Institute for Space Nuclear Power Studies
Department of Chemical and Nuclear Engineering
The University of New Mexico
Albuquerque, New Mexico 87131

AN INVESTIGATION OF VOIDS FORMATION MECHANISMS AND THEIR EFFECTS ON FREEZE AND THAW PROCESSES OF LITHIUM AND LITHIUM FLUORIDE

MOHAMED S. EL-GENK AND JAE-YOUNG YANG

FINAL REPORT NO. UNM-ISNPS-NAG 3-1045

NASA Grant No. NAG 3-1045
Performance Period May 1989 - January 1991

August 1991

Institute for Space Nuclear Power Studies
Department of Chemical and Nuclear Engineering
The University of New Mexico
Albuquerque, New Mexico 87131

AN INVESTIGATION OF VOIDS FORMATION MECHANISMS AND THEIR EFFECTS ON FREEZE AND THAW PROCESSES OF LITHIUM AND LITHIUM FLUORIDE

MOHAMED S. EL-GENK AND JAE-YOUNG YANG

FINAL REPORT NO. UNM-ISNPS-NAG 3-1045

NASA Grant No. NAG 3-1045
Performance Period May 1989 - January 1991

August 1991

TABLE OF CONTENTS

APPENDICES

AN INVESTIGATION OF VOIDS FORMATION MECHANISMS AND THEIR EFFECTS ON FREEZE AND THAW PROCESS OF LITHIUM AND LITHIUM FLUORIDE

ABSTRACT

In the first part of the research, the mechanisms of void formation during the cooldown and freezing of lithium coolant within the primary loop of SP-100 type systems are investigated. These mechanisms are: (a) homogeneous nucleation, (b) heterogeneous nucleation, (c) normal segregation of helium gas dissolved in liquid lithium, and (d) shrinkage of lithium during freezing. To evaluate the void formation potential due to segregation, a numerical scheme that couples the freezing and mass diffusion processes in both the solid and liquid regions is developed. The results indicated that the formation of He bubbles is unlikely by either homogeneous or heterogeneous nucleation during the cooldown process. However, homogeneous nucleation of He bubbles following the segregation of dissolved He in liquid Lithium ahead of the solid-liquid interface is likely to occur. Results also show that total volume of He void is insignificant when compared to that of shrinkage voids.

In viewing this, the subsequent research focuses on the effects of shrinkage void forming during freezing of lithium on subsequent thaw processes are investigated using a numerical scheme that is based on a single (solid/liquid) cell approach. The cases of lithium-fluoride are also investigated to show the effect of larger volume shrinkage upon freezing on the freeze and thaw processes. Results show that a void forming at the wall appreciably reduces the solid-liquid interface velocity, during both freeze and thaw, and causes a substantial rise in the wall temperature during thaw. However, in the case of Li, the maximum wall temperature was much lower than the melting temperature of PWC-

11, which is used as the structure material in the SP-100 system. Hence, it is concluded that a formation of hot spots is unlikely during the startup or restart of the SP-100 system.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background

The SP-100 Space Nuclear Power System, which is currently in the engineering development and testing phase, incorporates a fast flux, nuclear reactor which is cooled by a pumped liquid lithium (Li). The Li in the Primary Heat Transport System (PHTS), transports the reactor's thermal power to the Power Conversion Assemblies (PCAs), where it is partially converted into electricity. The residual heat is transported by liquid Li in the secondary loops from the Thermoelectric-Electromagnetic (TEM) pumps and the PCAs to the system's heat pipe radiator.

The operating temperatures of the primary and secondary Li coolants at the reactor nominal thermal power of 2.4 MW, are about 1350 K to 1375 K, and 800 K to 850 K, respectively. Because Li has a high melting point of ~454 K, it will be frozen during launch and upon thawing and heating up to a nominal operating temperature of 1350 K, Li will undergo about 20-25% increase in volume (see Figure 1). To minimize the stresses in the walls of the reactor containment and/or the piping system during startup in orbit, voids are introduced throughout the PHTS and each primary and secondary loop is equipped with a gas separator/accumulator and an accumulator with gas loaded bellows, respectively.

Following a planned or an accidental shutdown, liquid Li in the primary and secondary coolant loops of the SP-100 system will eventually freeze. In the current SP-100 system design, the freezing process is controlled only by the natural law of cooling. The increase in the Li density during cooldown and freezing will stimulate formation of voids. Depending on the distribution of these voids, many different situations will arise during the restart period of the SP-100 system. Homogeneously distributed voids within the solidified Li coolant, for example, would absorb the stress due to the volume increase of Li during the thaw process effectively. In this case, the stress in the walls of the coolant pipes, fuel pins and core container may not exceed those that cause deformation or failure of the materials. Ill posed voids, however, may lead to (a) mechanical failure of the coolant loop pipes and fuel elements due to excessive stresses, (b) thermal failure of structure or formation of hot spots during thaw. It is, therefore, important to understand their formation mechanisms and their effect on the system safety during rethaw.

1.2 Void Formation Mechanisms

In metal casting, volume shrinkage during freezing and bubble nucleation of dissolved gas atoms in the melt cause the formation of voids and results in a porous or leaky structure [Brick et al. (1977), Cahn & Haasen (1983)].

In the SP-100 system, helium (He) and tritium gas atoms are produced by neutron-lithium (n-Li) interactions in the reactor core. While tritium gas produced in the reactor core will readily diffuse through the niobium (Nb) walls to outer space, most of the He atoms will remain in the PHTS due to the very small diffusivity of He. McGhee, El-Genk and Rothrock (1989) have shown that Li coolant in the PHTS will be saturated with He gas within several days of operation and that the gas separators will only remove the amount of He gas in excess of the saturation concentration. It is, therefore, expected that He gas bubble might form, as in the metal casting process, during the cooldown and freeze processes of the Li coolant.

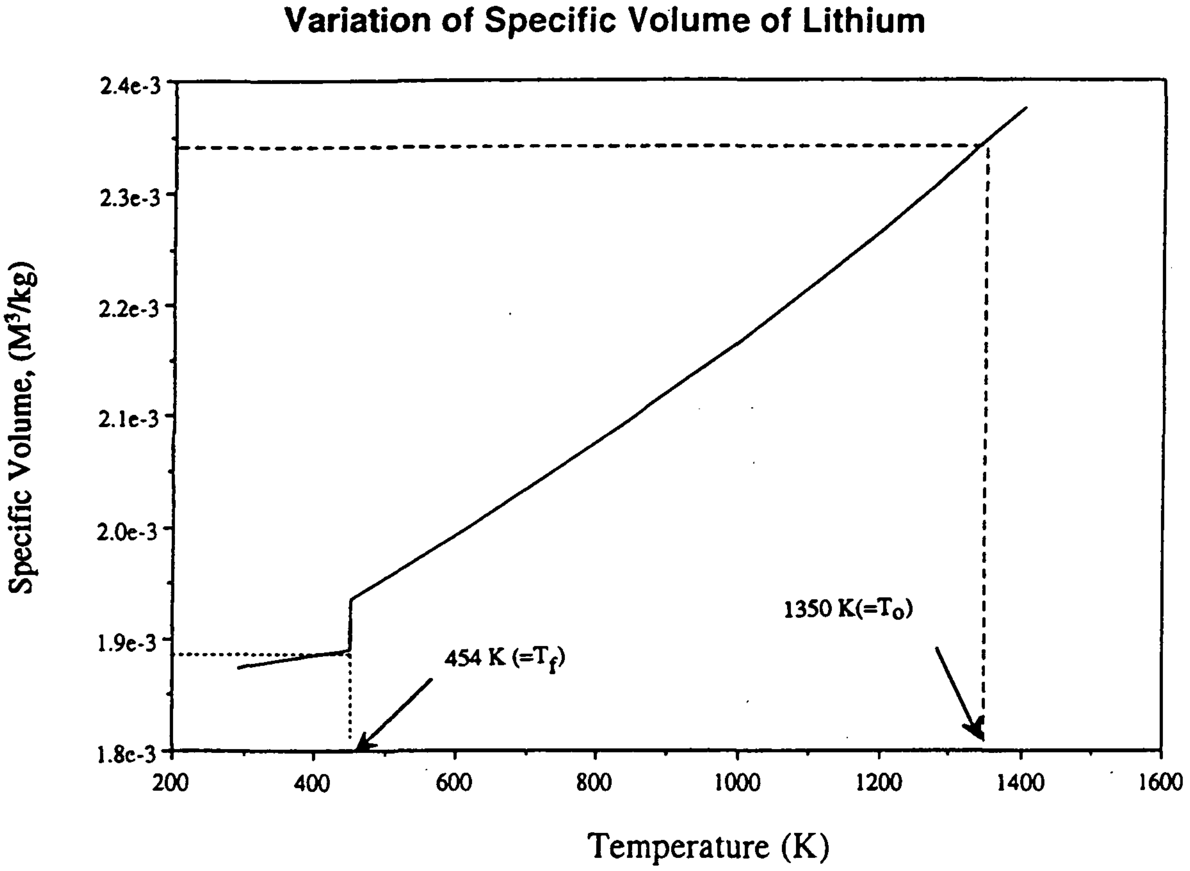


Figure 1. Variation of Specific Volume of Lithium

Two well known mechanisms of gas bubble formation are: (1) homogeneous nucleation (HN), and (2) heterogeneous nucleation (HTN). HN refers to the appearance of a gas nucleus in the bulk of liquid, away from container walls or suspended particles, while HTN refers to the formation of a gas nucleus on surfaces or suspended impurities. The gas concentration required for HN of bubbles could be orders of magnitude higher than the saturation concentration in Li. On the other hand, in a system rich in suspended impurities, the gas concentration required for bubble nucleation by HTN in liquid Li could be as low as the saturation concentration.

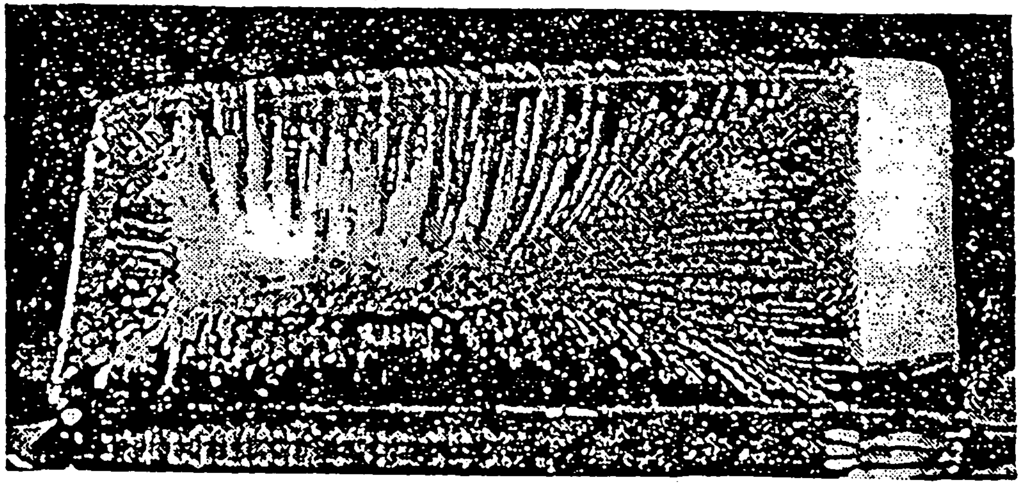
Another important mechanism of gas bubble formation during the freezing process is the segregation of dissolved gas atoms or molecules in a freezing medium. Segregation is a solute redistribution phenomenon during a freezing process. When the solubility limit of a solute in the solid phase is less than the solute concentration in the liquid phase, rejection of the solute by the S-L interface occurs as the freezing of the liquid progresses resulting in a buildup of the solute ahead of the S-L interface. Chalmers (1964) pointed out that in the case where gas is dissolved in a solidifying medium, segregation of gas in the liquid ahead of the solidification front can lead to the formation of gas voids. Figures 2a and 2b show gas void formation by segregation during the freezing processes of water and metal, respectively. In Chalmers (1964), seven different types of segregation are discussed including: (1) normal segregation, (2) grain boundary segregation, (3) cellular segregation, (4) dendritic segregation, (5) inverse segregation, (6) coring and intercrystalline segregation, and (7) gravity segregation. In microgravity, the last mechanism will not occur but the other six may occur, depending on the solidification process. The segregation of dissolved He atoms in liquid Li ahead of the solid-liquid (S-L) interface is expected to occur in the freezing process of Li coolant which may lead to the formation of gas bubbles during the process. The segregation mechanisms (2) - (6) require higher solidification velocities than that expected for normal segregation, type (1). Because of the high heat capacity and latent heat of fusion of Li and the low heat rejection rate, since radiation is the only mode of heat rejection in the current design of the SP-100 system, the cooldown and freezing processes are expected to be very slow, hence normal segregation will be the more likely mechanism for formation of He gas bubbles in the SP-100 system.

The formation of He gas bubbles by normal segregation will be accompanied by volume shrinkage of Li upon freezing stimulating the formation of shrinkage voids. These voids may occur either at the solid/wall interface or at the liquid/wall interface (see Figure 3). While the former will occur in the absence of adhesion between solidified Li and Nb wall, the latter is expected if liquid Li poorly wets Nb.

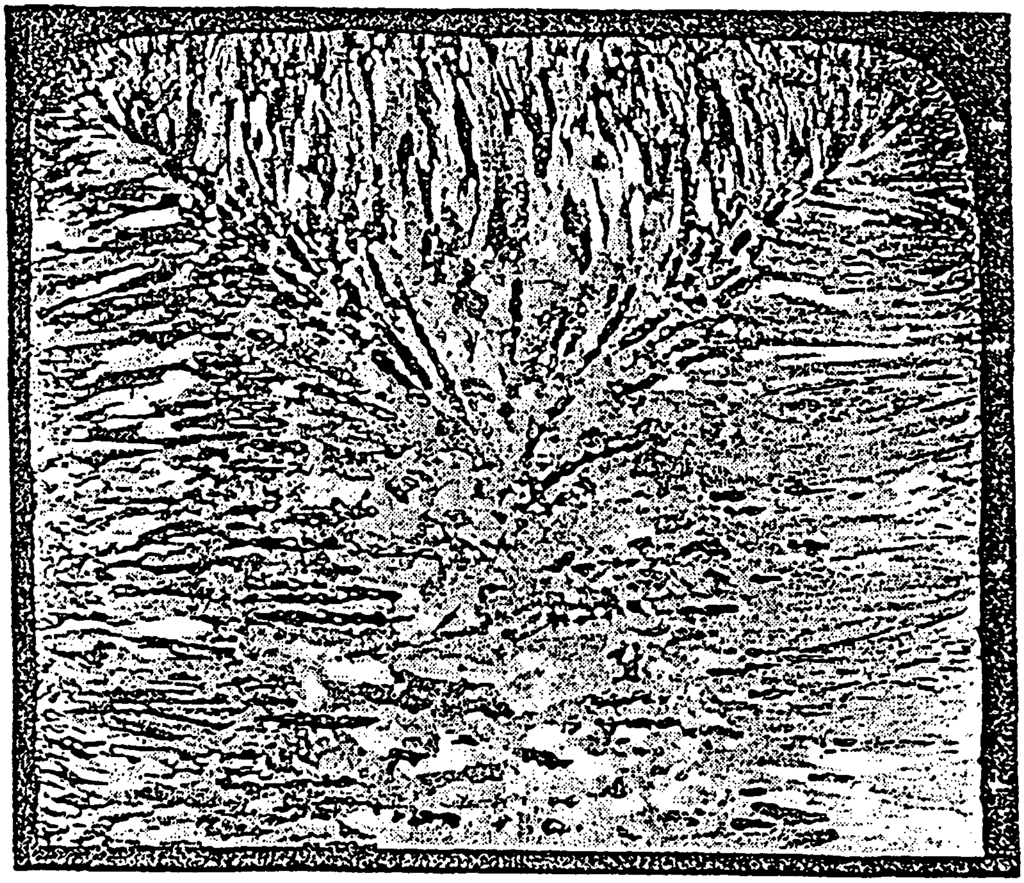
The potential for He gas bubble formation by HN, HTN, and normal segregation is investigated and the formation of voids due to volume shrinkage is quantified in Section 2. In Section 3, the research focuses on quantifying the thermal effects of shrinkage voids during the freeze and thaw processes.

1.3 Objectives

This research investigates: (1) the void formation mechanisms and the potential for forming voids during the cooldown and freezing processes of Li coolant in the SP-100 system after shutdown of the reactor in space, and (2) the thermal effect of void formation on the freeze and thaw processes of Li coolant. Also, investigated are the thermal effects of void formation on both freezing and subsequent thawing of LiF, which is being considered for thermal energy storage in space application by NASA. Results of Li and LiF are compared and discussed.

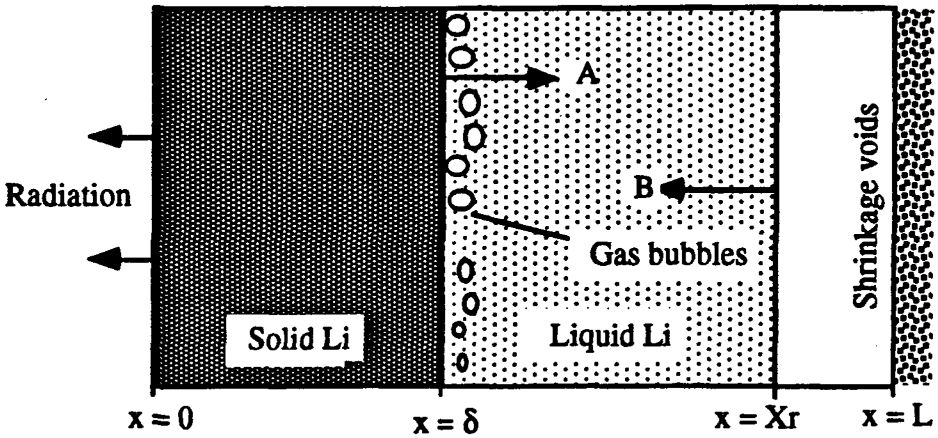


(a) Bubbles in ice cube .

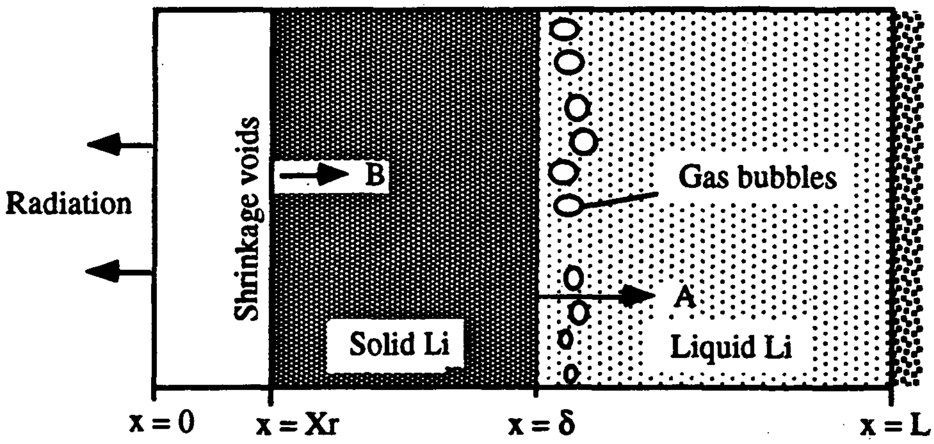


(b) Gas bubbles in a metal.

Figure 2. Gas Bubble Formation by Segregation during the Freezing Process of Water and Metal



(a) Shrinkage Void at Insulated Boundary



(b) Shrinkage Void at Wall /Solid Interface

Figure 3. Schematics of Void Formation in a Solidifying Lithium Coolant in a SP-100 System

2. VOID FORMATION MECHANISMS DURING THE COOLDOWN AND FREEZING PROCESSES OF LITHIUM COOLANT IN THE SP-100 SYSTEM

2.1 Homogeneous and Heterogeneous Nucleation of He Gas Bubbles During the Cooldown Process of Li Coolant

2.1.1 Theoretical Background

In order for a gas nucleus to be thermodynamically stable, its radius should be larger than a critical value. Van Stralen (1979) has shown that the supersaturation, SS_{HN}, necessary for HN of a gas nucleus of critical radius can be represented by:

$$SS_{HN} = \Delta C_h / C_{sat} = 2\sigma / r_{cr} P_L , \qquad (1)$$

where, $\Delta C_h = C_h - C_{sat}$. On the other hand, the supersaturation required for HTN of gas bubbles, SS_{HTN}, may be expressed as (Cole 1974):

$$SS_{HTN} = SS_{HN}\, F(Geo, \theta), \qquad (2)$$

where F is a complex function of wetting angle, θ, and surface geometry. For a flat surface F is only a function of the liquid-solid wetting angle, θ

$$F(\theta) = 1 - \frac{(2 + \cos\theta)(1 - \cos\theta)^2}{4} . \qquad (3)$$

Since, F varies between 0 and 1, SS_{HTN} can be as small as 0 (when liquid Li does not wet the container's wall or when the liquid is rich in solid or gas impurities) and as large as SS_{HN} (when liquid Li perfectly wets the wall or the liquid is pure). An expression for r_{cr} in equation (1) is (Cole 1974):

$$r_{cr} = \left\{ \frac{3kT}{4\pi\sigma} \ln\left(\frac{n_l kT}{hJ}\right) - \frac{E_d}{kT} \right\}^{0.5} . \tag{4}$$

In this equation E_d can be reasonably taken to lie between 0 and the energy of formation of a gas cluster of radius r_{cr} ($0 < E_d < 4\pi r_{cr}^2\sigma/3$), and J can be assumed to lie somewhere between 10^1 and 10^{12} nuc./m^3-s for what is normally thought of as a gas ebullition process. It can be shown that r_{cr} is approximately 1 nm, and is not a strong function of either J or E_d. Therefore, in the following assessment of the potential of He bubble nucleation during the cooldown process of liquid Li in SP-100 system, a value of $r_{cr} = 10^{-9}$ m is used. The values of C_{sat} are determined as functions of liquid Li pressure and temperature using the following correlation suggested by Slotnick et al. (1965):

$$C_{sat} = P_L \exp\left(-\frac{1752}{T} - 23.0\right) . \tag{5}$$

The reduction of Li coolant volume during the cooldown process will be compensated for by volume increase in the gas separator/accumulator (GSA). If we assume an ideal gas system in the GSA, P_L in Equation (1) can be calculated from the ideal gas law:

$$\frac{P_L V}{T} = \frac{P_o V_o}{T_o} . \tag{6}$$

The bubble nucleation potential (BNP) by either HN or HTN is defined as the ratio of actual supersaturation of the dissolved gas to supersaturation required for HN or HTN:

$$BNP = SS_{act}/SS_{HN} \text{ (or } SS_{HTN}\text{)}, \tag{7}$$

He gas bubbles will nucleate when BNP is greater than or equal to unity.

2.1.2 Results and Discussion

In order to assess the potential of He gas bubble nucleation in the SP-100 PHTS, the initial He concentration in Li coolant, C_o, is taken equal to saturation value, C_{sat}, at the nominal operating condition in the SP-100 prior to reactor shutdown (see Table 1). Using the system conditions listed in Table 1, the variations of P_L, Cs and BNP with Li temperature are calculated and the results are shown in Figures 4 and 5. In order for either HN or HTN of He bubble to occur the BNP should be higher than or equal to unity. However, as Figure 5 demonstrates, the BNP values for HN are much smaller than unity, hence HN of He gas bubbles is not possible throughout the entire cooldown process of Li coolant in SP-100. Furthermore, it can be deduced from Equation (2) and Figure 5 that in order for He gas bubbles to form by HTN, the value of F(θ) in Equation (3) needs to be as low as 2.5×10^{-3}, requiring a wetting angle $\theta = 150°$ (see Figure 6). Such high wetting angle is not generally expected in a liquid metal-metal substrate system as in the SP-100. Liquid Li is known to perfectly wet metallic surfaces. Also, dewetting does not occur upon cooling of Li below the wetting temperature (673.5 K) once it is heated above the wetting temperature and maintained at the temperature for a while [Hoffman (1990)]. Therefore, it could be concluded that the formation of He bubbles by HTN during the cooldown process in the SP-100 PHTS is highly unlikely, provided all the system conditions are remaining as fabricated.

2.2 Potential Formation of Intermetallic Compounds During the Cooldown Process of Li Coolant and Their Relaton to the Formation of He Gas Bubbles

It has been concluded in the previous section that the formation of He gas bubbles during the cooldown process of Li coolant is highly unlikely, provided all the system conditions are remaining as fabricated. However, should the characteristics of the metal substrate change due to the formation of intermetallic compounds, such as metal hydride or metal nitride of thickness as little as a few atom radii, the likelihood of dewetting of Li and the formation of He bubbles at the wall by HTN increases.

2.2.1 Potential Formation of Intermetallic Compounds

Because of their high compatibility with Li, Nb and Nb-1%Zr are used in the SP-100 system to fabricate the PHTS pipes and fuel cladding, respectively. As mentioned previously, the (n-Li) reactions in the reactor core produce tritium [McGhee, El-Genk and Rothrock (1989)]. Although the tritium's high diffusivity enables it to readily diffuse out from the PHTS, a small quantities of tritium may remain in the PHTS prior to reactor shutdown. The diffusivity of tritium in Nb rapidly decreases as the temperature of Nb is reduced [Alefeld and Volki (1978)], hence increasing the likelihood of Nb-T reaction during the cooldown process of Li in the SP-100 PHTS and the formation of Nb-T compounds at the inner surface of the pipes wall.

Another source for forming intermetallic compounds in SP-100 PHTS is nitrogen (N_2) gas in the nuclear fuel elements. Because, N_2 gas will be produced due to the dissociation of UN fuel during the operation of the SP-100 system. The amount of N_2 gas dissociated depends on the fuel temperature and burnup. Should the fuel cladding breach, N_2 gas will easily leak from the fuel into the Li coolant. It is also expected that N_2 gas will eventually diffuse through fuel cladding into Li coolant.

Table 2 shows that tritium and nitrogen readily react with Nb and form hydride and nitride at the reaction temperatures indicated. Since, the nominal operating temperature in the PHTS of the SP-100 system is about 1300 to 1350 K, formation of stable NbN is likely on the surface of fuel cladding or on the inner surfaces of the PHTS pipes. As the Li cools down after a reactor shutdown, within a temperature range of $673 < T < 1273$ K, the formation of a mixture of NbN and Nb_2N is expected. When the Li coolant temperature is below 973 K, the formation of LiT occurs and deposits saline LiT in the Li coolant. During the last stage of the cooldown process ($T < 573$ K), the formation of NbT is expected.

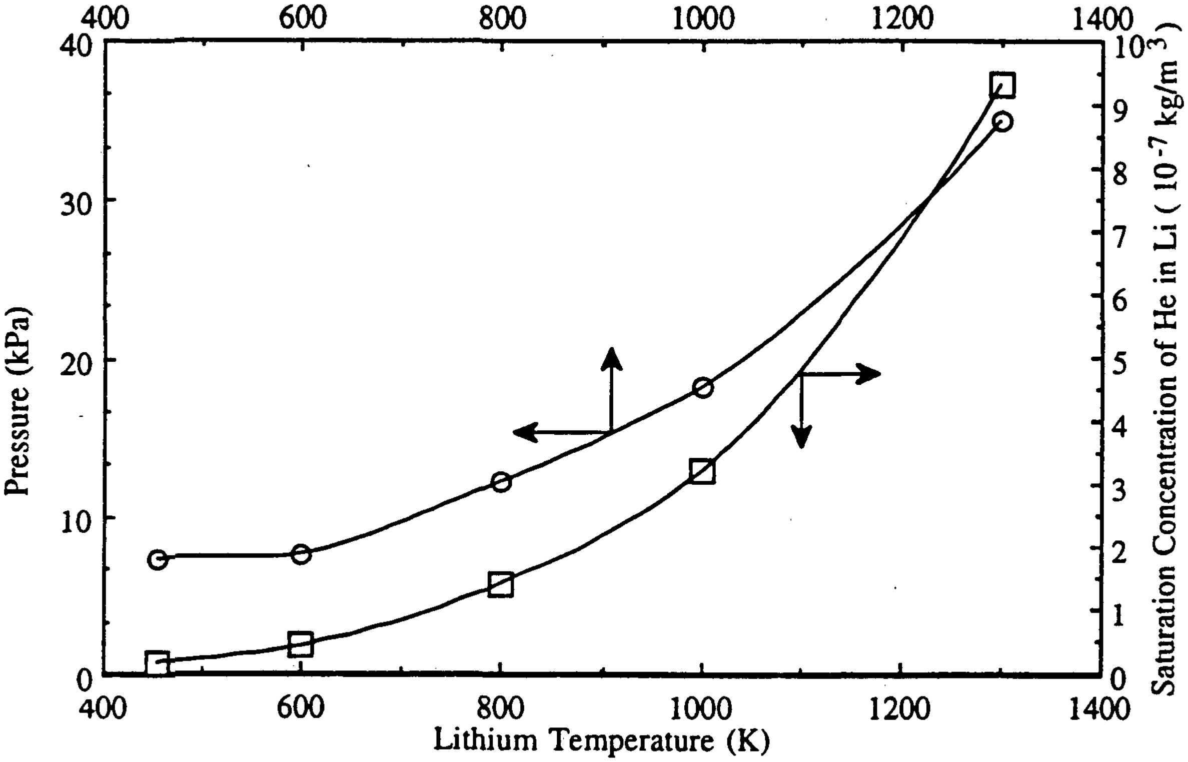


Figure 4. Variation of System Pressure and Saturation Concentration of He in Li with Lithium Temperature

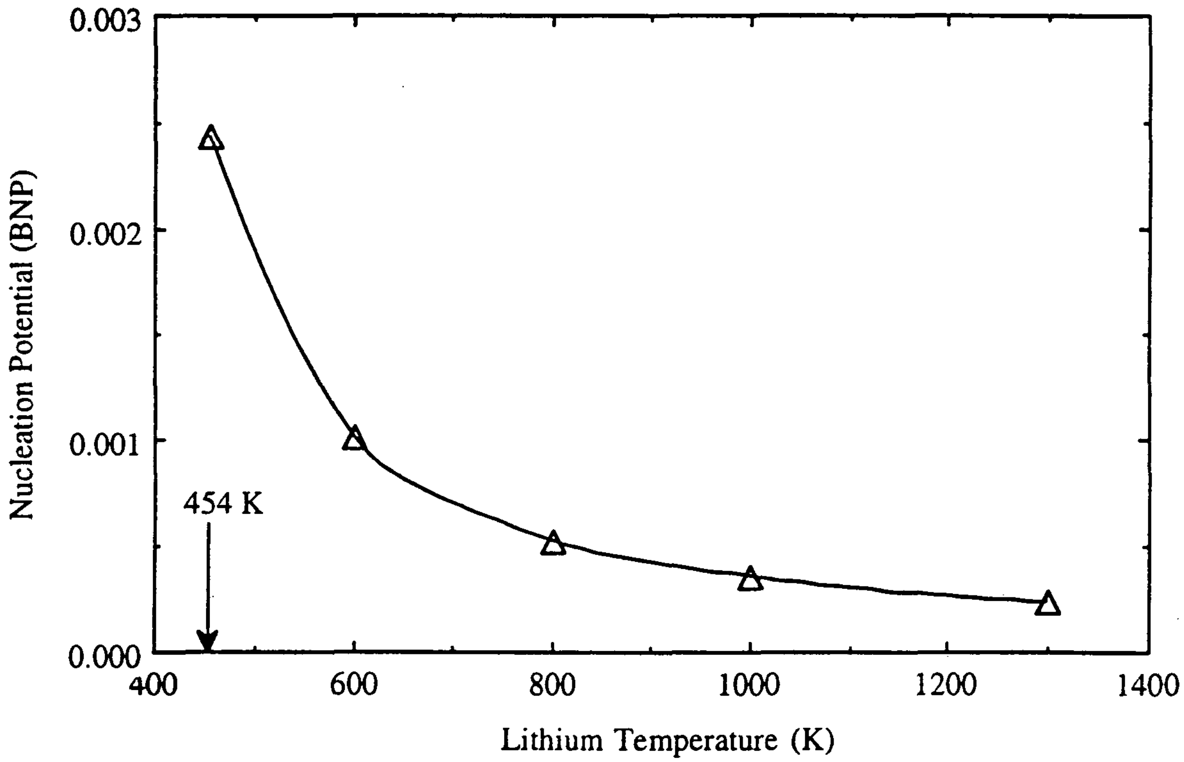


Figure 5. Homogeneous Nucleation Potential during the Cooldown Process of Li in SP-100 PHTS

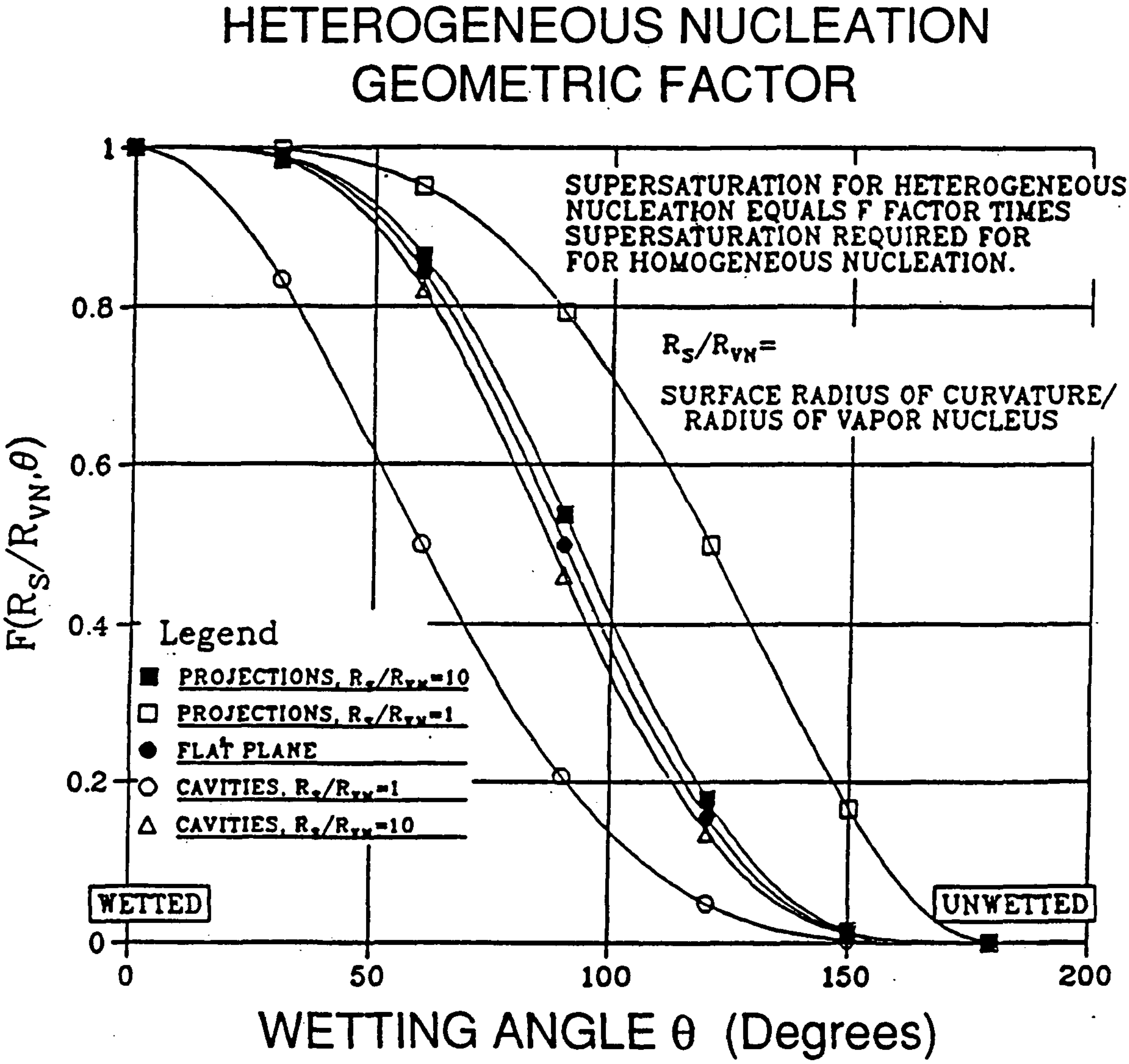


Figure 6. Heterogeneous Nucleation Geometric Factor

2.2.2 Discussion and Suggestion for Future Investigation

The deposit of LiT and the formation of NbT, NbN, and Nb_2N on inner surface of PHTS walls will increase the possibility of dewetting of Li from the walls, hence increasing the potential of void formation by HTN. However, in order to evaluate the potential, wetting angles of Li and mixture of Li and LiT on NbT, NbN, and Nb_2N are needed. Our literature search indicated that such information is not available, hence the potential of He gas bubble formation by HTN in conjunction with the formation and deposit of intermetallic compounds on the inner walls of coolant pipes, reactor core container, and the outer surface of fuel pins remain inconclusive. Experimental efforts to determine the wetting angles of Li on Nb-nitride and Nb-hydride substrates are, therefore, highly recommended.

2.3 He Gas Bubble Formation by Segregation of He Gas Atoms During the Freezing Process of Li Coolant

In the normal segregation process, the motion of He gas (solute) out of the solidified Li is in the direction of movement of solidification interface. The rate of He buildup ahead of the S-L interface is determined by the rate of He rejection from the solid at the solidification front and the rate of He diffusion away from the S-L interface into the liquid. The former process will tend to raise He concentration at the S-L interface, while the latter will reduce it. The rate of He rejection, $J(\delta(t))$, can be expressed as:

$$J(\delta(t)) = (C(\delta, t) - C_s^*)\, V(t)\,. \tag{8}$$

The solidification velocity, $V(t)(= d\delta(t)/dt)$, is obtained from the thermal analysis of the freezing process. In order to determine the potential of He gas formation by normal segregation, a physical model is developed in which the freezing of Li and the mass diffusion of He away from the S-L interface are coupled. In this model the void formation due to shrinkage of Li during freezing is also simulated.

2.3.1 Physical Model

For simplicity, a one dimensional system initially filled with liquid Li at its melting temperature is considered. The liquid Li is cooled by thermal radiation into space at one wall, x = 0, and is thermally insulated at the other wall, x = L. In Figure 3b, the shrinkage void forms at the wall/solidified crust interface, while in Figure 3a, it forms in the liquid at the insulated wall. In this chapter the model in Figure 3a is analyzed. The analysis of the model in Figure 3b will be described in chapter 3 in conjunction with the effect of shrinkage void formation on the freeze and thaw processes. In Figure 3a, the location and velocity of the boundary "B" of the liquid phase can be represented by $X_r = L - \Gamma\delta(t)$, $v(t) = -\Gamma(d\delta(t)/dt)$, respectively, where $\Gamma = (\rho_S/\rho_L - 1)$. The He concentration in liquid Li is assumed uniform prior to the initiation of the freezing process and equal to its saturation concentration in liquid Li at 1350 K (C_O = 4.26 x 10^{-6} kg/m^3). The segregation model consists of two coupled models: (a) freezing model of liquid Li, and (b) mass diffusion model of He gas in liquid Li ahead of the S-L interface. Because the problem in Figure 3a entails two moving boundaries, S-L interface "A" and shrinkage void/liquid interface "B", and a boundary condition of the third kind (radiation), it is highly nonlinear. Hence, an exact analytical solution is not attainable. Instead, a numerical approach is used to solve coupled freezing and gas diffusion processes.

Freezing Model of Liquid Li

For the one dimensional system shown in Figures 3a and 7a, the heat balance equations in the liquid and solid Li and initial and boundary conditions can be written as:

(A) Liquid Li ($\delta \le x \le X_r$)

$$\rho_L C_{pL}\left(\frac{\partial T_L}{\partial t} + v\frac{\partial T_L}{\partial x}\right) = \frac{\partial}{\partial x}\left(k_L\frac{\partial T_L}{\partial x}\right) , \qquad (9)$$

(B) Solid Li ($0 \le x \le \delta$)

$$\rho_s C_{ps}\frac{\partial T_s}{\partial t} = \frac{\partial}{\partial x}\left(k_s\frac{\partial T_s}{\partial x}\right) , \qquad (10)$$

(C) Initial and Boundary Conditions

$$T(x,0) = T_f, \tag{11a}$$

$$k_s \frac{\partial T}{\partial x}(0, t) = \varepsilon\sigma\left(T^4(0, t) - T_a^4\right) = h_r\left(T(0,t) - T_a\right), \tag{11b}$$

$$\frac{\partial T}{\partial x}(X_T, t) = 0, \tag{11c}$$

$$T_s(\delta, t) = T_L(\delta, t) = T_f, \tag{11d}$$

$$k_s \frac{\partial T_s}{\partial x}(\delta, t) - k_L \frac{\partial T_L}{\partial x}(\delta, t) = \rho_L \lambda\left(\frac{d\delta(t)}{dt} - v\right). \tag{11e}$$

Because of the high thermal conductivity of Li, the temperature differential in the frozen crust is expected to be small. Hence, constant properties can be assumed. When constant properties in each spatial mesh, Δx, within the solid and liquid Li are assumed, Equation 9 can be written in terms of enthalpy as [Shamsunder and Sparrow (1975), Hunter and Kuttler (1989)]:

$$\frac{\partial}{\partial t}\int_x^{x+\Delta x} \rho H \, dx + \left[\rho v H - k\frac{\partial T}{\partial x}\right]_x^{x+\Delta x} = 0 \tag{12}$$

The dimensionless nodal enthalpy and temperature are defined as:

$$\theta \equiv \frac{1}{\rho \Delta x}\int_{\Delta x} \rho \frac{(H - H_{sf})}{\lambda} \, dx, \qquad \phi \equiv \frac{C_p(T - T_f)}{\lambda} \tag{13}$$

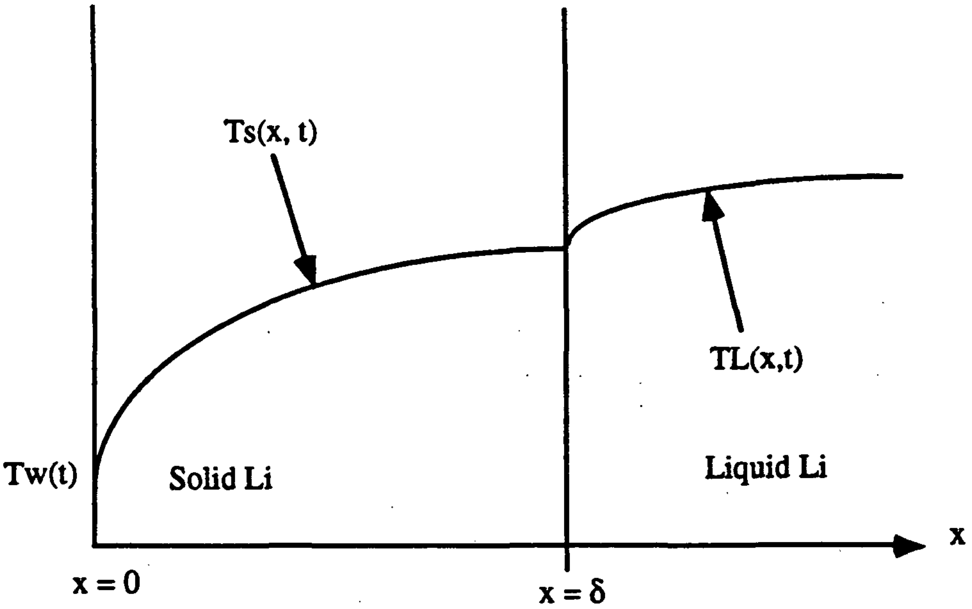
It has been shown by Shamsunder and Sparrow (1975), that θ and ϕ have the following relationships:

$$\phi = a + b\theta,$$

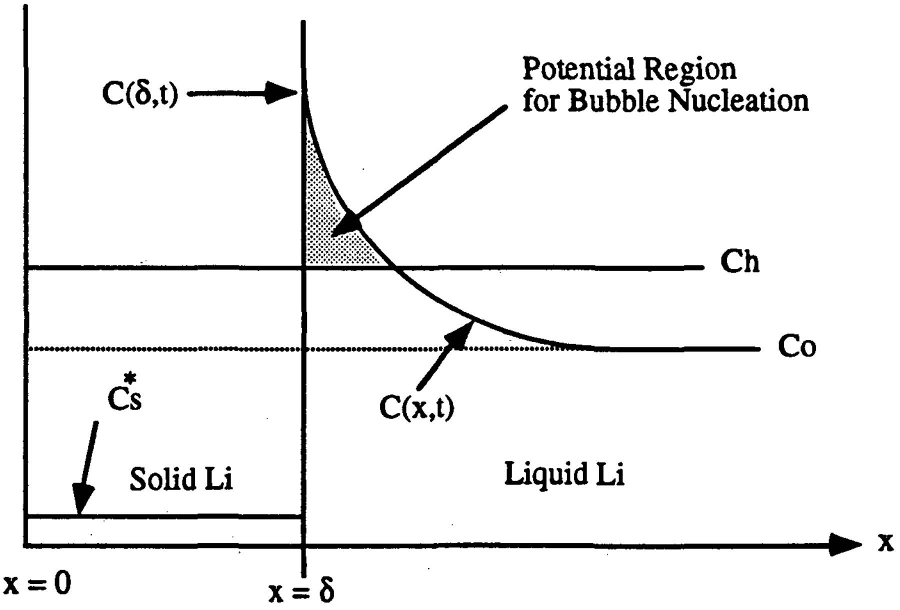
where,

$$a = 0, \; b = 1 \quad , \text{for } \theta < 0 \tag{14}$$

$$a = 0, \; b = 0 \quad , \text{for } 0 \leq \theta \leq 1.$$



(a) Freezing Process



(b) He Segregation Process

Figure 7. Schematic Representation of Freezing and He Segregation Processes

Using these dimensionless parameters, Equation (9) is transformed into:

$$\frac{\partial \theta}{\partial t} = \frac{1}{\Delta x \rho}\left[\frac{k}{C_p}\frac{\partial \phi}{\partial x}\right]_{x}^{x+\Delta x} - v\frac{\partial \theta}{\partial x} \tag{15}$$

It has been shown by Shamsunder and Sparrow (1975) that in the absence of convection term due to volume shrinkage, Equation (15) satisfies Equation (11e). It can also be shown that the same approach can be applied in the case with convection due to volume shrinkage to show that the energy conservation at the S-L interface is satisfied by Equation (15). Using backward finite difference formulation for the time derivative and forward difference formulation of the spacial derivatives Equation (15) is discretized as:

$$\frac{\theta_i^m - \theta_i^{m-1}}{\Delta t} = \frac{1}{\Delta x \rho_i}\left[\left(\frac{k}{C_p}\right)_e \frac{\left(\phi_{i+1}^{m-1} - \phi_i^m\right)}{\Delta x} - \left(\frac{k}{C_p}\right)_w \frac{\left(\phi_i^m - \phi_{i-1}^m\right)}{\Delta x}\right] + \Gamma\frac{\left(\delta^m - \delta^{m-1}\right)}{\Delta t}\frac{\left(\theta_{i+1}^{m-1} - \theta_i^m\right)}{\Delta x} . \tag{16}$$

The heat balance equation for the solid [Equation (10)] can be transformed and linearized as in Equations (15) and (16) by noting that v in the solid region is zero in and the last term of the Equation (16) drops out. The location of S-L interface, δ, is determined by $\delta = x_i - \theta_i \Delta x_i$ when the value of θ of the i-th node is less than 1 and greater than 0, since the frozen fraction of the node, α, can be conveniently related to θ as $\alpha = 1 - \theta$ from the definition of θ shown in Equation (13). The transformation and linearization of the initial and boundary conditions in Equation (11) are carried out consistently with those of the governing equations and the values of θ and ϕ at each time step are determined by a fully implicit scheme suggested by Shamsunder and Sparrow (1975). The program developed in accordance with the foregoing considerations (COPHASE.FOR) is listed in Appendix A-1 and input data required to run the program (COPHASE.INP) is listed in Appendix A-2.

Mass Diffusion Model of He Gas

The He segregation process during freezing of liquid Li is delineated in Figure 7b. The governing equation, initial and boundary conditions for He gas mass diffusion process in the liquid region ($x \geq \delta(t)$) are:

$$\frac{\partial C}{\partial t} + v\frac{\partial C}{\partial x} = D\frac{\partial^2 C}{\partial x^2}$$

(17)

$$C(x, 0) = C_0 \qquad (18a)$$

$$J(\delta(t)) = (C(\delta, t) - C_s^*)V(t) \qquad (18b)$$

$$\frac{\partial C}{\partial x}(X_r, t) = 0 \;. \qquad (18c)$$

Discretization of Equation (17) is done using Patankar's high flux approximation [Patankar (1980)], namely:

$$\frac{\Delta x_i}{\Delta t}(C_i^{n+1} - C_i^n) + J_{i,+} - J_{i,-} = 0 \;. \qquad (19)$$

The convection-diffusion flux at the left ($J_{i,-}$) and right ($J_{i,+}$) boundary of node i are defined as:

$$J_{i,sign} = vC_j + D_j^e F(Pe_j^e)(C_j - C_{j+1}), \; (j = i \text{ for sign} = + \text{ and } j = i\text{-}1 \text{ for sign} = -) \qquad (20a)$$

where,

$$D_j^e = 2D/\left(\Delta x_j + \Delta x_{j+1}\right), \qquad (20b)$$

$$F\left(Pe_j^e\right) = < 0, \left(1 - 0.1\left|Pe_j^e\right|\right)^5 > + < 0, - Pe_j^e >, \qquad (20c)$$

$$Pe_j^e = v/D_j^e. \qquad (20d)$$

The operator <a, b> is the largest of a and b. After linearizing the initial and boundary conditions in a consistent manner with that of the governing equation, He concentration in the liquid ahead of S-L interface, $C(\delta,t)$, and the values of C_i at each time step are determined by using a tridiagonal matrix solver.

Volume Estimate of He Gas Voids

As shown in Figure 7b, when the He gas concentration ahead of the S-L interface, $C(\delta,t)$, reaches the HN concentration, He bubbles will nucleate and grow by absorbing gas atoms adjacent to them. The bubble growth will continue until all the gas atoms in excess of the solubility limit, or saturation concentration, C_{sat}, have diffused into the bubbles. Because detailed analysis of the bubble nucleation and growth processes is beyond the scope of this work, it is assumed that He gas atoms in excess of C_h will be instantaneously contained in bubbles. The volume of these bubbles is calculated as:

$$V_{He} = \frac{n_{He} RT}{P}, \tag{21}$$

where,

$$n_{He} = \frac{1}{A_w} \int (C(x,t) - C_h)\,dx \quad . \tag{22}$$

The numerical integration of the Equation (22) is carried out within the liquid region where the local He concentration, $C(x,t)$, is greater than C_h. The program simulating the segregation process and the gas bubble evolution (SEGR.FOR) is listed in Appendix B-1 and input data required to run the program (SEGR.INP) is listed in Appendix B-2.

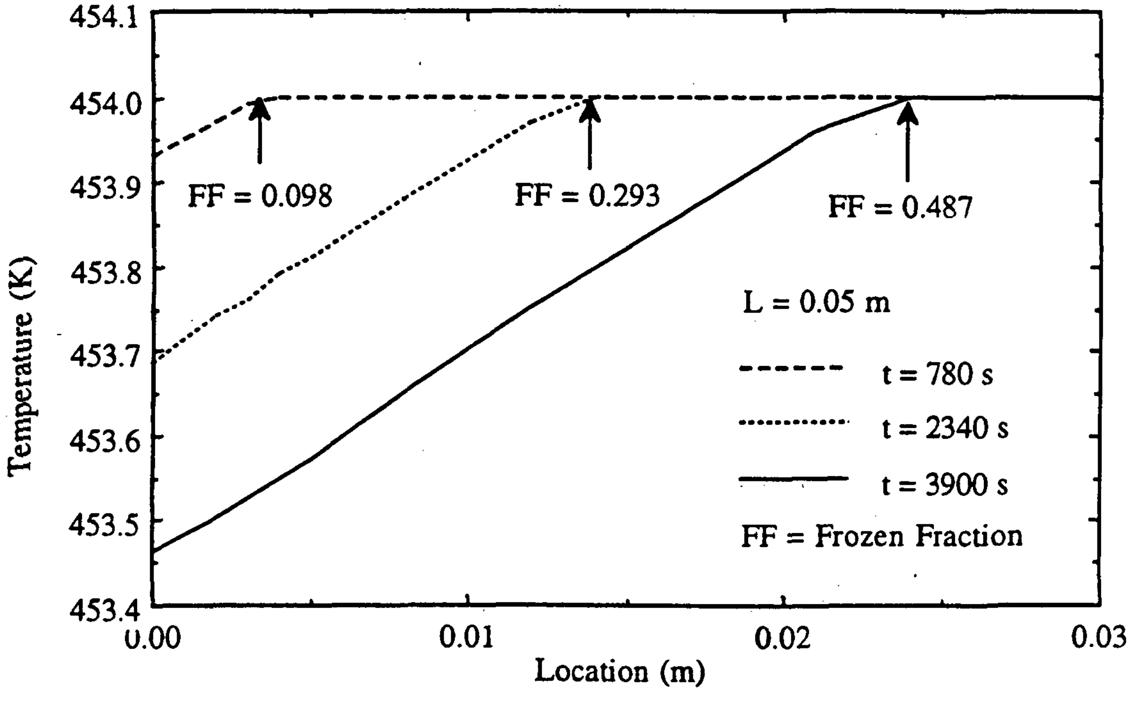
2.3.2 Results and Discussion

By using the thermal properties of Li, the base case parameters listed in Table 3 and the computer programs COPHASE.FOR and SEGR.FOR, the potential of He gas bubble formation by normal segregation and shrinkage of of Li during the freezing of Li coolant in the SP-100 primary coolant system is assessed and the results are presented and discussed in this subsection.

The segregation model described earlier is applied to three different coolant channel widths (0.01m, 0.05m, and 0.1m). The results are presented in Figures 8, 9 and 10. Figure 8 delineates the calculated temperature profiles and Figure 9 shows the calculated He concentration profiles during the freezing process of liquid Li. It can be seen in Figure 8 that the temperature profile in the solid Li is almost linear. Also, the solidification velocities and the wall temperatures are independent of coolant channel width (see Figure 10). The solidification velocity and rate of change in the wall temperature are approximately 8×10^{-6} m/sec and -2×10^{-4} K/sec, respectively. Note that the freezing process of Li in SP-100 is extremely slow, suggesting a steady-state treatment of the problem could be sufficient. Also, a constant wall heat flux approximation may be used in the analysis of freezing process of Li with radiative heat transfer boundary. The low freezing velocity of Li is due to its high heat capacity and high latent heat of fusion as well as the low cooling rate at the wall. Figure 11 presents the calculated frozen fraction of the solidified liquid as a function of channel width, L.

The results in Figures 12 and 13 show the volume of He gas bubbles formed by normal segregation and that of shrinkage voids during the freezing process of Li. The results indicated that the onset of He bubble nucleation occurs approximately 350 seconds after the initiation of freezing at which the thickness of solidified Li is only about 3 mm. These results suggest that formation of He gas bubble by normal segregation would occur throughout the PHTS of the SP-100 system during the Li freezing. It can be also seen in Figure 12 that the contribution of segregation bubble is only about 2% of the total volume of the void at the completion of solidification of each coolant channel. This corresponds to 0.055% of the volume of the coolant channels considered.

One may argue that this amount of gas bubbles will not be sufficient to effectively remove the stress in the walls of PHTS during a subsequent restart of the SP-100. However, if these bubbles are captured by the freezing front into the solid Li, they will grow to compensate the volume shrinkage of Li. Thus, upon restart, the He gas/shrinkage bubbles in the solid Li would effectively reduce the stress induced by the volume increase of Li. However, if the He gas bubbles are continuously driven by the solidification front, they will eventually collapse into the shrinkage voids. In order to minimize the stress during the restart of SP-100 and melting of Li, heating should be applied where the He/shrinkage void eventually form (see Figure 3). This would initially cause overheating of the wall (hot spot) due to the presence of the voids, but it will accommodate the increase in Li volume upon melting. It is not at all clear at present which of the two arrangement described in Figure 3 would occur in a solidifying Li in space. The subsequent research, therefore, focuses on the thermal effect of shrinkage void formation on the freeze and thaw processes of Li coolant in the SP-100 system.



(a) Temperature Profile

Figure 8. Calculated Temperature Profile

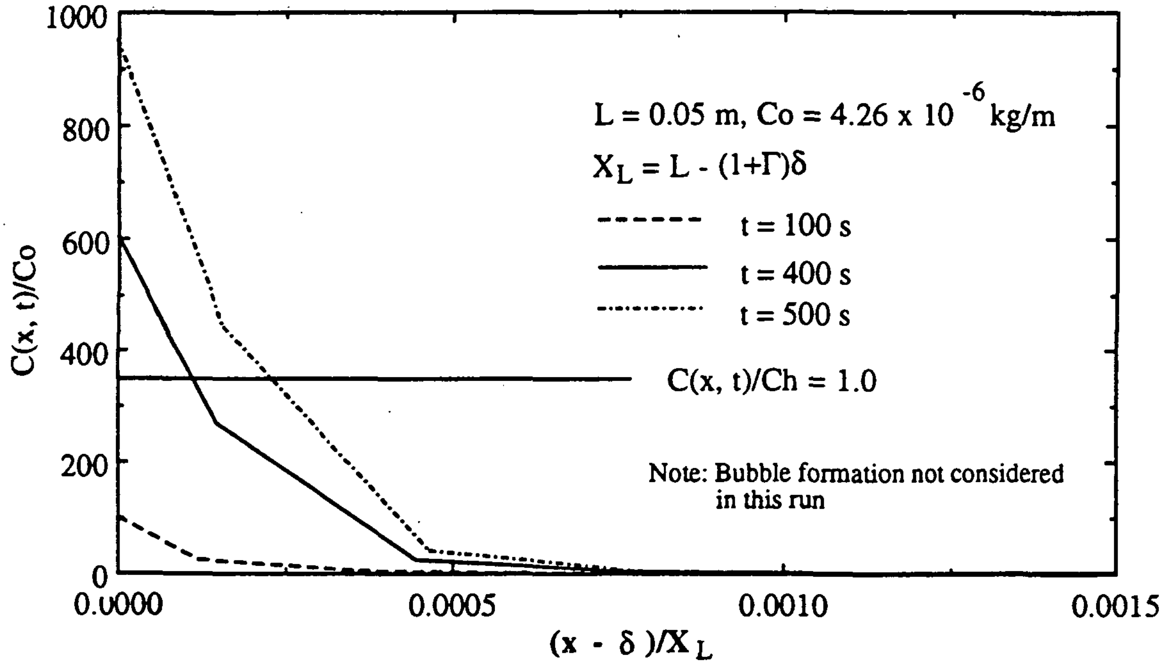


Figure 9. Calculated Dimensionless Concentration Profile

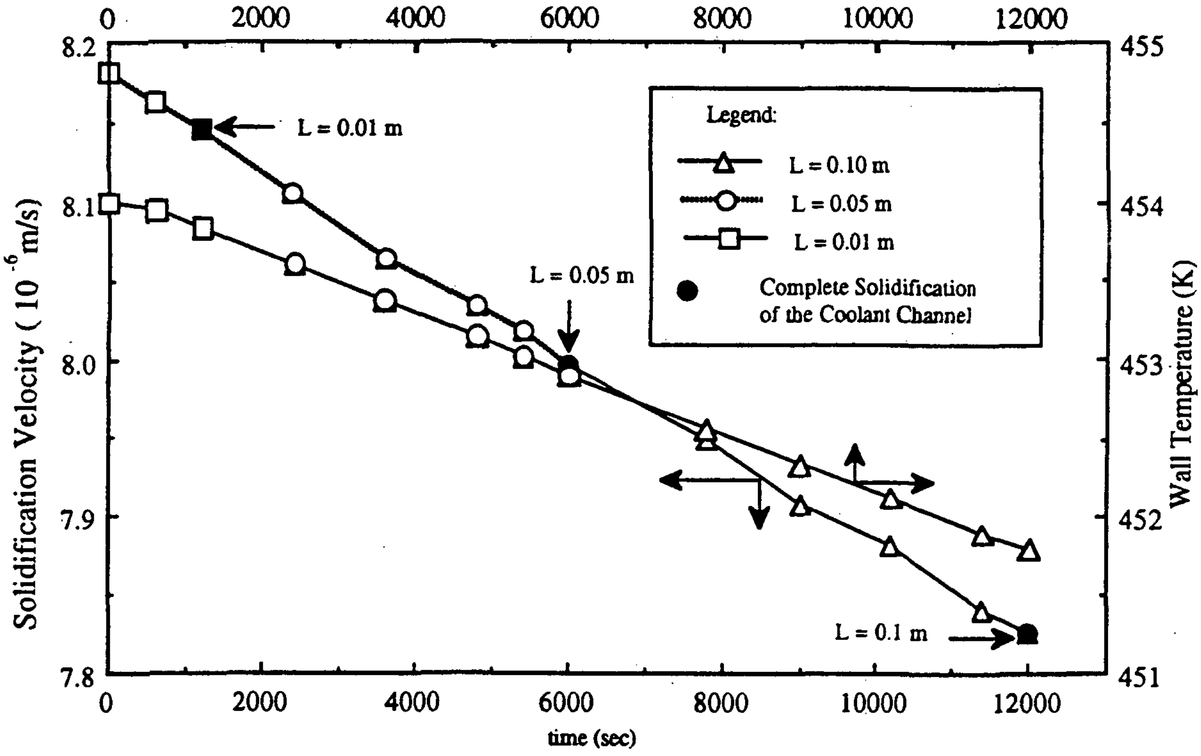


Figure 10. Variation of Solidification Velocity and Wall Temperature with Time for Different Coolant Channel Widths

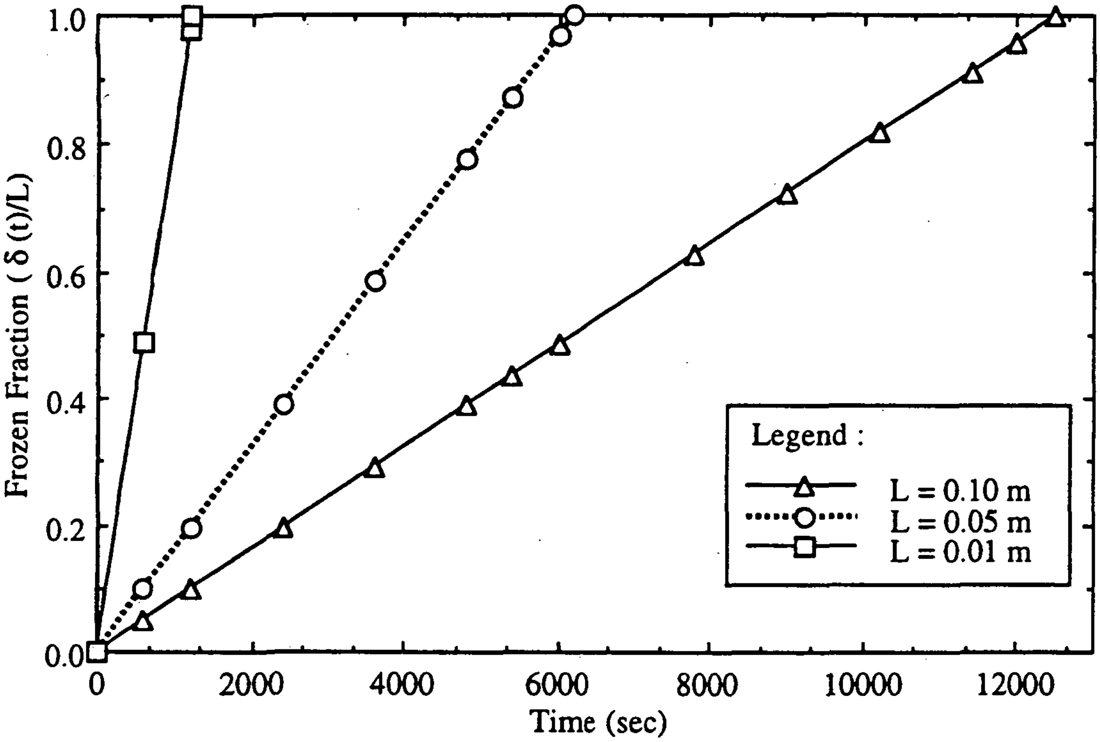


Figure 11. Variation of Frozen Fraction for Different Coolant Channel Widths

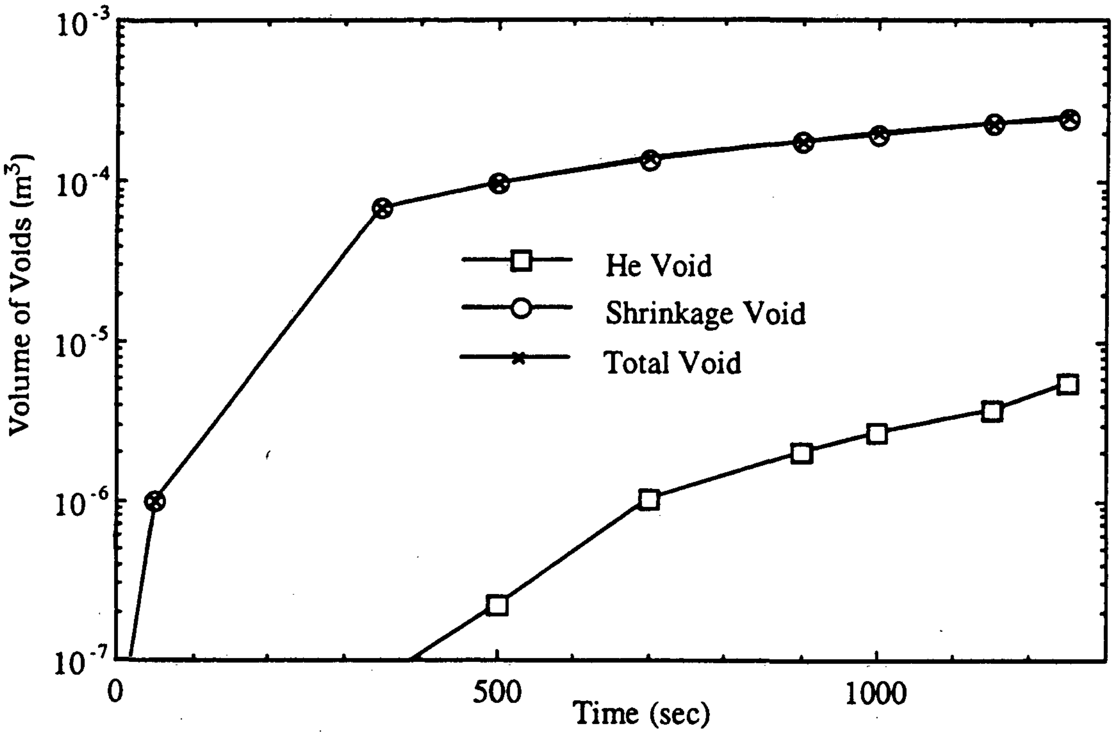


Figure 12. Evolution of Voids During Lithium Coolant Cooldown and Freezing (Coolant Channel Width = 0.01 m, Initial He Concentration = 4.26×10^{-6} kg/m^3)

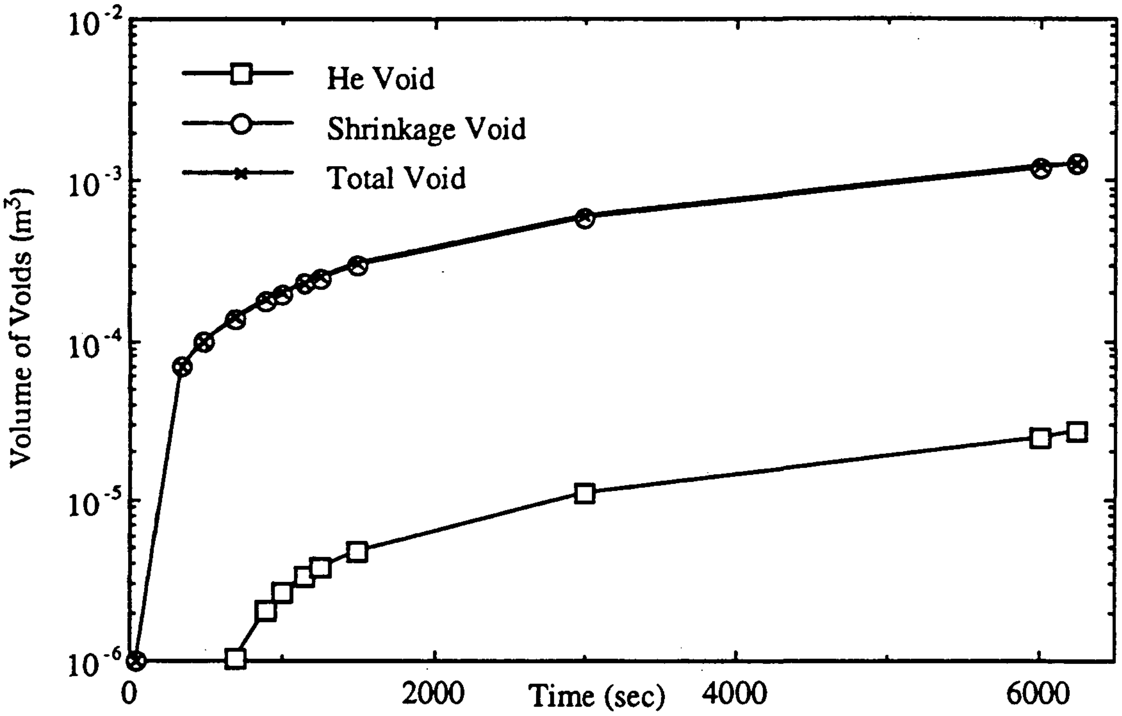


Figure 13. Evolution of Voids During Lithium Coolant Cooldown and Freezing, (Coolant Channel Width = 0.05 m, Initial He Concentration = 4.26 x 10^{-6} kg/m^3)

2.4 Nomenclature

English

A_W : Atomic weight (g/mole)
BNP : Potential for bubble nucleation
C_p : Heat capacity (J/kg.K)
Δx : Spacial mesh size (m)
C_s^* : Solubility limit in the solid (kg/m^3)
C : Gas concentration concentration (kg/m^3)
D : Diffusion coefficient (m^2/s)
E_d : Activation energy for diffusion of a molecule of gas through the liquid (J)
H : Enthalpy (J/kg)
h : Boltzman constant (6.26 x 10 $^{-34}$ J/s)
h_r : Radiative heat transfer coefficient (W/m^2.K)W//m^2.K)
J : Nucleation rate in Equation (4) (Nucleations/s)
J : Convection-diffusion flux (kg/m^2.s)
n_t : Number density of gas molecules (molecules/m^3)
k : Thermal conductivity (J/m.K)
L : Coolant channel width (m)
n_{He} : Number of moles of He gas
P : Pressure (Pa)
Pe : Peclet number in Equation (20d)
R : Gas Constant (8.3144 J/mole.K)
r_{cr} : Critical radius of bubble (m)
T : Temperature (K)
t : Time (s)

$V(t)$: S-L interface velocity (m/s)

V : Volume of the PHTS (m^3)

V_{He} : Volume of He bubbles (m^3)

v : Material velocity due to volume shrinkage (m/s)

Xr : Location of shrinkage void/liquid (or solid) interface (m)

x : Spacial coordinate (m)

Greek

δ : Solid crust thickness (m)

ε : Emissivity

ϕ : Dimensionless nodal temperature

λ : *Latent heat of fusion (J/kg)*

θ : Dimensionless nodal enthalpy or Wetting angle

α : Frozen fraction of a node

Γ : $\rho_S/\rho_L - 1$

ρ : Density (kg/m^3)

σ : Surface tension in Equation (1) (N/m)

σ : Stefan-Boltzman constant (5.67×10^{-8} $W/m^2 K^4$)

Superscripts

e : Right boundary of a node

n+1 or m : Present time step

n or m-1 : Previous time step

Subscripts

a : Ambient

SS : Supersaturation

act : Actual

e : Right boundary of a node

f : At melting temperature

HN, h: Homogeneous nucleation

HTN : Heterogeneous nucleation

i,j : i-th and j-th node, respectively

o : Initial condition

s : Solid

L : Liquid

sat : Saturation

w : Left boundary of a node

Table 1. The SP-100 System Nominal Operating Conditions.

Pressure	3.5×10^4 Pa
Temperature	1350 K
Li Mass in the PHTS	110 kg
Li volume in the PHTS	255 Liter
Gas Separator/Accumulator volume	35 Liter
Average He concentration	4.26×10^{-6} kg/m^3

Table 2. Temperature Ranges and the Free Energies of Formation of NbT, NbN and LiT in SP-100.

Reaction	Temperature	ΔF(kcal/mole)[a]
(1) NbN Formation	T>1273 K	- 50.1
(2) Mixture of Nb2N and NbN Formation	673 < T < 1273 K	- 54.0
(3) LiT Formation	773 < T < 973 K	- 16.7
(4) Full Dissociation of LiT	T > 973 K	
(5) NbT Formation	T < 573 K	- 3.2

a. The free energy of formation is the value at T = 298 K.

Table 3. Thermal Properties and Base Case Parameters Used in the Analysis of Helium Gas Segregation During Lithium Freezing

Material	Property	Value
Liquid Li	Thermal conductivity	42.8 J/s.m.K
	Density	517 kg/m^3
	Heat capacity	4.384 x 10^3 J/kg.K
	Thermal diffusivity	1.89 x 10^{-5} m^2/s
Solid Li	Thermal conductivity	75.8 J/s.m.K
	Density	530 kg/m^3
	Heat capacity	3.78 x 10^3 J/kg.K
	Thermal diffusivity	3.78 x 10^{-5} m^2/s
Base Case Parameters	Initial He concentration	4.26 x 10^{-6} kg/m^3
	Diffusivity of He in liquid Li	5.96 x 10^{-9} m^2/s
	Diffusivity of He in solid Li	Negligible
	Li Latent Heat of Fusion	4.321 x 10^5 J/kg
	Li Freezing Temperature	454 K

Table 4. Thermal Properties and Base Case Parameters used in the Analysis of Freeze and Thaw Processes of Li and LiF

Material	Properties	Liquid	Solid	Units
Li	k	42.8	75.8	W/m.K
	ρ	517	530	kg/m^3
	Cp	4384	3780	J/kg.K
	α	1.89×10^{-5}	3.78×10^{-5}	m^2/s
		H_{sl}	4.321×10^5	J/kg
	T_f	454		K
LiF	k	1.30	5.00	W/m.K
	ρ	1376	1792	kg/m^3
	Cp	2779	2470	J/kg.K
	α	3.40×10^{-5}	2.94×10^{-5}	m^2/s
	H_{sl}	3.473×10^5		J/kg
	T_f	1122		K
PWC-11	k		23.9	W/m.K
	ρ		6490	kg/m^3
	Cp		323.4	J/kg.K
	α		1.14×10^{-5}	m^2/s
	T_f		2740	K
LiF vapor	k	0.017		W/m.K
He gas	k	0.1678		W/m.K

Base Case Parameters

Ambient Temperature of Space	250	K
Temperature of Wall or Heat Source		
Freeze:	250	K
Thaw:	1350	K
Thickness of PCM at Liquid State	0.05	m
PWC-11 Wall Thickness	7.6×10^{-4}	m
Emissivity		
at outer surface of the wall	0.8	
at inner surface of the wall	0.5	

3. THE EFFECTS OF SHRINKAGE VOID FORMATION ON THE FREEZE AND THAW PROCESSES OF LITHIUM COOLANT

3.1 Physical Model

As mentioned in the previous chapter, because the contribution of gas voids volume to total voids forms in the SP-100 system is small (~2%), its effect on the thermal behavior during the freeze and thaw processes is expected to be small and may be negligible. Therefore, in this chapter, only the effect of shrinkage void formation on the freeze and thaw processes is considered. Instead, cases with lithium-fluoride are analyzed for the comparison's purpose and to satisfy the practical interests, since LiF exhibits much larger volume shrinkage upon freezing (23.2% compared to 2.5% for the case of Li) and it is known as an excellent phase change material (PCM) for thermal energy storage (TES) for solar dynamic power system. In addition, it is important to show the temperature variation in the heat transfer surface, the model being used in this analysis considers the effect of heat transfer wall.

A simple numerical scheme is developed to simulate two different modes of shrinkage void formation in a finite one dimensional system, namely: (1) a shrinkage void forming at solid-wall interface (or wall void WV afterwards) (Figure 14a); and, (2) a shrinkage void forming at the center (or adiabatic boundary) of the system (center void or CV, afterwards) (Figure 14b). The former will occur in the absence of good adhesion between solidified PCM and the wall, while the latter is expected if liquid poorly wets the wall. Modeling this problem requires solving the heat transfer equations with two moving boundaries, namely, the solid-liquid (S-L) interface and the PCM-void (P-V) interface, subject to radiative heat transfer at the wall.

Because an analytical solution for such non-linear problems is not attainable, a numerical solution based on a single (solid/liquid) cell approach is used. The problem studied is that of a one dimensional slab initially filled with PCM at its fusion temperature in a container having a wall thickness, t_c, symmetric about $x = L$ (see Figure 14). The wall is either radiatively cooled into space or radiatively heated from an isothermal heat source (e.g. a thaw heat pipe) at a temperature, T_a. The other side of the slab, $x = L$, is thermally insulated. The heat transfer within the void, that is forming during the freezing process, is by conduction and radiation. The conductance of the void is assumed to vary inversely with the size of void, i.e. $h_{vc} = k_v/\delta_v$ [Wilson and Solomon, 1986, Hunter and Kuttler, 1983]. The void is assumed to be filled with He gas and LiF vapor for the cases of Li and LiF, respectively. The sensible heat of the vapor is assumed negligible. Also, the temperature differential in the frozen crust is expected to be small because of the high thermal conductivity of Li. Hence, constant properties are assumed. During the freezing process, the location and velocity of the boundary "B" of the solid-represented by $X_r = \gamma S(t)$, $v(t) = \gamma(dS(t)/dt)$, respectively, where $\gamma = (1 - \rho_L/\rho_s)$. In Figure 14b, the boundary "B" representing the liquid-void interface moves in the opposite direction, hence $X_r = L - \Gamma\delta(t)$, and $v(t) = -\Gamma(d\delta(t)/dt)$, where $\Gamma = (\rho_s/\rho_L - 1)$.

During the thaw process, the location and velocity of the boundary "B" in Figure 14a can be represented as, $X_r = X_{ri} - \gamma\delta(t)$, and $v(t) = -\gamma(d\delta(t)/dt)$, respectively, where, X_{ri} is the initial thickness of void. In Figure 14b, $X_r = L - X_{ri} + \Gamma\delta(t)$, and $v(t) = \Gamma(d\delta(t)/dt)$.

Governing Equations

For the one-dimensional system shown in Figure 14 the temperature in the receding phases is uniform and equal to fusion temperature. The heat balance equations in the developing phase (subscript d), and in the heat transfer wall (subscript c) are:

Developing Phase

$$\rho_d C_{pd} \frac{\partial T_d}{\partial t} = \frac{\partial}{\partial x}\left(k_d \frac{\partial T_d}{\partial x}\right), \tag{22}$$

Heat Transfer Wall

$$\rho_c C_{pc} \frac{\partial T_c}{\partial t} = \frac{\partial}{\partial x}\left(k_c \frac{\partial T_c}{\partial x}\right), \tag{23}$$

Initial and Boundary Conditions

$$T(x, 0) = T_f , \tag{24}$$

$$T_s\big(S(t), t\big) = T_L\big(S(t), t\big) = T_f , \tag{25}$$

$$k_s \frac{\partial T_s}{\partial x}(S(t), t) - k_L \frac{\partial T_L}{\partial x}(S(t), t) = \rho_L H_{sl}\left(\frac{d\delta(t)}{dt} - v\right), \tag{26}$$

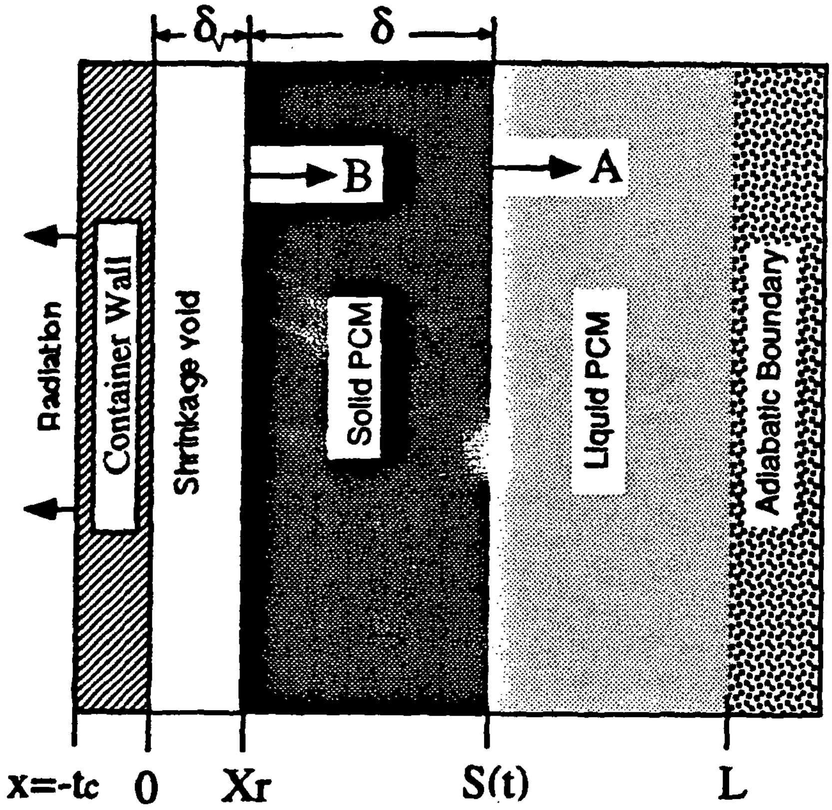
and

$$k_c \frac{\partial T}{\partial x}(-t_c, t) = \varepsilon\sigma\left(T^4(-t_c, t) - T_a^4\right), \tag{27}$$

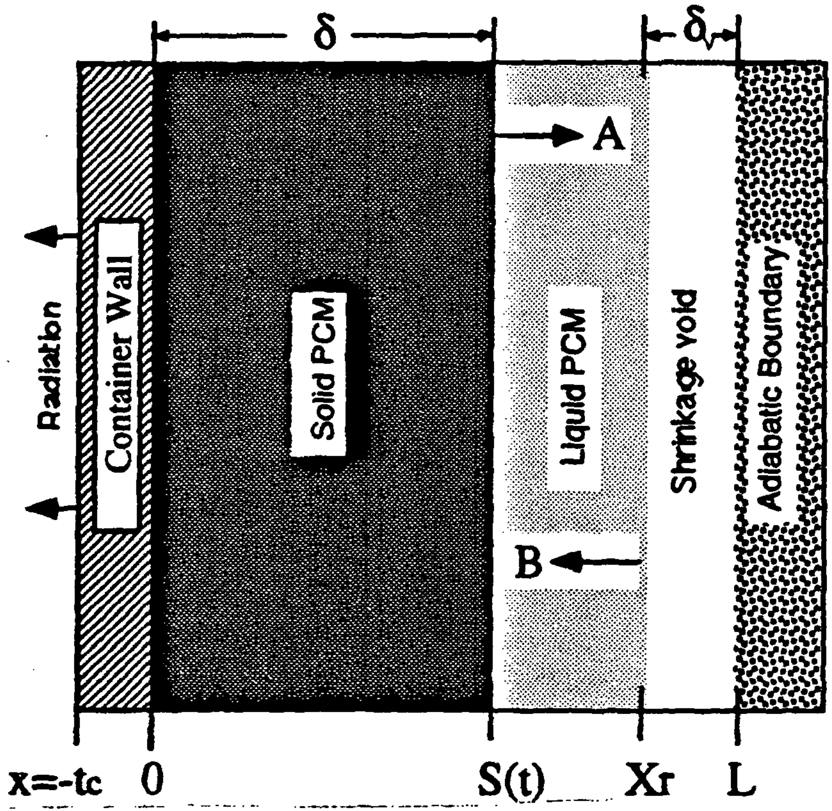
$$k_s \frac{\partial T}{\partial x}(X_r, t) = \varepsilon\sigma\left(T^4(X_r, t) - T^4(0, t)\right) + \frac{k_v}{\delta_v}\left(T(X_r, t) - T(0, t)\right), \quad \text{for Fig 14a,} \tag{28}$$

$$\frac{\partial T}{\partial x}(X_r, t) = 0, \quad \text{for Fig 14b.} \tag{29}$$

In a recent work by Yang and El-Genk (1991), it has been shown that when a radiative heat transfer boundary condition is applied to the freezing of Li, a linear temperature profile develops within the solid crust and quasi-steady state approximations can be used. When the thickness of a developing phase is small, a linear temperature profile may be used. The governing equations are linearized using a fully implicit scheme as follows:



(a) Shrinkage Void Forming at Solid/Wall Interface



(b)Shrinkage Void Forming at Center of System

Figure 14. Schematics of Shrinkage Void Formation in a Solidifying Lithium Coolant in a SP-100 System

Developing Phase

$$q''_1 - q''_{12} = \left[\rho_d C_{pd}\left(T_2^{n+1}\delta^{n+1} - T_2^{n}\delta^{n}\right) - \rho_d C_{pr} T_f\left(\delta^{n+1} - \delta^{n}\right)\right] / \Delta t, \tag{30}$$

where,

$$q''_1 = -h_{01}\left(T_1^{n+1} - T_a\right), \tag{31}$$

$$q''_{12} = -h_{12}\left(T_2^{n+1} - T_1^{n+1}\right), \tag{32}$$

$$q''_2 = -h_2\left(T_f - T_2^{n+1}\right), \tag{33}$$

$$h_{01} = \left(T_1^2 + T_a^2\right)\left(T_1 + T_a\right) / \left[1/\varepsilon\sigma + 4T_1/h_1\right], \tag{34}$$

$$h_1 = \frac{2K_c}{t_c}, \tag{35}$$

$$h_{12} = \left[1/h_1 + 1/h_v + 1/h_2\right]^{-1}, \tag{36}$$

$$h_2 = \frac{2K_d}{\delta}, \tag{37}$$

$$h_v = \frac{K_v}{\delta_v} + \varepsilon\sigma\left(T_3^2 + T_4^2\right)\left(T_3 + T_4\right), \tag{38}$$

$$T_3 = h_{12}\left[\left(\frac{1}{h_2} + \frac{1}{h_v}\right)T_1 + \frac{T_2}{h_1}\right], \tag{39}$$

$$T_4 = h_{12}\left[\frac{T_2}{h_2} + \left(\frac{1}{h_1} + \frac{1}{h_v}\right)T_2\right], \tag{40}$$

Heat Transfer Wall

$$q''_1 - q''_{12} = \rho_c C_{pc} t_c \frac{\left(T_1^{n+1} - T_1^{n}\right)}{\Delta t}. \tag{41}$$

The superscripts n and n+1 represent previous and present time steps. In order to solve for the temperatures of the wall and developing phase, the thickness of the developing phase, δ, should be known. An iterative procedure is used to determine δ by solving the linearized heat balance at the S-L interface:

$$\frac{2k_d}{\delta}\left(T_f - T_2^{n+1}\right) = (Pa)\rho_s H_{sl}\left(\frac{\delta^{n+1} - \delta^n}{\Delta t}\right), \tag{42}$$

where, Pa is equal to 1 for freezing and -1 for thawing.

The computer program developed to assess the effect of shrinkage void on the freezing process of Li and LiF (FREZ.FOR) and that for melting process (MELT.FOR) are listed in the Appendices C and D, respectively.

3.2 Results and Discussion

3.2.1 Comparison with Available Analytical Solution

The accuracy of the numerical scheme is verified by making a comparison with Wilson & Solomon's analytical solution for a constant wall temperature (Wilson and Solomon, 1986). The thermal properties and base case parameters used in this comparison and in the subsequent analyses are shown in Table 4. In this comparison, it is assumed that conduction in the void is the only mode of heat transfer. The results are shown in Figure 15. This figure shows that the numerical solution is in reasonable agreement with the analytical solution for the base cases of interest (error of less than 10%). However, the error in the freezing constant for the cases with a wall void is larger (as much as 26%) than that with a center void. Figure 15 also shows that for base case conditions (St= 1.78 and 6.20 for Li and LiF, respectively) freezing constants for the wall void cases are approximately 34% and 19% of those for the center void cases. It is also shown that in the center void cases, Li and LiF have similar freezing constants, while in the wall void cases, LiF has a much lower freezing constant than that for Li, due to the larger void size of the former.

3.2.2 Freezing with Radiative Heat Transfer Boundary

The results of the freezing process with a radiative boundary condition are shown in Figures (16) through (18). Figure 16 shows that because the difference between the fusion temperature of Li (454 K) and the ambient (250 K) is relatively small and the thermal conductivity of Li is high, the freezing velocity for Li is almost constant. Conversely, because of the high fusion temperature of the LiF (1122 K) the temperature differential between the wall and ambient is approximately three times higher than in the case of Li, the freezing velocity of LiF is an order of magnitude higher. As the void size and crust thickness increase (both have lower thermal conductance than in the case of Li), the wall temperature decreases, resulting in a lower freezing velocity. This velocity is still an order of magnitude higher than that of Li (see Figure 16). This figure also shows that while the location of the void negligibly affects the freezing of the Li, it significantly influences the freezing velocity of LiF. As Figure 16 demonstrates, the freezing velocity for LiF, in case of a void forming at the wall, is about half that in the case where the void forms at the opposite adiabatic boundary. Figures 17 and 18 shows that because the LiF fusion temperature is much higher and void size is larger that those of Li, the ratio of conduction to radiation heat transfer through the LiF void is about four orders of magnitude lower than that for Li.

3.2.3 Thaw with Radiative Heat Transfer Boundary

The thaw process with a radiative boundary has a practical importance to space applications. The thaw process begins by assuming that a shrinkage void has formed either at wall or at the center of a system (or adiabatic boundary). The size of void is equal to that resulting from a complete solidification of liquid initially filled in a container, having a width L. In the base case, void thickness is 1.23 mm for Li and 11.6 mm for LiF.

The results are delineated in Figures 19 through 21. At the start of the thaw process of Li the heat transfer to the frozen crust is limited by the heat transfer through the void. This causes a large fraction of the heat input to be absorbed in the wall, resulting in a higher wall temperature (see Figure 19). However, as the Li melts, reducing the size of the void, a smaller fraction of the heat input is absorbed in the wall, causing the rise rate of the wall temperature to decrease. Eventually, when the heat transferring through the void equals the heat input, the wall temperature reaches a maximum value (see Figure 20). Beyond this point, the wall temperature decreases with time as the void size becomes very small. Subsequently, the heat transfer rate through the void exceeds the heat input; the difference is compensated for by the decrease in the sensible heat of the wall. The variation of the heat flux ratio inside the void during the thaw process is delineated in Figure 21.

In the case of LiF, because the initial void size is about an order of magnitude larger than that of Li, the thaw of LiF is always limited by the heat transfer through the void. This causes the wall temperature to rise monotonically as the thaw process continues. As shown in Figure 19, the maximum wall temperature during Li thaw (990 K) is about two-thirds that of LiF (1336 K). The peak wall temperature during Li thaw is much lower than the melting temperature of the wall materials in the SP-100 (PWC-11), suggesting that a development of hot spots is unlikely during thaw in the SP-100 system.

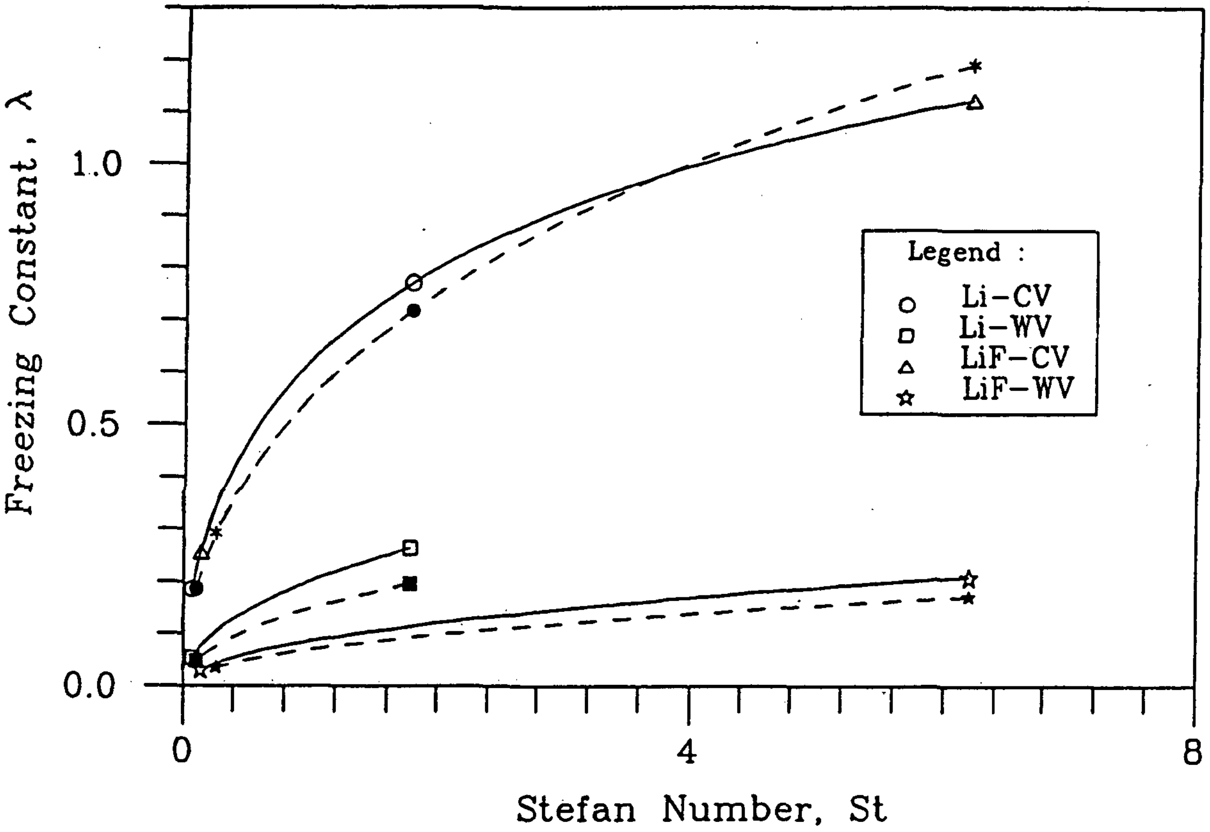


Figure 15. Variation of Freezing Constant, λ, with Stefan Number

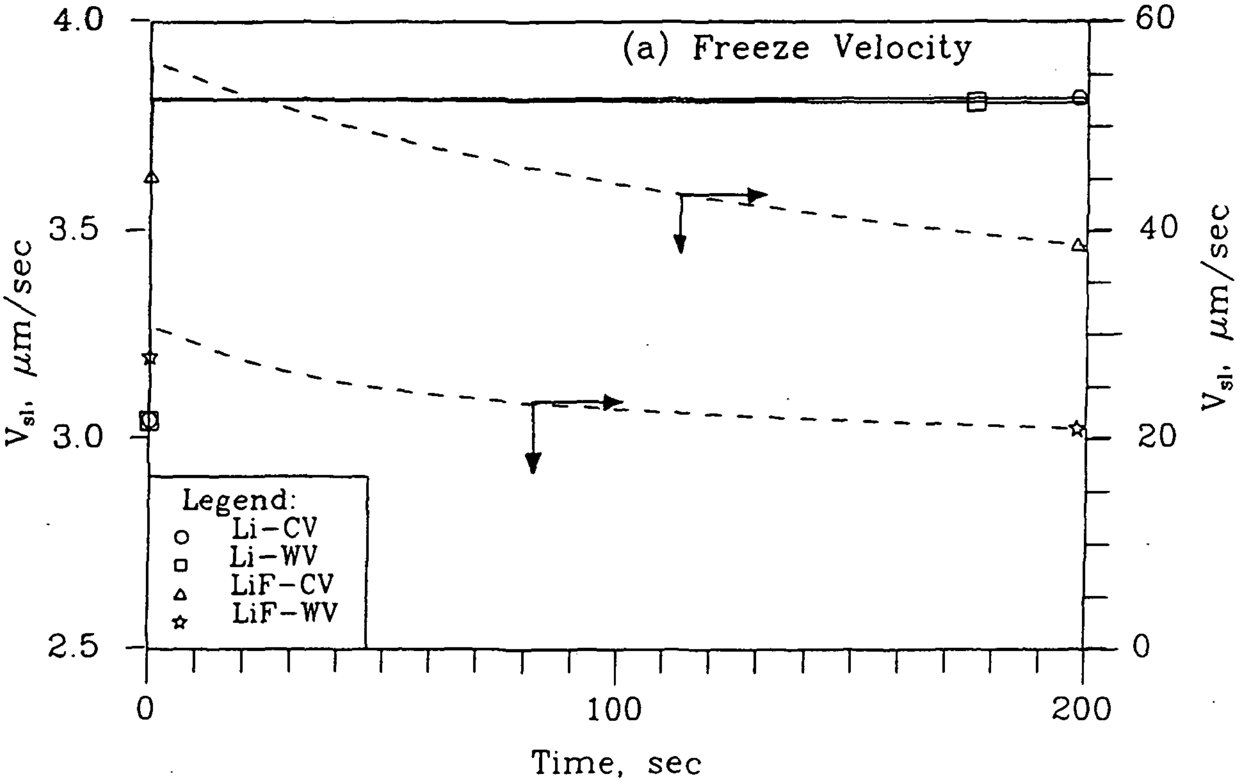


Figure 16. Calculated Solid-Liquid Interface Velocity During Freeze of Li and LiF (Radiative BC, Ta = 250 K)

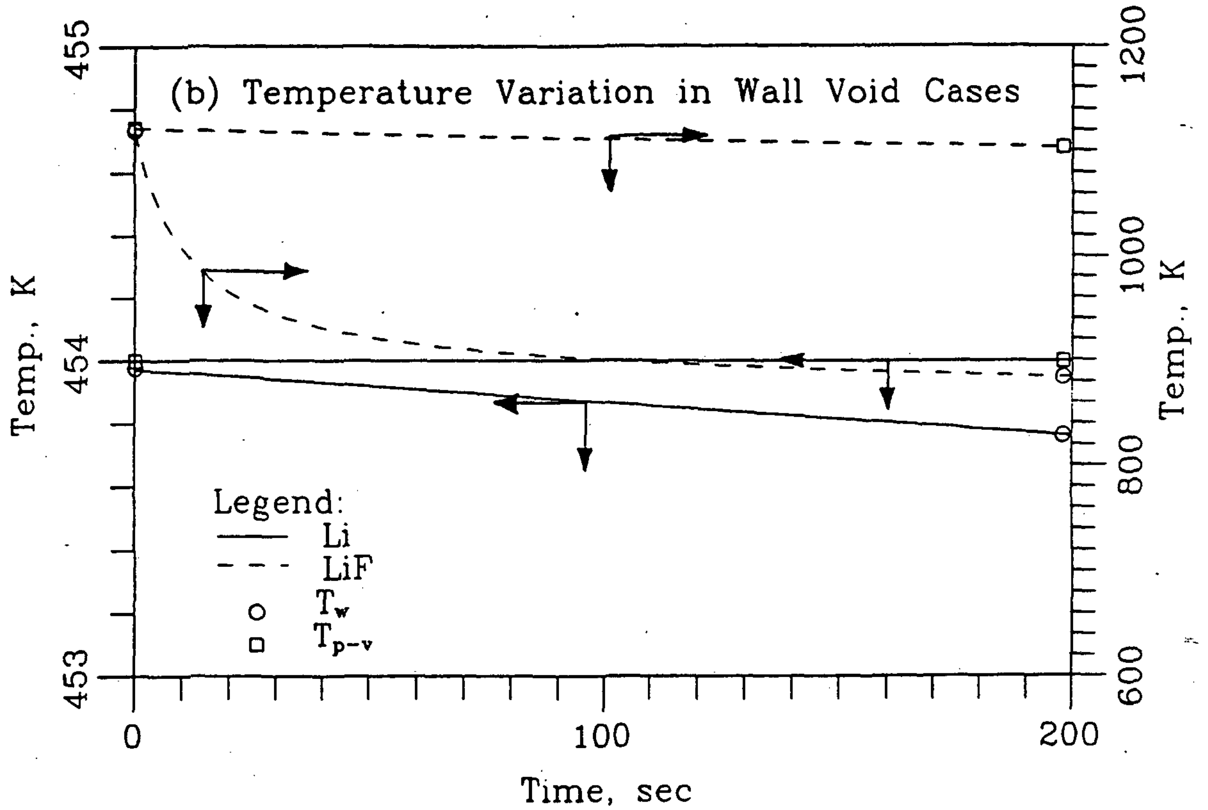


Figure 17. Temperature Variation in Wall Void Cases During Freeze of Li and LiF (Radiative BC, Ta = 250 K)

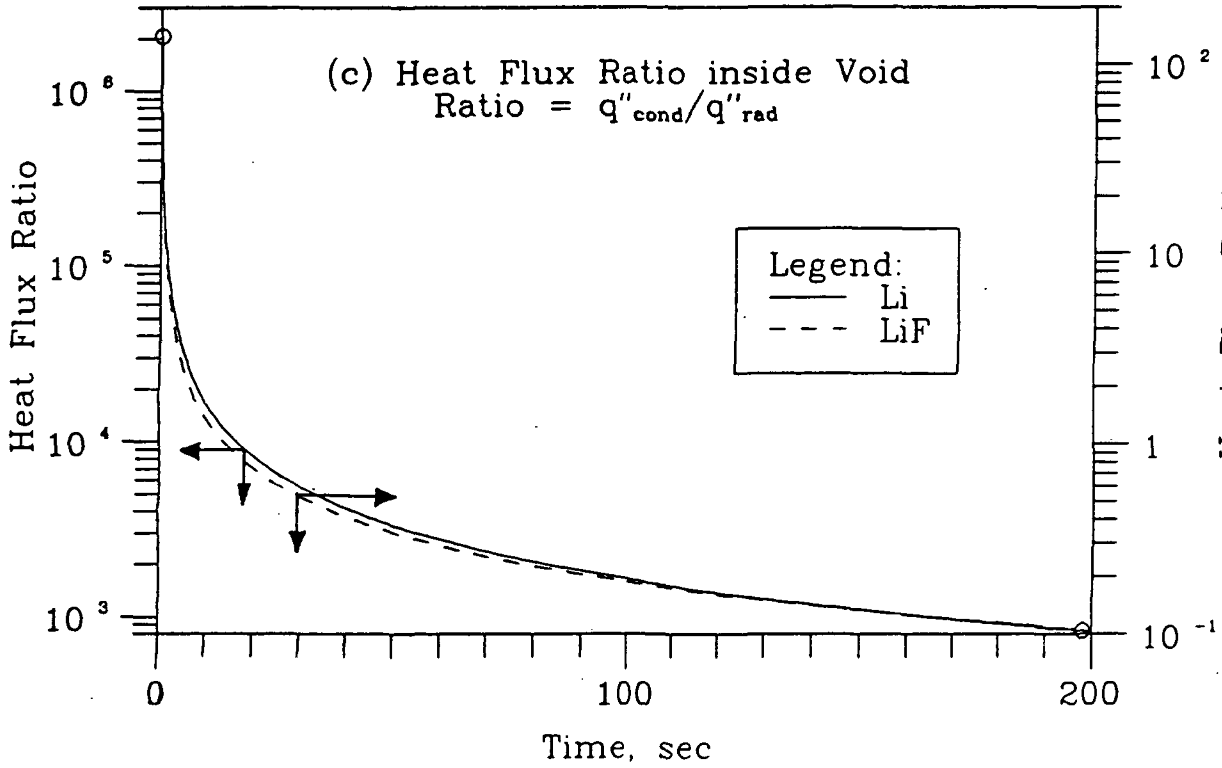


Figure 18. Calculated Heat Flux Ratio inside Void During Freeze of Li and LiF (Radiative BC, Ta = 250 K)

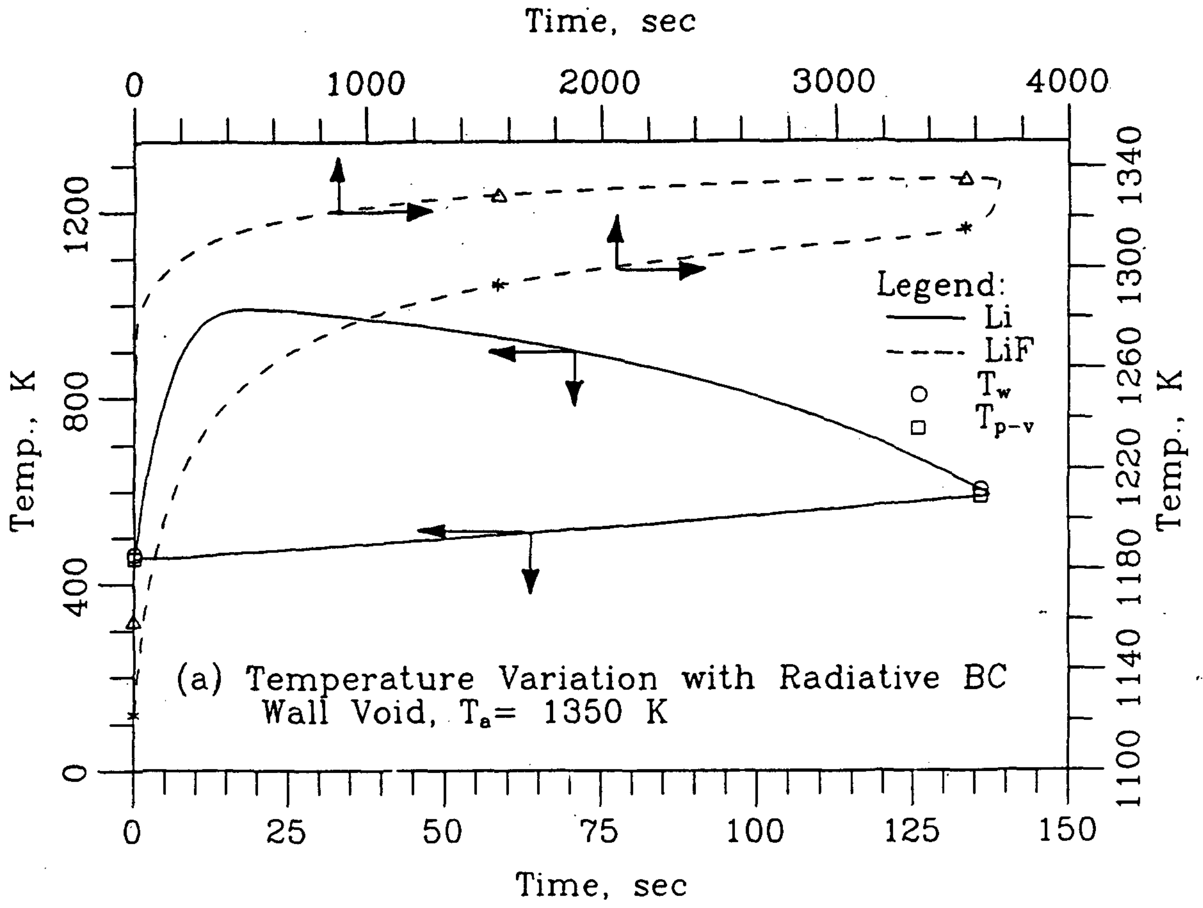


Figure 19. Temperature Variation, with Radiative BC Wall Void, Ta = 1350 K, During Thaw Process of Li and LiF

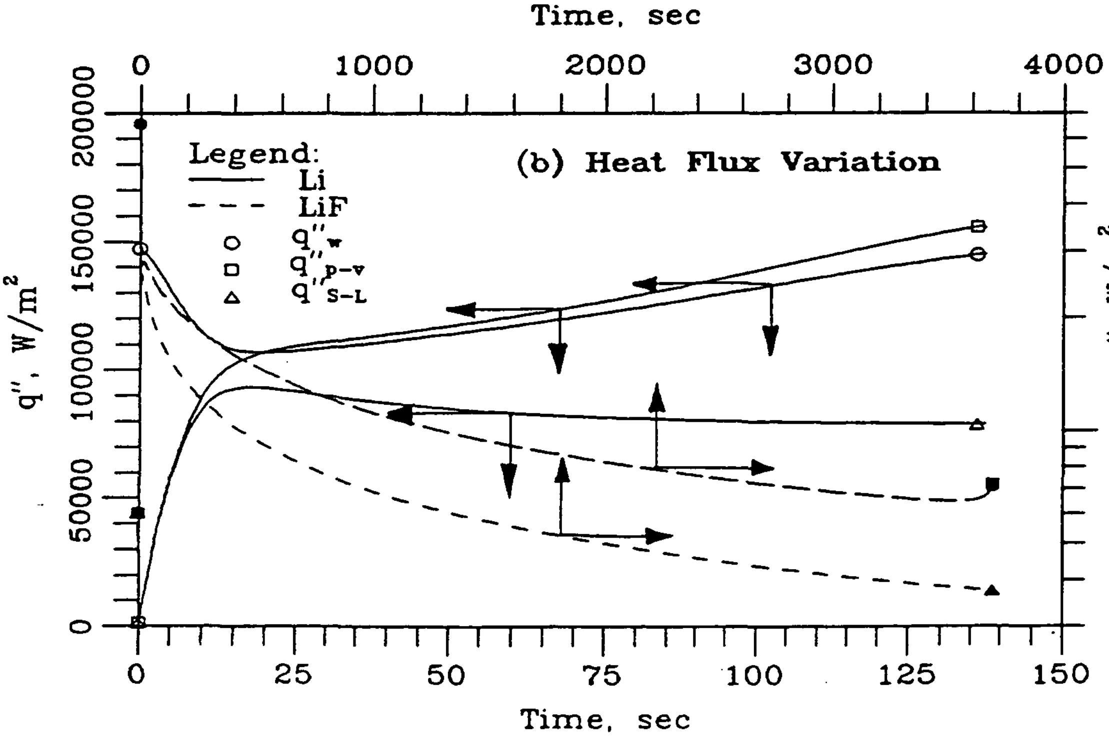


Figure 20. Heat Flux Variation During Thaw Process of Li and LiF (Wall Void, Radiative BC, Ta = 1,350 K)

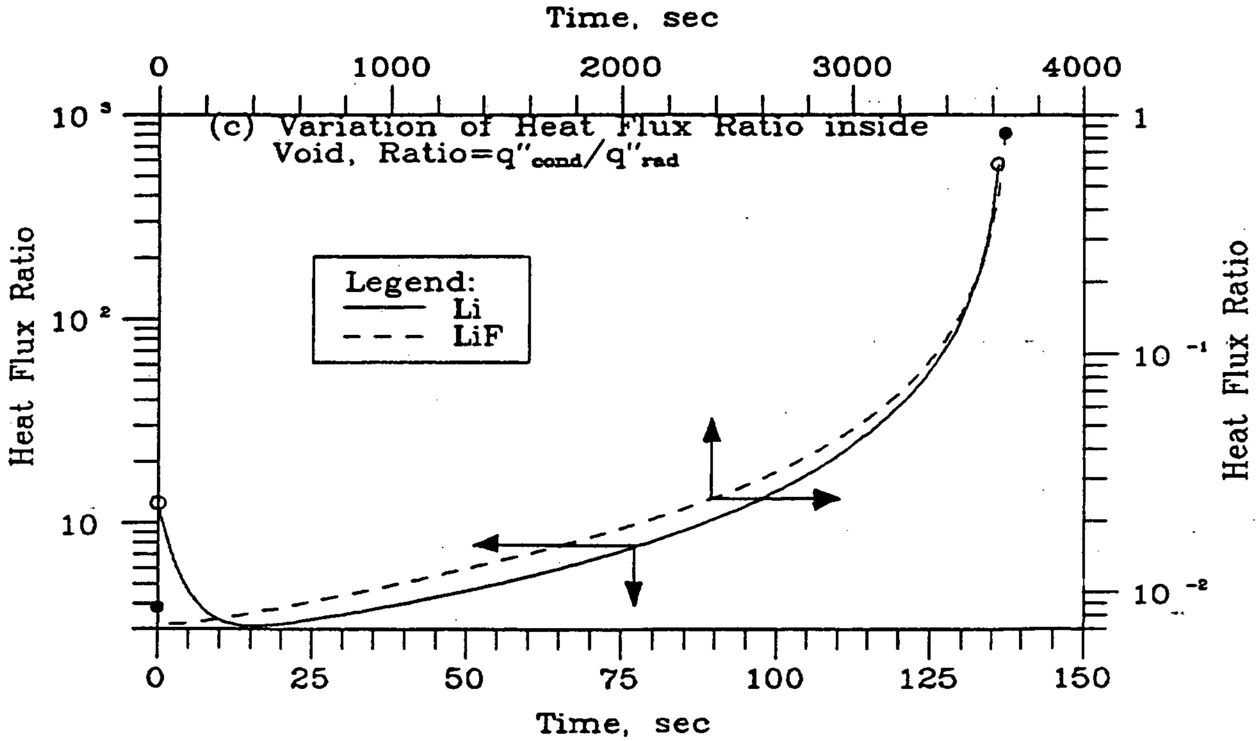


Figure 21. Variation of Heat Flux Ratio Inside Void During Thaw Process of Li and LiF (Wall Void, Radiative BC, Ta = 1350 K)

3.3 NOMENCLATURE

English

C_p : Heat capacity (J/kg.K)

H_{sl} : Latent heat of fusion (J/kg)

h_r : Radiative heat transfer coefficient (W/m^2.K)

h_{vc} : Thermal conductance of the void (W/m^2.K)

k : Thermal conductivity (J/m.K)

L : Coolant channel width (m)

st : Stefan number ()

T : Temperature (K)

T_1,T_2: Temperatures at the center of the wall and developing phase, respectively

t : Time (s)

t_c : Thickness of the wall (m)

v : Material velocity due to volume shrinkage (m/s)

Xr : Location of shrinkage void/liquid (or solid) interface (m)

Greek

α : Thermal diffusivity (m^2/s)

Γ : $\rho_s/\rho_L - 1$

δ : Solid crust thickness, (m)

δ_v : Void thickness, (m)

ε : Emissivity

ρ : Density (kg/m^3)

σ : Stefan-Boltzman constant (5.67 x 10 $^{-8}$ W/m^2.K)

Subscripts

c : Wall region
d : Developing phase
f : Fusion
L : Liquid
s : Solid
i : Initial
v : Void
01 : Between wall and heat source (sink)
12 : Between the centers of wall and developing phase
1 : Within the half of the wall
2 : Within the half of the developing phase
3 : At the wall-void interface
4 : At the PCM-void interface

4. SUMMARY AND CONCLUSIONS

It is shown that the formation of He gas voids either by HN or HTN during the cooldown process is unlikely. However, the formation of Nb-hydride and/or Nb-nitride in the SP-100 PHTS could increase the potential of Li-dewetting of the walls during cooldown, and hence the HTN of He bubbles. Analysis shows that HN of He bubble would occur by normal segregation during the freezing process of Li. Because of its high heat capacity and high latent heat of fusion as well as the low cooling rate of the wall, Li freezing is very slow. The results indicate that the total volume of He bubbles is insignificant (~2 %) compared to that due to volume shrinkage of Li during freezing. Hence the former is expected to play a minor role in relieving the stress in the walls during the restart of the SP-100 system. Depending on where heat is applied relative to the location of the shrinkage void, stress may or may not be induced in the PHTS structure during the restart of SP-100 system.

The effects of shrinkage void forming during freezing of lithium and lithium-fluoride on subsequent thaw processes are investigated using a numerical scheme that is based on a single (solid/liquid) cell approach. Results show that the formation or presence of a shrinkage void at the wall-PCM interface substantially reduces the S-L interface velocity while voids, forming at the center of the coolant ducts, do not influence the freeze and thaw processes.

Results also show that the presence of a void at the wall causes a substantial rise in the wall temperature during thaw. However, in the case of Li, the maximum wall temperature was much lower than the melting temperature of PWC-11, which is used as the structure material in the SP-100 system. Hence, it is concluded that a formation of hot spots is unlikely during the startup or restart of the SP-100 system.

ACKNOWLEDGEMENTS

This research is supported by NASA Lewis Research Center under research Grant No. NAG. 3-1045 to the University of New Mexico's Institute for Space Nuclear Power Studies.

REFERENCES

Alefeld, G. and J. Volki, Eds. (1978), *Hydrogen in Metals I*, Springer-Verlag, N.Y. pp 329-330.

Brick, R. M., A. W. Pense, and R. B. Gordon (1977), *Structure and Properties of Engineering Materials*, 4th ed., McGraw-Hill.

Cahn, R. W. and P. Haasen, Eds. (1983), *Physical Metallurgy*, 3rd ed., North-Holland Physics Publ., N.Y.

Chalmers, B. (1964), *Principles of Solidification*, John Wiley & Sons, New York.

Cole, R. (1974), "Boiling Nucleation," *Advances in Heat Transfer*, Vol. 10, pp 85-166, Academic Press, New York.

Hoffman, N. (1990), *Private Communication*, ETEC Div., Rockwell International Corp., Canoga Park, CA, April, 1990.

Hunter, L. W. and J. R. Kuttler (1983), "Cooling of a Slab with Thermal Contraction and Progressive Loss of Contact With a Cold Surface," *Trans. ASME*, 105, November 1983.

Hunter, L. W. and J. R. Kuttler (1989), "The Enthalpy Method for Heat Conduction Problems with Moving Boundaries," *J. of Heat Transfer*, **111**:239-242.

McGhee, J., M. S. El-Genk, and R. B. Rothrock (1989), "An Assessment of Helium Gas in SP-100 Class Reactors," *24th IECEC Conf.*, Crystal City, Washington, D.C., Aug. 1989.

Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow*, Hemisphere Pub. Co., Washington, D.C.

Shamsunder, N. and E. M. Sparrow (1975). "Analysis of Multidimensional Conduction Phase Change Via the Enthalpy Model," *J. of Heat Transfer*, August 1975.

Slotnick, H. et al. (1965), *The Solubility of Helium in Lithium and Potassium*, PWAC - 380, Pratt & Whitney Aircraft Co., Middletown, Conn., 24 February 1965.

van Stralen, Sjoerd and R. Cole, (1979), *Boiling Phenomena*, Vol. I & II, Hemisphere, Washington, D.C., pp 259 - 260.

Wilson, D. G. , and A. D. Solomon (1986). "A Stefan-Type Problem with Void Formation and its Explicit Solution," *IMA J. of Applied Mathematics*, 37, [1].

Yang, J. Y. and M. S. El-Genk (1991), "Mechanisms of Void Formation During Cooldown and Freezing of Lithium in SP-100 Type Systems," *Proc. of the 8th Symposium on Space Nuclear Power Systems*, Albuquerque, NM, 6-10 January 1991.

Appendix A-1

List of Computer Program COPHASE.FOR

```
      PROGRAM COPHASE
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON  /INPT1/ NT, TIME, NX, XL, HINIT, TINIT, TA, TF, TWALL,
     1         HSF, HLF, CONDSF, CONDLF, RHOSF, RHOLF, EMIT
      COMMON  /CONVE1/ EPS1, EPS2, EPS3
      COMMON  /OPTIN/ MLB, MRB, ILB, IRB, ICOMP, IENT, ITEM, SLOPT,
     &        PRTIME(20)
      COMMON  /MESHX/ XI(0:500), DX(500), DT
      COMMON  /DIMSV/ HX(0:500, 2), TX(0:500, 2), DELTA(0:2)
      COMMON  /DIMLV/ THETA(0:500, 2), PAI(0:500, 2), THEC(0:500)
      COMMON  /LOCPR/ RHOX(0:500,2), CPX(0:500,2), CONDX(0:500,2)
      COMMON  /TEMPO/ STHETA(0:500), STHEC(0:500), SPAI(0:500),
     &        SRHOX(0:500), SCPX(0:500), SCONDX(0:500), SHX(0:500),
     &        STX(0:500)
      COMMON  /THERP/ RHO(50), CP(50), COND(50), TEM(50), ENT(50)
      COMMON/PROP/HL,HS,CL,CS,CONL,CONS,RHL,RHS,TFF

      NAMELIST /OPTION/ MLB, MRB, ILB, IRB, ICOMP, IENT, ITEM,
     &        SLOPT, NPRT, NpStp
C ****  MLB = 1 FOR MOVING LEFT BOUNDARY
C       MRB = 1 FOR MOVING RIGHT BOUNDARY
C       ILB = INDEX FOR LEFT BOUNDARY CONDITIONS
C             0 = ADIABATIC BOUNDARY
C             1 = ISOTHERMAL WALL (TIME INDEPENDENT)
C             2 = ISOFLUX WALL (TIME INDEPENDENT)
C             3 = CONVECTIVE BOUNDARY
C             4 = RADIATIVE BOUNDARY
C       IRB = INDEX FOR LEFT BOUNDARY CONDITIONS
C             0 = ADIABATIC BOUNDARY
C             1 = ISOTHERMAL WALL (TIME INDEPENDENT)
C             2 = ISOFLUX WALL (TIME INDEPENDENT)
C             3 = CONVECTIVE BOUNDARY
C             4 = RADIATIVE BOUNDARY
CCCCCCC*** 9/26/1990 OPTION FOR IRB = 1 - 4 ARE NOT USED
C
C       ICOMP = 1 FOR COMPARISON WITH ANALYTICAL RESULT IF AVAILABLE
C       IENT =  1 FOR UNIFORM INITIAL ENTHALPY DISTRIBUTION
C               2 FOR NON-UNIFORM INITIAL ENTHALPY DISTRIBUTION
C                 (INPUT REQUIRED)
C       ITEM =  1 FOR UNIFORM INITIAL TEMPERATURE DISTRIBUTION
C               2 FOR NON-UNIFORM INITIAL TEMPERATURE DISTRIBUTION
C                 (INPUT REQUIRED)
C       SLOPT = OPTION FOR LOCATION OF INTERFACE
C       THREE DIFFERENT POSITION CAN BE CONSIDERED AS THE INTERFACE
C             = 0. FOR DELTA BASED ON THETA(X,T) = 0
C             = .5 FOR DELTA BASED ON THETA(X,T) = 1/2
C             = 1. FOR DELTA BASED ON THETA(X,T) = 1
C       PRTIME(I) = PRINT OUT RESULTS AT TIME PRTIME(I) (SEC)
C               IF NO INPUT IS MADE FOR THIS VARIABLE,
C               PRINT IS MADE ONCE AT THE TIME SPECIFIED BY
C               INPUT VARIABLE 'TIME' IN THE NAMELIST 'INPUT1'.

       NAMELIST /INPUT1/ NT, TIME, NX, XL, HINIT, TINIT, TF, HSF, HLF,
     1          CONDSF, CONDLF, RHOSF, RHOLF, EMIT, TWALL, HWALL, TA,
     2          CPS, CPL
       NAMELIST /CONVER/ EPS1, EPS2, EPS3
       OPEN(10, ERR=9999, FILE='COPHASE.INP', STATUS='OLD')
       OPEN(11, ERR=9999, FILE='COPHASE.VEL', STATUS='UNKNOWN')
       OPEN(12, ERR=9999, FILE='COPHASE.ENT', STATUS='UNKNOWN')
       OPEN(13, ERR=9999, FILE='COPHASE.TEM', STATUS='UNKNOWN')
       OPEN(14, ERR=9999, FILE='COPHASE.CRS', STATUS='UNKNOWN')

C ****  DEFAULT VALUES FOR OPTIONS
       DATA MLB/0/, MRB/0/, ILB/1/, IRB/0/, SLOPT/0./, IFLAG/0/
```

```
C ****  CONSTANTS
        DATA PHI/3.1415926/, BOLTZ/5.67D-08/
C ****  READ INPUT
        READ(10,OPTION)
        READ(10,INPUT1)
        READ(10,CONVER)
        HL = HLF
        HS = HSF
        CL = CPL
        CS = CPS
        CONL = CONDLF
        CONS = CONDSF
        RHL = RHOLF
        RHS = RHOSF
        TFF = TF

C ****  IDENTIFY TIME FOR PRINTOUT
        DPRT = TIME/DFLOAT(NPRT)
        PRTIME(1) = 0.
        DO 5 I=1,NPRT
5       PRTIME(I+1) = PRTIME(I)+DPRT
C ****  MESH GENERATION:
C       NX = NUMBER OF NODES
        NX1 = NX - 1
        DX1 = XL/DFLOAT(NX)
        DX101 = DX1 * 0.1

        XI(0) = 0.
        XI(NX) = XL

        DO 10 I=1,NX
        DX(I) = DX1
10      XI(I) = XI(I-1) + DX(I)

C ****  THE TIME STEP SIZE: UNIFORM
        DT = TIME/DFLOAT(NT)

C ****  PREPARATION FOR INITIAL CONDITIONS

        DELTA(0) = 0.
        DELTA(1) = 0.
        DELTA(2) = 0.
        DDELTADT = 0.
        SDELTA = 0.
        HF = HLF - HSF
        HO = HWALL
        SGAM = 1. - RHOLF/RHOSF
        IF(MLB .EQ.1) SGAM = -SGAM
        IF(MLB .EQ.0) SGAM = 0.

C ****  INITIAL TEMPERATURE, HEAT CAPACITY, THERMAL CONDUCTIVITY, &
C       DENSITY
C       DIMENSIONLESS ENTHALPY, THETA(I,M), TEMPERATURE, PAI(I,M)
C       AND ENTHALPY CORRECTION FACTOR, THEC(I)

        DO 20 J = 1,2
        DO 20 I = 1, NX
        HX(I,J) = HINIT
        CALL PROPERTY(HX(I,J),TX(I,J),CPX(I,J),RHOX(I,J),CONDX(I,J))
        THETA(I,J) = (HX(I,J) - HSF)/HF
        THEC(I) = - THETA(I,J)
        IF(THEC(I) .GT. 0.) THEC(I) = 0.
        IF(THEC(I) .LT. -1.) THEC(I) = -1.
        PAI(I,J) = THETA(I,J) + THEC(I)
20      CONTINUE
```

```
        RHOIN=RHOX(1,1)
        WRITE(*,1001) SNGL(RHOX(1,1)),SNGL(RHOX(2,1)),
     &          SNGL(RHOX(NX-1,1)),SNGL(RHOX(NX,1))
1001    FORMAT(/5X,'DENSITIES',E12.5,2X,E12.5,2X,E12.5,2X,E12.5)
C-----------------------------------------------------------------------
C ****  INITIAL VALUES FOR  CONSTANT TWALL< TF, PAIO = CONSTANT.
C-----------------------------------------------------------------------
        IF(ILB .EQ. 1) THEN

                CALL PROPERTY(HWALL,TWALL,CPXO,RHOXO,CONDXO)
                THETAO = (HWALL - HSF)/HF
                PAIO   = CPXO*(TWALL - TF)/HF
                THECO  = PAIO - THETAO
                HO = HWALL
                QOLD = 0.
        WRITE(*,21) SNGL(TWALL),SNGL(THETAO), SNGL(PAIO)

C-----------------------------------------------------------------------
C ****  INITIAL VALUES FOR RADIATIVE BOUNDARY
C-----------------------------------------------------------------------

        ELSEIF(ILB .EQ. 4) THEN
                THETOLD  = THETA(1,1)
                TWOLD    = TX(1,1)
                PAIOLD = PAI(1,1)
                HROLD = EMIT*BOLTZ*(TINIT**2. + TA**2.)*(TINIT + TA)
                QOLD = HROLD*(TINIT - TA)
                STE = CPX(1,1)*(TF - TA)/HF
                TA2 = TA**2.
        WRITE(*,22) SNGL(TWOLD),SNGL(THETOLD), SNGL(PAIOLD)
        ENDIF

C-----------------------------------------------------------------------
C ****  TOTAL SYSTEM ENERGY  [JOULE]
C-----------------------------------------------------------------------
        ETOT = 0.
        DO 25 I = 1, NX
25      ETOT = ETOT + RHOX(I,1)*HX(I,1)*DX(I)
        ETOTOLD = ETOT
        ETOTINIT = ETOT

        QTOT = 0.
        WRITE(12,26) SNGL(TF),SNGL(HF),SNGL(HLF),SNGL(HSF),
     &          SNGL(CPL),SNGL(CPS),
     &          SNGL(CONL),SNGL(CONS),SNGL(RHOLF),SNGL(RHOSF)
        WRITE(*,26) SNGL(TF),SNGL(HF),SNGL(HLF),SNGL(HSF),
     &          SNGL(CPL),SNGL(CPS),
     &          SNGL(CONL),SNGL(CONS),SNGL(RHOLF),SNGL(RHOSF)

C-----------------------------------------------------------------------
C ****  THE BEGINNING OF TIME LOOP *********************************************
        DO 9000 MA = 1, NT

        M=2
C ****  STORE PREVIOUS STEP VALUES IN THE TEMPORARY STORAGE VARIABLES
        DO 30 I=1,NX
        SHX(I)     = HX(I,M-1)
        STHEC(I)   = THEC(I)
        STHETA(I)  = THETA(I,M-1)
        SPAI(I)    = PAI(I,M-1)
        SRHOX(I)   = RHOX(I,M-1)
        SCPX(I)    = CPX(I,M-1)
        SCONDX(I)  = CONDX(I,M-1)
30      CONTINUE
        SDELTA     = DELTA(M-1)
```

```
C-----------------------------------------------------------------------
C ****  OUTER ITERATION ---- DELTA ITERATION
C-----------------------------------------------------------------------
        IDITER = 0
        DELTA(1) = DELTA(0) + DDELTADT*DT/2.    !ASSUMED CRUST THICKNESS
1025    CONTINUE
        IDITER = IDITER +1
        DDELTA = DELTA(1) - DELTA(0)

C---------------------------------------------------------------------I
C       INNER ITERATION                                               !
C---------------------------------------------------------------------!
C                                                                     !
C       CONVERGENCE ARE CHECKED IN EACH TIME STEP BY;                 !
C       SUMDIF = SUM(F(I,M)-SF(I))*DX(I))**2.), I=ILEND,IREND         !
C       SUMASS = SUM(SF(I)*DX(I))**2.)), I=ILEND,IREND                !
C                                                                     !
C               AND IF (SUMDIF/SUMASS)<= EPS1,                        !
C                THE CONVERGENCE ARE ACHIEVED                         !
C                                                                     !
C---------------------------------------------------------------------I

        NITER = 0
        SUMQINDT = QOLD
1050    CONTINUE

        NITER = NITER + 1

        SUMDIF = 0.
        SUMASS = 0.

C-----------------------------------------------------------------------
C ****  ADJUSTMENTS OF THE SIZE AND LOCATION OF THE BOUNDARY NODES
C-----------------------------------------------------------------------
C
C ****  INITIAL LOCATION OF BOUNDARY NODES
        IF(MLB .EQ. 0 .AND. MRB .EQ. 0) THEN
                XLEND = 0.
                XREND = XL
                ILEND = 1
                IREND = NX
C
C ****  LOCATION AND SIZE OF MOVING LEFT BOUNDARY NODE; MLB = 1
C
        ELSEIF(MLB .EQ. 1) THEN
                XLEND = SGAM*DELTA(M-1)
                XREND = XL
                ILEND = 1

                IF(XLEND .LT. XI(1)) THEN
                        ILEND = 1
                        GO TO 60

                ELSEIF(XLEND .GT. XI(NX1)) THEN
                        ILEND = NX
                        GO TO 60

                ELSE
                DO 50 IL=2,NX
                IF(XLEND .GT. XI(IL-1) .AND. XLEND .LE. XI(IL)) THEN
                        ILEND = IL
                        GO TO 60
                ENDIF
50              CONTINUE
```

```
              ENDIF

60            DXI = XI(ILEND) - XLEND
              IF(DXI .LT. DX101) THEN
                      ILEND = IL +1
                      DXI = XI(ILEND) - XLEND
              ENDIF
              DX(ILEND) = DXI
C
C ****  LOCATION AND SIZE OF MOVING RIGHT BOUNDARY NODE: MRB = 1
C
      ELSEIF(MRB .EQ. 1) THEN
              XREND = XL - SGAM*DELTA(M-1)
              XLEND = 0.
              ILEND = 1

              IF(XREND .GT. XI(NX1)) THEN
                      IREND = NX
                      GO TO 62

              ELSEIF(XREND .LE. XI(1)) THEN
                      IREND = 1
                      GO TO 62
              ELSE

              DO 61 IL=1,NX1
              IF(XREND .GT. XI(IL-1) .AND. XLEND .LE. XI(IL)) THEN
                      IREND = IL
                      GO TO 62
              ENDIF
61            CONTINUE
              ENDIF

62            DXI = XREND - XI(IREND)
              IF(DXI .LT. DX101) THEN
                      IREND = IL - 1
                      DXI = XREND - XI(IREND)
              ENDIF
              DX(IREND) = DXI

      ENDIF

      K = ILEND
      L = IREND

      ILEND1 = ILEND + 1
      IREND1 = IREND - 1
      IREND2 = IREND - 2

      DO 35 I = ILEND, IREND
35    SUMASS = SUMASS + (HX(I,M-1)*DX(I))**2.

C-----------------------------------------------------------------------
C ****  PRODUCTION RUN: LEFT BOUNDARY NODE
C-----------------------------------------------------------------------
C       ILB = 1 FOR CONSTANT WALL TEMPERATURE AT LEFT BOUNDARY
C
      IF(ILB .EQ. 1) THEN

C     ALM = CPX(K-1,M-1)*DX(K-1)/CONDX(K-1,M-1)
      ALO = CPX(K,M-1)*DX(K)/CONDX(K,M-1)
      ALP = CPX(K+1,M-1)*DX(K+1)/CONDX(K+1,M-1)

      A1 = 2.*DT/(DX(K)*RHOX(K,M-1))
      A2 = 1./(ALO + ALP)
```

```
      A3 = 1./ALO
C     A4 =

c     CHECK = STHETA(K) + A1*(A2*PAI(K+1,M-1) + A3*PAIO)
c
c     IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
c     THETA(K,M) = CHECK
c     THEC(K) = -THETA(K,M)
c
c     ELSEIF(CHECK .GT. 1.) THEN
c     THEC(K) = -1.
c     THETA(K,M) = ( STHETA(K) + A1*(A2*PAI(K+1,M-1)
c    &         + A3*PAIO - (A2+A3)*THEC(K)) ) / (1.+A1*(A2+A3))
c
c     ELSE
c     THEC(K) = 0.
c     THETA(K,M) = ( STHETA(K) + A1*(A2*PAI(K+1,M-1)
c    &         + A3*PAIO - (A2+A3)*THEC(K)) ) / (1.+A1*(A2+A3))
c     ENDIF

      CHECK = STHETA(K)*srhox(k)/rhox(k,m-1)
     &         -(1.-srhox(k)/rhox(k,m-1))*Hsf/Hf
     &         + A1*(A2*PAI(K+1,M-1) + A3*PAIO)

      IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
      THETA(K,M) = CHECK
      THEC(K) = -THETA(K,M)

      ELSEIF(CHECK .GT. 1.) THEN
      THEC(K) = -1.
      THETA(K,M) = ( STHETA(K)*srhox(k)/rhox(k,m-1)
     &         -(1.-srhox(k)/rhox(k,m-1))*Hsf/Hf
     &          + A1*(A2*PAI(K+1,M-1)
     &         + A3*PAIO - (A2+A3)*THEC(K)) ) / (1.+A1*(A2+A3))

      ELSE
      THEC(K) = 0.
      THETA(K,M) = ( STHETA(K)*srhox(k)/rhox(k,m-1)
     &         -(1.-srhox(k)/rhox(k,m-1))*Hsf/Hf
     &          + A1*(A2*PAI(K+1,M-1)
     &         + A3*PAIO - (A2+A3)*THEC(K)) ) / (1.+A1*(A2+A3))
      ENDIF

      HX(K,M) = HSF + THETA(K,M)*HF
      CALL PROPERTY(HX(K,M),TX(K,M),CPX(K,M),RHOX(K,M),CONDX(K,M))

      PAI(K,M) = THETA(K,M) + THEC(K)

      HDIF = ((HX(K,M) - HX(K,M-1))/HX(K,M-1))**2.
C---------------------------------------------------------------------
C     ILB = 2 FOR CONSTANT WALL HEAT FLUX BOUNDARY AT LEFT BOUNDARY
C---------------------------------------------------------------------
C
      ELSEIF(ILB .EQ. 2) THEN
      PRINT *,' '
      PRINT *,'  CONSTANT WALL HEAT FLUX BOUNDARY '

C---------------------------------------------------------------------
C     ILB = 4 FOR CONVECTIVETIVE BOUNDARY AT LEFT BOUNDARY
C---------------------------------------------------------------------
C
      ELSEIF(ILB .EQ. 3) THEN
      PRINT *,' '
      PRINT *,'  CONVECTIVE BOUNDARY '
```

```
C------------------------------------------------------------------------------------
C       ILB = 4 FOR RADIATIVE BOUNDARY AT LEFT BOUNDARY
C------------------------------------------------------------------------------------
C
        ELSEIF(ILB .EQ. 4) THEN

        ALPA = BOLTZ*EMIT*DX(K)/CONDX(K,M-1)/2.
        HRR = BOLTZ*EMIT*(TX(K,M-1)**2. + TA2)*(TX(K,M-1) + TA)
     &        /(1. + 4.*ALPA*TX(K,M-1)**3.)
        STE = CPX(K,M-1)*(TF - TA)/HF

C       ALM = CPX(K-1,M-1)*DX(K-1)/CONDX(K-1,M-1)
        ALO = CPX(K,M-1)*DX(K)/CONDX(K,M-1)
        ALP = CPX(K+1,M-1)*DX(K+1)/CONDX(K+1,M-1)

        A1 = 2.*DT/(DX(K)*RHOX(K,M-1))
        A2 = 1./(ALO + ALP)
        A3 = HRR/CPX(K,M-1)/2.
C       A4 =

c       CHECK = STHETA(K) + A1*(A2*PAI(K+1,M-1) - A3*Ste)
c
c       IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
c       THETA(K,M) = CHECK
c       THEC(K) = -THETA(K,M)
c
c       ELSEIF(CHECK .GT. 1.) THEN
c       THEC(K) = -1.
c       THETA(K,M) = ( STHETA(K) + A1*(A2*PAI(K+1,M-1) - (A2+A3)
c     &          * THEC(K) - A3*Ste) )/(1.+ A1*(A2+A3))
c       ELSE
c       THEC(K) = 0.
c       THETA(K,M) = ( STHETA(K) + A1*(A2*PAI(K+1,M-1) - (A2+A3)
c     &          * THEC(K) - A3*Ste) )/(1.+ A1*(A2+A3))
c       ENDIF
c
        CHECK = STHETA(K)*srhox(k)/rhox(k,m-1)
     &          -(1.-srhox(k)/rhox(k,m-1))*Hsf/Hf
     &           + A1*(A2*PAI(K+1,M-1) - A3*Ste)

        IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
        THETA(K,M) = CHECK
        THEC(K) = -THETA(K,M)

        ELSEIF(CHECK .GT. 1.) THEN
        THEC(K) = -1.
        THETA(K,M) = ( STHETA(K)*srhox(k)/rhox(k,m-1)
     &          -(1.-srhox(k)/rhox(k,m-1))*Hsf/Hf
     &           + A1*(A2*PAI(K+1,M-1) - (A2+A3)
     &          * THEC(K) - A3*Ste) )/(1.+ A1*(A2+A3))
        ELSE
        THEC(K) = 0.
        THETA(K,M) = ( STHETA(K)*srhox(k)/rhox(k,m-1)
     &          -(1.-srhox(k)/rhox(k,m-1))*Hsf/Hf
     &           + A1*(A2*PAI(K+1,M-1) - (A2+A3)
     &          * THEC(K) - A3*Ste) )/(1.+ A1*(A2+A3))
        ENDIF

        HX(K,M) = HSF + THETA(K,M)*HF
        CALL PROPERTY(HX(K,M),TX(K,M),CPX(K,M),RHOX(K,M),CONDX(K,M))

        PAI(K, M) = THETA(K, M) + THEC(K)

        ENDIF
C ***************** OPTIONS FOR OTHER BOUNDARY CONDITIONS **************
```

```
      HDIF = ((HX(K,M) - HX(K,M-1))/HX(K,M-1))**2.

      SUMDIF = SUMDIF + ((HX(K,M) - HX(K,M-1))*DX(K))**2.

      HRNEW = HR
C
C-----------------------------------------------------------------------
C ****  MAIN RUN: INTERIOR NODES
C-----------------------------------------------------------------------

      DO 200 I = ILEND1, IREND1

      ALM = CPX(I-1,M-1)*DX(I-1)/CONDX(I-1,M-1)
      ALO = CPX(I,M-1)*DX(I)/CONDX(I,M-1)
      ALP = CPX(I+1,M-1)*DX(I+1)/CONDX(I+1,M-1)

      A1 = 2.*DT/(DX(I)*RHOX(I,M-1))
      A2 = 1./(ALO + ALP)
      A3 = 1./(ALO + ALM)
C     A4 =
c     CHECK = STHETA(I) + A1*(A2*PAI(I+1,M-1) + A3*PAI(I-1,M))
c
c     IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
c     THETA(I,M) = CHECK
c     THEC(I) = -THETA(I,M)
c
c     ELSEIF(CHECK .GT. 1.) THEN
c     THEC(I) = -1.
c     THETA(I,M) = (STHETA(I) + A1*(A2*PAI(I+1,M-1) + A3*PAI(I-1,M))
c    &         - A1*(A2+A3)*THEC(I))/(1. + A1*(A2+A3))
c
c     ELSE
c     THEC(I) = 0.
c     THETA(I,M) = (STHETA(I) + A1*(A2*PAI(I+1,M-1) + A3*PAI(I-1,M))
c    &         - A1*(A2+A3)*THEC(I))/(1. + A1*(A2+A3))
c     ENDIF
c
      CHECK = STHETA(I)*srhox(i)/rhox(i,m-1)
     &        -(1.-srhox(i)/rhox(i,m-1))*Hsf/Hf
     &         + A1*(A2*PAI(I+1,M-1) + A3*PAI(I-1,M))

      IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
      THETA(I,M) = CHECK
      THEC(I) = -THETA(I,M)

      ELSEIF(CHECK .GT. 1.) THEN
      THEC(I) = -1.
      THETA(I,M) = (STHETA(I)*srhox(i)/rhox(i,m-1)
     &        -(1.-srhox(i)/rhox(i,m-1))*Hsf/Hf
     &         + A1*(A2*PAI(I+1,M-1) + A3*PAI(I-1,M))
     &        - A1*(A2+A3)*THEC(I))/(1. + A1*(A2+A3))

      ELSE
      THEC(I) = 0.
      THETA(I,M) = (STHETA(I)*srhox(i)/rhox(i,m-1)
     &        -(1.-srhox(i)/rhox(i,m-1))*Hsf/Hf
     &         + A1*(A2*PAI(I+1,M-1) + A3*PAI(I-1,M))
     &        - A1*(A2+A3)*THEC(I))/(1. + A1*(A2+A3))
      ENDIF


      HX(I,M) = HSF + THETA(I,M)*HF
      CALL PROPERTY(HX(I,M),TX(I,M),CPX(I,M),RHOX(I,M),CONDX(I,M))
      PAI(I,M) = THETA(I,M) + THEC(I)
```

```
C ****  CHECK CONVERGENCE FOR THETA OR THEC WHICH EVER IS CONVENIENT
        HDIF = ((HX(I,M) - HX(I,M-1))/HX(I,M-1))**2.

        SUMDIF = SUMDIF + ((HX(I,M) - HX(I,M-1))*DX(I))**2.

200     CONTINUE

C-----------------------------------------------------------------------
C ****  CLOSING RUN: RIGHT BOUNDARY NODE (IDIABATIC BC ONLY)
C-----------------------------------------------------------------------

        DX01 = DX(L) + DX(L-1)

        ALM = CPX(L-1,M-1)*DX(L-1)/CONDX(L-1,M-1)
        ALO = CPX(L,M-1)*DX(L)/CONDX(L,M-1)
C       ALP = CPX(L+1,M-1)*DX(L+1)/CONDX(L+1,M-1)

        A1 = 2.*DT/(DX(L)*RHOX(L,M-1))
        A2 = 1./(ALO + ALO)
        A3 = 1./(ALO + ALM)
C       A4 =

c       CHECK = STHETA(L) + A1*A3*PAI(L-1,M) - A4*THETA(L-1,M)
c
c       IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
c       THETA(L,M) = CHECK
c       THEC(L) = -THETA(L,M)
c
c       ELSEIF(CHECK .GT. 1.) THEN
c       THEC(L) = -1.
c       THETA(L,M) = (STHETA(L) + A1*A3*(PAI(L-1,M) - THEC(L))
c     &          - A4*THETA(L-1,M))/(1. + A1*A3)
c
c       ELSE
c       THEC(L) = 0.
c       THETA(L,M) = (STHETA(L) + A1*A3*(PAI(L-1,M) - THEC(L))
c     &          - A4*THETA(L-1,M))/(1. + A1*A3)
c       ENDIF

        CHECK = STHETA(L)*srhox(L)/rhox(L,m-1)
     &          -(1.-srhox(L)/rhox(L,m-1))*Hsf/Hf
     &           + A1*A3*PAI(L-1,M) - A4*THETA(L-1,M)

        IF(CHECK .GE. 0. .AND. CHECK .LE. 1.) THEN
        THETA(L,M) = CHECK
        THEC(L) = -THETA(L,M)

        ELSEIF(CHECK .GT. 1.) THEN
        THEC(L) = -1.
        THETA(L,M) = (STHETA(L)*srhox(L)/rhox(L,m-1)
     &          -(1.-srhox(L)/rhox(L,m-1))*Hsf/Hf
     &           + A1*A3*(PAI(L-1,M) - THEC(L))
     &          - A4*THETA(L-1,M))/(1. + A1*A3)

        ELSE
        THEC(L) = 0.
        THETA(L,M) = (STHETA(L)*srhox(L)/rhox(L,m-1)
     &          -(1.-srhox(L)/rhox(L,m-1))*Hsf/Hf
     &           + A1*A3*(PAI(L-1,M) - THEC(L))
     &          - A4*THETA(L-1,M))/(1. + A1*A3)
        ENDIF

        HX(L,M) = HSF + THETA(L,M)*HF
        CALL PROPERTY(HX(L,M),TX(L,M),CPX(L,M),RHOX(L.M).CONDX(L.M))
```

```
      PAI(L, M) = THETA(L, M) + THEC(L)

C **** CHECK CONVERGENCE FOR THETA OR THEC WHICH EVER IS CONVENIENT

      HDIF = ((HX(L,M) - HX(L,M-1))/HX(L,M-1))**2.

      SUMDIF = SUMDIF + ((HX(L,M) - HX(L,M-1))*DX(L))**2.

C-----------------------------------------------------------------------
C **** LOCATION OF INTERFACE
C-----------------------------------------------------------------------
C **** First, project THETA at x = 0.
      hi = dx(1)
      h1 = dx(2)
      h2 = dx(3)
      hih1 = hi+h1
      h1h2 = h1 + h2
      h12hi = h1 + 2.*hi
      hi2h1h2 = hi + 2.*h1 + h2
      h22hi2h1 = h2 + 2.*h1 + 2.*hi
      theta(0,m) = theta(1,m)*(2.*hi + h1)*h22hi2h1/(hih1*hi2h1h2)
     &         - theta(2,m)*hi*h22hi2h1/(hih1*h1h2)
     &         + theta(3,m)*hi*h12hi/(hi2h1h2*h1h2)

c     if (delta(m-1) .le. 0. .and. theta(0,m) .ge. SLOPT) then
c     delta(m) = 0.
c     go to 550
c     endif

      ISL1 = 1
c     ilendm1 = ilend - 1
      DO 541 I = ILEND, IREND1
      IF(THETA(I,M) .GE. 0. .AND. THETA(I,M) .LT. 1.) THEN
      ISL1 = I
      DELTA(M) = XI(ISL1) - DX(ISL1)*THETA(ISL1,M)
C     GO TO 551
c     ELSEIF(THETA(I,M) .LE. SLOPT .AND. THETA(I+1,M) .GT. SLOPT) THEN
c     IF(THETA(I,M) .LE. SLOPT .AND. THETA(I+1,M) .GT. SLOPT) THEN
c     ISL1 = I
c     DELTA(M) = XI(ISL1) - DX(ISL1)/2. + 2.*(SLOPT - THETA(ISL1,M))*
c    &         (THETA(ISL1+1,M)-THETA(ISL1,M))/(DX(ISL1)+DX(ISL1+1))
c     GO TO 551
      ENDIF
541   CONTINUE
551   CONTINUE

      if(theta(irend,m) .le. slopt) then
      delta(m) = xrend
      print *,'Complete solidification of the system'
      print *,'Calculation Terminated'
      go to 9999
c     go to 545
      endif

c     if(isl1 .eq. 1) then
      DELTA(M) = XI(ISL1) - DX(ISL1)*THETA(ISL1,M)
c     ELSE
c     DELTA(M) = XI(ISL1-1) - DX(ISL1-1)/2.
c    &         + 2.*(SLOPT - THETA(ISL1-1,M))*
c    &         (THETA(ISL1,M)-THETA(ISL1-1,M))/(DX(ISL1)+DX(ISL1-1))
c     endif

C     sumtheta = 0.
C     Do 555 i = ilend, isl1
C555   sumtheta = sumtheta + theta(i,m)*dx(i)
```

```
C       avgtheta = sumtheta/(xi(isl1) - xlend)
C       delta(m) = (xi(isl1) - xlend)*(1.-avgtheta)

c       isl1 = 1
c       DO 540 I = ILEND, IREND1
c       IF(THETA(I,M) .LE. SLOPT .AND. THETA(I+1,M) .GT. SLOPT) THEN
c       ISL1 = I
c       go to 545
c       ENDIF
c540    CONTINUE
c
c545    ximh = xi(isl1) - dx(isl1)/2.
c       xiph = xi(isl1) + dx(isl1+1)/2.
c
c       if(isl1 .eq. 1) then
c       xim3h = 0.
c       else
c       xim3h = xi(isl1) - dx(isl1) - dx(isl1-1)/2.
c       endif
c
c       if(ILB .eq. 1) then
c       upsl = xrend
c       else
c       upsl = 2.*xi(isl1)
c       endif
c
c       unsl = 0.
c       iiter = 0
c546    Xa = (upsl+unsl)/2.
c       iiter = iiter+1
c       fx = theta(isl1-1,m)*(Xa - ximh)*(Xa - xiph)
c     &          /((xim3h - ximh)*(xim3h - xiph))
c     &          +theta(isl1,m)*(Xa - xim3h)*(Xa - xiph)
c     &          /((ximh - xim3h)*(ximh - xiph))
c     &          +theta(isl1+1,m)*(Xa - xim3h)*(Xa - ximh)
c     &          /((xiph - xim3h)*(xiph - ximh))
c
c       write(*,2030) iiter, isl1 ,sngl(Xa), sngl(fx)
c2030   format(2x,'Niter=',i5,3x,'Nnode=',i5,3x,'Xa=',e12.5,
c     &          3x,'fx=',e12.5)

c       dfx = fx - SLOPT
c       if(dabs(dfx) .le. eps3) then
c       delta(m) = xa
c       go to 550
c       elseif(dfx .lt. 0.) then
c       unsl = xa
c       go to 546
c       else
c       upsl = xa
c       go to 546
c       endif
c550    CONTINUE

C       WRITE(*,609) SNGL(THETA(K,M)),SNGL(THETA(K+1,M)),
C     &          SNGL(THETA(L-1,M)),SNGL(THETA(L,M))

        CONVERGE = SUMDIF/SUMASS

C--------------------------------------------------------------------
C ****  CHECK ENERGY BALANCE
C--------------------------------------------------------------------
C
C       DETERMINE THE ENERGY OUTPUT WITHIN DT
C       FIRST, BY HEAT FLUX
```

```
      IF(ILB .EQ. 1) THEN
      DDXPAI = 2.*(PAI(K,M) - PAIO)/DX(K)
      QNEW = CONDX(K,M)*HF/CPX(K,M)*DDXPAI

      ELSEIF(ILB.EQ.4) THEN
      QNEW = HRR*(TX(K,M) - TA)
      ENDIF


      SUMQINDT = SUMQINDT + QNEW
      QOUT = QNEW*DT
C **** THEN, BY TOTAL SYSTEM ENERGY  [JOULE]
      ETOTNEW = 0.
      DO 705 I = ILEND, IREND
705   ETOTNEW = ETOTNEW + RHOX(I,M)*HX(I,M)*DX(I)

      ECOR = 0.
      DO 706 I = ILEND, IREND
706   ECOR = (RHOX(I,M) - SRHOX(I))*HSF*DX(I)
      EOUT = ETOTOLD - ETOTNEW + ECOR
      EBALANCE = (QOUT - EOUT) / EOUT *100.
      EBALRAT = ((QOUT - EOUT) / EOUT)**2.
      TSTPRAT = QOUT/EOUT

      DIF = 0.
      DO 560 I = ILEND, IREND
      DIF = DIF + ((HX(I,M) - HX(I,M-1))/HX(I,M-1))**2.
560   CONTINUE

      IF(DIF .GT. EPS1) THEN

      DO 801 JI = ILEND,IREND
      HX(JI,M-1)    = HX(JI,M)
      CALL PROPERTY(HX(JI,M-1),TX(JI,M-1),CPX(JI,M-1),
     &        RHOX(JI,M-1),CONDX(JI,M-1))
      THETA(jI,M-1) = (HX(jI,M-1) - HSF)/HF
      THEC(jI) = -THETA(jI,M-1)
      IF(THEC(jI) .GT. 0.) THEC(jI) = 0.
      IF(THEC(jI) .LT. -1.) THEC(jI) = -1.
      PAI(jI, M-1) = THETA(jI, M-1) + THEC(jI)
801   CONTINUE
      GO TO 1050

      ENDIF

      WRITE(*, 610) SNGL(EOUT), SNGL(QOUT), SNGL(EBALANCE),
     &         SNGL(TOTEOUT), SNGL(QTOT),
     &        SNGL(TOTBAL), isl1,sngl(delta(m))

C     WRITE(*,602) MA, IDITER, NITER, SNGL(CONVERGE), SNGL(SUMASS)

C     OUTER ITERATION ——- DELTA ITERATION
C     CONVERGENCE TEST FOR DELTA
C
C     IF(DELTA(M).GT.1.D-5 .AND. DELTA(M-1) .GT. 1.D-5) THEN
C     DIFDELTA = ((DELTA(M) - DELTA(M-1))/DELTA(M-1))**2.
C     ELSE
C     DIFDELTA = DELTA(M) - DELTA(M-1)
C     ENDIF
C
C     DELTA(M-1) = (DELTA(M-1)+DELTA(M))/2.
C     if(delta(m-1) .le. sdelta) delta(m-1)=sdelta
C
C     IF(DIFDELTA .GT. EPS2) THEN
```

```
C        DO 816 I = ILEND, IREND
C        HX(I,M-1) = HX(I,M)
C        CALL PROPERTY(HX(I,M-1),TX(I,M-1),CPX(I,M-1),
C     &          RHOX(I,M-1),CONDX(I,M-1))
C
C        THETA(I,M-1) = (HX(I,M-1) - HSF)/HF
C        THEC(I) = -THETA(I,M-1)
C        IF(THEC(I) .GT. 0.) THEC(I) = 0.
C        IF(THEC(I) .LT. -1.) THEC(I) = -1.
C        PAI(I, M-1) = THETA(I, M-1) + THEC(I)
C816     CONTINUE
C        GO TO 1025
C        ENDIF

C----------------------------------------------------------------------
C ****   TRANSFORMATION INTO DIMENSIONED VARIABLES
C----------------------------------------------------------------------
         DO 700 I = 1, NX

         HX(I,M) = THETA(I,M)*HF + HSF
         CALL PROPERTY(HX(I,M),TX(I,M),CPX(I,M),RHOX(I,M),CONDX(I,M))

700      CONTINUE

         TOTEOUT = ETOTINIT - ETOTNEW
         QTOT = QTOT + QOUT
         TOTBAL = QTOT/TOTEOUT

C ****   STORE PRESENT STEP VALUES IN THE PREVIOUS STEP VARIABLES TO ADVANCE
C        NEXT TIME STEP

         DO 710 I=1,NX
         THETA(I,M-1) = THETA(I,M)
         PAI(I,M-1)   = PAI(I,M)
         TX(I,M-1)    = TX(I,M)
         HX(I,M-1)    = HX(I,M)
         RHOX(I,M-1)  = RHOX(I,M)
         CPX(I,M-1)   = CPX(I,M)
         CONDX(I,M-1) = CONDX(I,M)
 710     CONTINUE
         DELTA(M-2) = SDELTA
         DELTA(M-1) = DELTA(M)
         DDELTADT = (DELTA(M-1) - DELTA(M-2))/DT
         QOLD = QNEW
         ETOTOLD = ETOTNEW
         FROFRAC = DELTA(M)/XL
         TIMEA = DT * DFLOAT(MA)
         TIMEB = DT * DFLOAT(MA+1)
         ma1 = ma/NpStp * NpStp
         if(ma .lt. 20 .or. ma1 .eq. ma) then
         deltaxl = delta(m)/XL
         WRITE(11,1133) SNGL(TIMEA), SNGL(DDELTADT), SNGL(DeltaXL),
      &          SNGL(Tx(ILEND,m))
         WRITE(14,1130) SNGL(TIMEA), SNGL(DELTA(M))
         endif
         DO 770 NK = 1,NPRT
         IF(TIMEA.LE.PRTIME(NK) .AND. TIMEB .GT.PRTIME(NK)) THEN
         WRITE(12,712) SNGL(PRTIME(NK))
         WRITE(13,712) SNGL(PRTIME(NK))
         WRITE(13,713) SNGL(FROFRAC)
 713     FORMAT(2X,'FROZEN FRACTION = ',E12.5)
         WRITE(*,712) SNGL(PRTIME(NK))

         ITIME = ITIME + 1
         WRITE(12, 610) SNGL(EOUT), SNGL(QOUT), SNGL(EBALANCE)
```

```fortran
     &          , SNGL(TOTEOUT), SNGL(QTOT),
     &          SNGL(TOTBAL), Isl1, sngl(delta(m))
        WRITE(*, 610) SNGL(EOUT), SNGL(QOUT), SNGL(EBALANCE)
     &          , SNGL(TOTEOUT), SNGL(QTOT),
     &          SNGL(TOTBAL), Isl1, sngl(delta(m))
        WRITE(12, 1110)
1110    FORMAT(/5X,'DIMENSIONLESS ENTHALPY DISTRIBUTION'/
     &         4X,'-----------------------------------'/
     &         5X,'POSITION       ENTHALPY')
        DO 8010 IJ = ilend, irend
8010    WRITE(12, 1120) SNGL(XI(IJ)),SNGL(THETA(IJ,M))
        WRITE(13, 1111)
1111    FORMAT(/5X,'TEMPERATURE DISTRIBUTION'/
     &         4X,'-----------------------------------'/
     &         5X,'POSITION       TEMPERATURE (K)')
        DO 8011 IJ = ilend, irend
8011    WRITE(13, 1120) SNGL(XI(IJ)),SNGL(TX(IJ,M))
1120    FORMAT(5X,E12.5,3X,E12.5)
1130    FORMAT(5X,E12.5,3X,E12.5,3X,E12.5)
        ENDIF
770     CONTINUE
9000    CONTINUE

C---------------------------------------------------------------
C       FORMAT STMTS
C---------------------------------------------------------------
21      FORMAT(/5X,'CONSTANT WALL TEMPERATURE LEFT BOUNDARY'/
     &         4X,'----------------------------------------'/
     &         5X,'Twall                      =',F10.5,' [K]'/
     &         5X,'Dimless Wall Enthalpy    =',E12.5/
     &         5X,'Dimless Wall Temperature =',E12.5)

22      FORMAT(/5X,'RADIATIVE LEFT BOUNDARY'/
     &         4X,'----------------------------------------'/
     &         5X,'Initial Twall              =',F10.5,' [K]'/
     &         5X,'Dimless Wall Enthalpy    =',E12.5/
     &         5X,'Dimless Wall Temperature =',E12.5)

26      FORMAT(/5X,'MATERIAL PROPERTIES'/
     &         4X,'-----------------------------------------------'/
     &         5X,'Tm [K] = ',F10.3,5x,'DHf [J/Kg] = ',F10.3/
     &         5X,'                      LIQUID          SOLID'/
     &         4X,'-----------------------------------------------'/
     &         5X,'H       [J/Kg]',3x,F10.2,2x,F10.2/
     &         5X,'Cp    [J/Kg-K]',3x,F10.2,2x,F10.2/
     &         5x,'K    [J/s-M-K]',3x,F10.2,2x,F10.2/
     &         5x,'Rho  [Kg/M**3]',3x,F10.2,2x,F10.2)

602     FORMAT(/5X,'TIME STEP NO. =',I5/
     &         5X,'NO. OF DELTA ITERATION =',I5/
     &         5X,'NO. OF INNER ITERATION =',I5/
     & 5X,'CONVERGENCE =',E12.5,'  SUMASS =',E12.5/)

609     FORMAT(/5X,'DIMENSIONLESS ENTHALPY DISTRIBUTION AT'/
     & 5X,'ILEND         ILEND+1         IREND-1         IREND'/
     & 4X,E12.5,2X,E12.5,2X,E12.5,2X,E12.5)

610     FORMAT(//5X,'ENERGY BALANCE AT THIS TIME STEP'/
     &          4X,'-------------------------------'/
     &          5X,'EOUT      = ',E12.5,'    [JOULES]'/
     &          5X,'QOUT      = ',E12.5,'    [JOULES]'/
     &          5X,'% ERROR   = ',E12.5//
     &          5X,'TOTAL ENERGY BALANCE UPTO NOW'/
     &          4X,'-------------------------------'/
     &          5X,'EOUT      = ',E12.5,'    [JOULES]'/
```

```
     &           5X,'QOUT      = ',E12.5,'    [JOULES]'/
     &           5X,'QOUT/EOUT = ',E12.5//
     &           5X,'NODE FREEZING=',I5/
     &           5X,'CRUST THICKNESS =',E12.5//)
712      FORMAT(/5X,'COMPUTED RESULTS AT TIME =',F12.2,' [SEC]')
1133     FORMAT(2x,E12.5,2x,E12.5,2x,E12.5,2x, E12.5)

9999     STOP
         END

         SUBROUTINE PROPERTY(H,T,CP,RHO,CON)
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/PROP/HL,HS,CPL,CPS,CONL,CONS,RHOL,RHOS,TF
         IF(H .GT. HL) THEN
         T = TF + (H - HL)/CPL
         CP = CPL
         RHO = RHOL
         CON = CONL
         ELSEIF(H .GE. HS) THEN
         T = TF
         CP = CPL - (HL - H)*(CPL - CPS)/(HL-HS)
         CON = CONL - (HL - H)*(CONL - CONS)/(HL-HS)
         RHO = RHOL - (HL - H)*(RHOL - RHOS)/(HL-HS)
         ELSE
         T = TF + (H - HS)/CPS
         CP = CPS
         RHO = RHOS
         CON = CONS
         ENDIF
         RETURN
         END
```

Appendix A-2

Input Data for the Computer Program COPHASE.FOR

```
$OPTION
MRB = 1,       ILB = 0,        IRB = 1,
NpStp = 5,
IDX = 0,
$END
$INPUT1
Fa = .5,       Xsbl = 0.05,
Ft = .5,       NT = 200,       TIME = 2.,      dt = .0005,
Fx = 1.05,     NX = 325,       XL =.05,
vopt = 0.,     Av = 7.80D-06,  Bv =0. ,        Cv = 0.,
Div = 1.,      CO = 4.26D-06,
NPRT = 10
$END
$CONVER
eps1 = 2.D-2
$END
```

Appendix B-1

List of Computer Program SEGR.FOR

```
$OPTION
ILB = 4,         NPRT = 10,        NpStp = 1
$END
$INPUT1
NT = 13,         TIME = 7800.,     NX=50,  XL = .05,
HINIT=1.132D6,
TINIT = 454.,    TF=454.,          TA=250.,
HSF=0.6999D6,    HLF=1.132D6,
CPS=3.963D3 ,    CPL=4.384D3,
CONDSF=71.20,    CONDLF=42.80,
RHOSF=530.,      RHOLF=517.,
EMIT=0.8,
TWALL=100.,      HWALL=.6D6
$END
$CONVER
EPS1 = 1.D-30, EPS2 = 1.D-12, EPS3 = 1.D-14
$END
```

```
      dtprt = time/dfloat(nprt)
      DO 5 I=1,nprt
      PRTIME(I) = dfloat(i)*dtprt
5     CONTINUE

C **** MESH GENERATION: STABILITY OF SOLUTION
C     find maximum velocity, Vmax
      if(idx .eq. 0) then
      Vmax = V(1)
      do 10 i = 1,nt
      if(Vmax .le. V(i)) then
      Vmax = V(i)
      endif
10    continue
      dx1 = 2.*Dc/Vmax/div

20    sum = 1
      do 25 i=2,nx
25    sum = sum + fx**dfloat(i-1)

      dxo = XL/sum
      Xi(0) = 0.
      do 30 i = 1, Nx
      dx(i) = dxo*fx**dfloat(i-1)
30    Xi(i) = Xi(i-1) + dx(i)

              if(dx1.lt.dx(1)) then
              nx = nx+1
              go to 20
              endif

      elseif(idx .eq.1) then
      dxs = xsbl/Dfloat(Nx)
      DO 35 i = 1,Nx
      dx(i) = dxs
35    Xi(i) = Xi(i-1) + dx(i)

      else
      Vmax = V(1)
      do 1011 i = 1,nt
      if(Vmax .le. V(i)) then
      Vmax = V(i)
      endif
1011  continue
      dx1 = 2.*Dc/Vmax/div

2011  sum = 1
      do 2511 i=2,nx
2511  sum = sum + fx**dfloat(i-1)

      dxo = XL/sum
      Xi(0) = 0.
      do 3011 i = 1, Nx
      dx(i) = dxo*fx**dfloat(i-1)
3011  Xi(i) = Xi(i-1) + dx(i)

              if(dx1.lt.dx(1)) then
              nx = nx+1
              go to 2011
              endif


      endif
      dxend = dx(1)/20.
      Nx1 = Nx-1
```

```
        Nx10 = Nx-10

C ****  PREPARATION FOR INITIAL CONDITIONS

        SGAM = RHOS/RHOL - 1.
        IF(MRB .EQ.0) SGAM = 0.

C ****  INITIAL CONCENTRATION

        DO 40 J = 1,2
        DO 40 I = 1, NX
        Cx(I,J) = Co
40      CONTINUE

C---------------------------------------------------------------------------
C ****  INITIAL TOTAL MASS OF GAS IN THE SYSTEM
C---------------------------------------------------------------------------
        Citot = 0.
        DO 50 I = 1, NX
50      Citot = Citot + Cx(I,1)*DX(I)
        Ctotold = Citot
c Co*XL

C ****  Bubble Volume Calculation
        ExcHe = Co*XL/AwHe
        VolHetot = ExcHe*Rg*To/Po

        CAdmass = 0.
        Cflux  = 0.
        DELTA = 0.
        iend = 0
        tr = 0.
        itime = 0
C---------------------------------------------------------------------------
C ****  THE BEGINNING OF TIME LOOP *******************************************
C---------------------------------------------------------------------------
9000    Continue
c       DO 9000 MA = 1, NT
        itime = itime + 1
        M=2
        ma = 1
        tr = tr + dt
        V(ma) = Av + Bv*tr + Cv*tr**2.

        G = -(1.+sGam)*V(ma)
        DELTA    = DELTA + v(ma)*DT
        Xliq = Xlo-(1.+sGam)*Delta

        if(xliq .le. dXend) then
        iend = 1
        endif

        print *,' '
        print *,'  S-L Front = ',Sngl(delta),' [M]'
        print *,'  Liquid Length = ',Sngl(Xliq),' [M]'

        if(MRB .eq. 1) then

        Xend = XLo - sGam*DELTA
        Vf = (XLo-Xend)/XLo
        print *,'  Void Forming At Right Boundary'
        print *,'  Void Location is ', sngl(Xend),' [M]'
        print *,'  Vol(Void)/Vol(Sys) =',sngl(Vf)

        else
```

```
      Xend = XLo
      Vf = 0.
      print *,'  No Void Formation'
      print *,'  Void Location is ', sngl(Xend),' [M]'
      print *,'  Vol(Void)/Vol(Sys) =',sngl(Vf)

      endif

C-----------------------------------------------------------------------
C ****  Determine Equivalent Interface Diffusion Coefficients and
C       Patanckar High Flux Function
C-----------------------------------------------------------------------
      DO 60 i = 1,Nx
      Di(i) = 2.*Dc/(DX(i)+DX(i+1))
      Pe = G/Di(i)
      Fpe(i) = FP(Pe)
60    Continue

C-----------------------------------------------------------------------
C ****  DETERMINE COEFFICIENTS FOR LEFT BOUNDARY NODE
C-----------------------------------------------------------------------
      DFL = Di(1)*FPe(1)
      Dm1 = Dc/DX(1)
      Pe1 = G/Dm1
      FPe1 = FP(Pe1)
      DFL1 = Dm1*FPe1
      Aw(1) = 0.
      Ae(1) = -Fa*DFL

      IF(ILB .EQ. 1) THEN                         !  J- = -Cs*[d(delta)/dt]
      Am(1) = DX(1)/dt + Fa*(DFL + G)
      S(1) = Dx(1)/dt*Cx(1,M-1) - Cs*v(ma)
     &          + (1.-Fa)* (DFL*Cx(2,m-1) - (G + DFL)*Cx(1,m-1))

      ELSEIF(ILB .EQ. 2) THEN                     !  J- = beta*(Ci - Cs)
      beta = 1./(1./v(ma) - DX(1)/2./Dc)
      Am(1) = DX(1)/dt + Fa*(DFL + G - beta)
      S(1) = Dx(1)/dt*Cx(1,M-1) - Cs*beta
     &          + (1.-Fa)* (DFL*Cx(2,m-1) + (beta - DFL - G)*Cx(1,m-1))

      ELSEIF(ILB .EQ. 3) THEN                     !  J- = 0.
      Am(1) = DX(1)/dt + Fa*(DFL + G)
      S(1) = Dx(1)/dt*Cx(1,M-1)
     &          + (1.-Fa)* (DFL*Cx(2,m-1) - (DFL + G)*Cx(1,m-1))

      ELSEIF(ILB .EQ. 4) THEN                     !  J- = (Co - Cs)*[d(delta)/dt]
      Am(1) = DX(1)/dt + Fa*(DFL + G)
      S(1) = Dx(1)/dt*Cx(1,M-1) + (Co - Cs)*V(ma)
     &          + (1.-Fa)* (DFL*Cx(2,m-1) - (DFL + G)*Cx(1,m-1))

      ELSEIF(ILB .EQ. 5) THEN                     !  J- = (C1 - CS)*[d(delta)/dt]
      Am(1) = DX(1)/dt + Fa*(DFL + G - v(ma))
      S(1) = Dx(1)/dt*Cx(1,M-1)  - Cs*V(ma)
     &          + (1.-Fa)* (DFL*Cx(2,m-1) - (DFL + G - v(ma))*Cx(1,m-1))

      ELSE                                        !  C(0, T) = Ch
      Am(1) = DX(1)/dt + Fa*(DFL + G + DFL1)
      S(1) = Dx(1)/dt*Cx(1,M-1)  + (G + DFL1)*Co
     &          + (1.-Fa)* (DFL*Cx(2,m-1) - (DFL + G + DFL1)*Cx(1,m-1))

      ENDIF

C-----------------------------------------------------------------------
C ****  DETERMINE COEFFICIENTS FOR INTERIOR NODES
```

```
C------------------------------------------------------------------------

        DO 70 I = 2, Nx1
        DFi = Di(i)*FPe(i)
        DFi1 = Di(i-1)*FPe(i-1)

        Aw(i) = -Fo*(DFi1 + G)
        Ae(i) = -Fo*DFi
        Am(i) = DX(i)/dt - (Ae(i) +Aw(i))
        S(i)  = Dx(i)/dt*Cx(i,M-1) - (1.-Fo)/Fo*(Ae(i)*Cx(i+1,m-1)
     &          -(Ae(i)+Aw(i))*Cx(i,m-1) + Aw(i)*Cx(i-1,m-1))

70      CONTINUE

C------------------------------------------------------------------------
C ****  DETERMINE COEFFICIENTS FOR RIGHT BOUNDARY NODE
C------------------------------------------------------------------------
        DFR = Di(Nx)*FPe(Nx)
        DFR1 = Di(Nx1)*FPe(Nx1)
        Aw(Nx) = -Fo*(DFR1 + G)
        Ae(Nx) = 0.

        IF(IRB .EQ. 1) THEN                        !  J+ = Co*G
        Am(Nx) = DX(Nx)/dt + Fo*DFR1
        S(Nx)  = Dx(Nx)/dt*Cx(Nx,M-1) - Co*G
     &          + (1.-Fo) * ( (G+DFR1)*Cx(Nx1,m-1)
     &          - DFR1*Cx(Nx,m-1))

        ELSEIF(IRB .EQ. 2) THEN                    !  Ci+1 = Ci
        Am(Nx) = DX(Nx)/dt -Aw(Nx)
        S(Nx)  = Dx(Nx)/dt*Cx(Nx,M-1)
     &          + (1.-Fo)/Fo*Aw(Nx)*(Cx(Nx1,m-1) - Cx(Nx,m-1))

        ELSEIF(IRB .EQ. 3) THEN                    !  Ci+1 = Co
        Am(Nx) = DX(Nx)/dt + Fo*DFR - Aw(Nx)
        S(Nx)  = Dx(Nx)/dt*Cx(Nx,M-1) + DFR*Co
     &          + (1.-Fo)*((DFR1 + G)*Cx(Nx1,m-1)
     &          - (DFR + G + DFR1)*Cx(Nx,m-1))

        ELSEIF(IRB .EQ. 4) THEN                    !  Ci+1 = 0.
        Am(Nx) = DX(Nx)/dt + Fo*DFR - Aw(Nx)
        S(Nx)  = Dx(Nx)/dt*Cx(Nx,M-1)
     &          + (1.-Fo)*((DFR1 + G)*Cx(Nx1,m-1)
     &          - (DFR + G + DFR1)*Cx(Nx,m-1))

        ELSE                                       !  J+ = CoG + DiFi(Ci-Co)
        Am(Nx) = DX(Nx)/dt + Fo*(DFR + DFR1 + G)
        S(Nx)  = Dx(Nx)/dt*Cx(Nx,M-1) - Co*(G - DFR)
     &          + (1.-Fo)*((DFR1 + G)*Cx(Nx1,m-1)
     &          - (DFR + DFR1 + G)*Cx(Nx,m-1))

        ENDIF

C------------------------------------------------------------------------
C ****  GET SOLUTION FOR PRESENT TIME STEP; TRIDIAGONAL MATRIX SOLVER
C------------------------------------------------------------------------

        CALL TDMA(sCx, Aw, Am, Ae, S, Nx)

C ****  STORE CALCULATED VALUES IN THE PRESENT VARIABLES
        DO 100 I=1,NX
        Cx(I,M) = sCx(I)
c       Cx(i,m) = Cx(i,m)/(1. - (1. + sGam)*v(m)*dt/xl)
        dx(i) = dx(i)*(1. - (1. + sGam)*v(m)*dt/xl)
```

```
      xi(i) = xi(i-1) + dx(i)
100   CONTINUE

      Cmax = Cx(1,m)
      Imax = 1
      Do 299 i = 1,Nx
      if(Cmax .LE. Cx(i,m)) then
      Cmax = Cx(i,m)
      Imax = I
      endif
299   continue

      CxCh = Cx(Imax,m)/Ch
      WRITE(11, 6030) SNGL(TR), CxCh
      itck = itime/Npstp*Npstp
      if(itck .eq. itime .or. itime .lt. 20 .or. iend .eq.1) then
      WRITE(14, 6030) SNGL(TR), Delta
      endif

      print *,' '
      print *,'Time=', SNGL(TR),'   Cx(1)=', Sngl(Cx(1,m))
      Print *,'Max. C Appears at i =',imax,'  Cx(Imax) =',
     &        Sngl(Cx(Imax,m))

C-----------------------------------------------------------------------
C ****  CHECK MASS BALANCE
C-----------------------------------------------------------------------
C ****  DETERMINE MASS FLUX AT LEFT BOUNDARY

      IF(ILB .EQ. 1) THEN                        !  J- = -Cs*[d(delta)/dt]
      FluxL = - Cs*V(mo)

      ELSEIF(ILB .EQ. 2) THEN                    !  J- = beta*(Ci - Cs)
      FluxL = beta*(Fo*Cx(1,m)+(1.-Fo)*Cx(1,m-1) - Cs)

      ELSEIF(ILB .EQ. 3) THEN                    !  J- = 0.
      FluxL = 0.

      ELSEIF(ILB .EQ. 4) THEN                    !  J- = (Co - Cs)*[d(delta)/dt]
      FluxL = (Co - Cs)*V(mo)

      ELSEIF(ILB .EQ. 5) THEN                    !  J- = (Ci - Cs)*[d(delta)/dt]
      FluxL = (Fo*Cx(1,m) + (1.-Fo)*Cx(1,m-1) - Cs)*v(mo)

      ELSE                                       !  C(0, T) = Ch
      FluxL = (G + DFL1)*Co - DFL1*(Fo*Cx(1,m) + (1. - Fo)*Cx(1,m-1))
      ENDIF

C-----------------------------------------------------------------------
C ****  DETERMINE MASS FLUX AT RIGHT BOUNDARY

      IF(IRB .EQ. 1) THEN                        !  J+ = Co*G
      FluxR = Co*G

      ELSEIF(IRB .EQ. 2) THEN                    !  Ci+1 = Ci
      FluxR = G*(Fo*Cx(Nx,m) + (1.-Fo)*Cx(Nx,m-1))

      ELSEIF(IRB .EQ. 3) THEN                    !  Ci+1 = Co
      FluxR = (G + DFR)*(Fo*Cx(Nx,m) + (1.-Fo)*Cx(Nx,m-1))
     &        - DFR*Co

      ELSEIF(IRB .EQ. 4) THEN                    !  Ci+1 = 0.
      FluxR = (G + DFR)*(Fo*Cx(Nx,m) + (1.-Fo)*Cx(Nx,m-1))

      ELSE                                       !  J+ = CoG + DiFi(Ci-Co)
```

```
        FluxR = Co*(G - DFR)
     &          + (G + DFR)*(Fo*Cx(Nx,m) + (1.-Fo)*Cx(Nx,m-1))

        ENDIF

C------------------------------------------------------------------------
C       Obtain total mass at the time step

        Citot = 0.
        DO 200 I = 1,Nx
        Citot = Citot + Cx(i,m)*dx(i)
200     continue
        avgCi = Citot/Xliq

        Admass = (FluxL - FluxR)*dt
        CAdmass = CAdmass + Admass
        Error1 = Citot - Admass - Ctotold
        BAL1 = Error1/Ctotold *100.

C ****  Bubble Volume Calculation
        nnuc = 0
        sumdx = 0.
        Cexcess = 0.
        if(Cx(imax,m) .GT. Ch) then
        if(istop .eq.1) then
        write(12, 2115) sngl(tr)
        do 2118 i = 1,nx,2
        cxco = cx(i,m)/co
        xixliq = xi(i)/xliq/100.
2118    write(12, 2116) xixliq, cxco
2116    format(5x,d20.10,2x,d20.10)
2115    format(5x,'time = ',f10.3,' sec')
        print *,'optional stop at first occurrence of Cx(imax.m) > Ch'
        stop
        endif

        DO 205 i = 1, Nx
        if(Cx(i,m) .GE. Ch) then
        Cexcess = Cexcess + (Cx(i,m) - Ch)*dx(i)
        endif
205     continue

        if(Cexcess .GT. CAdmass) then
c       AvailC = Cexcess - CAdmass
        AvailC = Cexcess
c       CAdmass = 0.
        do 2051 i = 1, Nx

        if(Cx(i,m) .GE. Ch) then
        Cx(i,m) = Ch
        Sumdx = Sumdx + dx(i)
        Inuc = 1
        Nnuc = Nnuc + 1
        Nucnode(Nnuc) = i
        endif

2051    continue
        endif

        if(AvailC .GT. 0.) then
        THeMole = AvailC/AwHe
        HeMole = THeMole/Trnuc
        AddVol =  HeMole*Rg*To/Po
        VolHe = VolHe + AddVol
```

```
         if(volHe .GT. VolHetot) then
         AdjCx = (VolHe - VolHetot)*Po*AwHe/(Rg*To)
         VolHe = VolHetot
         Redist = AdjCx/Sumdx
         DO 206 i = 1, Nnuc
206      Cx(Nucnode(i),m) = Cx(Nucnode(i),m) + Redist*dx(Nucnode(i))

         else
         AdjCx = 0.
         endif

         endif

         endif

         VolSh = Delta*sGam
         Voltot = VolHe + VolSh
         Write(*,6080) sngl(AddVol), sngl(VolHe), sngl(VolSh), sngl(Voltot)
         WRITE(*, 6020) SNGL(Citot), SNGL(Admass), SNGL(Ctotold),
      &   SNGL(Error1), SNGL(BAL1)

C ****   STORE PRESENT STEP VALUES IN THE PREVIOUS STEP VARIABLES TO ADVANCE
C        NEXT TIME STEP
         DO 300 I=1,NX
         Cx(I,M-1) = Cx(I,M)
300      CONTINUE
         Ctotold = Citot - Cexcess + AdjCx
         AvailC = 0.
C----------------------------------------------------------------------------
C ****   TRANSFORMATION INTO DIMENSIONLESS VARIABLES
C----------------------------------------------------------------------------

         TB = Tr + DT
         CxCh = Cx(Imax,m)/Ch
         itck = itime/NpStp*NpStp

         if(itck .eq. itime .or. itime .eq. 1
      &          .or. Inuc .EQ. 1 .or. iend .eq.1) then
         Write(13,6090) sngl(TR), sngl(VolHe), sngl(VolSh), sngl(Voltot)
         endif
         print *,' '
         print *,'Time=', SNGL(TR),'   Cx(1)=', Cx(1,m)

         DO 500 IJ = 1,NPRT

         IF(TR .LE. PRTIME(IJ) .AND. TB .GT.PRTIME(IJ)) THEN
         WRITE(*,6050) SNGL(Tr)
         WRITE(12,6050) SNGL(Tr)
         WRITE(12,6040)
         IPFlag = 1
         ENDIF
500      CONTINUE
         if(IPFlag .eq. 1) then
         do 400 ik = 1,nx,2
         CxCh=Cx(ik,m)/Ch
         CxCo = Cx(ik,m)/Co
         XiXLiq = Xi(ik)/XLiq
         Arg = (xi(ik)-dx(ik)/2.)/(2.*DSQRT(Dc*tr))
         Anal = (Ca*(1. - DERF(Arg)) + Co*DERF(Arg))/Co
400      write(12,2119) XiXLiq,  CxCo, Anal
2119     format(5x,d20.10,2x,d20.10,2x,d20.10)

         IPFlag = 0
         endif
         Inuc = 0
```

```
        if(iend .ne. 1) then
        go to 9000
        endif
c9000   CONTINUE

C-----------------------------------------------------------------------
C       FORMAT STMTS
C-----------------------------------------------------------------------

5010    FORMAT(5X,D12.5,3X,D12.5)

6001    format(1x,'L =',F10.4, ' m')

6010    FORMAT(/5X,'INITIAL CONCENTRATION DISTRIBUTION'/
     &          5x,E12.5,2X,E12.5,2X,E12.5,2X,E12.5)

6020    FORMAT(//5X,'MASS BALANCE AT THIS TIME STEP'/
     &          4X,'-----------------------------------------'/
     &          5X,'  Total Mass   = ',E12.5,' [Kg/M**3]'/
     &          5X,'  Accumulation = ',E12.5/
     &          5x,'  Initial Mass = ',E12.5/
     &          4X,'-----------------------------------------'/
     &          5x,'  Difference   = ',E12.5,' [Kg/M**3]'/
     &          5X,'  % ERROR      = ',E12.5)

6030    FORMAT(3X,E12.5,3X,D20.10)

6040    FORMAT(/5x,'He Concentration Distribution'/
     &          4x,'-----------------------------------------'/
     &          5x,'Position(M)   Concentration [Kg/M**3]'/
     &          4x,'-----------------------------------------')

6050    FORMAT(/5X,'COMPUTED RESULTS AT TIME =',E12.5,' [SEC]')

6070    FORMAT(3X,E12.5,3X,D20.10)
6080    format( 5x,'Information for Void'/
     &          4x,'---------------------------------------------'/
     &          5x,'Added He Vol.           =',e12.5,'  [M**3]'/
     &          5x,'Total He Vol.           =',e12.5/
     &          5x,'Vol. of Shrinkage Void =',E12.5/
     &          5x,'Total Void Volume       =',E12.5//)
6090    format(2x,e12.5,2x,e12.5,2x,e12.5,2x,e12.5)

9999    STOP
        END


c ******************************************************************
c TRIDIAGONAL MATRIX SOLVER : aWi.X(i-1) + aPi.X(i) + aEi.X(i+1) = Si
c ******************************************************************

      SUBROUTINE TDMA(X, aW,aP,aE,S, N)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION X(1), aW(1),aP(1),aE(1),S(1)
      DIMENSION P(1000),Q(1000)
c
c     (  aW(1) = 0.   ,  aE(N) = 0.  )
c
      P(1) = -aE(1)/aP(1)
      Q(1) =   S(1)/aP(1)
c
      DO 10 i=2,N
        DENUM =  aP(i)+aW(i)*P(i-1)
        P(i) =  -aE(i)/DENUM
```

```
          Q(i) = ( S(i)-oW(i)•Q(i-1) ) /DENUM
 10     CONTINUE
c
        X(N) = Q(N)
c
        DO 20 i=N-1,1 , -1
          X(i) = P(i)•X(i+1)+Q(i)
 20     CONTINUE
c
        RETURN
        END
```

Appendix B-2

Input Data for Computer Program SEGR.FOR

```fortran
      PROGRAM SEGRGATE
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION PRTIME(20), Nucnode(1000)
      COMMON  /CONVE1/ EPS1, EPS2, EPS3
      COMMON  /MESHX/ Xi(0:1000), dx(1000), sdx(1000), DT
      COMMON  /CONCE/ Cx(0:1000,2), Di(0:1000), FPe(0:1000),
     &        sCx(1000)
      COMMON  /VELOC/ v(0:1000), T(0:1000), vel(0:2), tm(0:2)
      COMMON  /MATRIX/ Aw(1000), Am(1000), Ae(1000), S(1000)
      NAMELIST /INPUT1/ NT, TIME, NX, XL, Co, Cs, Ch, Dc, vopt,
     &        Av, Bv, Cv, Div, nprt, fx, ft, Fa, dt, xsbl, Ca
      NAMELIST /CONVER/ EPS1, EPS2, EPS3
      NAMELIST /OPTION/ MLB, MRB, ILB, IRB, Npstp, IDX, istop

C ****  IRB = 0 : Adiabatic Right Boundary
C           = 1 : Constant supply of gas at initial concentration
c                 with the speed of (solidification+volume shrinkage)

C ****  Function F(Pe): Patanckar High Flux
      FP(z) = DMAX1(0.D0 , 1.D0 - 0.1D0*DABS(z))**5.D0
     &        + DMAX1(0.D0, -z)

      OPEN(10, ERR=9999, FILE='SEGR6.INP', STATUS='OLD')
      OPEN(11, ERR=9999, FILE='SEGR6.Cmt', STATUS='UNKNOWN')
      OPEN(12, ERR=9999, FILE='SEGR6.Cxi', STATUS='UNKNOWN')
      OPEN(13, ERR=9999, FILE='SEGR6.Vol', STATUS='UNKNOWN')
      OPEN(14, ERR=9999, FILE='SEGR6.Del', STATUS='UNKNOWN')
      OPEN(15, ERR=9999, FILE='SEGR6.Tem', STATUS='UNKNOWN')
      OPEN(20, ERR=9999, FILE='vel.Tem', STATUS='old')

C ****  DEFAULT VALUES FOR OPTIONS
      DATA MLB/0/, MRB/0/, ILB/1/, IRB/0/, Vopt/0/
      DATA dt/ 5./

C ****  CONSTANTS
      DATA PHI/3.1415926/, Dc/5.96D-09/, Rhos/530./, Rhol/517./
      DATA Ch/1.737D-03/, Co/4.26D-06/, Cs/1.59D-08/,
     &       Csat/1.59D-08/
      DATA Po/7330./, To/454./, Rg/8.3143/, AwHe/4.0026D-03/
C ****  READ  NAMELIST INPUT
      READ(10,OPTION)
      READ(10,INPUT1)
      READ(10,CONVER)
      write(11,6001) XL
      write(12,6001) XL
      write(13,6001) XL
      write(14,6001) XL
      write(15,6001) XL
C ****  THE TIME STEP SIZE: UNIFORM

      DTin = TIME/DFLOAT(NT)

      IF(VOPT .GT. 0.) THEN
      DO 1 I=1,NT
1             READ(20,5010) T(I),V(I)
      ELSE
              DO 2 I=1,NT
              T(I) = dFLOAT(I)*dtin
2             V(I) = Av + Bv*t(i) + Cv*t(i)**2.
      ENDIF

      XLo = XL
      Nxo = Nx

C ****  IDENTIFY PRINTOUT TIME
```

Appendix C

List of Computer Program FREZ.FOR

```
c       write(6, 6002)
        write(7, 6003)
c       write(8, 6004)
        write(9, 6005)
        write(10,6006)
        write(17,6007)
        write(18,6008)
        write(19,6009)
        endif

        if(iPCM .ne. 1) then     ! LiF calculation
C LiF properties
        Tf= Tff
        Ti=Tif
        xKs=xKsf
        xKl=xKlf
        Cps=Cpsf
        rhos=rhosf
        rhol=rholf
        Hsl=Hslf
        xKv=xKvf
        endif

        n=5
        nx=2

        if(iopt .eq. 1) go to 9000


c       if(ivoid .eq. 1) then
c       do = 0.00001
c       slo = do
c       else
        do = 0.
        slo = do
c       endif
        dos = do

        do 10 i=1,n
        To(i)=Ti
        Ts(i)=Ti
 10     continue

        Hl = Hsl + Cps*Tf
        eps3o = eps3
        epso = eps1
        eps12 = dsqrt(eps1)
        Gam = 1. - rhol/rhos
        rGam = Gam/(1.-Gam)

        As = xKs/rhos/Cps
        Ac = xKc/rhoc/Cpc
        a1 = rhoc*Cpc*tc/dt
        a2 = rhos*Cps/dt

        hcs = 2.*xKc/tc
        Al = Emitout*Boltz/hcs
        SumLam=0.
        do 50 it=1, nt

        ind = it/npr*npr - it
        time = dfloat(it)*dt
        Erdo = 1.D30
        if(it .eq. 1) then
        dmax = do + 2.*eps3
```

```
      else
      dmax = do + 2.*vel*dt
      endif
      dmin = do
      iout = 0

20    continue
      iout = iout + 1

C Assume ds; new solid crust thickness
      ds = (dmax+dmin)/2.
      dvs = rGam*ds
      if( ivoid .eq. 1) then
      sls = ds + dvs
      else
      sls = ds
      endif

C Equivalent heat transfer coefficients
C At the outer wall of the cladding, ho1

      Den = 1./Emitout/Boltz + 4.*Ts(1)**3/hcs
      ho1 = (Ts(1)**2+Ta**2)*(Ts(1)+Ta)/Den

C within the void

      hss = 2.*xKs/ds
      hcvs = xKv/dvs
      hr2s = Emitin*Boltz*(Ts(4)**2 + Ts(3)**2)*(Ts(4) + Ts(3))
      hv = hcvs + hr2s
c     hv = hcvs

      IF(iBC .eq. 1) THEN

C Overall between T1 and T2
C2    This considers heat transfer across the void
      if(ivoid .eq. 1) then
      h12 = 1./(1./hcs + 1./hv + 1./hss)
      else
C BC1 This does not consider heat transfer across the void
      h12 = 1./(1./hcs + 1./hss)
      endif

C BC1 Radiative Boundary
      aw(1) = 0.
      ap(1) = ho1 + h12 + a1
      ae(1) = -h12
      s(1) = a1*To(1) + ho1*Ta

      aw(2) = -h12
      ap(2) = h12 + hss + a2*ds
      ae(2) = 0.
      s(2) = rhos*Hl/dt*(ds-do) + a2*do*To(2) + hss*Tf

      call tdma(Tn,aw,ap,ae,s,nx)

      dT5 = Al*(Tn(1)**4.-Ta**4)/(1.+4.*Al*Tn(1)**3)
      Tn(5) = Tn(1)-dT5

C BC1 This considers heat transfer across the void
      if(ivoid .eq. 1) then
      Tn(4) = h12*(Tn(1)/hss + (1./hcs + 1./hv)*Tn(2))
      Tn(3) = h12*(Tn(2)/hcs + (1./hss + 1./hv)*Tn(1))
      else
C BC1 This does not consider heat transfer across the void
```

```
      Tn(4) = h12*(Tn(1)/hss + 1./hcs*Tn(2))
      Tn(3) = h12*(Tn(2)/hcs + 1./hss*Tn(1))
      endif


      ELSE
C BC2 Constant wall temperature
      Tn(1) = Ta
      Tn(5) = Tn(1)
      Tn(3) = Tn(1)

      if(ivoid .eq. 1) then
C BC3 Constant wall temperature, with void
      h12 = 1./(1./hss + 1./hv)
      else
C BC2 Constant wall temperature, without void
      h12 = hss
      endif

      Tn(2) = (a2*do*To(2) + rhos*Hl*(ds-do)/dt
    & + hss*Tf + h12*Tn(1))/(h12 + hss + a2*ds)

      if(ivoid .eq. 1) then
      Tn(4) = (hv*Tn(1) + hss*Tn(2))/(hss + hv)
      else
      Tn(4) = Tn(1)
      endif

      ENDIF

C Check energy balance of the system
C Heat flux at the left boundary of the system

      Qls = -ho1*(Tn(1)-Ta)

C Heat flux at the left boundary of the PCM

      Qlp = -h12*(Tn(2)-Tn(1))
C For constant wall temperature only
      if(iBC .eq. 2) Qls = Qlp

C Heat flux at the right boundary of the system, x=xLB + delta
C which is equal to Heat flux at the right boundary of the solid PCM

      Qrs = -rhos*Hsl*(ds-do)/dt

C Total heat accumulated in the system within dt
c Fully implicit scheme
      EQp = (Qlp - Qrs)*dt
      EQs = (Qls - Qrs)*dt

C Energy accumulated within the cladding
      if(iBC .eq. 1) then
      Ec = tc*rhoc*Cpc*(Tn(1) - To(1))
      else
      Ec = 0.
      endif

C Energy accumulated within the solid crust
      Ep = rhos*Cps*(ds*Tn(2) - do*To(2))-(ds-do)*rhos*Hl

C Total Energy accumulated in the system
      Ea = Ec + Ep

      Ers = (EQs - Ea)/Ea
```

```
      Erp = (EQp - Ep)/Ep

      REcEp = Ec/Ep
      vel = (ds-do)/dt

      if(it .eq. 1 .or. ind .eq. 0) then
c     write(*,6080) iout, sngl(Ec),sngl(Ep),sngl(REcEp),sngl(vel)
      endif

C  Check the convergence

      Cx = 2.*xKs*(Tf-Tn(2))*dt/rhos/Hsl
      if(Cx .LT. 0.) Cx = 0.
      Cy = do**2 + 4.*Cx
      dsc = (dsqrt(Cy)+do)/2.

      Ero = (ds - dsc)/ds
      AEro = dabs(Ero)
      Erdn = dabs(ds - dsc)

      if(it .eq. 1 .or. ind .eq. 0) then
c     write(*,6020) sngl(Erdo), sngl(Erdn), sngl(Erp),sngl(Ers)
      endif

      if(Erdn .LE. eps1) go to 30
      if(dsc .GT. ds) then
      dmin = ds
      Erdo = Erdn
      go to 20
      else
      dmax = ds
      Erdo = Erdn
      go to 20
      endif

30    continue

      if(ivoid .eq. 1) then
      Qrin = -Emitin*Boltz*(Tn(4)**4 - Tn(3)**4)
      Qcin = -hcvs*(Tn(4)-Tn(3))
      qrat = Qcin/Qrin
      endif
      Stef = Cps*(Tf - Tn(5))/Hsl
      FC = (sls-slo)/dt*dsqrt(time/As)
      SumLam = SumLam + FC
      do = ds
      slo = sls
      dos = ds
      do 40 i=1,n
      Ts(i)=Tn(i)
      To(i)=Ts(i)
40    continue
      if(iopt .ne. 1) then
      if(it .eq. 1 .or. ind .eq. 0) then

      print *,' '
      print *,' '
      print *,'Sol. converged'
      aQls = dabs(Qls)
      aQlp = dabs(Qlp)
      aQrs = dabs(Qrs)
      write(7,  6010) sngl(time),sngl(Tn(1))
      write(10, 6010) sngl(time),sngl(Tn(2))
c     write(8,  6010) sngl(time),sngl(Tn(3))
      write(9,  6010) sngl(time),sngl(Tn(4))
```

```
c       write(6, 6010) sngl(time),sngl(Tn(5))
        write(11, 6010) sngl(time),sngl(FC)
        write(12, 6010) sngl(time),sngl(vel)
        write(13, 6010) sngl(time),sngl(Ers)
        write(14, 6010) sngl(time),sngl(ds)
        write(15, 6010) sngl(time),sngl(qrat)
        write(16, 6010) sngl(time),sngl(REcEp)
        write(17, 6010) sngl(time),sngl(aQls)
        write(18, 6010) sngl(time),sngl(aQlp)
        write(19, 6010) sngl(time),sngl(aQrs)
        endif
        endif
50      continue

        avFC = SumLam/DFloat(nt)
        write(10,8010)
        write(11,8011) Sngl(Stef)
        write(12,8012)
        write(13,8013)
        write(14,8014)
        write(15,8015)
        write(16,8016)
        write(19,8017)
        go to 9999


9000    continue
C This proc. calculates Average Freezing constant, avgFC, for given time
C period
C       in this proc. Wall Temp., Tw, is varied from 250 K
C        to 450 K for Lithium            ; Tf(Li) = 454 K
C        to 1120 K for Lithium Fluoride  ; Tf(LiF) = 1122 K
        if(iPCM .eq. 1) then
        Tu = 450.
        else
        Tu = 1120.
        endif
        Tl = 250.

        dTe = (Tu-Tl)/25.
        Tlow = Tl - dTe

        do 160 iTw=1,25
        Tw = Tlow + dTe*Dfloat(iTw)
        Ta = Tw
        print *,'Tw = ',sngl(Tw)
c       if(ivoid .eq. 1) then
c       do = 0.00001
c       slo = do
c       else
        do = 0.
        slo = do
c       endif
        dos = do

        do 110 i=1,n
        To(i)=Ti
        Ts(i)=Ti
110     continue

        Hl = Hsl + Cps*Tf
        eps3o = eps3
        epso = eps1
        eps12 = dsqrt(eps1)
        Gam = 1. - rhol/rhos
```

```
        rGam = Gam/(1.-Gam)

        As = xKs/rhos/Cps
        Ac = xKc/rhoc/Cpc
        a1 = rhoc*Cpc*tc/dt
        a2 = rhos*Cps/dt

        hcs = 2.*xKc/tc
        Al = Emitout*Boltz/hcs
        SumLam=0.
        do 150 it=1, nt

        ind = it/npr*npr - it
        time = dfloat(it)*dt
        Erdo = 1.D30
        if(it .eq. 1) then
        dmax = do + 2.*eps3
        else
        dmax = do + 2.*vel*dt
        endif
        dmin = do
        iout = 0

120     continue
        iout = iout + 1

C Assume ds; new solid crust thickness
        ds = (dmax+dmin)/2.
        dvs = rGam*ds
        if( ivoid .eq. 1) then
        sls = ds + dvs
        else
        sls = ds
        endif

C Equivalent heat transfer coefficients
C At the outer wall of the cladding, ho1

        Den = 1./Emitout/Boltz + 4.*Ts(1)**3/hcs
        ho1 = (Ts(1)**2+Ta**2)*(Ts(1)+Ta)/Den

C within the void

        hss = 2.*xKs/ds
        hcvs = xKv/dvs
        hr2s = Emitin*Boltz*(Ts(4)**2 + Ts(3)**2)*(Ts(4) + Ts(3))
c       hv = hcvs + hr2s
        hv = hcvs

        IF(IBC .eq. 1) THEN

C Overall between T1 and T2
C2      This considers heat transfer across the void
        if(ivoid .eq. 1) then
        h12 = 1./(1./hcs + 1./hv + 1./hss)
        else
C BC1   This does not consider heat transfer across the void
        h12 = 1./(1./hcs + 1./hss)
        endif

C BC1 Radiative Boundary
        aw(1) = 0.
        ap(1) = ho1 + h12 + a1
        ae(1) = -h12
        s(1) = a1*To(1) + ho1*Ta
```

```
      aw(2) = -h12
      ap(2) = h12 + hss + a2*ds
      ae(2) = 0.
      s(2) = rhos*Hl/dt*(ds-do) + a2*do*To(2) + hss*Tf

      call tdma(Tn,aw,ap,ae,s,nx)

      dT5 = Al*(Tn(1)**4.-Ta**4)/(1.+4.*Al*Tn(1)**3)
      Tn(5) = Tn(1)-dT5

C BC1  This considers heat transfer across the void
      if(ivoid .eq. 1) then
      Tn(4) = h12*(Tn(1)/hss + (1./hcs + 1./hv)*Tn(2))
      Tn(3) = h12*(Tn(2)/hcs + (1./hss + 1./hv)*Tn(1))
      else
C BC1  This does not consider heat transfer across the void
      Tn(4) = h12*(Tn(1)/hss + 1./hcs*Tn(2))
      Tn(3) = h12*(Tn(2)/hcs + 1./hss*Tn(1))
      endif


      ELSE
C BC2 Constant wall temperature
      Tn(1) = Ta
      Tn(5) = Tn(1)
      Tn(3) = Tn(1)

      if(ivoid .eq. 1) then
C BC3 Constant wall temperature, with void
      h12 = 1./(1./hss + 1./hv)
      else
C BC2 Constant wall temperature, without void
      h12 = hss
      endif

      Tn(2) = (a2*do*To(2) + rhos*Hl*(ds-do)/dt
     & + hss*Tf + h12*Tn(1))/(h12 + hss + a2*ds)

      if(ivoid .eq. 1) then
      Tn(4) = (hv*Tn(1) + hss*Tn(2))/(hss + hv)
      else
      Tn(4) = Tn(1)
      endif

      ENDIF

C Check energy balance of the system
C Heat flux at the left boundary of the system

      Qls = -hol*(Tn(1)-Ta)

C Heat flux at the left boundary of the PCM

      Qlp = -h12*(Tn(2)-Tn(1))
C For constant wall temperature only
      if(iBC .eq. 2) Qls = Qlp

C Heat flux at the right boundary of the system, x=xLB + delta
C which is equal to Heat flux at the right boundary of the solid PCM

      Qrs = -rhos*Hsl*(ds-do)/dt

C Total heat accumulated in the system within dt
c Fully implicit scheme
```

```
      EQp = (Qlp - Qrs)*dt
      EQs = (Qls - Qrs)*dt

C Energy accumulated within the cladding
      if(iBC .eq. 1) then
      Ec = tc*rhoc*Cpc*(Tn(1) - To(1))
      else
      Ec = 0.
      endif

C Energy accumulated within the solid crust
      Ep = rhos*Cps*(ds*Tn(2) - do*To(2))-(ds-do)*rhos*Hl

C Total Energy accumulated in the system
      Ea = Ec + Ep

      Ers = (EQs - Ea)/Ea
      Erp = (EQp - Ep)/Ep

      REcEp = Ec/Ep
      vel = (ds-do)/dt

C  Check the convergence

      Cx = 2.*xKs*(Tf-Tn(2))*dt/rhos/Hsl
      if(Cx .LT. 0.) Cx = 0.
      Cy = do**2 + 4.*Cx
      dsc = (dsqrt(Cy)+do)/2.

      Ero = (ds - dsc)/ds
      AEro = dabs(Ero)
      Erdn = dabs(ds - dsc)

      if(it .eq. 1 .or. ind .eq. 0) then
c     write(*,6020) sngl(Erdo), sngl(Erdn), sngl(Erp),sngl(Ers)
      endif

      if(Erdn .LE. eps1) go to 130
      if(dsc .GT. ds) then
      dmin = ds
      Erdo = Erdn
      go to 120
      else
      dmax = ds
      Erdo = Erdn
      go to 120
      endif

130   continue

      Stef = Cps*(Tf - Tn(5))/Hsl
      FC = (sls-slo)/dt*dsqrt(time/As)
c     FC = sls/2./dsqrt(As*time)
      SumLam = SumLam + FC
c     print *,'It=',it,'    Fc = ',FC

      do = ds
      slo = sls
      dos = ds
      do 140 i=1,n
      Ts(i)=Tn(i)
      To(i)=Ts(i)
140   continue

      if(it .eq. 1 .or. ind .eq. 0) then
```

```
c       print *,' '
c       print *,' '
c       print *,'Sol. converged'
        endif
150     continue

        avgFC = SumLam/DFloat(nt)
        write(11,6010) sngl(Stef),sngl(avgFC)
160     continue

        write(11,8020)

9999    stop

6001    format(2x,'Generate a plot.'/2x,'input data.')
c6002   format(2x,'"','T1','"')
6003    format(2x,'"','T2','"')
c6004   format(2x,'"','T3','"')
6005    format(2x,'"','T4','"')
6006    format(2x,'"','T5','"')
6007    format(2x,'"','Cladding','"')
6008    format(2x,'"','PCM-Void','"')
6009    format(2x,'"','S-L Intf','"')

6010    format(2x,E12.5,2x,e14.7)

6020    format(2x,'Erdo=',E12.5,2x,'Erdn=',E12.5,2x,'Erp=',e12.5,2x,
     &  'Ers=',e12.5)

6080    format(///2x,'iout=',i5/
     &  2x,'Ec=',e12.5,2x,'Ep=',e12.5,2x,'Ratio=',e12.5,
     &  2x,'vel=',e12.5)

8010    format(2x,'end of data.'/2x,'title text is ','"','Temperature',
     &  ', K','"','.'/2x,'go.')

8011    format(2x,'end of data.'/2x,'title text is ','"','Freezing',
     &  ' Constant, St = ',E12.5,'"','.'/2x,'go.')

8012    format(2x,'end of data.'/2x,'title text is ','"','Vsl, M/sec',
     &  '"','.'/2x,'go.')

8013    format(2x,'end of data.'/2x,'title text is ','"','Error in ',
     &  'Energy Balance','"','.'/2x,'go.')

8014    format(2x,'end of data.'/2x,'title text is ','"','Delta, M','"'
     &  ,'.'/2x,'go.')

8015    format(2x,'end of data.'/2x,'title text is ','"','Heat Flux',
     &  ' Ratio inside Void, Qcond/Qrad','"','.'/2x,'go.')

8016    format(2x,'end of data.'/2x,'title text is ','"','Energy Ratio',
     &  ' in the System, dEc/dEs','"','.'/2x,'go.')

8017    format(2x,'end of data.'/2x,'title text is ','"','Heat Flux at',
     &  ' Various Position, W/M**2','"','.'/2x,'go.')

8020    format(2x,'end of data.'/2x,'title text is ','"','Freezing',
     &  ' Constant vs Stefan Number','"','.'/2x,'go.')

        end

c ******************************************************************
```

```
c TRIDIAGONAL MATRIX SOLVER : aWi.X(i-1) + aPi.X(i) + aEi.X(i+1) = Si
c ********************************************************************

      SUBROUTINE TDMA(X, aW,aP,aE,S, N)
        implicit double precision (a-h, o-z)
      DIMENSION X(1), aW(1),aP(1),aE(1),S(1)
      DIMENSION P(1000),Q(1000)
c
c     (  aW(1) = 0.   ,  aE(N) = 0.  )
c
      P(1) = -aE(1)/aP(1)
      Q(1) =    S(1)/aP(1)
c
      DO 10 i=2,N
        DENUM =  aP(i)+aW(i)*P(i-1)
        P(i) =  -aE(i)/DENUM
        Q(i) = ( S(i)-aW(i)*Q(i-1) ) /DENUM
 10   CONTINUE
c
      X(N) = Q(N)
c
      DO 20 i=N-1,1 , -1
        X(i) = P(i)*X(i+1)+Q(i)
 20   CONTINUE
c
      RETURN
      END
```

```
      program freeze
C This program uses dichotomic search for S-L front
C
C This program calculates temperatures T(i), i=1,5
C i=1, outer surface of the wall
C  =2, middle of the solid crust
C  =3, inner surface of cladding
C  =4, PCM-Void interface
C  =5, outer surface of cladding
C
      implicit double precision (a-h, o-z)
      character*8 adum
      dimension Tst(2,2000),Qst(3,2000), Timst(2000), Vst(2000)
      dimension To(5),Ts(5), Tn(5)
      dimension aw(1000), ap(1000), ae(1000), s(1000)
      data Ta/250./, dt/.1/, nt/2000/, Emitin/.8/, Emitout/.8/,
     &  Boltz/5.67D-08/, npr/10/, xSL/1./,iTpr/500/
C LiF properties
     &  Tff/1122./, Tif/1122./, xKsf/4.9997/, xKlf/1.3/,
     &  Cpsf/2469.74/, rhosf/1792./, rholf/1376./, Hslf/347271./,
C Thermal conductivity of LiF vapor
     &  xKvf/0.016954/,
C Li properties
     &  Tf/454./, Ti/454./, xKs/75.8/, xKl/42.8/,
     &  Cps/3780./, rhos/530./, rhol/517./, Hsl/432100./,
C Thermal conductivity of He gas, at 200 F
     &  xKv/0.16781/,
C Properties of cladding material
     &  tc/0.00076/, Cpc/323.4/, rhoc/6490./, xKc/23.885/,
     &  eps1/1.D-12/, eps2/1.75/, eps3/0.01/, tcm/2740./
C Initial size of void, thickness of the system
      data iopt/1/,iPCM/2/, ivoid/1/, iBC/2/, q1/25000./


c     open(6, err=9999, file='T1.dat',  status='unknown')
      open(7, err=9999, file='T2.dat',  status='unknown')
c     open(8, err=9999, file='T3.dat',  status='unknown')
      open(9, err=9999, file='T4.dat',  status='unknown')
      open(10,err=9999, file='T5.dat',  status='unknown')
      open(11,err=9999, file='FC.dat',  status='unknown')
      open(12,err=9999, file='Vel.dat', status='unknown')
      open(13,err=9999, file='Ers.dat', status='unknown')
      open(14,err=9999, file='SL.dat',  status='unknown')
      open(15,err=9999, file='Qrat.dat',status='unknown')
      open(16,err=9999, file='Erat.dat',status='unknown')
      open(17,err=9999, file='Qcld.dat',status='unknown')
      open(18,err=9999, file='Qsol.dat',status='unknown')
      open(19,err=9999, file='Qdel.dat',status='unknown')
      open(20,err=9999, file='Ttot.dat',status='unknown')
      open(21,err=9999, file='Qtot.dat',status='unknown')
c     open(22,err=9999, file='Tpro.dat',status='unknown')
c     open(23,err=9999, file='Tpro.dat',status='unknown')
c     open(24,err=9999, file='Tpro.dat',status='unknown')

      write(11,6001)
      IF(iopt.ne.1) then

      write(7, 6001)
      write(12,6001)
      write(13,6001)
      write(14,6001)
      write(15,6001)
      write(16,6001)
      write(17,6001)
```

Appendix D

List of Computer Program MELT.FOR

```
      program melt
C This program uses dichotomic search for S-L front
C
C This program calculates temperatures T(i), i=1,5
C i=1, middle of the wall
C  =2, middle of the liquid
C  =3, inner surface of cladding
C  =4, PCM-Void interface
C  =5, outer surface of cladding
C
C during a melting processes
      implicit double precision (a-h, o-z)
      character*8 adum
      dimension Tst(2,2000),Qst(3,2000), Timst(2000), Vst(2000)
      dimension To(5),Ts(5), Tn(5)
      dimension aw(1000), ap(1000), ae(1000), s(1000)
      data Ta/1350./, dt/1./, nt/20000/, Emitin/.5/, Emitout/.8/,
     &   Boltz/5.67D-08/, npr/10/, xSL/0.05/,iTpr/500/
C LiF properties
     &   Tff/1122./, Tif/1122./, xKsf/4.9997/, xKlf/1.3/,
     &   Cpsf/2469.74/, rhosf/1792./, rholf/1376./, Hslf/347271./,
C Thermal conductivity of LiF vapor
     &   xKvf/0.016954/,
C Li properties
     &   Tf/454./, Ti/454./, xKs/75.8/, xKl/42.8/,
     &   Cps/3780./, rhos/530./, rhol/517./, Hsl/432100./,
C Thermal conductivity of He gas, at 200 F
     &   xKv/0.16781/,
C Properties of cladding material
     &   tc/0.00076/, Cpc/323.4/, rhoc/6490./, xKc/23.885/,
     &   eps1/1.D-12/, eps2/1.75/, eps3/0.01/, tcm/2740./
C Initial size of void, thickness of the system
      data iopt/0/,iPCM/2/, ivoid/1/, iBC/1/, q1/30000./
      data dvi/0.0001/, imachine/1/
      if(imachine .eq.1) then
      open(6, err=9999, file='TTV.dat',  status='unknown')
      open(7, err=9999, file='TQR.dat',  status='unknown')
      else
      open(6, err=9999, file='T1.dat',  status='unknown')
      open(7, err=9999, file='T2.dat',  status='unknown')
      open(8, err=9999, file='T3.dat',  status='unknown')
      open(9, err=9999, file='T4.dat',  status='unknown')
      open(10,err=9999, file='T5.dat',  status='unknown')
      open(11,err=9999, file='dv.dat',  status='unknown')
      open(12,err=9999, file='Vel.dat', status='unknown')
      open(13,err=9999, file='Ers.dat', status='unknown')
      open(14,err=9999, file='SL.dat',  status='unknown')
      open(15,err=9999, file='Qrat.dat',status='unknown')
      open(16,err=9999, file='Erat.dat',status='unknown')
      open(17,err=9999, file='Qcld.dat',status='unknown')
      open(18,err=9999, file='Qsol.dat',status='unknown')
      open(19,err=9999, file='Qdel.dat',status='unknown')
      endif

1     format(2x,'Generate a plot.'/2x,'input data.')
2     format(2x,'"','T1','"')
3     format(2x,'"','T2','"')
4     format(2x,'"','T3','"')
5     format(2x,'"','T4','"')
6     format(2x,'"','T5','"')
7     format(2x,'"','Cladding','"')
8     format(2x,'"','PCM-Void','"')
9     format(2x,'"','S-L Intf','"')

      if(imachine .ne. 1) then
```

```
      write(6,1)
      write(11,1)
      write(12,1)
      write(13,1)
      write(14,1)
      write(15,1)
      write(16,1)
      write(17,1)

      write(6,2)
      write(7,3)
      write(8,4)
      write(9,5)
      write(10,6)
      write(17,7)
      write(18,8)
      write(19,9)
      endif

      n=5
      nx=2

      if(iPCM .ne. 1) then    ! LiF calculation
C LiF properties
      Tf= Tff
      Ti=Tif
      xKs=xKsf
      xKl=xKlf
      Cps=Cpsf
      rhos=rhosf
      rhol=rholf
      Hsl=Hslf
      xKv=xKvf
      endif


      Hs = Cps*Tf
      Hl = Hsl + Hs
      Cpl = Hl/Tf
      eps3o = eps3
      epso = eps1
      eps12 = dsqrt(eps1)
      Gam = 1. - rhol/rhos
      rGam = (1.-Gam)/Gam

      do = 0.000
      dos = do

C Estimate the thickness of PCM and void
C based on the assumption that the system was originally filled with liquid
C at its fusion temperature

      dvi = xSL*Gam
      xPCM = xSL - dvi
      dvo = dvi

      do 10 i=1,n
      To(i)=Ti
      Ts(i)=Ti
 10   continue

      All = xKl/rhol/Cpl
      As = xKs/rhos/Cps
      Ac = xKc/rhoc/Cpc
      a1 = rhoc*Cpc*tc/dt
```

```
      a2 = rhol*Cpl/dt

      hcs = 2.*xKc/tc
      Al = Emitout*Boltz/hcs

      do 3000 it=1, nt

      ind = it/npr*npr - it
      time = dfloat(it)*dt
      Erdo = 1.D30
      if(it .eq. 1) then
      dmax = do + 3.*eps3
      else
      dmax = do + 3.*vel*dt
      endif
      dmin = do
      iout = 0

100   continue
      iout = iout + 1

C Assume ds; new solid crust thickness
      ds = (dmax+dmin)/2.

C Equivalent heat transfer coefficients
C At the outer wall of the cladding, ho1

      Den = 1./Emitout/Boltz + 4.*Ts(1)**3/hcs
      ho1 = (Ts(1)**2+Ta**2)*(Ts(1)+Ta)/Den

C within the void
      dvs = dvo - (ds-do)*Gam
      if(dvo .lt. 1.d-06) then
      ivoid = 0
      dvs = 1.d-06
      endif
      his = 2.*xKl/ds
      hcvs = xKv/dvs
      hr2s = Emitin*Boltz*(Ts(4)**2 + Ts(3)**2)*(Ts(4) + Ts(3))
      hv = hcvs + hr2s

C Overall between T1 and T2
C2    This considers heat transfer across the void
      IF(iBC .ne. 2) THEN
      if(ivoid .eq. 1)then
      h12 = 1./(1./hcs + 1./hv + 1./his)
      else
C1    This does not consider heat transfer across the void
      h12 = 1./(1./hcs + 1./his)
      endif

C BC1 Radiative Boundary
      aw(1) = 0.
      ae(1) = -h12
      if(iBC .eq. 1) then
      ap(1) = ho1 + h12 + a1
      s(1) = a1*To(1) + ho1*Ta
      else
      ap(1) = h12 + a1
      s(1) = a1*To(1) + q1
      endif
C BC1
      aw(2) = -h12
      ap(2) = h12 + his + a2*ds
      ae(2) = 0.
```

```
      s(2) = rhol*Hl/dt*(ds-do) + a2*do*To(2) + hls*Tf
C BC1
      call tdma(Tn,aw,ap,ae,s,nx)

      if(iBC .eq. 1) then
      dT5 = Al*(Tn(1)**4.-Ta**4)/(1.+4.*Al*Tn(1)**3)
      Tn(5) = Tn(1)-dT5
      else
      Tn(5) = Tn(1) + q1/hcs
      endif

C BC1  This considers heat transfer across the void
      if(ivoid .eq. 1) then
      Tn(4) = h12*(Tn(1)/hls + (1./hcs + 1./hv)*Tn(2))
      Tn(3) = h12*(Tn(2)/hcs + (1./hls + 1./hv)*Tn(1))
      else
C BC1  This does not consider heat transfer across the void
      Tn(4) = h12*(Tn(1)/hls + 1./hcs*Tn(2))
      Tn(3) = h12*(Tn(2)/hcs + 1./hls*Tn(1))
      endif

      ELSE

C BC2 Constant wall temperature, without void
      if(ivoid .eq. 0) then
      h12 = hls
C BC3 Constant wall temperature, with void
      else
      h12 = 1./(1./hls + 1./hv)
      endif

      Tn(1) = Ta
      Tn(2) = (a2*do*To(2) + rhol*Hl*(ds-do)/dt
     & + hls*Tf + h12*Tn(1))/(h12 + hls + a2*ds)

C BC2 Constant wall temperature,
      Tn(5) = Tn(1)
      Tn(3) = Tn(1)
C BC2 without void
      if(ivoid .eq. 0) then
      Tn(4) = Tn(1)
      else
C BC2 with void
      Tn(4) = (hv*Tn(1) + hls*Tn(2))/(hls + hv)
      endif

      ENDIF

C Check energy balance of the system
C Heat flux at the left boundary of the system

      Qls = -ho1*(Tn(1)-Ta)

C Heat flux at the left boundary of the PCM

      Qlp = -h12*(Tn(2)-Tn(1))
C For constant wall temperature heat flux at the outer surface of
C cladding should be equal to that for inner surface of clladding

      if(iBC .eq. 2) Qls = Qlp
      if(iBC .eq. 3) Qls = q1

C Heat flux at the right boundary of the system, x=xLB + delta
C which is equal to Heat flux at the right boundary of the solid PCM
```

```
      Qrs = -hls*(Tf-Tn(2)) - rhol*Hsl*(ds-do)/dt
C for melting process, heat conducted at the RHS of liquid
C only by temperature gradient
      Qrs1 = -hls*(Tf-Tn(2))

C Total heat accumulated in the system within dt
c Fully implicit scheme
      EQp = (Qlp - Qrs)*dt
      EQs = (Qls - Qrs)*dt

C Energy accumulated within the cladding (Temperature variation in the
C cladding is considered only when the radiative and constant heat flux
C boundary conditions)
      if(iBC .ne. 2) then
      Ec = tc*rhoc*Cpc*(Tn(1) - To(1))
C For constant wall temp Ec = 0.
      else
      Ec = 0.
      endif

C Energy accumulated within the liquid PCM
      Ep = rhol*Cpl*(ds*Tn(2) - do*To(2))-(ds-do)*rhol*Hs

C Total Energy accumulated in the system
      Ea = Ec + Ep

      Ers = (EQs - Ea)/Ea
      Erp = (EQp - Ep)/Ep

      REcEp = Ec/Ep
      vel = (ds-do)/dt

c     if(it .eq. 1 .or. ind .eq. 0) then
c     write(*,6080) it,iout,sngl(Ec),sngl(Ep),sngl(REcEp),sngl(vel)
c     endif
c6080 format(///2x,'it=',i5,'                iout=',i5/
c    & 2x,'Ec=',e12.5,2x,'Ep=',e12.5,2x,'Ratio=',e12.5,
c    & 2x,'vel=',e12.5)

C  Check the convergence

      Cx = 2.*xKl*(Tn(2)-Tf)*dt/rhol/Hsl
      if(Cx .LT. 0.) Cx = 0.
      Cy = do**2 + 4.*Cx
      dsc = (dsqrt(Cy)+do)/2.

      Ero = (ds - dsc)/ds
      AEro = dabs(Ero)
      Erdn = dabs(ds - dsc)

c     if(it .eq. 1 .or. ind .eq. 0) then
c     write(*,6020) sngl(Erdo), sngl(Erdn), sngl(Erp),sngl(Ers)
c     endif
c6020 format(2x,'Erdo=',E12.5,2x,'Erdn=',E12.5,2x,'Erp=',e12.5,2x,
c    & 'Ers=',e12.5)

      if(Erdn .LE. eps1) go to 2000
      if(dsc .GT. ds) then
      dmin = ds
      Erdo = Erdn
      go to 100
      else
      dmax = ds
      Erdo = Erdn
      go to 100
```

```
        endif

2000    continue

        if(ivoid .eq. 1) then
        Qrin = -Emitin*Boltz*(Tn(4)**4 - Tn(3)**4)
        Qcin = -hcvs*(Tn(4)-Tn(3))
        qrat = Qcin/Qrin
        endif

        do = ds
        dos = ds
        dvo = dvs
        do 170 i=1,n
        Ts(i)=Tn(i)
        To(i)=Ts(i)
170     continue
        if(it .le. 50 .or. ind .eq. 0 .or. do .gt. xSL) then

c       print *,' '
c       print *,' '
c       print *,'Sol. converged'

        aQls = dabs(Qls)
        aQlp = dabs(Qlp)

c       aQrs = dabs(Qrs)
C for melting process, heat conducted at the RHS of liquid
C only by temperature gradient
        aQrs = dabs(Qrs1)
        if(imachine .eq. 1) then
        write(6, 6200) sngl(time),sngl(Tn(1)),sngl(Tn(2)),sngl(Tn(4)),
     &  sngl(vel)
        write(7, 6200) sngl(time),sngl(aQls),sngl(aQlp),sngl(aQrs),
     &  sngl(Qrat)
        else
        write(7,  6010) sngl(time),sngl(Tn(1))
        write(10, 6010) sngl(time),sngl(Tn(2))
        write(8,  6010) sngl(time),sngl(Tn(3))
        write(9,  6010) sngl(time),sngl(Tn(4))
        write(6,  6010) sngl(time),sngl(Tn(5))
        write(11, 6010) sngl(time),sngl(dvs)
        write(12, 6010) sngl(time),sngl(vel)
        write(13, 6010) sngl(time),sngl(Ers)
        write(14, 6010) sngl(time),sngl(ds)
        write(15, 6010) sngl(time),sngl(qrat)
        write(16, 6010) sngl(time),sngl(REcEp)
        write(17, 6010) sngl(time),sngl(aQls)
        write(18, 6010) sngl(time),sngl(aQlp)
        write(19, 6010) sngl(time),sngl(aQrs)
        endif
        endif
6010    format(2x,E12.5,2x,e14.7)
6200    format(1x,f8.2,1x,4e14.5)
        if(do .gt. xSL)then
        write(*,2990)
2990    format(//2x,'Calculation Terminated on Complete melting of',
     &  ' the System')
         go to 3001
        endif
        if(Tn(1) .GE. tcm) then
        write(*,2991)
2991    format(//2x,'Calculation Terminated on Wall Temp. Exceeds',
     &  ' Melting Temp.')
         go to 3001
```

```
      endif
3000  continue
3001  continue
      if(imachine .ne. 1) then
      write(10,8010)
8010  format(2x,'end of data.'/2x,'title text is ','"','Temperature',
     & ', K','"','.'/2x,'go.')

      write(11,8017)
8017  format(2x,'end of data.'/2x,'title text is ','"','Void',
     & '; Size, M','"','.'/2x,'go.')

      write(12,8011)
8011  format(2x,'end of data.'/2x,'title text is ','"','Vsl, M/sec',
     & '"','.'/2x,'go.')

      write(13,8012)
8012  format(2x,'end of data.'/2x,'title text is ','"','Error in ',
     & 'Energy Balance','"','.'/2x,'go.')

      write(14,8013)
8013  format(2x,'end of data.'/2x,'title text is ','"','Delta, M','"'
     & ,'.'/2x,'go.')

      write(15,8014)
8014  format(2x,'end of data.'/2x,'title text is ','"','Heat Flux',
     & ' Ratio inside Void, Qcond/Qrad','"','.'/2x,'go.')

      write(16,8015)
8015  format(2x,'end of data.'/2x,'title text is ','"','Energy Ratio',
     & ' in the System, dEc/dEs','"','.'/2x,'go.')

      write(19,8016)
8016  format(2x,'end of data.'/2x,'title text is ','"','Heat Flux at',
     & ' Various Position, W/M**2','"','.'/2x,'go.')
      endif

9999  stop
      end


c ********************************************************************
c TRIDIAGONAL MATRIX SOLVER : aWi.X(i-1) + aPi.X(i) + aEi.X(i+1) = Si
c ********************************************************************

      SUBROUTINE TDMA(X, aW,aP,aE,S, N)
        implicit double precision (a-h, o-z)
      DIMENSION X(1), aW(1),aP(1),aE(1),S(1)
      DIMENSION P(1000),Q(1000)
c
c     (  aW(1) = 0.   ,  aE(N) = 0.  )
c
      P(1) = -aE(1)/aP(1)
      Q(1) =   S(1)/aP(1)
c
      DO 10 i=2,N
        DENUM =  aP(i)+aW(i)*P(i-1)
        P(i) =  -aE(i)/DENUM
        Q(i) = ( S(i)-aW(i)*Q(i-1) ) /DENUM
 10   CONTINUE
c
      X(N) = Q(N)
c
      DO 20 i=N-1,1 , -1
        X(i) = P(i)*X(i+1)+Q(i)
```