COMPUTATIONAL TECHNIQUES FOR PREDICTING THE MOTION OF A TRANSIENT SOLIDIFICATION FRONT

I. D. Wedgwood.

Thesis submitted for the Degree of Doctor of Philosophy

of the University of St. Andrews

ABSTRACT

The classical Stefan problem may be described by the immersing of a cold body into an effectively infinite bath of warm liquid, so creating a frost layer and a moving solidification front as the interface between the liquid and the newly formed solid. There are many existing techniques for the solution of such a problem, however no one enjoys universal application. The problem discussed in this thesis has features that render most of the existing techniques inappropriate. There is two way motion of the solidification front; a rapid freezing followed by relatively slow thawing. In addition, the size of the frost layer is small in comparison to the other areas where significant heat transfer is taking place.

The proposed solution is one of a variation of the usual immobilising technique, using different transformations in different regions and mapping the semi-infinite liquid region onto a finite interval. This technique is then compared to a perturbation series solution for a dip soldering problem, as well as in the examination of two special cases with known large time solutions.

As a further comparison to the proposed solution, an investigation is made of a previously developed and commonly used technique, which involves the tracking of the solidification front across a fixed spatial grid. This technique is shown to have certain limitations and inherent inaccuracy, thus justifying the need to develop a new method such as the immobilising technique.

Finally, the special case of immersing a finite cold body in a finite warm liquid bath of the same material is examined. This reduces the problem to a two region configuration, which is solved by both the immobilising technique and the fixed grid method.

I Ian Duncan Wedgwood hereby certify that this thesis has been composed by myself, that it is a record of my own work, and that it has not been accepted in partial or complete fulfilment of any other degree or professional qualification.

Signed

Date 25th August 1993

I was admitted to the Faculty of Science of the University of St.Andrews under Ordinance General No. 12 in October 1990 and as a candidate for the Degree of Ph.D. in October 1991.

Signed .....

Date 25th August 1993

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate to the Degree of Ph.D.

Signature of Supervisor ..... Date 21st Feb 94

In submitting this thesis to the University of St. Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker.

ACKNOWLEDGEMENTS

I would especially like to thank Dr. Graham Bell for his supervision and encouragement throughout the duration of my research and the drafting of this thesis.

I would also like to thank Graeme Dykes and Douglas McAndrew for their much appreciated assistance in the field of chemistry.

Finally, I wish to thank my parents for their support during my University career.

TABLE OF CONTENTS

§3: Solution of the one-dimensional Stefan problem using a fixed grid technique.

§4: An extension of the immobilising transformation technique to freezing and thawing of finite bodies.

APPENDIX - program listings for the algorithms used in this thesis

REFERENCES

Nomenclature list

T	:	temperature
T_0	:	ambient temperature of liquid
T_f	:	freezing temperature of liquid
T_c	:	initial cold body temperature
U	:	transformed temperature
U_B	:	transformed initial cold body temperature
A	:	cross-sectional area of material
τ	:	time
t	:	transformed time, $\frac{\kappa\tau}{a^2}$
x	:	displacement
X,Z	:	transformed displacement, as defined in text
x_0,X_0	:	freeze front displacement
a	:	size of cold body
κ	:	thermal diffusivity
K	:	thermal conductivity
ρ	:	density
L	:	latent heat of fusion
c	:	specific thermal capacity
ϕ	:	ratio of conductivities of cold body and solidified layer
β	:	ratio of conductivities of solidified layer and liquid
η	:	ratio of diffusivities of subscripted regions
α	:	reciprocal of Stefan number for process
$_{1,2,3}$	:	subscripts representing cold body, solidified layer and liquid respectively

§1. Introduction.

§1.1: The classical Stefan problem.

One of the most studied areas in numerical heat transfer is of problems involving a change of phase and thus exhibiting a moving boundary as the interface between the two phases. Clearly the solution of such problems is of great relevance in areas involving melting and solidification, however analogues of such systems lie across a diverse spectrum of disciplines. A discussion of these problems was first published in 1891 by J. Stefan [21] and concerned the melting and freezing of polar ice. It is because of this that problems involving freezing are known as Stefan problems.

A simple system can be described by the immersing of a cold body in a warm liquid bath. If the immersed body is cold enough, then a frost layer begins to form at the interface between the immersed body and the liquid, thus forming a third region. This frozen layer is bounded by the surface of the immersed body and a moving surface of separation of the two phases. This surface, or freeze front as it will be known in this work, may advance and retreat depending on the system studied. Attention is usually focused on the prediction of the position of the freeze front throughout the lifetime of the frost layer. At first sight the problem seems a simple one, however heat is absorbed or liberated at the freeze front as it progresses and the thermal characteristics of the two phases on either side of the front are usually different, so causing great difficulties. The system has been shown to be non-linear [4] so that special solutions must be determined and cannot be superposed.

The most obvious areas of application for solutions to Stefan problems concern the manufacture of castings and the prediction of the effects of winter freezing. The particular problem considered in this thesis may be interpreted as a model of the processes of soldering, welding, tinning and latent heat storage. In particular latent heat storage systems will be considered.

Tinning is the application of a thin layer of material to a component, or wire, in order to facilitate an electrical connection. The tinning process is usually one of simply immersing the cold component into molten metal, thus describing exactly the Stefan

problem given above. Most interest in recent years has focused on problems in the field of civil engineering, such as in the design of roadways in cold climates, as well as studies of ground temperature. It is clearly essential to have numerical techniques available for calculating frost depth or thawing times for frozen soil [9].

Beyond these examples involving phase change / recrystallisation, there exists other diverse applications. In biology for instance, muscle tissue functions with the presence of oxygen. If this oxygen is not present, then a certain degree of anaerobic respiration takes place producing lactic acid† ($CH_3{\cdot}CH(OH){\cdot}COOH$). It becomes necessary to discover just how deep into the tissue this oxygen has diffused, converting the lactic acid into easily managed carbon dioxide and water, in the equation:

$$CH_3{\cdot}CH(OH){\cdot}COOH + 3O_2 \overset{\text{enzyme}}{\rightarrow} 3H_2O + 3CO_2$$

As an analogy to the phase change problem, the presence of either oxygen or lactic acid can be considered as a separate phases and the neutral interface between them a freeze front. In this way, the problem is reduced to one of ascertaining the position of this interface within the muscle tissue, along with the levels of oxygen and lactic acid in each region [43].

Similarly in chemistry, it is possible to react two solids or crystals together as opposed to the usual solution reactions. This area is very important in pharmaceutical manufacture and is known as solid state chemistry. Often yields are higher than the equivalent solution reaction and most important, synthetic routes are possible for which no viable solution synthesis is known. The two reactants are merely placed in contact with one another and a third region is formed at the interface as the product of the reaction. It is obviously necessary to determine the depth of product at any time and the overall lifetime of the reaction.

Significant work has also been made into methods of capturing and storing solar energy [14,30,49] and this is the origin of the problem to be considered here. It is necessary to find a method of storing a large amount of solar heat using as small a size as possible. One approach involves using latent heat as opposed to sensible heat, as latent

† Racemic 2-hydroxy-propanoic acid

heat allows higher energy storage densities than say the heating of rocks or water. The latent heat utilised may be heat of fusion, i.e. melting of solids, or heat of crystallisation, i.e. melting a crystal lattice. If salt crystallisation methods are adopted, then energy is also stored at a constant salt crystallisation temperature which can be matched to that required by the application [14].

Essentially, solar energy heats a block of salt crystal, thus liquefying it by liberation of its water of crystallisation†. In the process the salt gains large amounts of energy, which can then be liberated at a later time by recrystallisation. The salt most commonly used is called Glaubers salt (hydrated sodium sulphate; $Na_2SO_4 \cdot 10H_2O$) also known as *mirabilite* ‡. The storage system uses the dehydration reaction of the salt, thus utilising the latent heat of crystallisation. Since the method relies on the repeated dehydration and rehydration of the crystal form of the salt, it is necessary to eliminate the possibility of the formation of the anhydrous salt (not in crystalline form) which would degrade the system. Thus the salt is used in a hydrated form of 68.2% salt to 31.8% water, giving a stable composition.

After having developed a suitable phase change medium to collect energy from the sun during daylight hours, the next problem lies in how to retrieve the stored energy during darkness. The main difficulty is that crystal growth at discharge occurs at the coldest points in the storage system. So when conventional heat exchange equipment is used, the heat transfer surface becomes blocked by a crystal mantle and poor heat transfer conditions result [14]. Also, all salts supercool to a marked extent, thus giving erratic performance. However, this may be overcome by the introduction of nucleating agents, provided they are not degraded by repeated cycles of crystallisation. There is also a problem since this is a multi-phase heterogeneous system. Due to these problems an efficiency of 20-30% when using Glaubers salt is not uncommon. As will be apparent, the design of latent heat storage systems is a subject of continuing research.

---

† The water present in hydrated compounds. These compounds when crystallised from solution retain a definite amount of water.

‡ Formed in salt lakes, deposited by hot springs or resulting from the action of volcanic gases on sea water.

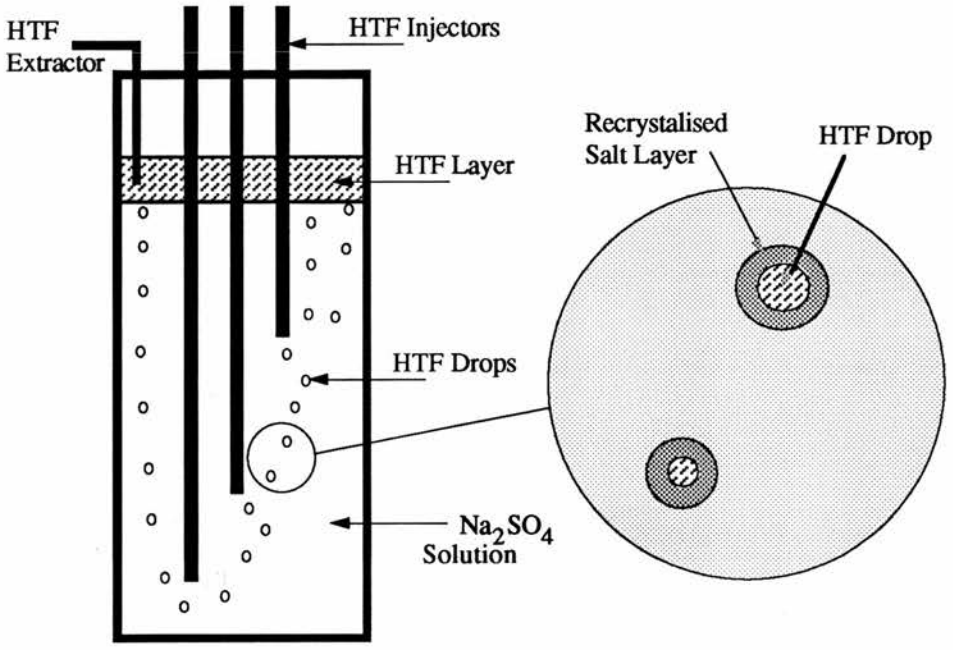

Figure 1.1a: Experimental design of solar storage heat transfer column.

One possible configuration to overcome these difficulties involves an immiscible heat transfer liquid bubbling directly through the hydrated salt to remove the heat from it, see Figure 1.1a. This method is superior to that of conventional heat exchange, since relatively fine crystals are formed under dynamic agitation conditions, preventing the build up of large crystalline balls which would choke the system. This system is also more homogeneous and thus has a higher heat transfer rate.

The heat transfer fluid has to be chosen so as to have certain characteristic properties:

i) it must have low solubility or to be preferably insoluble in the process fluid (Glaubers salt in this case);

ii) it must have no tendency to form stable emulsions with the process fluid;

iii) it must have sufficient specific gravity difference to permit rapid disengagement,

iv) it must have a high heat capacity.

The process column must also be tall enough, so that any crystal growth around each bubble rehydrates before the bubble reaches the extraction point. The bubble size is regulated to be fairly small, in practice up to about 10mm diameter. The three bubble inlet tubes are necessary, since if agitation ceases, crystal growth occurs around the inlets possibly blocking them. Any blockages may require up to a few hours to overcome, so

by having three tubes the chances that at least one bubble inlet remains functional at any time are increased greatly.

It becomes necessary to predict the maximum crystalline growth size and the growth lifetime. This is to prevent choking and to enable extraction of the heat transfer fluid at the optimum time. Thus an algorithm designed to solve the classical Stefan problem provides a direct means of approximating the mantle thickness and lifetime of a particular bubble, provided that the crystal growth remains small. The growth can then be interpreted as essentially one dimensional. The latent heat of fusion constants are exchanged for latent heat of rehydration. If large growth occurs, then a spherical geometry solution would have to be developed, but most methods used to solve the one-dimensional problem are readily transferable to both cylindrical and spherical geometries.

§1.2: Existing numerical techniques.

In attempts to solve Stefan problems, authors have previously tried a profusion of different angles on the concepts involved. Most have their strengths, all have their limitations. Comparative studies of the methods listed below are given in books by Carslaw and Jaeger [4], and Crank [1,12] and also in work by Fox [26] and Furzeland [46].

Of all the techniques tried, probably the best known is the enthalpy method, first described by Eyres et al in 1946 [41]. With this method, instead of working entirely in terms of the temperature of a material, an enthalpy function is defined which represents the total heat content per unit mass of the material. The main advantage is the ease of implementation. There are no conditions to be satisfied at the freeze front, which has no need to be accurately tracked. There is also no need to consider the regions on either side of the freeze front separately. It becomes easy to introduce a 'mushy' region (phase change occurring over a range rather than at a single temperature).

The main problems with this method appear on running the numerical algorithm developed. The method often exhibits unusual behaviour in the numerical solution [7,9]. There are problems with the appearance of rogue plateaux in graphs of freeze front displacement over the lifetime of the frost layer [8, 19]. Attention is usually drawn to the

solution during the freezing stage and thus there is limited investigation of the thawing stage. A comparative study of the various alternative progressions of the method are given by Bonacina [37].

In order to overcome the peculiarities in the numerical solution, authors have tried alternative numerical implementations (Voller [29], Lemmon [51]). Further advancements of this method have been to progress to higher dimensions [36] and also to apply it to a multitude of differing physical problems. M.E. Rose [18] later introduced the concept of weak solution to solve hyperbolic equations involving shocks and Atthey used the method in work on spot welding [42].

It is also possible to utilise this enthalpy technique using finite elements as opposed to finite differences. Work on phase change using this approach was first carried out by Comini *et al* [52] and the corresponding algorithm improved by Morgan *et al* [53]. Rolph and Bathé [56] modified the method by iterating both enthalpy and heat flow in the transient region. Comparative studies of such techniques are given by Bathé [50], Dalhuijsen and Segal [55] and more recently Tamma and Namburu [54].

Another commonly used method is that of the heat balance integral, first proposed in 1966 by Goodman [24]. Essentially, a polynomial relationship consistent with the boundary conditions is assumed for the dependence of the temperature on the space variable. The one-dimensional heat flow equation is integrated with respect to the space variable x and the assumed temperature profile is inserted. This integral equation expresses the overall heat balance of the system. The equation is solved to obtain the motion of the freeze front and the resultant changes in the temperature profile.

The advantages of such a technique is that it is easy to apply to problems involving temperature dependent thermal properties or change of phase. Poots [45] applied this method to problems involving liquids at fusion temperature (effectively single phase problems) using quadratic temperature profiles.

The main difficulty with this method is the sensitivity to the choice of temperature profile, shown in results presented by Langford [35]. A higher order polynomial does not necessarily generate better results than lower order ones [5,26]. It is also difficult to predict the accuracy achievable by a particular profile [20].

Noble [28] suggested that the best way forward for such a method would be repeated spatial subdivision using quadratic profiles in each subregion. Successive subdivision however, only improves the solution slowly and a better rate of improvement would be desirable [20]. The method is essentially limited to the one-dimensional case, however solution of the cylindrical geometry problem is given by Bell [6].

In addition to these methods, a whole host of techniques have been developed utilising front tracking, front fixing, graph construction, series expansions and even mechanical methods.

The isotherm migration method was proposed by Chernousko and independently by Dix and Cizek in 1970 [1]. Here the system is reformulated using x=x(u,t) thus creating isotherms. The partial differential equations describing the system are thus rewritten using this and the new equations are solved by finite difference methods [31].

Douglas and Gallie in 1955 [32] developed a variable time step implicit method such that the moving boundary always coincides with a grid line in space at each time level. This method is limited in that there are no obvious ways to extend it to higher dimensions, however Goodling and Khader [12] develop solutions to the cylindrical and spherical geometry problems. Gupta and Kumar [47] used a difference form of the moving boundary condition to update the time step, so as to avoid instabilities developing as the frost depth increased. The same authors also later applied this work to solving the oxygen diffusion problem [38].

A more obscure technique known as the transmission line method was proposed by Johns in 1977 [23]. Here the system is represented by a graph consisting of nodes and branches, itself a discrete description of the physical process. This configuration is made time discrete by considering pulses travelling between the graph nodes in finite Δt. The branches have characteristic resistance representing thermal diffusivity and / or non-linear capacitance to represent change of phase. This work was later applied to the melting / freezing of ice / water [23].

The use of series expansions of the freeze front position was first described by Evans *et al* in 1950 [33]. Tadjbakhsh and Liniger [2] later generated a similar perturbation expansion for the freeze front position in terms of the Stefan number and compared the

results with those generated by a front tracking method described by Crank [13]. Lock in 1969 [40] applied a wide range of boundary conditions to the method and included an estimate of the errors involved.

Of course the above techniques are merely a few from a selection. Others not described include the method of straight lines [39], variational inequality [27], mechanical [17], the polygonal method [48], transformation to an eigenvalue problem [13], front tracking [13,25] (also examined in §3), relaxation [34], approximation of the Neumann solution [10] and front fixing [3,11] (the basis of work examined in §2 and §4). Nearly all the methods proposed are tested against the problem of the progressive freezing of water where the interface moves as a square root of time. However, each is usually described with some particular problem in mind and there is still no one method that enjoys universal application.

§1.3: A guide to this thesis.

The problem to be discussed in §2 has features that render most of the existing techniques inappropriate. As will be seen there is two way motion at the interface; a rapid freezing followed by relatively slow thawing. In addition, the size of the frost layer is small in comparison to the other areas where significant heat transfer is taking place.

A variation of the usual immobilisation technique is described in §2. The technique developed is compared to a perturbation series solution developed by Tadjbakhsh and Liniger [2] for a dip soldering problem. A further technique is investigated in §3 in way of a comparison, which involves a front tracking method proposed by Crank [13]. Both techniques developed in §2 and §3 are then applied to the special case of a cold finite body immersed in a finite warm liquid of the same material. A brief summary is included as §5.

All computer programs used in this thesis were developed by the author in Sun Pascal v2.1 and run on a Sun SPARCstation1+ (4/65). Graphical output was obtained from UNIRAS v6.2a in EPSF form and 'pasted' into this thesis.

Due to the small size of processors used, the programs developed ran over large time scales, from approximately 3 hours for the immobilisation method developed in §2 to well over 48 hours for the front tracking method given in §3.

§2. A novel solution of the one-dimensional Stefan problem using an immobilising transformation technique.

§2.1: Introduction.

The fundamental aim of this chapter is to formulate a solution to the one-dimensional classical Stefan† problem by developing a novel immobilising transformation technique and solving the resultant equations by using an explicit finite difference numerical algorithm. The Stefan problem is that of immersing a finite cold body in a bath of warm liquid. The immersed body has a temperature lower than the freezing point of the liquid and hence freezes the adjacent liquid, so inducing the growth of a small frost layer. This may further grow or shrink again depending on the relative values of the thermal characteristics of the materials involved. The solution developed should ideally provide a full history of the depth of the frost layer, along with the corresponding temperature profiles across the spatial regions.

The solution generated here will involve the immobilisation of the freeze front using transformation techniques, however in order to model the system most effectively, first the true nature of the problem must be extracted from any inherent extraneous physical information, so maintaining generality as far as possible.

In simple terms, there are three spatial regions representing the immersed cold body, the frost layer and the liquid, each with a temperature profile across it. The boundary between the regions representing the frost layer and the liquid is moving, thus altering the relative size of each region in a continuous process. With the intention of preserving generality, the following physical simplifications are made:

i) all the materials and phases thereof are considered to be homogeneous and thus the thermal characteristics across them are uniform in each region.

ii) a planar (one-dimensional) system is examined, since any aforementioned multidimensional contrivances or spatial complexities require prior investigation of the planar solution. It is the aim of the author that any ensuing solutions may be easily adapted to a higher dimensional system, however formulation of such solutions is not attempted in this work.

---

† Stefan problems are explained in §1

iii) the size of the frost layer is considered small enough with respect to the size of the liquid such that any effects of immersing the cold body in the liquid do not reach the extremities of the liquid bath. It thus becomes practical to examine an essentially infinite liquid region, however the solution of problems involving a finite liquid region is undertaken as an extension of this model in §4.

iv) the existence of the line of symmetry of the system at the centre of the immersed body is exploited by considering only one half space.

v) the densities of the liquid and the frost layer are deemed to be equal (a parameter defined as the ratio of these two densities may be incorporated later), and thus there is no change in volume connected with the transition between the solid and liquid phases.

vi) the interface between the frost layer and the liquid region is considered to be a discrete line by exclusion of any interphase states, indicating that change of phase occurs at a single temperature. Mushy or plastic regions, as these interphase states are usually known, occur most readily when dealing with the melting of alloys as described in §1 and methods here could possibly be applied to these states by considering the mushy region as a separate phase, however this will not be attempted in this work.

viii) such effects as convection currents, mass diffusion and variation of thermal properties with temperature are neglected. restrictive?

Thus the number of governing parameters for the process are reduced and are:

ρ : density of the solid and liquid materials,

κ : thermal diffusivity of the materials,

$T(x,\tau)$: temperature distribution,

x : displacement from a fixed origin,

τ : elapsed time from immersing the cold body,

K : thermal conductivity of the materials

and L : latent heat of solidification† of the liquid.

† The difference in enthalpies (a measure of atomic order) of the substance in its two states. A material may exist in two phases at a single temperature, the difference in internal energies between the two phases being the latent heat of fusion of the material.

The subscripts 1,2,3 used identify the immersed body, the frost layer and the liquid region respectively.

§2.2: Physical configuration.

The physical layout of the system is given in Bell and Wedgwood [15] and is shown in Figure 2.2a.

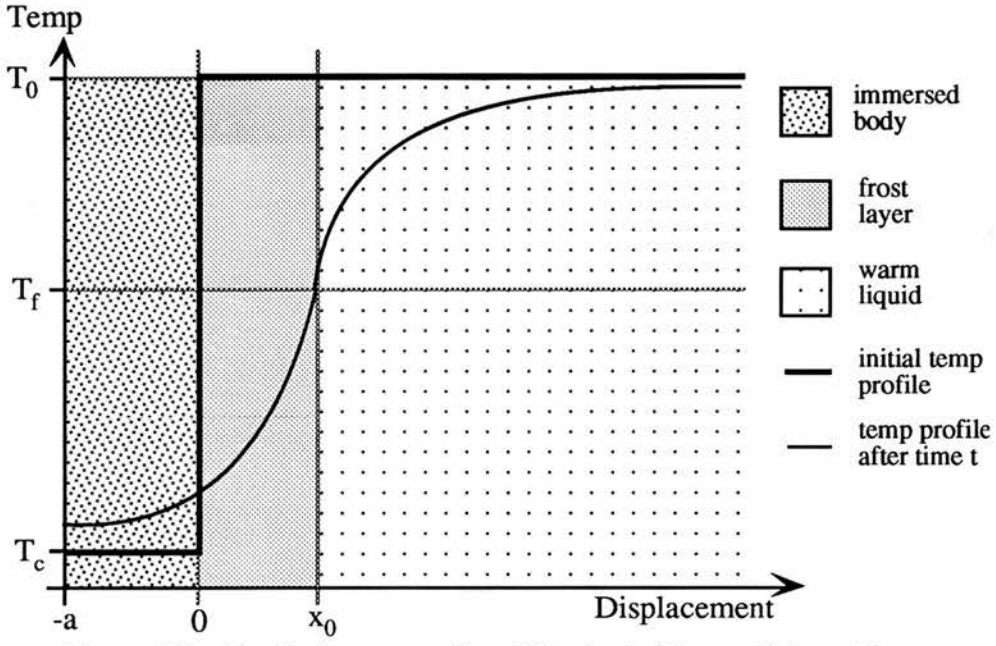

Figure 2.2a: Graphical representation of the physical layout of the problem.

A cold body at temperature T_c is immersed in a warm liquid at ambient temperature T_0, where $T_c < T_f < T_0$ and T_f is the freezing temperature of the liquid. The resulting heat transfer process has three stages.

Firstly, there is a very short-lived solidification stage as the initially cold body freezes the surrounding liquid and a solidified layer of thickness X_0 forms on the surface of the body. Since the warm liquid is of sufficiently large extent, then a second, and much slower, thawing stage occurs where the heat supplied by the liquid to the solidified layer is greater than that being gained by the cold body. The frozen layer melts and the temperature of the cold body continues to increase. This second stage is terminated as the frozen layer disappears and the surface of the cold body reaches the temperature T_f. From that point on there is a pure conduction phase which is relatively easy to solve by traditional methods and is not considered in this thesis.

The difficulty in tackling such melting and freezing problems is deciding how best to treat the moving interface between the solidified layer and the liquid, as the behaviour of the solidified layer differs substantially throughout its existence. The boundary X_0, defined as the displacement from the surface of the cold body, may be left to move across a fixed grid mesh, as in the following chapter, or it may be immobilised by the use of appropriate transformations. The benefits of immobilising the moving boundary are that it is possible to keep a fixed number of grid points within each region, eliminating the need for tracking procedures to ascertain the boundary position at any time. Also, since the boundary is fixed, the regions may be divided into equally spaced sub-intervals eliminating the problem of having to deal with variable mesh spacings necessitating the use of interpolation techniques.

§2.3: Basic equations representing the system.

The mathematics for this system is based on the relevant diffusion equations for the three regions:

$$\frac{\partial T}{\partial \tau} = \kappa_i \frac{\partial^2 T}{\partial x^2} \quad , i=1,2,3 \qquad \textbf{(2.3a)}$$

together with the boundary conditions imposed by consideration of the physical system.

The boundary condition at x=-a, the representation of the centre of the immersed body is dictated by the symmetry of the system about this point. Due to this symmetry there can be no heat flow across this boundary and so the corresponding temperature gradient must also be identically zero:

$$\left.\frac{\partial T}{\partial x}\right|_{-a+} = 0 \quad \text{on} \quad x = -a. \qquad \textbf{(2.3b)}$$

At x=0, the interface between the immersed body and the frost layer, there is no phase change. There is heat flux continuity across the boundary such that any sensible heat

leaving the frost layer must register as sensible heat gained by the immersed body. This gives the boundary condition as:

$$K_1 \left.\frac{\partial T}{\partial x}\right|_{0-} = K_2 \left.\frac{\partial T}{\partial x}\right|_{0+} \quad \text{on } x=0. \qquad \textbf{(2.3c)}$$

Probably the most important feature of the system and indeed the governing factor is the boundary condition at $x=x_0(\tau)$, the interface between the frost layer and the liquid. Here both phase change and flux continuity have to be considered. Any difference in heat flux across the boundary is due to the latent heat of fusion absorbed / released in the change of phase from solid to liquid or vice versa:

$$K_2 \left.\frac{\partial T}{\partial x}\right|_{x_o-} - K_3 \left.\frac{\partial T}{\partial x}\right|_{x_o+} = L\,\rho_2 \frac{dx_0}{d\tau} \quad \text{on } x=x_0 \qquad \textbf{(2.3d)}$$

Also, on this boundary, the temperature is that of the fixed phase change temperature of the frost layer / liquid for all time and so:

$$T[x_0(\tau),\tau] = T_f \ . \qquad \textbf{(2.3e)}$$

The liquid bath occupies the semi-infinite region $x>x_0(\tau)$ and it may be assumed that the temperature as $x\to\infty$ remains as the ambient liquid temperature T_0 for all time, since any effects of immersing the cold body never reach the liquid extremities. Thus:

$$T(x,\tau) \to T_0 \quad \text{as } x \to \infty,\ \tau > 0. \qquad \textbf{(2.3f)}$$

In addition, the initial conditions must be considered. As shown in Figure 2.2a, the immersed body has a uniform temperature field T_c at time $\tau=0$ since the body has had no time to warm after being plunged into the liquid:

$$T[x,0] = T_c \quad -a \le x \le 0. \qquad \textbf{(2.3g)}$$

Furthermore there is initially no frost layer, so the freeze front is at the surface of the immersed cold body, considered to be the origin:

$$x_0(0) = 0. \qquad \textbf{(2.3h)}$$

Finally, the temperature profile of the liquid is uniform at its ambient temperature T_0, since no cooling effects from the immersed body have had time to take place:

$$T(x,0) = T_0 \quad x > 0. \qquad \textbf{(2.3i)}$$

§2.4: Non-dimensionalising.

In order to proceed, the equations (2.2a-i) are non-dimensionalised so as to reduce the number of characterising parameters in the problem, thus simplifying it slightly. This is done by introducing the transformations:

$$U = \frac{T - T_f}{T_0 - T_f} , \quad t = \frac{k_3 \tau}{a^2} \quad \text{and} \quad X = \frac{x}{a} , \qquad \textbf{(2.4a)}$$

where the displacement of the freeze front is now denoted by:

$$X_0(t) = \frac{x_0(t)}{a} .$$

These transformations have the effect of mapping the temperature field from (T_c,T_f,T_0) onto $(U_B,0,1)$ and so any further equations only involve one temperature parameter U_B, defined by:

$$U_B = \frac{T_c - T_f}{T_0 - T_f} . \qquad \textbf{(2.4b)}$$

After these transformations, the physical space becomes that illustrated in Figure 2.4a and the diffusion equations (2.3a) in the immersed cold body, frost layer and liquid bath become:

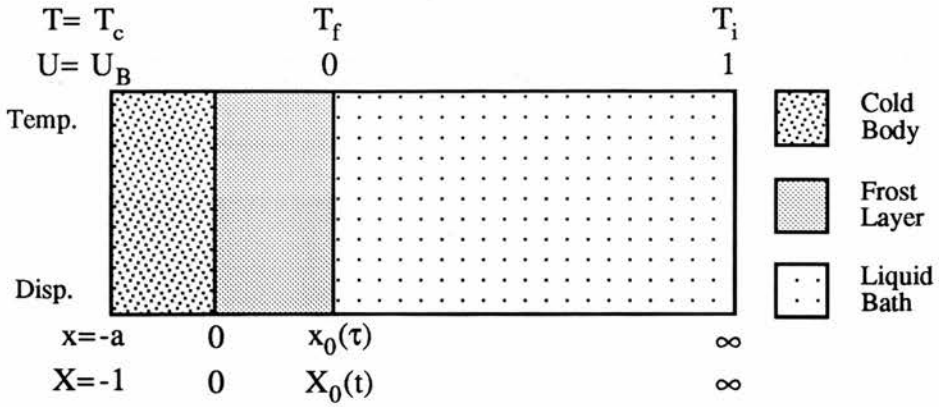

Figure 2.4a: Schematic layout of the system after non-dimensionalising.

$$\frac{\partial U}{\partial t} = \eta_1 \frac{\partial^2 U}{\partial X^2} \quad , \qquad -1 < X < 0 \; ; \tag{2.4c}$$

$$\frac{\partial U}{\partial t} = \eta_2 \frac{\partial^2 U}{\partial X^2} \quad , \qquad 0 < X < X_0 \; ; \tag{2.4d}$$

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial X^2} \quad , \qquad X_0 < X \tag{2.4e}$$

respectively. Similarly the boundary conditions, equations (2.3b-f) transform to become:

$$\frac{\partial U}{\partial X} = 0 \qquad \text{on} \quad X = -1 \; ; \tag{2.4f}$$

at the centre of the immersed cold body,

$$\phi \left. \frac{\partial U}{\partial X} \right|_{0-} - \left. \frac{\partial U}{\partial X} \right|_{0+} = 0 \, , \qquad \text{on} \quad X=0. \tag{2.4g}$$

at the immersed body / frost layer interface, along with

$$\beta \left. \frac{\partial U}{\partial X} \right|_{X_o-} - \left. \frac{\partial U}{\partial X} \right|_{X_o+} = \alpha \frac{dX_0}{dt} \qquad \text{on} \quad X=X_0. \tag{2.4h}$$

and

$$U[X_0(t),t] = 0 \,, \qquad \textbf{(2.4i)}$$

at the moving freeze front $X_0(t)$ and

$$U(X,t) \rightarrow 1 \qquad \text{as } X \rightarrow \infty, \qquad \textbf{(2.4j)}$$

representing uniform temperatures well away from the effects of the heat transfer zone.

The initial conditions, equations (2.3g-i), are similarly transformed into:

$$U(X,0) = U_B \,, \qquad -1 \leq X < 0 \,; \qquad \textbf{(2.4k)}$$

$$U(X,0) = T_0 \,, \qquad 0 < X \,; \qquad \textbf{(2.4l)}$$

and $$X_0(0) = 0 \,. \qquad \textbf{(2.4m)}$$

The dimensionless thermal parameters $\alpha, \beta, \phi, \kappa_i$ present in the above transformed equations are defined as:

$$\alpha = \frac{L}{c_3 (T_0 - T_f)} \,, \quad \beta = \frac{K_2}{K_3} \,, \quad \phi = \frac{K_1}{K_2}$$

and $$\eta_i = \frac{\kappa_i}{\kappa_3} \,, \; i=1,2 \qquad \textbf{(2.4n)}$$

where α is the reciprocal of the Stefan number for the process.

§2.5: Immobilising transform technique.

In the problem as described above, the behaviour of the frost layer differs substantially throughout its existence. In an attempt to adopt a uniform approach, the moving boundary is immobilised by introducing appropriate transformations in the

immersed body and the frost layer. The two equations used are standard immobilisation techniques adopted by many authors and map the physical space (-1,0), $(0,X_0)$ representing the immersed body and frost layer respectively onto the regions (-1,0), (0,1). They are:

$$z_1 = X \qquad \text{in the immersed body,} \qquad \textbf{(2.5a)}$$

and

$$z_2 = \frac{X}{X_0(t)} \qquad \text{in the frozen layer,} \qquad \textbf{(2.5b)}$$

Essentially any difficulty now lies with how to handle the liquid region in the half space $X > X_0(t)$. The technique adopted takes into account the whole semi-infinite liquid region, as opposed to treating it merely as a very large finite region◊. The physical space $(X_0(t),\infty)$ is simply mapped onto the region (1,2) by use of the transformation:

$$z_3 = 2 - \frac{X_0(t)}{X} \; . \qquad \textbf{(2.5c)}$$

This technique, along with the concept of using different transformations in the different regions, is believed to be novel.

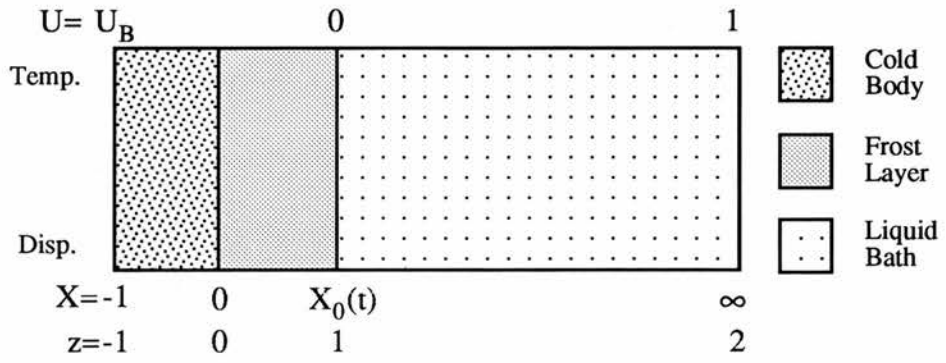

Figure 2.5a: Schematic layout of the system after immobilising transformations.

Under these transformations the physical space becomes that illustrated in Figure 2.5a and the diffusion equations (2.4c-e) become:

◊ The idea being that more points need to be placed in the active region, which an equally spaced set of grid points cannot achieve.

$$\frac{\partial U}{\partial t} = \eta_1 \frac{\partial^2 U}{\partial z_1^2} \quad , \qquad -1 < z_1 < 0 \; ; \tag{2.5d}$$

$$\frac{\partial U}{\partial t} = \frac{\eta_2}{X_0^2} \frac{\partial^2 U}{\partial z_2^2} + \frac{z_2}{X_0} \frac{dX_0}{dt} \frac{\partial U}{\partial z_2} \quad , \qquad 0 < z_2 < 1 \; ; \tag{2.5e}$$

$$\frac{\partial U}{\partial t} = \frac{(2 - z_3)^4}{X_0^2} \frac{\partial^2 U}{\partial z_3^2} + \frac{(2 - z_3)}{X_0} \frac{dX_0}{dt} \frac{\partial U}{\partial z_3} - 2\frac{(2 - z_3)^3}{X_0^2} \frac{\partial U}{\partial z_3} \quad , \qquad 1 < z_3 < 2. \tag{2.5f}$$

The boundary conditions, equations (2.4f-j), transform in a similar manner to give:

$$\left. \frac{\partial U}{\partial z_1} \right|_{-1} = 0 \quad , \qquad \text{at } z = -1 \tag{2.5g}$$

the centre of the immersed cold body ,

$$\phi \left. \frac{\partial U}{\partial z_1} \right|_{0-} - \frac{1}{X_0} \left. \frac{\partial U}{\partial z_2} \right|_{0+} = 0 \, , \qquad \text{at } z_1 = z_2 = 0 \, , \tag{2.5h}$$

the interface of the immersed body and the frost layer,

$$\beta \left. \frac{\partial U}{\partial z_2} \right|_{X_0(t)-} - \left. \frac{\partial U}{\partial z_3} \right|_{X_0(t)+} = \alpha X_0 \frac{dX_0}{dt} \tag{2.5i}$$

and

$$U = 0 \qquad \text{at } z_2 = z_3 = 1 \, , \tag{2.5j}$$

the interface between the frost layer and the liquid, together with:

$$U = 1 \qquad \text{at } z_3 = 2 \, , \tag{2.5k}$$

which corresponds to the liquid remaining at the ambient temperature at distances well away from the solidification zone.

The initial conditions, equations (2.4k-m), are also transformed to become:

$$U = U_B \ , \quad \text{for } -1 \leq z_1 < 0 \, . \qquad \textbf{(2.5l)}$$

$$U = 1 \, , \quad \text{for } 1 < z_3 \leq 2 \, , \qquad \textbf{(2.5m)}$$

and

$$X_0(0) = 0 \ . \qquad \textbf{(2.5n)}$$

Despite removing the difficulty of treating the moving and infinite boundaries, this set of transformations has other consequences. Firstly, the transformed equations are non-linear and involve the unknown variable $X_0(t)$, but this appears to be common to all other such immobilisation techniques and it is usually easier to approximate slightly more awkward equations than to cater for an unknown boundary line. Secondly, since $X_0(0) = 0$ (there is no initially frost layer), the dimensionless temperature U is undefined in the region (0,1) at time t=0. Although this may seem to flaw the solution somewhat, it is in fact no different to the situation in the physical co-ordinates. The mathematical model has a discontinuity at X=0 for time t=0, which corresponds to the sudden temperature change at the surface of the immersed body as it is plunged into the warm liquid. It is therefore necessary, as is common in such situations, to overcome this initial discontinuity by use of a starting solution to define U in the frost layer (0,1).

§2.6: Starting solution.

In order to generate a suitable temperature profile, frost depth and the front speed to be used as a starting solution, the whole system is examined for very small time. For small t, heat transfer is confined to a region close to the surface of the immersed cold body and any effects due to the size of the body may be neglected. A suitable starting procedure is thus the Neumann method as described by Carslaw and Jaeger [4], whereby the dimensionless

temperature at the start time $t=t_0$ consists of error function solutions. The initial motion of the freeze front is thus approximated by:

$$X_0(t) \approx \lambda \sqrt{\eta_2 t} \quad \text{for } 0 < t \leq t_0 , \tag{2.6a}$$

and the corresponding temperature profiles in the three regions are determined from the error function profiles:

$$U_i = A_i \operatorname{erf}\left[\frac{x}{2\sqrt{\eta_i t}}\right] + B_i \qquad i=1,2,3, \tag{2.6b}$$

where the constants A_i and B_i are found by applying the relevant boundary conditions. The parameter λ in equation (2.6a) is found by substituting the newly found temperature profiles, equations (2.6b), into the freeze front condition, equation (2.5i).

With the intention of simplifying this starting solution slightly, it is assumed that if t is small enough, then very little effect, if any, of immersing the cold body into the liquid is initially felt by the body.† This removes a degree of freedom from the starting solution by fixing the temperature profile in the immersed body to be uniform at the initial dimensionless temperature U_B. Thus the relevant temperature profiles in the three regions at the start time t_0 are given by:

$$U_1 = U_B , \qquad -1 \leq x_1 \leq 0 , \tag{2.6c}$$

$$U_2 = U_B \left[1 - \frac{\operatorname{erf}\left\{\frac{x}{2\sqrt{\eta_2 t}}\right\}}{\operatorname{erf}\left\{\frac{\lambda}{2}\right\}} \right] , \qquad 0 \leq x_2 \leq x_0(t) , \tag{2.6d}$$

$$U_3 = \frac{\operatorname{erf}\left\{\frac{x}{2\sqrt{t}}\right\} - \operatorname{erf}\left\{\frac{\lambda\sqrt{\eta_2}}{2}\right\}}{\operatorname{erfc}\left\{\frac{\lambda\sqrt{\eta_2}}{2}\right\}} , \qquad x_0(t) \leq x_2 , \tag{2.6e}$$

† The validity of this assumption is considered in §2.9

where λ is the solution of the transcendental equation gained by substituting these temperature profiles, equations (2.6c-e), into the freeze front condition (2.5i) to give:

$$\frac{\alpha\,\lambda\,\sqrt{\eta_2\pi}}{2} + \frac{\beta\,U_B\,e^{-\lambda^2/4}}{\sqrt{\eta_2}\,\mathrm{erf}(\lambda/2)} + \frac{e^{-\eta_2\lambda^2/4}}{\mathrm{erfc}(\lambda\sqrt{\eta_2}/2)} = 0 \qquad \textbf{(2.6f)}$$

which is solved using a Newton-Raphson technique.

§2.7: Numerical method.

Having developed the set of transformed equations and generated a suitable starting solution, we are able to go ahead and generate the full solution to the problem. This is found by approximating these equations by explicit finite difference formulae and deriving a numerical algorithm to progress the solution over small time increments from the starting profiles. An explicit method is chosen so as to simplify any algorithm used. Any approximated equations should be kept as lucid as possible in order to understand the logistics of the problem and not to lose the essence of the behaviour of the system in complex numerical calculations.

Standard finite difference approximations are adopted in the algorithm. These are:

$$\left.\frac{\partial U}{\partial z}\right|_{z=qh} \approx \frac{U^p_{q+1} - U^p_{q-1}}{2h},$$

$$\left.\frac{\partial U}{\partial t}\right|_{t=pk} \approx \frac{U^{p+1}_q - U^p_q}{k},$$

$$\left.\frac{\partial^2 U}{\partial z^2}\right|_{z=qh} \approx \frac{U^p_{q+1} - 2\,U^p_q + U^p_{q-1}}{h^2}, \qquad \textbf{(2.7a)}$$

$$\left.\frac{\partial U}{\partial z}\right|_{z=qh+} \approx \frac{-3\,U^p_q + 4\,U^p_{q+1} - U^p_{q+2}}{2h},$$

$$\left.\frac{\partial U}{\partial z}\right|_{z=qh-} \approx \frac{3\,U^{p}_{q} - 4\,U^{p}_{q-1} + U^{p}_{q-2}}{2h},$$

where h is the spatial mesh size and k is the time increment representing Δx and Δt respectively. Given the complexity of the transformed equations (2.5d-i) there is little gained from using higher order approximations.

The algorithm used is simply to progress the solution from the starting profile in steps of k in the order:

1. Select a suitable time increment k from the available stability bounds (see below).
2. Update the temperature profiles in the immersed body, frost layer and liquid region from the explicit approximations of the transformed diffusion equations (2.5d-f).
3. Estimate the speed of the freeze front from the moving boundary equation (2.5i).
4. Update the position of the front $X_0(t)$ using the speed calculated in stage '3'.
5. Estimate the new values for temperature U at z = -1 and z = 0 from the fixed boundary equations (2.5g-h).
6. Return to stage 1.

Since explicit approximations are used for the non-linear partial differential equations, it becomes necessary to select the time increment k with some care. In order to prevent the possible development of instabilities, the time step is selected to ensure that cancellation errors are avoided when updating the temperature fields. Thus, in the case of the approximations for the immersed cold body where:

$$\frac{U^{p+1}_{q} - U^{p}_{q}}{k} = \eta_1 \left\{ \frac{U^{p}_{q+1} - 2\,U^{p}_{q} + U^{p}_{q-1}}{h^2} \right\} \qquad \textbf{(2.7b)}$$

if the coefficient of U^{p}_{q} is to be chosen so as to eliminate cancellation errors, the following bound is obtained:

$$\left\{1 - \frac{2\,\eta_1\,k}{h^2}\right\} > 0\,, \qquad \textbf{(2.7c)}$$

which may be satisfied for the correct choice of k. Whence, during the execution of the algorithm, k is required to satisfy the smaller of the following bounds:

$$k < \frac{h^2}{2\eta_1}\,, \qquad \textbf{(2.7d)}$$

$$k < \frac{X_0^2\,h^2}{2\eta_2} \qquad \textbf{(2.7e)}$$

and

$$k < \frac{X_0^2\,h^2}{2}\,, \qquad \textbf{(2.7f)}$$

which arise from the explicit approximations of each of the transformed diffusion equations in the immersed body, frost layer and liquid region respectively.† These are reasonably accurate guides to the time increment available, however they are not as strict as would be desired. A suitable tolerance must be chosen for the bounds, since if k must be less than a certain value k_{bound} then k should be taken as tol*k_{bound} where tol < 1. In most cases tol is taken to be about 0.5 which seems a reasonable enough value to eliminate any instabilities, but keep k as large as possible. The value of $X_0(t)$ is initially very small and hence approximations are very sensitive; the bounds equations (2.7e and f) being the most restrictive depending on the value of η_2. However, since the stability bounds rely on the size of $X_0(t)$, the size of time step k can be adjusted at each cycle of the algorithm in order to achieve a vastly more efficient solution.

Given the novel features of the transformation and the potentially sensitive nature of the approximated equations, the computational procedure is tested for two simpler special cases for which there exists known solutions.

† The spatial grid size h is taken to be constant across the three regions.

§2.8: Continuous freezing - the immersed body temperature held fixed at its initial value.

If the transformed immersed body temperature is taken as fixed at its initial value U_B for all time then the process reduces to that of continuous freezing of a warm liquid.

In this case, the starting solution for the algorithm is the formal mathematical solution, whereby the freeze front position is given by:

$$X_0(t) = \lambda\sqrt{\eta_2 t} \quad \text{for all time,} \qquad \textbf{(2.8a)}$$

with error function temperature profiles in the frost layer and liquid region given by equations (2.6d and e) - see Carslaw and Jaeger [4].

The algorithm is modified to accommodate the exclusion of the immersed cold body by only utilising the transformed equations (2.5e,f,i-n), the boundary condition at the interface between the immersed body and the frost layer, equation (2.5h) being simply replaced by:

$$U = U_B \qquad \text{at } z=0. \qquad \textbf{(2.8b)}$$

So as to test the algorithm, values for the thermal parameters α, β etc. are required and here they are chosen to correspond to those used by Tadjbakhsh and Liniger [2] and represent the process of dip soldering of a copper rod by immersion in a eutectic solder (62% tin, 38% lead) bath. Thus for the algorithm here, the corresponding dimensionless thermal parameters are

$$\left.\begin{array}{llll} \alpha & = 9.7846 & U_B & = -0.48837 \\ \beta & = 2.0377 & \phi & = 8.3519 \\ \eta_1 & = 7.7050 & \eta_2 & = 2.0360 \end{array}\right\} \qquad \textbf{(2.8c)}$$

which give rise to a constant of proportionality in equation (2.6f) of $\lambda = 0.26854$.

The spatial mesh size h is taken to be 0.05 throughout, thus dividing each region into 20 sub-intervals.◊ Due to the value of η_2 being greater than unity, the relevant stability bound is thus equation (2.7e).

◊ This configuration will be used in the majority of the work that follows.

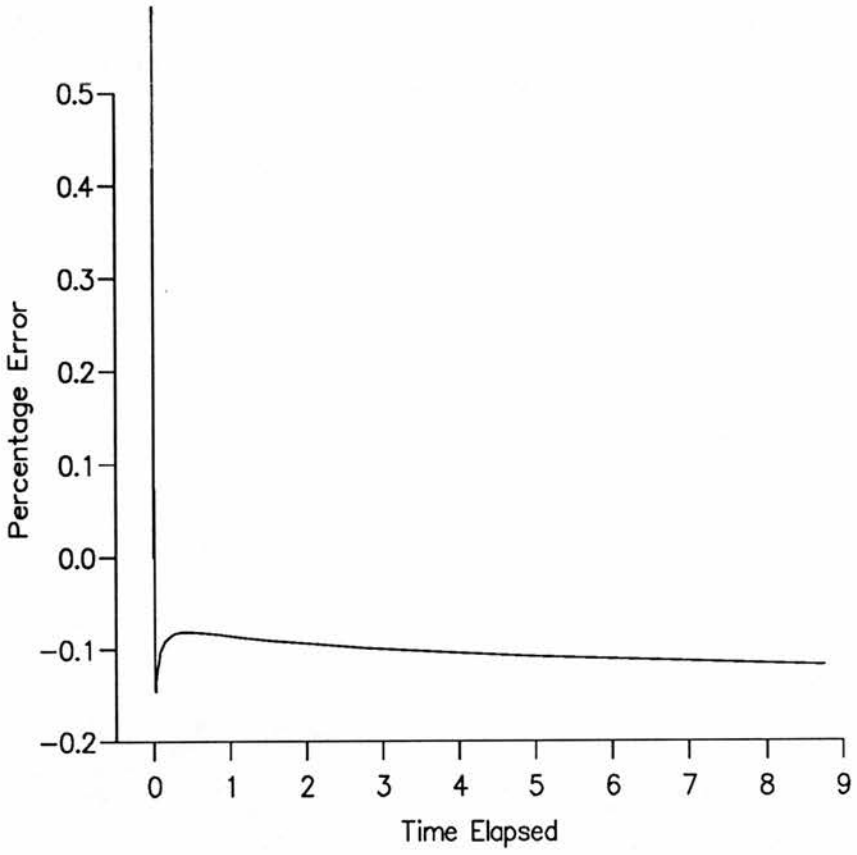

Figure 2.8a: Percentage error in the estimation of the location of the freeze front for the case of continuous freezing for large time.

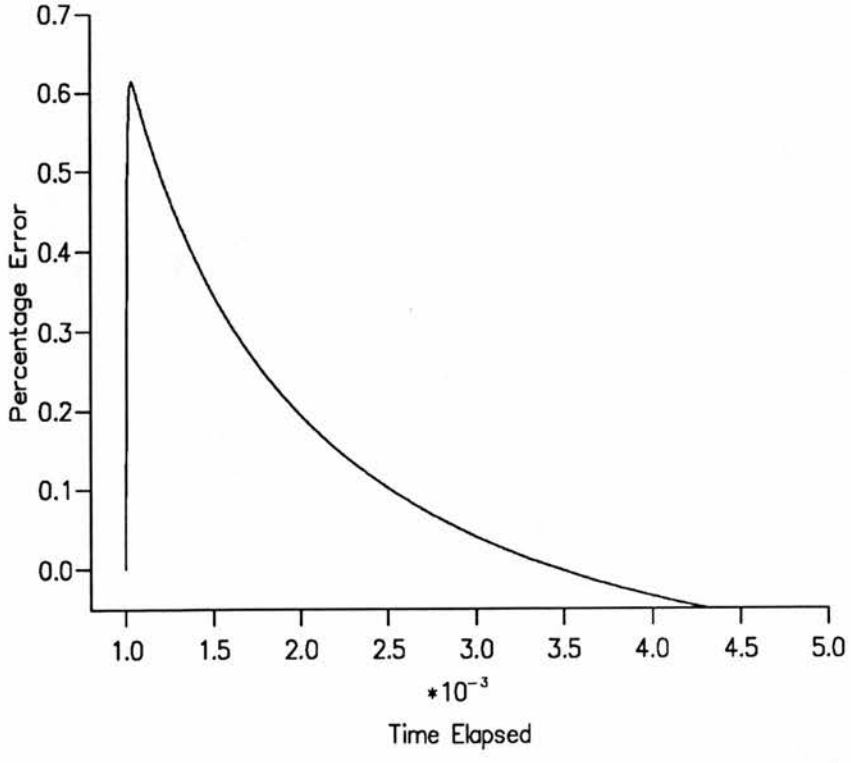

Figure 2.8b: Percentage error in the estimation of the location of the freeze front for the case of continuous freezing for small time.

The algorithm is implemented for parameters (2.8c) and cycled for a large number of time steps. Figures 2.8a and b show the resultant percentage error between the numerically generated front position and the theoretical position given by equation (2.8a), the correlation being strikingly accurate. The initial error, albeit small, quickly dies away and becomes almost negligible (percentage error $\approx 0.1\%$) over a relatively long time span.

This initial error ($\approx 0.6\%$) is probably due to an ill-distribution of spatial mesh points along the early temperature profile, so causing a slight misrepresentation of the true profile, or the inevitable inaccuracy when dealing with a very small X_0 and the drastic size differences between the three regions. However, the rather unusual manoeuvre of mapping a semi-infinite region onto the finite interval (1,2) seems to generate excellent results when the solidification region is growing and the immersed body temperature is held fixed at its initial value.

§2.9: Limiting growth - the liquid temperature initially at its freezing point.

The second special case arises if the temperature of the immersed body is once again allowed to vary from its initial value, but the initial temperature T_i of the liquid region is its freezing temperature T_f. In this case the number of regions to be considered again reduces from three to two. The reason for examining this scenario is that the frost layer should develop to a limiting depth after large time, the steady state occurring when the temperature of the immersed body rises to the freezing point of the liquid.

The limiting thickness of the frost layer can be calculated by considering the net heat transfer between the initial conditions and the steady state, such that the total heat gained by the immersed body must balance with the latent heat released by the liquid during solidification. In the original physical parameters this may be written as:

$$(T_f - T_c) \rho_1 c_1 a A = L \rho_2 x_0 A . \qquad \textbf{(2.9a)}$$

This equation is rearranged to yield a value for the limiting position of the freeze front:

$$\lim_{t\to\infty} \left[\frac{x_0(t)}{a} \right] = \frac{(T_f - T_c) \rho_1 c_1}{L \rho_2} . \qquad \textbf{(2.9b)}$$

The same numerical algorithm as in section §2.8 is to be used to solve this configuration, however the equations have to be adapted slightly, since in this system $T_0 = T_f$ and thus the transformation equation (2.4a) becomes inappropriate. Instead, the modified transformations:

$$U = \frac{T - T_f}{T_f} \quad , \qquad t = \frac{k_2 \tau}{a^2} \tag{2.9c}$$

are used, together with equations (2.5a and b) as before. Under these transformations the diffusion equations now become:

$$\frac{\partial U}{\partial t} = \eta \frac{\partial^2 U}{\partial z_1^2} \quad , \qquad -1 < z_1 < 0 \ ; \tag{2.9d}$$

in the immersed cold body together with

$$\frac{\partial U}{\partial t} = \frac{1}{X_0^2} \frac{\partial^2 U}{\partial z_2^2} + \frac{z_2}{X_0} \frac{dX_0}{dt} \frac{\partial U}{\partial z_2} \quad , \quad 0 < z_2 < 1 \ ; \tag{2.9e}$$

in the frost layer. The boundary conditions at the centre of the immersed body and at the interface between the immersed body and the frost layer, equations (2.5g and h), remain unaltered, however the boundary condition on the moving front now becomes

$$\beta \left. \frac{\partial U}{\partial z_2} \right|_{1-} = \alpha X_0 \frac{dX_0}{dt} \quad . \tag{2.9f}$$

since the spatial derivative of temperature in the liquid bath is zero. The adapted parameters involved in these equations are defined as:

$$\alpha = \frac{L \rho_2 \kappa_2}{K_2 T_f} , \quad \phi = \frac{K_1}{K_2} , \quad \beta = \frac{K_2}{K_3} , \quad \eta = \frac{\kappa_1}{\kappa_2} \ .^{\dagger}$$

† Note that ϕ and β are the same as in the previous section.

Having successfully formulated these equations which are used to generate the numerical algorithm, it is now necessary to formulate a starting solution, as outlined earlier, from error function profiles. Again the starting solution may be simplified if it assumed that the temperature profile of the immersed cold body does not alter appreciably from its initial value if the start time t_0 is small. Thus the starting temperature profile in the immersed body and frost layer are again equations (2.6c and d). The parameter λ is found from substituting equation (2.6d) and $U_3 = 0$ into the moving boundary equation (2.9f) to yield:

$$\alpha \lambda \sqrt{\pi}\, \mathrm{erf}\left\{ \frac{\lambda}{2} \right\} + 2\, U_B\, e^{-\lambda^2/4} = 0. \qquad \textbf{(2.9g)}$$

The numerical algorithm remains essentially the same as for the full three region configuration and the test parameters are again those used by Tadjbakhsh and Liniger [2], giving a transformed parameter regime:

$$\left.\begin{array}{ll} \alpha = 1.3782 & U_B = -0.068852 \\ \phi = 8.3519 & \eta = 3.7845. \\ \lambda = 0.31351 & \end{array}\right\} \qquad \textbf{(2.9h)}$$

Substituting this regime into (2.9b) gives a theoretical limiting solidification depth of:

$$\lim_{t\to\infty}\left[\frac{x_0(t)}{a} \right] = 0.11025. \qquad \textbf{(2.9i)}$$

The numerical algorithm is thus cycled for large time (of the order t = 100) so that the limiting depth is effectively reached, for various start times t_0. The limit is deemed to be reached when the front speed is less than 10^{-9}. It is expected that the size of t_0 should not have a dramatic effect on the limiting depth provided that it is reasonably small. The resultant frost depths are listed for varying start times t_0 in Table 2.9a. Clearly any assumption that the starting time t_0 has little effect on the final outcome is misguided. In fact the error is divided roughly by a factor of $\sqrt{10}$ as the start time t_0 is divided by a factor of 10. This obviously leads to the questioning of just how significant is the start time t_0 and why should it have such a drastic effect on this comparatively simple configuration.

Start Time t_0	Limiting Depth $\frac{x_0}{a}$	Percentage Error
1×10^{-1}	0.21233	92.6
1×10^{-2}	0.14276	29.4
1×10^{-3}	0.12072	9.49
1×10^{-4}	0.11374	3.16
1×10^{-5}	0.11152	1.15
1×10^{-6}	0.11082	0.519
1×10^{-7}	0.11060	0.318
1×10^{-8}	0.11053	0.254

Table 2.9a: The effect of altering the start time t_0 on the limiting depth of the frost layer for the special case $T_0 = T_f$.

Effectively, the difficulty that has arisen is how to handle the trade-off between the accuracy of the solution and the computer efficiency when generating this solution. The later the start time chosen, the larger $X_0(t)$ and thus the larger the available time increment and the greater the efficiency, but this gives less accurate solutions, since the starting solution obviously becomes a less reliable approximation to the temperature profiles at time t_0. In order to overcome this difficulty, the starting solution may be modified, since the assumption that the temperature profile in the immersed body varies negligibly in the time $t < t_0$ may be unfounded.

If the starting immersed body temperature is allowed to vary from its initial value, then consider the modified starting solution with error function profiles:

$$U_1 = (U_0 - U_B)\,\mathrm{erf}\left\{\frac{x}{2\sqrt{\eta\, t}}\right\} + U_0\,, \quad -1 \le X \le 0, \qquad \textbf{(2.9j)}$$

$$U_2 = U_0\left[1 - \frac{\mathrm{erf}\left\{\frac{x}{2\sqrt{\eta\, t}}\right\}}{\mathrm{erf}\left\{\frac{\lambda}{2}\right\}}\right], \quad 0 \le X \le X_0(t), \qquad \textbf{(2.9k)}$$

in the immersed body and frost layer respectively, where the temperature U_0 at the interface between the two regions is given by:

$$U_0 = \frac{\phi\, U_B\, \mathrm{erf}\left\{\frac{\lambda}{2}\right\}}{\phi\, \mathrm{erf}\left\{\frac{\lambda}{2}\right\} + \sqrt{\eta}}\,.^{\dagger} \qquad \textbf{(2.9l)}$$

The parameter λ is now found by solving the transcendental equation gained by substituting these profiles into the transformed moving boundary condition (2.9f) to give:

$$\frac{\alpha\, \lambda\, \sqrt{\pi}}{2} + \frac{\phi\, U_B\, e^{-\lambda^2/4}}{\phi\, \mathrm{erf}\left\{\frac{\lambda}{2}\right\} + \sqrt{\eta}} = 0\,. \qquad \textbf{(2.9m)}$$

Again the algorithm is run for large time and yields results as in Table 2.9b. Clearly the adapted starting solution is much better and the error for larger starting times is much improved.

Start Time t_0	Limiting Depth $\frac{x_0}{a}$	Percentage Error	Number of Time Steps x 10^6	Points Affected In Cold Body
1 x 10^{-1}	0.10164	7.81	0.90	21
1 x 10^{-2}	0.10522	4.56	1.00	21
1 x 10^{-3}	0.11163	1.25	1.10	12
1 x 10^{-4}	0.11221	1.78	1.25	4
1 x 10^{-5}	0.11104	0.717	1.35	1
1 x 10^{-6}	0.11067	0.381	1.50	1
1 x 10^{-7}	0.11055	0.274	1.65	1
1 x 10^{-8}	0.11052	0.240	1.80	1

Table 2.9b: The effect of altering the start time t_0 on the limiting depth of the frost layer using the modified starting solution for the special case $T_0 = T_f$.

As the start time is decreased, the estimated large time position of the freeze front (column 2) converges towards the theoretical value given by equation (2.9b) with a corresponding increase in the computation required (column 4). The last column in Table 2.9b shows a representation of the starting distribution of temperatures across the mesh points in the immersed cold body at $t = t_0$. If the temperature at $t = t_0$ differs by at

† It is interesting to note that this value is independent of the start time t_0.

large the temperature at all 21 mesh points are affected and the starting solution is invalid, since t_0 is stipulated to be such that heat transfer is confined to a region close to the surface of the cold body. However, if t_0 is too small then only the boundary temperature differs from T_c. Given the mesh spacing of $h = 0.05$ and the parameters used, a value $t_0 = 0.001$ is a realistic compromise. However for any further starting solutions developed, the initial temperature distribution across the mesh points and the start time t_0 itself should be a matter of close scrutiny.

§2.10: Full three region problem.

Having validated the computational procedure in terms of the previous two special cases the full three region problem may be tackled with some confidence. After examining the consequences of using a modified starting solution in the previous section, the full problem starting solution is adapted to cater for a change in the immersed body temperature profile by using the error function profiles:

$$U_1 = (U_0 - U_B)\,\mathrm{erf}\left\{\frac{x}{2\sqrt{\eta_1 t}}\right\} + U_0\,, \quad -1 \le X \le 0, \qquad \textbf{(2.10a)}$$

$$U_2 = U_0\left[1 - \frac{\mathrm{erf}\left\{\frac{x}{2\sqrt{\eta_2 t}}\right\}}{\mathrm{erf}\left\{\frac{\lambda}{2}\right\}}\right]\,, \quad 0 \le X \le X_0(t), \qquad \textbf{(2.10b)}$$

$$U_3 = \frac{\mathrm{erf}\left\{\frac{x}{2\sqrt{t}}\right\} - \mathrm{erf}\left\{\frac{\lambda\sqrt{\eta_2}}{2}\right\}}{\mathrm{erfc}\left\{\frac{\lambda\sqrt{\eta_2}}{2}\right\}}\,, \quad X_0(t) \le X \qquad \textbf{(2.10c)}$$

where U_0 is the temperature on the interface between the immersed body and the frost layer:

$$U_0 = \frac{\phi\, U_B \,\mathrm{erf}\left\{\frac{\lambda}{2}\right\}\sqrt{\eta_2}}{\phi\,\mathrm{erf}\left\{\frac{\lambda}{2}\right\}\sqrt{\eta_2} + \sqrt{\eta_1}} \quad \dagger \qquad \textbf{(2.10d)}$$

As before, the parameter λ is calculated from the moving boundary condition such that:

$$\frac{\alpha\,\lambda\sqrt{\pi\,\eta_2}}{2} + \frac{\beta\,\phi\,U_B\, e^{-\lambda^2/4}}{\phi\,\mathrm{erf}\left\{\frac{\lambda}{2}\right\}\sqrt{\eta_2} + \sqrt{\eta_1}} + \frac{e^{-\lambda^2\eta_2/4}}{\mathrm{erfc}\left\{\frac{\lambda\sqrt{\eta_2}}{2}\right\}} = 0\,. \qquad \textbf{(2.10e)}$$

The parameter regime adopted is given in (2.8c) with a spatial grid of h=0.05 for all three regions.

Difficulties immediately begin to arise, since if the start time t_0 is taken to be $t_0 = 0.001$ (an adequate decision in the limiting depth scenario), then the temperature profiles across the three regions are not as expected. The number of points affected in region 1, the immersed body, is much less than anticipated. Even to 15 significant figures no points are changed from the starting temperature U_B.

Start Time	Initial Front Position	Initial Front Speed	Time to Stat. Point	Maximum Depth
0.02	0.02114	0.5286	0.2350	0.07020
0.01	0.01495	0.7475	0.2140	0.06476
0.005	0.01057	1.057	0.1995	0.06020
0.001	0.004728	2.364	0.1805	0.05211
0.0001	0.001495	7.475	0.1710	0.04724
0.00001	0.0004728	23.64	0.1701	0.04669

Table 2.10a: The effects of altering the start time on the position of the stationary point.

Judging from the little change in affected points for various start times, it would be straightforward to assume that little effect, if any, is felt by the full solution for varying t_0. However, Figure 2.10a shows the freeze front history for various start times and illustrates a drastic change in maximum depths achieved for various t_0. These changes are listed in

† Again this must be recognised as being independent of the start time t_0 which appears to be inappropriate, however this shortfall is overlooked for the time being.

Table 2.10a. It is thus obvious that finding the correct full solution may hinge on determining an optimum start time t_0. Since the root cause of any behavioural change of the system seems unlikely to be stemming from the number of points affected in the immersed cold body, attention must be turned elsewhere.

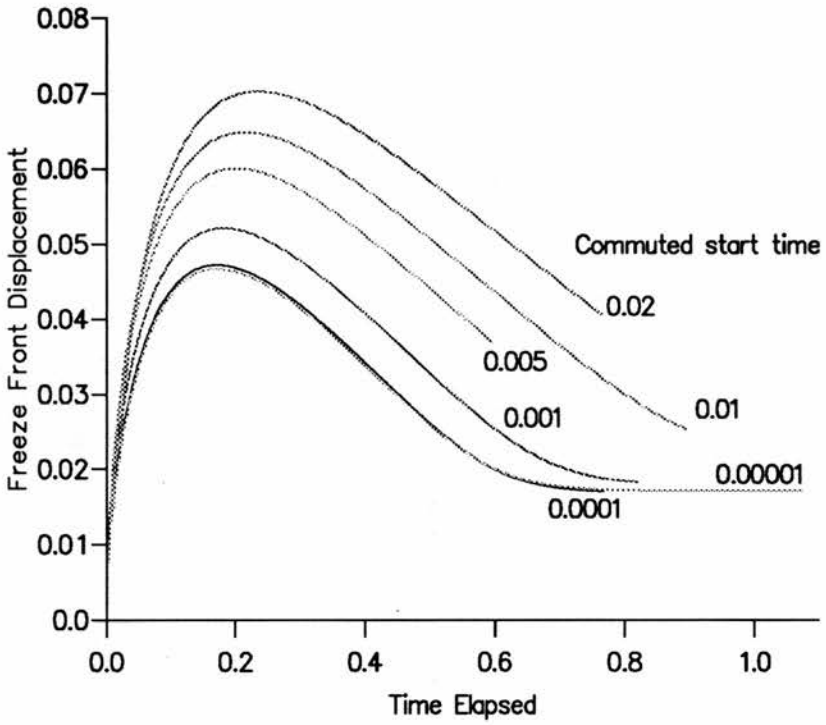

Figure 2.10a: Freeze front position histories for various start times t_0.

The mesh point distribution in the frost layer is determined by:

$$X = X_0 \, h \, i \quad , \qquad i = 1..20 \qquad \textbf{(2.10f)}$$

which essentially states that no matter what value of start time is taken (and thus the corresponding value of X_0), the mesh points always have the same linear distribution across the temperature profile. Similarly in the liquid region the mesh points are determined by:

$$X = \frac{X_0}{(2 - h \, i)} \quad , \quad i = 20..40 \qquad \textbf{(2.10g)}$$

and this again gives the same relative distribution, in this case non-linear, of mesh points along the temperature profile no matter what the value of t_0.

The solution to this dilemma lies in the essence of the behaviour of the system as a whole. By examining the whole system for certain t_0 it is not the immediate effect of switching from the starting solution to the numerical solution that governs the final result, moreover it is the way that the whole system evolves for a short time after the transition. Examination of differing starting temperature profiles fails to give significant information on the would be net effect of switching to the numerical solution. So in order to determine an optimum t_0, the numerical solutions are generated for, say, a few thousand time steps after the transition, for various t_0 and the results compared.

An optimum starting solution would essentially be one whereby the transition to the numerical solution is as smooth as possible. Examination of the freeze front position over the transition yields very little, however examination of the front speed reveals the crux of the matter.

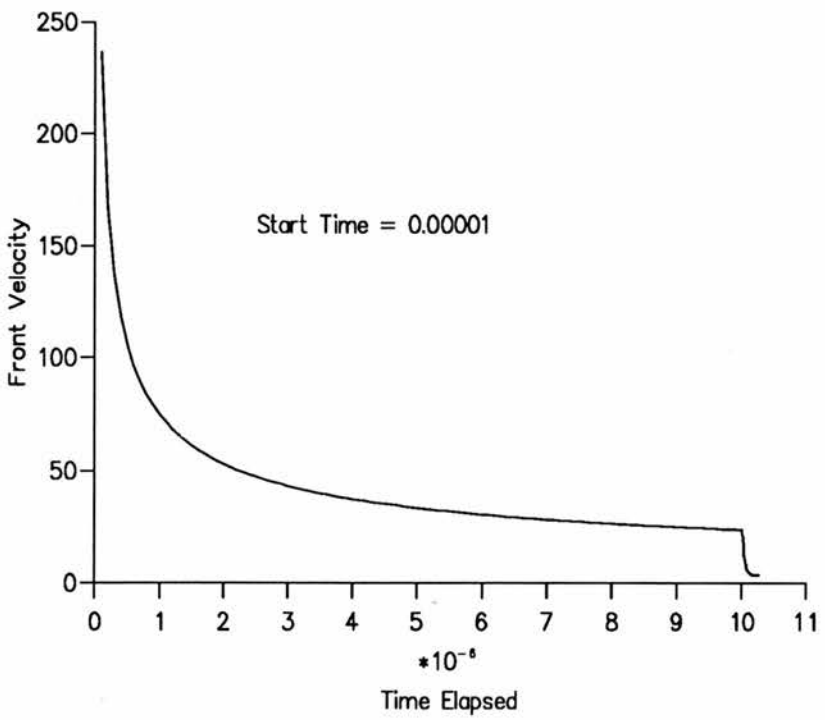

Figure 2.10b: Transition from starting solution to numerical solution for front velocity for start time t_0=0.00001.

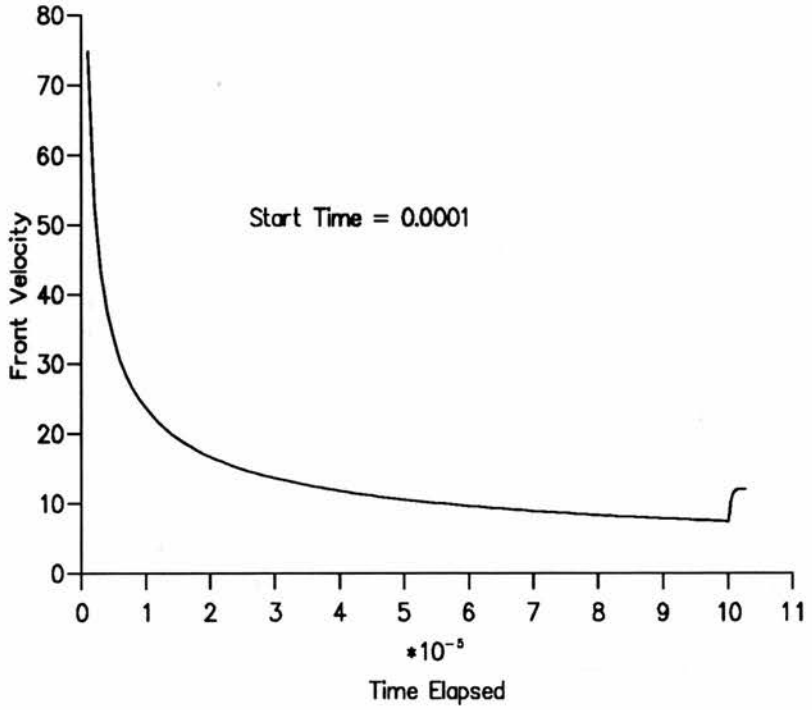

Figure 2.10c: Transition from starting solution to numerical solution for front velocity for start time t_0=0.0001.

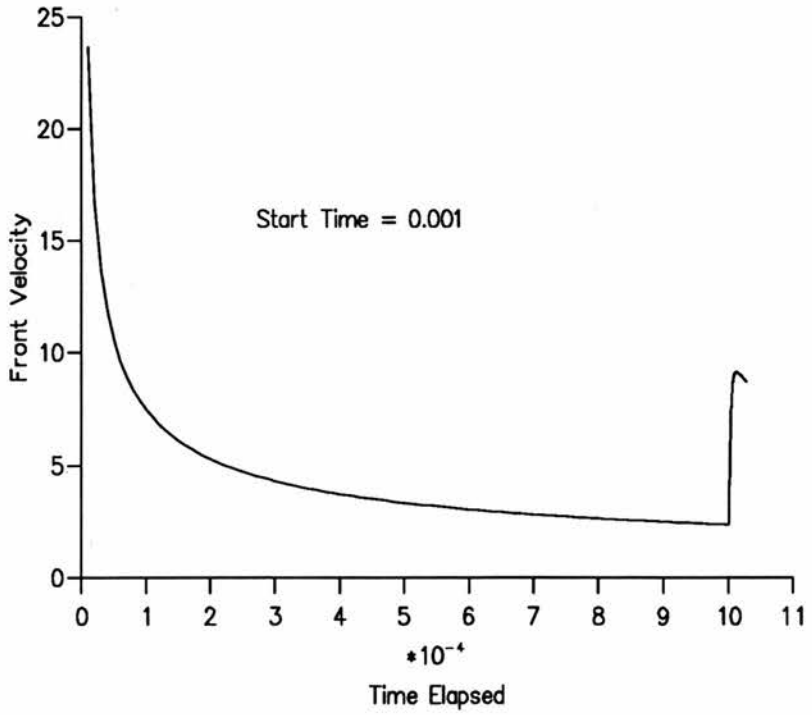

Figure 2.10d: Transition from starting solution to numerical solution for front velocity for start time t_0=0.001.

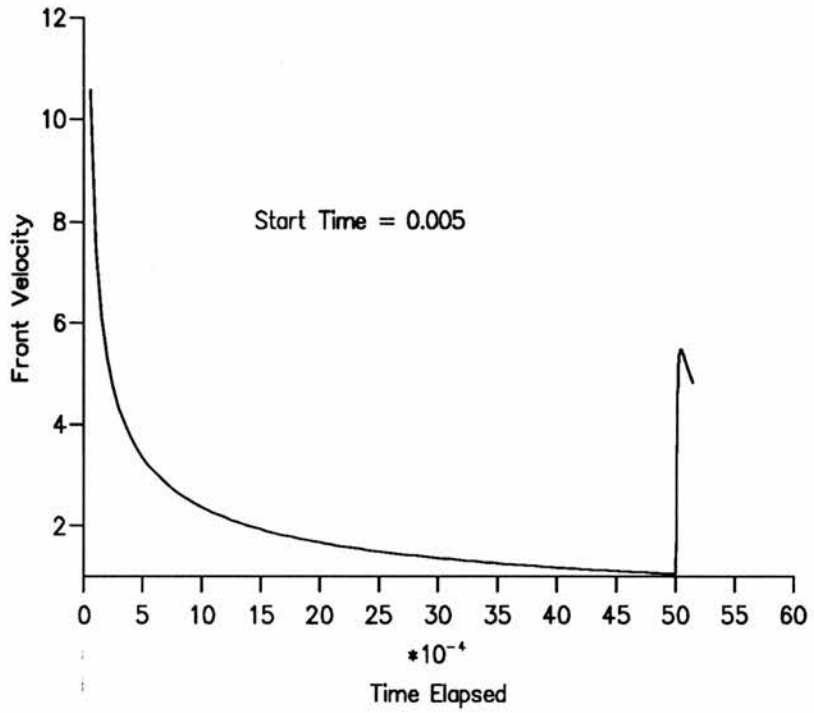

Figure 2.10e: Transition from starting solution to numerical solution for front velocity for start time t_0=0.005.

Figures 2.10b-e show the transition from the starting solution to the numerical solution for the freeze front velocity for various start times. Obviously the transitions for $t_0 > 0.0001$ are not smooth and thus it would be logical to assume that results generated from these t_0 would be inaccurate. However for $t_0 = 0.0001$ and $t_0 = 0.00001$, the transitions are far smoother in front velocity and thus full front histories generated from these start times would sensibly be deemed to be more accurate. This theory is upheld by re-examination of Figure 2.10a where for $t_0 > 0.0001$ the freeze front histories are inconsistent. However for the smoother transitions of $t_0 = 0.0001$ and $t_0 = 0.00001$ the curves match well.

The obvious next stage is to ask why this should occur. From a physical point of view, the starting solution is only valid for small time and thus it would seem logical that the smaller the start time, the better the result. However, for very small time, the frost layer becomes extremely small and the discretisation process becomes inaccurate; the machines available for processing unable to cope with the necessity to manipulate such small

quantities and vastly differing relative dimensions of the regions involved. Thus a limit occurs whereby the accuracy cannot be improved and in fact gets slowly worse. It should be noted at this time that the starting value of U_0 is independent of the starting time which is obviously incorrect. The immersed cold body remains too cold by not allowing change in U_0 which implies a larger overall life span of the frost layer as well as a greater maximum depth achieved.

It seems a rational assumption to suggest by the overall match of the freeze front histories for $t_0 = 0.0001$ and $t_0 = 0.00001$ that these results are the best available and thus, with efficiency in mind, an optimum start time would be to use $t_0 = 0.0001$ in the work that follows for this particular set of parameters. Thus the freeze front lifetime utilised is that illustrated in Figure 2.10f, showing the moving front position throughout the growing phase and beyond.

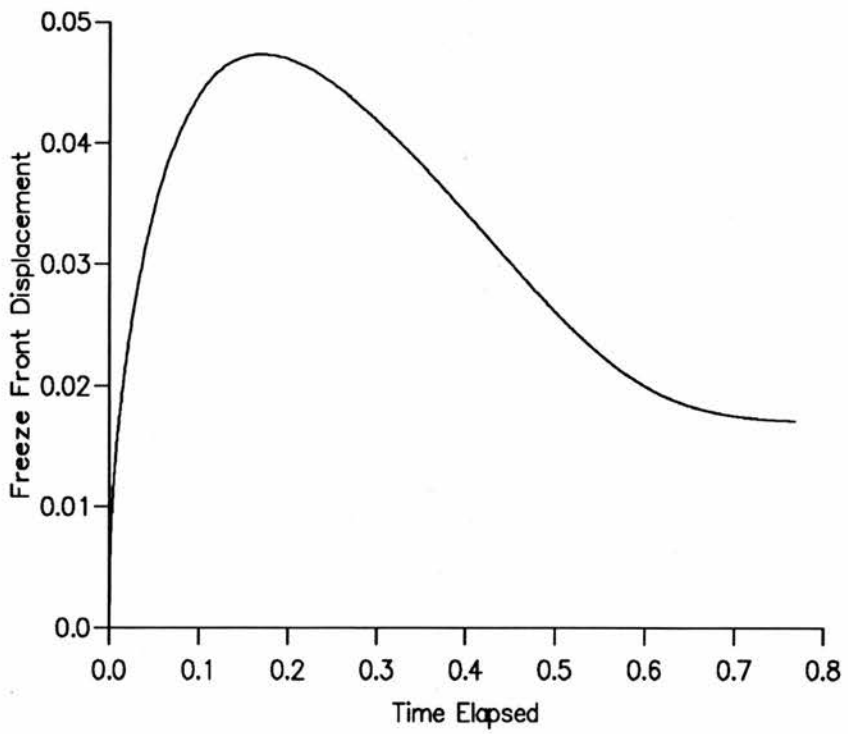

Figure 2.10f: Freeze front history for optimum start time $t_0 = 0.0001$ for parameters (2.8c) during the growing phase and beyond.

As expected the front moves out to a maximum depth and then begins to return, however it fails to return completely to its initial position at $X_0(t) = 0$. This is obviously

incorrect; the infinite heat reservoir of the warm liquid region must inevitably force the front back to its initial position eradicating the frost layer completely. Nevertheless, the existence of a uniform motion to a stationary point and the initial returning is promising and it is these characteristics which are tested against other authors' results. The comparison chosen is that of an asymptotic expansion for the front position $X_0(t)$ by Tadjbakhsh and Liniger [2] as described below.

§2.11: Single term asymptotic expansion for the front position X_0.

Tadjbakhsh and Liniger [2] develop an asymptotic expansion for the freeze front position $X_0(t)$. The first term only of this expansion is easily found in terms of a small parameter ε, such that $X_0(t)$ is given by:

$$X_0 = \varepsilon\,\sigma = \varepsilon\,(\sigma_0 + \varepsilon\,\sigma_1 + \varepsilon^2\,\sigma_2 + ...) \quad , \qquad \textbf{(2.11a)}$$

where $$\varepsilon = \frac{(1 - \beta U_B)}{\eta_2\,\alpha} \quad ,$$

and all other parameters are defined as before. The term ε is the ratio of the sum of initial temperature differences between the two media and the phase change temperature T_f to the latent heat of fusion. Thus it is essentially a representation of the Stefan number for the problem.

To gain a first order approximation, an expression for term σ_0 must be determined and doing so yields:

$$\sigma_0 = \frac{\varphi\,F(\tau)}{\mu} - \frac{2\,(1 - \varphi)\,\sqrt{\tau}}{\sqrt{\pi\kappa}} \quad , \qquad \textbf{(2.11b)}$$

where $$\varphi = \frac{-\beta U_B}{1-\beta U_B} \quad \text{and} \quad \mu = \frac{K_2}{K_1}\sqrt{\frac{\kappa_1}{\kappa_2}} \quad . \qquad \textbf{(2.11c)}$$

The function $F(\tau)$ is the Theta function for zero argument and is given by:

$$F(\tau) = \frac{8}{\pi^2} \sum_{n=0}^{\infty} \frac{1}{(2n+1)^2} \left\{ 1 - \exp\left[- \frac{(2n+1)^2 \pi^2 \tau}{4} \right] \right\} . \qquad \textbf{(2.11d)}$$

Since for the numerical solution developed thus far the term τ has order 1, terms above n=0 in equation (2.11d) are negligible so giving:

$$F(\tau) \approx \frac{8}{\pi^2} \left\{ 1 - \exp\left[\frac{-\pi^2 \tau}{4} \right] \right\} . \qquad \textbf{(2.11e)}$$

In order to establish the position and time of the stationary point; X_{max} and $\tau(X_{max})$ it is necessary to solve:

$$\frac{d\sigma_0}{d\tau} = 0,$$

which from equation (2.11b) gives:

$$\frac{\varphi \, f(\tau)}{\mu} - \frac{(1 - \varphi)}{\sqrt{\pi \kappa \tau}} = 0 \qquad \textbf{(2.11f)}$$

where the term $f(\tau)$ is merely the derivative of $F(\tau)$ such that:

$$f(\tau) = \frac{dF(\tau)}{d\tau} \approx 2 \exp\left\{ \frac{-\pi^2 \tau}{4} \right\}, \qquad \textbf{(2.11g)}$$

for τ of the order 1. Substituting equation (2.11g) back into equation (2.11f) gives an expression for τ to be solved giving a value for τ_{max}:

$$\frac{2\varphi}{\mu} \exp\left\{ \frac{\pi^2 \tau}{4} \right\} - \frac{(1 - \varphi)}{\sqrt{\pi \kappa \tau}} = 0 . \qquad \textbf{(2.11h)}$$

Solving equation (2.11h) yields the time to the stationary point τ_{max} and this value is substituted back into equation (2.11b) along with equation (2.11c) to give a value for σ_0 and thus a maximum frost depth X_{max} from equation (2.11a).

Table 2.11a shows a comparison of the estimated location of the stationary point for the numerical solution and the asymptotic solution.

	X_{max}	Turn Time
Computed	0.04724	0.1710
Ref [2]	0.06248	0.1685

Table 2.11a: Comparison of estimated location of the stationary point for the numerical solution and the asymptotic solution.

The estimates for the turn time match very well (only about 1.5% difference) which is encouraging, however the maximum depth estimates are not in good agreement (approximately 24% difference). Tadjbakhsh and Liniger [2] themselves found similar discrepancies for this parameter regime and attributed them to be due to the relatively large size of ε in the expansion ($\varepsilon \approx 0.1$ for parameters 2.8c). Since the asymptotic expansion is only to first order and ε is not very small, it is obviously expected that fairly substantial errors in the approximation of $X_0(t)$ would appear. However when calculating the turn time, the zeros of an expansion for the speed are found. In an expansion of this kind, zeros of the complete expansion are essentially zeros of each term considered, thus a single term would give a fairly accurate zero of the complete expansion. It is expected therefore that even the single term of this asymptotic series would give a fairly accurate estimate of the turn time.

Assessing the results of this comparison between the numerical solution and the asymptotic expansion gives reasonable confidence in the accuracy of the numerical solution, however the penetration depth is not fully substantiated as of yet.

The next stage of the process is to examine the numerical solution for the time after the stationary point, since here the curve should return to the line $X_0(t) = 0$ however it fails to do so, as illustrated in Figure 2.10f.

§2.12: Examining the solution beyond the stationary point.

Beyond the stationary point there is obviously some need for further investigation. Barring the possibility of rounding errors taking over in the numerical algorithm (since the depth remains bounded and indeed particularly inert), effectively the only possibility available is a problem due to the transformation techniques used. The difficulty is postulated to be due to the misrepresentation of the temperature profiles by the mesh point distribution used. Obviously the region in which this error is most likely to occur is the liquid, since here the transformation equation (2.5c) distributes the mesh points with a large tendency towards the freeze front.

While the solidified layer is growing there is every reason to believe that the transformation equation (2.5c) will provide a sensible distribution of mesh points over the active part of the liquid region. During the growing phase the temperature changes in the liquid are confined to a fairly small spatial region. Once the solidified layer starts to melt however, and X_0 begins to retreat to zero, the effect of the transformation equation (2.5c) will be to concentrate nearly all the mesh points close to the freeze front and this over-concentration become increasingly exaggerated as time increases towards t^* (the time the freeze front returns to its initial position). However, during this time, the volume of liquid affected by the heat transfer region is continuously growing, and consequently, the temperature distribution within the liquid is poorly represented. Therefore during the melting stage it becomes appropriate to simplify the problem and adopt a different technique in the liquid region. To ensure a sensible distribution of mesh points over the liquid region, transformation equation (2.5c) is replaced by a linear transformation:

$$z_3 = 1 + \frac{X - X_0}{10 - X_0} , \qquad \textbf{(2.12a)}$$

which assumes that there is effectively no heat transfer beyond the point X = 10. The value X=10 is suitably large for the cases considered here, but a larger value may be required in other circumstances. The transformed diffusion equation in the liquid region now becomes:

$$\frac{\partial U}{\partial t} = \frac{1}{(10-X_0)^2}\frac{\partial^2 U}{\partial z_3^2} + \frac{20}{(10-X_0)^2}\frac{dX_0}{dt}\frac{\partial U}{\partial z_3} , \quad 1 < z_3 < 2 . \qquad \textbf{(2.12b)}$$

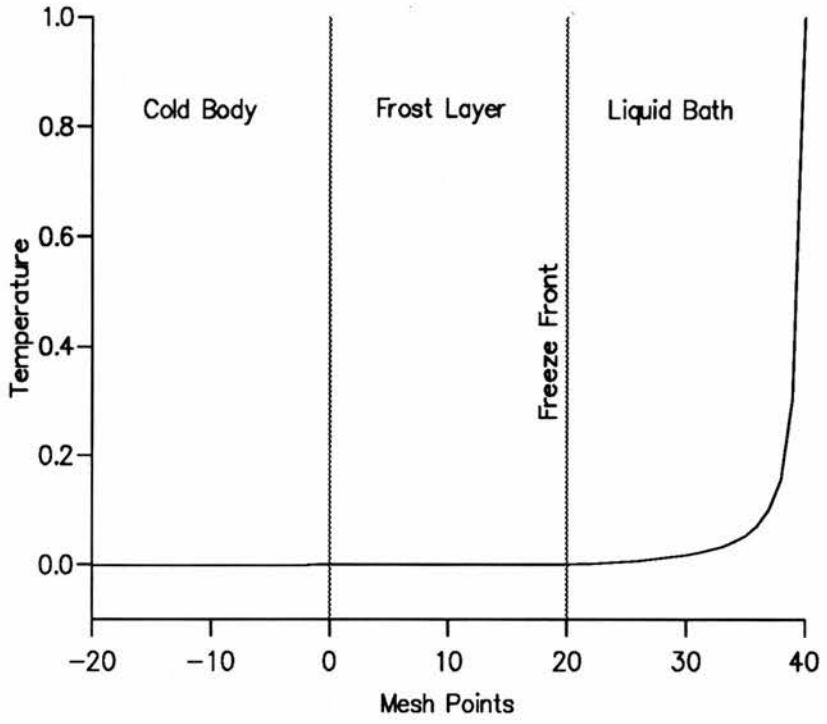

Figure 2.12a: Temperature distribution at a time after the stationary point.

Figure 2.12a shows the original temperature distribution at a time after the stationary point for the test case and it is plain to view the rapidity with which the temperatures in the immersed body and frost layer reach the freezing temperature T_f. The values of the dimensionless temperature U in regions 1 and 2 become very small (with almost linear profiles) shortly after thawing commences. It thus becomes appropriate to consider the temperature profiles in these regions to be identically that of the freezing temperature ($U \equiv 0$), essentially eliminating the need to solve equations in these regions. This assumption leads to a modification of the moving boundary condition, equation (2.5i), since the first term is eliminated and the new transformation, equation (2.12a), yields:

$$\frac{-1}{(10-X_0)}\frac{\partial U}{\partial z_3} = \alpha X_0 \frac{dX_0}{dt} \quad , \qquad \textbf{(2.12c)}$$

thus reducing the heat transfer process to a single phase problem.

Now, all that remains is the decision as to when to change to the new transformation. Ideally, the new transformation, equation (2.12a), should be introduced as soon as the liquid temperature becomes poorly represented by the original set of mesh points and the setting of the temperature profiles in the immersed body and the frost layer to zero should occur when the profiles fall below some tolerance. These two conditions may seem difficult to implement, however with the experience of the transition from the starting solution to the numerical solution in mind, the most obvious solution is to ensure the smooth transition from the original algorithm to the newly adopted one.

The next difficulty lies with the fact that the spatial grid points in the newly transformed liquid region do not coincide with those used during the earlier part of the solution. To overcome this, the temperature values at the new mesh points are generated by fitting a spline approximation across the temperature values for the old mesh points in the liquid region, and effectively reading off the new values.

The technique implemented is that of cubic splines. Given a function f defined over n+1 points $x = 0,..,n$, along with the derivatives at the end points, it is possible to generate a system of cubic splines to find values intermediate to the knots x_i. The solution is given by:

$$S(x) = \sum_{r=-3}^{n-1} a_r \, C(x - r - 2) \quad , \quad 0 \le x \le n \qquad \textbf{(2.12d)}$$

where C is the spline function and is defined as:

$$C(x) = \begin{cases} \frac{2}{3} - \frac{x^2}{2}(2 - |x|) & 0 \le |x| \le 1 \\ \frac{1}{6}(2 - |x|)^3 & 1 \le |x| \le 2 \\ 0 & 2 \le |x| \end{cases}$$

The n+1 coefficients $a_{-2},....,a_{n-2}$ are determined from the value of the function f across the original mesh points and the derivatives of the function at the end points, giving a system of simultaneous equations which can be represented in a matrix form by:

$$\mathbf{M\,a} = \mathbf{b} \qquad \textbf{(2.12e)}$$

where:

$$\mathbf{a}^T = [a_{-2}, a_{-1}, ..., a_{n-2}] \ ,$$

$$\mathbf{b}^T = [6f(0) + 2f'(0), 6f(1), ..., 6f(n-1), 6f(n) - 2f'(n)]$$

and M is the tridiagonal matrix:

$$\mathbf{M} = \begin{bmatrix} 4 & 2 & 0 & 0 & 0 & . & . & . & 0 \\ 1 & 4 & 1 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 4 & 1 & 0 & . & . & . & 0 \\ . & & & . & & & & & . \\ . & & & & . & & & & . \\ . & & & & & . & & & . \\ 0 & . & . & . & 0 & 1 & 4 & 1 & 0 \\ 0 & . & . & . & 0 & 0 & 1 & 4 & 1 \\ 0 & . & . & . & 0 & 0 & 0 & 2 & 4 \end{bmatrix} .$$

The remaining coefficients a_{-3} and a_{n-1} are calculated from:

$$f'(0) = \frac{1}{2}(a_{-1} - a_{-3})$$

$$f'(n) = \frac{1}{2}(a_{n-1} - a_{n-3}).$$

This spline method is thus applied directly to the temperature profiles for the old transformation and the matrix equation (2.12e) is solved using Gaussian elimination†. Thus the new temperature values for the non-coincident points are found directly from equation (2.12d). Since the temperature profile in the liquid region is well behaved, these splines give a very good representation of the dimensionless temperature U for any possible values between knots.

Having formulated the method of changing from one transformation to another, the technique now has to be applied to the temperature profiles at a time after the stationary

† See appendix for program.

point. If the splines technique were to be implemented in the main algorithm after each time step, then the efficiency would be drastically reduced. Thus in order to find an optimum changeover time, some other system must be developed. The simplest way found is to dump the temperature profiles and freeze front information to separate files at regular intervals throughout the lifetime of the frost layer. The splines technique is then implemented to each of these files in turn and the new algorithm is cycled for a few time steps (10 steps seems to work readily) from each set of data. The transition time t_1 is chosen to be that of the file whose data yields the smoothest evolution in the freeze front velocity to the new algorithm.

Of course, this technique of choosing an optimum t_1 may be fully automated to home in to a high degree of accuracy, although for use here, the technique is more than adequate.

Dump time t_1	Speed at time t_1	Speed after 10 cycles	Subsequent life-span of layer	Percentage error from Ref [2]
0.4282	-0.11928321	-0.11924364	0.8151	10.81
0.4521	-0.10800353	-0.10797321	0.8349	8.644
0.4731	-0.09735301	-0.09733194	0.8540	6.554
0.4920	-0.08795862	-0.08794600	0.8699	4.815
0.5089	-0.08010922	-0.08010383	0.8828	3.403
0.5244	*-0.07379403*	*-0.07379459*	*0.8933*	*2.254*
0.5385	-0.06883813	-0.06884345	0.9014	1.368
0.5516	-0.06502620	-0.06503525	0.9085	0.5909
0.5639	-0.06214540	-0.06215734	0.9137	0.02188
0.5753	-0.06000466	-0.06001881	0.9182	0.4705
0.5862	-0.05844156	-0.05845737	0.9221	0.8973

Table 2.12a: The effect of the transition time t_1 on the resultant latter half of the lifetime of the frost layer.

Figures 2.12b and c show the difference in subsequent freeze front motion for various transition times, both for a short time just after each transition and for the subsequent lifetime of the layer. Obviously depending on the transition time taken, the life span of the layer is substantially different thus necessitating a certain degree of selectivity in choosing the optimum transition. Table 2.12a shows transitional information for sample data dumps for the test parameters considered. The first three columns show the time of the dump, the

freeze front position and the freeze front velocity respectively from the original algorithm. The fourth column shows the front velocity shortly after the transition, giving an idea of the smoothness in the evolution from one transformation to the next. The fifth column shows the subsequent life span of the frost layer and this is compared to the value of $t^* = 0.9139$ calculated from the asymptotic expansion technique used by Tadjbakhsh and Linger [2]. From the table, the smoothest transition from one transformation to the next occurs when $t_1 = 0.52437$ (*italic*) and this yields a frost life span of $t^* = 0.8933$ giving a difference from the asymptotic expansion estimate of approximately 2.25%. This result obviously compares favourably and permits confidence in the ability of the immobilising transformation technique to predict with a fair degree of accuracy the subsequent life span of the frost layer (shown in Figure 2.12d).

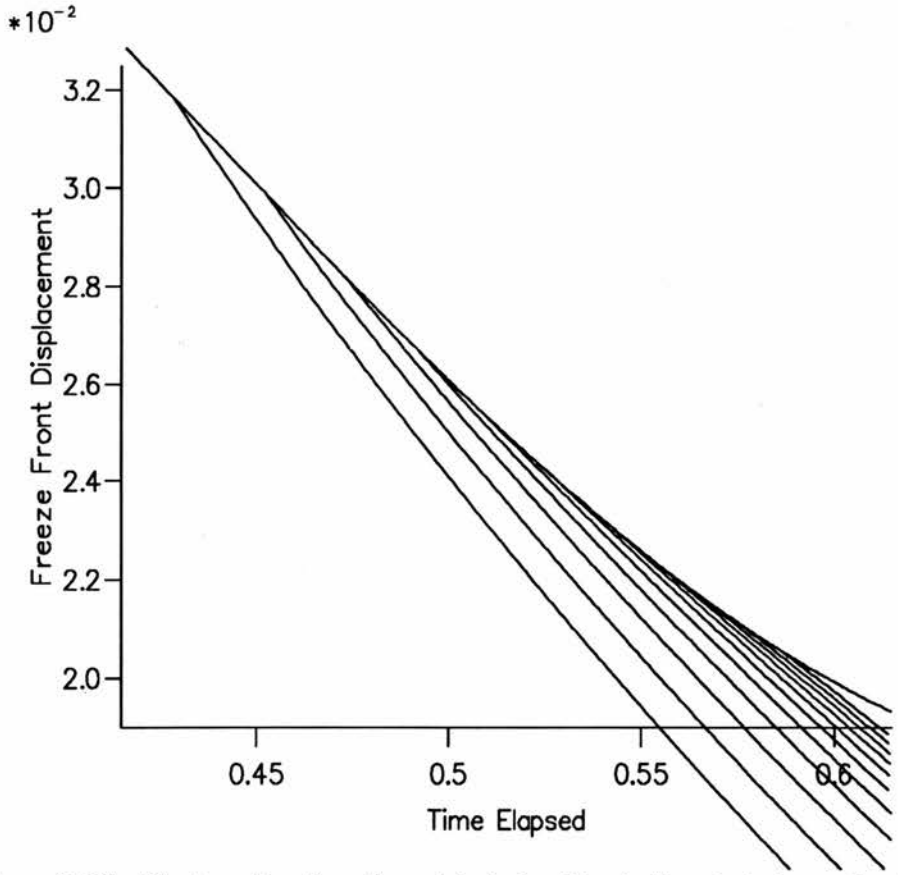

Figure 2.12b: The transition from the original algorithm to the adapted one, showing the ability to predict an optimum changeover time t_1.

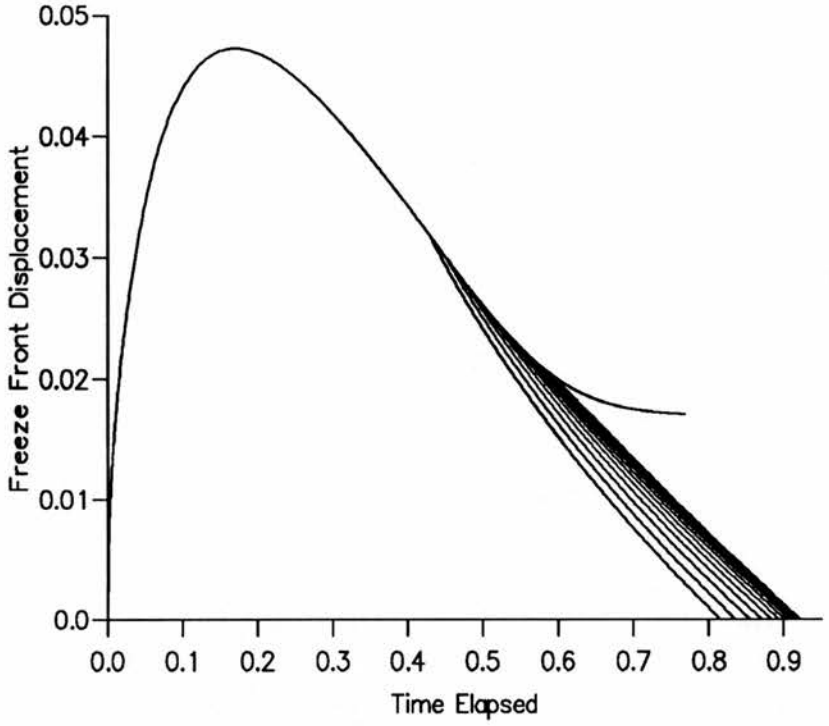

Figure 2.12c: Subsequent lifetime of frost layer in transition to linear transformation.

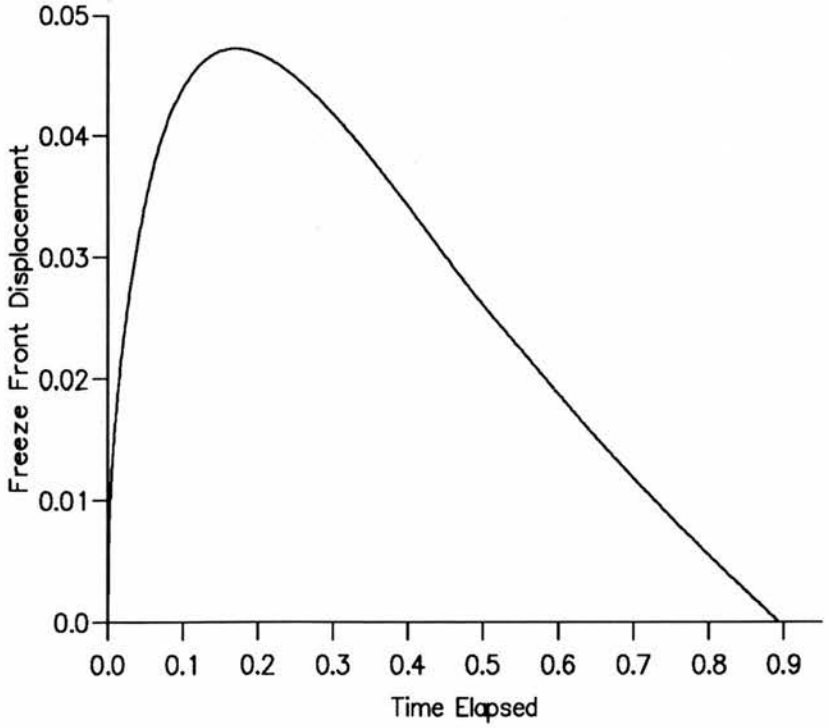

Figure 2.12d: The subsequent full lifetime of the frost layer.

§2.13: The effects of varying the characterising parameters on the solidification process.

Various factors give a reasonable amount of assurance in the overall accuracy of the model:

i) the accuracy in the approximation of the layer lifetime attained from the smoothest transition between transformations.

ii) the ability to predict with precision the time calculated to reach the stationary point.

iii) the accuracy in the maximum depth achieved.

Thus it becomes possible to examine the consequences of altering the dimensionless parameters β, ϕ, η_1, η_2, U_B and α in the relevant equations and formulate a physical appraisal of these effects. In the following analysis, the parameters are varied individually from a basic regime whilst holding the others fixed. The basic regime is:

$$\left.\begin{array}{llll} \alpha = 10.0 & U_B = -0.50 \\ \beta = 2.0 & \phi = 8.0 \\ \eta_1 = 8.0 & \eta_2 = 2.0 \end{array}\right\} \qquad \textbf{(2.13a)}$$

i) The effect of altering β:

β represents the ratio of the thermal conductivities of the solidified layer and the liquid bath. It appears in the moving boundary equation governing the speed of the freeze front. Figure 2.13a and Table 2.13a show the effects of altering β on the parameter λ (determining the initial position and the velocity of the freeze front at the start time t_0) and the change to the lifetime statistics of the frost layer. From the table, it can be seen that as β increases, the initial speed of the freeze front increases and thus it progresses to a greater depth over a longer time period and the frost layer has a greater overall lifetime.

In physical terms, consider the conductivity in the frost layer to be constant. If β is increased, then the relative conductivity of the liquid bath is lower (the conductivity of the immersed body remaining constant). Thus heat passes more readily to the frost layer from the immersed body than from the liquid bath. This in turn means that the effect of the immersed cold body is more readily transferred to the moving front through the solidified layer. Thus the temperature differential at the front is more likely to speed up its progress.

It should be noted that although negative solutions for λ exist, these are physically inappropriate. In the event of a negative λ, the equations dictating heat flow across both the interfaces between the three regions, along with the diffusion equation in the solidified layer, become invalid.

β	U_0	λ	$X_{0,max}$	$t(X_{0,max})$	t^*
1.0	-0.02702	0.025316	0.01912	0.1250	0.4693
2.0	-0.09063	0.098182	0.04686	0.1617	0.8870
3.0	-0.13122	0.158005	0.07765	0.2374	1.692
4.0	-0.16024	0.209746	0.1117	0.3189	2.974
5.0	-0.18238	0.255825	0.1524	0.4228	4.988
6.0	-0.20002	0.297460	0.1874	0.5417	8.475

Table 2.13a: The effect of altering the parameter β on the stationary point and subsequent life span of the layer.

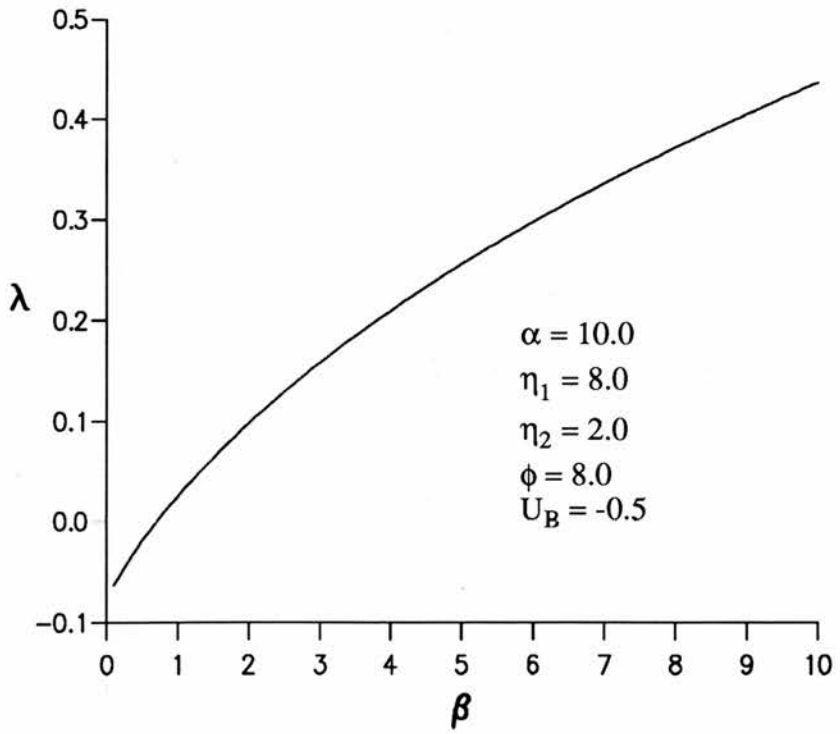

Figure 2.13a: Graphical representation of the effect of altering the parameter β on the starting solution parameter λ.

ii) The effect of altering ϕ:

ϕ represents the relative conductivity of the immersed cold body with respect to the frost layer. Thus if ϕ is increased, this may be interpreted as an increase in the conductivity of the immersed body, whilst holding the conductivity of the solidified layer constant. The conductivity of the liquid also remains constant. Thus, the effects of the immersed body are conducted more quickly to the frost layer than the effects of the liquid bath. This in turn makes the frost layer colder and thus is more liable to grow. The penetration depth will increase and the time scales involved will be larger, which is upheld by Figure 2.13b and Table 2.13b.

ϕ	U_0	λ	$X_{0,max}$	$t(X_{0,max})$	t^*
6.0	-0.05130	0.067577	0.02884	0.1187	0.6117
8.0	-0.09063	0.098182	0.04686	0.1617	0.8870
10.0	-0.12761	0.121631	0.05701	0.2048	1.157
12.0	-0.16047	0.139849	0.06924	0.2582	1.556

Table 2.13b: The effect of altering the parameter ϕ on the stationary point and subsequent life span of the layer.

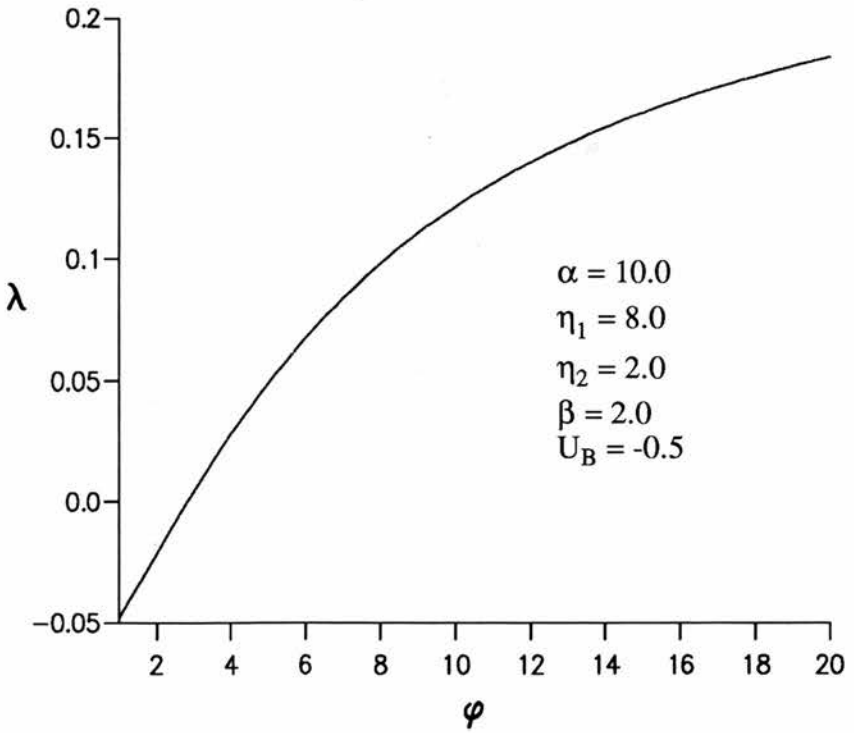

Figure 2.13b: Graphical representation of the effect of altering the parameter ϕ on the starting solution parameter λ.

Mathematically, ϕ is used in the calculation of the updated temperature at the interface between the immersed cold body and the frost layer (equation 2.5h). If ϕ is increased then the value U_0^{p+1} decreases, thus giving the appropriate colder frost layer and so increasing the maximum frost depth, and the value of λ increases so creating a larger frost depth and freeze front velocity at the start time.

ii) The effect of altering η_1:

η_1 represents the relative size of the value of thermal diffusivity in the immersed cold body with respect to the liquid region. If this value is large, then heat is more liable to diffuse quickly through the immersed cold body hence warming it more rapidly. This in turn causes the developing freeze layer to penetrate to a lesser depth and thus gives a shorter layer lifetime. Figure 2.13c and Table 2.13c uphold this pattern since as η_1 increases, the initial speed (given by λ) decreases, the maximum depth achieved decreases, as too does the layer lifetime.

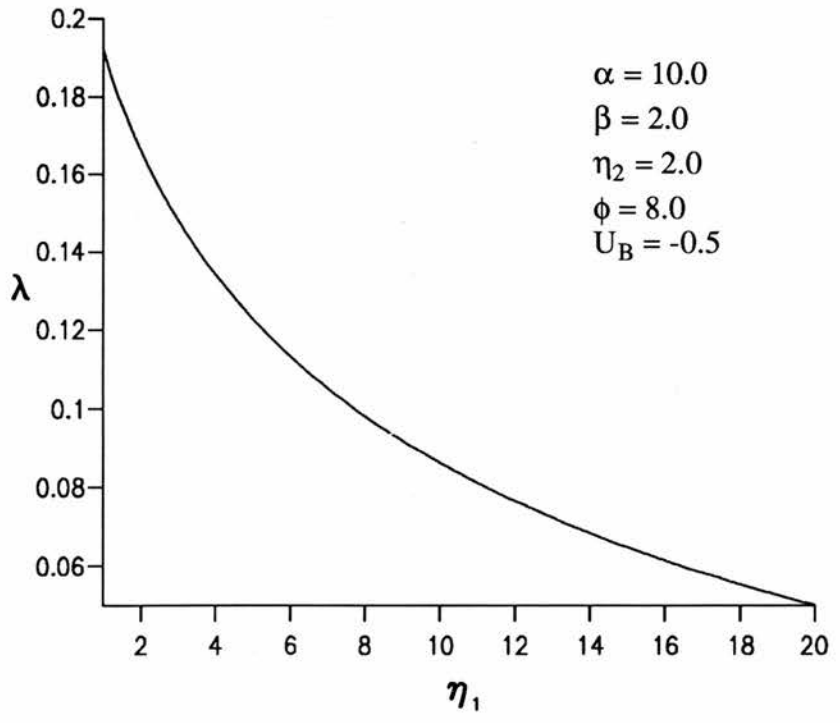

Figure 2.13c: Graphical representation of the effect of altering the parameter η_1 on the starting solution parameter λ.

η_1	U_o	λ	$X_{o,max}$	$t(X_{o,max})$	t^*
2.0	-0.21382	0.165919	0.2035	1.737	12.54
4.0	-0.14969	0.134087	0.09889	0.5035	3.183
6.0	-0.11396	0.113404	0.05895	0.2430	1.174
8.0	-0.09063	0.098182	0.04686	0.1617	0.8870
10.0	-0.074105	0.086255	0.03263	0.1084	0.5447
12.0	-0.06177	0.076538	0.03002	0.08689	0.4351

Table 2.13c: The effect of altering the parameter η_1 on the stationary point and subsequent life span of the layer.

ii) The effect of altering η_2:

η_2 represents the relative size of the value of the thermal diffusivity in the frost layer with respect to the liquid region. This value should play no role in the overall penetration depths and times, since the tendency of heat to diffuse from the immersed cold body or liquid is unaltered by this parameter. This can be readily seen in Table 2.13d.

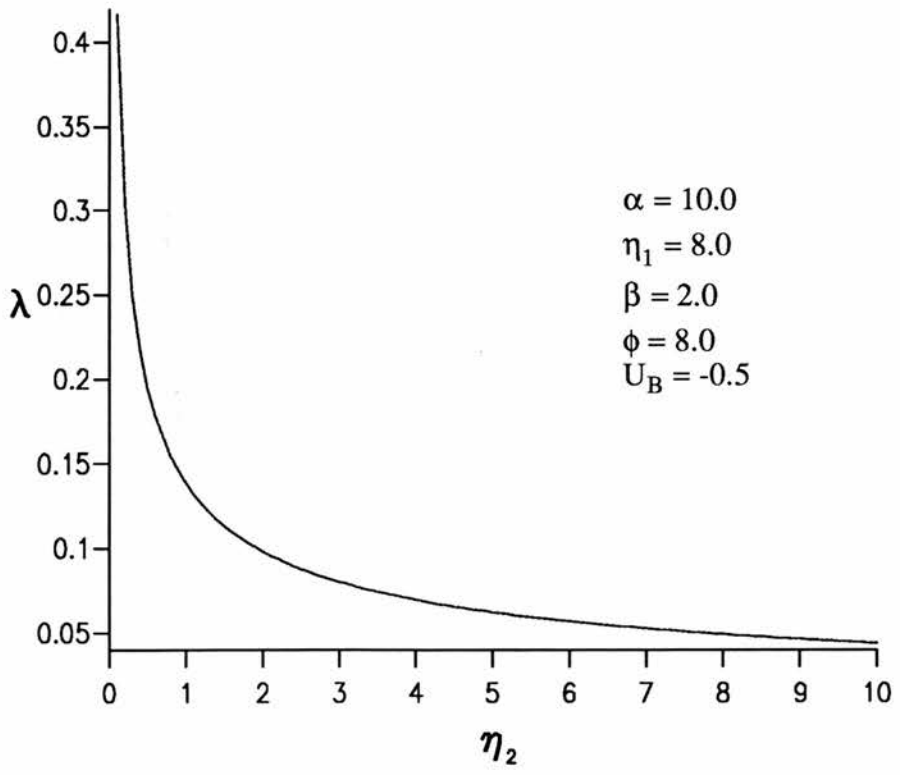

Figure 2.13d: Graphical representation of the effect of altering the parameter η_2 on the starting solution parameter λ.

However, η_2 plays a major part in the initial behaviour of the freeze front, as can be seen from the change in λ with respect to η_2 from Figure 2.13d. As η_2 is increased, the initial speed does not alter too drastically. The relationship between η_2 and λ in the equation $X_0 = \lambda \sqrt{\eta_2 t}$ is such that as η_2 gets larger, λ must become significantly smaller.

η_2	U_0	λ	$X_{0,max}$	$t(X_{0,max})$	t^*
1.0	-0.09035	0.138436	0.04681	0.1632	0.8871
2.0	-0.09063	0.098182	0.04686	0.1617	0.8870
3.0	-0.09073	0.080246	0.04688	0.1619	0.8855
4.0	-0.09077	0.069530	0.04689	0.1609	0.8860
6.0	-0.09082	0.056799	0.04690	0.1611	0.8857
8.0	-0.09084	0.049202	0.04690	0.1616	0.8861

Table 2.13d: The effect of altering the parameter η_2 on the stationary point and subsequent life span of the layer.

ii) The effect of altering U_B:

U_B is probably the parameter whose behaviour is most easy to predict. U_B gives a measure of the temperature of the immersed cold body relative to the liquid region and is always negative. If this is reduced, then the penetration depth and frost lifetime are increased. This again is easily recognisable from Table 2.13e. Also, since the initial difference in temperatures of the immersed body and the liquid region increases as U_B decreases, then so too must the initial freeze front depth and velocity, upheld by the larger size of the parameter λ in Figure 2.13e.

U_B	U_0	λ	$X_{0,max}$	$t(X_{0,max})$	t^*
-0.25	-0.01351	0.025316	0.01888	0.1215	0.4929
-0.5	-0.09063	0.098182	0.04686	0.1617	0.8870
-0.75	-0.19684	0.158005	0.07765	0.2374	1.692
-1.0	-0.32048	0.209746	0.1218	0.3394	2.725
-1.25	-0.45595	0.255825	0.1524	0.4228	5.479
-1.5	-0.60006	0.297640	0.1996	0.5487	7.864

Table 2.13e: The effect of altering the parameter U_B on the stationary point and subsequent life span of the layer.

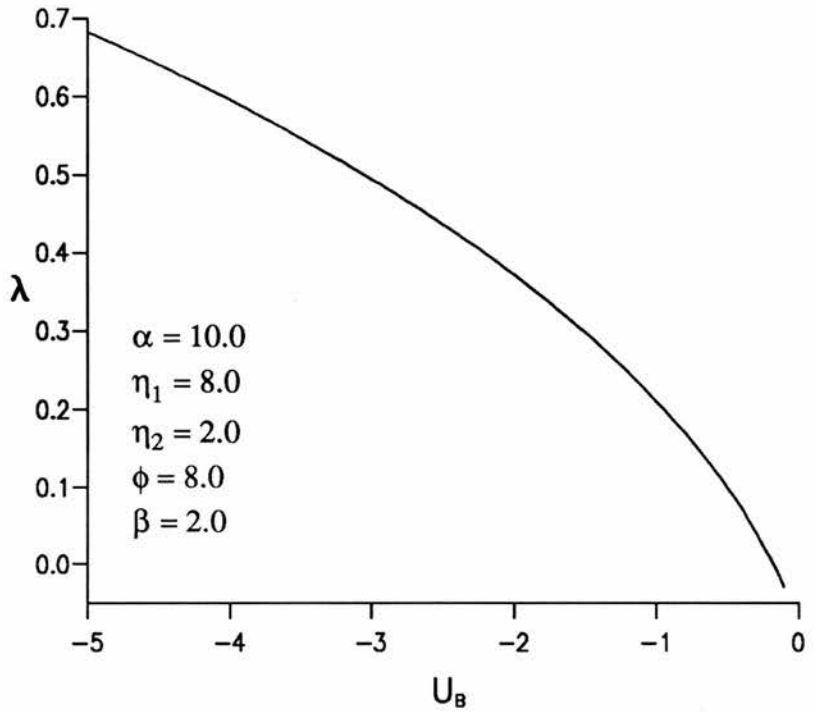

Figure 2.13e: Graphical representation of the effect of altering the parameter U_B on the starting solution parameter λ.

ii) The effect of altering α:

The behaviour of the parameter α is slightly more tricky to assess. α is the reciprocal of the Stefan number for the process. It gives a representation of the amount of heat absorbed or emitted for a change of state of the liquid bath. If α is increased, then a larger amount of latent heat is released on solidification of the liquid, which in turn will go to heat the immersed cold body. Thus the penetration depth will be reduced, the initial speed will be reduced (characterised by the size of the parameter λ in Figure 2.13f), as too will the time to maximum penetration. However, at a time after the stationary point of the boundary, heat re diffusing back in from the liquid goes to changing the phase of the solidified layer. Since the latent heat of fusion is larger, more heat is required and thus the return speed is reduced. This increases the overall life span of the layer as can be seen from the last column in the Table 2.13f.

α	U_0	λ	$X_{0,max}$	$t(X_{0,max})$	t^*
2.5	-0.16534	0.219798	0.1102	0.1993	0.8145
5.0	-0.12857	0.153677	0.07516	0.1809	0.8532
7.5	-0.10609	0.119482	0.05750	0.1703	0.8658
10.0	-0.09063	0.098182	0.04686	0.1617	0.8870
12.5	-0.07925	0.083515	0.03988	0.1570	0.9564
15.0	-0.07049	0.072752	0.03499	0.1563	1.044

Table 2.13f: The effect of altering the parameter α on the stationary point and subsequent life span of the layer.

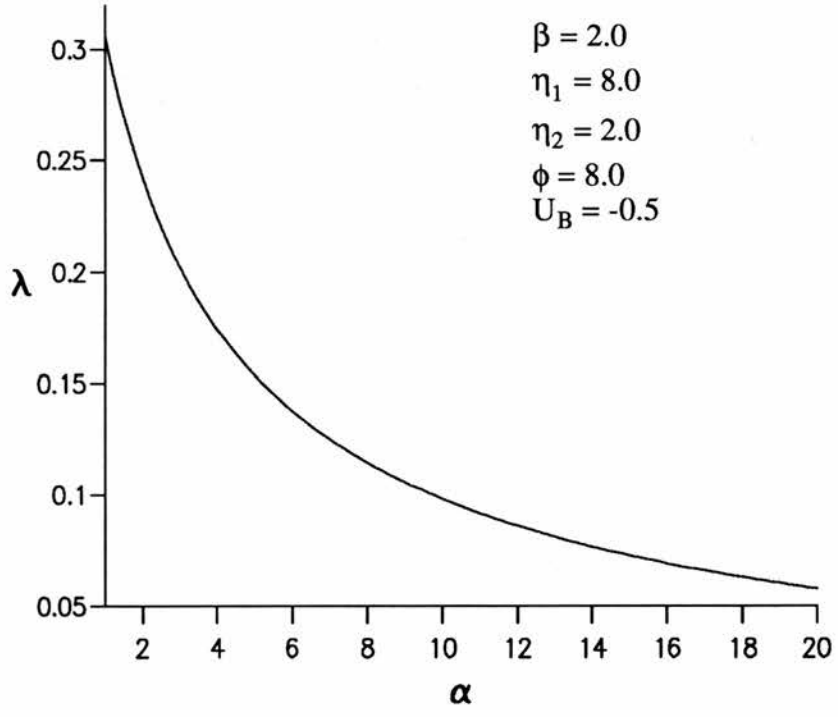

Figure 2.13f: Graphical representation of the effect of altering the parameter α on the starting solution parameter λ.

§2.14: Summary of the immobilising transformation technique.

The aim of this chapter has been to solve the full three region problem generated by immersing a cold body in a warm liquid thus generating a transient frost layer. This is achieved by use of immobilising transformations in the frost layer and the liquid region so as to fix the freeze front boundary to enable its position to be calculated for all time. A starting solution consisting of error function profiles is used to seed a numerical algorithm

of explicit finite difference approximations to the partial differential equations generated. At a time after the subsequent stationary point of the freeze front, the transformation technique becomes unsuitable due to the rapidity in the rise in temperature of the immersed cold body and frost layer to the freezing point of the phase change medium and also the misrepresentation of the relevant temperature profiles due to a maldistribution of spatial mesh points in the liquid region. Thus a different linear transformation is used in the liquid region to map the temperature profile more accurately, along with the temperature profiles in the immersed body and frost layer being considered to be uniform at the freezing temperature. This change in transformation requires a spline technique to produce the new array of temperature values at the newly formed mesh points which are not coincident with the old transformations mesh points.

The use of these processes are validated by consideration of two special cases which essentially reduce the problem to a two region configuration. The case of continuous freezing is such that the temperature of the immersed body remains constant, so inducing perpetual freezing of the liquid region. The accuracy attained gives great confidence in the techniques ability to predict the freeze front position and corresponding temperature profiles. However, the second case involving a limiting front depth shows an alarming consequence of altering the size of the start time and the corresponding starting temperature profiles. This questioning of the size of the start time is carried over into the full three region problem and the answer is deemed to lie in the smoothness of transition from the starting solution to the numerical solution. The starting solution dictates that the temperature value at the interface between the immersed body and the frost layer is constant at U_0 for all possible start times. This obviously must be incorrect, since as soon as the numerical algorithm is cycled beyond the starting solution, the temperature value at this boundary must rise. Thus it would seem logical to assume that a better starting solution would allow U_0 to vary. This is impossible for the method considered here and so the only way around this problem is to take an early enough start time to allow the smoothest possible transition to the numerical solution.

The results given by this numerical algorithm are compared with those generated by an asymptotic solution developed by Tadjbakhsh and Liniger [2]. The results for the time to

stationary point and the frost layer life span are very encouraging, however the results for the maximum depth are inconsistent, which may have been anticipated from the possible accuracy of the asymptotic expansion for the depth. The value generated by the numerical algorithm here appears to be correct, however any justification of this still has to be achieved.

§3. Solution of the one-dimensional Stefan problem using a fixed grid technique.

§3.1: Introduction.

Probably the best means of validating the results gained by the immobilising transformation technique in §2 is to attempt a solution of the same problem by an intrinsically differing method. The method adopted is one described by Crank [13] and involves solution by a fixed grid technique. This method is used by other authors [2,3] for the same purpose and hence the reason it is chosen here. The computational process is similar to the immobilising transformation technique, but instead of immobilising the freeze front so that a fixed number of spatial mesh points appear in each region at all times, the freeze front is allowed to track across a spatial grid and its position is calculated at each step. This method is far less efficient than the immobilisation technique due to the inability to increase the size of the time increment available as the freeze front progresses.

The aim of this chapter initially is to solve the same classical Stefan problem as in §2 in order to draw a comparison between the two methods. Ideally the results generated by this method will substantiate those generated in the previous chapter. However, the limitations of the method, as demonstrated here, actually show the computational complexity of the problem under investigation and the necessity for the development of a new technique, such as the immobilising transformation technique.

The problem tackled is identical to that in §2 and thus all the physical assumptions §2.1(i-vii) concerning uniformity, symmetry etc. are again implemented and hence the parameters occurring in the following theory are the same as those given in §2.1.

§3.2: Interpolation around the freeze front.

The physical system is identical to that shown in Figure 2.2a and indeed the mathematics of the system is identical as far as the diffusion and boundary conditions are concerned; so the non-dimensionalised equations 2.4(c-n) hold as before. The fundamental difference with this method, as opposed to the immobilising transformation technique, is that the freeze front is allowed to move across a fixed grid. A typical freeze front position is shown in Figure 3.2a.

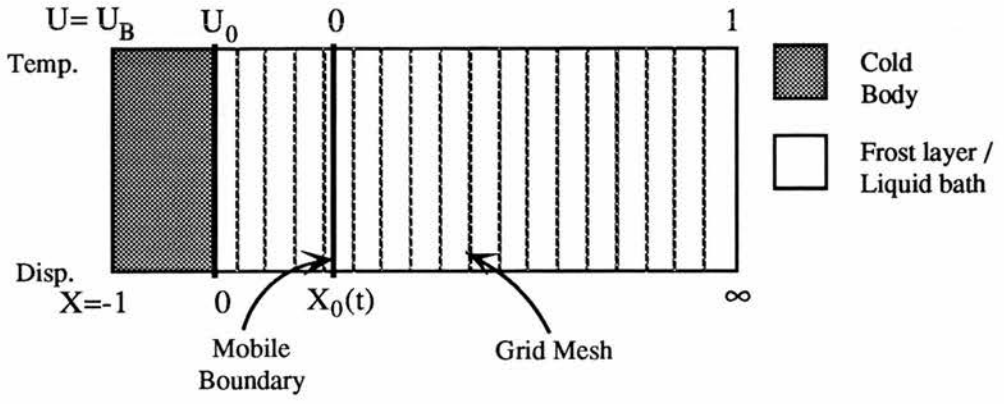

Figure 3.2a: A typical system configuration.

With a fixed grid and boundary condition (2.4h) to be satisfied at the freeze front $X_0(t)$, it is necessary to track the location of the front in part of the calculation. Also, an approximation of the heat equations at the points adjacent to the front has to be considered. The usual explicit finite difference approximations used in the previous chapter rely on equal grid spacings and hence some system must be developed to overcome this.

Consider the standard Lagrangian interpolation formula on (n+1) distinct points a_j

$$f(x) = \sum_{j=0}^{n} l_j(x)\, f(a_j) \quad , \qquad \textbf{(3.2a)}$$

where $$l_j(x) = \frac{p_n(x)}{(x-a_j)\, p'_n(a_j)}$$

and $$p_n(x) = (x-a_0)(x-a_1)\ldots(x-a_{n-1})(x-a_n).$$

Restricting to a three point formula, i.e. n=2, then

$$\frac{1}{2}\frac{d^2f(x)}{dx^2} = \frac{f(a_0)}{(a_0-a_1)(a_0-a_2)} + \frac{f(a_1)}{(a_1-a_0)(a_1-a_2)} + \frac{f(a_2)}{(a_2-a_0)(a_2-a_1)} \qquad \textbf{(3.2b)}$$

and $$\frac{df(x)}{dx} = l'_0(x)f(a_0) + l'_1(x)f(a_1) + l'_2(x)f(a_2) \quad , \qquad \textbf{(3.2c)}$$

where $$l'_0(x) = \frac{(x-a_1) + (x-a_2)}{(a_0-a_1)(a_0-a_2)} \, ,$$

$$l'_1(x) = \frac{(x-a_2) + (x-a_0)}{(a_1-a_2)(a_1-a_0)},$$

and $$l'_2(x) = \frac{(x-a_0) + (x-a_1)}{(a_2-a_0)(a_2-a_1)}.$$

The numerical procedure is as follows. The cold body, region 1 is divided into 20 subintervals of length $h_1 = \frac{1}{20}$ with mesh points numbered q=-20 to 0. For regions 2 and 3, the frost layer and liquid, equal intervals of length h are selected with grid points numbering q=1,2,... . Suppose that the value of $X_0(t)$ lies between the r^{th} and $(r+1)^{th}$ grid point. For points away from the freeze front, the diffusion equations are approximated using the same explicit finite difference equations as in §2, giving:

$$U_q^{p+1} = U_q^p + k\kappa_i \left\{ \frac{U_{q+1}^p - 2U_q^p + U_{q-1}^p}{h_i^2} \right\}, \qquad \eta_i = \frac{\kappa_i}{\kappa_3}, \qquad \textbf{(3.2d)}$$

where $-20 < q < 0$ for i=1, $0 < q < r-1$ for i=2 and $r+1 < q$ for i=3. The points -20 and 0 are the locations of the centre of the cold body and the cold body surface, respectively and the points (r-1, r, r+1) are grid points adjacent to the front location.

The boundary condition (2.4f) at X=-1 becomes:

$$U_{-20}^{p+1} = \frac{4U_{-19}^p - U_{-18}^p}{3}, \qquad \textbf{(3.2e)}$$

where U_{-20}^{p+1} is an estimate of the temperature at x=-a. For the purposes of computation, assume that X = hn is sufficiently far away from the frost layer as to be effectively undisturbed by the heat transfer process. The size of h will be assigned later. At the surface of the cold body, X = 0 (the grid point q=0), the boundary condition (2.4g) is approximated as:

$$3\left\{ \phi + \frac{h_1}{h} \right\} U_0^p = \phi \left\{ 4U_{-1}^p - U_{-2}^p \right\} + \frac{h_1}{h} \left\{ 4U_1^p - U_2^p \right\}. \qquad \textbf{(3.2f)}$$

At the extremity of the liquid, assumed to be hn ($>>X_0$), the liquid temperature is assumed to remain at the initial ambient temperature T_0 so that:

$$U_n^{p+1} = 1$$

for all p (all time).

All that remains is to consider how to advance the solution around the freeze front. Consider a typical freeze front position as shown in Figure 3.2b. The freeze front has progressed to be in the interval between the r^{th} and $r+1^{th}$ grid points. The exact position of the front is given relative to the $r-1^{th}$ grid line as wh, with $1 \leq w \leq 2$.

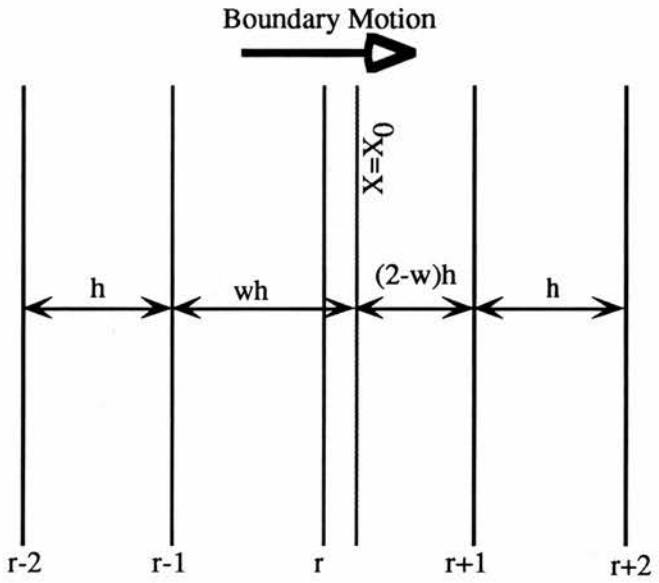

Figure 3.2b: A typical position of the freeze front relative to the spatial mesh.

When updating the solution, the immersed body temperature profiles are calculated from the approximated diffusion equation (3.2d, i=1), along with the frost layer temperatures for the mesh points up to the $r-2^{th}$ grid point from equation (3.2d, i=2). To progress the solution at the points r-1 and r+1 (the r^{th} point is omitted for the time being) a Lagrangian formulated diffusion equation is used. By substituting equation (3.2b) into the non-dimensionalised diffusion equation (2.4c), and effectively interpolating over the old temperatures at the points r-2, r-1 and X_o (ignoring the r^{th} point), an appropriate approximation is obtained:

$$U_{r-1}^{p+1} = U_{r-1}^{p} + \frac{2\,k\,\kappa_2}{h^2}\left\{ \frac{U_{r-2}^{p}}{w^p+1} - \frac{U_{r-1}^{p}}{w^p} \right\} \qquad \textbf{(3.2g)}$$

Similarly, the new temperature value of the r+1th mesh point is found by interpolating over the old temperature values at the points X_0, r+1, r+2 giving:

$$U_{r+1}^{p+1} = U_{r+1}^{p} + \frac{2\,k}{h^2}\left\{ \frac{U_{r+2}^{p}}{3 - w^p} - \frac{U_{r+1}^{p}}{2 - w^p} \right\}. \qquad \textbf{(3.2h)}$$

The remaining liquid temperatures are calculated using the approximated diffusion equation (3.2d, i=3) from the r+2th mesh point upwards.

In order to track the freeze front it is still necessary to ascertain its position in terms of w^{p+1}. This is achieved by substituting the Lagrangian interpolation approximation, equation (3.2c), into the moving boundary equation (2.4h), giving:

$$\frac{\beta}{h_2}\left\{ \frac{w^p U_{r-2}^{p}}{w^p+1} - \frac{(w^p+1)U_{r-1}^{p}}{w^p} \right\} - \frac{1}{h_2}\left\{ \frac{(w^p-2)U_{r+2}^{p}}{3-w^p} + \frac{(w^p-3)U_{r+1}^{p}}{w^p-2} \right\} =$$

$$= \alpha\frac{dX_0}{dt} = \frac{\alpha\,h}{k}\left\{ w^{p+1} - w^p \right\}, \qquad \textbf{(3.2i)}$$

which is then used to calculate the updated value w^{p+1}.

The temperature value at the rth mesh point is omitted from consideration. However when the freeze front passes from one grid interval to an adjacent one, that is when $w^{p+1} < 1$ or $w^{p+1} > 2$, then it is necessary to re-establish its value before continuing. For the case when the frost depth is increasing, this is achieved by Lagrangian interpolation over the points r-2, r-1 and X_0. The interpolation is taken over these three points since the rth grid line lies within the region ([r-1]h,X_0) and all these points lie within the solidified layer, hence:

$$U_r^{p+1} = \frac{(1-w^p)\,U_{r-2}^{p+1}}{1+w^p} + \frac{2\,(w^p-1)\,U_{r-1}^{p+1}}{w^p}. \qquad \textbf{(3.2j)}$$

The theory covered by Crank [13] only covers the case when the solidified layer is growing, but for the system here, the solution must also be developed when thawing occurs. When w^{p+1} becomes less than unity, the temperature value U_r is found by interpolating over the points X_0, r+1 and r+2, since the r^{th} grid line lies within the region $(X_0,[r+1]h)$ and thus all these points lie within the liquid region:

$$U_r^{p+1} = \frac{2\,(2-w^p)\,U_{r+1}^{p+1}}{3-w^p} - \frac{(2-w^p)\,U_{r+2}^{p+1}}{4-w^p} . \qquad \textbf{(3.2k)}$$

§3.3: Stability.

The approximations used are again explicit by nature and so it is necessary to find a suitably small time increment k so that the system of equations remains stable. This is found, as before, by assuring that no cancellation of terms occurs in the approximated diffusion equations. For the above system, k must be bounded above by the least of the following, being the stability bounds for the immersed cold body, frost layer and liquid bath respectively:

$$k \leq \frac{h_1^2}{2\,\eta_1} , \qquad \textbf{(3.3a)}$$

$$k \leq \frac{h^2}{2\,\eta_2} , \qquad \textbf{(3.3b)}$$

$$k \leq \frac{h^2}{2} . \qquad \textbf{(3.3c)}$$

These bounds are calculated for the parameter regime (2.8c). Here the mesh size in the liquid bath is restricted to be very fine compared to that in the immersed body (as will become apparent) and thus the second bound, equation (3.3b), is the appropriate one.

§3.4: A slight modification of the Crank fixed grid technique.

The procedure described above is essentially that of Crank [13]. The system is configured as in Figure 3.4a, with 20 mesh points in the immersed cold body and 2000 available for the frost layer and liquid bath, these representing the spatial region (X=0,10).

Note that the value X=10 is an estimated distance whereby the temperature of the liquid beyond that point is essentially the initial liquid temperature T_0. The spatial grid size in the frost layer and liquid bath is h = 0.005.

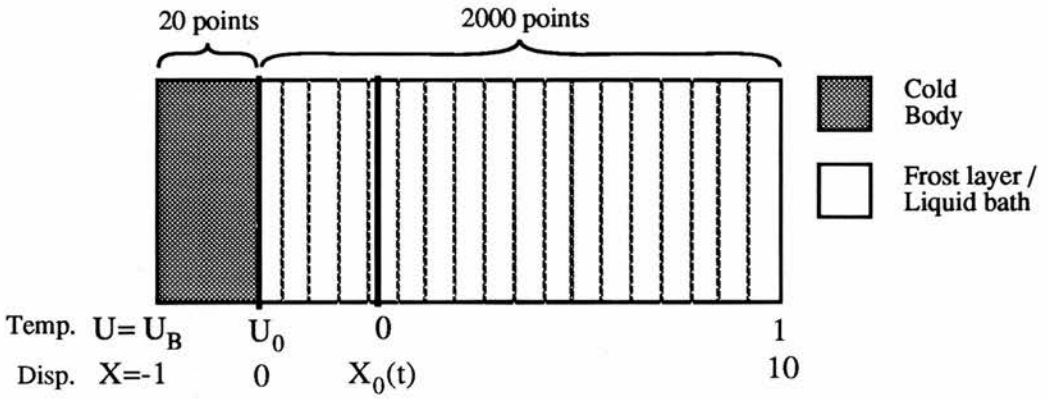

Figure 3.4a: Configuration of the system using 20 mesh points in the immersed body and 2000 points available for the frost layer and liquid bath.

Since the problem solved here is identical to that solved in §2, the starting solution used is the same, giving a profile for the temperatures in the three regions, an initial boundary position X_0 and its velocity at a computational start time t_0.

The first real difficulty arises upon consideration of a possible starting time for the numerical algorithm. In order to progress the system from the starting solution, the moving boundary equation (3.2i) requires temperature values at the r-2^{th}, r-1^{th}, r+1^{th} and r+2^{th} mesh points. This is illustrated in Figure 3.4b. Consequently to enable availability

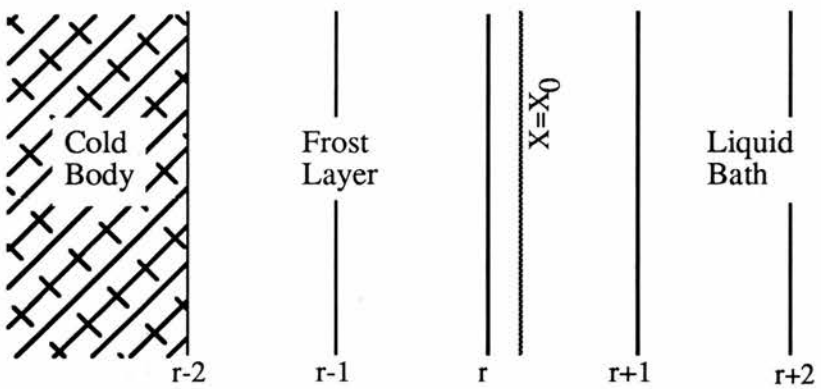

Figure 3.4b: Algorithm minimum starting condition.

of an r-2^{th} grid point, the freeze front has to have moved sufficiently far from the surface of the cold body. As a very minimum the freeze front must in fact be at least as far out as to

lie between the third and fourth mesh points, so that X_0 must be strictly greater than 2h (=0.01) to start the algorithm. For the parameters (2.8c) used here, this gives a bound for t_0 as:

$$t_0 = \frac{X_0^2}{\eta_2 \lambda^2} > \frac{10^{-4}}{\eta_2 \lambda^2} \approx 0.01. \qquad \textbf{(3.4a)}$$

The bound (3.4a) is the minimum possible start time for the numerical algorithm to function. However, it is desirable to choose a start time t_0 such that enough mesh points initially lie within the frost layer in order to represent the temperature profile across it more accurately. Too late a start time is obviously inappropriate, so here the start time is taken to be $t_0 = 0.02$, being well beyond the minimum (3.4a), allowing at least 4 mesh points to be initially present in the frost layer. It should be noted that this start time is far from optimum with respect to the smoothness of the transition from the starting solution to the numerical algorithm. On further investigation of this transition, it is clear that a start time of the order $t_0 < 10^{-3}$ would be more appropriate if at all possible. This is effectively unavoidable due to the strict bound placed on the size of the start time by equation (3.4a) and the necessity to accurately represent the temperature profile in the frost layer.

Since this method involves non-standard difference approximations, the stability bound generated, equation (3.3b) is checked for the parameters (2.8c) before the full algorithm is tried. For the parameters used the initial theoretical stability bound is thus:

$$k < 6.14 \times 10^{-6}. \qquad \textbf{(3.4b)}$$

The algorithm is run for a few steps from the start time t_0 for differing time increments k, and the resultant stability is shown in Figure 3.4c. The estimate for the bound on k is obviously a good one, since beyond that the graph shows unstable peaks and troughs. Therefore, with k constrained by (3.4b) the algorithm may be advanced with a reasonable amount of confidence.

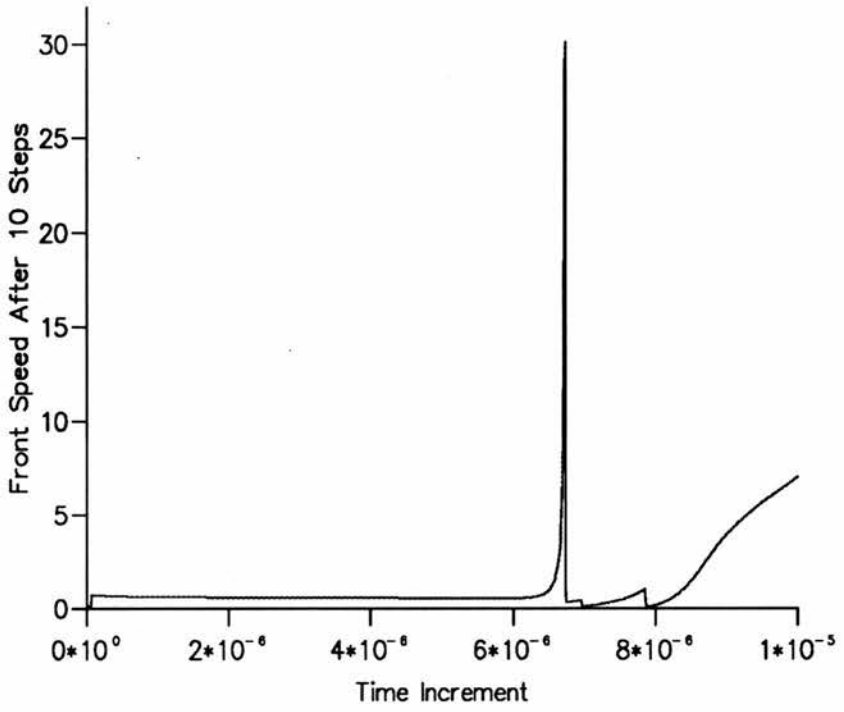

Figure 3.4c: Graph showing stability for differing time increments k.

The algorithm is hence cycled for a reasonable time span and the results examined. Figure 3.4d shows a graph of the progress of the front speed throughout the lifetime of the layer. This value should exhibit a smooth curve since the freeze front should not show any spurious acceleration, however the figure shows a completely different story.

Peaks occur for the full lifetime of the frost layer along the graph. On closer examination they occur when the front passes through a mesh point which happens when the value w_P reaches 2 during solidification. This is a result of the factor $(2-w_P)^{-1}$ in equation (3.2i) which becomes very large for w_P close to 2. Obviously when this method was first formulated in 1956, only larger space and time increments were available due to the limited processing power and so w_P never approached 2 to any accuracy. However, on the scales of spatial accuracy to which the calculations are performed here, a division by zero is effectively taking place.

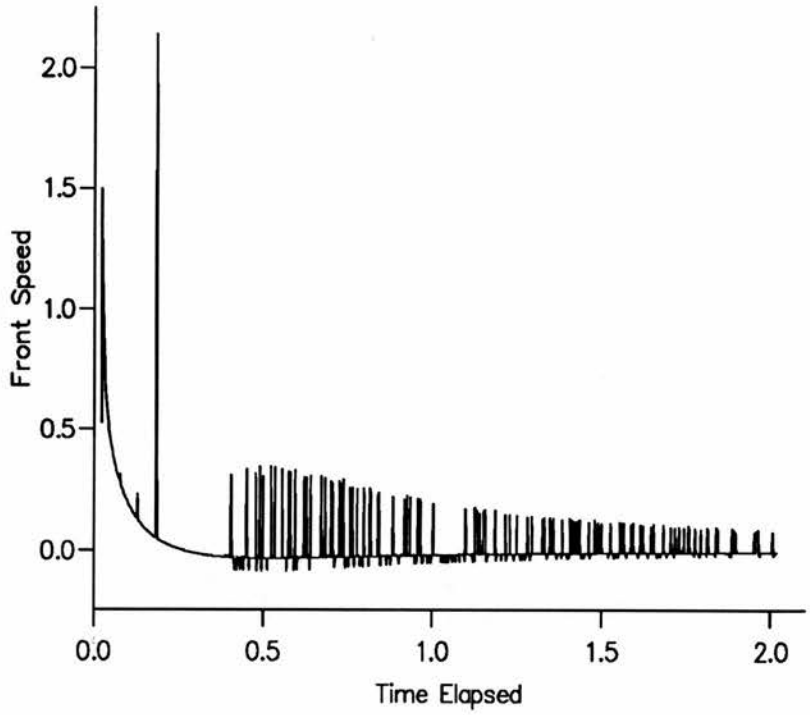

Figure 3.4d: Graph showing the freeze front speed for the frost layer lifetime.

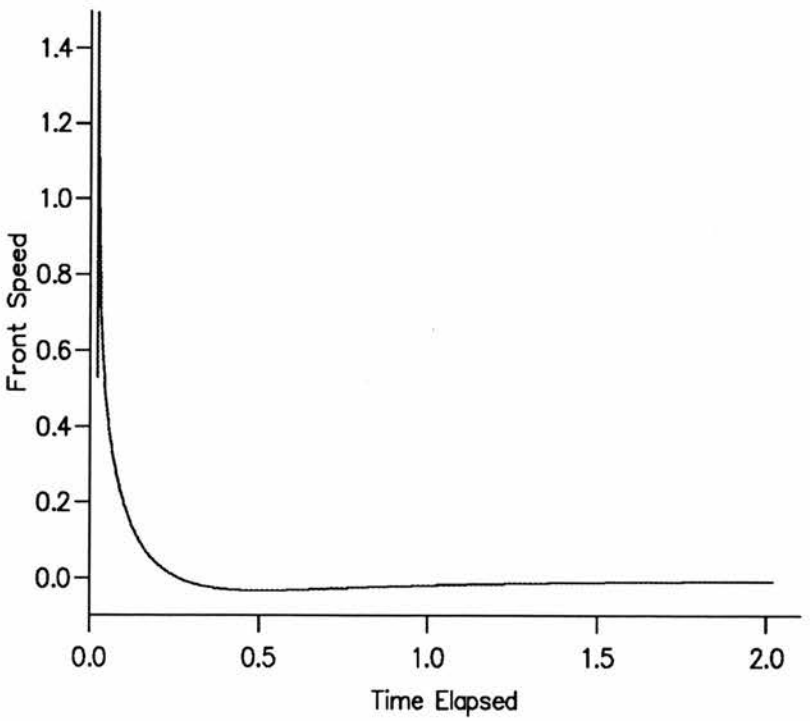

Figure 3.4e: Graph of the freeze front speed for the frost lifetime using modified changeover grid positions.

To overcome this, the algorithm is rerun using the same mesh spacings, but instead of changing the value of r when w_P approaches 2 or 1, the change occurs when w_P approaches 1.5 or 0.5. This completely avoids the problem of division by zero which occurs if $w_P=2$ (or indeed $w_P=0$ see equation (3.2i)). The graph in Figure 3.4e shows a smooth curve for the front speed over the lifetime of the frost, thus eradicating the instability peaks intrinsic in Crank's original formulation.

The results for the location of the stationary point of the moving boundary are shown in Table 3.4a. Also shown is the similar results for the immobilising transformation technique with start time $t_0 = 0.02$ and those found from the perturbation technique developed by Tadjbakhsh and Liniger [2]. As mentioned previously, the optimum start time for the immobilising transformation technique is $t_0 = 0.0001$ and so this comparison is merely an attempt to compare like with like.

	X_{max}	Turn time
transform t_0=0.001	0.04680	0.1720
Tadjbakhsh et al	0.06248	0.1685
transform t_0=0.02	0.0702	0.235
fixed grid t_0=0.02	0.0701	0.259

Table 3.4a: Estimated location of stationary point using the three methods available.

The results compare very well; the difference in $X_{0,max}$ is less than 1% and the difference in the turn time is about 9%. Superficially, the results of the fixed grid approach with the very large number of mesh points, 2020 in all, might seem more reliable. However, by examining X_0 and r over the frost lifetime it is discovered that X_0 only reaches a maximum value of r=14 out of 2000 available points, giving a loss in accuracy due to the relatively small number of mesh points in the frost layer. This comparison between the size of n and the small value of r illustrates the inappropriateness of fixed grid algorithms for this type of problem and hence the need for an alternative strategy.

Despite these problems, the results match very well whilst the frost layer is growing, however any similarity disappears after X_0 reaches its maximum value, as in Figure 3.4e.

It is quite common for authors to only examine the frost depth up to the stationary point and not to quote results after it. Indeed, authors such as Padmanabhan and Krishnamurthy [3] approximate the thawing stage in their calculations by simply extrapolating linearly the frost depth curve back to zero. However, it is obvious from Figure 3.4f that the thawing stage needs a great deal of further investigation, and this is undertaken here.

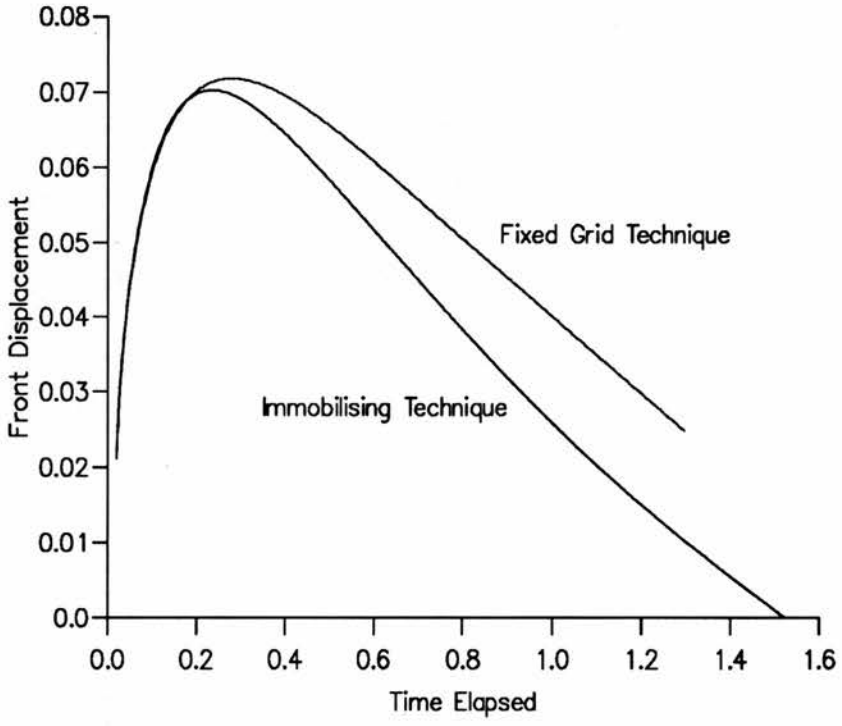

Figure 3.4f: Graphical comparison of the fixed grid technique using 2020 mesh spacings and the immobilising transformation technique using 60 mesh spacings for start time t_0=0.02.

§3.5: Improving the accuracy of the fixed grid technique by introducing a dual mesh size system.

In an effort to increase the maximum value of r attained at the stationary point, the spacings may be compressed into the region (X=0,2), whilst still keeping the 2000 sub-intervals in the solidified layer / liquid region; see Figure 3.5a.

Thus the boundary X_0 will progress to a maximum grid point value of r≈70. Hence more grid points are compressed into the transient solidified region ($X=0,X_0$), in an effort

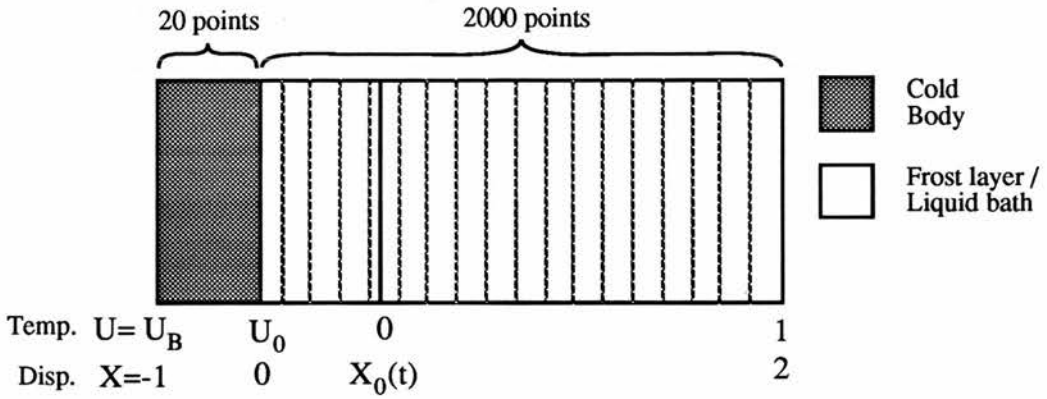

Figure 3.5a: Amended layout of the fixed grid problem using the spatial region (X=0,2).

to improve the accuracy. Since the grid spacings are reduced to one tenth of their original size, the time increment k used must be decreased accordingly to maintain stability. This in turn means more time steps for the full lifetime of the frost layer and hence a drastic reduction in the efficiency of the method.

Start time =0.02	X_{max}	turn time
fixed grid (0,10)	0.0701	0.259
fixed grid (0,2)	0.0717	0.279
immobilising technique	0.0702	0.235

Table 3.5a: Comparison of estimated location of stationary point using compressed grid spacings, coarse spacings and immobilising technique.

Table 3.5a shows a comparison between the immobilising technique, a (X=0,10) fixed grid and a (X=0,2) fixed grid technique, all with starting time t_0=0.02. Considering that we have reasonable confidence in the immobilising technique, it is seen that the idea of compressing all 2000 grid spacings into the region (X=0,2) yields an unsatisfactory degradation in the accuracy of the time to reach the stationary point t_{turn}. The value X=2 should be a spatial point where any effect of the heat transfer process is insignificant. Figure 3.5b shows the resultant temperature profile in the spatial region (X=0,2) around the time t_{turn} and indicates that this assumption is far from true. The system is thus reverted to one with a spatial region (X=0,10).

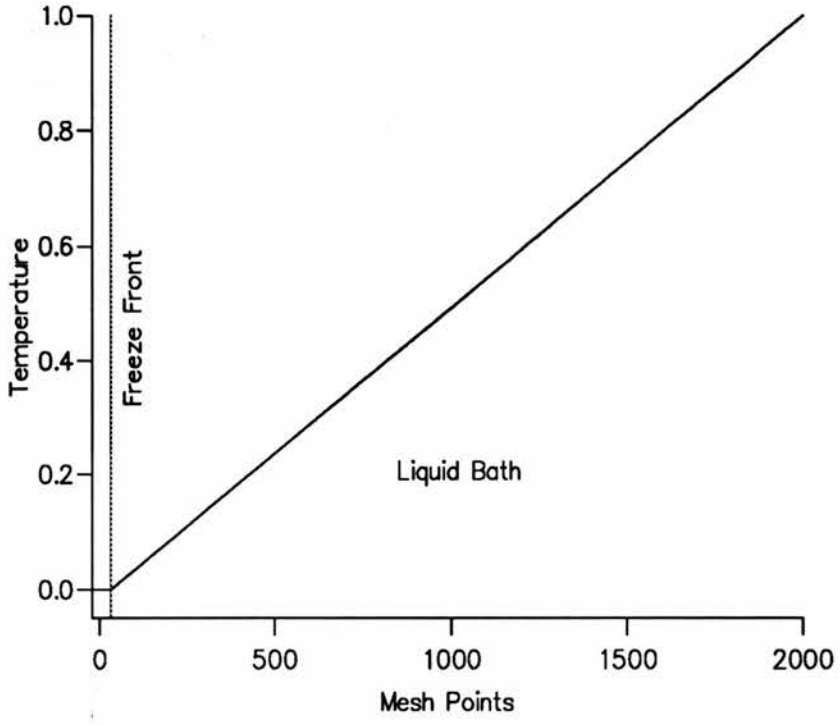

Figure 3.5b: Temperature profile at a time around the stationary point.

Having reverted back to a spatial region (X=0,10), the task in hand is to compress as many points as possible into the transient region, but not so many as to deem the process too massive to solve within a reasonable time span. Clearly the simplest way is to use two different spatial mesh sizes in the liquid bath. Thus the region (0,10) is divided into two sub-regions, such that the transient frost layer always remains within one sub-region, as in Figure 3.5c.

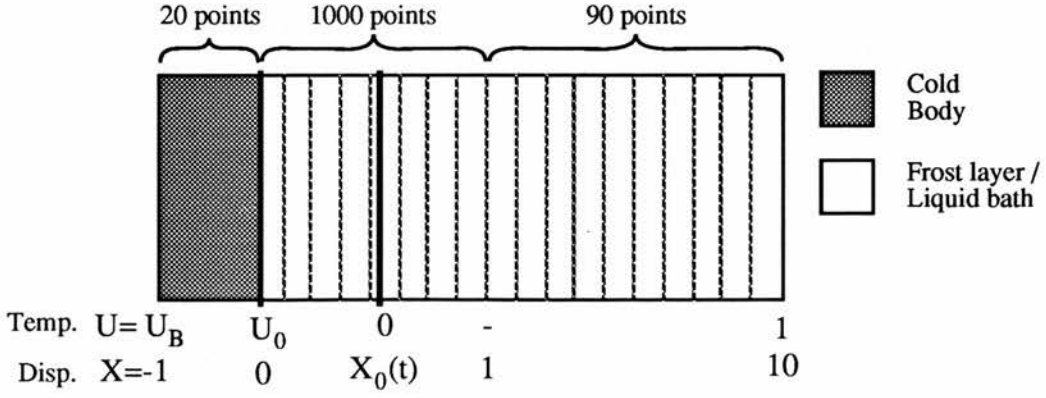

Figure 3.5c: Double mesh size fixed grid layout used.

As previously shown, the maximum frost depth $X_{0,max}$ is much less than unity and thus a possible two grid system would be to split the spatial region (X=0,10) into sub-regions (X=0,1) and (X=1,10). The range (X=0,1) has a fine mesh of size h=0.001 in order to follow the transitional period more effectively, by forcing more grid points into the transient frost layer. The sub-region (X=1,10) has a much coarser mesh of size h=0.1, so as to cover a large enough spatial range efficiently.

This system is solved in exactly the same way as the previous ones, however there is an additional boundary equation at X=1, the interface between the two mesh sizes. The equation used is generated by simply making an approximation of the usual diffusion equation at the interface. This is done by utilising the Lagrangian formulation of the double derivative equation (3.2b). There appears to be no violent temperature changes at this interface, so this technique should be sufficient in the circumstances. Thus at the interface of the two grids, which corresponds to the mesh point q=1000, an appropriate approximation of the diffusion equation is:

$$U_{1000}^{p+1} = U_{1000}^{p} + 2k\left\{ \frac{U_{999}^{p}}{h_2(h_2+h_3)} - \frac{U_{1000}^{p}}{h_2 h_3} + \frac{U_{1001}^{p}}{h_3(h_2+h_3)} \right\} \quad \textbf{(3.5a)}$$

Of course, the starting solution remains the same as before and the numerical algorithm is cycled for large time, giving results shown in Figure 3.5d.

This arrangement clearly yields more satisfactory results, however there is still a discrepancy between the two techniques for the starting time t_0=0.02 beyond the stationary point. Also, superimposed onto the results is the immobilising technique results graph with the more realistic starting time of t_0=0.0001. Ideally we would hope to gain some match between the fixed grid results and this curve, but the concept of an optimum start time appears to be causing great difficulties. It would seem an obvious step to make the liquid mesh so fine so as to be able to start the algorithm early enough to find an optimum t_0.

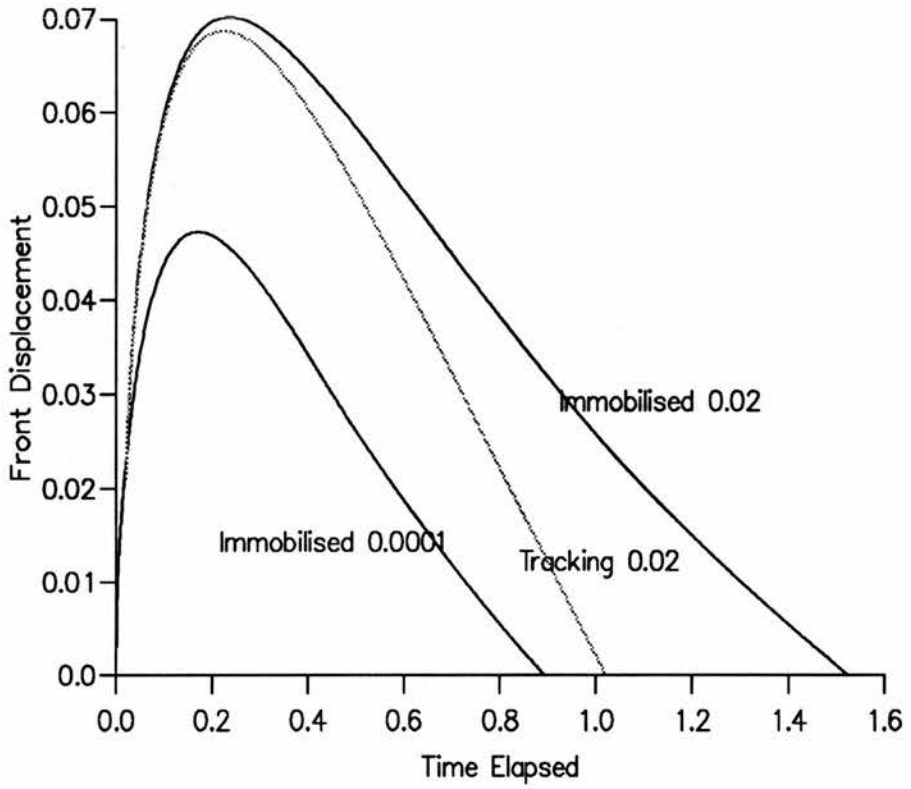

Figure 3.5d: Comparison of fixed grid (X=0,10) dual mesh size method with immobilising technique with starting time t_0=0.02 and immobilising technique with starting time t_0=0.0001.

However, if a finer mesh is utilised, the stability condition (3.3b) dictates that a smaller time increment is necessary. Also, since more mesh points would be used, then more calculations per time step would be necessary. This too would exacerbate the problem. In its present form the fixed grid technique requires approximately 45 hours of processing time and any increase in this would be ridiculous to propose. (Note that this is compared to less than 3 hours for the immobilising transformation technique for the same problem).

In an attempt to justify the optimum start time argument, another path is available. In §2, two special cases were examined; that of continuous freezing and that of limiting growth. Both these cases have known large time solutions. In the case of continuous freezing, the starting solution is the formal mathematical solution and thus we would expect that the start time used would bear no effect on the accuracy achieved. The case of limiting growth on the other hand would be expected to be inaccurate for start times such

as t_0=0.02, but improve as this time is decreased towards the optimum. It is these two cases that are examined.

§3.6: Continuous freezing - the immersed body temperature held fixed at its initial value.

As explained in §2.8, the temperature profile in the immersed body is held fixed at its initial temperature U_B for all time, so that the solution to the problem is that of the starting solution. The freeze front position is thus governed by the equation:

$$X_0 = \lambda \sqrt{\eta_2 t} \ .$$

The equations used are again altered slightly to cater for this change. Since the cold body temperature profile remains at a fixed temperature U_B, it can be omitted from the algorithm, as too can the equation at X=0, the immersed cold body / solid interface, since here $U=U_B$ for all time. The other equations used in the method remain unchanged.

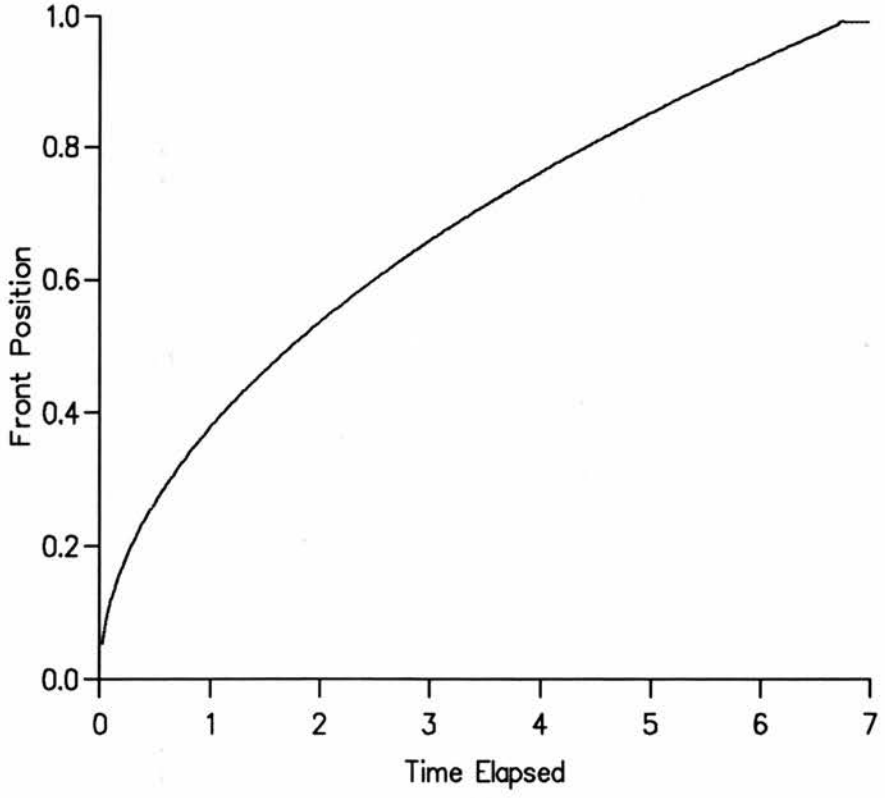

Figure 3.6a: Graph of displacement against time for fixed grid technique holding the cold body temperature fixed.

Since the formal mathematical solution is that of the starting solution, it is postulated that the start time t_0 is irrelevant, thus it is chosen to be t_0=0.02. The grid set up remains the same as before.

Again the parameters (2.8c) from Tadjbakhsh and Liniger [2] are used to test the method, and freeze front lifetime is shown in Figure 3.6a. The percentage error in the calculated boundary displacement X_0 is plotted against time in Figure 3.6b.

The percentage error remains very small (less than 3%) over the frost lifetime until $t \approx 6.75$, which corresponds to the time whereby $X_0 \approx 1$, and hence the freeze front reaches the bounds of the fine mesh and the start of the coarse mesh. At this stage the algorithm becomes invalid and the process is terminated

This result is an excellent one and is exactly as predicted.

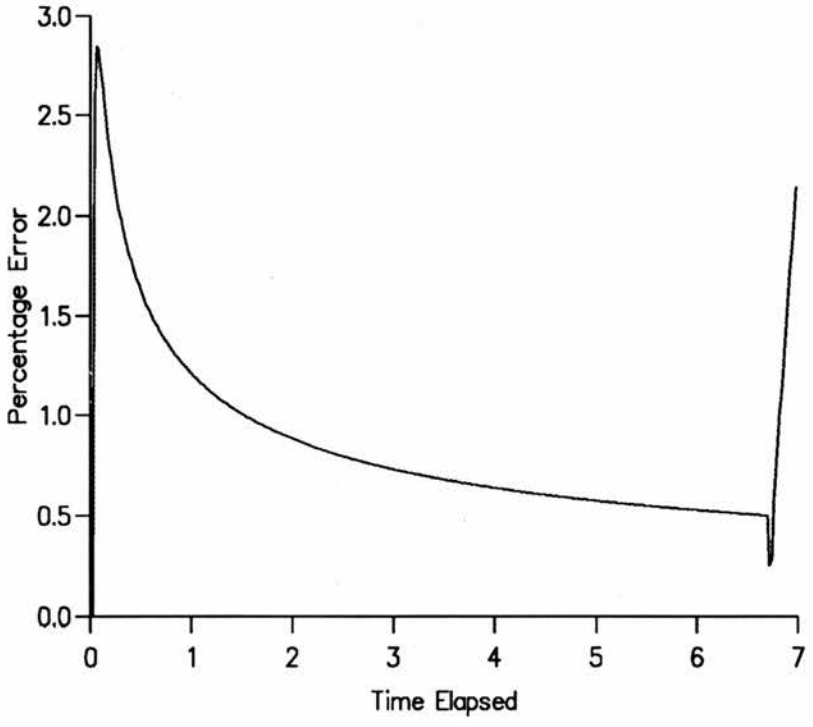

Figure 3.6b: Graph showing the percentage error in the boundary displacement for the lifetime of the method.

§3.7: Limiting growth - the liquid temperature initially at its freezing temperature.

As explained previously, if the temperature of the cold body is again allowed to vary, but the initial temperature of the liquid is its freezing temperature T_f, then the freeze front progresses to a limiting depth governed by

$$\lim_{t\to\infty}\left[\frac{X_0(t)}{a}\right] = \frac{(T_f - T_c)\,\rho_1\,c_1}{L\,\rho_2}. \qquad \textbf{(3.7a)}$$

and for the parameter regime (2.8c), this becomes

$$\lim_{t\to\infty}\left[\frac{X_0(t)}{a}\right] = 0.11025. \qquad \textbf{(3.7b)}$$

This case is a great deal easier to implement for the fixed grid method than the immobilising transform technique, since the liquid temperature profile described in §3.5 is merely set to U=0. Again the algorithm is run for large time, such that the front speed is less than 10^{-8} and so the freeze front is deemed to have reached its limiting value. Table 3.3 shows the results for varying start times t_0.

Start Time	Limiting Depth	Time Elapsed	% Error
0.02	0.126956	2.018	15.2
0.01	0.124009	2.008	12.5
0.005	*0.121185*	*3.005*	*9.92*

Table 3.7a: The effect of t_0 on limiting depth of the freeze front.

Again the results are as expected. Here the starting solution is not the exact solution and the accuracy is improved with the earlier start times (towards the optimum). The smallest possible start time for the given grid is t=0.01 which corresponds to r=2, so the result given for the start time t=0.005 is such that the algorithm is initially invalid. To reduce the start time t_0 further a finer grid has to be employed. Further compression of the liquid grid mesh in the full problem is deemed to be a ridiculous burden on CPU time. It is

obvious that fixed grid techniques of this type are not the most effective method of solution to the problem in question.

§3.8: Summary of the fixed grid technique.

Initially the aim of this chapter was to validate the results obtained by using the immobilising transformation technique in §2. This validation was to be achieved by use of a fixed grid technique proposed by Crank [13] and utilised by other authors for the same purpose. The method uses Lagrangian interpolation across the moving boundary to cater for the unequal grid spacings which occur at the freeze front and is far less efficient than the immobilising technique.

The system of equations is first tested on a trial run of the full solution, revealing a flaw in the original theory. This flaw was eradicated by changing the spatial position whereby the freeze front is judged to have moved into an adjacent grid neighbourhood.

Having overcome this flaw, effectively the main problem was due to the inability to allow the freeze front to progress over a large enough number of grid points. Ideally more points should be in the solidified region. A first attempt to overcome this was by merely compressing all the mesh points used into a smaller spatial region. This proved to be incorrect, since this smaller region failed to successfully represent the complete temperature distribution in the liquid bath. The remedy to the problem was deemed to lie in a dual mesh system. A fine grid was used in the active region incorporating the frost layer and a coarse grid was used across the remaining liquid to represent a larger spatial region more efficiently.

The results gained from this method are good up until the stationary point, after which any resemblance between the two methods rapidly disappears. The problem is deemed to lie in the choice of a possible start time t_0 for the fixed grid algorithm. The mathematical structure of the technique dictates a minimum t_0. However, the evidence of chapter 2 suggests that an optimum time exists for the smooth transition from the starting solution to the numerical solution. The optimum value is well below the lower bound on t_0 imposed by the grid spacing. This is upheld by examination of two special cases, one for which t_0 is irrelevant and the other for which it is found to be crucial. The only possible way around

this problem would be to introduce a much finer mesh in the liquid region. This is not deemed to be a worthwhile venture, since the efficiency of the process would be hundreds of times lower. With the CPU time already at a high, this proposal is rejected.

It is therefore concluded that the use of fixed grid techniques of this type is inappropriate for a problem of this kind, where very small regions are involved. Hence this must cast doubt on the validity of the results of Padmanabhan and Krishnamurthy [3]. However, the method is shown to be useful for problems such as those described in §4.

§4. An extension of the immobilising transformation technique to freezing and thawing of finite bodies.

§4.1: Introduction.

An interesting progression of the problems discussed in §2 and §3 is to consider the immersing of a cold body in a finite liquid bath of the same material, as opposed to an infinite one of a different material. This arrangement has certain features which the previous problem did not possess:

i) The number of regions to consider is reduced from three to two.

ii) The inevitability of the return of the freeze front to its original position is eliminated, since there is no longer an infinite quantity of heat in the liquid bath. Thus, unlike the single characteristic motion of the infinite liquid problem, the resulting heat transfer process may progress in many different ways dictated by the parameters involved in the system.

iii) The cold body can melt after the thawing of a frost layer.

iv) The cold body can melt initially. Thus, negative values of λ in the starting solution are meaningful.

v) Since the warm liquid region is finite, steady state solutions are also possible which are independent of the initial motion of the freeze front. If the liquid region is relatively large, the process may be terminated when the immersed cold body thaws completely leaving purely liquid, or conversely, if the liquid region is small enough it may freeze completely leaving purely solid.

The aim of this chapter is to model this problem using an immobilising transformation technique similar to that developed in §2 and solve it using a numerical algorithm involving explicit finite difference approximations to the equations formulated. The results from this system will then be compared with the results generated using the fixed grid technique developed in §3 and also with the known steady state solutions. The physical simplifications §2.1(i-vii) concerning uniformity, symmetry etc. barring the infinite approximation 2.1(iii) are again implemented and thus the governing parameters are exactly those given in §2.1.

§4.2: Physical configuration.

The physical layout of the system is the same as that given in Bell and Wedgwood [16] and is shown in Figure 4.2a.

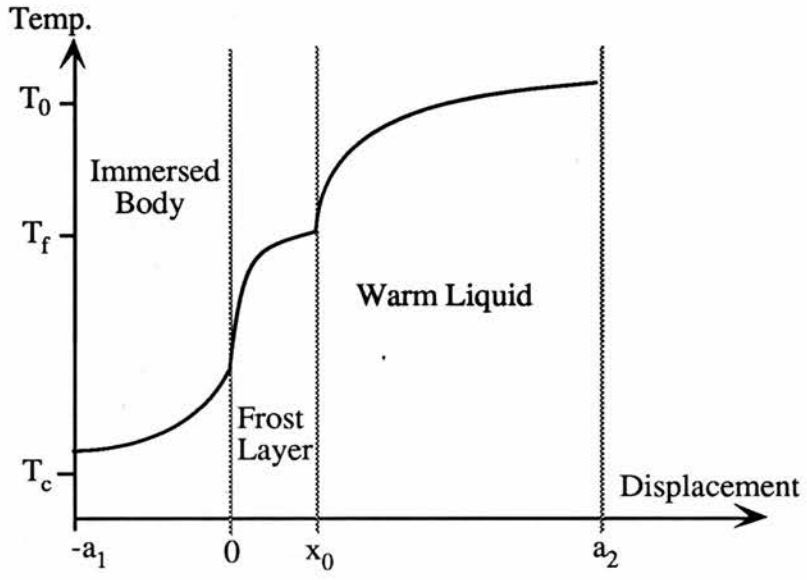

Figure 4.2a: Typical freezing temperature profile for finite bodies.

The system may be characterised by considering a frozen body at temperature T_c being immersed in a finite bath of warm liquid of the same material at temperature T_0 where $T_c < T_f < T_0$, T_f representing the freezing temperature of the liquid. The immersed body has size a_1 and the liquid has size a_2, the origin taken to be the position of the solid / liquid interface at time t = 0. In physical and computational terms, this system is a slight simplification of that described earlier and published in [15], since there are essentially only two finite regions of solid and liquid material. However the problems are compounded by the fact that the front may approach either boundary a_1 or a_2, and may cross the initial freeze interface at the origin (illustrated in Figure 4.2b which shows the possible formats of freeze front history for large time). This causes difficulties in choosing transformations since if the freeze front approaches a_1 say, indicating total thawing, the solid region becomes very small, yet in the numerical algorithm it will contain the same number of spatial mesh points.

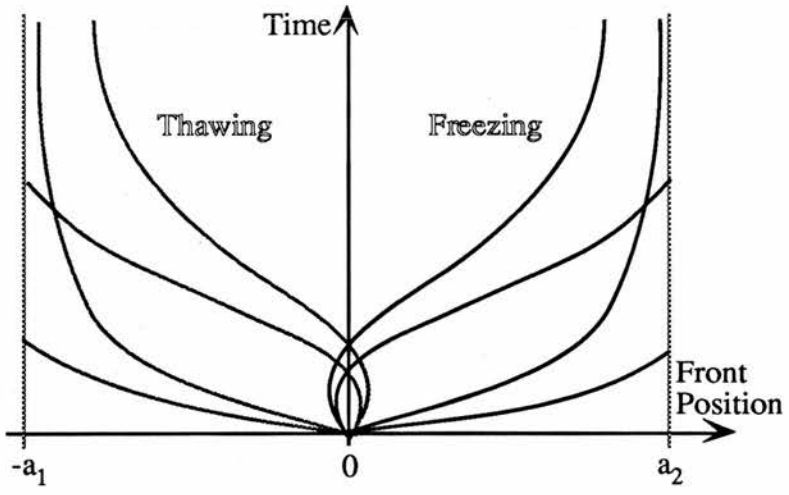

Figure 4.2b: Possible large time solutions, indicating the feasibility of many freeze front motions.

§4.3: Basic equations representing the system.

Mathematically the system is very similar to that of the infinite liquid case, and can be described by diffusion equations in the two regions:

$$\frac{\partial T_i}{\partial \tau} = \kappa_i \frac{\partial^2 T_i}{\partial x^2} \quad , \qquad i = 1,2, \tag{4.3a}$$

where 1 denotes the solid and 2 denotes the liquid, together with the relevant boundary conditions. At the boundary $x=-a_1$ there exists a line of symmetry of the system and thus no heat flow occurs across it:

$$\left.\frac{\partial T}{\partial x}\right|_{-a_1} = 0 \qquad x = -a_1. \tag{4.3b}$$

Since the thermal parameters of the immersed cold body will be the same as those of the frost layer there is no justification for considering the interface between them. The flow of heat across this interface is continuous. However, at the freeze front $x=x_0(\tau)$, two conditions hold. Firstly the temperature on the interface is always the freezing temperature:

$$T(x_0(\tau),\tau) = T_f \qquad x=x_0(\tau), \tag{4.3c}$$

and secondly there is heat balance during the phase change:

$$K_1 \frac{\partial T}{\partial x}\bigg|_{x_0-} - K_2 \frac{\partial T}{\partial x}\bigg|_{x_0+} = L\,\rho_1 \frac{dx_0}{dt} \; . \tag{4.3d}$$

There are various possibilities for a boundary condition at the boundary a_2 involving heat flux across it, but the simplest case is that of no heat transfer consistent with the liquid being insulated (zero heat flow), which is mathematically equivalent to the boundary condition imposed at a_1, thus:

$$\frac{\partial T}{\partial x}\bigg|_{a_2} = 0 \; . \tag{4.3e}$$

The initial conditions for the process are:

$$T(x,0) = T_c \qquad -a_1 < x < 0 \quad , \tag{4.3f}$$

representing the initial immersed cold body temperature being uniform at temperature T_c

$$x_0(0) = 0 \qquad , \tag{4.3g}$$

representing the initial solid / liquid interface being at origin, and

$$T(x,0) = T_0 \qquad 0 < x < a_2 \quad , \tag{4.3h}$$

representing the ambient temperature of the liquid being uniform at the temperature T_0.

§4.4: Transformation technique.

The number of governing parameters is reduced by non-dimensionalising as before, thus simplifying the equations using the transformations:

$$U = \frac{T - T_f}{T_0 - T_f} \quad \text{and} \quad t = \frac{\kappa_2 \tau}{a_1^2} \, . \qquad \textbf{(4.4a)}$$

The sizes of the two regions are eliminated by considering a parameter defined as the ratio of the two. This is gained by using:

$$X = \frac{x}{a_1} \quad \text{and} \quad X_1 = \frac{a_2}{a_1} \, . \qquad \textbf{(4.4b)}$$

The system now looks like that shown in Figure 4.4a.

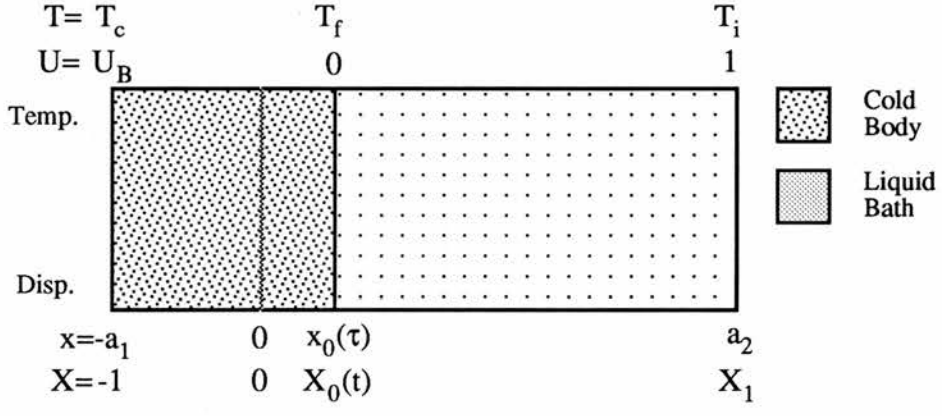

Figure 4.4a: System after non-dimensionalising, before immobilisation.

The main difficulty with the immobilising transformation technique is obviously how to go about immobilising the freeze front. In §4.1 it was explained that the freeze front may ultimately return and pass through its initial position to assume a negative value, thus indicating thawing of the original immersed body. This implies that at some point in time the freeze front will pass through x=0. Therefore, terms involving $[x_0(t)]^{-1}$ must be avoided in the transformed equations, since this creates a division by zero as the freeze

front passes through the origin. With this in mind it is suggested that the following transformations be used:

$$z = \frac{X - X_0}{1 + X_0} \quad \text{in the solid region;} \qquad \textbf{(4.4c)}$$

$$z = \frac{X - X_0}{X_1 - X_0} \quad \text{in the liquid region.} \qquad \textbf{(4.4d)}$$

These transformations map the immersed cold body and the liquid bath i.e. the regions $(-1,X_0)$ and (X_0,X_1) onto the regions $(-1,0)$ and $(0,1)$ respectively, thus holding the freeze front fixed at $z=0$. The diffusion equations (4.3a) governing the process become:

$$\frac{\partial U}{\partial t} = \frac{\eta}{(1 + X_0)^2}\frac{\partial^2 U}{\partial z^2} + \frac{(1 + z)}{(1 + X_0)}\frac{dX_0}{dt}\frac{\partial U}{\partial z}, \quad -1 < z < 0; \qquad \textbf{(4.4e)}$$

$$\frac{\partial U}{\partial t} = \frac{1}{(X_1 - X_0)^2}\frac{\partial^2 U}{\partial z^2} + \frac{(1 - z)}{(X_1 - X_0)}\frac{dX_0}{dt}\frac{\partial U}{\partial z}, \quad 0 < z < 1; \qquad \textbf{(4.4f)}$$

where $X_0(t) = \frac{x_0(t)}{a_1}$ and $\eta = \frac{\kappa_1}{\kappa_2}$.

Both these transformations satisfy the condition imposed concerning terms $[x_0(t)]^{-1}$, however as the freeze front approaches $x = a_2$ or $x = -a_1$ problems will undoubtedly still occur. When $X_0 \to -1$ or $X_0 \to X_1$ the corresponding diffusion equation becomes singular, so further considerations will have to be made.

The boundary conditions, equations (4.3b-e) transform in a similar manner to give at the centre of the immersed body:

$$\left.\frac{\partial U}{\partial z}\right|_{-1} = 0 . \qquad \textbf{(4.4g)}$$

Likewise at the extremity of liquid region:

$$\left.\frac{\partial U}{\partial z}\right|_{z=1} = 0 \quad . \qquad \textbf{(4.4h)}$$

At the freeze front the boundary conditions become:

$$\frac{\beta}{(1 + X_0)} \left.\frac{\partial U}{\partial z_1}\right|_{0-} - \frac{1}{(X_1 - X_0)} \left.\frac{\partial U}{\partial z_2}\right|_{0+} = \alpha \frac{dX_0}{dt} \qquad \textbf{(4.4i)}$$

and $\qquad U = 0 \qquad$ at $z = 0$,

This moving boundary condition also suffers from the problem of division by zero as $X_0 \rightarrow -1$ or $X_0 \rightarrow X_1$, so this equation too must be altered in the event of total thawing or freezing. The dimensionless thermal parameters α and β which occur in these transformed equations are described as:

$$\alpha = \frac{L}{c_2 (T_0 - T_f)} \quad \text{and} \quad \beta = \frac{K_1}{K_2} \quad ,$$

again α essentially being the reciprocal of the Stefan number for the process.

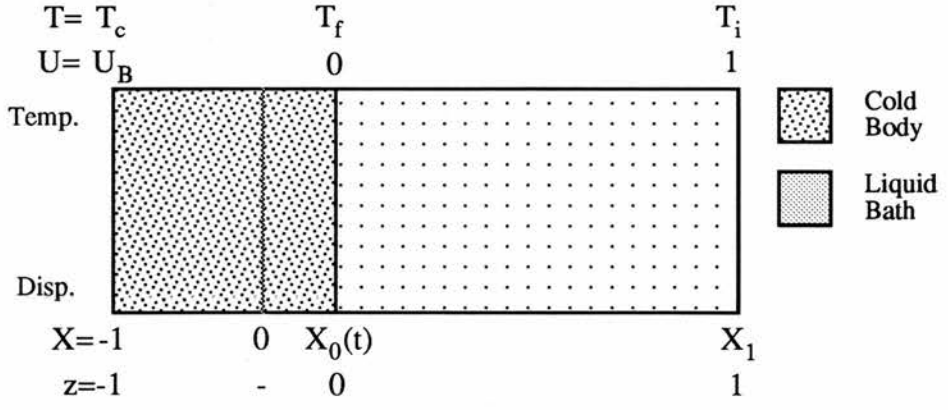

Figure 4.4b: System after transformation.

The initial conditions are transformed in a similar manner to become:

$$X_0(0) = 0 \ ,$$

$$U = 1 \ , \text{ for } \ 1 < z \leq 2 \ ,$$

and $$U = U_B = \frac{T_c - T_f}{T_0 - T_f} \ , \text{ for } \ -1 \leq z_1 < 0 \ .$$

§4.5: Starting solution.

In the same way as for the infinite liquid case, the mathematical model contains a discontinuity at X=0 when t=0 and thus a small time starting solution must be used to overcome this. If the time elapsed is taken to be very small then effects due to the sizes of the immersed cold body and liquid region may be ignored, since heat transfer is confined to the locality of the interface between the two regions. A suitable starting process is thus to assume error function profiles for the dimensionless temperature U as described by Carslaw and Jaeger [4]. The initial velocity of the freeze front is thus

$$X_0(t) = \lambda \sqrt{t} \quad \text{for } 0 < t \leq t_0 \ . \qquad \textbf{(4.5a)}$$

The dimensionless temperature profiles in the two regions are:

$$U = U_B \left\{ \frac{\operatorname{erf}\left[\frac{\lambda}{2\sqrt{\eta}}\right] - \operatorname{erf}\left[\frac{x}{2\sqrt{\eta_L}}\right]}{\operatorname{erf}\left[\frac{\lambda}{2\sqrt{\eta}}\right] + 1} \right\} \qquad \textbf{(4.5b)}$$

in the solid and

$$U = \frac{\operatorname{erf}\left[\frac{x}{2\sqrt{t}}\right] - \operatorname{erf}\left[\frac{\lambda}{2}\right]}{\operatorname{erfc}\left[\frac{\lambda}{2}\right]} \qquad \textbf{(4.5c)}$$

in the liquid respectively.

The parameter λ is found by substituting these temperature profiles, equations (4.5b & c) into the moving boundary condition, equation (4.3d) to give a transcendental equation:

$$\frac{\alpha \lambda \sqrt{\pi}}{2} + \frac{\beta U_B e^{-\lambda^2/4\eta}}{\sqrt{\eta}\left\{\operatorname{erf}\left[\frac{\lambda}{2\sqrt{\eta}}\right] + 1\right\}} + \frac{e^{-\lambda^2/4}}{\operatorname{erfc}\left[\frac{\lambda}{2}\right]} = 0, \qquad \textbf{(4.5d)}$$

which is solved for λ using a Newton Raphson algorithm.

§4.6: Numerical method.

Having developed the set of transformed equations and generated a suitable starting solution, the full solution to the problem is found by approximating these equations by explicit finite difference formulae and deriving a numerical algorithm to progress the solution over small time increments from the starting profiles. The explicit approximations used are identical to those in §2 and are given in equation (2.7a). The computational procedure for updating the unknowns is essentially of the same structure as that used in §2.7. Due to the sensitive nature of such explicit approximations, it is necessary to eliminate possible cancellation errors. This is done by selecting the time increment Δt such that all terms remain positive, as before, thus yielding bounds:

$$\Delta t < \frac{h^2(1+X_0)^2}{2\eta} \qquad \textbf{(4.6a)}$$

and

$$\Delta t < \frac{h^2(X_1-X_0)^2}{2}, \qquad \textbf{(4.6b)}$$

which arise from the explicit approximations of each of the transformed diffusion equations in the solid and liquid region respectively.†

Both of these bounds become appropriate at some stage during the freeze front lifetime since neither remains smaller than the other throughout the process.

For the results that follow the common mesh size h in the two regions is taken to be 0.05 throughout. The best time to switch from the starting solution to the numerical algorithm will be discussed at a later stage.

§4.7: Initial motion and steady states of the freeze front.

Unlike the infinite liquid region scenario, the freeze front may initially progress so as to freeze the liquid or to thaw the solid thus implying positive or negative initial velocity. It is also possible, given the right thermal parameters and initial conditions, for the initial velocity of the front to be zero. The initial velocity of the front is fundamental to the solution, however behaviour of the system is complex so the effects of altering the parameters α, β, η and U_B are examined. A basic parameter regime is taken as follows:

$$\eta = 12 \qquad \beta = 2 \qquad \alpha = 2 \qquad U_B = -2 \tag{4.7a}$$

which corresponds to a value for the velocity characterisation parameter as $\lambda = 0.0611839$ thus indicating initial freezing, albeit slowly.

Given this regime, the effect on λ of altering each of these parameters in turn holding the others fixed is examined in Figures 4.7a-d.

† The spatial grid size h is constant across the two regions.

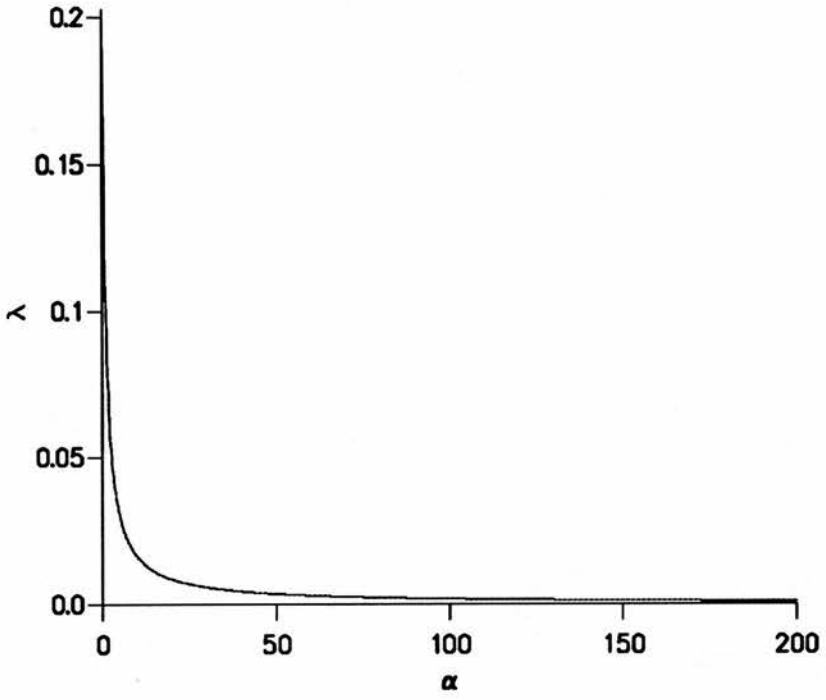

Figure 4.7a: The effect of varying α on the value of λ.

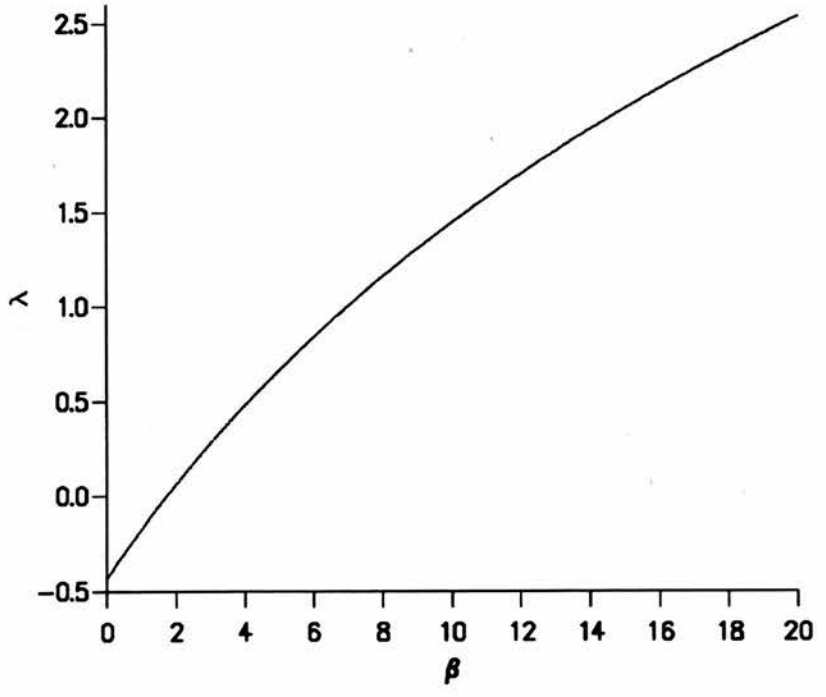

Figure 4.7b: The effect of varying β on the value of λ.

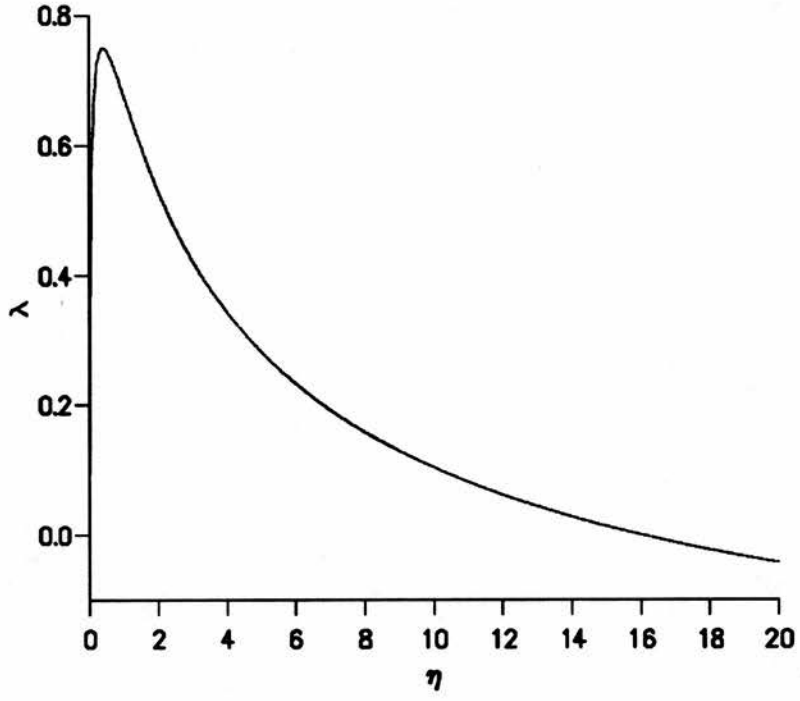

Figure 4.7c: The effect of varying η on the value of λ.

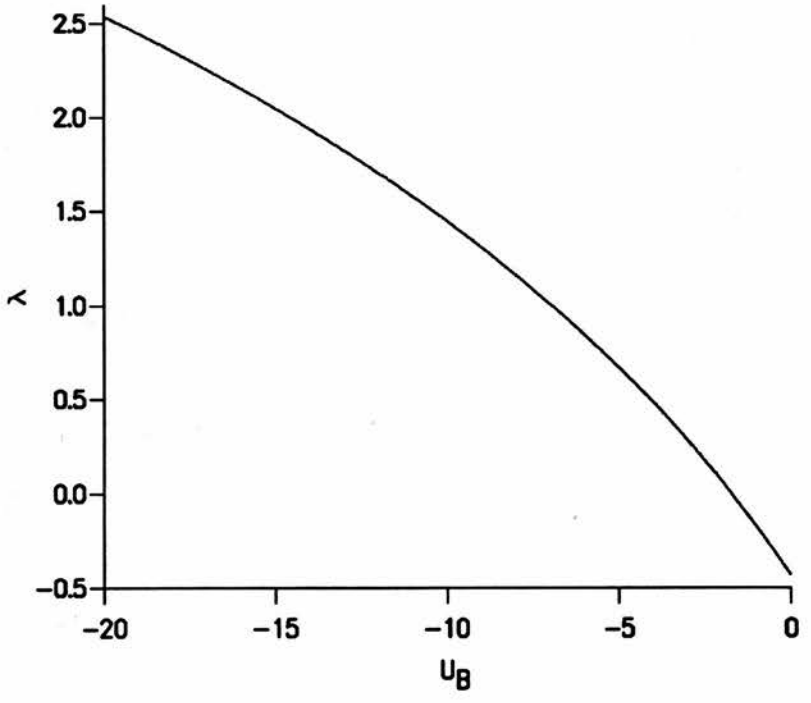

Figure 4.7d: The effect of varying U_B on the value of λ.

As can be seen from the graphs in Figure 4.7a-d, the effects of each parameter on λ are relatively simple, however it is only β and U_B that truly dictate the sign of λ and it is these that are most easily varied to examine different starting scenarios. The effect of each parameter on the value of λ is most easily characterised by determining a single expression involving all the parameters which dictates the initial direction of motion of the front. This is done by considering the case when $\lambda \to 0$ in equation 4.5d. By either setting $\lambda = 0$ in equation 4.5d or by generating a small λ expansion, the limiting case corresponds to the parameters satisfying:

$$\frac{\beta\, U_B}{\sqrt{\eta}} + 1 = 0. \qquad \textbf{(4.7b)}$$

It is thus appropriate to define a new parameter ϑ from this such that

$$\vartheta = -\{\, \beta\, U_B + \sqrt{\eta}\, \}. \qquad \textbf{(4.7c)}$$

This parameter ϑ has the property that $\lambda = 0 \Rightarrow \vartheta = 0$ and the sign of λ is the same as the sign of ϑ (see Figure 4.7e), so it is possible to determine the direction of initial motion of the freeze front quickly and simply given any parameter regime.

In physical parameters the equation (4.7c) is given by

$$\vartheta = -\left\{ \frac{K_1}{K_2}\left[\frac{T_c - T_f}{T_0 - T_f}\right] + \sqrt{\frac{\kappa_1}{\kappa_2}}\ \right\}. \qquad \textbf{(4.7d)}$$

This expression yields little in the way of directly understanding the initial motion of the freeze front, but it does show that the initial velocity is dependent on the relative sizes of thermal conductivity and diffusivity in the two regions as well as the initial temperature distribution relative to the freezing temperature.

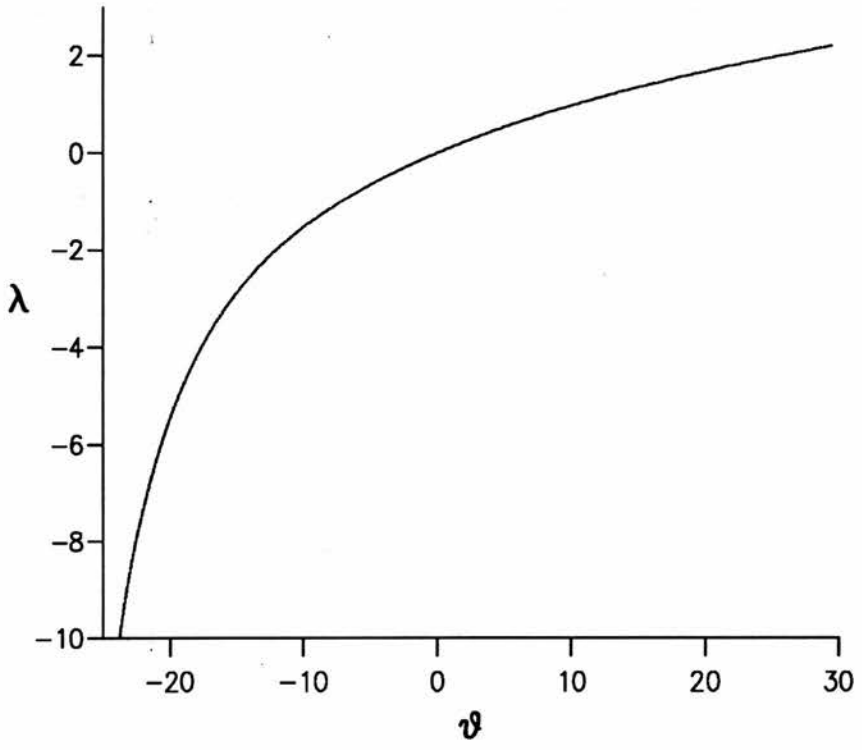

Figure 4.7e: Graph showing the relationship between the velocity constant λ and the characterisation parameter ϑ.

The freeze front motion is also dependent on the existence of any limiting large time solutions. These are examined by considering the net loss or gain of heat from the two regions once a steady state has been established:

$$\text{Gain in heat of solid} = c_1 \rho_1 a_1 (T_f - T_c) \qquad \textbf{(4.7e)}$$

$$\text{Loss of heat of liquid} = c_2 \rho_2 a_2 (T_0 - T_f) + L \rho_2 X_0. \qquad \textbf{(4.7f)}$$

Equating these and non-dimensionalising gives

$$X_1 + \alpha X_0 + \frac{U_B \beta}{\eta} = 0. \qquad \textbf{(4.7g)}$$

For a given set of material parameters, it is obvious that by altering the value of X_1 (the relative size of the liquid region with respect to the immersed cold body), the limiting front position may be predicted and is independent of initial motion of the front, since the value of X_1 has no bearing on the value of λ. For the parameter regime, equation (4.7a), this becomes:

$$\lim_{t\to\infty}\left[\frac{x_0(t)}{a_1}\right] = \frac{1}{6}(1 - 3X_1)\,. \qquad \textbf{(4.7h)}$$

This gives a means of checking results obtained, especially if the limiting position of the front remains within the spatial bounds of the system i.e. $-1 < X_0 < X_1$, which corresponds to $^1/_9 < X_1 < {}^7/_3$ for the parameters here. For the case when the bodies are the same size, $X_1 = 1$ and the limiting front position should be at $X_0 = -1/3$.

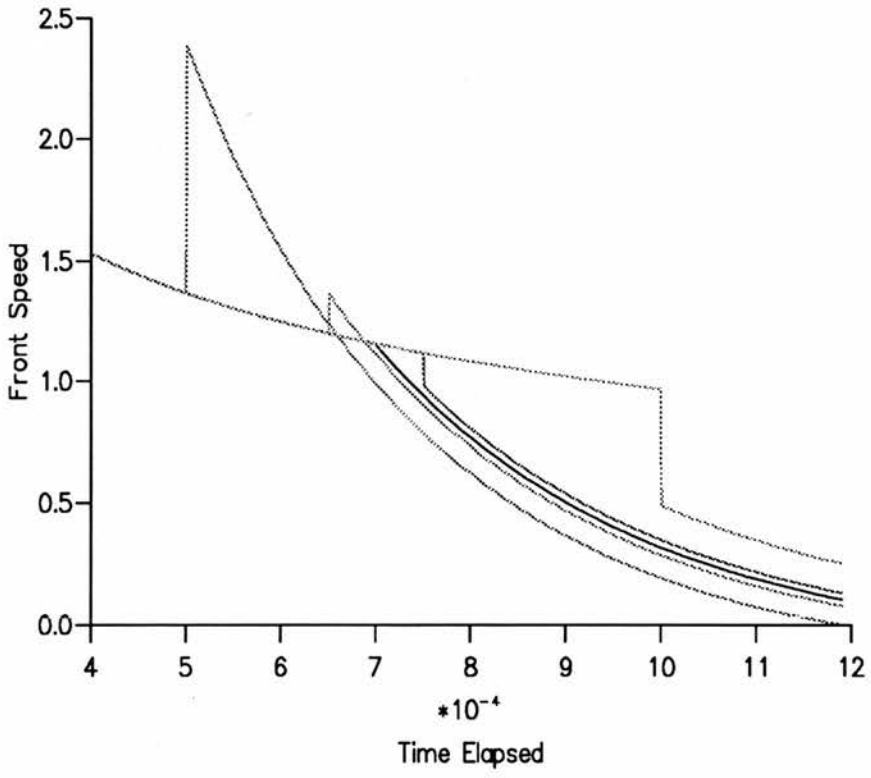

Figure 4.7f: The transition in front speed from the starting solution to the numerical solution showing the discontinuities encountered for start times other than the optimum one.

The start time t_0 is chosen for each parameter regime to ensure that a smooth transition is made from the starting solution to the numerical solution as in the previous two chapters. Here the system seems to be slightly more sensitive to the value chosen, as seen from Figure 4.7f which shows fairly substantial shifts in front speeds across the transition. Obviously the optimum start time is $t_0 = 0.0007$ from Figure 4.7f, however this value still produces a slight change in front speed in the change from the starting solution to the numerical algorithm, illustrated upon closer investigation in Figure 4.7g. The shift may seem to be harsh, but this start time gives by far and away the best transition and thus this discrepancy would appear to be unavoidable.

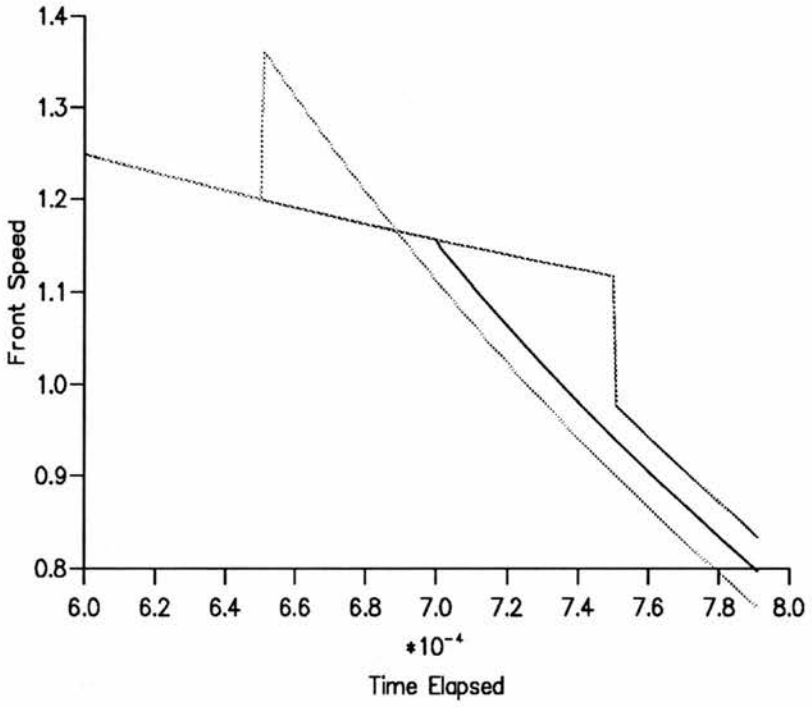

Figure 4.7g: The transition from the starting solution to the numerical solution showing the slight variation in front speed for the optimum commuted start time $t_0 = 0.0007$.

The front history calculated using the immobilising technique is shown in Figures 4.7h and i and clearly shows a good accuracy in the limiting front position (calculated $X_0 = -0.33987325$ which represents an error of less than 2.0%).

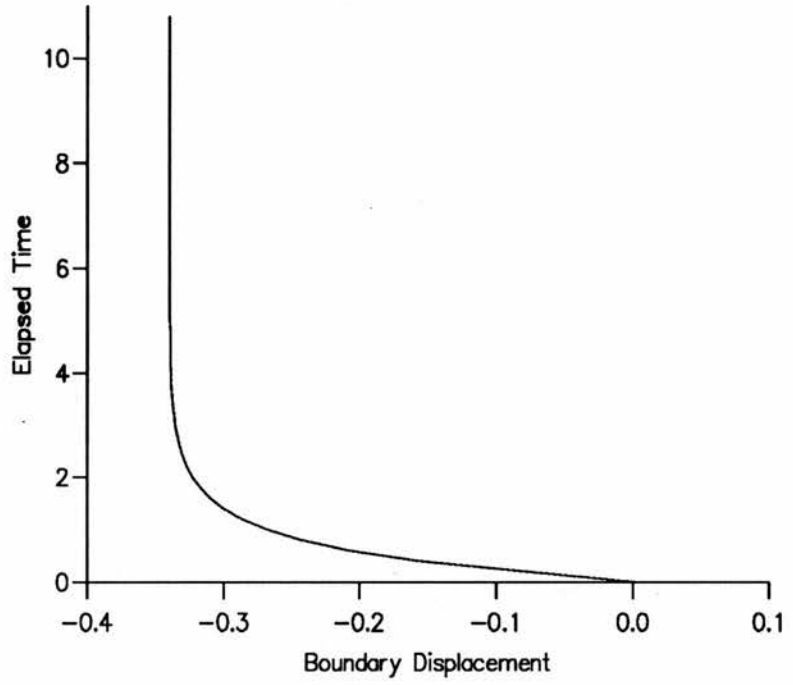

Figure 4.7h: Front position calculated using the immobilising transformation technique for the case $X_1 = 1$ (immersed body and liquid region initially the same size) for large time.

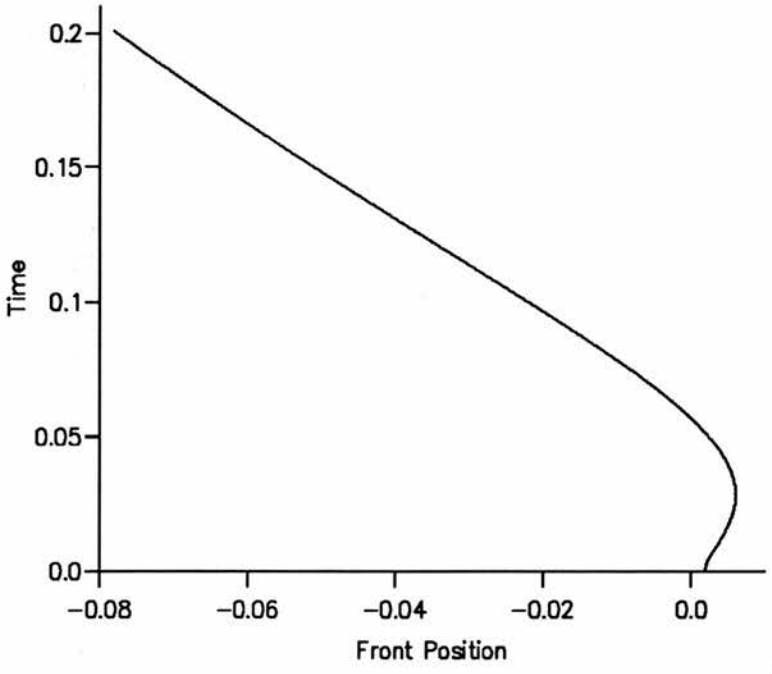

Figure 4.7i: Front position calculated using the immobilising transformation technique for the case $X_1 = 1$ for small time.

§4.8: Fixed grid technique.

Essentially the best way to check the validity of the earlier stages of the front lifetime calculated by this technique is to match it against an intrinsically differing method. The method adopted is that of using a fixed grid technique first described by Crank [13] and developed in §3. The two regions are again divided into a number of smaller sub-intervals and the front is tracked across these using a Lagrangian interpolation approximation of the boundary partial differential equations. The temperature profile away from the boundary is updated using explicit finite difference approximations of the diffusion equations as before, however the freeze front may exist (in fact spends virtually all of its time) between mesh points. Thus if the freeze front is to be considered in any calculations, a method must be used to cater for the uneven grid spacings caused by an intermediate position of the moving boundary. Thus an interpolation technique is adopted as in §3.

Here the initial solid and liquid regions are each sub-divided into 100 intervals. The front is allowed to progress from its starting position at a time t_0 across the mesh. A typical front position is shown in Figure 4.8a where the front is in the interval between the r^{th} and $r+1^{th}$ mesh points. The temperature profile is updated across the mesh points ...r-2, r-1, x_0, r+1, r+2..., the r^{th} mesh point being omitted from the equations for the time being.

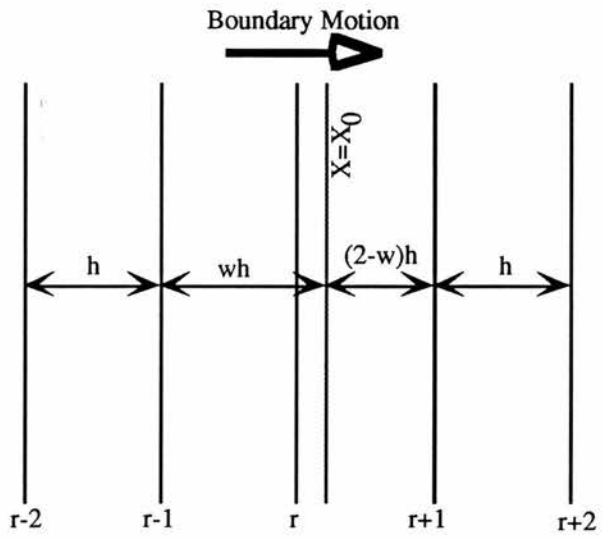

Figure 4.8a: Motion of the freeze front across the fixed grid.

When running the algorithm, the solid temperature profile is calculated up to the r-2th mesh point (note that all mesh spacings are equally sized) using the usual finite difference approximation to the diffusion equation. To calculate the new temperature value at the r-1th mesh point, a Lagrangian formulated diffusion equation is developed by interpolating over the old temperatures at the points r-2, r-1 and x_0 (ignoring the rth point) giving

$$U_{r-1}^{p+1} = U_{r-1}^{p} + \frac{2\,\kappa\,k}{h^2}\left\{ \frac{U_{r-2}^{p}}{wp+1} - \frac{U_{r-1}^{p}}{wp} \right\} \qquad \textbf{(4.8a)}$$

h being the common mesh spacing in both the solid and liquid regions and (r+w)h being the front position at any time. Similarly, the new temperature value of the r+1th mesh point is found by interpolating over the old temperature values at the points x_0, r+1, r+2 giving

$$U_{r+1}^{p+1} = U_{r+1}^{p} + \frac{2\,k}{h^2}\left\{ \frac{U_{r+2}^{p}}{3 - wp} - \frac{U_{r+1}^{p}}{2 - wp} \right\}. \qquad \textbf{(4.8b)}$$

The remaining liquid temperature profile over the points r+2, r+3... is calculated in a similar way to the solid temperature profile by finite difference approximation of the relevant diffusion equation in the liquid region.

Having generated equations to develop the temperature profiles in the two regions, all that remains is to formulate a method to track the freeze front. This is done by making a Lagrangian interpolation approximation to the usual moving boundary equation giving

$$\frac{\beta}{h}\left\{ \frac{wpU_{r-2}^{p}}{wp+1} - \frac{(wp+1)U_{r-1}^{p}}{wp} \right\} - \frac{1}{h}\left\{ \frac{(wp-2)U_{r+2}^{p}}{3-wp} + \frac{(wp-3)U_{r+1}^{p}}{wp-2} \right\} =$$

$$= \alpha\frac{dx_0}{dt} = \frac{\alpha\,h}{k}\{\, wp+1 - wp \,\} \qquad \textbf{(4.8c)}$$

which is used to calculate the updated value $wp+1$ at each cycle.

Since the freeze front must progress across the fixed grid, at some stage it must cross from one grid space to an adjacent one. In this case, the value of w^p must change by unity and since the relevant boundary mesh points have changed, a value must be found for the previously ignored temperature U_r.

When w^p increases by unity then the temperature value U_r is found by Lagrangian interpolation over the points r-2, r-1 and x_0. The interpolation is taken over these three points since the r^{th} grid line lies within the region $([r-1]h,x_0)$ and so all these points lie within the solidified layer:

$$U_r^{p+1} = \frac{(1-wp)\, U_{r-2}^{p+1}}{1+wp} + \frac{2\,(wp-1)\, U_{r-1}^{p+1}}{wp} \,. \qquad \textbf{(4.8d)}$$

The procedure presented by Crank [13] only covers the case when the solidified layer is growing, but for the system here, the solution must also be developed when thawing occurs. When w^p decreases by unity the temperature value U_r is found by interpolating over the points x_0, r+1 and r+2 since the r^{th} grid line lies within the region $(x_0,[r+1]h)$ and thus all these points lie within the liquid region:

$$U_r^{p+1} = \frac{2\,(2-wp)\, U_{r+1}^{p+1}}{3-wp} - \frac{(2-wp)\, U_{r+2}^{p+1}}{4-wp} \,. \qquad \textbf{(4.8e)}$$

The starting profile used is the same as that used earlier and the parameters are those in equation (4.7a). Again the start time is taken to be such that the transition from starting solution to numerical solution is the smoothest. To check the validity of the immobilising technique X_1 is chosen as unity, corresponding to a limiting front position of $X_0 = -1/3$. Figure 4.8b compares the results for the immobilising transform technique, the fixed grid technique and the theoretical front position for large time. Clearly the match is very good, but since more sub-intervals are used in the fixed grid method a better representation of the temperature profile is gained and its large time limit is slightly more accurate

(X_0 calculated = –0.33431539 which represents an error of only 0.3% compared to the immobilising error of 2.0%).

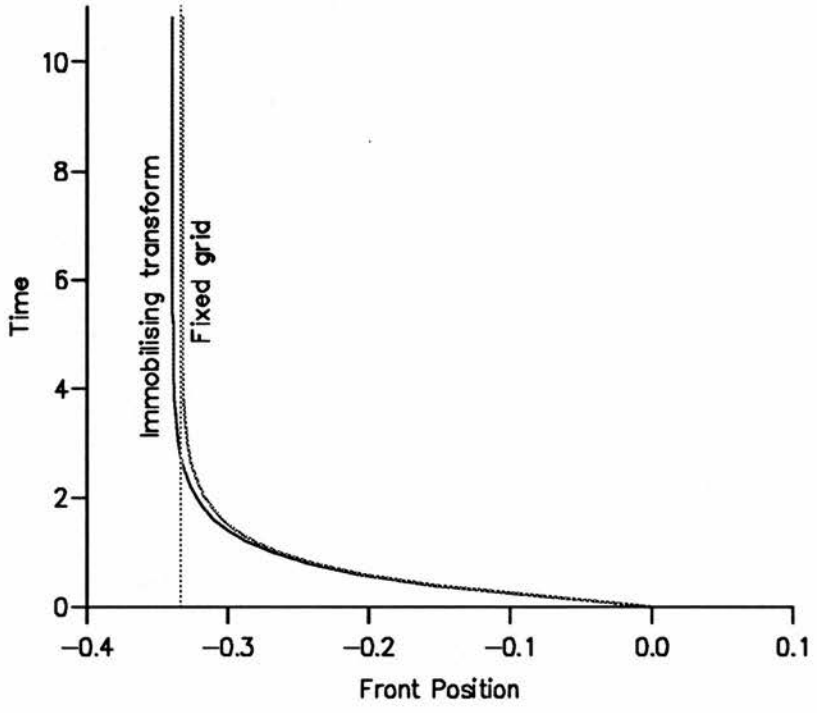

Figure 4.8b: Graphical comparison of the immobilising transformation technique and the fixed grid method for a finite liquid and $X_1 = 1.0$ for parameters 4.7a for large time.

Examination of the small time comparison between the two methods reveals a possible source of the slightly larger error for the immobilising transform technique. Both methods show a smooth transition from the starting solution to the numerical algorithm. For the immobilising transform technique this occurs at $t_0 = 0.0007$. The freeze front progresses in much the same way as for the fixed grid technique until a time about $t \approx 0.0018$ and then a kink occurs. The source of this anomaly is unknown, since all motion up to that time and during the brief period during which this kink occurs is smooth. The kink is not due to any sort of numerical instability, since even if the time step is decreased by a factor of one million the identical behaviour is exhibited. Clearly if the kink were not present then the

freeze front would follow much the same path as the fixed grid solution and indeed would generate a better result. This issue remains unresolved.

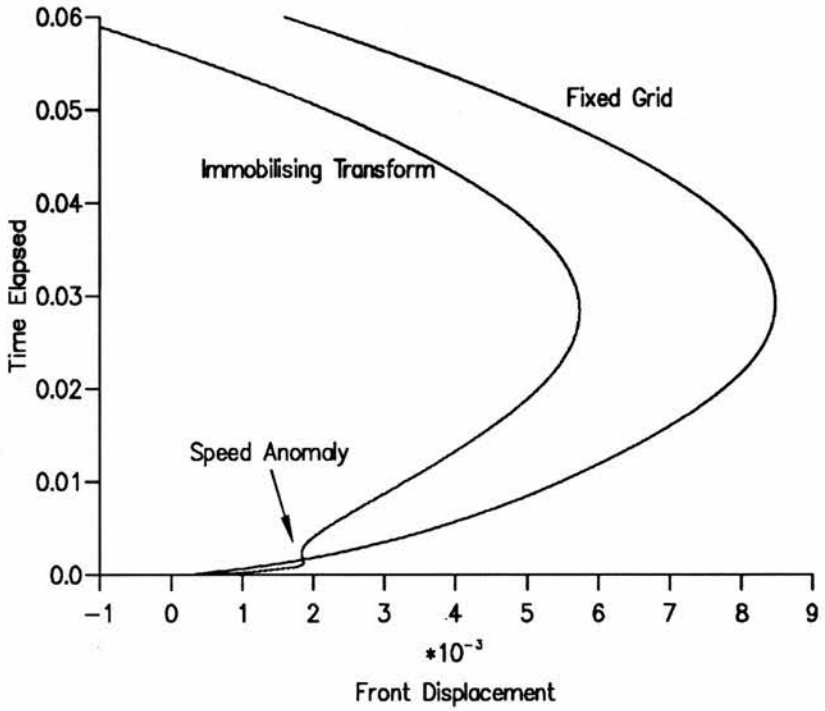

Figure 4.8c: Graphical comparison of the immobilising transformation technique and the fixed grid method for a finite liquid and $X_1 = 1.0$ for parameters 4.7a for small time.

§4.9: Total freezing/thawing scenarios.

Returning to the use of the immobilising technique, by altering the value of X_1 whilst utilising the parameters in equation 4.7a, the initial motion of the boundary remains the same as one of freezing, but the steady state solution can be varied to literally any position across the spatial regions. Figure 4.9a shows results involving initial freezing but varying steady state solutions for small values of X_1.

Table 4.9a shows a representation of the effect of altering the value of the size ratio X_1 on the limiting freeze front position. The algorithm is cycled until such a time that it is deemed that the limit has effectively been reached. For the results here, this time is such

that the front speed is less than 10^{-9}. The last column shows the resultant error in the

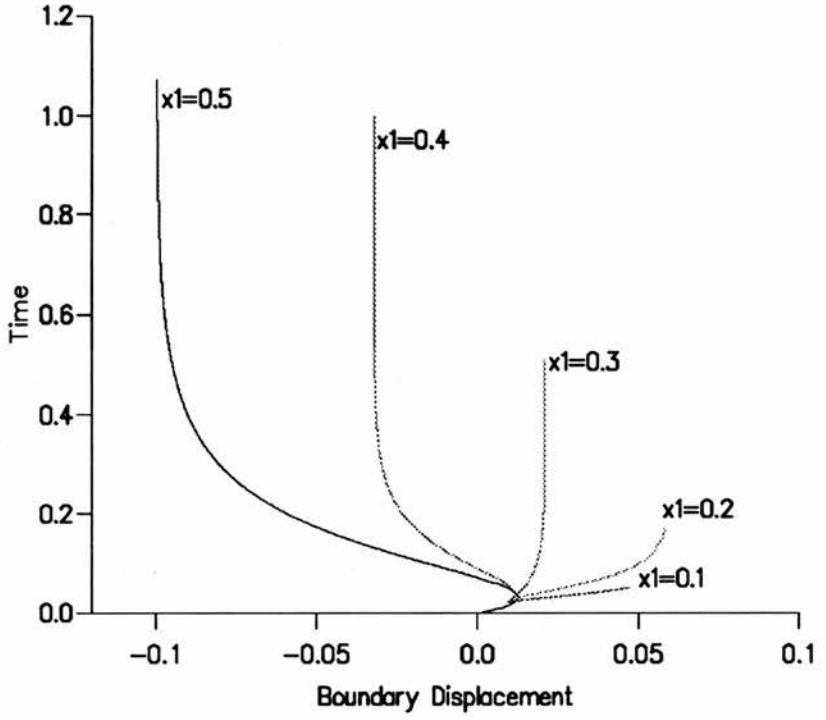

Figure 4.9a: The effect of altering the value X_1 on the steady state solution.

method and this is effectively kept well below the 3% mark. This gives a certain degree of confidence in the accuracy of the technique over a range of values for X_1. The method is most efficient for cases whereby the freeze front does not have a limiting position close to the extremities of either body, since if this occurs the time step at each cycle must be reduced accordingly (equations 4.6a and b). However, even in these instances, this technique is drastically more efficient than the fixed grid technique whilst giving a comparable degree of accuracy. It also should be noted that in altering the value of X_1 the fixed grid method requires a complete restructuring of the grid spacings if the new value of X_1 is not of the same order of magnitude as the old value, thus making the method even less desirable.

X_1	Front Limit		Time Elapsed	% Error In Calc. Limit
	Theoretical	Calculated		
0.1	Total Freezing Of Liquid Bath			
0.2	0.0666667	0.0656157	0.94	1.58
0.3	0.0166667	0.0170181	1.02	2.11
0.4	-0.0333333	-0.0336927	1.65	1.08
0.5	-0.0833333	-0.0813098	3.56	2.43
1.0	-0.333333	-0.339873	12.3	1.96
2.0	-0.833333	-0.859979	36.0	3.20
2.4	Total Thawing Of Cold Body			

Table 4.9a: The effects of altering the value X_1 on the limiting freeze front position.

The final problem remaining is that of what happens as the front approaches either boundary X=-1, X_1 causing a division by zero in the appropriate diffusion equation (Note that the freeze front remains within the body provided that $^1/_9 < X_1 < {}^7/_3$). An example of this is when X_1=3 thus giving a theoretical limiting value of X_0 = -4/3 which is outside the realms of the system, hence corresponding to a solution of complete thawing.

The problem of the division by zero is overcome by examining the relevant temperature profiles in the two regions as the front approaches a boundary. The temperature profile for the case of X_1=3 for time t ≈ 4.9 corresponding to a front position of X_0=-0.83 is shown in Figure 4.9b. It is clear that the temperatures in the vanishing region are so close to the freezing temperature T_f (U=0) as to be negligible. This simplification allows the smaller region to be omitted from considerations and the moving boundary equation is simplified accordingly:

$$\frac{-1}{(X_1 - X_0)} \frac{\partial U}{\partial z_2} \Bigg|_{0+} = \alpha \frac{dX_0}{dt} \qquad \textbf{(4.9a)}$$

removing the potential division by zero as $X_0 \rightarrow 0$

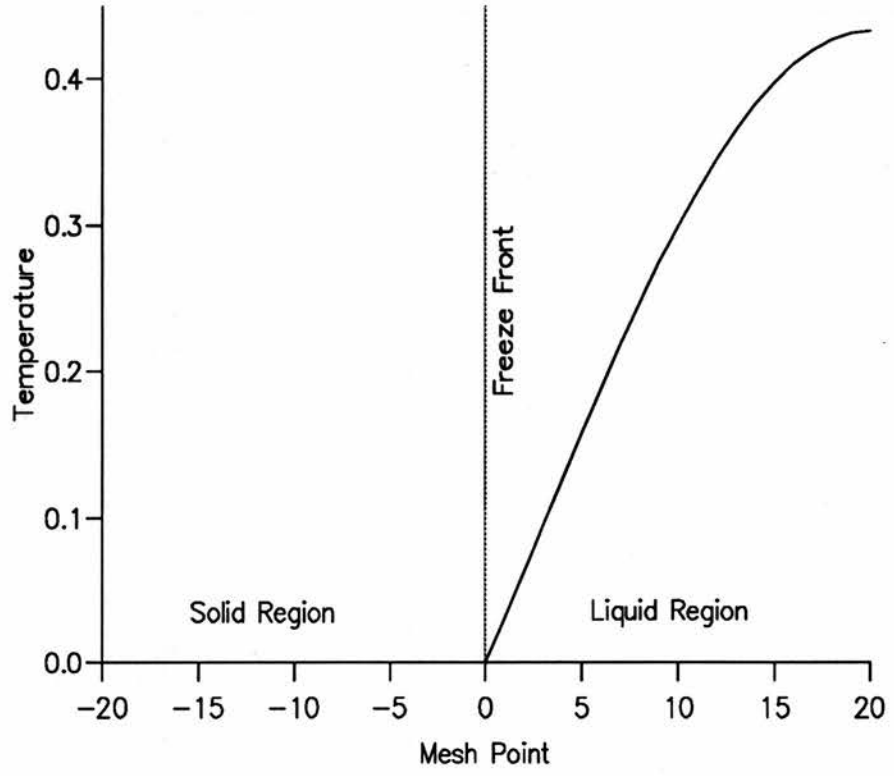

Figure 4.9b: Temperature profile for $X_1 = 3$ after time $t \approx 4.9$ (approaching total thawing)

The transition to the omission of the solid, from its inclusion is made at time such that it is as smooth as possible and the front position progresses until X_0=-1 when the system obviously becomes invalid. After this time any heat profiles remaining flatten to a uniform temperature distribution. Figure 4.9c shows the complete lifetime of the solid including the transition to the omission of the temperature profile in that region.

Situations may also occur which involve the freeze front approaching the boundary $X = X_1$ thus indicating total freezing. For the parameter regime 4.7a this may occur for $X_1 \leq {}^1/_9$. For $X_1 = {}^1/_{20}$, the initial motion of the freeze front is shown in Figure 4.9d. Again by examining the temperature profiles at time $t \approx 0.014$ (Figure 4.9e) it is obvious that in the liquid region is essentially at the freezing temperature and thus may be omitted from further calculations.

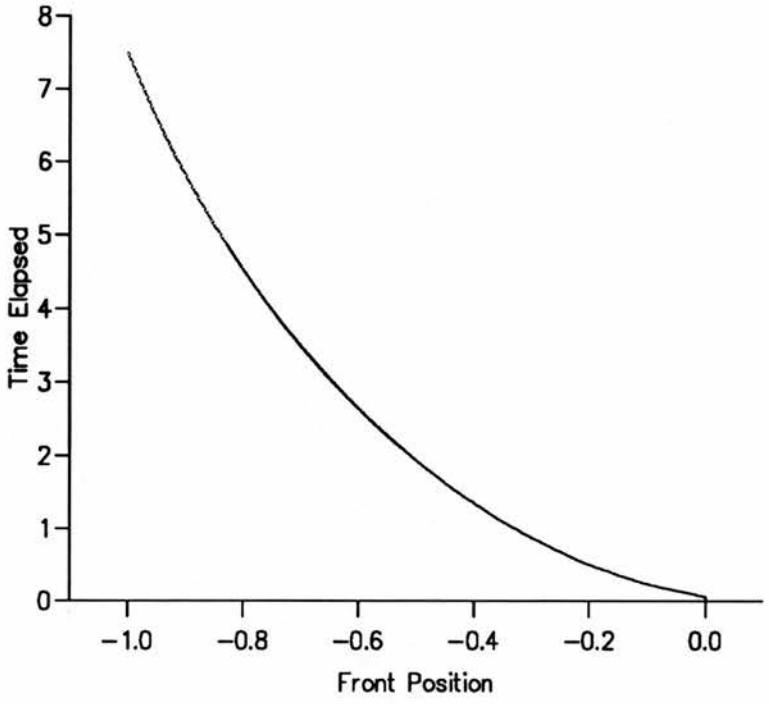

Figure 4.9c: Full lifetime of solidification front for X_1=3, yielding total thawing.

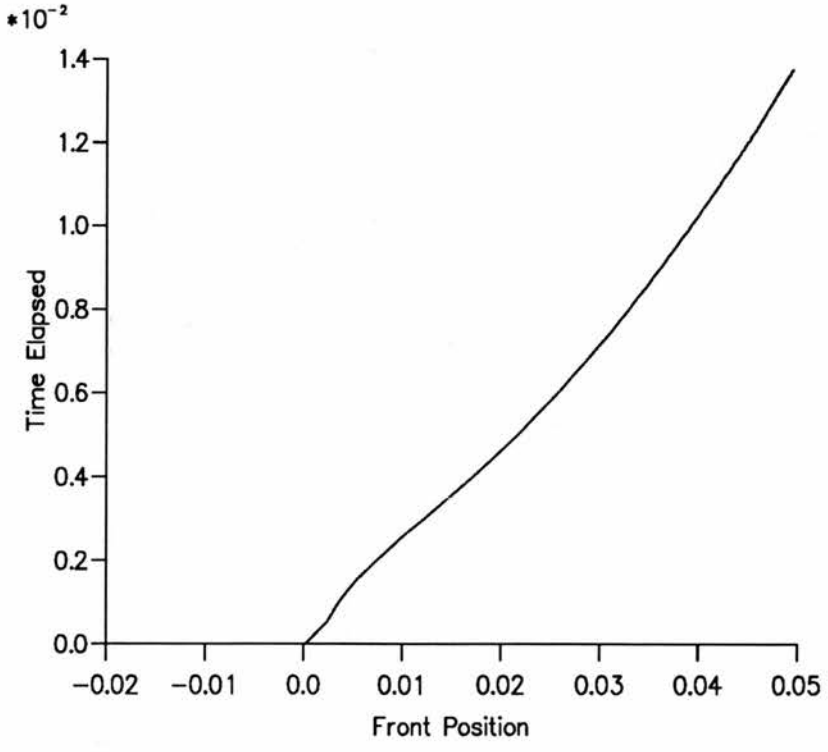

Figure 4.9d: Initial motion of the freeze front for $X_1 = {}^1/_{20}$.

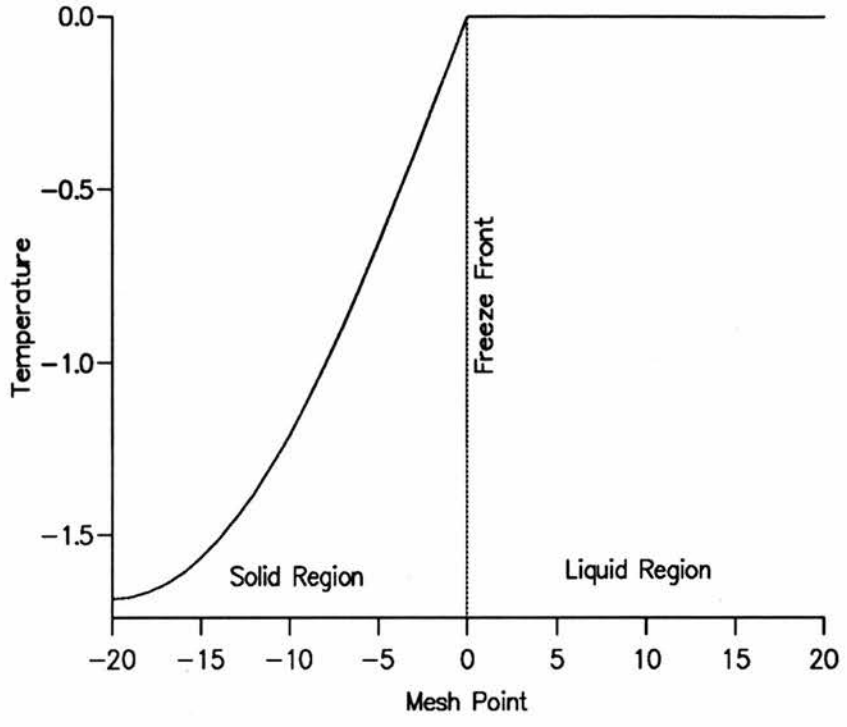

Figure 4.9e: Temperature profile for $X_1 = {}^1/_{20}$ after time $t \approx 0.014$ (approaching total freezing).

The moving boundary equation (4.4i) has to be modified since the spatial derivative of temperature in the liquid bath is now zero:

$$\frac{\beta}{(1 + X_0)} \frac{\partial U}{\partial z_1} \Big|_{0-} = \alpha \frac{dX_0}{dt} \qquad \textbf{(4.9b)}$$

The transition from the original solution to the adapted solution is again made as smooth as possible, by consideration of an optimum changeover time, and the resultant complete freeze front lifetime is shown in Figure 4.9f. The change occurs at $t \approx 0.0138$ by which time the freeze front has progressed to $X_0 \approx 0.0496$.

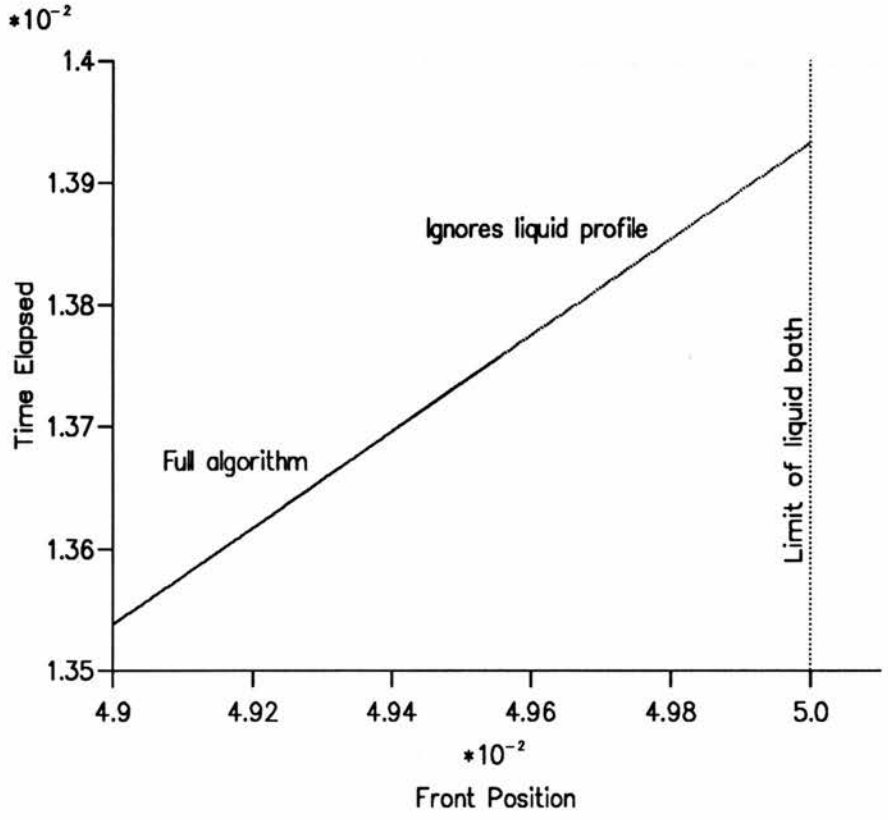

Figure 4.9f: The transition from the full algorithm to that omitting the liquid bath, thus allowing the development of the full history of the freeze front.

The immobilising transformation method developed obviously works well for positive values of λ. However, the method should be validated for the unusual case when $\lambda = 0$ (and possibly for negative values). The simplest regime to give a zero value for λ is to adopt:

$$\eta = 16 \qquad \beta = 2 \qquad \alpha = 2 \qquad U_B = -2 \,. \tag{4.9c}$$

These values alter the limiting front position for $X_1 = 1.0$ to be at:

$$\lim_{t\to\infty} \left[\frac{x_0(t)}{a_1} \right] = -0.375 \,. \tag{4.9d}$$

The starting solution is used up to a time $t_0 = 0.0006$ whereby the transition to the numerical algorithm is the smoothest and the resultant front position history is shown in Figure 4.9g.

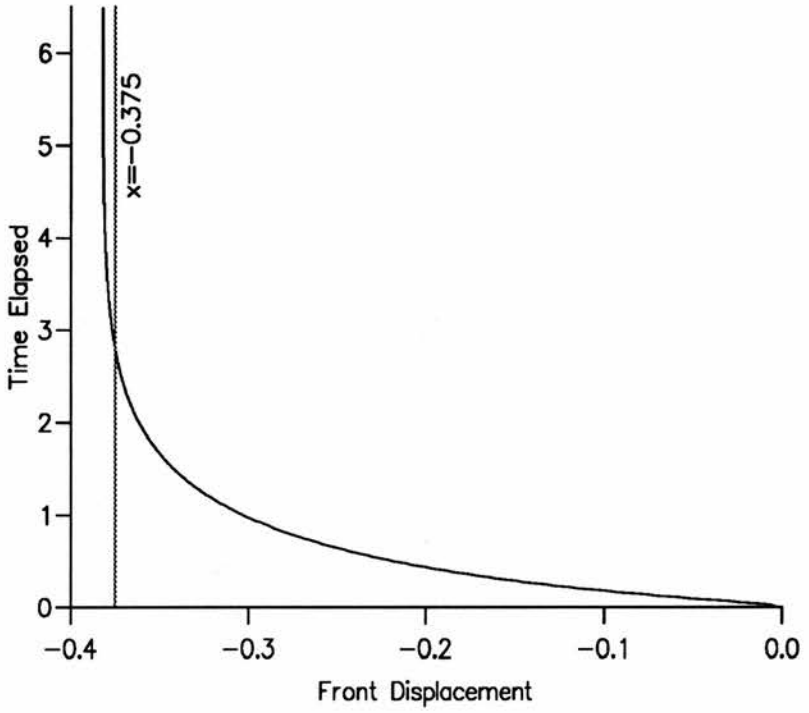

Figure 4.9g: Freeze front history for $X_1 = 1.0$ and an initially motionless boundary.

The results are clearly good although there is a slight discrepancy between the calculated limiting front position and the theoretical one. The numerical algorithm predicts a limiting freeze front position to be at $X_0 = -0.38225855$ compared with the theoretical limit of $X_0 = -0.375$. This corresponds to an error of around 1.9% which is excellent considering that only a small number of grid spacings are used.

The case when $\lambda = 0$ is most unusual since initially heat flow from the liquid goes completely to heating the solid and none is 'lost' as latent heat in any phase change. However, after a small time, the relative sizes of the two regions begins to induce motion of the freeze front. An interesting development of this would be to examine a system by

which the limiting position of the freeze front is also at $X_0 = 0$ but to have the thermal characteristics of the two phases as being different from one another.

§4.10: Finite liquid region summary and conclusions.

The object of this chapter was to solve the problem of immersing a finite cold body in a finite liquid bath and develop a solution involving the moving interface between the two regions and the relevant temperature profiles across them. This was achieved by two methods; firstly an immobilising transformation technique similar to that in §2 and secondly a fixed grid technique similar to that in §3. The numerical solutions for these two methods are progressed from an error function starting solution to either a state of total freezing or thawing or to a steady state intermediate to these two.

The results developed are extremely accurate with respect to the theoretical boundary position of the steady state and also in the match between the two methods. However, fewer problems are likely to occur with a finite liquid region as opposed to an infinite one, purely by the fact that a semi-infinite region provides a tricky hurdle to negotiate. This is upheld by the comparative ease of solution and the accuracy gained for relatively few mesh spacings. Nevertheless, this chapter provides insight into an area which holds great physical interest.

§5. Summary and Conclusions.

In order to solve the classical Stefan problem, there are clearly many techniques available (see §1). The work presented in this thesis is an attempt to track with greater accuracy the rapidly moving freeze front for the one-dimensional case of a cold body immersed in a warm liquid bath. This was done by returning to a more simplistic approach using a front fixing or immobilising technique. The results gained by this method were then checked against a front tracking method proposed by Crank [13] and similarly a perturbation expansion proposed by Tadjbakhsh and Liniger [2].

Obviously other methods were tried and discarded before and during the development of the immobilising transformation technique. For instance, the zone of influence in the liquid bath may itself be considered as a region, as opposed to the bath as a whole. This gives rise to an 'influence front' (X_1 say) moving through the bath. Behind X_1 the liquid temperature has deviated from its ambient sufficiently to be considered to have been influenced and ahead of it, the temperatures remain unchanged. At first sight this method may seem to be a valid one, however analysis of this type failed due to the extremely rapid progression of X_1. Subsequent discretisation of the liquid region failed to place enough mesh points in the active region, close to the freeze front.

Another method tried was one of matching a piecewise polynomial across the three regions. However, even utilising fractional powers, it was found to be impossible to generate coefficients such that all boundary conditions were satisfied. This technique would be deemed to be of low accuracy even on succeeding to generate coefficients and was attempted merely as a means of checking the results derived by the other methods.

The results generated by the immobilising transformation technique appear to be accurate. There is a reasonable match between these results and those generated by the fixed grid technique and the perturbation expansion. The additional special cases of continuous freezing and limiting frost depth also give excellent results when solved using the immobilising transformation technique. Furthermore, the immobilising techniques used in §4 give very good results over a broad range of thermal parameters.

The main difficulties encountered were in the investigation of the fixed grid technique undertaken in §3. Here the obvious problem is one of the mesh configuration and consequently the computational efficiency of the method. Difficulties arise due to the bound placed on the minimum start time invoked by the grid configuration itself. An optimum start time whereby the transition from the starting solution to the numerical algorithm is as smooth as possible cannot be utilised due to this bound. Hence the method of solution is deemed inaccurate. As previously explained in §3, the only way to reduce the start time further is to use a finer mesh. This vastly reduces the efficiency of the method due to the need to reduce the time increment used and also the subsequent increase in the number of calculations per time step. The method already imposes a huge burden on CPU time and thus any increase in this is considered impractical if not impossible. Hence further investigation of this method is not attempted.

Speed of solution is possibly the only factor requiring attention. The immobilising transformation technique is drastically more efficient than the fixed grid technique, however certain other methods may give a rough solution in a fraction of the time. This should not be considered as a great downfall of the method, since it can track rapid freeze front motion where other techniques fail. It must be possible to increase the efficiency of the process by some sort of fine tuning of the numerical procedures, but this has not been attempted. Another possibility is the use of implicit finite difference approximations instead of explicit ones, however for a large number of mesh points, this change may not necessarily improve efficiency.

It would be desirable if this method could be easily applied to the multi-dimensional problem, but it far from obvious how this could be achieved. Nevertheless, transition to cylindrical or spherical geometries is very straightforward in terms of the transformation used. Investigation of the cylindrical geometry problem shows a much more rapid progression of the freeze front. This causes difficulties in the transition from a possible starting solution (which in itself is difficult to develop - the one-dimensional starting solution in some circumstances may be inappropriate) to the numerical procedure. The technique may also be modified to take into account a possible mushy region at the phase change boundary. This ability is often seen as the strength of such methods as the

enthalpy technique. The plasticised phase would itself have to be considered as an individual region, so producing a further moving boundary condition. Only one possible set of boundary conditions is examined here, so continuation of this work may be to modify this set for different physical conditions.

In summary, the immobilising transformation technique developed can be considered as a definite improvement on the methods available for solution of the classical Stefan problem. Its use is clear due to the vast array of diverse applications available in this area of study.

Appendix

The following programs are examples of those used for the generation of results given in the previous chapters. All are written in Sun Pascal v2.1.

```
{       Calculates the parameter lambda appearing in the starting solution for the      }
{       full three-regioned classical Stefan problem by use of a Newton-Raphson         }
{       method                                                                          }

program lamda;

   const
        pi=3.141592654;
        kap1=8.0;
        kap2=2.0;
        beta=2.0;
        phi=8.0;
        ub=-1.25;
        alpha=10.0;

   var
        lam,u0,kap3:double;
        count,count2:integer;

    function erf(y:double):double;
{       Since this version of Pascal contains a basic few mathematical functions,       }
{       the error function module used from an external C file                          }
        external c;

    function f(x:double):double;
{       Returns the value of f(lambda) for the latest value of lambda                   }
        var
        f1,f2,f3:double;
        begin
        f1:=alpha*x*sqrt(kap2*pi)/2;
        f2:=beta*ub*phi*exp(-x*x/4)/(erf(x/2)*sqrt(kap2)*phi+sqrt(kap1));
        f3:=exp(-x*x*kap2/4)/(1-erf(x*sqrt(kap2)/2));
        f:=f1+f2+f3;
        end;

    function fdash(x:double):double;
{       Returns the derivative of f for the latest value of lambda                      }
        var
        f4,f5,f6:double;
        begin
        f4:=alpha*sqrt(kap2*pi)/2;
        f5:=x*(phi*erf(x/2)*sqrt(kap2)+sqrt(kap1))/2+phi*sqrt(kap2)*exp(-x*x/4)
/sqrt(pi);
        f5:=-f5*beta*phi*ub*exp(-x*x/4)/sqr(phi*erf(x/2)*sqrt(kap2)+sqrt(kap1));
        f6:=exp(-x*x*kap2/4)*sqrt(kap2/pi)-x*kap2*(1-erf(x*sqrt(kap2)/2))/2;
        f6:=f6*exp(-x*x*kap2/4)/sqr(1-erf(x*sqrt(kap2)/2));
        fdash:=f4+f5+f6;
        end;

{       Main program...runs the Newton-Raphson algorithm and then calculates            }
{       the starting solution temp at the immersed body / solidified layer boundary     }
begin
        lam:=0.1;
        for count:=1 to 500 do
        lam:=lam-f(lam)/fdash(lam);
```

```
        writeln('Final value for lamda=',lam);
        writeln('Final value for f(lam)=',f(lam));
        kap3:=kap2/kap1;
        u0:=phi*ub*erf(lam/2)*sqrt(kap3)/(phi*erf(lam/2)*sqrt(kap3)+1);
        writeln('u0=',u0);
end.
```

```
{       Generates the start data for the program which runs the main algorithm      }
{       for the full three regioned one-dimensional classical Stefan problem        }

program startsoln;

    const
        startfile='3rf41.s';            { file to generated }
        kap1=8.0;
        kap2=6.0;
        beta=2.0;
        phi=8.0;
        ub=-0.5;
        alpha=10.0;
        u0=-0.0908210281044598;
        lambda=0.0567994553709387;
        tstart=0.001;
        h=0.05;
        p=0;

    var
        u:array[-20..40,0..1] of double;
        x0,x0dot,x1,x1dot:array[0..1] of double;
        ts,x:double;
        c1,q,c2:integer;
        f:file of double;
        path:string;

    function erf(y:double):double;
{       Since this version of Pascal contains a basic few mathematical functions,   }
{       the error function module used from an external C file                      }
        external c;

    function small1(y,time:double):double;
{       Generates the small time temperature profile in the immersed cold body      }
        begin
        small1:=(u0-ub)*erf(y/(2*rtkap1*time))+u0;
        end;

    function small2(y,time:double):double;
{       Generates the small time temperature profile in the solidified layer        }
        begin
        small2:=u0*(1-erf(y/(2*sqrt(kap2*time)))/erf(lamda/2));
        end;

    function small3(y,time:double):double;
{       Generates the small time temperature profile in the warm liquid bath        }
        var
        term1:double;
        begin
        term1:=erf(y/(sqrt(time)*2))-erf(lamda*rtkap2/2);
        small3:=term1/(1-erf(lamda*rtkap2/2));
        end;

{       Main program...calculates freeze front position and speed, and temperature  }
{       profiles in the cold body, solidified layer & liquid region respectively. Then }
{       writes data to the file to be used by the algorithm                         }
begin
        rewrite(f,startfile);
        x0[p+1]:=lamda*sqrt(kap2*tstart);
        x0dot[p+1]:=0.5*lambda*sqrt(kap2/tstart);
```

```
        for c1:=0 to 20 do              {immersed body}
        begin
        x:=-1+c1*h;
        u[c1-20,p+1]:=small1(x,tstart);
        end;
        for c1:=1 to 20 do              {solidified layer}
        begin
        x:=x0[p+1]*c1*h;
        u[c1,p+1]:=small2(x,tstart);
        end;
        for c1:=0 to 19 do              {liquid bath}
        begin
        x:=x0[p+1]/(1-c1*h);
        u[c1+20,p+1]:=small3(x,tstart);
        end;
        u[40,p+1]:=1.0;
        u[20,p+1]:=0.0;

        write(f,tstart);
        write(f,x0[p+1]);
        write(f,x0dot[p+1]);
        for c1:=-20 to 40 do
        write(f,u[c1,p+1]);
end.
```

```
{       Full three regioned 1-D Stefan problem using one temp variable u and 60      }
{       space steps, for a finite body immersed in a liquid of a differing material  }
{                              Part One                                              }

program infin_liq_pt1;

    const
         startfile='/home/maths/idw/3RF/3rf.01/3rf01.start';
        outputfile='3rf01.out';
        kap1=8;
        kap2=2.0;
        beta=2.0;
        phi=8.0;
        ub=-0.5;
        alpha=10.0;
        tol=0.1;
        fail=10.0;
        finish=8000000;
        h=0.05;
        p=0;

    var
        u:array[-20..40,0..1] of double;
        x0,x0dot:array[0..1] of double;
        tnow,x,term1,term2,term3,twominusz,ts:double;
        c1,q,tsteps,c2:integer;
        f,g:text;
        g2,g1:file of double;
        failsign,path:string;

    function pow(x1,y1:double):double;
{       A failing of the language Pascal its its inability to raise a number x1 to the  }
{       power y1 and hence the operation has to be completed by an external C file      }
        external c;

    function dudz(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at     }
{       the point q1                                                                    }
        begin
        dudz:=(u[q1+1,p]-u[q1-1,p])/(2*h);
        end;

    function d2udz2(q1:integer):double;
{       Calculates the finite difference approximation of the spatial double            }
{       derivative at the point q1                                                      }
        begin
        d2udz2:=(u[q1+1,p]-2*u[q1,p]+u[q1-1,p])/sqr(h);
        end;

    function duplus(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at     }
{       the point q1 from above                                                         }
        begin
        duplus:=(-3*u[q1,p]+4*u[q1+1,p]-u[q1+2,p])/(2*h);
        end;

    function duminus(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at     }
{       the point q1 from below                                                         }
        begin
```

```
        duminus:=(3*u[q1,p]-4*u[q1-1,p]+u[q1-2,p])/(2*h);
        end;

    procedure resetvars;
{       Sets the latest set of information to be the old set...cycles one time step      }
        var
        ct:integer;
        begin
        u[20,p+1]:=0.0;
        u[40,p+1]:=1.0;
        for ct:=-20 to 40 do
        u[ct,p]:=u[ct,p+1];
        x0[p]:=x0[p+1];
        x0dot[p]:=x0dot[p+1];
        end;

    procedure printout;
{       Writes current freeze front information to the output file                      }
        begin
        writeln(f,tnow:11:8,' ',x0[p+1]:11:8);
        end;

    procedure solveeqns(k:double);
{       Solves the approximated equations at each time step of the algorithm in        }
{       the order...update cold body & solid temps, solid/liquid boundary, liquid      }
{       temps, freeze front position x0, lower solid boundary, upper liquid boundary.  }
        begin
        resetvars;
        tnow:=tnow+ts;
        for q:=-20 to -1 do             {cold body}
        u[q,p+1]:=u[q,p]+k*kap1*d2udz2(q);
        for q:=1 to 19 do               {solid}
        u[q,p+1]:=u[q,p]+kap2*k*d2udz2(q)/sqr(x0[p])+q*h*k*x0dot[p]*dudz(q)/x0[p];
        x0dot[p+1]:=(beta*duminus(20)-duplus(20))/(x0[p]*alpha);
        for q:=21 to 39 do              {liquid}
        begin
        twominusz:=2-q*h;
        term1:=x0dot[p]*twominusz*dudz(q)/x0[p];
        term2:=2*pow(twominusz,3)*dudz(q)/sqr(x0[p]);
        term3:=pow(twominusz,4)*d2udz2(q)/sqr(x0[p]);
        u[q,p+1]:=u[q,p]+k*(term1-term2+term3);
        end;
        x0[p+1]:=x0[p]+k*x0dot[p+1];
        term1:=4*u[1,p]-u[2,p];
        term2:=x0[p]*phi*(4*u[-1,p]-u[-2,p]);
        u[0,p+1]:=(term1+term2)/(3+3*x0[p]*phi);
        u[-20,p+1]:=(4*u[-19,p]-u[-18,p])/3;
        end;

    procedure findtimestep;
{       Calculates an optimum time step at each cycle of the algorithm                 }
        var
        counter:integer;
        begin
        ts:=tol*sqr(h*x0[p+1])/2;
        counter:=0;
        repeat
        ts:=ts*10;
        counter:=counter+1;
        until  ts>=100.0;
```

```
        ts:=trunc(ts)*pow(10,-counter);
        end;

    procedure stopprog;
{       Terminates program if algorithm is diverging and generates a warning file      }
        var
        count:integer;
        begin
        c2:=finish;
        printout;
        writeln(f,'failed');
        for count:=-20 to 40 do
        writeln(f,'u[',count:2,',p]=',u[count,p]);
        failsign:=outputfile+'FAILED';
        rewrite(g,failsign);
        writeln(g,outputfile,' has failed');
        close(g);
        end;

    procedure partprint;
{       Generates data files every 1/100th of the total number of time steps           }
        var
        c1,dummy,d1,d2:integer;
        begin
        dummy:=round(c2*100/finish);
        d1:=dummy div 10;
        d2:=dummy mod 10;
        path:='/home/maths/idw/3RF/3rf.01/'+outputfile+chr(d1+48)+chr(d2+48);
        open(g2,path,'new');
        write(g2,tnow);
        write(g2,x0[p+1]);
        write(g2,x0dot[p+1]);
        for c1:=-20 to 40 do
        write(g2,u[c1,p+1]);
        close(g2);
        end;

{       Main program...loads in start data and controls the running of the algorithm    }
begin
        open(g1,startfile,'old');
        reset(g1);
        read(g1,tnow);
        read(g1,x0[p+1]);
        read(g1,x0dot[p+1]);
        for c1:=-20 to 40 do
        read(g1,u[c1,p+1]);
        close(g1);
        path:='/home/maths/idw/3RF/3rf.01/'+outputfile;
        open(f,path,'new');
        resetvars;
        findtimestep;
        printout;
        partprint;

        c2:=0;
        repeat
        findtimestep;
        solveeqns(ts);
        if c2 mod round(finish/100)=0 then partprint;
        if c2 mod round(finish/1000)=0  then printout;
```

```
        c2:=c2+1;
        if abs(x0[p])>fail then stopprog;
        until c2>=finish;

        writeln(outputfile,' done');
        close(f);
end.
```

```
{       Calculates the temp profiles in the three regions for the modified algorithm    }
{       used after the stationary point for the full three-regioned classical Stefan    }
{       problem. These profiles are found by matching a piecewise set of splines        }
{       across the old temp profile and reading off the new temps                       }

program spline;

   const
        n=25;
        folderin='3rf.37/partfiles/3rf37.out';
        folderout='3rf.37/3rf3.s.';
        start=1;
        finish=73;

   var
        a,b,func:array[-3..n] of double;
        m:array[-2..n-2,-2..n-2] of double;
        cold:array[1..40] of double;
        i,j,k,d1,d2,dummy:integer;
        x,y,z,x0,x0dot,duplus,duminus,tnow:double;
        f,g:file of double;
        f1,g1:text;
        path:array[0..255] of char;

    function pow(x2,y2:double):double;
{       A failing of the language Pascal its its inability to raise a number x1 to the  }
{       power y1 and hence the operation has to be completed by an external C file      }
        external c;

    function C(y1:double):double;
{       Calculates the value of the function C in the cubic splines approximation §2    }
        var term17:double;
        begin
        term17:=2/3-sqr(y1)*(2-abs(y1))/2;
        if abs(y1)>1 then term17:=pow(2-abs(y1),3)/6;
        if abs(y1)>2 then term17:=0;
        C:=term17;
        end;

{       Main program...loads in data dump from previous algorithm, creates matrix      }
{       M in splines theory §2 and solves for coefficients using Gaussian elimination. }
{       Writes new temp profile to starting file for modified algorithm                 }
begin
        for dummy:=start to finish do
        begin
        d1:=dummy div 10;
        d2:=dummy mod 10;
        path:=folderout+chr(d1+48)+chr(d2+48);
        open(g,path,'new');
        path:=folderin+chr(d1+48)+chr(d2+48);
        open(f,path,'old');
        reset(f);
        read(f,tnow);
        read(f,x0);
        read(f,x0dot);
        for k:=1 to 40 do
        begin
        read(f,x);
        cold[k]:=x;
        end;
```

```
for k:=0 to 20 do
begin
read(f,x);
func[k]:=x;
end;
for k:=-3 to -1 do
func[k]:=k*func[1];
for k:=21 to n do
func[k]:=func[20];

for k:=-1 to n-2 do
b[k]:=6*func[k];
duplus:=(-3*func[0]+4*func[1]-func[2])/2;;
duminus:=0;
b[-2]:=6*func[0]+2*duplus;
b[n-1]:=6*func[n]-2*duminus;

for i:=-2 to n-2 do
        for j:=-2 to n-2 do
        begin
        if i=j then m[i,j]:=4;
        if i=j-1 then m[i,j]:=1;
        if i=j+1 then m[i,j]:=1;
        m[-2,-1]:=2.0;
        m[n-2,n-3]:=2.0;
        end;

for i:=-2 to n-3 do
begin
      b[i]:=b[i]/m[i,i];
       b[i+1]:=b[i+1]-b[i]*m[i+1,i];
    for j:=-2 to n-2 do
        begin
        m[i,n-4-j]:=m[i,n-4-j]/m[i,i];
        m[i+1,n-4-j]:=m[i+1,n-4-j]-m[i,n-4-j]*m[i+1,i];
        end;
end;
b[n-2]:=b[n-2]/m[n-2,n-2];
m[n-2,n-2]:=m[n-2,n-2]/m[n-2,n-2];

for k:=-3 to n do
a[k]:=0.0;
a[n-2]:=b[n-2];
for i:=-2 to n-3 do
begin
a[n-5-i]:=b[n-5-i];
for j:=n-5-i to n-5 do
a[n-5-i]:=a[n-5-i]-m[n-5-i,j+1]*a[j+1];
end;

a[-3]:=a[-1]-duplus*2;
a[n-1]:=a[n-3]+2*duminus;

write(g,tnow);
write(g,x0);
write(g,x0dot);
for k:=1 to 40 do              {immersed body/ frost layer}
write(g,cold[k]);

for k:=0 to 20 do              {liquid bath}
```

```
        begin
        x:=k*(10-x0)/20+x0;
        z:=2-x0/x;
        z:=(z-1)*20;
        y:=0;
        for j:=-3 to n-1 do
        y:=y+a[j]*C(z-j);
        write(g,y);
        end;

        close(g);
        close(f);
        end;
end.
```

```
{       Full three regioned 1-D Stefan problem using one temp variable u and 60      }
{       space steps, for a finite body immersed in a liquid of a differing material  }
{                              Part Two                                              }

program inf_liq_pt2;

    const
         startfile='3rf3.s.';
        outputfile='3rf3.out01';
        kap1=8.0;
        kap2=2.0;
        beta=2.0;
        phi=8.0;
        ub=-0.5;
        alpha=10.0;
        tol=0.8;
        fail=10.0;
        ts=0.0001;
        finish=10000;
        h=0.05;
        p=0;
        loopst=20;
        loopfin=99;

    var
        u:array[-20..40,0..1] of double;
        x0,x0dot:array[0..1] of double;
        testvar:array[loopst..loopfin,0..1] of double;
        tnow,x,term1,term2,term3,twominusz,bestdiff:double;
        c1,q,tsteps,c2,loopvar,bestone,d1,d2:integer;
        f,g:text;
        g2,g1:file of double;
        failsign,path:string;

    function dudz(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at  }
{       the point q1                                                                  }
        begin
        dudz:=(u[q1+1,p]-u[q1-1,p])/(2*h);
        end;

    function d2udz2(q1:integer):double;
{       Calculates the finite difference approximation of the spatial double         }
{       derivative at the point q1                                                   }
        begin
        d2udz2:=(u[q1+1,p]-2*u[q1,p]+u[q1-1,p])/sqr(h);
        end;

    function duplus(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at  }
{       the point q1 from above                                                      }
        begin
        duplus:=(-3*u[q1,p]+4*u[q1+1,p]-u[q1+2,p])/(2*h);
        end;

    procedure resetvars;
{       Sets the latest set of information to be the old set...cycles one time step  }
        var
        ct:integer;
        begin
```

```
        u[20,p+1]:=0.0;
        u[40,p+1]:=1.0;
        for ct:=20 to 40 do
        u[ct,p]:=u[ct,p+1];
        x0[p]:=x0[p+1];
        x0dot[p]:=x0dot[p+1];
        end;

    procedure printout;
{       Writes current freeze front information to the output file                }
        begin
        writeln(f,tnow:11:8,' ',x0[p+1]:11:8);
        end;

  procedure solveeqns1(k:double);
{       Solves the approximated equations at each time step of the algorithm in   }
{       the order...update solid/liquid boundary, liquid temps, freeze front position x0 }
        begin
        resetvars;
        tnow:=tnow+ts;
        x0dot[p+1]:=-duplus(20)/((10-x0[p])*alpha);
        for q:=21 to 39 do              {liquid}
        begin
        term1:=d2udz2(q)/sqr(10-x0[p])+20*x0dot[p]*dudz(q)/sqr(10-x0[p]);
        u[q,p+1]:=u[q,p]+k*term1;
        end;
        x0[p+1]:=x0[p]+k*x0dot[p+1];
        end;

    procedure stopprog;
{       Terminates program if algorithm is diverging and generates a warning file }
        var
        count:integer;
        begin
        c2:=finish;
        printout;
        writeln(f,'failed');
        for count:=-20 to 40 do
        writeln(f,'u[',count:2,',p]=',u[count,p]);
        end;

{       Main program...loads in start data, runs algorithm for 10 time steps and finds }
{       an optimum changeover time. The best data is then reloaded, the full algorithm }
{       is run from that set of data                                              }
begin
        for loopvar:=loopst to loopfin do
        begin
        d1:=loopvar div 10;
        d2:=loopvar mod 10;
        path:=startfile+chr(d1+48)+chr(d2+48);
        open(g1,path,'old');
        reset(g1);
        read(g1,tnow);
        read(g1,x0[p+1]);
        read(g1,x0dot[p+1]);
        for c1:=-20 to 40 do
        read(g1,u[c1,p+1]);
        close(g1);
        resetvars;
        solveeqns1(ts);
```

```
        testvar[loopvar,0]:=x0dot[p+1];
        for c2:=0 to 10 do
        solveeqns1(ts);
        testvar[loopvar,1]:=x0dot[p+1];
        bestdiff:=abs(testvar[loopvar,1]-testvar[loopvar,0]);
        writeln(loopvar,'       ',bestdiff);
        end;
        bestone:=loopst;
        bestdiff:=abs(testvar[loopst,1]-testvar[loopst,0]);
        for loopvar:=loopst to loopfin do
        begin
        if abs(testvar[loopvar,1]-testvar[loopvar,0])<bestdiff then
        begin
        bestone:=loopvar;
        bestdiff:=abs(testvar[loopvar,1]-testvar[loopvar,0]);
        end;
        end;

        d1:=bestone div 10;
        d2:=bestone mod 10;
        path:=startfile+chr(d1+48)+chr(d2+48);
        writeln('Using ',path);
        open(g1,path,'old');
        reset(g1);
        read(g1,tnow);
        read(g1,x0[p+1]);
        read(g1,x0dot[p+1]);
        for c1:=-20 to 40 do
        read(g1,u[c1,p+1]);
        close(g1);
        rewrite(f,outputfile);
        resetvars;
        printout;
        c2:=0;
        repeat
        solveeqns1(ts);
        if (round(c2*1000/finish)=c2*1000/finish) then printout;
        c2:=c2+1;
        if abs(x0[p])>fail then stopprog;
        if x0[p]<0.0 then stopprog;
        until c2>=finish;
        writeln(outputfile,' done');
        close(f);
end.
```

```
{       Solution of the full three-regioned classical Stefan problem using a front      }
{       tracking method proposed by Crank. Algorithm uses a modified configuration      }
{       consisting of a dual mesh system in the frost layer / liquid bath               }

program fixing_dual;

    const
        startfile='CRANK/crank3.0.0005';
        outputfile='crank0005.out';
        kap1=7.705035971;
        kap2=2.035971223;
        beta=2.037735849;
        phi=8.351851852;
        ub=-0.488372093;
        alpha=9.784020316;
        lambda=0.06118391;
        fail=10.0;
        u0=-0.098779510438656;
        finish=1000;
        h1=0.05;
        h2=0.001;
        h3=0.2;
        p=0;
        k=0.0000001;

    var
        u:array[-20..1100,0..1] of double;
        x0,x0dot,w:array[0..1] of double;
        tnow,x,xdot,term1,term2,term3,term4:double;
        c1,q,c2,r:integer;
        f,g:text;
        g2,g1:file of double;
        failsign,path:string;

     function d2udz2(q1:integer;h:double):double;
{       Calculates the finite difference approximation of the spatial double            }
{       derivative at the point q1                                                      }
        begin
        d2udz2:=(u[q1+1,p]-2*u[q1,p]+u[q1-1,p])/sqr(h);
        end;

     procedure resetvars;
{       Sets the latest set of information to be the old set...cycles one time step     }
{       Also updates the value of wp if the freeze front has changed mesh spacings      }
        begin
        for q:=-21 to 1100 do
        u[q,p]:=u[q,p+1];
        x0[p]:=x0[p+1];
        w[p]:=w[p+1];
        if w[p]>=1.5 then
          begin
                u[r,p]:=(1-w[p])*u[r-2,p]/(1+w[p])+2*(w[p]-1)*u[r-1,p]/w[p];
            r:=r+1;
             w[p]:=w[p]-1.0;
          end;
        if w[p]<0.5 then
          begin
                u[r,p]:=2*(1-w[p])*u[r+1,p]/(2-w[p])-(1-w[p])*u[r+2,p]/(3-w[p]);
            r:=r-1;
            w[p]:=w[p]+1;
```

```
        end;
      end;

  procedure printout;
{     Writes current freeze front information to the output file                }
      begin
      writeln(f,tnow:11:8,' ',x0dot[p+1]:11:8);
      end;

  procedure solveeqns;
{     Solves the approximated equations at each time step of the algorithm in   }
{     the order...update cold body, cold/solid boundary, lower cold boundary,   }
{     solid temps, freeze front temps, freeze front position, liquid temps, upper }
{     liquid boundary.                                                          }
      begin
      resetvars;
      tnow:=tnow+k;
      for q:=-20 to -1 do              {immersed cold body}
      u[q,p+1]:=u[q,p]+k*kap1*d2udz2(q,h1);
      u[0,p+1]:=(50*(4*u[1,p]-u[2,p])+phi*(4*u[-1,p]-u[-2,p]))/(150+3*phi);
      u[-20,p+1]:=(4*u[-19,p]-u[-18,p])/3;
      for q:=1 to r-2 do               {solidified layer}
      u[q,p+1]:=u[q,p]+kap2*k*d2udz2(q,h2);

      u[r-1,p+1]:=u[r-1,p]+2*kap2*k*(u[r-2,p]/(w[p]+1)-u[r-1,p]/w[p])/sqr(h2);
      u[r+1,p+1]:=u[r+1,p]+2*k*(-u[r+1,p]/(2-w[p])+u[r+2,p]/(3-w[p]))/sqr(h2);

      term1:=(w[p]-3)*u[r+1,p]/(w[p]-2)+(w[p]-2)*u[r+2,p]/(3-w[p]);
      term2:=w[p]*u[r-2,p]/(w[p]+1)-(w[p]+1)*u[r-1,p]/w[p];
      w[p+1]:=w[p]+k*(beta*term2-term1)/(alpha*sqr(h2));

      x0dot[p+1]:=h2*(w[p+1]-w[p])/k;
      x0[p+1]:=x0[p]+k*x0dot[p+1];

      for q:=r+2 to 999 do             {liquid fine mesh}
      u[q,p+1]:=u[q,p]+k*d2udz2(q,h2);

      for q:=1001 to 1099 do           {liquid coarse mesh}
      u[q,p+1]:=u[q,p]+k*d2udz2(q,h3);

      u[1000,p]:=(200*u[999,p]+u[1001,p])/201;
      u[1100,p]:=1.0;
      end;

  procedure stopprog;
{     Terminates program if algorithm is diverging and generates a warning file }
      var
      count:integer;
      begin
      c2:=finish;
      printout;
      writeln(f,'failed');
      for count:=-20 to 1100 do
      writeln(f,'u[',count:2,',p]=',u[count,p]);
      failsign:=outputfile+'FAILED';
      rewrite(g,failsign);
      writeln(g,outputfile,' has failed');
      close(g);
      end;
```

```
    procedure partprint;
{       Generates data files every 1/100th of the total number of time steps        }
        var
        c1,dummy,d1,d2:integer;
        begin
        dummy:=round(c2*10/finish);
        d1:=dummy div 10;
        d2:=dummy mod 10;
        path:='/home/maths/idw/CRANK/'+outputfile+chr(d1+48)+chr(d2+48);
        open(g2,path,'new');
        write(g2,tnow);
        write(g2,x0[p+1]);
        write(g2,x0dot[p+1]);
        for c1:=-20 to 1100 do
        write(g2,u[c1,p+1]);
        close(g2);
        end;

{       Main program...loads in start data and controls the running of the algorithm  }
begin
        path:='/home/maths/idw/'+startfile;
        open(g1,path,'old');
        reset(g1);
        read(g1,tnow);
        read(g1,x0[p+1]);
        read(g1,x0dot[p+1]);
        for c1:=-20 to 1100 do
        read(g1,u[c1,p+1]);
        close(g1);
        rewrite(f,outputfile);
        r:=trunc(x0[p+1]*1000);
        w[p+1]:=1+x0[p+1]*1000-r;
        if w[p+1]>=1.5 then
        begin
        w[p+1]:=w[p+1]-1;
        r:=r+1;
        end;
        if w[p+1]<0.5 then
        begin
        w[p+1]:=w[p+1]+1;
        r:=r-1;
        end;
        for c1:=1 to 99 do
        begin
        xdot:=lambda*0.5/sqrt(tnow*c1/100);
        writeln(f,tnow*c1/100:11:8,'        ',xdot:11:8);
        end;
        resetvars;
        printout;
        partprint;

        c2:=0;
        repeat
        solveeqns;
        if c2 mod round(finish/100)=0  then printout;
        if c2 mod round(finish/10)=0  then partprint;
        c2:=c2+1;
        if abs(x0[p])>fail then stopprog;
        if x0[p]<=0 then stopprog;
        if abs(x0dot[p+1])>1000 then stopprog;
```

```
        until c2>=finish;
        partprint;
        writeln(outputfile,' done');
        close(f);
end.
```

```
{       Special case of the three regioned 1-D Stefan problem using one temp       }
{       variable u and 40 space steps for a finite body immersed in a finite liquid  }
{       of the same material.                                                        }

program finite_liquid;

    const
         startfile='/home/maths/idw/1Dfin/x10.2.s';
        outputfile='1Dfin0.2.out';
        eta=12.0;
        beta=2.0;
        ub=-2.0;
        x1=0.2;
        lambda=0.0611839;
        alpha=2.0;
        tol=0.5;
        fail=10.0;
        finish=80000;
        h=0.05;
        p=0;

    var
        u:array[-20..20,0..1] of double;
        x0,x0dot:array[0..1] of double;
        tnow,x,xdot,term1,term2,term3:double;
        oneminusz,ts:double;
        c1,q,tsteps,c2,d1,d2:integer;
        f,g:text;
        g2,g1:file of double;
        failsign,path:string;

     function pow(x1,y1:double):double;
{       A failing of the language Pascal its its inability to raise a number x1 to the  }
{       power y1 and hence the operation has to be completed by an external C file      }
        external c;

    function dudz(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at  }
{       the point q1                                                                 }
        begin
        dudz:=(u[q1+1,p]-u[q1-1,p])/(2*h);
        end;

    function d2udz2(q1:integer):double;
{       Calculates the finite difference approximation of the spatial double         }
{       derivative at the point q1                                                   }
        begin
        d2udz2:=(u[q1+1,p]-2*u[q1,p]+u[q1-1,p])/sqr(h);
        end;

    function duplus(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at  }
{       the point q1 from above                                                      }
        begin
        duplus:=(-3*u[q1,p]+4*u[q1+1,p]-u[q1+2,p])/(2*h);
        end;

    function duminus(q1:integer):double;
{       Calculates the finite difference approximation of the spatial derivative at  }
{       the point q1 from below                                                      }
```

```
        begin
        duminus:=(3*u[q1,p]-4*u[q1-1,p]+u[q1-2,p])/(2*h);
        end;

    procedure resetvars;
{       Sets the latest set of information to be the old set...cycles one time step      }
        var
        ct:integer;
        begin
        u[0,p+1]:=0.0;
        for ct:=-20 to 20 do
        u[ct,p]:=u[ct,p+1];
        x0[p]:=x0[p+1];
        x0dot[p]:=x0dot[p+1];
        end;

    procedure printout;
{       Writes current freeze front information to the output file                      }
        begin
        writeln(f,tnow:11:8,' ',x0[p+1]:11:8,'        ',x0dot[p+1]:11:8);
        end;

    procedure solveeqns(k:double);
{       Solves the approximated equations at each time step of the algorithm in        }
{       the order...update solid temps, solid/liquid boundary, liquid temps, freeze    }
{       front position x0, lower solid boundary, upper liquid boundary.                }

        begin
        resetvars;
        tnow:=tnow+ts;
        for q:=1 to 19 do
        begin
        term1:=x0dot[p]*q*h*dudz(q-20)/(1+x0[p]);
        term2:=eta*d2udz2(q-20)/sqr(1+x0[p]);
        u[q-20,p+1]:=u[q-20,p]+k*(term1+term2);
        end;
        x0dot[p+1]:=(beta*duminus(0)/(1+x0[p])-duplus(0)/(x1-x0[p]))/alpha;
        for q:=1 to 19 do
        begin
        oneminusz:=1-q*h;
        term1:=x0dot[p]*oneminusz*dudz(q)/(x1-x0[p]);
        term2:=d2udz2(q)/sqr(x1-x0[p]);
        u[q,p+1]:=u[q,p]+k*(term2+term1);
        end;
        x0[p+1]:=x0[p]+k*x0dot[p+1];
        u[-20,p+1]:=(4*u[-19,p]-u[-18,p])/3;
        u[20,p+1]:=(4*u[19,p+1]-u[18,p+1])/3
        end;

    procedure findtimestep;
{       Calculates an optimum time step at each cycle of the algorithm                 }
        var
        d1,d2:double;
        begin
        d1:=0.5*sqr(h*(1+x0[p+1]))/eta;
        d2:=sqr(h*(x1-x0[p+1]))/2;
        if d1<d2 then ts:=tol*d1 else ts:=tol*d2;
        if ts>0.0001 then ts:=0.0001;
        end;
```

```
    procedure stopprog;
{       Terminates program if algorithm is diverging                                    }
        var
        count:integer;
        begin
        c2:=finish;
        printout;
        writeln(f,'failed');
        for count:=-20 to 20 do
        writeln(f,'u[',count:2,',p]=',u[count,p]);
        failsign:=outputfile+'FAILED';
        rewrite(g,failsign);
        writeln(g,outputfile,' has failed');
        close(g);
        end;

    procedure partprint;
{       Generates data files every 1/100th of the total number of time steps            }
        var
        c1,dummy,d1,d2:integer;
        begin
        dummy:=round(c2*100/finish);
        d1:=dummy div 10;
        d2:=dummy mod 10;
        path:='/home/maths/idw/1Dfin/data/'+outputfile+chr(d1+48)+chr(d2+48);
        if c2>=finish then path:='/home/maths/idw/1Dfin/data/'+outputfile+'last';
        open(g2,path,'new');
        write(g2,tnow);
        write(g2,x0[p+1]);
        write(g2,x0dot[p+1]);
        for c1:=-20 to 20 do
        write(g2,u[c1,p+1]);
        close(g2);
        end;

{       Main program...loads in start data, generates start solution freeze front      }
{       profile and controls the running of the algorithm                               }
begin
        open(g1,startfile,'old');
        reset(g1);
        read(g1,tnow);
        read(g1,x0[p+1]);
        read(g1,x0dot[p+1]);
        for c1:=-20 to 20 do
        read(g1,u[c1,p+1]);
        close(g1);
        rewrite(f,outputfile);
        for c1:=1 to 9 do
        begin
        x:=lambda*sqrt(tnow*c1/10);
        xdot:=lambda*0.5/sqrt(tnow*c1/10);
        writeln(f,tnow*c1/10:11:8,' ',x:11:8,'      ',xdot:11:8);
        end;
        resetvars;
        findtimestep;
        printout;

        c2:=0;
        repeat
        solveeqns(ts);
```

```
        if c2 mod round(finish/100)=0 then partprint;
        if c2 mod round(finish/1000)=0  then printout;
        findtimestep;
        c2:=c2+1;
        if x0[p+1]>x1 then stopprog;
        if x0[p+1]<-0.999 then stopprog;
        until c2>=finish;

        partprint;
        writeln(outputfile,' done');
        close(f);
end.
```

REFERENCES

1. J. Crank, *Free and Moving Boundary Problems,* Clarendon Press, Oxford, (1984).

2. I. Tadjbakhsh and W. Liniger. 'Free boundary problems with regions of growth and decay - Analysis of heat transfer in dip soldering process.' *Q.J. Mech. Appl. Math.*, vol **17**(2), pp. 141-155, (1964).

3. P.V. Padmanabhan and M.V. Krishnamurthy. 'Freezing around a finite heat sink immersed in an infinite phase change medium.' *Int. J. Heat and Fluid Flow*, vol **6**(2), pp.104 - 112, (1985).

4. H.S. Carslaw and J.C. Jaeger, *Conduction of Heat in Solids*, 2nd ed., Chapter 11, Oxford University Press, (1959).

5. G.E. Bell, 'A refinement of the heat balance integral method applied to the melting problem.' *Int. J. Heat and Mass Transfer,* **21** pp1357-61 (1978).

6. G.E. Bell, 'Solidification of a liquid about a cylindrical pipe.' *Int. J. Heat and Mass Transfer,* **22** pp1681-6 (1979).

7. G.E. Bell, 'On the performance of the enthalpy method.' *Int. J. Heat and Mass Transfer,* **25** pp587-9 (1982).

8. V. Voller and M. Cross, 'Accurate solutions of moving boundary problems using the enthalpy method.' *Int. J. Heat and Mass Transfer,* **24** pp545-556 (1980).

9. L.E. Goodrich, 'Efficient numerical technique for one-dimensional thermal problems with phase change.' *Int. J. Heat and Mass Transfer,* **21** pp615-621 (1978).

10. S.W. Churchill and J.P. Gupta, 'Approximations for conducting with freezing or melting.' *Int. J. Heat and Mass Transfer,* **20** pp1251-3 (1977).

11. H.G. Landau, 'Heat conduction in a melting solid.' *Q. J. Appl. Math.* **8**(1) pp81-94 (1950).

12. J. Crank, *Mathematics of Diffusion*.2nd Edn, pp323-324. Clarendon Press, Oxford (1975).

13. J. Crank, 'Two methods for the numerical solution of moving-boundary problems in diffusion and heat flow.' *Q. J. Appl. Math.* **10** pp220-231 (1957).

14. A.E. Fouda *et al*, 'Solar storage systems using salt hydrate latent heat and direct contact heat.' *Solar Energy* **25** pp437 - 444 (1980).

15. G.E. Bell and I.D. Wedgwood, 'Predicting the lifetime and extent of a transient solidified layer', *Proc. 2nd Int. Conf. Heat Transfer* , Vol **1**: Conduction, radiation and phase change, pp351 - 363 (1992).

16. G.E. Bell and I.D. Wedgwood, 'Freezing / thawing around a frozen body immersed in a finite bath of liquid of the same material.' In *Numerical Methods in Thermal Problems* **8** (Ed. R.W. Lewis) pp127-138, Pineridge Press, Swansea, U.K. (1993).

17. J. Crank, 'A nostalgic look at the mechanical solution of Stefan problems.' *Bull. I.M.A.* **11** pp32-33 (1975).

18. M.E. Rose, 'A method for calculating solutions of parabolic equations with a free boundary.' *Math. Comput.* **14** pp249-256 (1960).

19. V.R. Voller, M. Cross and P.G. Walton, 'Assessment of weak solution numerical techniques for solving Stefan problems.' In *Numerical Methods in Thermal Problems* (Ed. R.W. Lewis and K. Morgan) pp172-181 Pineridge Press, Swansea, U.K. (1979).

20. G.E. Bell, 'The accurate solution of one-dimensional melting problems by the heat balance integral method' In *Numerical Methods in Thermal Problems* (Ed. R.W. Lewis and K. Morgan) pp196-203 Pineridge Press, Swansea, U.K. (1979).

21. J. Stefan, 'Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere' *Ann. Phys. u. Chem.* (Wiedermann) Neue.Folge. **42** pp269-86 (1891).

22. J.S. Goodling and M.S. Khader, 'A numerical solution for outward solidification problems' In *Numerical Methods in Thermal Problems* (Ed. R.W. Lewis and K. Morgan) pp204-215 Pineridge Press, Swansea, U.K. (1979).

23. G. Butler and P.B. Johns, 'The solution of moving boundary heat problems using the TLM method of numerical analysis' In *Numerical Methods in Thermal Problems* (Ed. R.W. Lewis and K. Morgan) pp189-195 Pineridge Press, Swansea, U.K. (1979).

24. T.R. Goodman, 'Applications of integral methods to transient non-linear heat transfer' In *Advances in Heat Transfer* (Ed. T.F. Irvine Jr and J.P. Hartnett) **1** pp51-122 Academic Press (1964).

25. W.D. Murray and F. Landis, 'Numerical and machine solutions of transient heat conduction problems involving melting or freezing' *J. Heat Transfer* **81** pp106-112 (1959).

26. L. Fox, 'What are the best numerical methods?' In *Moving Boundary Problems in Heat Flow and Diffusion* (Ed. J.R. Ockendon and W.R. Hodgkins) pp210-241 Oxford University Press, Oxford (1975).

27. G. Duvaut, 'The solution of a two-phase Stefan problem by a variational inequality' In *Moving Boundary Problems in Heat Flow and Diffusion* (Ed. J.R. Ockendon and W.R. Hodgkins) pp173-181 Oxford University Press, Oxford (1975).

28. B. Noble, 'Heat balance methods in melting problems' In *Moving Boundary Problems in Heat Flow and Diffusion* (Ed. J.R. Ockendon and W.R. Hodgkins) pp208-209 Oxford University Press, Oxford (1975).

29. V.R.Voller, 'Implicit finite-difference solutions of the enthalpy formulation of Stefan problems' *I.M.A J. Num. Anal.* **5** pp201-214 (1985).

30. C. Bellecci and M. Conti, 'Phase change thermal storage: transient behaviour analysis of a solar reciever / storage module using the enthalpy method' *Int. J. Heat Mass Transfer* **36** pp2157-2163 (1993).

31. J. Crank and R.D. Phahle, 'Melting Ice by the isotherm migration method' *Bull. I.M.A.* **9** pp12-14 (1973).

32. J. Douglas,Jr. and T.M. Gallie,Jr., 'On the numerical integration of a parabolic differential equation subject to a moving boundary condition' *Duke Mathematical Journal* **22** pp557-571 (1955).

33. G.W. Evans II, E. Isaacson and J.K.L. MacDonald, 'Stefan-like problems' *Quart. Appl. Math* **8** pp312-319 (1950).

34. D.N. De G. Allen and R.T. Severn, 'The application of relaxation methods to the solution of non-elliptic partial differential equations' *Quart. J. Mech. App. Math* **5** pp447-454 (1952).

35. D. Langford, 'The heat balance integral method' *Int. J. Heat Mass Transfer* **16** pp2424-2428 (1973).

36. A.B. Crowley, 'Numerical solution of Stefan problems' *Int. J. Heat Mass Transfer* **21** pp215-218 (1978).

37. C. Bonacina, G. Comini, A. Fasano and M. Primicerio, 'Numerical solution of phase change problems' *Int. J. Heat Mass Transfer* **16** pp1825-1832 (1973).

38. R.S. Gupta and D. Kumar, 'Complete numerical solution of the oxygen diffusion problem involving a moving boundary' *Comput. Meth. appl. Mech. Engng.* **29** pp233-239 (1981).

39. G.H. Meyer, 'One-dimensional parabolic free boundary problems' *SIAM J. Num. Anal.* **19**(1) pp17-34 (1977).

40. G.S.H. Lock, 'On the use of asymptotic solutions to plane ice-water problems' *Journal of Glaciology* **8**(53) pp285-300 (1969).

41. N.R. Eyres, D.R. Hartree, J. Ingham, R. Jackson, R.J. Sarajant and J.B. Wagstaff, 'The calculation of variable heat flow in solids' *Phil. Trans. Roy. Soc.* **240** pp1-57 (1946).

42. D.R. Atthey, 'A finite difference scheme for melting problems' *J. Inst. Maths Applics* **13** pp353-366 (1974).

43. A.V. Hill, 'The diffusion of oxygen and lactic acid through tissues' *Proc. Roy. Soc. B* **104** pp39-96 (1928).

44. J. Crank and R.S. Gupta, 'A method for solving moving boundary problems in heat flow using cubic splines or polynomials' *J. Inst. Maths Applics.* **10** pp296-304 (1972).

45. G. Poots, 'On the application of integral-methods to the solution of problems involving the solidification of liquids initially at fusion temperature' *Int. J. Heat Mass Transfer* **5** pp525-531 (1962).

46. R.M. Furzeland, 'A comparitive study of numerical methods for moving boundary problems' *J. Inst. Maths Applics* **26** pp411-429 (1980).

47. R.S. Gupta and Dhirendra Kumar, 'A modified variable time step method for the one-dimensional Stefan problem' *Computer Methods in Applied Mechanics and Engineering* **23** pp101-109 (1980).

48. G. Borgioli, E. Di Benedetto and M. Ughi, 'Stefan problems with nonlinear boundary conditions: the polygonal method' *ZAMM* **58** pp539-546 (1978).

49. A. D. Solomon, M. D. Morris, J. Martin and M. Olszewski, 'The development of a simulation code for a latent heat thermal energy storage system in a space station' ORNL - 6213, Oak Ridge National Laboratory, Oak Ridge, Tennessee (1986).

50. K.J. Bathé, *Finite element procedures in engineering analysis,* Prentice Hall, New Jersey (1982).

51. E.C. Lemmon, 'Phase change techniques for finite element conduction codes' in *Proc. 1st Int. Conf. Num. Meths. in Thermal Problems,* University College, Swansea, July 1979 pp149-158 Pineridge Press, Swansea (1979).

52. G. Comini, S. Del Guidice, R.W. Lewis and O.C. Zienkiewicz, 'Finite element solution of non-linear heat conduction problems with special reference to phase change' *Int. J. Num. Meths. Eng.* **8**, pp613-624 (1974).

53. K. Morgan, R.W. Lewis and O.C. Zienkiewicz, 'An improved algorithm for heat conduction problems with phase change' *Int. J. Num. Meths. Eng.* **12** pp1191-1195 (1978).

54. K.K. Tamma and R.R. Namburu, 'Recent advances, trends and new perspectives via enthalpy-based finite element formulations for application to solidification problems' *Int. J. Num. Meths. Eng.* **30** pp803-820 (1990).

55. A.J. Dalhuijsen and A. Segal, 'Comparison of finite element techniques for solidification problems' *Int. J. Num. Meths. Eng.* **23** pp1807-1829 (1986).

56. W.D. Rolph and K.J. Bathé, 'An efficient algorithm for analysis of non-linear heat transfer with phase change' *Int. J. Num. Meths. Eng.* **18** pp119-134 (1982).