO(3),HEATSL,COND3(2,2)       SO
          COMMON/RAD1/START,GRAD,VEL(3),INIFLG                          SO
          COMMON/BOUN/KBOUN                                             SO
          COMMON/DIREC/RINCPE                                           SO
          DIMENSION GK(NOMAX2,1),GF(1),X(2,1),Y(2,1),THETA(2,1),        SO
     1              PX(1),XP(1),YP(1),NEL1(NYMM1,NXMM1,1),              SO
     2              DSX(1),DSY(1),XPO(1),YPO(1),                        SO
     3              XX(2,4),T(4),XXP(2,2),NEL2(NYMM1,NXMM1,1)           SO
C                                                                       SO
          NBOUN2=NOBOUN *2                                              SO
          CALL CLEAR(GK,GF,NOMAX2,NBOUN2,4)                             SO
          DO 110 NP= 1,2                                                SO
          DO 10 M= 1,NOBOUN-1                                           SO
          DO 20 K=1,4                                                   SO
          IF(NP.EQ.1)THEN                                               SO
              NOUM= NEL1(M,NX(1)-1,K)                                   SO
          ELSE                                                          SO
              NOUM= NEL2(M,NX(2)-1,K)                                   SO
          ENDIF                                                         SO
          XX(1,K)= X(NP,NOUM)                                           SO
          XX(2,K)= Y(NP,NOUM)                                           SO
```

```
 20       T(K)= THETA(NP,NOUM)                                          SO
C                                                                       SO
          IF(NP.EQ.1) THEN                                              SO
                 COND= CONDK(1)                                         SO
                 RI= 1.0000000                                          SO
            ELSE                                                        SO
                 COND= -CONDK(2)                                        SO
                 RI= -1.00000000                                        SO
          ENDIF                                                         SO
          IP= M                                                         SO
          JP= M +1                                                      SO
          XXP(1,1)= XP(IP)                                              SO
          XXP(1,2)= XP(JP)                                              SO
          XXP(2,1)= YP(IP)                                              SO
          XXP(2,2)= YP(JP)                                              SO
          CALL FPHASE(GF,XX,XXP,IP,JP,COND,T,RI,NP)                     SO
 10       CONTINUE                                                      SO
 110      CONTINUE                                                      SO
C                                                                       SO
          DO 30 M=1, NOBOUN-1                                           SO
          IP= M                                                         SO
          JP= M+1                                                       SO
          XXP(1,1)= XP(IP)                                              SO
          XXP(1,2)= XP(JP)                                              SO
          XXP(2,1)= YP(IP)                                              SO
          XXP(2,2)= YP(JP)                                              SO
 30       CALL KPHASE(GK,NOMAX2,XXP,IP,JP)                              SO
          CALL APLBC(GK,GF,NOMAX2)                                      SO
          CALL TRIB(GK,NOMAX2,NBOUN2,4)                                 SO
          CALL RHSB(GK,PX,GF,NOMAX2,NBOUN2,4)                           SO
C                                                                       SO
          DO 40 I= 1,NYMAX                                              SO
          XPO(I)= XP(I)                                                 SO
 40       YPO(I)= YP(I)                                                 SO
C                                                                       SO
          DO 50 I= 1,NOBOUN                                             SO
          IDUM= I*2 -1                                                  SO
          JDUM= I*2                                                     SO
          DSX(I)= PX(IDUM)                                              SO
          DSY(I)= PX(JDUM)                                              SO
          XP(I)= XPO(I) + DSX(I)                                        SO
 50       YP(I)= YPO(I) + DSY(I)                                        SO
          RINCRE= DSX(NOBOUN)                                           SO
C                                                                       SO
 70       CALL MERGE(Y,XP,YP)                                           SO
          IF(KBOUN.EQ.1) START= XP(1)                                   SO
          IF(KBOUN.EQ.2)THEN                                            SO
              LDUM= NINT( (NOBOUN+1)/2.00)                              SO
              START= XP(LDUM)                                           SO
          ENDIF                                                         SO
          RETURN                                                        SO
          END                                                           SO
C                                                                       SO
C*********************************************************************** SO
          SUBROUTINE WHERE(X,Y,THETA,XO,YO,TO,NELO1,NELO2,NYMM1,NXMM1)  SO
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                SOI
      COMMON/CODE1/NOD(2),NX(2),NY(2),NODEMX,NYMAX                       SOI
      COMMON/CODE2/NOBOUN                                                SOI
      COMMON/CODE4/NODO(2),NXO(2),NYO(2),NBOUNO,NODMXO                   SOI
      COMMON/CODE6/HH1,HH2,HH3,TSAT,THETAH,THETSF                        SOI
      COMMON/DIREC/PINCRE                                                SOI
      DIMENSION X(2,1),Y(2,1),THETA(2,1),XO(2,1),YO(2,1),TO(2,1),        SOI
     1          NELO1(NYMM1,NXMM1,1),XK(4),YK(4),                        SOI
     2          NELO2(NYMM1,NXMM1,1)                                     SOI
C                                                                        SOI
C     CALCULATE NODAL TEMPERATURES AT NEW MESH POINTS                    SOI
      DO 1 NP= 1,2                                                       SOI
      DO 2 I= 1,NYMAX                                                    SOI
 2    THETA(NP,I)= TO(NP,I)                                              SOI
      DO 3 I= 1,NOBOUN                                                   SO
      J= NOD(NP) +I                                                      SO
 3    THETA(NP,J)= TSAT                                                  SO
C                                                                        SOI
      DO 10 JN= 1,NYMAX                                                  SO
      DO 20 IN= 2,NX(NP)-1                                               SOI
      L= NYMAX *(IN-1) +JN                                               SOI
      DO 30 K= 1,NXO(NP)-1                                               SOI
      KM= NYMAX*(K-1) +JN                                                SO
      KP= KM + NYMAX                                                     SO
      DUM= ( XO(NP,KP) - X(NP,L) )*( XO(NP,KM) - X(NP,L) )               SO
      IF(DUM.GT.0.0D0)GO TO 30                                           SO
      THETA(NP,L)= ( TO(NP,KP)- TO(NP,KM) )*( X(NP,L)- XO(NP,KM) )       SO
     1            / ( XO(NP,KP) - XO(NP,KM) ) + TO(NP,KM)                SO
      GO TO 20                                                           SO
 30   CONTINUE                                                           SO
 20   CONTINUE                                                           SOI
 10   CONTINUE                                                           SOI
C                                                                        SOI
C     NODAL TEMPERATURES FOR THE INTERFACE NODES                         SO
C     IN THE BLANKET REGION                                              SOI
      NDUM= NOD(NP) + NOBOUN                                             SOI
      IDUM= NYMAX - NOBOUN +1                                            SOI
      MDUM= NDUM + IDUM                                                  SOI
      IF(PINCRE .GT.0.D0)THEN                                            SOI
           RINTRP= PINCRE/(XO(NP,MDUM)- XO(NP,NDUM))                     SOI
           DO 44 JN= 1,IDUM-1                                            SOI
           L= NDUM+JN                                                    SOI
           K= L+IDUM                                                     SOI
 44        THETA(NP,L)= (TO(NP,K)-TO(NP,L))*RINTRP + TO(NP,L)            SO
      ELSE                                                               SOI
           RINTRP=(X(NP,NDUM)-XO(NP,NDUM-NYMAX))/                        SOI
     1             (XO(NP,NDUM) - XO(NP,NDUM-NYMAX))                     SO
           DO 55 JN= 1,IDUM-1                                            SO
           L= NDUM+JN                                                    SO
           K= L- NYMAX                                                   SO
 55        THETA(NP,L)= (TO(NP,L)-TO(NP,K))*RINTRP + TO(NP,K)            SO
      ENDIF                                                              SO
 1    CONTINUE                                                           SO
C                                                                        SO
      RETURN                                                             SO
```

FILE: SOLIC6 FORTRAN A VM/SP CMS RELEASE 3.1, CFD - U.C. BERKELEY

```
      END                                                               S
```

*LAWRENCE BERKELEY LABORATORY*
*TECHNICAL INFORMATION DEPARTMENT*
*UNIVERSITY OF CALIFORNIA*
*BERKELEY, CALIFORNIA 94720*