RE-491

SEGREGATION EFFECTS DURING SOLIDIFICATION IN WEIGHTLESS MELTS

(Interim Report)

August 1974

RESEARCH DEPARTMENT

GRUMMAN AEROSPACE CORPORATION
BETHPAGE NEW YORK

SEGREGATION EFFECTS DURING SOLIDIFICATION IN WEIGHTLESS MELTS

INTERIM REPORT

(Reporting Period: 5 July 1973 to 4 June 1974)

RE-491

by

Chou Li[†]

and

Morris Gershinsky[‡]

Research Department
Grumman Aerospace Corporation
Bethpage, New York 11714

Prepared in partial fulfillment of
Contract NAS 8-29662

for the

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center
Alabama 35812

August 1974

[†]Research Department, Grumman
Aerospace Corporation

[‡]Grumman Data Systems Corporation

Approved by: Charles E. Mack, Jr.
Charles E. Mack, Jr.
Director of Research

This report was prepared by Grumman Aerospace Corporation for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration in partial fulfillment of Contract NAS 8-29662. Our program on Segregation Effects in Weightless Melts has been partially supported by NASA/Marshall under the above contract.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ACKNOWLEDGMENT

The authors wish to thank Mr. R. C. Ruff of Marshall Space Flight Center for directing and monitoring this contract; they also wish to thank Dr. J. Bredt of NASA Headquarters in Washington, D.C. for helpful discussions.

SUMMARY

This report discusses the work partially supported under NASA Contract NAS 8-29662, "Segregation Effects During Solidification in Weightless Melts." The contract covers the period from July 5, 1973 to July 4, 1974.

During the contract period, the generalized problem of determining the temperature and solute concentration profiles during directional solidification of binary alloys with surface evaporation has been mathematically formulated. Realistic initial and boundary conditions have been defined, and a computer program has been developed and checked out.

The program computes the positions of two moving (evaporation and solidification) boundaries and their velocities of movement, and also the temperature and solute concentration profiles in the semi-infinite material body at selected instants of time.

The program has the following unique features:

- Two moving boundaries are involved, i.e., the evaporative boundary and freezing boundary
- Surface evaporation, and its related effects such as material loss, evaporative segregation, and surface cooling due to the heat of evaporation, have been considered
- Surface temperature is realistically determined by the combined effect of heat radiation, evaporative cooling, and thermal diffusion

- Material parameters such as solid and liquid densities, specific heats, thermal conductivities, mass diffusivities, and latent heat of fusion or evaporation, can all vary with both the temperature and composition
- Realistic phase diagrams involving curved liquidus and solidus lines are used

Our computer simulation work on solidification clearly shows that constitutional supercooling readily occurs and within-melt nucleation must then happen, particularly with reduced effective liquid mass transfer under zero-gravity conditions. Such results enabled us to explain and correlate some perplexing space solidification phenomena observed on Skylab, e.g., E. McKannan's weld (M551) and Prof. Adams' braze (M552) results (see Monthly Progress Reports Nos. 10 and 11). Detailed and quantitative application of the results of this computer program, however, still awaits the gathering of pertinent crystal growth data. A final report is expected to be written after these data are gathered and correlated.

INTRODUCTION

Space processing is moving closer to reality. Bigger, better, and more uniform single crystals of important semiconductors and welds or brazes of improved properties have already been made in space, as reported in the Third Space Processing Symposium at Marshall Space Flight Center. Although processing of structural materials may certainly have a profit potential in the long range, it appears that the high cost per pound of single-crystal electronic and optical materials makes these materials the most desirable contenders for immediate profitable returns from space processing. A selected single crystal study is, therefore, highly desirable to help us understand the segregation effects during solidification in weightless melts.

Important tools for understanding these segregation effects are analytic solutions or computer programs that simulate or predict what actually happens during space manufacturing. Such solutions and programs, furthermore, are probably necessary in space processing and other experiments where available time and experimental facilities are limited, the cost per sample or experiment is very high, and yet only a limited total number of tests or test samples can be conducted.

Theoretical predictions often greatly save time while computer simulation saves cost. Specifically, analytic solutions and computer programs allow us to answer many questions during the planning or execution of space experiments on material solidification, such as learning

- What phenomena are most important and what other phenomena are negligible

- Which influences are favorable to our understanding of weightless solidification and which are not
- What conditions lead to optimal combination of the favorable influences or elimination of the unfavorable ones
- What sample and processing conditions should be used
- What is the best way to analyze the resultant samples for understanding a particular phenomenon or influence
- How to save time and money — that is, how to maximize scientific return

We have developed a number of analytic solutions relating to solidification and evaporation (Refs. 1-3). Several important computer programs have also been developed. Some of these solutions and programs were developed under our Contract NAS 8-27891, and they are already proving useful in correlating actual experimental results (Refs. 4 and 5).

These analytic solutions and computer programs are, however, still in their early stages of development. The physical models involved are very simple and require considerable improvements to be used for other applications. It is, therefore, an important objective of this contract to refine and improve these models and the resultant analytic solutions and computer programs.

These refined solutions and programs are more widely useful, have greater predictive value, and provide more accurate results. Such accuracies are absolutely necessary to separate the rather subtle zero-gravity effects on solidification, in the presence of noise due to other unavoidable or unanticipated but ever-present

miscellaneous effects. As a result of this continued work, more efficient space experiments and greater scientific returns appear possible. More meaningful solidification experiments and fuller utilization of the unique space environment may also result.

The predicted results of our refined solutions and programs should, of course, first be checked with selected experiments. Another objective of this contract is, therefore, to design unique experiments to correlate the numerical results to actual solidification processes. This work is yet to be reported.

Review of Previous Contract

Under our NASA Contract NAS 8-27891, "Segregation Effects During Solidification in Weightless Melts" (Ref. 3), two types of melt segregation effects were studied: evaporative segregation, or segregation due to surface evaporation, and freezing segregation, or segregation due to liquid-solid phase transformation.

These segregation effects are closely related. In fact, evaporative segregation always precedes freezing segregation to some degree and must often be studied prior to performing meaningful solidification experiments. This is particularly true since evaporation may cause the melt composition, at least at the critical surface regions or layers, to be affected manyfold, often within seconds, so that at the surface region or layer the melting point and other thermophysical properties, nucleation characteristics, base for undercooling, and critical velocity to avoid constitutional supercooling, may be completely unexpected.

To predict the segregation effects of solidification time and temperature and to correlate these predictions with actual experimental data, "normal evaporation equations" were developed (Refs. 1, 4-6). An evaporative congruent temperature (or equi-evaporative

temperature) was then defined and listed for various binary or ternary alloys. Knowing these congruent temperatures and the solute and solvent evaporating rates, one can predict the type (solute depletion or enrichment) and magnitude of compositional or constitutional changes on the critical melt surface. One application of this unique temperature is to explain, predict, or plan "anomalous" evaporative or constitutional melting (on cooling) or solidification (on heating) experiments. We then computed for a simple model the reactive jetting forces due to surface evaporation and, in particular, showed that these forces can be very substantial on a differentially heated sample and may completely destroy the unique zero-gravity environment in space manufacturing (Ref. 7). In addition, these jetting forces may initiate surface deformation and vibration or other fluid disturbances, and may even produce some convection currents not normally anticipated. These studies also showed which sample materials are preferable, which should be avoided, and what impurities are harmful in producing excessive jetting or effective as stabilizing influences. The relationship between normal evaporation and normal freezing was then considered. Finally, applications of evaporation to space manufacturing concerning material loss and dimensional control, compositional changes, evaporative purification, surface cooling, materials standards, and freezing data interpretation were briefly described.

In the area of segregation due to solidification, we explained in some detail the normal freezing process and its successful use in the semiconductor industry. Various constitutional diagrams demonstrated the desirability of using nonconstant segregation coefficient techniques in metallurgical studies. We then stated the basic normal freezing differential equation, together with its

solutions for cases where the liquidus and solidus are quadratic, cubic, high-degree polynomial, and exponential functions of the melt temperature. The meaning of constant segregation coefficient was discussed, together with the associated errors due to curvatures of the liquidus and solidus lines and the best value of constant segregation coefficient for a given solidification experiment. Numerical methods for computing the normal freezing behavior were then given. Finally, as an example, the steady state solidification of the Ni-Sn system under conditions of limited liquid diffusion and nonconstant segregation coefficients was described. This system was studied in the M553 experiment on Skylab.

IMPORTANCE OF EVAPORATION

Evaporation is important in space melting and solidification for the following reasons:

- Significant evaporation of alloy components always occurs at high temperatures in space vacuum environments
- High-temperature evaporation of alloys is generally a neglected area of systematic research. Yet, unless the complete evaporative segregation behavior is understood and analyzed, solidification and its related segregation effects may not be properly studied because of ill-defined initial conditions. Before the liquid alloy can be controllably solidified or even melted, there is invariably some surface evaporation to cause changes in composition, freezing temperature, supercooling characteristics, nucleation and growth morphology conditions, and the like
- Controlled space evaporation probably most closely meets the requirements of our model of normal evaporation. We may thus be able to obtain material purity or evaporation standards, thermal properties, or even such basic thermodynamic properties as heat of evaporation, activity coefficients, and sticking coefficients that are difficult or impossible to obtain on earth

- Evaporation is a much simpler process than freezing, since the former does not involve such complicated phenomena as nucleation, phase transformation, and constitutional or nonconstitutional supercooling. Thus, in normal evaporation for specific geometries or alloy systems, we may ideally isolate and investigate such other phenomena as heat conduction or radiation, liquid or solid diffusion, fluid dynamics, and convection currents. Exact knowledge of these phenomena is necessary to understand solidification

- Evaporation causes surface cooling due to the heat of evaporation. This evaporative cooling effect is particularly important in low-melting materials (Ref. 8)

- Different rates of evaporation at various surface regions give rise to unbalanced forces and momenta that may produce erratic or unwanted accelerations, surface distortions and vibrations, exceedingly large "equivalent gravities," and possibly new types of powerful convection currents in zero-gravity conditions

- Evaporation may cause the surface composition of certain unwanted or unsuspected impurities to be increased a thousandfold or millionfold within seconds so that the layer's melting point and other thermophysical properties, nucleation characteristics, base for undercooling, and critical velocity to avoid constitutional supercooling may be completely

unexpected. In fact, anomalous "constitutional" or evaporative melting on cooling, or solidification on heating, is possible because of surface evaporation. In addition, very large artificial gravities (e.g., 10 g), strong fluid disturbances, or even new and significant convection currents may be produced from surface evaporation. These phenomena have been observed in the M553 movies, according to Dr. Martin Tobin of Westinghouse Co., Pa.

The much greater evaporative segregation effects, if unaccounted for, would almost certainly conceal any minor or subtle zero-gravity effects, particularly in the presence of other unknown or uncontrolled effects. Definitive space solidification work should probably, therefore, be preceded by an evaporative compatibility study of the sample materials and their possible associated impurities. In fact, evaporation is almost certain to be very important or so overwhelming that the effect of zero-gravity or freezing segregation may be masked or even reversed. A freely suspended molten drop in space may, for example, have its surface solute concentration greatly enriched (as much as a millionfold), by neglected and undetectable trace impurities within seconds of its deployment. We are then dealing at the critical surface layer with a completely new and unanticipated alloy having an entirely different composition, melting point, surface tension, thermophysical properties, latent heat of fusion, undercooling and nucleation characteristics, growth morphology, and the like.

From this we can also see that any analytical, numerical, or experimental study on solidification may yield completely unexpected or irrelevant results if the important and ever-present evaporation phenomena is not adequately taken care of. This is

particularly true in the study of nucleation, undercooling, and space manufacturing. Another important aspect of the present contractual work is to incorporate this generally neglected evaporation phenomena to define the exact initial and boundary conditions before and during the alloy solidification process.

COMPUTER PROGRAM WITHOUT SURFACE EVAPORATION AND RADIATION

Solidification, even in one-g, is a complicated process involving a multitude of interrelated phenomena such as mass and heat transfer, phase change, and fluid motion. Comprehensive reviews on solidification have been given, for example, by Chalmers (Ref. 9), Tiller (Ref. 10), Christian (Ref. 11), and Li (Ref. 12).

Solidification in zero-g is still very complicated. Here, gravitational force is negligibly small, but other effects as a result become important. For example, surface tension often plays a dominant role in determining the sample shape, processing technique and the resulting contamination level of the processed samples. Evaporation is another ever-present, complicating or dominating factor, but one that may be used to advantage when understood. Neglected, or improperly controlled evaporation may drastically change sample surface composition, fluid motion, equivalent gravities, nucleation, and undercooling characteristics as previously described. The previous program, under Contract NAS 8-27891, however, does not deal with evaporation.

Mathematical Definition of Solidification Problem

To understand thoroughly solute segregation either from combined evaporation and solidification, or in single-crystal growth, one requires a complete characterization of the (mass) diffusion and temperature fields in the solid crystal and remaining melt. The zero-gravity effect on the solidification may be overshadowed by other effects invariably present (such as evaporation) in any such growth process — a condition necessitating that such characterization be accurately defined. Unfortunately, the coupled partial differential equations for the diffusion and temperature

fields are generally not solvable. Although special case solutions have been given for some types of usually physically nonsatisfying, two-phase Stefan problems, for the general case solution we must resort to numerical computations. Existing numerical methods are always subject to such unrealistic assumptions as constancy of interfacial velocity, temperature or temperature gradients, segregation coefficients, diffusion constants, and other material thermophysical properties.

Under NAS 8-27891, a number of computer programs were developed to study the unidirectional solidification of a binary alloy. These programs employ analytical and numerical methods. The analytic program is based on some closed-form solutions of a simple model and gives results for our numerical program to compare. The model for the analytic program deals with a binary alloy at a constant temperature and concentration throughout the initial liquid melt, with the surface temperature instantaneously dropped below the liquidus temperature. The liquid-solid interface temperature is assumed constant, and the concentrations of the alloy at the interface are given by the phase diagram having curved liquidus and solidus lines. In addition, the interface boundary plane moves according to a square root law relative to the solidification time. The program also allows the interface temperature and interface boundary to vary from these fixed rules, but in practice the variation is negligible and not above the computer error level (Ref. 3).

Although covered in detail in the final report on NAS 8-27891, the mathematical formulation of the model is presented below for the sake of completeness.

We deal in unidirectional solidification with a liquid binary alloy to be directionally solidified into two phases, liquid and

solid. We consider the liquid alloy to be semi-infinite with original (at $t = 0$) temperature T_o and concentration C_o. Solidification occurs when the temperature at $x = 0$ is changed from T_o to a lower value T_1, either instantaneously or gradually, so that T_1 is below the temperature T_2 at which the liquid mixture at concentration C_o can be in equilibrium with a solid phase. As solidification occurs, the solid phase grows and its boundary is located at $x = y(t)$, and the interface temperature at this point is $T_i(t)$. The partial differential equations describing the solidification process are the following:

$$a_s^2 \frac{\partial^2 T_s}{\partial x^2} = \frac{\partial T_s}{\partial t} \quad , \quad D_s \frac{\partial^2 C_s}{\partial x^2} = \frac{\partial C_s}{\partial t} \quad \text{for} \quad 0 < x < y(t) \qquad (1)$$

$$a_\ell^2 \frac{\partial^2 T_\ell}{\partial x^2} = \frac{\partial T_\ell}{\partial t} \quad , \quad D_\ell \frac{\partial^2 C_\ell}{\partial x^2} = \frac{\partial C_\ell}{\partial t} \quad \text{for} \quad y(t) < x < \infty \qquad (2)$$

where the variables T, C represent the temperature and concentration (of solute in solvent) and the subscripts ℓ, s denote the liquid and solid phases, respectively. The thermal and mass diffusion coefficients a_s, a_ℓ, D_s, D_ℓ are assumed constant. The following conditions are usually assumed throughout:

(a) $T_\ell(x,0) = T_o$ and $C_\ell(x,0) = C_o$

(b) $T_\ell(\infty,t) = T_o$ and $C_\ell(\infty,t) = C_o$

(c) $T_s\big(y(t),t\big) = T_\ell\big(y(t),t\big) = T_i(t)$

(d) $C_s\big(y(t),t\big) = f_s\big(T_i(t)\big)$

(e) $C_\ell\big(y(t),t\big) = f_\ell\big(T_i(t)\big)$

(f) $\rho\gamma\dot{y}(t) = k_s \dfrac{\partial T_s}{\partial x} - k_\ell \dfrac{\partial T_\ell}{\partial x}$ for $x = y(t)$

(g) $\Big[f_s\big(T_i(t)\big) - f_\ell\big(T_i(t)\big)\Big]\dot{y}(t) = D_\ell \dfrac{\partial C_\ell}{\partial x} - D_s \dfrac{\partial C_s}{\partial x}$

for $x = y(t)$

In many cases, it is also assumed

(h) $y(t) = \alpha \sqrt{t}$.

Equation (a) describes the condition that the original mixture is all liquid at temperature T_o and concentration C_o. Equation (b) is a consequence of the semi-infinite nature of the mixture so that at any time t, the portion near infinity is unchanged. Equation (c) assumes that at the solid-liquid interface plane there is an interface temperature $T_i(t)$ and that both the solid and liquid phases at $x = y(t)$ have this temperature. There is no discontinuity in temperature. Equations (d) and (e) state that the concentrations of solid and liquid at the interface are given by the solidus and liquidus curves, respectively, of the constitutional diagram for the alloy. Equation (f) connects the derivative of the moving boundary with the redistribution of temperature and Eq. (g) connects the same boundary with that of concentration. Equation (h) relates the position of the interface boundary to the solidification time t.

The conditions on $T_s(0,t)$ and $C_s(0,t)$ are not fixed in our discussion, and a number of alternatives are considered:

1. $T_s(0,t) = T_1(t)$ with $T_1(t)$ equal to a constant for all t;

2. linear, $T_1(t) = T_o + t(T_1 - T_o)/s$ for $t < s$ and $T_1(t) = T_1$ for $t \geq s$;

3. exponential, $T_1(t) = T_1 + (T_o - T_1)\, e^{-t/s}$ so $T_1(0) = T_o$ and $T_1(\infty) = T_1$.

For $C_s(0,t)$ the conditions considered are $C_s(0,t) = C_1$ usually taken $C_s(T_2)$ or at times a condition conserving mass between 0 and ∞.

The two approaches we have pursued may be designated as analytic and numerical. The numerical approach can be applied to all three conditions on temperature whereas the analytic approach holds only the case of constant temperature instantaneously applied. A variant of this analytic method to apply to linear varying temperature has been investigated.

An analytic solution to the coupled partial differential equations (1) and (2) subject to the initial and boundary conditions (a) through (g) has been given (Ref. 13). A numerical program has been designed for the analytic solution.

These numerical programs developed under NAS 8-27891 are based upon finite difference approximations of the partial and ordinary derivatives and involve a variable spacing (for improved computing efficiency). The programs have given acceptable results and compared well with the reference analytic solution, where comparable. The basic physical properties such as densities, diffusivities, specific heats, thermal conductivities, and heat of fusion have been held to be constant, and independent of temperatures and concentrations.

COMPUTER PROGRAM DEVELOPED UNDER PRESENT CONTRACT

Under the present contract, we have extended the programs to allow for reasonable variation of these physical properties. The approach that has first been taken is to base the values of these physical properties upon extrapolated values of temperature and concentration, and then to determine the values of temperature and concentration. The process is then repeated by re-evaluations of the physical properties. Other modifications of our original program are: 1) to store the physical properties for each of the mesh points and to employ the appropriate quantities at each step, and 2) to recheck the mass and heat diffusion equations to make certain that the constancy of these properties is not assumed.

An additional major program modification has been the inclusion of evaporation effects. This includes evaporation before solidification that is mathematically identical to the problem of simple solidification in binary alloys. After solidification starts, significant evaporation may still exist. We then have to deal with two moving (solid-gas and solid-liquid) boundaries located at $y(t)$ for evaporation and at $z(t)$ for solidification, as will be described.

Modification of the initial and boundary conditions a-h has also been made to make the problem more physically meaningful. One such modification is to include a surface heat radiative loss term involving T^4. This term affects the convergence of the problem and creates the need for different algorithms. As reported previously, (Ref. 14), the surface cooling due to evaporation is negligible for many metallic systems such as nickel and iron alloys, or other higher melting materials, and has not been studied in detail under this contract.

To obtain solutions for realistic boundary conditions and to include the various mass transfer effects, numerical solutions of the partial differential equations of heat and mass transfer are required. We have again used the finite difference method to obtain the numerical solution.

The boundary conditions for surface temperature include radiation cooling as given by the Stefan-Boltzmann equation and also include evaporative cooling for both components of the alloy. Raoult's law has been assumed in determining the evaporation rates. At the interface it is assumed that the temperature and concentration relationships for each phase are given by the constitutional diagram for the alloy. The temperature dependence of the thermal and mass diffusion coefficients are allowed for each phase.

GENERALIZED SOLIDIFICATION WITH SURFACE EVAPORATION

Evaporative Solidification of a Binary Alloy

Given a semi-infinite binary alloy melt, initially at concentration C_o and temperature T_o, we consider the solidification of the alloy due to surface heat loss by evaporation and radiation (Fig. 1). There are two separate regimes to be considered. The first is concerned with temperature and concentration variations before solidification begins; the evaporation causes the original liquid-vapor boundary to change. Thus, we have a moving boundary problem. The second regime begins with the solidification which introduces a boundary between the freezing solid and remaining liquid phases whose compositions, we assume, follow the phase diagram, i.e., solidus and liquidus curve relations hold. Consequently, after solidification begins, there are two moving boundaries: one is the evaporative boundary and the other is the freezing or solidification boundary.

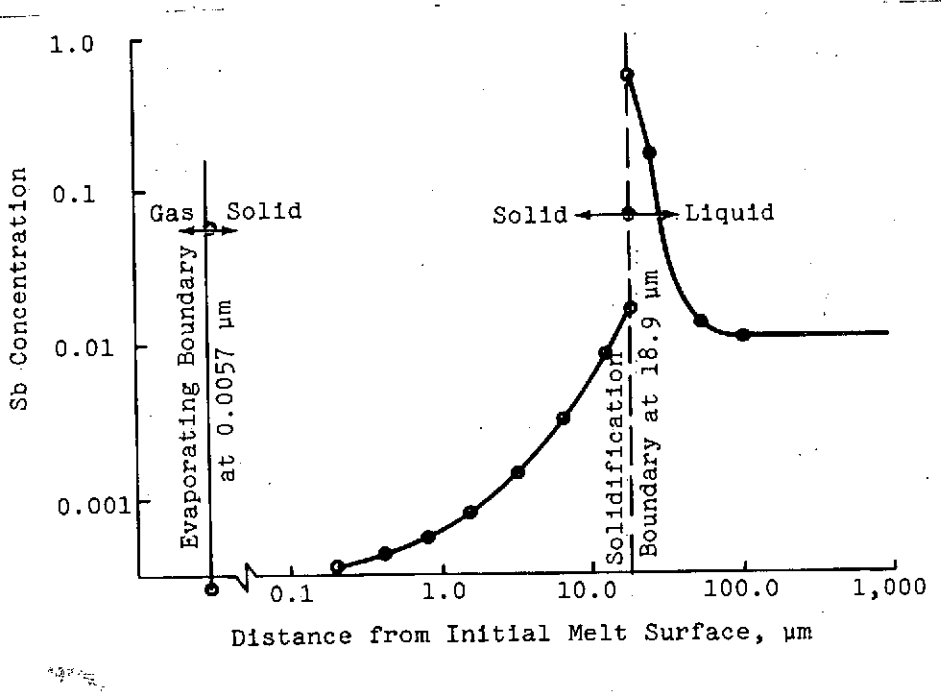

Fig. 1 Evaporation-Solidification of 0.01 Sb-Ge Initially at 970°C at 0.01 Second

Equations at the Evaporative Boundary

We denote the evaporative boundary as $x = y(t)$ where $y(0) = 0$. The evaporation rates in $mol/m^2/sec$ for pure solute and solvent are, respectively (Ref. 15),

$$U = K_e \, 10^{A_u - B_u/T} \, (M_u T_s)^{-\frac{1}{2}}$$

$$V = K_e \, 10^{A_v - B_v/T} \, (M_v T_s)^{-\frac{1}{2}}$$

where $K_e = 5.83 \times 10^{-5}$, M_u, M_v are molecular weights for solute and solvent atoms, T_s is the evaporating surface temperature in degree K, and A_u, B_u, A_v, B_v are the evaporating constants for solute and solvent, respectively. If ρ_u and ρ_v are the solute and solvent densities, then

$$\frac{dy}{dt} = \frac{UM_u C}{\rho_u} + \frac{VM_v(1 - C)}{\rho_v}$$

where C is the concentration at the moving boundary.

The heat loss rate equation at the boundary due to radiation and evaporation is given by

$$\frac{\partial T}{\partial t} = - \epsilon\sigma T_s^4 - U\gamma_u C - V\gamma_v(1 - C)$$

where ϵ is emissivity coefficient, σ the Stefan-Boltzmann constant, and γ_u and γ_v are specific heats for solute and solvent, respectively.

The equation for the rate of concentration change is

$$\frac{\partial C}{\partial t} = - (U - V)C$$

Since the evaporative boundary is a moving one, and since both the evaporation temperature T and solute concentration C are functions of distance $x = y(t)$ and time t, i.e., $T = T(x,t)$ and $C = C(x,t)$, the total derivatives $\frac{dT}{dt}$ and $\frac{dC}{dt}$ may be obtained from the partials, i.e.,

$$\frac{dT}{dt} = \frac{\partial T}{\partial t} + \left(\frac{\partial T}{\partial x}\right)_{x=y}\left(\frac{dy}{dt}\right)$$

$$\frac{dC}{dt} = \frac{\partial C}{\partial t} + \left(\frac{\partial C}{\partial x}\right)_{x=y}\left(\frac{dy}{dt}\right)$$

where $\frac{\partial T}{\partial x}$ and $\frac{\partial C}{\partial x}$ are evaluated at the moving boundaries.

Given $\frac{dy}{dt}$, $\frac{dT}{dt}$, and $\frac{dC}{dt}$, we can integrate for y, T, and C for the moving boundary using a modified Euler method.

$$v^{(i)}_{t+\Delta t} = v_t + \Delta t \left(\frac{dv}{dt}\right)_t$$

$$v^{(i+1)}_{t+\Delta t} = v_t + \frac{\Delta t}{2}\left[\left(\frac{dv}{dt}\right)_t + \left(\frac{dv}{dt}\right)^{(i)}_{t+\Delta t}\right]$$

where $\frac{dv^{(i)}}{dt}$ is the value of the derivative at time $t + \Delta t$ using the value $v^{(i)}$ for v.

To determine $\frac{\partial T}{\partial x}$ at time t and $t + \Delta t$ requires knowledge of the distribution of temperatures at both times. Those at time $t + \Delta t$ are initially approximated by an extrapolation and are corrected using an approximated value of the temperature of the evaporating boundary with the heat diffusion difference equations. Since the change in temperature at the boundary is greatest due to the heat of evaporation, more iterations are applied to determine

it than to the determination of temperature distribution by means of diffusion equations. Similar considerations hold for the determination of $\frac{\partial C}{\partial x}$ and $\frac{dC}{dt}$.

The computations of the position of the evaporation boundary [$y = y(t)$], temperature (T), and solute concentration (C) at this boundary constitute an initial value problem in ordinary differential equations. Thus, given $y_o = 0$, $T = T_o$, $C = C_o$, at time $t = 0$, and given also the equations for velocity of movement of this boundary dy/dt, and rate of change of temperature and solute concentration dT/dt, and dC/dt, we can determine for selected times the values of y, T, and C. The method used is an iterated Euler scheme:

$$y_{n+1} = y_n + \frac{\Delta t}{2} (y'_n + y'_{n+1})$$

where the initial value y'_{n+1} is taken as y'_n. This scheme must be connected to the problem of determining the temperature and solute concentration distribution within the semi-infinite body because the derivatives dy/dt, DT/dt, and dC/dt depend upon these quantities. The first step is to determine a first approximation of the temperature and solute concentration by extrapolation and then correct these values from the newly approximated values of the boundary position and the temperature and concentration thereat.

Start of Solidification

To determine the time when solidification has begun, the boundary temperature is compared with the temperature obtained by the inverse function for the liquidus curve evaluated at the boundary concentration. If the former is greater, then solidification has not yet begun. If it is smaller, then solidification has begun. In order to avoid an exact iterative procedure to determine the instant of solidification and to follow it up by a starting procedure for the first time interval thereafter, a simplified

approach has been taken that introduces a small error in the evaporative boundary and freezing boundary. By allowing the temperature to be below the solidification temperature by a small amount and by assuming that the temperatures at both boundaries are the same, a starting value of $x = z(t)$ of the freezing boundary is determined so that the loss in concentration due to solidification is compensated by the gain in concentration at the liquidus. Given the new temperature TI2 below the temperature at which solidification begins, we compute CSS = FS(TI2) and CLL = FL(TI2), the corresponding solid and liquid concentrations given by the phase diagram. To determine DEL2 = ZI2 - YI2, the distance between the evaporative boundary and solid-liquid interface, we assume that the solid is entirely at concentration CSS, and the liquid varies linearly from CLL to CC(II2), the concentration at the first mesh point x(II2) after the evaporative boundary. The total concentration is to equal the concentration in the whole regime had no solidification taken place. We assume it to be CL2 computed at YI2 and to vary linearly to CC(II2) at x(II2). This yields the equation

$$\text{CSS} * \text{DELZ} + \big(\text{CLL} + \text{CC(II2)}\big)/2 * \big(\text{x(II2)} - \text{YI2} - \text{DELZ}\big)$$

$$= \big(\text{CL2} + \text{CC(II2)}\big)/2 * \big(\text{x(II2)} - \text{YI2}\big)$$

Hence

$$\text{DELZ} * [\text{CSS} - \big(\text{CLL} + \text{CC(II2)}/2\big)] = \big(\text{x(II2)} - \text{YI2}\big) * (\text{CL2} - \text{CLL})/2$$

where

$$\text{DELZ} = (\text{CLL} - \text{CL2})/2 * \big(\text{x(II2)} - \text{YI2}\big) \div \big(\text{CLL} + \text{CC(II2)}/2 - \text{CSS}\big)$$

Then

$$\text{ZI2} = \text{YI2} + \text{DELZ} \quad \text{and} \quad \frac{dz}{dt} = \frac{\text{DELZ}}{\text{DELTS}} .$$

This enables us to begin the next time step with initial values for $y(t_s)$, $z(t_s)$, $\frac{dy}{dt}$, $\frac{dz}{dt}$, $T\left(y(t_s)\right) = T\left(z(t_s)\right) = TI2$, and $C\left(y(t_s)\right) = CSS$, $C_s\left(z(t_s)\right) = CSS$, $C_\ell\left(z(t_s)\right) = CLL$.

The Two-Boundary Problem-Derivative Estimation

The equations at the freezing boundary are those given in the Grumman Final Report RE-458 to Contract NAS 8-27891 (Ref. 3), with the exception that the freezing boundary is now called $x = z(t)$ and not $x = y(t)$ as in Eq. 49 c-g. At every time step we must compute (in addition to the temperature and concentration at the evaporation boundary) the temperature at the freezing boundary. The concentrations are determined by the phase diagram. The method we employ is that which determines T (the solidification temperature) and $\frac{dz}{dt}$ by means of Eq. (49) f,g. Having obtained $\frac{dz}{dt}$ we obtain $z(t)$ by means of a modified Euler method. Since the Eq. (49) f,g required approximation for $\left(\frac{\partial T_s}{\partial x}\right)_{z,t}$ and $\left(\frac{\partial C_s}{\partial x}\right)_{z,t}$, we must develop techniques for these approximations appropriate to various situations for mesh points. In addition, for the computation of $\frac{dy}{dt}$, $\frac{dT}{dt}$, and $\frac{dC}{dt}$ at the evaporative boundary, we also need $\left(\frac{\partial T}{\partial x}\right)_{y,t}$ and $\left(\frac{\partial C}{\partial x}\right)_{y,t}$. When there are two mesh points between y and z, then the techniques alluded to above are available. This involves determining $\frac{\partial^2 T}{\partial x^2}$ at both y and z and the same for $\frac{\partial^2 C}{\partial x^2}$. When there is only one mesh point between y and z, then $\frac{\partial^2 T}{\partial x^2}$ at both points are the same. When there are no mesh points between y and z, then we can assume either that $\frac{\partial^2 T}{\partial x^2}$ is zero and hence $\left(\frac{\partial T}{\partial x}\right)_y = \left(\frac{\partial T}{\partial x}\right)_z = \frac{T(z) - T(y)}{z - y}$ or that

$\frac{\partial^2 T}{\partial x^2} = k \frac{\partial T}{\partial x}$ and hence $\left(\frac{\partial T}{\partial x}\right)_y = \left(1 - \frac{z - y}{2} k\right)\frac{\partial T}{\partial x}$ and $\left(\frac{\partial T}{\partial x}\right)_z = \left(1 + \frac{z - y}{2} k\right)\frac{\partial T}{\partial x}$. The choice of k must be small so that $\frac{\partial^2 T}{\partial x^2} = k \frac{\partial^2 T}{\partial x^2} = k^2 \frac{\partial T}{\partial x}$ is negligible. Thus, since $z-y$ is also very small this option is indistinguishable from $\frac{\partial^2 T}{\partial x^2} = 0$. We have three cases: 1) no mesh points between two boundaries and we assume $\frac{\partial^2 T}{\partial x^2} = 0$, $\left(\frac{\partial T}{\partial x}\right)_y = \left(\frac{\partial T}{\partial z}\right)_z$, 2) one mesh point between y and z when $\frac{\partial^2 T}{\partial x^2}$ is obtained from the three points and $\left(\frac{\partial T}{\partial x}\right)_y = \frac{T(z) - T(y)}{z - y} - \frac{(z - y)}{2} \frac{\partial^2 T}{\partial x^2}$ and $\left(\frac{\partial T}{\partial x}\right)_z = \frac{T(z) - T(y)}{z - y} + \frac{(z - y)}{2} \frac{\partial^2 T}{\partial x^2}$, and 3) when two or more mesh points, say x_i and x_{i+1}, are between y and z so that we can compute $\left(\frac{\partial^2 T}{\partial x^2}\right)_z$ and $\left(\frac{\partial^2 T}{\partial x^2}\right)_y$ separately and distinct. Then $\left(\frac{\partial T}{\partial x}\right)_y = \frac{T(x_i) - T(y)}{x_i - y} - \frac{(x_i - y)}{2} \left(\frac{\partial^2 T}{\partial x^2}\right)_y$ and $\left(\frac{\partial T}{\partial x}\right)_z = \frac{T(z) - T(x_{i+1})}{z - x_{i+1}} + \frac{(z - x_{i+1})}{2} \left(\frac{\partial^2 T}{\partial x^2}\right)_z$.

In general, it is necessary to compute $\frac{\partial^2 T}{\partial x^2}$ and $\frac{\partial^2 C}{\partial x^2}$ in three ways, two ways indicated above for the solid regime and a third for the liquid side of the freezing boundary. It is similarly necessary to compute $\frac{\partial T}{\partial x}$ and $\frac{\partial C}{\partial x}$ in three ways.

Boundary and Mesh Points

When boundary points come close to mesh points, the computation of derivatives may be vitiated by closeness to mesh point.

Therefore, tests are made to determine when such closeness occurs as usually expressed in terms of a decimal fraction of the interval. In that case, the reference point is moved to the next mesh point and the values of T and C at the skipped mesh point are obtained by linear interpolation. This interpolation depends on which side of the solid-liquid interface the mesh point lies. For the evaporative boundary similar considerations hold.

Solution for Remaining Points

The solution for the remaining points is obtained as in the Final Report previously mentioned, pages 3-14 and 3-15 (Ref. 3). One change is, however, necessary because the first mesh point (or more) are no longer under consideration if the evaporative boundary has passed them. The subroutine TRIST is used to solve for the remaining points. In this subroutine we compute the values of temperature and concentration at intermediate mesh points when given the values at the two extreme mesh points. We replace the values at the mesh point to the left of the evaporative boundary by those at the evaporative boundary point, before solving for the intermediate points. This can be done without destroying any useful information since that mesh point is no longer used in the computations. The subroutine TRIST does not depend upon equal spacing or any regular spacing and therefore can accommodate this usage.

Convergence

The convergence problem is the crux of the program. Oscillation tends to cause the needed quantities to overflow. Thus, tests must be made on all the quantities to contain them within reasonable bounds. The subroutine MOTON is used to check the monotonicity of these consecutive points. In addition, the temperature at the evaporative boundary is necessarily less than the temperature at

the freezing boundary. This condition is always imposed in the program.

In addition, the solution for the solidification temperature and freezing boundary derivative (especially the latter) involves very rapidly changing quantities. More iterations should, therefore, be expended in this part of the program. Fewer iterations are needed for determining the evaporative boundary, and the temperature and concentration at that boundary. The program allows five iterations in the former for each of the latter. The number of iterations of the latter is used in a manner analogous to that described in Final Report RE-458 (Ref. 3).

An input quantity NIT (usually a multiple of 4) gives the maximum number of iterations. When NIT/2 iterations occur and convergence is not reached, the time step size is halved. This process is continued until either convergence is attained or the minimum step allowed by the program has been iterated NIT + 1 times. In this case the program may stop or continue on using the nonconverged quantities. Very often these quantities are sufficiently smooth so that convergence will occur on the next interval and the program gives satisfactory results.

However, if the program proceeds with the minimum step and the maximum number of iterations, the results may be spurious. In case of overflow, there is no doubt of it. Otherwise the user must look at results to decide whether he finds them reasonable.

IMPROVED COMPUTER PROGRAM

The complete computer program for the generalized solidification problem is listed herein (see appendix), together with a glossary explaining the special names used in the program. This computer program has the following unique features:

- Surface evaporation, and its related effects such as material loss, evaporative segregation, and surface cooling due to the heat of evaporation, have been considered
- Material parameters such as solid and liquid densities, specific heats, thermal conductivities, mass diffusivities, and latent heat of fusion or evaporation, are allowed to vary with the temperature and composition
- Realistic phase diagrams involving curved liquids and solidus lines are used
- Two moving boundaries are involved, i.e., the evaporative boundary and freezing boundary
- Surface temperature is determined by the combined effect of heat radiation, evaporative cooling, and thermal diffusion

Use of Computer Program

The computer program works well if the following three input program parameters are properly chosen: 1) time step size (DELT), 2) grid spacing (DELX), and 3) maximum iteration count (NIT).

The solidification boundary is sensitive to the grid spacing. This is because in passing through a mesh point, discontinuity in the computation occurs for the following reasons. We compute the derivatives in terms of the temperatures and concentrations at the discrete mesh points. When one mesh point is dropped because solidification occurs near it, the derivative based on a substituting new mesh point is discontinuous with that based on the previous mesh point. Though this discontinuity can be reduced by using a smaller time step size, it would be a self-defeating strategy. An alternative is to accept the discontinuous results as they occur, advantages being taken of the fact that the program corrects itself. Although the derivative dz/dt is large when the solid-liquid interface passes through a mesh point, it becomes smaller thereafter thereby correcting the solidification boundary position.

The frequency of this self-correction depends on the grid spacing. Too small a grid spacing would cause too frequent self-corrections. Too large a grid spacing, on the other hand, would obscure the rapid temperature variations around the solidification boundary. This indicates that a proper choice of the grid spacing is required to achieve an optimal tradeoff between accuracy and computing time. There is another tradeoff between the time step size and maximum iteration count for optimal computing results.

Since each evaporation-solidification problem represents a different and unique physical situation, each case must be dealt with separately. However, based on our experience, the following guidelines would be helpful:

The first consideration for the choice of the grid spacing is the behavior of the evaporation boundary after solidification begins. If the evaporation boundary is virtually stationary as

compared to the solidification boundary, the grid spacing should be chosen so that the evaporation boundary is within the first mesh interval (between the first and second mesh points). If, on the other hand, the evaporation boundary is moving at velocities comparable to those of the solidification boundary, then the grid spacing can be selected more freely. The major consideration in this case is the relationship between the grid spacing and the time step size. For a fixed time step size, the grid spacing should be chosen so that at least four time intervals (of step sizes) occur before the solidification boundary passes through a mesh point.

In cases where the evaporation boundary is virtually stationary, one must experiment to determine an optimum time step size in terms of accuracy and computing time. The conditions of the experiment are as follows. Set both the minimum time step size (DELTM) and the time printing interval (DELP) to zero. Setting the time printing interval to zero will cause the computer to print out every computer time step. Setting the minimum time step size to zero will not cause the program to cut back indefinitely but will use, as the minimum, the time step size divided by 1024. By examining the computed results, one can see at what time step sizes the program is running. By examining the actual iteration count (IT), one can see if the program is converging or not. If not converging repeatedly, a smaller time step size is indicated. If the program is converging most of the time, then the minimum time step size can be set at the level of the most frequent time step size and the actual iteration count re-examined to see if the program still converges most of the time. For long runs, the time printing interval must not be zero or small, but must be chosen in consideration with the amount of the required output.

To improve the computing time on long runs, one should consider enlarging the grid spacing as suggested above as one of the tradeoffs. In addition, one may change the maximum iteration count upwards or downwards to also improve the computing time.

Our computer program has the capability for assuming equal or unequal (doubling) mesh point spacings. Our experience, as indicated in Tables 1-4, shows that the unequal spacing scheme gives practically the same accuracy with far less computations as compared with the equal spacing scheme. This may be due to the rapidity at which the temperature declines at the evaporation boundary. Other physical situations may give different results and may indicate that the equal spacing scheme should be used.

The program input parameters consist of a set of integers IX, IAM, NIT, IM, N, and NCN; and a set of real numbers DELX, DELT, DELTM, DELP, TF, and S. IX is the maximum number of mesh points to be used in the program. Present, IX 28. IAM is the spacing option indicator. If IAM equals 0, the points of mesh are equally spaced with grid spacing DELX. If IAM = 1, an unequal spacing is indicated. The first two intervals are equal and set to DELX. Thereafter, each interval is double the previous interval in spacing. NIT is the maximum number of iterations as interpreted in the context of halving the time step size. If the step is begun at the minimum time step, the NIT is the maximum number of iterations allowed. IM is the number of mesh points in actual use. The input value of IM introduces the minimum number of mesh points to be used. Thereafter additional mesh points are added as required by a substantial change in temperature at next to last mesh point, that is, 1 degree below the initial temperature. IM is increased until IM is equal to IX.

TABLE 1 VARIATION OF TEMPERATURE (°C) AT EVAPORATIVE BOUNDARY FOR VARIOUS COMPUTATION SCHEMES

Scheme	I	II	III	IV	V
Grid Spacing	0.01 equal	0.01 unequal	0.001 equal	0.001 unequal	0.0001 cm unequal
time, ms					
0.2	966.5	966.5	966.5	966.5	966.5
0.6	959.4	959.4	959.4	959.4	959.5
1.4	945.7	945.7	945.7	945.7	945.8
1.8	938.9	938.9	938.9	938.9	939.0
2.0	935.5	935.5	935.5	935.5	935.6
2.05	934.7	934.7	934.7	934.7	N.C.
2.075	934.2	934.2	934.2	934.2	N.C.
2.0875	934.0	934.0	934.0	934.0	N.C.
2.09375	933.9	933.9	933.9	933.9	N.C.
2.1	933.8	933.8	933.8	933.8	933.9
2.1125	933.6	933.6	933.6	933.6	933.7*
2.1375	933.2	933.2	933.2	933.2	933.3*
2.1875	932.4	932.4	932.4	932.4*	932.5*
2.2875	930.8	930.8	930.8	930.8*	930.9*
2.4875	927.7	927.7	927.2	927.7*	927.7*
2.8875	921.5	921.5	921.5	921.5*	921.5*
3.6875	909.3	909.3	909.3	909.3*	909.3*
5.2875	885.8	885.8	885.8		

*Hand interpolations
N.C. not computed

TABLE 2 VARIATION OF POSITION (μm) OF EVAPORATIVE BOUNDARY WITH TIME FOR VARIOUS COMPUTATION SCHEMES

Scheme	I	II	III	IV	V
Grid Spacing	0.01 equal	0.01 unequal	0.001 equal	0.001 unequal	0.0001 cm unequal
time, ms					
0.2	0.122	0.122	0.122	0.122	0.122
0.6	0.351	0.351	0.351	0.351	0.351
1.4	0.752	0.752	0.752	0.752	0.752
1.8	0.927	0.927	0.927	0.927	0.928
2.0	1.009	1.009	1.009	1.009	1.010
2.05	1.029	1.029	1.029	1.029	N.C.
2.075	1.039	1.039	1.039	1.039	N.C.
2.0875	1.044	1.044	1.044	1.044	N.C.
2.09375	1.046	1.046	1.047	1.047	N.C.
2.1	1.047	1.047	1.047	1.047	1.050
2.1875	1.048	1.048	1.048	1.048*	1.052
2.2875	1.049	1.049	1.049	1.048*	1.053
2.4875	1.051	1.051	1.051	1.050*	1.055
2.8875	1.055	1.055	1.055	1.054*	1.059
3.6875	1.061	1.061	1.062	1.061	1.066
5.2875	1.072	1.072	1.072	-	-

*Hand interpolations
N.C. not computed

TABLE 3 VARIATION OF POSITION (μm) OF SOLID-LIQUID INTERFACE

Scheme	I	II	III	IV	V
Grid Spacing	0.01 equal	0.01 unequal	0.001 equal	0.001 unequal	0.0001 cm unequal
time, ms					
0.21	0.205	0.204	0.109	0.109	0.106
0.24875	0.211	0.211	0.179	0.180	0.408
0.28875	0.229	0.229	0.350	0.350*	1.03
0.36875	0.283	0.283	0.866	0.860*	2.76
0.52875	0.429	0.429	2.291	N.C.	N.C.

TABLE 4 VARIATION OF TEMPERATURE (°C) AT SOLID-LIQUID INTERFACE

Scheme	I	II	III	IV	V
Grid Spacing	0.01 equal	0.01 unequal	0.001 equal	0.001 unequal	0.0001 cm unequal
time, ms					
0.21	933.8	933.8	933.8	933.8	933.9
0.24875	927.7	927.7	927.7	927.7*	927.9*
0.28875	924.5	921.4	921.5	921.4*	921.6*
0.36875	909.3	909.3	909.3	909.3*	909.4*
0.52875	885.8	885.8	885.8		

*Hand interpolations
N.C. not computed

NONCN is a nonconvergence option. Failure to converge occasionally is not necessarily an indication of unacceptable results. Therefore, it is desirable to continue computations and examine the results to see if they are acceptable. This is done by setting NONCN to 1. If NONCN is set at 0, the nonconvergent results are not printed unless called for by the print interval. If NONCN is -1, the program stops on nonconvergent results.

The quantity DELX is the grid spacing. Equal spacing and unequal double spacing both make use of this quantity as indicated in the discussion of IAM. The quantity DELT is the maximum time interval (step size) for computation. The quantity DELTM is the input minimum time interval. The program uses as its actual minimum the larger of the quantities DELT/1024 and DELTM. Thus, even if DELTM is set at 0, the number of halving on cutting back the time interval is at most 10 ($2^{10} = 1024$). The program in its presolidification phase starts with its actual time step DELTS set to DELT/8 and allows it to build up to DELT by quick convergence.

On the other hand, near the beginning of solidification, DELTS is allowed to cut back to as small as DELT/256 in order to find an acceptable start of solidification. After solidification has begun, then the restriction of DELTS is between DELTK and DELT. If halving reduces DELTS below DELTK, it is set to DELTK. The quantity DELP is the present interval. If DELP = 0, then every time step is printed. TPR is the time for outputing results. TPR is set originally to DELP and after printout is reset to TPR + DELP. The program prints results if the time TIME1 at the end of the time step equals or exceeds TPR. The program does not attempt to set DELTS so that TPR = TIME1. This is only a slight inconvenience when the print interval is large as compared to DELTM. Generally, we would like DELTM to be set close to the most frequent

DELTS provided that failure to converge does not ensue on a regular basis. TF is the final time of program. This means that if TIME1 equals or exceeds TF, no additional time steps are taken.

The decimal quantity S between 0 and 0.5 is used to determine closeness to a mesh point. If the boundary point (either evaporation or solidification) is such that it exceeds the point that divided the mesh interval surrounding the boundary point in the ratio (1-S)/S, then the mesh reference point for computation is moved to the next mesh point. The introduction of S is to make the transition due to passing a mesh point less abruptly discontinuous. The best values of S are between 0.05 and 0.15. For computations on the solid side of the solidification boundary, we continue to use the old mesh points until the boundary point passes the point that divides the new mesh interval about the solidification point in the ratio S/(1-S). This strategy causes a gradual transition from one mesh point to another. The integers II1, II2 are used as reference point indicators for the solid and liquid sides, respectively. For the evaporation boundary, II3 is used to indicate which points are used. II4 is used only to indicate the first mesh point to the left of the evaporation boundary.

Typical Computer Input

The definitions of the various inputs fed into the computer are given in the Glossary of Program Parameters. Typical input values are as follows:

IX	=	28 = maximum number of mesh points
IAM	=	1, unequal, doubled grid spacing
NIT	=	20, maximum number of iterations
IM	=	16, actual number of points in mesh
NONCN	=	0, allowing the program to continue when non-convergence occurs with no special printout of these results.

The alloy phase diagrams are determined from the five constants ET, EA, EB, EC, and ED, which define the liquidus C_ℓ and solidus lines C_s as two functions of the temperature, T:

$$C_\ell(T) = ED \times (ET - T)^2 + EC \times (ET - T)$$

$$C_s(T) = EB \times (ET - T)^2 + EA \times (ET - T)$$

In our example of 10 mole percent ($C_o = 0.10$) of antimony in germanium initially uniform at 970°C ($T_o = 970$)

ET = melting point of pure germanium = 956°C
EA = 0.128812×10^{-3}
EB = -0.82218×10^{-7}
EC = 0.466678×10^{-2}
ED = -0.60466×10^{-5}

The evaporation constants for the solvent and solute as defined previously under "The Equation at the Evaporative Boundary" are:

AU = $A_u = 0.1115 \times 10^2$
BU = $B_u = 0.863 \times 10^4$
EMU = $M_u = 0.2435 \times 10^3$
AV = $A_v = 0.1171 \times 10^2$
BV = $B_v = 0.1803 \times 10^5$
EMV = $M_v = 0.726 \times 10^2$
EK = $K_e = 5.833 \times 10^{-4}$

The diffusion coefficient of the solute in solvent in the solid and liquid states are, respectively

DS = $D_s = 0.10 \times 10^{-6}$
DL = $D_\ell = 0.10 \times 10^{-3}$

The density ρ, and latent heat of evaporation, γ, of the pure solvent are, respectively,

RHO = ρ_v = 5.32

GAMMA = γ_v = 160

Corresponding values for pure solute are:

RHOU = ρ_u = 6.68

GAMMAU = γ_u = 39

The above give two derived quantities:

ALS = $a_\ell^2 = k_\ell/\rho_v c$

ASS = $a_s^2 = k_s/\rho_v c$

where

CEE = c = 0.740 x 10^{-1} = specific heat

The two input parameters in the surface radiation terms are:

EE = ϵ = 0.55 = emissivity coefficient, and

SIG = σ = 0.136 x 10^{-7} = Stefan-Boltzmann constant.

Computer Output

The first line of computer outputs gives the program input parameters IX, IAM, NIT, IM, and NONCN, which are defined previously and also in the "Glossary." The next two lines of computer output give the phase diagram constants (ET, EA, EB, EC, and ED) and evaporation constants (AU, BU, EMU, AV, BV, EMV, and EK), respectively. The next printouts are for CEE, DS, DL, TO, CO, XKL, RHO, GAMMA, RHOU, GAMMAU, EE, SIG, T2, and COO, where T2 is the temperature when solidification begins for the melt of initial solute concentration CO.

The computed numbers are then outputed as follows:

IT = actual iteration count

IM = number of points in mesh

II1 = grid point reference for solid side of mesh

II2 = grid point reference for liquid side of mesh

II3 = grid point reference for evaporation boundary

II4 = grid point reference for point after evaporation boundary

These printouts are then followed by the computed values associated with the evaporation boundary: time, location y, concentration C, temperature T, extent of points X(IM), current time interval DELTS, dy/dt DYDT1, dC/dt DCDT1, dT/dt DTDT1. If solidification has not begun, then there is no information about the solidification boundary. Otherwise, we have position of the solidification boundary z computer language (ZI1), solid solute concentration C_s (CS1), liquid solute concentration C_ℓ (CL1), temperature T (TI1), and rate of movement dz/dt (DZDT1). All decimal outputs are five per line with excess going to the next lines.

Representative Computed Results

The study of the effect of varying the grid spacing DELX on the computed results is summarized in Tables 1 through 4. The five cases considered are:

Case I: DELX = 0.01 cm with equal spacing

Case II: DELX = 0.01 cm with unequal spacing

Case III: DELX = 0.001 cm with equal spacing

Case IV: DELX = 0.001 cm with unequal spacing

Case V: DELX = 0.0001 cm with unequal spacing

Tables 1 and 2 indicate the insensitivity of the evaporation boundary and its temperature to grid spacing, provided that the spacing is always larger than the evaporation boundary point. Tables 3 and 4 involve solidification boundary and show that the temperature is insensitive to DELX but that the solidification boundary is quite sensitive to the choice of DELX. Thus, it is important to use DELX sufficiently small so that the solidification boundary movement is fully exhibited and not stunted by a large grid spacing relative to which the boundary size is small. The spacing affects the evaluation of the first and second temperature partial derivatives with time which are larger in absolute values for smaller spacings, due to more rapid temperature changes near the boundaries.

The figures (Figs. 2-4), prepared from the computed results in Tables 1-4, indicate the superiority of unequal over equal grid spacing. For DELX = 0.01 cm, where the spacing is coarse, little difference is found in the temperature distribution. For DELX = 0.001 cm, there is greater difference between the two because the equal spacing has limited the semi-infinite body to 27 (0.001 cm) and fixes the temperature at the end point to 970°C, thus not allowing the temperature to decline as rapidly as it should. For DELX = 0.0001 cm, the equal spacing method could not work at all because 27 (0.0001 cm) is too small a range to define a semi-infinite body even for the small time constants under consideration.

The second set of graphs, Fig. 3 (grid spacing DELX = 0.001), shows wide disparity between equal and unequal spacing whereas the first set of graphs (Fig. 2) (DELX = 0.01) shows good agreement. The smaller DELX needs more points to simulate the semi-infinite, one dimensional case and when restricted to IX = 28, fails to

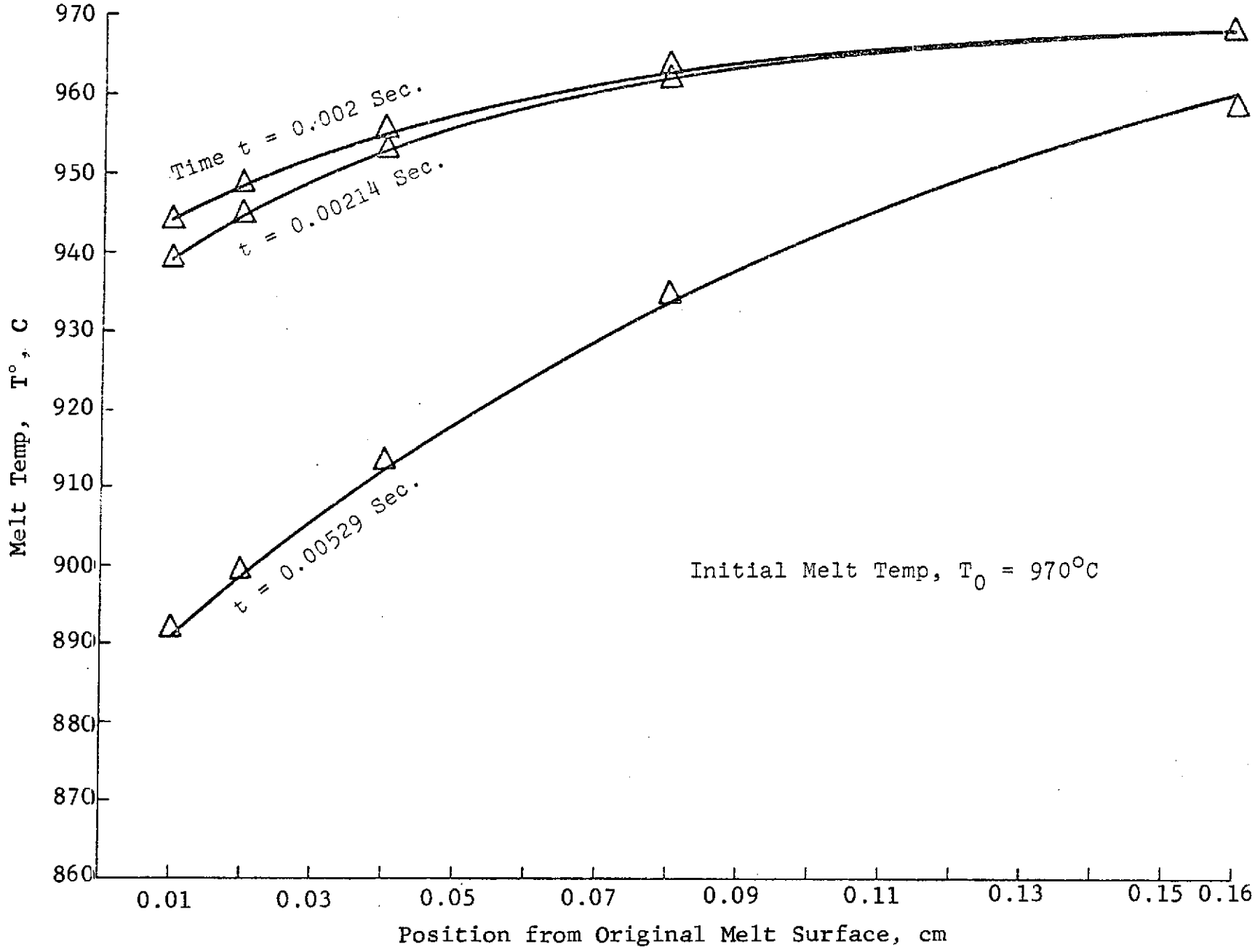

Fig. 2 Computed Temperature Distribution in Semi-infinite Body at Different Times, t (initial grid spacing = 0.01 cm)

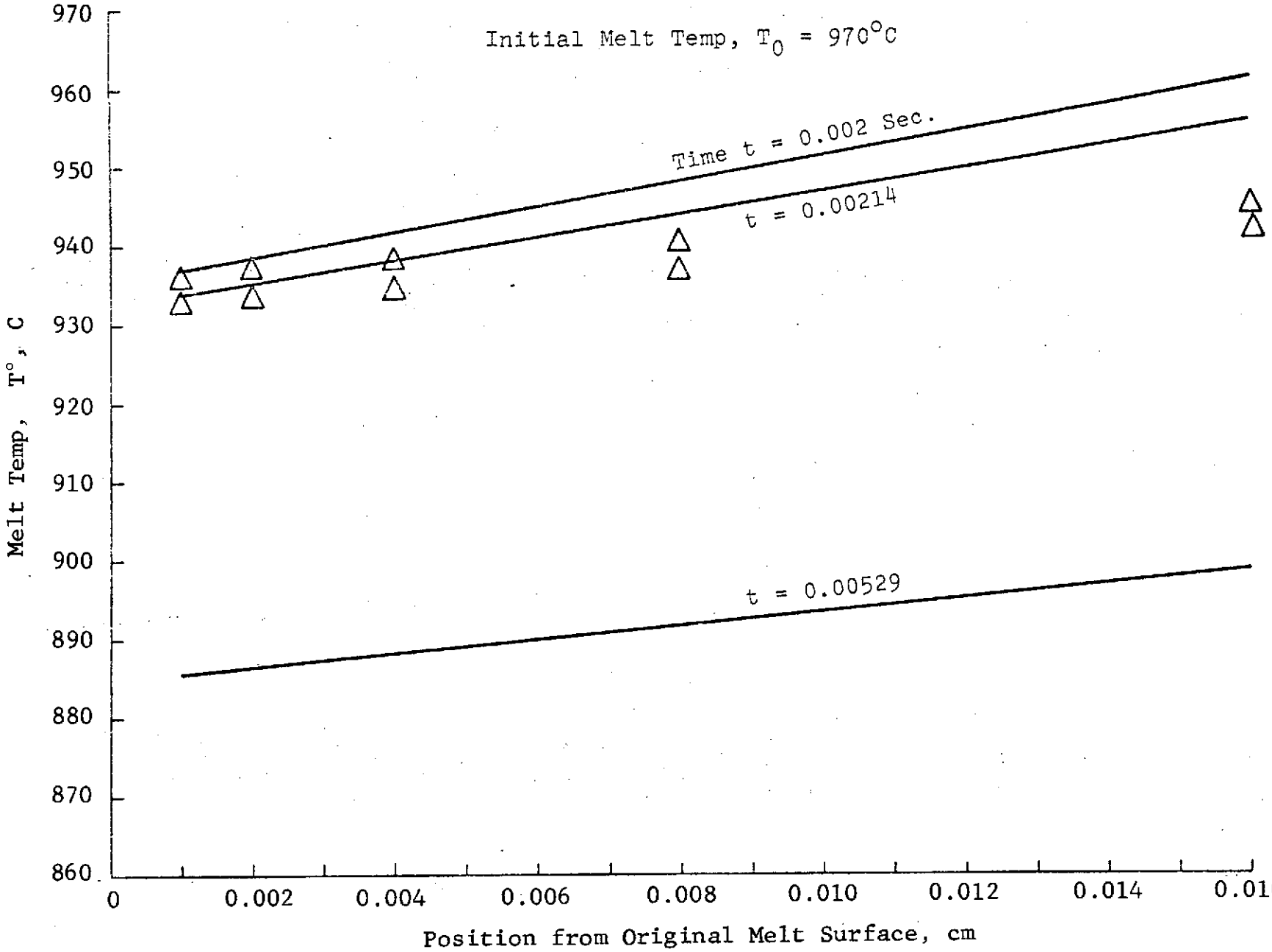

Fig. 3 Computed Temperature Distribution in the Semi-infinite Body at Different Times, t (initial grid spacing = 0.001 cm)

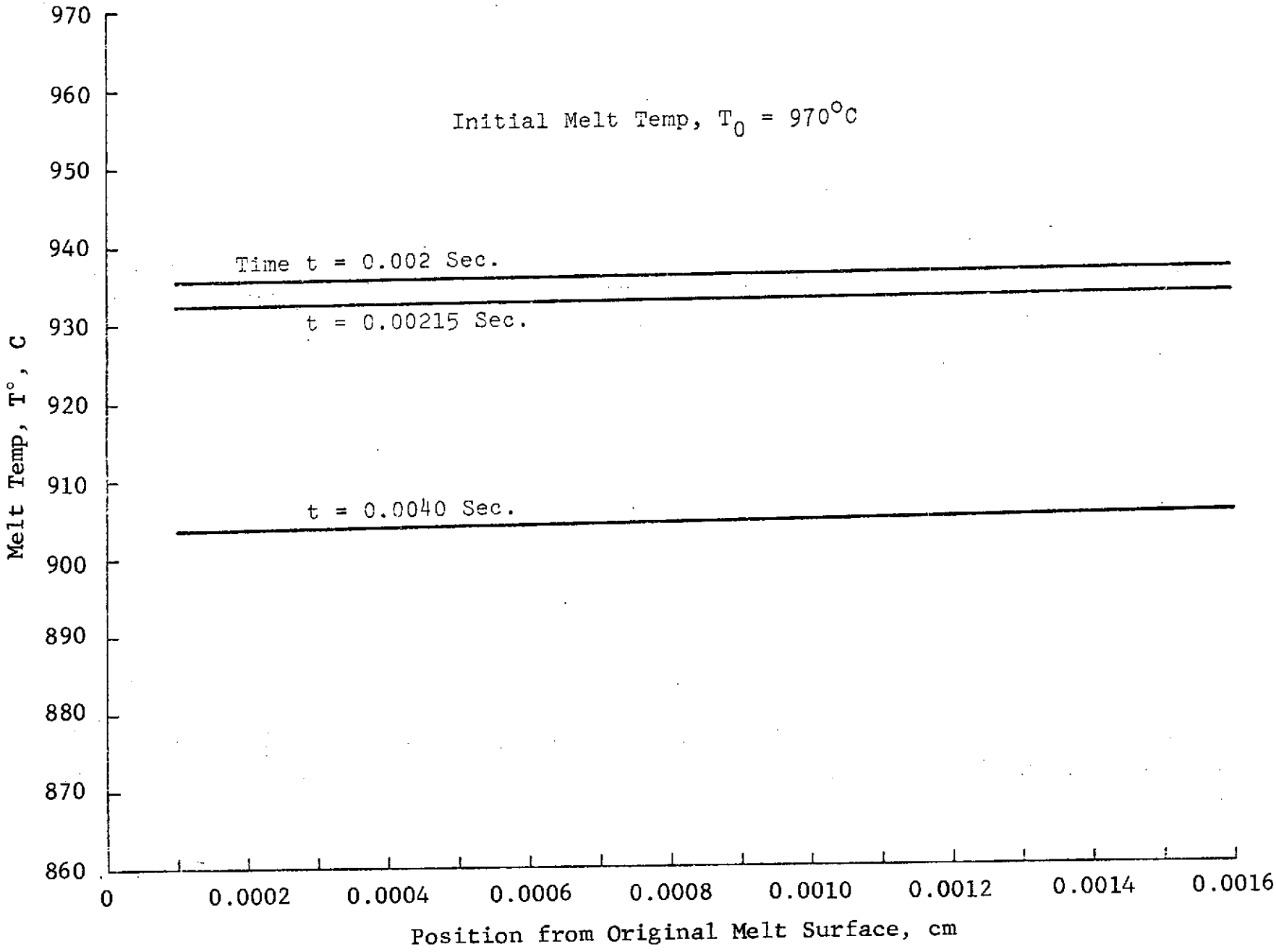

Fig. 4 Computed Temperature Distribution in Semi-infinite Body at Different Times, t (initial grid spacing = 0.0001 cm)

allow temperature away from the evaporating surface to decline rapidly because it is artificially pegged at x = 28 (0.001) to 970°. The unequal spacing needs but six points to give equivalent extension and, when given 10 or 11 points, can adequately span a sufficient distance to simulate semi-infinity. At smaller DELX (0.0001) one cannot even attempt to use equal spacing without modifying the behavior at the last mesh point. For unequal spacing, 16 points will adequately represent the semi-infinite body for the times under consideration.

Checking Program

To check the program, GaAs single crystals will be grown with controlled dopant type, dopant concentration, growth rate, and temperature gradient, as shown in Table 5. The dopant concentration, carrier mobilities, defect contents, ... will be measured along several sections on each crystal. The results will be statistically analyzed and presented in future reports.

TABLE 5 GaAs CRYSTAL GROWTH SCHEDULE

Crystal No.	Dopant	Concentration	Growth Rate in./hr	Temperature Gradient °C/in.
1	Te	1×10^{17}	0.16	8
2	Si	5×10^{18}	0.22	8
3	Cr	5×10^{18}	0.28	6
4	Si	5×10^{17}	0.16	6
5	Zn	5×10^{18}	0.16	4
6	Cr	1×10^{18}	0.16	10
7	Te	1×10^{18}	0.22	6
8	Zn	1×10^{18}	0.28	8
9	Cr	5×10^{17}	0.10	8
10	Si	1×10^{18}	0.10	4
11	Si	1×10^{17}	0.28	10
12	Zn	1×10^{17}	0.10	6
13	Te	5×10^{17}	0.28	4
14	Zn	5×10^{17}	0.22	10
15	Cr	1×10^{17}	0.22	4
16	Te	5×10^{18}	0.10	10
17				
18				
19	To be grown after the results of above crystals			
20	are analyzed.			

CONCLUSIONS

We have developed a computer simulation program to study the phenomena of directional combined evaporation and solidification in binary alloys. A realistic phase diagram involving curved liquidus lines is used, and the program can be used for cases where the solid and liquid material parameters (e.g., specific heat, conductivity, diffusivity, ...) are functions of both temperature and solute concentration. The program works well if one follows the guidelines outlined in the report. The computed output results include the locations and velocities of movement of both the evaporation and solidification boundaries, and the temperature and concentration profiles in the semi-infinite alloy body at selected instants of time.

APPENDIX I

IMPROVED COMPUTER PROGRAM

```
      DIMENSION X(28),T(28),TT(28),C(28),CC(28),                        BIN00010
     *TEM(10)      ,A(84)                                               BIN00020
      NAMELIST /INVAR/CEE,DS,DL,T0,C0,XKL,RHO,GAMMA,RHOU,GAMMAU,EE,SIG  BIN00030
     *,ET,EA,EB,EC,ED,AU,BU,EMU,AV,BV,EMV,EK,IX                         BIN00040
      D2(X,F,Y,G,Z,H)=((H-G)/(Z-Y)-(F-G)/(X-Y))/(Z-X)*2.                BIN00050
      ABS1(X)=AMAX1(1.,ABS(X))                                          BIN00060
      UE(V)=EK*(10.**(AU-BU/(273.12+V)))/SQRT(EMU*(273.12+V))           BIN00070
      VE(V)=EK*(10.**(AV-BV/(273.12+V)))/SQRT(EMV*(273.12+V))           BIN00080
      FS(V)=(EB*(ET-V)+EA)*(ET-V)                                       BIN00090
      FL(V)=(ED*(ET-V)+EC)*(ET-V)                                       BIN00100
      XCL(V)=ET-2.*V/(EC+SQRT((EC)**2+4.*(ED)*(V)))                     BIN00110
      XCSL(V)=ET+2.*V/((EC-EA)+SQRT((EA-EC)**2+4.*(EB-ED)*V))           BIN00120
      DFL(V)=-(2.*ED*(ET-V)+EC)                                         BIN00130
      II=1                                                              BIN00140
      IO=8                                                              BIN00150
      READ(II,100) IX,IAM,NIT,IM,NONCN                                  BIN00160
  100 FORMAT (16I5)                                                     BIN00170
      NITH=NIT/2                                                        BIN00180
      NITQ=NITH/2                                                       BIN00190
      NITL=NITH+NITQ                                                    BIN00200
   64 READ(II,101) ET,EA,EB,EC,ED                                       BIN00210
      AQUAN=-(EA-EC)**2/(4.*(EB-ED))                                    BIN00220
  101 FORMAT(7E10.0)                                                    BIN00230
      READ(II,101) AU,BU,EMU,AV,BV,EMV,EK                               BIN00240
      READ(II,101) CEE,DS,DL,T0,C0,XKL,RHO,GAMMA    ,RHOU,GAMMAU,EE,SIG BIN00250
      ALS=XKL/(RHO*CEE)                                                 BIN00260
      XKS=1.1*XKL                                                       BIN00270
      ASS=1.1*ALS/1.03                                                  BIN00280
      AS=SQRT(ASS)                                                      BIN00290
      AL=SQRT(ALS)                                                      BIN00300
      READ(II,101) DELX,DELT,DELTM,DELP,TF,S                            BIN00310
      DELTK=AMAX1(DELTM,DELT/1024.)                                     BIN00320
      T2=XCL(C0)                                                        BIN00330
      C00=FL(T2)                                                        BIN00340
  201 DO 1 I=1,IX                                                       BIN00350
      IF(I-2) 2,3,4                                                     BIN00360
    2 X(1)=0.                                                           BIN00370
      GO TO 1                                                           BIN00380
    3 X(2)=DELX                                                         BIN00390
      GO TO 1                                                           BIN00400
    4 IF(IAM) 5,5,6                                                     BIN00410
    5 X(I)=X(I-1)+DELX                                                  BIN00420
      GO TO 1                                                           BIN00430
    6 X(I)=X(I-1)+X(I-1)                                                BIN00440
    1 CONTINUE                                                          BIN00450
  999 WRITE(IO,100)IX,IAM,NIT,IM,NONCN                                  BIN00460
      WRITE(IO,102)ET,EA,EB,EC,ED                                       BIN00470
      WRITE(IO,102)AU,BU,EMU,AV,BV,EMV,EK                               BIN00480
      WRITE(IO,102)CEE,DS,DL,T0,C0,XKL,RHO,GAMMA ,RHOU,GAMMAU,EE,SIG,T2,BIN00490
     *C00                                                               BIN00500
      TPR=DELP                                                          BIN00510
      RAT=1.                                                            BIN00520
      TSI1=T0                                                           BIN00530
      TI1=T0                                                            BIN00540
      CSL1=C0                                                           BIN00550
```

```
      YI1=0.                                                                    BIN00560
      ZI1=0.                                                                    BIN00570
      II1=2                                                                     BIN00580
      II2=2                                                                     BIN00590
      II3=2                                                                     BIN00600
      D2T2=0.                                                                   BIN00610
      D2T3=0.                                                                   BIN00620
      D2T4=0.                                                                   BIN00630
      D2C3=0.                                                                   BIN00640
      D2C4=0.                                                                   BIN00650
      D2C2=0.                                                                   BIN00660
      D2T1=0.                                                                   BIN00670
      D2C1=0.                                                                   BIN00680
      DTLDX=0.                                                                  BIN00690
      DTSDX=0.                                                                  BIN00700
      D2C5=0.                                                                   BIN00710
      D2T5=0.                                                                   BIN00720
      D2C6=0.                                                                   BIN00730
      D2T6=0.                                                                   BIN00740
      TIME =0.                                                                  BIN00750
      DCDX0=0.                                                                  BIN00760
      DTDX0=0.                                                                  BIN00770
      DELTS=DELT/8.                                                             BIN00780
        TIME1=TIME+DELTS                                                        BIN00790
      DO 10 I=  1,IM                                                            BIN00800
      C(I)=C0                                                                   BIN00810
      CC(I)=C0                                                                  BIN00820
      TT(I)=T0                                                                  BIN00830
   10 T(I)=T0                                                                   BIN00840
      IFL=0                                                                     BIN00850
      IFS=0                                                                     BIN00860
      III=II2                                                                   BIN00870
   14 IT=0                                                                      BIN00880
      IF(IFS.EQ.1)GO TO 199                                                     BIN00890
      IF(IFL) 11,11,20                                                          BIN00900
   11 U0=UE(T0)                                                                 BIN00910
      V0=VE(T0)                                                                 BIN00920
      IFI=1                                                                     BIN00930
  199 IF(IFS.NE.0)CSL1=CS1                                                      BIN00940
      IF(IFS.EQ.1) IFS=2                                                        BIN00950
      DYDT0=U0*EMU*CSL1/RHOU+V0*EMV*(1.-CSL1)/RHO                               BIN00960
       HB0=-EE*SIG*(273.12+TSI1)**4-U0*GAMMAU*CSL1-V0*GAMMA*(1.-CSL1)           BIN00970
      DCDT0=DCDX0*DYDT0-(U0-V0)*CSL1                                            BIN00980
      DTDT0=DTDX0*DYDT0+HB0                                                     BIN00990
   20 YI2=YI1+DELTS*DYDT0                                                       BIN01000
      IF(IFS.EQ.0) ZI2=YI2                                                      BIN01010
      IF(IFS.NE.0) ZI2=ZI1+DELTS*DZDT0                                          BIN01020
      IF(ZI2.GT.X(II2+1)) ZI2=(X(II2)+YI2)/2.                                   BIN01030
      TSI2=TSI1+DELTS*DTDT0                                                     BIN01040
      TI2=TSI2                                                                  BIN01050
      CSL2=CSL1+DELTS*DCDT0                                                     BIN01060
      IF(IFS.EQ.0)CL2=CSL2                                                      BIN01070
   77 IF(IFS.EQ.0)GO TO 777                                                     BIN01080
      IF (IFL.GT.1)GO TO 877                                                    BIN01090
      III=0                                                                     BIN01100
```

```
  877 CS2=FS(TI2)                                                        BIN01110
      CL2=FL(TI2)                                                        BIN01120
  777 D2C2=D2(ZI2,CL2,X(II2),CC(II2),X(II2+1),CC(II2+1))                 BIN01130
      CC(II2)=(CC(II2)+C(II2)+.5*DELTS*(D2C1+D2C2)*DL      )/2.          BIN01140
       CALL MOTON(X(II2),CC(II2),ZI2,CL2,X(II2+1),CC(II2+1))             BIN01150
      IF(ZI2.LT.X(II2-1))CC(II2-1)=CC(II2)+(X(II2-1)-X(II2))*            BIN01160
     *(CL2-CC(II2))/(ZI2-X(II2))                                         BIN01170
      IF(II1-II3-1)83,87,84                                              BIN01180
   87 XP=YI2                                                             BIN01190
      CP=CSL2                                                            BIN01200
      GO TO 184                                                          BIN01210
   84 XP=X(II1-2)                                                        BIN01220
      CP=CC(II1-2)                                                       BIN01230
  184 D2C4=D2(XP,CP,X(II1-1),CC(II1-1),ZI2,CS2)                          BIN01240
      CC(II1-1)=C(II1-1)+.5*DELTS*(D2C3+D2C4)*DS                         BIN01250
      CALL MOTON(X(II1-1),CC(II1-1),XP,CP,ZI2,CS2)                       BIN01260
   83 IF(II2.EQ.II1.OR.ZI2.LT.X(II2-1)) GO TO 85                         BIN01270
      IF(II2-II3.GT.1)GO TO 185                                          BIN01280
      XP=YI2                                                             BIN01290
      CP=CSL2                                                            BIN01300
      GO TO 51                                                           BIN01310
  185 XP=X(II2-2)                                                        BIN01320
      CP=CC(II2-2)                                                       BIN01330
   51 CC(II2-1)=CP+(X(II2-1)-XP)*(CS2       -CP)/                        BIN01340
     *(ZI2      -XP)                                                    BIN01350
   85 D2T2=D2(ZI2,TI2,X(II2),TT(II2),X(II2+1),TT(II2+1))                 BIN01360
      IF (D2T2.GT.0.)D2T2=0.                                             BIN01370
      TT(II2)=(TT(II2)+T(II2)+.5*DELTS*(D2T1+D2T2)*ALS      )/2.         BIN01380
      CALL MOTON(X(II2),TT(II2),ZI2,TI2,X(II2+1),TT(II2+1))              BIN01390
      IF (IFS.EQ.0)GO TO 485                                             BIN01400
  485 IF(ZI2.LT.X(II2-1))TT(II2-1)=TT(II2)+(X(II2-1)-X(II2))*            BIN01410
     *(TI2-TT(II2))/(ZI2-X(II2))                                         BIN01420
      IF(II1-II3-1) 69,169,269                                           BIN01430
  169 TP=TSI2                                                            BIN01440
      XP=YI2                                                             BIN01450
      GO TO 16                                                           BIN01460
  269 TP=TT(II1-2)                                                       BIN01470
      XP=X(II1-2)                                                        BIN01480
   16 D2T4=D2(XP,TP,X(II1-1),TT(II1-1),ZI2,TI2)                          BIN01490
      TT(II1-1)= T(II1-1)+.5*DELTS*(D2T3+D2T4) *ASS                      BIN01500
      CALL MOTON(X(II1-1),TT(II1-1),XP,TP,ZI2,TI2)                       BIN01510
   69 IF(II2.EQ.II1.OR.ZI2.LT.X(II2-1)) GO TO 86                         BIN01520
   52 IF(II2.LT.II3-1)GO TO 186                                          BIN01530
      XP=YI2                                                             BIN01540
      TP=TSI2                                                            BIN01550
      GO TO 352                                                          BIN01560
  186 XP=X(II2-2)                                                        BIN01570
      TP=TT(II2-2)                                                       BIN01580
  352 TT(II2-1)=TP +(X(II2-1)-XP)*(TI2       -TP)/                       BIN01590
     *(ZI2      -XP)                                                    BIN01600
   86 IF(IFS.EQ.0)GO TO 299                                              BIN01610
      DCLDX=(CL2-CC(II2))/(ZI2-X(II2))                                   BIN01620
      IF (D2C2.GT.0.)  DCLDX=DCLDX-D2C2*(X(II2)-ZI2)/2.                  BIN01630
      IF (D2C2.LT.0.)D2C2=0.                                             BIN01640
      DTLDX=(TI2-TT(II2))/(ZI2-X(II2))                                   BIN01650
```

```
      IF (D2T2.LT.0.)DTLDX=DTLDX-.5*D2T2*(X(II2)-ZI2)                    BIN01660
      IF(II3.EQ.II1)GO TO 386                                            BIN01670
      XP=X(II1-1)                                                        BIN01680
      TP=TT(II1-1)                                                       BIN01690
      CP=CC(II1-1)                                                       BIN01700
      GO TO 486                                                          BIN01710
  386 XP=YI2                                                             BIN01720
      TP=TSI2                                                            BIN01730
      CP=CSL2                                                            BIN01740
      DCSDX=(CS2-CP)/(ZI2-XP)                                            BIN01750
      DTSDX=(TI2-TP)/(ZI2-XP)                                            BIN01760
      GO TO 686                                                          BIN01770
  486 DCSDX=(CP-CS2)/(XP-ZI2)   -D2C4*(XP-ZI2)/2.                        BIN01780
      DTSDX=(TP-TI2)/(XP-ZI2) -D2T4*(XP-ZI2)/2.                          BIN01790
  686 DZDT=(DL*DCLDX-DS*DCSDX)/(CS2-CL2)                                 BIN01800
       DZDTT=(XKS*DTSDX-XKL*DTLDX)/(RHO*GAMMA)                           BIN01810
      FSL=DZDT*(CS2-CL2)/DZDTT                                           BIN01820
      IF(FSL.GT.0..OR.FSL.LT.AQUAN )GO TO 772                            BIN01830
      TII=XCSL(FSL)                                                      BIN01840
      GO TO 771                                                          BIN01850
  772 TII=RT                                                             BIN01860
  771 COE1=XKS/(ZI2-XP)                                                  BIN01870
      COE2=XKL/(X(II2)-ZI2)                                              BIN01880
  586 TI=(RHO*GAMMA*DZDT+COE1*(TP-D2T4*.5*(XP-ZI2)**2)+COE2*(TT(II2)-.5  BIN01890
     **D2T2*(X(II2)-ZI2)**2))/(COE1+COE2)                                BIN01900
  773 IF (TI.LT.TSI2.AND.TI.LT.TII) TI=TII                               BIN01910
      IF (TI.LT.TSI2.OR.TI.GT.TT(II2+1))TI=TT(II2)                      BIN01920
      IF (DZDT.LT.0..AND.DZDTT.GT.0.)  DZDT=DZDTT                        BIN01930
      IF(ABS(TI-TI2)-1.E-5*ABS1(TI+TI2))587,587,770                      BIN01940
  587 IF(ABS(DZDT-DZDT1)-1.E-3*ABS1(DZDT+DZDT1))298,298,770              BIN01950
  770 TI2=(TI+TI2)/2.                                                    BIN01960
      IF (DZDT .LT.0.)DZDT=DZDT1                                         BIN01970
      DZDT1=(DZDT+DZDT1)/2.                                              BIN01980
      ZI2=ZI1+.5*DELTS*(DZDT1+DZDT0)                                     BIN01990
      IF(ZI2.GT.X(II2+1))ZI2=(X(II2)+YI2)/2.                             BIN02000
      IIT=IIT+1                                                          BIN02010
      IF(IIT.GT.5)  GO TO 298                                            BIN02020
      GO TO 877                                                          BIN02030
  298 IF(TI.GT.TT(II2))TI=TT(II2)                                        BIN02040
      IF(TI.LT.TSI2)TI=TSI2                                              BIN02050
      IF(II1-II3-1)398,498,598                                           BIN02060
  598 D2T6=D2(YI2,TSI2,X(II3),TT(II3),X(II3+1),TT(II3+1))                BIN02070
      D2C6=D2(YI2,CSL2,X(II3),CC(II3),X(II3+1),CC(II3+1))                BIN02080
      DTDX1=(TSI2-TT(II3))/(YI2-X(II3))-D2T6*(X(II3)-YI2)/2.             BIN02090
      DCDX1=(CSL2-CC(II3))/(YI2-X(II3))-D2C6*(X(II3)-YI2)/2.             BIN02100
      GO TO 599                                                          BIN02110
  498 DTDX1=DTSDX+D2T4*(XP-ZI2)                                          BIN02120
      DCDX1=DCSDX+D2C4*(XP-ZI2)                                          BIN02130
      D2T6=D2T4                                                          BIN02140
      D2C6=D2C4                                                          BIN02150
      GO TO 599                                                          BIN02160
  398 DCDX1=DCSDX                                                        BIN02170
      DTDX1=DTSDX                                                        BIN02180
      GO TO 599                                                          BIN02190
  299 DTDX1=(TT(II2)-TI2)/(X(II2)-YI2)-.5*D2T2*(X(II2)-YI2)              BIN02200
```

```
      DCDX1=(CC(II2)-CL2)/(X(II2)-YI2)-.5*D2C2*(X(II2)-YI2)                BIN02210
  599 U1=UE(TSI2)                                                          BIN02220
      V1=VE(TSI2)                                                          BIN02230
      HB1=-EE*SIG*(273.12+TSI2)**4-U1*GAMMAU*CSL2-V1*GAMMA*(1.-CSL2)       BIN02240
      DYDT1=U1*EMU*CSL2/RHOU+V1*EMV*(1.-CSL2)/RHO                          BIN02250
      DCDT1=DCDX1*DYDT1-(U1-V1)*CSL2                                       BIN02260
      DTDT1=DTDX1*DYDT1+HB1                                                BIN02270
      IF (IFL.LT.2) GO TO 76                                               BIN02280
      IF(IFS.GE.1)GO TO 174                                                BIN02290
      IF(TSI2.GT.TT(II2)) GO TO 399                                        BIN02300
      TT2=XCL(CSL2)                                                        BIN02310
      IF(TSI2.GT.TT2) GO TO 174                                            BIN02320
      IF(TSI2.GE.TT2-.05)GO TO 400                                         BIN02330
      IF (DELTS.LE.DELTK) GO TO 400                                        BIN02340
      GO TO 399                                                            BIN02350
   76 YY=YI1+.5*(DYDT0+DYDT1)*DELTS                                        BIN02360
      IF(IFS.EQ.0)ZI=YI                                                    BIN02370
      TSI =TSI1+.5*(DTDT0+DTDT1)*DELTS                                     BIN02380
      IF(TSI.LT.0.)TSI=TSI1                                                BIN02390
      CSL=CSL1+.5*(DCDT0+DCDT1)*DELTS                                      BIN02400
      IF (IFS.EQ.0) GO TO 73                                               BIN02410
      IF(IIT.GT.5)GO TO 70                                                 BIN02420
      IF(TSI.GT.TT(II2).OR.TSI.GT.TI2) GO TO 70                            BIN02430
   73 IF(ABS(YI-YI2)-1.E-6*ABS1(YI+YI2))74,74,70                           BIN02440
   74 IF(ABS(CSL-CSL2)-1.E-4*ABS1(CSL+CSL2))75,75,70                       BIN02450
   75 IF(ABS(TSI-TSI2)-1.E-5*ABS1(TSI+TSI2))7,7,70                         BIN02460
   70 TSI2=(TSI+TSI2)/2.                                                   BIN02470
      IF (IFS.EQ.0)TI2=TSI2                                                BIN02480
      YI2=(YI+YI2)/2.                                                      BIN02490
      IF (IFS.EQ.0)ZI2=YI2                                                 BIN02500
      CSL2=(CSL+CSL2)/2.                                                   BIN02510
      IF(IFS.EQ.0)CL2=CSL2                                                 BIN02520
      IF(ZI2.LT.(1.-S)*X(II2)+S*X(II2-1)) GO TO 24                         BIN02530
   22 IF(II1-II2) 24,96,96                                                 BIN02540
   96 II2=II2+1                                                            BIN02550
   46 D2C1=D2(ZI1,CL1,X(II2),C(II2),X(II2+1),C(II2+1))                     BIN02560
      D2T1=D2(ZI1,TI1,X(II2),T(II2),X(II2+1),T(II2+1))                     BIN02570
   24 IF (IT-NITQ) 48,174,160                                              BIN02580
  160 IF(IT-NITH) 48,47,48                                                 BIN02590
   47 IF(DELTS-DELTK) 48,48,53                                             BIN02600
   53 DELTS=DELTS/RAT                                                      BIN02610
  152 IF(RAT-1.) 153,153,154                                               BIN02620
  153 RAT=2.*RAT                                                           BIN02630
       DELTS=DELTS/2.                                                      BIN02640
       GO TO 152                                                           BIN02650
  154 TIME1=TIME+DELTS                                                     BIN02660
  212 DO 45 I=II3,IM                                                       BIN02670
      TT(I)=(TT(I)+(RAT-1.)*T(I))/RAT                                      BIN02680
   45 CC(I)=(CC(I)+(RAT-1.)*C(I))/RAT                                      BIN02690
  148 RAT=1.                                                               BIN02700
   48 IT=IT+1                                                              BIN02710
      IF(IT.EQ.NITH+1)GO TO 20                                             BIN02720
      IF (IT-NITL) 161,174,161                                             BIN02730
  161 IF(IT-NIT) 77,77,402                                                 BIN02740
  402 IF(DELTS.GT.DELTK) GO TO 399                                         BIN02750
```

```
      IFL=2                                                              BIN02760
      GO TO 77                                                           BIN02770
   26 IF (NONCN) 99,48,48                                                BIN02780
    7 IFL=IFL+1                                                          BIN02790
      DZDT1=DZDT                                                         BIN02800
      IF(IFS.NE.0) ZI2=ZI1+.5*DELTS*(DZDT0+DZDT1)                        BIN02810
      TSI2=TSI                                                           BIN02820
      YI2=YI                                                             BIN02830
      IF(IFS.EQ.0)ZI2=YI2                                                BIN02840
      CSL2=CSL                                                           BIN02850
      IF (IFS.NE.0) TI2=TI                                               BIN02860
      IF (TI2.LT.TSI2)TI2=TSI2                                           BIN02870
      IF (IFS.EQ.0) TI2=TSI2                                             BIN02880
      IF(IFS.EQ.0)CL2=CSL2                                               BIN02890
      GO TO 77                                                           BIN02900
  399 IF(DELTS.LE.DELTK) GO TO 26                                        BIN02910
      IT=NITH                                                            BIN02920
      IFL=1                                                              BIN02930
      GO TO 153                                                          BIN02940
  117 IF(TIME.EQ.TIME1)TIME1=TIME1    +DELTS                             BIN02950
      TIME=TIME1                                                         BIN02960
       IF (RAT.NE.2..AND.RAT.NE.0.) DELTS=DELTS/RAT                      BIN02970
      RAT=1.                                                             BIN02980
      IF(IT-NITQ) 82,82,81                                               BIN02990
   82 IF(DELTS.GT.DELT/2.) GO TO 81                                      BIN03000
      DELTS=DELTS+DELTS                                                  BIN03010
      RAT=2.                                                             BIN03020
   81 TIME1=TIME1+DELTS                                                  BIN03030
  282 D2C1=D2C2                                                          BIN03040
      D2C3=D2C4                                                          BIN03050
      D2C5=D2C6                                                          BIN03060
      D2T3=D2T4                                                          BIN03070
      D2T1=D2T2                                                          BIN03080
      D2T5=D2T6                                                          BIN03090
      DYDT0=DYDT1                                                        BIN03100
      DCDT0=DCDT1                                                        BIN03110
      U0=U1                                                              BIN03120
      V0=V1                                                              BIN03130
      DTSDX1=DTSDX                                                       BIN03140
      DCSDX1=DCSDX                                                       BIN03150
      IF(IFS.NE.0) DCDX0=DCDX1                                           BIN03160
      IF(IFS.NE.0) DTDX0=DTDX1                                           BIN03170
      IF(IFS.NE.0) DZDT0=DZDT1                                           BIN03180
      CSL1=CSL2                                                          BIN03190
      TSI1=TSI2                                                          BIN03200
      YI1=YI2                                                            BIN03210
      ZI1=ZI2                                                            BIN03220
      IF(IFS.NE.0)CS1=CS2                                                BIN03230
      HB0=HB1                                                            BIN03240
      CL1=CL2                                                            BIN03250
      TI1=TI2                                                            BIN03260
      IF(YI2.LT.X(II3))GO TO 410                                         BIN03270
      II3=II3+1                                                          BIN03280
      D2T5=D2(YI1,TSI1,X(II3),TT(II3),X(II3+1),TT(II3+1))                BIN03290
      D2C5=D2(YI1,CSL1,X(II3),CC(II3),X(II3+1),CC(II3+1))                BIN03300
```

```
 410 IF(II1.EQ.II2.OR.ZI2.LT.S*X(II2)+(1.-S)*X(II2-1))GO TO 33          BIN03310
     II1=II2                                                           BIN03320
     IF(II1-II3-1)33,310, 110                                          BIN03330
 310 XP=YI1                                                            BIN03340
     CP=CSL1                                                           BIN03350
     TP=TSI1                                                           BIN03360
     GO TO 210                                                         BIN03370
 110 XP=X(II1-2)                                                       BIN03380
     CP=CC(II1-2)                                                      BIN03390
     TP=TT(II1-2)                                                      BIN03400
 210 D2C3=D2(XP,CP,X(II1-1),CC(II1-1),ZI1,CL1)                         BIN03410
     D2T3=D2(XP,TP,X(II1-1),TT(II1-1),ZI1,TI1)                         BIN03420
     IF(II1-II3.NE.1)GO TO 33                                          BIN03430
     D2T5=D2T3                                                         BIN03440
     D2C5=D2C3                                                         BIN03450
     GO TO 33                                                          BIN03460
 174 III=II2                                                           BIN03470
     IF(ZI2.LT.X(II2-1))III=III-1                                      BIN03480
     IF (II1-II3.LT.2) GO TO 18                                        BIN03490
     IF(II1-II3.GT.2)GO TO 29                                          BIN03500
     TT(II3)=(T(II3)/DELTS+.5*ASS*D2T5                                 BIN03510
    *+TT(II3+1)*ASS/(X(II3+1)-X(II3))/(X(II3+1)-YI2)+TSI2*ASS/(X       BIN03520
    *(II3)-YI2)/(X(II3+1)-YI2))/(1./DELTS+ASS/(X(II3+1)-YI2)*(1./(X(II3BIN03530
    *+1)-X(II3))+1./(X(II3)-YI2)))                                     BIN03540
     CC(II3)=(C(II3)/DELTS+.5*DS*D2C5                                  BIN03550
    *+CC(II3+1)*DS/(X(II3+1)-X(II3))/(X(II3+1)-YI2)+CSL2*DS/(X(I       BIN03560
    *I3)-YI2)/(X(II3+1)-YI2))/(1./DELTS+DS/(X(II3+1)-YI2)*(1./(X(II3+1)BIN03570
    *-X(II3))+1./(X(II3)-YI2)))                                        BIN03580
     GO TO 18                                                          BIN03590
  29 X(II3-1)=(YI2+YI1)/2.                                             BIN03600
     C(II3-1)=CSL1                                                     BIN03610
     CC(II3-1)=CSL2                                                    BIN03620
     T(II3-1)=TSI1                                                     BIN03630
     TT(II3-1)=TSI2                                                    BIN03640
     CALL TRIST(X,T,TT,II3-1,III-1,ASS,DELTS,A)                        BIN03650
     CALL TRIST (X,C,CC,II3-1,III-1,DS,DELTS,A)                        BIN03660
     GO TO 18                                                          BIN03670
  27 IF(T(IM-1)-T0+1.E0)    30,32,32                                   BIN03680
  30 IF (IM-IX) 15,32,32                                               BIN03690
  15 IM=IM+1                                                           BIN03700
     T(IM)=T0                                                          BIN03710
     C(IM)=C0                                                          BIN03720
  32 TT(IM)=T(IM)                                                      BIN03730
     CC(IM)=C(IM)                                                      BIN03740
     GO TO 14                                                          BIN03750
  18 IF(DELTS.LT.DELTK)DELTS=DELTK                                     BIN03760
     CALL TRIST(X,T,TT,III,IM,ALS,DELTS,A)                             BIN03770
     CALL TRIST(X,C,CC,III,IM,  DL,DELTS,A)                            BIN03780
      DO 21 I=II2,IM                                                   BIN03790
      IF (I.EQ.II2) GO TO 21                                           BIN03800
  21 CONTINUE                                                          BIN03810
 219 IF (IFL.EQ.2) GO TO 117                                           BIN03820
     GO TO 48                                                          BIN03830
  33 IF(TIME -TPR) 50,34,34                                            BIN03840
  50 IF (NONCN) 98,98,54                                               BIN03850
```

```
   54 IF(IT-NIT) 98,98,42                                                                BIN03860
   34 IF(DELP.EQ.0.)GO TO 42                                                             BIN03870
      TPR=TPR+DELP                                                                       BIN03880
   42 WRITE(IO,100) IT,IM ,II1,II2 ,II3                                                  BIN03890
      WRITE(IO,102)TIME,YI1,CSL1,TSI1,U1,V1,HB1,DYDT1,DTDT1,DCDT1,DELTS                  BIN03900
      IF(IFS.NE.0)WRITE(IO,102)ZI1,CS1,CL1,TI1,DZDT1                                     BIN03910
  143 WRITE(IO,102) (TT(I),I=II3,IM),(CC(I),I=II3,IM)                                    BIN03920
  102 FORMAT(5E14.6)                                                                     BIN03930
      IF(TIME -TF) 98,99,99                                                              BIN03940
   98 DO 97 I=II3,IM                                                                     BIN03950
      TTT=TT(I)+RAT*(TT(I)-T(I))                                                         BIN03960
      T(I)=TT(I)                                                                         BIN03970
      TT(I)=TTT                                                                          BIN03980
   97 CONTINUE                                                                           BIN03990
       DO 197 I=II3,IM                                                                  BIN04000
      CCC=CC(I)+RAT*(CC(I)-C(I))                                                         BIN04010
      C(I)=CC(I)                                                                         BIN04020
  197 CC(I)=CCC                                                                          BIN04030
       IFL=1                                                                             BIN04040
       GO TO 27                                                                          BIN04050
  400 CSS=FS(TI2)                                                                        BIN04060
      CLL=FL(TI2)                                                                        BIN04070
      DELZ=(CLL-CL2)/((CLL+CC(II2))-2.*CSS)*(X(II2)-YI2)                                 BIN04080
      ZI2=YI2+DELZ                                                                       BIN04090
      DZDT0=DELZ/DELTS                                                                   BIN04100
      DZDT1=DZDT0                                                                        BIN04110
      CL2=CLL                                                                            BIN04120
      CS2=CSS                                                                            BIN04130
      DCDX0=0.                                                                           BIN04140
      DTDX0=0.                                                                           BIN04150
      DTDX1=0.                                                                           BIN04160
      WRITE(IO,103)TIME1,YI2,TI2,ZI2,CS2,CL2,DELZ,DELTS,DZDT0                            BIN04170
  103 FORMAT('  SOLIDIFICATION HAS BEGUN '/(5E14.6))                                    BIN04180
      IFS=1                                                                              BIN04190
      GO TO 174                                                                          BIN04200
   99 READ(II,INVAR)                                                                     BIN04210
      IF(IX.GT.0)GO TO 999                                                               BIN04220
      STOP                                                                               BIN04230
      END                                                                                BIN04240
```

APPENDIX II

GLOSSARY OF PROGRAM PARAMETERS

IX = maximum number of points in mesh, $\leq$ 28

IAM = spacing option: 0 indicates equal, 1 unequal doubling

NIT = maximum iteration count

IM = number of points in mesh

NONCN = nonconvergence option: 1 indicates proceed and printout
0 indicates proceed but do not printout
-1 indicates program stop

IT = actual iteration count

NITH = half of NIT

NITQ = quarter of NIT

NITL = 3/4 NIT

IFL = indicator of convergence: 2 on convergence,
< 2 before convergence

IFS = indicator of beginning of solidification: IFS = 0 before solidification,
IFS > 0 after solidification

II1 = grid point reference for solid side of mesh

II2 = grid point reference for liquid side of mesh

II3 = grid point reference for evaporation boundary

II4 = grid point reference for point after evaporation boundary

III = grid point reference for point after solidification boundary

DELT = maximum time interval (step size)

DELTM = minimum time interval

DELTK = larger of quantities DELT/1024 and DELTM

DELTS = current time interval

TIME = time at beginning of time interval

TIME1 = time at end of time interval

DELP = time print interval

TPR = time for printing results

TF = final time

YI, YI1, YI2 = values of y (evaporation boundary)

ZI1, ZI2 = values of z (solid-liquid boundary)

TI, TII, TI1, TI2 = temperatures at solid-liquid boundary

TSI, TSI1, TSI2 = temperatures at evaporation boundary

CSL, CSL1, CSL2 = concentration at evaporation boundary

CS1, CS2 = concentration of solid at solid-liquid boundary

CL1, CL2 = concentration of liquid at solid-liquid boundary

DZDT, DZDTT, DZDT0, DZDT1 = $\frac{dz}{dt}$ derivative of solid-liquid boundary

DYDT0, DYOT1 = $\frac{dy}{dt}$ derivative of evaporation boundary

DTDT0, DTDT1 = $\frac{dT}{dt}$ derivative of temperature at evaporation boundary

DCDT0, DCDT1 = $\frac{dC}{dT}$ derivative of concentration at evaporation boundary

DTDX0, DTDX1 = $\left(\frac{\partial T}{\partial x}\right)_y$ partial derivative for temperature at evaporation boundary

DTSDX = $\left(\frac{\partial T_s}{\partial x}\right)_z$ partial derivative of temperature in solid at boundary

DTLDX = $\left(\frac{\partial T_L}{\partial x}\right)_z$ partial derivative of temperature in liquid at boundary

DCDX0, DCDX1 = $\left(\frac{\partial C}{\partial x}\right)_y$ partial derivative of concentration at evaporation boundary

DCSDX = $\left(\frac{\partial C_s}{\partial x}\right)_z$ partial derivative of concentration in solid at solid-liquid boundary

DCLDX = $\left(\frac{\partial C_L}{\partial x}\right)_z$ partial derivative of concentration in liquid at solid-liquid boundary

D2T1, D2T2 = $\left(\frac{\partial^2 T_2}{\partial x^2}\right)_z$ at liquid side of solid-liquid boundary

D2T3, D2T4 = $\left(\frac{\partial^2 T_s}{\partial x^2}\right)_z$ at solid side of solid-liquid boundary

D2T5, D∂T6 = $\left(\frac{\partial^2 T}{\partial x^2}\right)_y$ at evaporation boundary

D2C1, D2C2 = $\left(\frac{\partial^2 C_2}{\partial x^2}\right)_z$ at liquid side of solid-liquid boundary

D2C3, D2C4 = $\left(\frac{\partial^2 C_s}{\partial x^2}\right)_z$ at liquid side of solid-liquid boundary

D2C3, D2C4 = $\left(\frac{\partial^2 C_s}{\partial x^2}\right)_z$ at solid side of solid-liquid boundary

D2C5, D2C6 = $\left(\frac{\partial^2 C}{\partial x^2}\right)_y$ at evaporation boundary

UO, U1 = rates of evaporation of solute

VO, V1 = rates of evaporation of solvent

HBO, HB1 = heat balance sum of evaporation and radiation terms

EMU, EMV = molecular weight of solute and solvent atoms

RHOU, RHO = density of solute and solvent

GAMMAU, GAMMA = specific heats of solute and solvent

AU, BU, AV, BV, EK = evaporation constants for solute and solvent

UE, VE = arithmetic function definition for evaporation rates

ET, EA, EB, EC, ED = phase diagram constants

FS, FL = arithmetic functions for solidus and liquidus curves

AS, AL	= temperature diffusion coefficients
ASS, ALS	= squares of temperature diffusion coefficients
XKS, XKL	= k_s, k_ℓ for interphase boundary equation
DS, DL	= mass (concentration) diffusion coefficient
EE, SIG	= radiation constants ϵ, σ
TO, CO	= initial temperature and concentration distribution
COO	= equals CO

APPENDIX III

LIST OF SYMBOLS

A_ℓ	$k_\ell/\rho_v c$
A_s	$k_s/\rho_v c$
A_u, B_u	evaporating constants for solute
A_v, B_v	evaporating constants for solvent
c	specific heat, cal/g/°C
C	solute concentration
C_ℓ	solute concentration in liquid
C_o	initial solute concentration in liquid
C_s	solute concentration in solid
D_s	density of solid, g/cm^3
D_ℓ	density of liquid, g/cm^3
E_T	melting point of solvent, °C
E_A, E_B	coefficients of solidus
E_C, E_D	coefficients of liquidus
f_ℓ	liquidus equation
f_s	solidus equation
K_e	constant in evaporation equations = 5.83×10^{-5}
k_ℓ	thermal conductivity of liquid, cal/sq cm/cm/sec/°C
k_s	thermal conductivity of solid, cal/sq cm/cm/sec/°C
M_u	molecular weight of solute
M_v	molecular weight of solvent

S	time constant
t	time, second
T	temperature, °C
T_i	temperature at solid-liquid interface, °C
T_ℓ	temperature in liquid, °C
T_o	initial melt temperature, °C
T_s	temperature in solid, or surface temperature, °C
T_1	surface temperature, °C
T_2	final surface temperature, °C
U	solute evaporating rate, $mol/cm^2/sec$
V	solvent evaporating rate, $mol/cm^2/sec$
v	position and temperature-dependent variable
x	distance from initial melt surface, cm
y	distance at phase change boundary, cm
$\dot{y}$	rate of movement of phase change boundary, cm/sec
z	distance at phase change boundary, cm
α	constant
ϵ	emissivity coefficient
ρ	density, g/cm^3
ρ_u	density of solute, g/cm^3
ρ_v	density of solvent, g/cm^3
γ	latent heat of fusion, cal/g
γ_u	latent heat of fusion, or specific heat, of solute, cal/g or cal/g/°C
γ_v	latent heat of fusion, or specific heat, of solvent, cal/g or cal/g/°C
σ	Stefan-Boltzmann constant = 1.35×10^{-6}

REFERENCES

1. Li, C. H., Phys. Stat. Solidi 15, 3 and 419, 1966.

2. Li, C. H., "Review of Grumman Studies in Metal Solidification," Grumman Research Memorandum RM-525, 1971.

3. Li, C. H., Final Report, NASA Contract NAS 8-27891, Grumman Research Report RE-458, June 1972.

4. Mukerjee, J. L., Gupta, K. P., and Li, C. H., "Evaporative Segregation in 80%-20% Cr and 60%-40% Ni Alloys," Grumman Research Memorandum RM-552, October 1972; J. Vac. Sci. 11, 33, 1974

5. Mukerjee, J. L., Gupta, K. P., and Li, C. H., "Purification Kinetics of Beryllium during Vacuum Induction Melting," Grumman Research Memorandum RM-553, 1971.

6. Zinsmeister, G., Vakuum-Tech., No. 8, 223, 1964.

7. Li, C. H., "Evaporation in Space Processing," AIAA meeting, Boston, July 1974.

8. Reichman, J., "Solidification of Metal Spheres in Vacuum," Grumman Research Memorandum RM-544, June 1972.

9. Chalmers, B., Solidification, Wiley, New York, 1969.

10. Tiller, W., "Solidification" in Physical Metallurgy, Ed. R. W. Cahn, Wiley, New York, 1965, pp. 385-441.

11. Christian, J. W., The Theory of Transformations in Metals and Alloys, Pergamon Press, New York, 1965, pp. 527-593.

12. Li, C. H., "Solidification," to be published in Materials Science Series, Academic Press, New York, 1975.

13. Rubenstein, L. I., The Stefan Problem, Amer. Math. Soc., Providence, Rhode Island, 1971.

14. Li, C. H., Progress Report No. 2 under NASA Contract NAS 8-27891, March 1972.

15. Dushman, S., Scientific Foundations of Vacuum Techniques, Wiley, New York, 1962.