Numerical Computation of Moving Boundary Phenomena

A Thesis
Submitted to The
Faculty of Engineering
for The Degree of
Doctor of Philosophy

By

Mohamed ZERROUKAT

Department of Mechanical Engineering
University of Glasgow

I dedicate this thesis to my parents, brothers and sister
for their encouragement, support and understanding.

Acknowledgements

I would like to thank my supervisor, Dr. C.R. Chatwin, for his support, encouragement and modesty, which has shown throughout the course of this work.

My thanks also are due to:

- Professor D.C. Gilles, from the Computing Science Department, for his valuable discussions on many occasions.
- All the staff of the Department of Mechanical Engineering, especially the LOSEG group, for providing the necessary means to acheive this work.
- Professor B.F. Scott for the facilities he has provided, and for financial assistance to attend a conference.
- My friends for their support and understanding.

Finally, the financial assistance by the Algerian government is gratefully acknowledged as without it my studies would not have been possible.

ABSTRACT

When matter is subjected to a gradient of: temperature, pressure, concentration, voltage or chemical potential a phase change may occur, which for dynamic processes will be separated by moving boundaries between the adjacent phases. Transport properties vary considerably between phases, consequently any change in phase modifies the rate of transport of: energy, momentum, charge or matter which are fundamental to the behaviour of many physical systems. Such dynamic multi-phase problems have, for historical and mathematical reasons, become known as either: *Stefan problems* or *Moving Boundary Problems* (MBPs).

In most engineering applications the analysis of these problems is often impossible without recourse to numerical schemes which utilise either: finite difference or finite element methods. The success of finite element methods is their ability to handle complex geometries; however, they are time consuming and less amenable to vectorisation than finite difference techniques which, because of their greater simplicity in formulation and programming, continue to be the more popular choice.

Several finite difference schemes are available for the solution of moving boundary problems; however, there are some difficulties associated with each method. Each time a new numerical scheme is developed, it has the aim of improving either, or both, the accuracy and the computational performance. For solving one–dimensional moving boundary problems, the variable time step grid is the best approach in terms of simplicity and computational efficiency. Due to the fact that the time step is variable the implicit recurrence formulae, which are stable for any mesh size, have always been used with this type of discretisation of the space time domain. It will be shown in the course of this thesis that the implicit methods are very inaccurate when used with relatively large time steps; hence, the immediate conclusion may be made – that implicit variable time step methods may not be sufficiently accurate to solve moving boundary problems where the boundary is moving with a relatively slow velocity.

The proposed idea, of combining real and virtual grid networks and using new explicit finite difference equations, eliminates the loss of accuracy associated with implicit methods, when the time step is large, and offers higher computational performance. The new finite difference equations are based on the approach of making the finite difference substitution into the solution of the partial differential equation rather than into the partial differential equation itself, which is the classical approach. A new numerical scheme for two-phase Stefan problems which will be referred to as the EVTS method is

developed and the solution is compared to other numerical methods as well as the analytic solution.

Furthermore, the EVTS method is modified to solve implicit moving boundary problems (oxygen diffusion problem), in which an explicit relation containing the velocity of the moving boundary is absent. The resulting method achieves similar results to other more complex and time consuming methods.

A further numerical scheme referred to as the ZC method is developed to deal with heat transfer problems involving three phases (or 2 moving boundaries) which appear and disappear during the process. To the knowledge of the author, a finite difference method for such a problem does not exist. For validation, numerical results are compared with those of the conservative finite element method of Bonnerot and Jamet, which is the only other method available to solve two-moving boundary problems.

Finally, a new finite difference solution for non–linear problems is developed and applied to laser heat treatment of materials. The numerical results are in good agreement with published experimental results.

CONTENT

page

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1. Introduction

Most physical processes can be modelled as boundary value problems, in which the solution of differential equations has to satisfy certain conditions on the boundaries of a prescribed domain. However, in many important processes involving the changing states of matter, a boundary separating two different phases develops during the process. In these problems, the position of the boundary is not known *a priori*, but has to be determined as an integral part of the solution. The term "*Moving Boundary Problems (MBPs)*" is associated with time-dependent boundary problems, where the position of the Moving Boundary (MB) has to be determined as function of time and space. Moving boundary problems, also known as Stefan problems, were studied as early as 1831 by Lame and Clapeyron [1]. However, J. Stefan was given the major credit due to a sequence of papers [2,3] which resulted from his study of the melting of the polar ice cap around 1890.

The formulation of MBPs requires not only the initial and boundary conditions to be known, as in boundary value problems, but two more conditions are needed on the moving boundary; one to determine the boundary itself and the other to complete the solution of partial differential equations governing the process in each region (formulation examples are given in appendix A).

The applications of these problems are mainly but not exclusively concerned with: fluid flow in porous media, diffusion problems, heat transfer involving phase transformation, shock waves in gas dynamics and cracks in solid mechanics [4].

Moving boundaries also occur in many processes associated with: the metal, glass, plastic and oil industries; preservation of foodstuffs; statistical decision theory, heat treatment of semi–conductors; cryosurgery; astrophysics; meteorology; geophysics and plasma physics [5,6].

In the early years, experimental analyses were the only means available to give an understanding of physical processes. However, with the advent of high speed digital computers, mathematical modelling and computer simulation are often the cheapest and fastest means to give a broad understanding of the real process. This permits the simulation of the performance of new product designs and the assessment of quality even before production, which results in greater savings in terms of time and money.

Due to the wide range of applicability in engineering and science, in the last two decades MBPs have drawn considerable attention from mathematicians, engineers and scientists alike. As analytical solutions are often impossible for most engineering and science applications, recourse is often made to numerical analysis.

Finite Difference Methods (FDMs) have been used extensively for the numerical solution of MBPs and only in recent years, Finite Element Methods (FEMs) have been introduced. The great advantage of FEMs is their ability to handle complex geometries; however, it is well acknowledged that they are time consuming and less suitable to vectorisation than FDMs.

Finite difference methods for solving MBPs can be classified into:

(i) Fixed Grid Methods (FGMs),

(ii) Variable Space–Step Methods (VSSMs),

(iii) Variable Time–Step Methods (VTSMs).

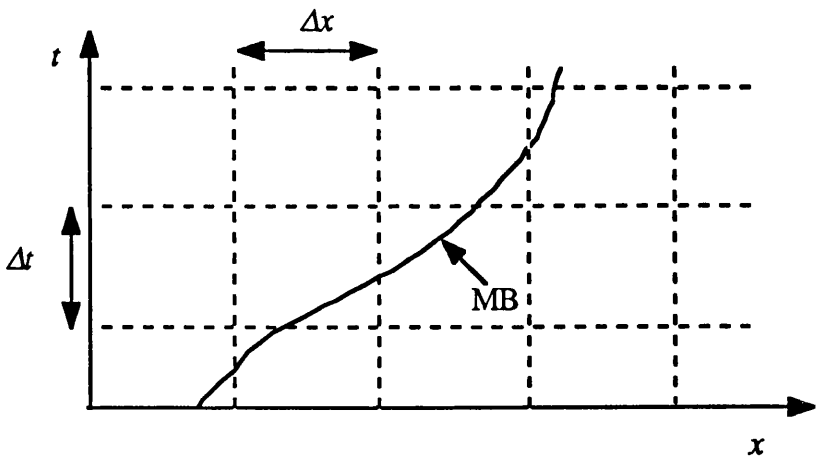

Figure 1.1: Position of the MB in a fixed grid network

FGMs, where the Moving Boundary (MB) is often located between two neighbouring grid points (Figure 1.1), break down when the boundary moves a distance larger than a space increment during a time step. This restriction, upon the velocity of the moving

boundary, may considerably increase the array size (memory) requirements and the cpu-time if computations are to be performed for extended times. The possibility of break down of the scheme is avoided in VSSMs, by using a variable space elements (as shown in Figure 1.2).

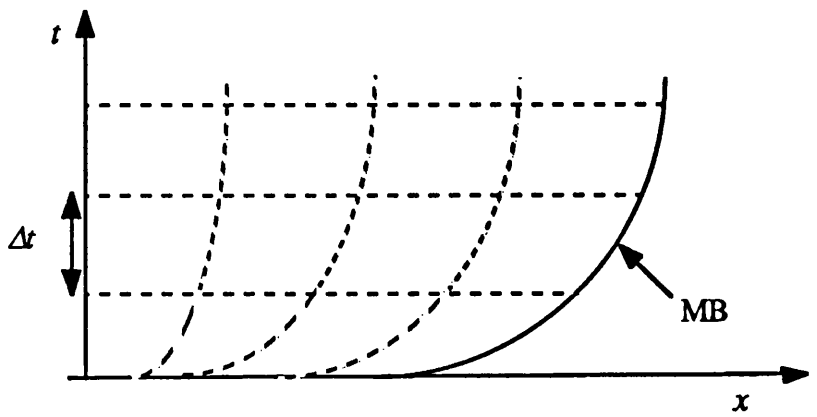

Figure 1.2: Position of the MB in a variable space grid network

However, these methods present considerable computational difficulties (underflow and overflow) at the beginning of computation when the MB is too close to the fixed boundaries, and loses accuracy at later times, due to the enlargement of space elements which is a consequence of the displacement of the MB.

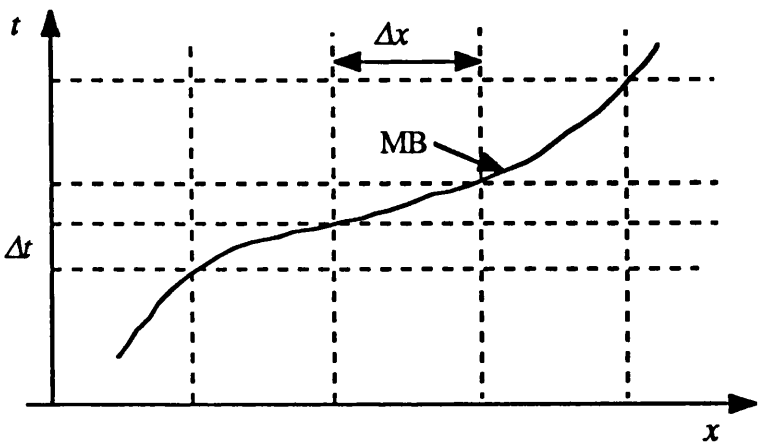

Figure 1.3: Position of the MB in a variable time grid network

By using VTSMs – where a variable time grid network is adopted (Figure 1.3) rather than a space variable grid – the problems associated with FGMs and VSSMs are avoided, this makes VTSMs the most attractive approach. However, as the accuracy of VTSMs depends on the accuracy of recurrence formulae such as Crank-Nicolson and fully implicit equations which are very inaccurate when time step becomes relatively large; VTSMs may not be sufficiently accurate if used to solve MBPs where the moving boundary has a relatively slow velocity.

1.2. Layout of the thesis

Chapter 2 reviews some of the well known numerical methods frequently used to solve moving boundary problems. The methods are briefly described in order to illustrate the essence of each approach. This chapter also contains important bibliographies which will assist any reader who has a profound interest in the topic.

A comparative study of the finite difference equations, which represents the basis of any finite difference scheme, is given in chapter 3. The study shows that the implicit methods, which are stable for any mesh size, are very inaccurate when used with large time steps. The combination of the proposed Virtual Sub-Interval Elimination Technique (VSIET) and the explicit finite difference equations offers higher performances than the implicit equations.

Due to the inaccuracy of the implicit finite difference equations when the Fourier number is relatively large, a new numerical scheme which will be referred to as the Explicit Variable Time Step (EVTS) method is developed for solving two–phase Stefan problems. Numerical results show that the EVTS method out–performs the implicit methods in current use. This is the subject of chapter 4.

In chapter 5, the EVTS method is modified to solve implicit moving boundary problems (the oxygen diffusion problem). Numerical results compare very favourably with those due to earlier authors.

In chapter 6, the EVTS method is extended to deal with multiple moving boundary problems, where the problem can have more than one moving boundary simultaneously. The method is applied to the collapse of a solid wall, where both melting and vaporisation interfaces appear and disappear during the process. Numerical results compare very well with those of the finite element solutions of Bonnerot and Jamet [7].

In chapter 7, a new finite difference equation for non-linear problems is developed and applied to the laser heat treatment of materials. Numerical results are in good agreement with published experimental results.

1.3. References

1. M.M. Lame and B.P.E. Clapeyron, *Ann. Chem. Phys.* **47**, 250-256 (1831).
2. J. Stefan, *Sber. Akad. Wiss. Wien.* **98**, 473-484 (1889).
3. J. Stefan, *Ann. Chem. Phys.* **42**, 269-286 (1891).
4. J. Crank, *Free moving boundary problems*, Clarendon Press, Oxford (1984).
5. M. Furzeland, *J. Inst. Math. Appl.* **26**, 411-429 (1980).
6. E. Magenes (ed.), *Free boundary Problems vols. I and II*, Instituo Nazionale di Alta Matematica Francescon Severi, Rome (1980).
7. R. Bonnerot and P. Jamet, *J. Comput. Phys.* **41**, 357-378 (1981).

2 Numerical Methods for Moving Boundary Problems

Summary

In this chapter, some of the well known numerical methods which can be used to solve moving boundary problems are reviewed. These methods compute, at each time step, the position of the moving boundary as well as the temperature at each grid point of the space–time domain. Most of the numerical methods cited in this chapter are described briefly; however, due to their utility in a later chapter, the heat balance and the variable time step methods are explained more fully.

Nomenclature

a	Material Thickness	λ	Latent heat of fusion/unit volume
c	Specific heat	Δt	Time step
h	Heat transfer coefficient	Δx	Space element
H	Enthalpy		*Subscripts*
K	Thermal conductivity	i	Space step index
M	Position of the interface	i_b	Interface node index
L	Latent heat of fusion	j	Time step index
r	Fourier number	m	Melting
t	Time variable	s	Solid
T	Temperature	l	Liquid
x	Space variable		*Superscript*
α	Diffusivity	k	Iteration index
ρ	Material density		

2.1. Introduction

Research on moving boundary problems started as early as 1831. However, with the onset of the computer revolution, MBPs have been the focus of intensive research during the seventies and eighties, during which a large number of numerical schemes have been developed. It is not practical to review all of them in this chapter; however, many surveys and conference reports [1–14] are available which contain up to date accounts of the mathematical developments in this field and illustrate wide ranging applications in physics, biology and engineering.

Finite difference methods have been used extensively for solving moving boundary problems. However, before presenting the numerical techniques, the well known Goodman method [15] is illustrated; this is an analytical method which is often used as a reference against which to validate the numerical methods.

2.2. The heat balance integral method (Goodman)

By integrating the one dimensional heat flow equation with respect to the space variable x, and inserting the boundary conditions, Goodman [15] produced an integral equation which expresses the overall heat balance of the system. The successive steps in Goodman's method are:

(i) Assume a particular form for the dependence of the temperature on the space variable which is consistent with the boundary conditions, e.g. a polynomial relationship.

(ii) Integrate the heat flow equation with respect to the space variable over the appropriate interval and substitute the assumed temperature distribution to obtain the heat balance integral.

(iii) Solve the integral equation to obtain the motion of the phase change boundary and the time dependence of the temperature distribution.

2.2.1. Heat balance method for the one-phase ice melting problem

The heat balance method can be conveniently illustrated by solving the one-phase melting-ice problem in one space dimension as defined by the following non-dimensional equations:

$$\frac{\partial T}{\partial t}=\frac{\partial^2 T}{\partial x^2} \quad , \quad 0\le x\le M(t) \quad , \quad t>0 \tag{2.1}$$

$$T(x,t)=1 \quad , \quad x=0 \quad , \quad t>0 \tag{2.2}$$

$$T(x,t)=0 \quad , \quad x>0 \quad , \quad t=0 \tag{2.3}$$

$$M(0)=0 \tag{2.4}$$

$$T(x,t)=0 \quad , \quad x=M(t) \quad , \quad t>0 \tag{2.5}$$

$$-\frac{\partial T}{\partial x}=\frac{dM}{dt} \quad , \quad x=M(t) \quad , \quad t>0 \tag{2.6}$$

Integration of (2.1) with respect to x, from $x=0$ to $x=M(t)$, gives:

$$\frac{d}{dt}\int_0^{M(t)} T(x,t)\,dx=-\left(\frac{dM}{dt}+\frac{\partial T(0,t)}{\partial x}\right) \tag{2.7}$$

Assuming a temperature distribution in the water phase given by:

$$T(x,t)=\xi(t)\{x-M(t)\}+\varphi(t)\{x-M(t)\}^2 \tag{2.8}$$

which satisfies (2.5). It is convenient to modify the condition (2.6) to the form:

$$\left(\frac{\partial T}{\partial x}\right)^2=\frac{\partial^2 T}{\partial x^2} \quad , \quad x=M(t) \tag{2.9}$$

by using the standard formula:

$$\frac{dT}{dt}=\frac{\partial T}{\partial x}\frac{dM}{dt}+\frac{\partial T}{\partial t}=0 \quad , \quad x=M(t) \tag{2.10}$$

to replace (dM/dt) in (2.6). Taking into consideration that $T(0,t)=1$, $x=0$, (2.8) gives: $1=-\xi M+\varphi M^2$ and on differentiation, $\partial T/\partial x=\xi$, $\partial^2 T/\partial x^2=2\varphi$ at $x=M(t)$, so that the substitution into (2.9) yields:

$$\xi=\frac{1-\sqrt{3}}{M(t)} \quad , \quad \varphi=\frac{\xi M+1}{M^2} \tag{2.11}$$

Substitution of (2.8) incorporating $\xi(t)$ and $\varphi(t)$ from (2.11), and integrating the left side of (2.7) gives:

$$\frac{1}{6}\left(1+\sqrt{3}\right)\frac{dM}{dt}=-\frac{dM}{dt}-(\xi-2\varphi M) \tag{2.12}$$

Further elimination of ξ and φ from (2.12) gives:

$$M\frac{dM}{dt}=\frac{6\left(3-\sqrt{3}\right)}{7+\sqrt{3}}=\frac{1}{2}\beta^2 \qquad \text{where} \qquad \beta=\sqrt{\frac{12\left(3-\sqrt{3}\right)}{7+\sqrt{3}}} \tag{2.13}$$

from which it follows that:

$$M(t)=\beta\sqrt{t} \tag{2.14}$$

By substitution of (2.14) and (2.11) into (2.8), the temperature distribution which represents the solution of the problem, defined by (2.1) to (2.6), is readily obtained.

2.2.2. Heat balance method for two-phase melting of a finite slab problem

The heat balance method was applied to solve the melting of a finite slab of thickness a initially at a uniform temperature that is below the melting point. The problem is described by the following equations:

$$\frac{\partial T_l}{\partial t}=\alpha_l\frac{\partial^2 T_l}{\partial x^2} \quad , \quad 0\le x\le M(t) \tag{2.15}$$

$$\frac{\partial T_s}{\partial t} = \alpha_s \frac{\partial^2 T_s}{\partial x^2} \quad , \quad M(t) \le x \le a \tag{2.16}$$

$$\left.\begin{aligned} K_l \frac{\partial T_l}{\partial x} - K_s \frac{\partial T_s}{\partial x} &= \lambda \frac{dM}{dt} \\ T_l(x,t) = T_s(x,t) &= T_b \end{aligned}\right\} \quad , \quad x = M(t) \tag{2.17}$$

where the suffixes l and s denote liquid phase and solid phase respectively. The solution of the problem during the pre-melting stage is not presented since standard solutions are readily available in [16]. Goodman and Shea [17] have also solved this stage by the heat balance method.

Following the three stages (i) to (iii) mentioned in section 2.2, (2.15) is integrated from $x=0$ to $x=M(t)$ and (2.16) from $x=M(t)$ to $x=a$ to obtain the integral equations:

$$\alpha_l \left\{ \left(\frac{\partial T_l}{\partial x} \right)_{M(t)} - \left(\frac{\partial T_l}{\partial x} \right)_0 \right\} = \int_0^M \frac{\partial T_l}{\partial x} dx = \frac{d}{dt} \int_0^M T_l \, dx - T_b \frac{dM}{dt} \tag{2.18}$$

$$\alpha_s \left\{ \left(\frac{\partial T_s}{\partial x} \right)_a - \left(\frac{\partial T_s}{\partial x} \right)_{M(t)} \right\} = \frac{d}{dt} \int_M^a T_s \, dx + T_b \frac{dM}{dt} \tag{2.19}$$

Elimination of $(\partial T_l / \partial x)$ and $(\partial T_s / \partial x)$ at $x = M(t)$ from (2.18) and (2.19) making use of (2.17) yields:

$$\left(\frac{K_s}{\alpha_s} T_b - \frac{K_l}{\alpha_l} T_b - \lambda \right) \frac{dM}{dt} + \frac{K_l}{\alpha_l} \frac{d\theta_l}{dt} + \frac{K_s}{\alpha_s} \frac{d\theta_s}{dt} = K_s \left(\frac{\partial T_s}{\partial x} \right)_{x=a} - K_l \left(\frac{\partial T_l}{\partial x} \right)_{x=0} \tag{2.20}$$

where

$$\theta_l = \int_0^M T_l(x,t) \, dx \quad , \quad \theta_s = \int_M^a T_s(x,t) \, dx \tag{2.21}$$

Suitable temperature distributions are then chosen:

$$T_l(x,t) = T_b + \xi_l(t)\{x - M(t)\} + \varphi_l(t)\{x - M(t)\}^2 \tag{2.22}$$

$$T_s(x,t) = T_b + \xi_s(t)\{x - M(t)\} + \varphi_s(t)\{x - M(t)\}^2 \tag{2.23}$$

The parameters $\xi_l(t)$, $\xi_s(t)$, $\varphi_l(t)$ and $\varphi_s(t)$ are determined so that the temperature distributions, (2.22) and (2.23), satisfy the boundary conditions. For example, with the following boundary conditions:

$$-K_l\left(\frac{\partial T_l}{\partial x}\right)_{x=0} = F = \text{const.} \quad , \quad K_s\left(\frac{\partial T_s}{\partial x}\right)_{x=a} = 0 \quad , \quad T_b = 0 \tag{2.24}$$

Goodman and Shea [17] arrive at the following differential equations:

$$-\lambda\frac{dM}{dt} + \frac{K_l}{\alpha_l}\frac{d\theta_l}{dt} + \frac{K_s}{\alpha_s}\frac{d\theta_s}{dt} = F \tag{2.25}$$

$$\theta_l = M^2\left(\frac{F}{2K_l} - \frac{1}{3\alpha_l}\frac{d\theta_l}{dt}\right) \tag{2.26}$$

$$\theta_s = \frac{(a-M)^2}{3\alpha_s}\frac{d\theta_s}{dt} \tag{2.27}$$

Equation (2.25) is obtained by substitution of (2.24) into (2.20); the evaluation of $\xi_l(t)$ and $\varphi_l(t)$ using the conditions (2.24) and (2.18), and using the first integral of (2.21) gives (2.26). Equation (2.27) is derived by first evaluating $\xi_s(t)$ and $\varphi_s(t)$ using (2.24) and (2.19) and finally using the second of (2.21).

To solve the equations (2.25) to (2.27), the following initial conditions are needed:

$$M(t_m) = 0 \quad , \quad \theta_l(t_m) = 0 \quad , \quad \theta_s(t_m) = \frac{-F\,a^2}{3K_s} \tag{2.28}$$

$\theta_s(t_m)$ of (2.28) is obtained by integration with respect to x from $x = 0$ to $x = a$ of the temperature distribution $T_s(x,t_m)$ at the time when melting starts which is given by Goodman and Shea [17] as:

$$T_s(x,t_m) = \frac{-aF}{2K_s}\left\{\frac{x}{a}\left(2 - \frac{x}{a}\right)\right\} \tag{2.29}$$

Goodman and Shea obtained the solution of (2.25) to (2.27) by lengthy mathematical manipulations. As an alternative method, solution of (2.25) to (2.27), can be obtained easily using a standard method such as Runge–Kutta.

Many authors [18–24] have applied the heat balance method to different problems, as well as studied ways of improving and easing the mathematical analysis. It is well

acknowledged that the mathematical manipulations required for the heat balance method, for anything other than relatively simple problems, can be very lengthy and complicated. Furthermore, the choice of a satisfactory approximation to the temperature distribution is a major difficulty in the method. For example, the use of a high order polynomial for the temperature profile makes the approach highly complicated, and does not necessarily improve the accuracy of the solution.

The most promising way of improving the heat balance method was suggested by Noble [25]. The idea was to use spatial subdivisions and assume a quadratic profile in each sub–region, as in finite element methods. Bell [26] modified Noble's approach to solve a single phase melting problem, by using equal subdivisions of the temperature rather than subdividing the space variable. The solution of a system of differential equations provides the position of each isotherm identified by a subdivision, including that of the melting boundary, at successive times.

2.3. Fixed grid finite difference methods

The heat flow equation is solved by using finite difference replacements for the derivatives in order to compute values of temperature $T_{i,j}$, at $x_i = i\Delta x$ and time $t_j = j\Delta t$ on a fixed grid in the (x,t) plane. At any time $t_j = j\Delta t$, the phase change boundary will usually be located between two neighbouring grid points; for example: between $i_b\Delta x$ and $(i_b + 1)\Delta x$, as shown in Figure 2.1.

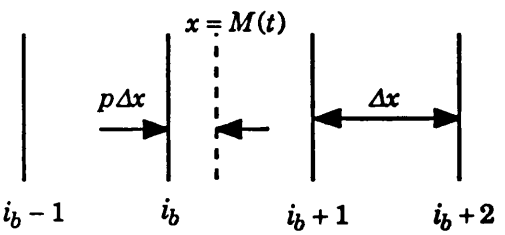

Figure 2.1: Location of the moving boundary in a fixed grid

The numerical solution of the single-phase problem, defined by equations (2.1) to (2.6), provides a simple example and proceeds as follows: the temperature $T_{i,j+1}$ is given by:

$$T_{i,j+1} = T_{i,j} + \left(\frac{\Delta t}{\Delta x^2}\right)\left\{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}\right\} \tag{2.30}$$

By using three point Lagrange interpolation [27], instead of (2.30), the temperature at $x = i_b\Delta x$ is calculated using:

$$T_{i_b,j+1} = T_{i_b,j} + \left(\frac{2\Delta t}{\Delta x^2}\right)\left\{\frac{1}{p_j+1}T_{i_b-1,j} - \frac{1}{p_j}T_{i_b,j}\right\} \tag{2.31}$$

The variation of the position of the moving boundary is given by:

$$p_{j+1} = p_j - \left(\frac{\Delta t}{\lambda\,\Delta x^2}\right)\left\{\frac{p_j}{p_j+1}T_{i_b-1,j} - \frac{p_j+1}{p_j}T_{i_b,j}\right\} \tag{2.32}$$

Various finite difference schemes have been proposed for approximating both the boundary conditions and the partial differential equation at the neighbouring grid points. For instance, Murray and Landis [28] used a fixed grid network. In particular, in the sub-space containing the fusion front at any time, they introduce two fictitious temperatures, one obtained by quadratic extrapolation from temperatures in the solid region and the other from temperatures in the liquid region. The fusion temperature and the current position of the interface are incorporated in the fictitious temperatures, which are then substituted into the standard approximation, such as (2.30), to compute temperatures near the interface instead of using special formulae like (2.31). For the motion of the fusion front, an expression analogous to (2.32) is used based on the Taylor extrapolation formulae. Furzeland [29] suggested using an approximation for $(\partial T / \partial x)$ on the MB with a fictitious temperature to be eliminated by use of (2.30).

The advantages of implicit finite difference equations have been investigated by Ehrlich [30], Koh *et al.* [31], Saitoh [32], and Bonnerot and Jamet [33]. For example Meyer [34] obtained a second order approximation for the movement of the MB by using a three-point backward formula.

Ciment and Guenther [35] employed a method of spatial mesh refinement on both sides of the MB, previously analysed by Ciment and Sweet [36]. They also incorporated the idea used earlier by Douglas and Gallie [37], of adjusting the time step so that the MB coincides with one of the refined mesh points.

Huber [38] used an implicit scheme for approximating the heat flow equation in each time step, but he tracked the MB explicitly using the Stefan condition; this approach was later simplified by Rubinstein [4] and Fasano and Primicerio [39].

Lazaridis [40] used explicit finite difference approximations on a fixed grid to solve two-phase solidification problems in both two and three space dimensions. He wrote the heat balance condition on the MB in Patel's expressions [40] and developed numerical schemes based on an auxiliary set of differential equations, which express the fact that the MB is an isotherm. These auxiliary forms are compatible with Patel's expressions in enabling the boundary movement to be computed in the direction of the co-ordinate axes, rather than along the normal to itself. Standard finite difference approximations to the heat flow equation were used at grid points far enough from the MB. Near the boundary,

formulae for unequal intervals were incorporated into the auxiliary equations. To avoid loss of accuracy associated with singularities which can arise when the MB is too near a grid point, localised quadratic temperature profiles were used. The mathematical manipulations are very lengthy and complex indeed.

2.4. Variable space grid finite difference methods

Many ways of modifying the grid have been proposed, all with the aim of avoiding the increased complication, and loss of accuracy, associated with unequal intervals near the MB in the fixed grid networks. Murray and Landis [28] kept the number of space intervals between a fixed and moving boundary constant, for example equal to q, for all time. The space interval, $\Delta x = M(t)/q$, is different for each time step. The MB is always on the q^{th} grid line. They differentiated partially with respect to x. Thus, for the line $x_i = i\Delta x$ they had:

$$\left(\frac{\partial T}{\partial t}\right)_i = \left(\frac{\partial T}{\partial x}\right)_t \left(\frac{dx}{dt}\right)_i + \left(\frac{\partial T}{\partial t}\right)_x \tag{2.33}$$

Murray and Landis assumed that a general grid point at x moved according to the expression:

$$\frac{dx_i}{dt} = \frac{x_i}{M(t)} \frac{dM}{dt} \tag{2.34}$$

The one–dimensional heat equation becomes:

$$\left(\frac{\partial T}{\partial t}\right)_i = \frac{x_i}{M(t)} \frac{dM}{dt} \left(\frac{\partial T}{\partial t}\right) + \frac{\partial^2 T}{\partial x^2} \tag{2.35}$$

The position of the moving boundary $M(t)$ is updated at each time step by using, for example, a suitable finite difference form of the boundary condition ($dM/dt = -\partial T/\partial x$) on $x = M(t)$. The method was extended by Heitz and Westwater [41] to convection problems and by Tien and Churchill [42] to cylindrical coordinates.

Crank and Gupta [43] avoided the complications due to the unequal grid size near the MB by moving the whole uniform grid system with the velocity of the MB. They described two ways of obtaining the interpolated values of temperature at the new grid points, to be used for the next time step, based on cubic splines or polynomials. Later Gupta [44,45] avoided the interpolations by using a Taylor's expansion of space and time variables and derived an equation which can be seen as a particular case of (2.35).

2.5. Variable time step grid finite difference methods

Douglas and Gallie [37] rather than using a fixed time step and searching for the position of the moving boundary decided to determine, as part of the solution, a variable time step such that the MB always coincides with a grid line in space. They used the fully implicit finite difference equation, this leads to a system of linear equations to be solved at each time step.

Gupta and Kumar [46] formulated the same set of finite difference equations as Douglas and Gallie but they used the interface condition to update the time step Δt. They thus avoided the instability which develops as the depth of the MB increases; the instability is generated as the time step enters an infinite loop because it becomes very sensitive to rounding errors. Goodling and Khader [47] incorporated the finite difference form of the MB condition into the system of equations to be solved. The system is solved for an arbitrary value of the temperature of the node neighbouring the MB, which is then updated from the boundary condition. Gupta and Kumar [48] found this latter method fails to converge as the computation progresses in time because it is too sensitive to a small change in temperature. Gupta and Kumar's results are in good agreement with: those obtained by the other variable time step methods and by the integral method of Goodman. Gupta and Kumar [49] also adapted the method of Douglas and Gallie to solve the oxygen diffusion problem (implicit boundary condition). Their results compares favourably with results obtained by Hansen and Hougaard [50] and by Dahmardah and Mayers [51].

To illustrate these methods, consider the following non-dimensional problem:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le M(t) \quad , \quad t > 0 \tag{2.36}$$

$$\frac{\partial T}{\partial x} = hT + \psi \quad , \quad x = 0 \quad , \quad t > 0 \tag{2.37}$$

$$T(x,t) = 1 \quad , \quad M(t) \le x \le 1 \quad , \quad t \ge 0 \tag{2.38}$$

$$\frac{dM}{dt} = \frac{\partial T}{\partial x} \quad , \quad x = M(t) \quad , \quad t > 0 \tag{2.39}$$

$$M(0) = 0 \tag{2.40}$$

where $x = M(t)$ is the distance of the interface from $x = 0$.

The region $0 \le x \le 1$ is divided into N sections of length $\Delta x = 1/N$. The time step Δt_j, at each time step j, is chosen such that the boundary moves a distance Δx during that interval of time. Any point (x_i, t_j) in the (x,t) domain is given by the coordinates:

$$\left(x_i = i\,\Delta x, t_j = \sum_{m=0}^{j-1} \Delta t_m \right)$$

where Δt_m denotes the time interval in which the boundary moves from $x = m\Delta x$ to $x = (m+1)\Delta x$.

Suppose the moving boundary is moving from the nodal point (i_b, j) to $(i_b + 1, j+1)$, then the problem is to compute the time step Δt_j necessary for that move as well as the temperature distribution – $T_{i,j+1}, i = 0, N$ – at the time t_{j+1}, from a known temperature distribution at the time t_j (see Figure 2.2).

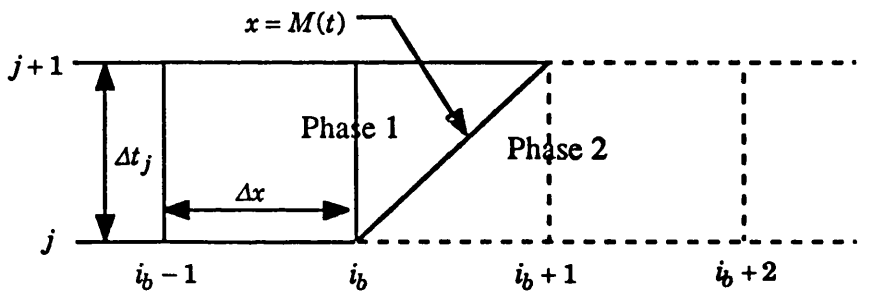

Figure 2.2: Movement of the MB $x = M(t)$ in a variable time step grid network

2.5.1. The Extended Douglas and Gallie's (EDG) method

Let $T_{i,j}$ be the temperature at the coordinates (x_i, t_j). Replacing the left side of (2.36) by backward difference at the point (x_i, t_j) and the right side by central difference, gives:

$$\frac{T_{i,j+1} - T_{i,j}}{\Delta t_j} = \alpha \left(\frac{T_{i-1,j+1} - 2T_{i,j+1} + T_{i+1,j+1}}{\Delta x^2} \right) \qquad (2.41)$$

Rearrangement of (2.41) gives:

$$(1 + 2r_j) T_{i,j+1} - r_j \left(T_{i-1,j+1} + T_{i+1,j+1} \right) = T_{i,j} \quad , \quad i = 1, i_b \qquad (2.42)$$

where $r_j = \alpha\, \Delta t_j / \Delta x^2$.

In order to determine Δt_j, both sides of (2.36) are integrated with respect to x, from $x = 0$ to $x = M(t)$, making use of (2.37)

$$\frac{d}{dt}\left(\int_0^{M(t)} T(x,t)\,dx - (\alpha+1)M(t)\right) = -\alpha(hT(0,t) - \psi) \tag{2.43}$$

Further integration of (2.43) with respect to t, and making use of (2.40), gives:

$$\alpha\psi t + \alpha h\int_0^t T(0,t)\,dt = (\alpha+1)M - \int_0^M T(x,t)\,dx \tag{2.44}$$

The finite difference replacement of (2.44), when $t = t_{j+1}$ is made to give:

$$\alpha\psi t_{j+1} + \alpha h\sum_{p=1}^{j+1}\left(T_{0,p}\,\Delta t_{p-1}\right) = (\alpha+1)(j+1)\Delta x - \Delta x\sum_{i=1}^{i_b+1} T_{i+1,j+1} \tag{2.45}$$

After manipulation, (2.45) gives:

$$\Delta t_j = \frac{\left\{(\alpha+1)(j+1)\Delta x - \alpha\psi t_j + \alpha h\sum_{p=1}^{j}\left(T_{0,p}\,\Delta t_{p-1}\right) - \Delta x\sum_{i=1}^{i_b+1} T_{i+1,j+1}\right\}}{\alpha\left(h + \psi T_{0,j+1}\right)} \tag{2.46}$$

It should be noted that in (2.45), the finite difference replacement of the integrals have been made such that the value of Δt_j, obtained from (2.46), matches with the one obtained by satisfying the boundary conditions (2.37) and (2.39). The special case of $i_b = 1$ is discussed in section 2.5.3.

By choosing a suitable estimate of Δt_j, the temperature at any grid point is computed from (2.42) and the estimated value of Δt_j is subsequently re-evaluated from (2.46). This iterative process is repeated until the desired accuracy in Δt_j is achieved. The k^{th} iteration for solving the system of $(i_b + 1)$ linear equations is:

$$\left(1+2r_j^k\right)T_{i,j+1}^k - r_j^k\left(T_{i-1,j+1}^k + T_{i+1,j+1}^k\right) = T_{i,j} \quad , \quad i = 1, i_b \tag{2.47}$$

$$T_{1,j+1}^k - (1+h\Delta x)T_{0,j+1}^k = \psi\Delta x \tag{2.48}$$

where $r_j^k = \alpha\,\Delta t_j^k / \Delta x^2$ and $T_{i_b+1,j+1}^k = 1$ from (2.38) for all k.

Similarly, the $(k+1)^{th}$ iteration for calculating Δt_j can be written from (2.46) as:

$$\Delta t_j^{k+1} = \frac{\left\{(\alpha+1)(j+1)\Delta x - \alpha\psi t_j + \alpha h \sum_{p=1}^{j}\left(T_{0,p}\,\Delta t_{p-1}\right) - \Delta x \sum_{i=1}^{i_b+1} T_{i+1,j+1}^{k}\right\}}{\alpha\left(h+\psi T_{0,j+1}^{k}\right)} \tag{2.49}$$

Choosing Δt_j^0 equal to Δt_{j-1} which is already known, $T_{i,j+1}^0, i=0,i_b$ are determined from (2.47) and (2.48). These values are in turn used to determine the new estimate of Δt_j^1 from (2.49). Δt_j^1 is substituted in (2.47) and (2.48) to give a new temperature distribution $T_{i,j+1}^1, i=0,i_b$. This process is repeated until the desired accuracy in Δt_j is achieved.

2.5.2. Goodling and Khader's (GK) method

Goodling and Khader [47] used the same set of simultaneous equations, (2.47) and (2.48), to determine the temperatures at time t_{j+1} at the k^{th} iteration. The boundary condition (2.39) is replaced by the following finite difference equation:

$$\frac{1}{\Delta x}\left(T_{i_b+1,j+1}^{k} - T_{i_b,j+1}^{k}\right) = \frac{\Delta x}{\Delta t_j^k} \tag{2.50}$$

giving:

$$\Delta t_j^k = \frac{\Delta x^2}{1 - T_{i_b,j+1}^{k}} \tag{2.51}$$

The method suggests that after choosing a value of $T_{i_b,j+1}^0$, Δt_j^0 is recalculated from (2.51). The value of Δt_j^0 is used to calculate $T_{i,j+1}^0, i=0,i_b$ using (2.47). Equation (2.48) is then tested for the accuracy for the selection of $T_{i_b,j+1}^0$. If (2.48) is not satisfied, $T_{i_b,j+1}$ is estimated again and a new Δt_j is obtained from (2.51), which in turn is used to calculate new values of $T_{i,j+1}, i=0,i_b$. This process is repeated until (2.48) is satisfied within a specified error. However, the way in which $T_{i_b,j+1}$ is selected was not elaborated on by the authors.

Gupta and Kumar [48], in a comparative study of numerical methods for moving boundary problems, tackle this point. They tried taking $T_{i_b,j+1}^0$ equal to $T_{i_b-1,j}$ as a suitable estimate; then kept on giving a small increment to it successively until a change of sign in the error in (2.48) is detected; then $T_{i_b,j+1}$ is interpolated either by the method of chords or the bisection method. They found that this procedure does not always work. In the problem they solved [48] the procedure fails after some time; the method of solving

(2.47) becomes unstable and the error in (2.48) behaves in an irregular manner. Therefore, they adopted the following approach: instead of estimating $T_{i_b,j+1}$, they estimated Δt_j^0 from the following:

$$\Delta t_j^0 = \Delta t_{j-1} + \left(\Delta t_{j-1} - \Delta t_{j-2}\right) + \varepsilon \tag{2.52}$$

where ε is small. $T^0_{i_b,j+1}$ is calculated from (2.49); values of $T^0_{i,j+1}, i = 0, i_b$ are then determined from (2.47); (2.48) is checked to ensure that it is within the specified error; if not, Δt_j is incremented by approximately 0.2% and the process is repeated until a change of sign in the error occurs; then the value of Δt_j is interpolated by the method of bisection.

2.5.3. The Modified Variable Time Step (MVTS) method

In this method [48] the same set of simultaneous equations, (2.47) and (2.48), are to be solved as in the EDG and GK methods. The interface condition (2.39) is written as:

$$\Delta t_j^{k+1} = \frac{\Delta x^2}{1 - T^k_{i_b,j+1}} \tag{2.53}$$

Choosing Δt_j^0 equal to Δt_{j-1} initially, the temperature distribution $T^0_{i,j+1}, i = 0, i_b$ is obtained using (2.47) and (2.48). Using the value of $T^0_{i_b,j+1}$ in (2.53) the new estimate Δt_j^1 is obtained. The process is repeated until the specified error in Δt_j is reached.

The first time step Δt_0 is calculated by combining both the boundary condition (2.37) and the interface condition (2.39). The finite difference of (2.37) gives:

$$\frac{1}{\Delta x}\left(T_{1,1} - T_{0,1}\right) = hT_{0,1} + \psi \tag{2.54}$$

and the finite difference of (2.39) gives:

$$\frac{1}{\Delta x}\left(T_{0,1} - T_{1,1}\right) = \frac{\Delta x}{\Delta t_0} \tag{2.55}$$

Combining (2.54), (2.55) and remembering that the moving boundary is at the node (1,1) and therefore $T_{1,1} = 1$, we obtain:

$$\Delta t_0 = \frac{\Delta x(1 + h\Delta x)}{(h + \psi)} \tag{2.56}$$

In sections 2.5.1, 2.5.2 and 2.5.3 the computational procedure, of passing from one time step to another, for each method is explained. At the beginning of computation where the moving boundary is moving from $x = 0$ to $x = \Delta x$, using any of the EDG, GK and MVTS methods, the time step Δt_0 is calculated from (2.56) without iterations and the temperature at $x = 0$ is calculated using either (2.54) or (2.55).

2.6. The enthalpy formulation techniques

These techniques are based on the relationships which exist between the different thermodynamic properties of the material such as enthalpy, heat capacity and latent heat of fusion. These methods are strictly limited to melting (freezing) problems, where the enthalpy formulation automatically satisfies the jump in heat flux across an isothermal moving boundary. For instance, these methods cannot be used for implicit moving boundary problems, where the moving boundary is present due to physical phenomena other than a phase change [52].

2.6.1. Post-iterative (Isothermal)

This is probably the simplest method [53,54]. For each node the usual finite difference or finite element equations are used. For the nodes where the phase change is occurring the temperature is set back to the temperature of transition and the equivalent energy is added to an enthalpy budget for that node. Once the budget is equal to the latent heat for the volume associated with that node, the temperature is allowed to vary according to the Fourier conduction equation.

2.6.2. Post-iterative (Mushy)

This is similar to the isothermal case except that a mushy range (see appendix A) is also taken into consideration [55,56]. The latent heat is released over a temperature range, and is assumed to be a function of temperature (generally linear) in that range. The procedure is as in 2.6.1 except that the temperature will be set to:

$$T = T_2 - \left(\frac{\Delta H}{L}\right)\Delta T \tag{2.57}$$

where T_2 is the temperature at the top end of the mushy range; ΔH, L and ΔT are heat in enthalpy budget, latent heat of fusion and difference of temperature of the mushy range. Again both implicit and explicit finite difference, as well as finite element formulations can be used for the partial differential equations.

2.6.3. Apparent heat capacity

In this technique [57-59] the latent heat effect is considered by increasing the heat capacity in the phase change temperature range. For example, if a linear release of the latent heat in the range is assumed, the apparent capacity is

$$c_{ap} = \begin{cases} c & T < T_1 \quad \text{Solid region} \\ c_{mu} & T_1 \le T \le T_2 \quad \text{Mushy region} \\ c & T > T_2 \quad \text{Liquid region} \end{cases} \tag{2.58}$$

where $c_{mu} = c + (L/\Delta T)$. The apparent capacities are calculated using the temperatures at the nodes, the resulting finite difference or finite element equations are based on these apparent capacities.

2.6.4. Enthalpy method

In the enthalpy method [60 – 64], the heat conduction equation is written as:

$$\rho \frac{\partial H}{\partial t} = K \nabla^2 T \tag{2.59}$$

The enthalpy function of temperature is given by:

$$H(T) = \begin{cases} cT & T < T_1 \quad \text{Solid region} \\ cT + \dfrac{L(T - T_1)}{T_2 - T_1} & T_1 \le T \le T_2 \quad \text{Mushy region} \\ cT + L & T > T_2 \quad \text{Liquid region} \end{cases} \tag{2.60}$$

Hunter and Kuttler [65] combined the enthalpy method with Kirchhoff and a co-ordinate transformation to put moving boundary problems in a simple form; where, all the non-linearity in physical properties of the material can be concentrated in the functional relation between enthalpy and temperature.

2.6.5. Pham's method

This technique [59] combined the enthalpy and the heat capacity methods. Pham uses a three time level finite difference scheme. For the one–dimensional heat equation, this can be written as:

$$\rho c_{i,j+1}\left(T_{i,j+2}-T_{i,j}\right)=\left(\frac{2K\Delta t}{3\Delta x^2}\right)\Big\{\ T_{i+1,j+2}+T_{i+1,j+1}+ \\ T_{i+1,j}+T_{i-1,j+2}+T_{i-1,j+1}+T_{i-1,j}-2T_{i,j+2}-2T_{i,j+1}-2T_{i,j}\ \Big\} \tag{2.61}$$

where $c_{i,j+1}$ is calculated as:

$$c_{i,j+1}=\frac{\Delta H^*}{\left(T^*-T_{i,j}\right)} \tag{2.62}$$

where

$$\Delta H^*=\left(\frac{2K\Delta t}{\Delta x^2}\right)\left(T_{i+1,j+1}+T_{i-1,j+1}-2T_{i,j+1}\right) \tag{2.63}$$

and

$$T^*=f_T\left\{f_H\left(T_{i,j}\right)+\Delta H^*\right\} \tag{2.64}$$

where f_T is the function to calculate temperature from enthalpy and f_H is the function to calculate enthalpy from temperature. When $T_{i,j+2}$ has been estimated using (2.61), it is corrected from

$$T_{i,j+2}(\text{corrected})=f_T\left\{f_H\left(T_{i,j}\right)+c_{i,j+1}\left(T_{i,j+2}-T_{i,j}\right)\right\} \tag{2.65}$$

If $c_{i,j+1}$ was underestimated causing $T_{i,j+2}$ to pass the freezing temperature, (2.65) will reset $T_{i,j+2}$ back to the freezing range.

2.6.6. Effective heat capacity

In this technique [66], a temperature profile is assumed between the nodes; instead of calculating an apparent capacity based on the nodal temperature, an effective capacity is calculated based on the integral through the nodal volume and the assumed temperature profile. The integral to be evaluated is:

$$c_{eff}=\frac{1}{V}\int c_{ap}\,dV \tag{2.66}$$

By evaluating (2.66) at each time step, it is insured that the latent heat effect is considered at all time steps, even if the mushy range happens to fall between two nodes. Similarly to previous methods, the explicit or implicit finite difference as well as finite element formulation can be utilised.

2.6.7. Tacke's method

This method [64] is based on an explicit finite difference formulation and it is identical to the enthalpy method except for the element that contains the phase front. If an element i_b contains the phase interface, the interface position is calculated using the enthalpy H_{i_b} of the element i_b

$$H_{i_b} = L(1-\xi)+\frac{c}{2}(T_{eR}-T_b)(1-\xi)-\frac{c}{2}(T_b-T_{eL})\xi \tag{2.67}$$

where T_{eR} and T_{eL} is the temperature of the element to the right and left of the MB, respectively; ξ is the fraction of nodal volume which is solid. Once the interface position is known, the heat flux density into and out of the element can be defined as:

$$\Phi_{Left} = \frac{K_{Left}\left(T_b - T_{i_b-1}\right)}{\Delta x\left(\frac{1}{2}+\xi\right)}, \tag{2.68}$$

$$\Phi_{Right} = \frac{K_{Right}\left(T_{i_b+1} - T_b\right)}{\Delta x\left(\frac{3}{2}-\xi\right)} \tag{2.69}$$

If the interface passes from one element to another in a time step, the enthalpies are corrected to account for the errors in the assumed heat flux densities.

2.6.8. Blanchard and Fremond's method

Blanchard and Fremond [67] use the freezing index and the homograph approximation. The homograph approximation is an approximation of the variation of the liquid content μ with temperature. It is expressed as:

$$\mu = \frac{T-T_b}{2\left(\eta+|T-T_b|\right)}+1 \tag{2.70}$$

where η is a constant in the homograph approximation. If T is greater than T_b, μ from (2.70) approaches unity, while if T is less than T_b, μ approaches zero. The specific energy E of the solidifying or melting material is given by:

$$E = cT + L\mu \tag{2.71}$$

The freezing index u is defined as:

$$u = K\int_0^t T\,dt \qquad \text{or} \qquad T = \frac{1}{K}\frac{du}{dt} \tag{2.72}$$

The energy equation can be rewritten, using (2.71) and (2.72), as:

$$E(t) - E(0) = \frac{1}{\rho}\nabla^2 u \tag{2.73}$$

In this method, equation (2.73) is solved instead of the original Fourier equation. Both the explicit and implicit finite difference as well as finite element formulation can be utilised.

2.7. Conclusion

Research into numerical solutions for moving boundary problems deals mainly with developing new schemes or extending the existing ones to more general applications of these problems. However, few attempts have been made to evaluate the best method. Fox [68], in his article "*what are the best methods*", concluded that considerable research is required to answer this question. Furzeland [52], compared the accuracy, computational efficiency, ease of programming and the ability to handle complex problems. He concluded that there is no method where all the criteria are fulfilled. Poirier and Salcudean [69] also compared different numerical schemes for different problems; however, they concluded that the best numerical method for ice–melting, for example, is not necessarily the best for metal–melting. The choice of the numerical scheme depends not only on the nature of the problem but also on the priorities set by the user on: accuracy, computational cost and ease of programming.

The accuracy of numerical schemes depend largely on the accuracy of the recurrence formulae such as the classical Euler explicit, the Crank–Nicolson and the fully implicit equations. On the other hand, the refinement of the mesh size also increases the accuracy of the solution, but the storage and execution time increases. However, from the point of view of programming ease and computing economy, variable time step methods, with implicit finite difference equations, are very attractive in that they avoid complications and reduce the storage and memory size requirements.

2.8. References

1. S.G. Bankoff, *Adv. Chem. Eng.* **5**, 75–150 (1964).
2. J.C. Muehlbauer and J.E. Sunderland, *Appl. Mech. Rev.* **8**, 951–959 (1965).
3. B.A. Boley, *Nucl. Eng. Des.* **18**, 377–399 (1972).
4. L.I. Rubinstein, *Trans. Math. Monographs* **27**, Am. Math. Soc., Providence, R.I. (1971).
5. R.M. Furzeland, TR/76, Dept. Math., Brunel University (1977).
6. K.H. Hoffman (ed.), Frei Universitat (Berlin) Fachbereich Mathematik Vol I-III (1977).
7. J. Crank, TR 81004, Royal Military College of Science (1981).
8. L. Fox, in: *A survey of numerical methods for partial differential equations*, I. Gladwell and R. Wait (eds.), Clarendon Press, Oxford (1979).
9. J.R. Ockendon and W.R. Hodgkins (eds.), *Moving boundary problems in heat flow and diffusion*, Clarendon Press, Oxford (1975).
10. D.G. Wilson and P.T. Boggs (eds.), *Moving boundary problems*, Academic Press, New York (1978).
11. R.M. Furzeland, *Bull. Inst. Math. Appl.* **15**, 172–176 (1979).
12. E. Magenes (ed.), *Free boundary problems vols I and II*, Instituto Nazionale di Alta Matematica Francesco Severi, Rome (1980).
13. J. Albrecht, L. Collatz and K.H. Hoffman (eds.), Proceedings of the Oberwolfach conference on free boundary problems (1980).
14. A. Fasano and M. Primicerio, Free boundary problems–theory and applications Vol I and II, *Research Notes in Mathematics* **78** and **79**, Pitman, London (1983).
15. T.R. Goodman, *Trans. ASME* **80**, 335–342 (1958).
16. H.S. Carslaw and J.C. Jaeger (eds.), *Conduction of heat in solids*, Clarendon Press, Oxford (1959).
17. T.R. Goodman and J.J. Shea, *J. Appl. Mech.* **27**, 16–24 (1960).
18. G. Poots, *Int. J. Heat Mass Transfer* **5**, 525–531 (1962).
19. T.J. Lardner and F.V. Pohle, *J. Appl. Mech.* **28**, 310–312 (1961).
20. B.A. Boley and L. Estenssoro, *Mech. Res. Comm.* **4**, 271–279 (1977).
21. M. Imber and P.N.S. Huang, *Int. J. Heat Mass Transfer* **16**, 1951–1954 (1973).
22. G.E. Bell, *Int. J. Heat Mass Transfer* **21**, 1357–1361 (1978).
23. B.A. Boley, *Int. J. Heat Mass Transfer* **16**, 2035–2041 (1973).
24. D. Langford, *Int. J. Heat Mass Transfer* **16**, 2424–2428 (1973).
25. B. Noble, In Ockendon and Hodgkins (ref. 9), 208–209 (1975).
26. G.E. Bell, *Int. J. Heat Mass Transfer* **22**, 1681–1686 (1979).
27. J. Crank (ed.), *The mathematics of diffusion*, Clarendon Press, Oxford (1975).
28. W.D. Murray and F. Landis, *Trans. ASME J. Heat Transfer* **81**, 106–112 (1959).

29. R.M. Furzeland, TR/77, Dept. Math., Brunel University (1977).
30. L.W. Ehrlich, *J. Ass. Comput. Math.* **5**, 161–176 (1958).
31. J.C. Y Koh, J.F. Price and R. Colony, *Prog. Heat Mass Transfer* **2**, 225–247 (1969).
32. T. Saitoh, *Mem. Sagami Inst. Technol.* **6**, 1–14 (1972).
33. R. Bonnerot and P. Jamet, *Int. J. Numer. Meth. Eng.* **8**, 811–220 (1974).
34. G.H. Meyer, TR/62, Dept. Math., Brunel University (1976).
35. M. Ciment and R.B. Guenther, *Appl. Anal.* **4**, 39–62 (1974).
36. M. Ciment and R.A. Sweet, *J. Comput. Phys.* **12**, 513–525 (1973).
37. J. Douglas and J.M. Gallie, *Duke Math. J.* **22**, 557–571 (1955).
38. A. Huber, *Z. Angew. Math. Mech.* **19**, 1–21 (1939).
39. A. Fasano and M. Primicerio, *Z. Angew. Math. Mech.* **53**, 341–348 (1973).
40. A. Lazaridis, *Int. J. Heat Mass Transfer* **13**, 1459–1477 (1970).
41. W.L. Heitz and J.W. Westwater, *Int. J. Heat Mass Transfer* **13**, 1371–5 (1970).
42. L.C. Tien and S.W. Churchill, *A. I. Ch. E. Jl.* **11**, 790–793 (1965).
43. J. Crank and R.S. Gupta, *J. Inst. Math. Appl.* **10**, 296–304 (1972).
44. R.S. Gupta, Ph.D. Thesis, Brunel University (1973).
45. R.S. Gupta, *Comput. Meth. Appl. Mech. Eng.* **4**, 143–152 (1974).
46. R.S. Gupta and D. Kumar, *Comput. Meth. Appl. Mech. Eng.* **23**, 101–109 (1974).
47. J.S. Goodling and M.S. Khader, *J. Heat Transfer* **96**, 114–115 (1974).
48. R.S. Gupta and D. Kumar, *Int. J. Heat Mass Transfer* **24**, 251–259 (1981).
49. R.S. Gupta and D. Kumar, *Comput. Meth. Appl. Mech. Eng.* **29**, 233–239 (1981).
50. E.B. Hansen and P. Hougaard, *J. Inst. Math. Appl.* **13**, 385–398 (1974).
51. H.O. Dahmardah and D.F. Mayers, *IMA J. Numer. Analysis* **3**, 81–85 (1973).
52. R.M. Furzeland, *J. Inst. Math. Appl.* **26**, 411–429 (1980).
53. G.M. Dusinberre, *Trans. ASME* **67**, 703 (1945).
54. P.C. Doherty, *U.S. Geological Survey Computer Contribution* **4**, (1970).
55. M. Salcudean and A. Mashaie, Rep. II Mech. Eng. Dept., University of Ottawa (1983).
56. M. Salcudean and Z. Abdullah, Proceedings of the VIII International Heat Transfer Conference and Exhibition, San Francisco CA, 459–464 (1986).
57. H.T. Hashemi and C.M. Sliepcevich, *Chem. Eng. Prog. Symp. Series* **63**, 34–41 (1967).
58. G. Comini, S. Del Guidice, R.W. Lewis and O.C. Zienkiewicz, *Int. J. Numer. Meth. Eng.* **8**, 613–624 (1974).
59. Q.T. Pham, *Int. J. Heat Mass Transfer* **28**, 2079–2084 (1985).
60. G.H. Meyer, *SIAM J. Numer. Anal.* **10**, 522–538 (1973).

61. N. Shamsunder and E.M. Sparrow, *Tans. ASME J. Heat Transfer* **79**, 333–340 (1975).
62. A.B. Crowley, *Int. J. Heat Mass Transfer* **21**, 215–219 (1978).
63. G.E. Bell and A.S. Wood, *Int. J. Numer. Meth. Eng.* **19**, 1583–1592 (1983).
64. K.H. Tacke, *Int. J. Num. Meth. Eng.* **21**, 543–554 (1985).
65. L.W. Hunter and J.R. Kuttler, *Trans. ASME J. Heat Transfer* **111**, 239–242 (1989).
66. D. Poirier, MSc Thesis, University of Ottawa (1986).
67. D. Blanchard and D. Fremond, *Int. J. Numer. Meth. Eng.* **20**, 757–771 (1984).
68. L. Fox, in Ockendon and Hodgkins (ref. 9), 210–241 (1975).
69. D. Poirier and M. Salcudean, *Trans. ASME J. Heat Transfer* **110**, 333–340 (1988).

3 Computational Performance of Finite Difference Methods

Summary

Solutions of transient heat conduction problems are usually obtained utilising finite difference methods. These however depend on recurrence formulae such as the Crank-Nicolson, fully implicit and Euler. These and more recently developed equations, are tested on three different problems and their relative performances are compared using a Computational Performance Factor (*CPF*), which takes into account: the accuracy, cpu–time and computer memory requirement. Overall the explicit finite difference equations combined with the virtual sub–interval elimination technique is seen to offer higher performance compared to the implicit equations.

Nomenclature

a	Material Thickness	r_f	A fixed Fourier number (≤0.5)
AS	Array size ($=n_x \times n_t$)	t	Time variable
c	Specific heat	T	Temperature
CPF	Computational performance factor	x	Space variable
h	Heat transfer coefficient	α	Diffusivity
K	Thermal conductivity	ρ	Material density
MPD	Maximum percentage deviation	Δt	Time step
n_x	Total number of space elements	Δx	Space element
n_t	Total number of time steps		*Subscripts*
PD	Percentage deviation	i, j	Space / time step index
p	Number of virtual sub-time steps		*Superscript*
r	Fourier number	m	Virtual sub-time step index

3.1. Introduction

In heat transfer problems where an analytical solution is not possible: finite difference, finite element or boundary element methods have to be applied, to compute the temperature distribution in the space–time domain. However, these methods depend – for their implementation – on the recurrence formulae such as the Crank-Nicolson, fully implicit and Euler equations [1-5].

Solving one-dimensional heat conduction problems, requires the solution of the partial differential equation

$$\rho c \frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le a \quad , \quad 0 \le t \le \tau \tag{3.1}$$

with prescribed boundary conditions at $x=0$ and $x=a$, and known initial conditions. Analytical solutions of (3.1) are available in [6] for limited boundary and initial conditions. However, in most engineering problems, the boundary conditions are complex and with the wide availability of high speed digital computers, recourse is often made to numerical methods.

Using the implicit methods, the temperature distribution at each time step is determined by solving a system of n_x (if the total thickness a is divided into n_x grid points) simultaneous equations with n_x unknowns; whereas using the explicit methods the temperature at each grid point is determined directly from the temperature of the same and neighbouring grid points evaluated for the previous time step. Due to stability constraints associated with explicit equations, an algorithm using this approach may require a large array size if computations are to be performed for an extended time

period. On the other hand implicit methods do not have such constraints, which makes implicit methods more practical when the Fourier number is relatively large.

To avoid the large array size requirements and stability constraints associated with explicit equations, a Virtual Sub–Interval Elimination Technique (VSIET), similar to that used in moving boundary problems [7], is incorporated to maintain stability for any mesh size without increasing the array size requirements.

Most authors [8-11] compare different numerical methods in terms of accuracy (error) only, this however may not be a sufficient criteria to make a judgement on whether one method is better than another. For instance, if a method achieves an accuracy of 0.01% greater than the accuracy of another method but with twice the cpu-time and memory requirements, this may make the first method better when an error of 0.01% is crucial, but less attractive when the cpu-time is more important than a 0.01% error. Therefore the objective of this study is to compare the different methods in terms of a Computational Performance Factor (*CPF*), which takes into account the three main computational parameters (accuracy, cpu-time and computer memory requirements). In this comparative study, besides the well known Euler, Crank-Nicolson and fully implicit equations – the exponential equation [12] and the implicit weighted time step method [10,13] are also considered.

3.2. Methods of solution

If equation (3.1) with prescribed boundary conditions has to be solved using finite difference methods; then a is divided into n_x space sections of length $\Delta x = a/n_x$, and τ is divided into n_t time steps of length $\Delta t = \tau / n_t$.

Let $T_{i,j}$ be the temperature at $x_i = i\Delta x\,; i = 0, n_x$ at time $t_j = j\Delta x\,; j = 0, n_t$. To compute the temperature distribution at a time t_{j+1} from that of t_j, the following methods can be used:

(i) The Crank-Nicolson finite difference method [14]:

$$T_{i,j+1} = T_{i,j} + \frac{r}{2}\left\{\left(T_{i-1,j} - 2T_{i,j} + T_{i+1,j}\right) + \left(T_{i-1,j+1} - 2T_{i,j+1} + T_{i+1,j+1}\right)\right\} \tag{3.2}$$

where $r = \alpha \dfrac{\Delta t}{\Delta x^2}$ and $\alpha = K/(\rho c)$

(ii) The fully implicit finite difference method [15]:

$$T_{i,j} = (1-2r)T_{i,j+1} - r\left(T_{i-1,j+1} + T_{i+1,j+1}\right) \tag{3.3}$$

(iii) The weighted time step method [10,13]:

$$T_{i,j+1} = T_{i,j} \exp\left\{-\Delta t\, D_{i,j}\right\} \tag{3.4}$$

where the matrix $D_{i,j}$ is given by:

$$D_{i,j} = A_{i,k}^{-1} B_{k,j} \tag{3.5}$$

$A_{i,j}$ and $B_{i,j}$ are matrices defined in [10]

(iv) The Euler equation combined with VSIET:

$$T_{i,j+1} = T_{i,j}^{p} \tag{3.6}$$

where

$$T_{i,j}^{m} = (1-2r_c)T_{i,j}^{m-1} + r_c\left(T_{i-1,j}^{m-1} + T_{i+1,j}^{m-1}\right) \quad , \quad m = 1,p \tag{3.7}$$

$$p = \text{int}\left(\frac{r}{r_f}\right) + 1 \tag{3.8}$$

$$r_c = \frac{r}{p} \tag{3.9}$$

where the notation $T_{i,j}^{m}$ denotes the temperature at $x_i = i\Delta x$ at a virtual time $t_j^m = \left(j + \frac{m}{p}\right)\Delta t$; "int (y)" stipulates the largest integer less than y; r_f is a fixed value of Fourier number ($r_f \leq 0.5$).

(v) The exponential equation combined with VSIET:

$$T_{i,j+1} = T_{i,j}^{p}$$

where

$$T_{i,j}^{m} = T_{i,j}^{m-1} \exp\left\{-r_c \frac{2T_{i,j}^{m-1} - \left(T_{i-1,j}^{m-1} + T_{i+1,j}^{m-1}\right)}{T_{i,j}^{m-1}}\right\} \quad , \quad m = 1,p \tag{3.10}$$

Using the implicit methods, the temperature distribution at each time step is calculated by solving a system of n_x equations. Each equation contains three unknowns, except the equations for the first and last nodes which contain only two unknowns (by combining the boundary condition and the Fourier equation). The system is tri-diagonal and its solution is readily obtained by using the Gauss elimination method.

The VSIET consists of sub-dividing the real time step into p (equation (3.8)) Virtual Sub-Steps (VSS). Computation of the temperature distribution at each VSS is computed

explicitly (temperatures at the boundary nodes are calculated using a dummy node outside the domain with fictitious temperature) from that of the previous VSS. Each time a new VSS is completed, the previous VSS is eliminated (for more details see [7]) and the temperature distribution at the last VSS of each real time step, is assigned to the corresponding time (equation (3.6)).

In order to compare their performances, these methods are applied to three different problems where analytical solutions exist. For the purpose of comparison, in all the test problems the array size was the same for all the methods. The Computational Performance Factor (*CPF*), which is used to compare the different methods, is defined as

$$CPF = \frac{(1/MPD)}{AS \times CPU} \tag{3.11}$$

where AS is the array size (i.e. total number of grid points, $AS = (n_x + 1)(n_t + 1)$), CPU is the cpu-time spent by the method and MPD is the Maximum Percentage Deviation from the analytical solution registered anywhere in the grid:

$$MPD = \max\left\{\left(\left|PD_{i,j}\right|, i = 0, n_x\right), j = 0, n_t\right\} \tag{3.12}$$

where $PD_{i,j}$ is the percentage deviation from the analytical value at the grid point (i, j) given by:

$$PD_{i,j} = 100 \times \left(\frac{T_{i,j}^{Nu} - T_{i,j}^{An}}{T_{i,j}^{An}} \right) \tag{3.13}$$

where the superscripts Nu and An refer to temperature calculated Numerically and Analytically, respectively.

3.3. Test problems

Problem A

This consists of one–dimensional transient heat conduction through a slab, which initially has a triangular temperature distribution, the surfaces are maintained at zero temperature at all times. The governing equations are:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le 1 \quad , \quad t > 0 \tag{3.14}$$

$$T(0,t) = T(1,t) = 0.0 \tag{3.15}$$

$$T(x,0)=\begin{cases}\psi x & 0\le x\le \frac{1}{2}\\ \psi(1-x) & \frac{1}{2}\le x\le 1\end{cases} \tag{3.16}$$

The analytical solution is given by [ref. 6, pp. 104]:

$$T(x,t)=\frac{4\psi}{\pi^2}\sum_{n=1}^{\infty}\left\{\frac{1}{n^2}\sin\left(\frac{n\pi}{2}\right)\sin(n\pi x)\exp\left[-\alpha t(n\pi)^2\right]\right\} \tag{3.17}$$

Problem A, when $\alpha=1.0$ and $\psi=2.0$, is solved by methods (i), (ii), (iii), (iv) and (v) (section 3.2) which will be referred to as CN, FI, WTS, EuVSS and ExVSS respectively. A comparison of numerical results is given in Tables 3.1, 3.2 (for $r=10.0$) and 3.3 (for $r=40.0$), using $n_x=n_t=20$. These are discussed in section 3.4.

Table 3.1 (Problem A): Comparison of solutions of FI and CN at the slab centre. Percentage deviation (PD [%]) from the analytical solution are also tabulated.

	FI		CN		Analytic
j	$T_{k,j}$	$PD_{k,j}$	$T_{k,j}$	$PD_{k,j}$	$T_{k,j}$
1	0.6888	7.0931	0.5637	-12.3592	0.6432
2	0.5330	7.4739	0.5440	9.6920	0.4959
3	0.4226	9.2703	0.3493	-9.6820	0.3868
4	0.3376	11.7588	0.3313	9.6626	0.3021
5	0.2705	14.5868	0.2117	-10.3175	0.2361
6	0.2169	17.6023	0.2038	10.4981	0.1844
7	0.1740	20.7438	0.1270	-11.8503	0.1441
8	0.1396	23.9879	0.1262	12.1066	0.1126
9	0.1120	27.3266	0.0755	-14.1270	0.0880
10	0.0899	30.7586	0.0787	14.4693	0.0687

Note: the node k=int(n_x/2) is the node at the slab centre; r=10.0

Table 3.2 (Problem A): Comparison of solutions of WTS, ExVSS and EuVSS. Percentage deviation (PD [%]) from the analytical solution are also tabulated.

	WTS		ExVSS		EuVSS		Analytic
j	$T_{k,j}$	$PD_{k,j}$	$T_{k,j}$	$PD_{k,j}$	$T_{k,j}$	$PD_{k,j}$	$T_{k,j}$
1	0.6807	5.8409	0.6451	0.2938	0.6439	0.1048	0.6432
2	0.5286	6.5863	0.4977	0.3548	0.4963	0.0799	0.4959
3	0.4188	8.2954	0.3884	0.4114	0.3869	0.0369	0.3868
4	0.3341	10.5830	0.3035	0.4635	0.3021	-0.0154	0.3021
5	0.2671	13.1487	0.2373	0.5142	0.2359	-0.0702	0.2361
6	0.2137	15.8633	0.1855	0.5648	0.1842	-0.1254	0.1844
7	0.1710	18.6766	0.1450	0.6153	0.1438	-0.1809	0.1441
8	0.1369	21.5710	0.1133	0.6655	0.1123	-0.2369	0.1126
9	0.1096	24.5405	0.0886	0.7155	0.0877	-0.2927	0.0880
10	0.0877	27.5844	0.0693	0.7657	0.0685	-0.3481	0.0687

Note: the node k=int(n_x/2) is the node at the slab centre; r=10.0

Table 3.3 (Problem A): Comparison of percentage deviations (PD [%]) from the analytical solution of FI, CN, WTS, ExVSS and EuVSS solution at different time step and at the slab centre position.

	FI	CN	WTS	ExVSS	EuVSS	Analytic
j	$PD_{k,j}$	$PD_{k,j}$	$PD_{k,j}$	$PD_{k,j}$	$PD_{k,j}$	$T_{k,j}$
1	39.3090	-56.5155	37.7851	0.4768	- 0.0783	0.3021
2	84.0123	86.2224	81.5747	0.6793	- 0.3618	0.1126
3	147.7231	-251.6444	143.1260	0.8830	- 0.6454	0.0420
4	234.5848	488.6956	226.4239	1.0873	- 0.9262	0.0156
5	352.1880	-1248.9570	338.4726	1.2902	- 1.2086	0.0058
6	511.2041	2843.0334	489.0356	1.4959	- 1.4895	0.0022
7	726.1621	-7017.5547	691.3118	1.7009	- 1.7678	0.0008
8	1016.7266	16983.9336	963.0533	1.9046	- 2.0483	0.0003
9	1409.4824	-42065.726	1328.1135	2.1124	- 2.3259	0.0001
10	1940.3745	104251.750	1818.5382	2.3182	- 2.0266	0.0000

Note: the node k=int(n_x/2) is the node at the slab centre; Fourier number r =40.

Problem B

The slab is initially at constant temperature T_0 throughout. At $t = 0$ the slab is heated by a constant heat flux F at one surface and thermally insulated on the other:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le 1 \quad , \quad t > 0 \tag{3.18}$$

$$\frac{\partial T}{\partial x} = 0 \quad , \quad x = 0.0 \quad , \quad t > 0 \tag{3.19}$$

$$-K \frac{\partial T}{\partial x} = F = \text{const.} \quad , \quad x = 1.0 \quad , \quad t > 0 \tag{3.20}$$

$$T(x,0) = T_0 \tag{3.21}$$

The analytical solution is given by [ref. 6, pp. 112]:

$$T(x,t) = T_0 + \frac{2F(\alpha t)^{1/2}}{K} \sum_{n=0}^{\infty} \left\{ \text{ierfc}\left(\frac{2n+1-x}{2(\alpha t)^{1/2}} \right) + \text{ierfc}\left(\frac{2n+1+x}{2(\alpha t)^{1/2}} \right) \right\} \tag{3.22}$$

Numerical results for problem B ($\alpha = 0.4 \times 10^{-5}$, $T_0 = 10.0$, $F = 5 \times 10^3$ and $K = 19.3$) using the FI, CN, EuVSS and ExVSS methods ($r = 30.0$ and $n_x = n_t = 20$) are shown in Table 3.4.

Table 3.4 (Problem B): Comparison of solutions of FI, CN, ExVSS and EuVSS at different time steps (j) for different space grid lines (i). Percentage deviation ($PD_{i,j}$ [%]) from the analytical solution are also tabulated. Fourier number r =30.

	FI		CN		ExVSS		EuVSS		Analytic
j	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$
$i=0$									
1	14.4641	36.1696	10.7593	1.2914	10.6469	0.2331	10.5894	-0.3084	10.6221
6	93.1772	10.8978	82.0227	-2.3780	84.5730	0.6574	83.9543	-0.0790	84.0207
10	173.7317	7.8045	163.4193	1.4054	161.7820	0.3894	161.0758	-0.0488	161.1544
14	255.4144	6.9377	245.2222	2.6704	239.5092	0.2784	238.7423	-0.0426	238.8441
20	378.0920	6.3782	367.9272	3.5182	356.0598	0.1793	355.2534	-0.0476	355.4226
$i=n_x=20$									
1	87.8593	-2.4401	67.0669	-25.528	90.8958	0.9317	90.0057	-0.0567	90.0568
6	219.9615	3.6004	210.9782	-0.6307	212.8857	0.2677	212.2566	-0.0286	212.3173
10	303.0122	4.2626	293.0459	0.8333	291.2397	0.2119	290.5393	-0.0291	290.6240
14	384.9236	4.4925	374.8130	1.7479	369.0127	0.1733	368.2537	-0.0327	368.3745
20	507.6238	4.6742	497.4917	2.5849	485.5896	0.1306	484.7622	-0.0400	484.9561

Problem C

The slab initially has an exponential temperature distribution. At $t=0$, one surface is exposed to a variable heat flux and the other surface loses heat by convection

$$\frac{\partial T}{\partial t}=\alpha\frac{\partial^2 T}{\partial x^2} \quad , \quad 0\le x\le 1 \quad , \quad t>0 \tag{3.23}$$

$$\frac{\partial T}{\partial x}=-\exp(\alpha t) \quad , \quad x=0.0 \quad , \quad t>0 \tag{3.24}$$

$$\frac{\partial T}{\partial x}=-T(x,t) \quad , \quad x=1.0 \quad , \quad t>0 \tag{3.25}$$

$$T(x,0)=\exp(-x) \tag{3.26}$$

The analytical solution is given by:

$$T(x,t)=\exp(\alpha t-x) \tag{3.27}$$

Numerical results for problem C, when $\alpha=2.5$, are shown in Table 3.5 using FI, CN, ExVSS and EuVSS when $r=25.0$ and $n_x=n_t=20$.

Table 3.5 (Problem C): Comparison of solutions of FI, CN, ExVSS and EuVSS at different time steps (j) for different space grid lines (i). Percentage deviation (PD [%]) from the analytical solution are also tabulated. Fourier number r =25.

	FI		CN		ExVSS		EuVSS		Analytic
j	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$	$PD_{i,j}$	$T_{i,j}$
$i=0$									
2	1.1475	1.2697	1.1436	0.9198	1.1334	0.0192	1.1333	0.0116	1.1331
7	1.5885	2.5626	1.5766	1.7914	1.5493	0.0330	1.5491	0.0161	1.5488
12	2.1862	3.2680	2.1650	2.2697	2.1178	0.0390	2.1174	0.0168	2.1170
16	2.8164	3.6098	2.7862	2.4985	2.7194	0.0426	2.7188	0.0176	2.7183
20	3.6240	3.8302	3.5827	2.6459	3.4919	0.0459	3.4910	0.0189	3.4903
$i=n_x=20$									
2	0.4223	1.3135	0.4202	0.7900	0.4169	0.0169	0.4169	0.0073	0.4169
7	0.5888	3.3294	0.5817	2.0856	0.5700	0.0444	0.5699	0.0129	0.5698
12	0.8144	4.5717	0.8016	2.9335	0.7792	0.0533	0.7789	0.0155	0.7788
16	1.0518	5.1783	1.0334	3.3397	1.0005	0.0528	1.0001	0.0140	1.0000
20	1.3555	5.5695	1.3302	3.5996	1.2847	0.0558	1.2842	0.0137	1.2840

3.4. Numerical results and discussion

Table 3.1 shows that for problem A, the CN method oscillates around the analytical solution, whilst the FI method does not exhibit such oscillations but it is less accurate than CN. Furthermore, it can be seen from Table 3.1 that the error accumulation in FI is greater than that in CN as time progresses. Table 3.2 shows that WTS is more accurate than FI but less accurate than CN; however, for ExVSS and EuVSS the maximum relative error is less than 1% whereas for the FI, CN and WTS methods the maximum errors are around 31%, 14% and 28% respectively. It is evident from Table 3.3 that the implicit methods become very inaccurate when the time step length is relatively large. Such errors may be seen from the engineering point of view as not acceptable, whereas the error using both EuVSS and ExVSS, which is around 2%, is still satisfactory even with very large Fourier numbers.

Similarly, numerical results for the problem B and C, which are shown in Tables 3.4 and 3.5 respectively, show that the explicit equations combined with VSIET out–perform the implicit schemes. The ExVSS method may exhibit computational difficulties (the argument of the exponential function may exceed the limit allowed for the compiler's intrinsic function) if a high heat flux is prescribed at the boundaries. Such difficulties can be avoided by approximating the exponential function, at the boundary nodes, by the first three terms of the Taylor series expansion.

In Table 3.6, a comparison of run times with and without vectorisation, for the different methods, is shown. Provided that the number of elements stays constant, the

cpu-time for the implicit methods remain approximately the same irrespective of the time step length, but the cpu-time for ExVSS and EuVSS increases as r increases. Table 3.6 also shows that the speed up due to vectorisation (*sdv*) for the explicit methods is greater than that of implicit methods, this however is due to the fact that the explicit methods – where there is no recurrence – are more suitable for vectorisation than implicit procedures.

Table 3.6: Comparison of cpu-time (x 10^2 s) without and with vectorisation for FI, CN, ExVSS and EuVSS for problem A.

	FI		CN		ExVSS		EuVSS	
r	*cpu*	*sdv*	*cpu*	*sdv*	*cpu*	*sdv*	*cpu*	*sdv*
00.25	0.4351	3.498	0.5580	4.161	0.3500	5.120	0.2127	6.147
00.35	0.4375	3.371	0.5674	4.244	0.5809	5.100	0.3038	5.899
00.50	0.4366	3.398	0.5648	4.212	0.5837	4.988	0.3030	5.872
00.80	0.4342	3.435	0.5620	4.163	0.8176	4.778	0.3926	5.723
01.00	0.4379	3.434	0.5634	4.146	0.8158	4.668	0.3942	5.738
02.00	0.4345	3.446	0.5658	4.241	1.5190	4.582	0.6688	5.597
03.00	0.4347	3.554	0.5609	4.208	2.2230	4.495	0.9399	5.411
06.00	0.4407	3.571	0.5598	4.241	4.3670	4.423	1.7550	5.375
09.00	0.4398	3.420	0.5600	4.124	6.2370	4.367	2.4750	5.327
12.00	0.4413	3.348	0.5570	4.179	8.3600	4.282	3.2900	5.335
15.00	0.4352	3.498	0.5635	4.190	10.210	4.256	4.0120	5.290
20.00	0.4412	3.402	0.5659	4.131	13.760	4.234	5.3630	5.289

Note – For each method, and for a given value of r, the first and second value corresponds to the cpu-time used by the method without and with vectorisation respectively.
cpu (vect.) = *cpu* (without vect.)/*sdv*, where *sdv* is the speed up due to vectorisation. *cpu* and *sdv* are specific to the IBM390/150VF, Computer Centre, University of Glasgow, U.K.

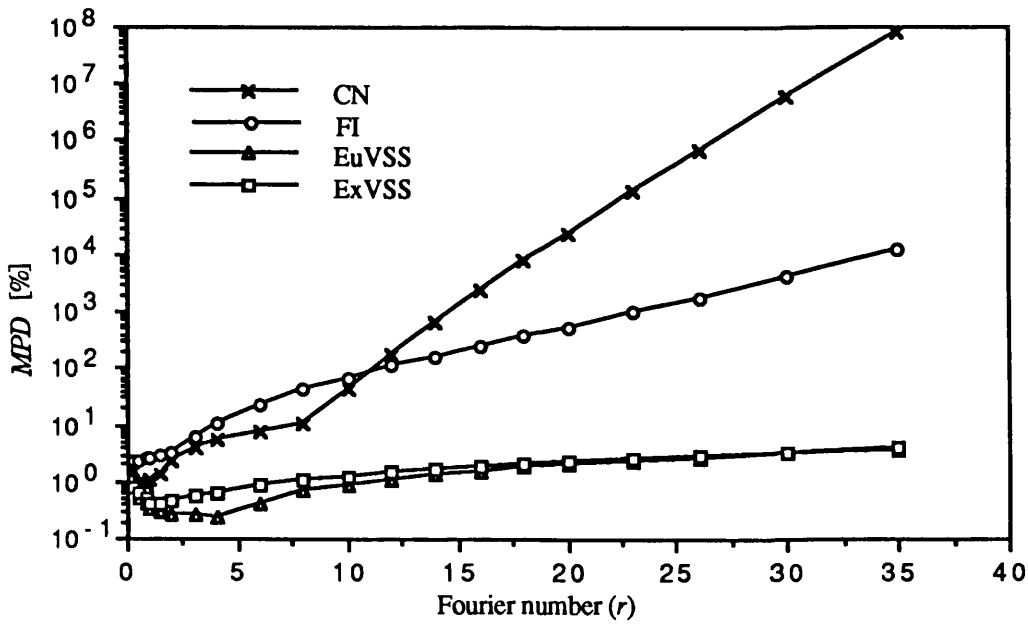

Figure 3.1: Variation of *MPD* with the Fourier number r for the different methods. Problem A; r_f=0.35, α=0.5, n_x=n_t=20 and λ=4.0

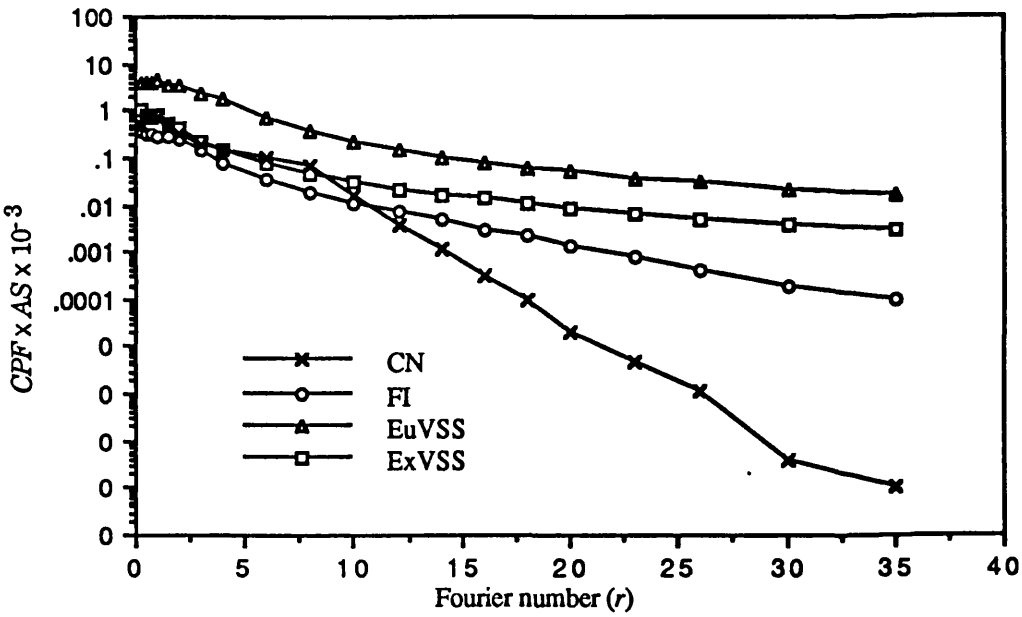

Figure 3.2: Comparison of computational performance factor *CPF* for the different methods. Problem A; r_f=0.35, α=0.5, n_x=n_t=20 and λ=4.0

To investigate the performance of the different methods over a large range of values of r, the three test problems are solved with r varying from 0.3 to 35. The performance of each method is assessed by plotting the maximum percentage deviation *MPD* and the computational performance factor *CPF* against r. Figure 3.1 shows that for problem A, the accuracy of both EuVSS and ExVSS is always superior to that achieved by CN and FI (this holds true also for the WTS method, because its accuracy lies between that of CN and FI with higher cpu-time). It also shows that when r >10, FI and CN give unacceptable errors.

Having regard to both the cpu-time and accuracy, the performances of the different methods are shown more clearly by Figure 3.2, where the *CPF* (the *CPF* is calculated using the cpu-time spent by the method using vectorisation) for EuVSS is the highest for all values of r used in the computation. ExVSS has a similar accuracy to EuVSS (Figure 3.1) but with a poorer *CPF*, this is mainly due to the greater cpu-time used for evaluating the exponential functions.

Figure 3.3 shows that for problem B, the *MPD* for FI and CN decreases when r is less than 3.0, this is due to the increase of temperature and consequently a decrease of the relative error. However, when the Fourier number becomes large, the difference between the numerical and analytical solutions becomes larger than the rise in temperature, which causes an increase in the relative error. Figure 3.4 shows again that EuVSS and ExVSS out–perform the implicit methods. It shows also that the *CPF* for FI and CN increases when r< 3 due to the decrease in the error shown in Figure 3.2, as the cpu-time remains approximately the same provided that the number of nodes remains fixed (see Table 3.6).

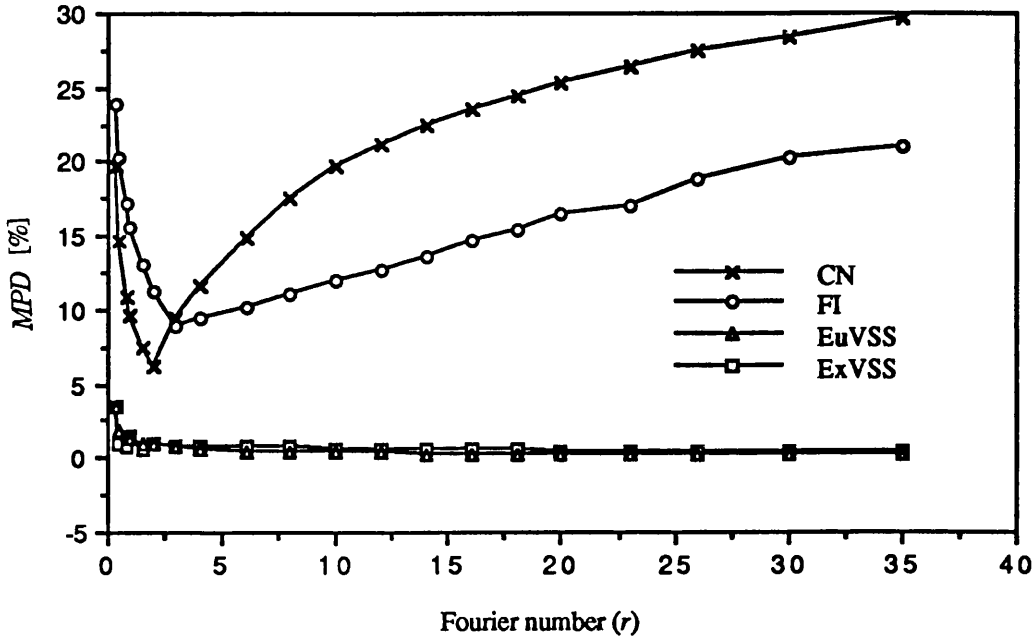

Figure 3.3: Variation of *MPD* with the Fourier number r for the different methods. Problem B; r_f=0.35, α =0.5x10^{-4}, F=3000, K=20, T_0=15.0 and n_x=n_t =20

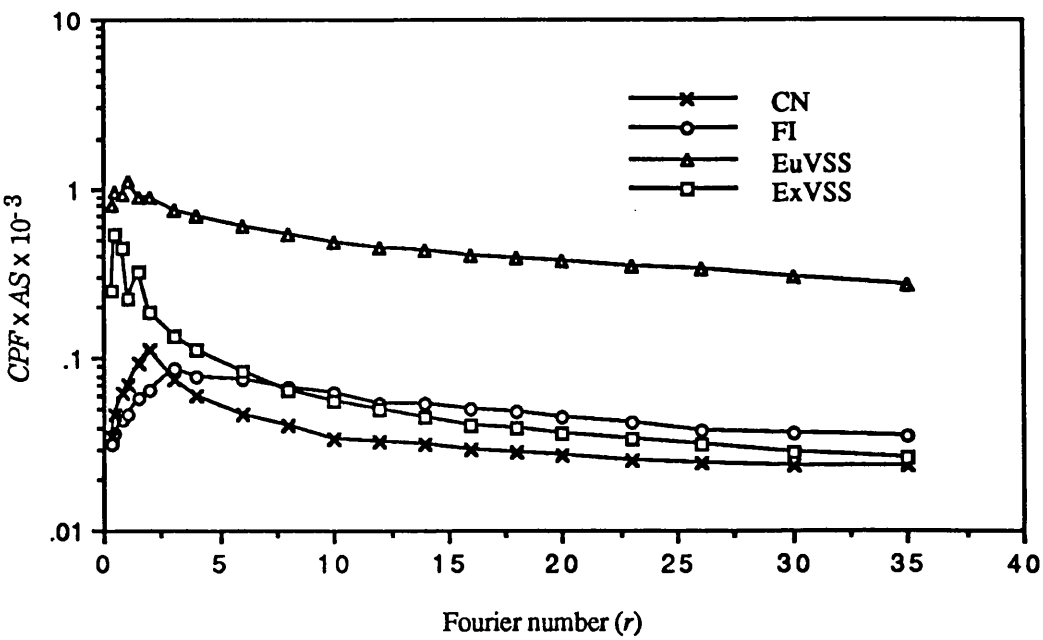

Figure 3.4: Comparison of computational performance factor *CPF* for the different methods. Problem B; r_f=0.35, α =0.5x10^{-4}, F=3000, K=20, T_0=15.0 and n_x=n_t =20

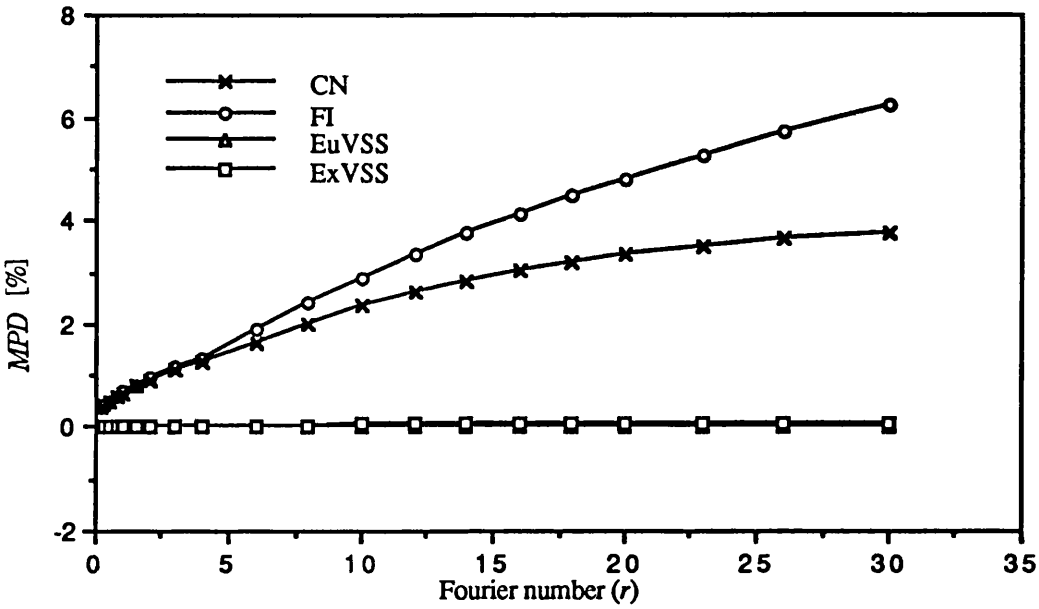

Figure 3.5: Variation of *MPD* with the Fourier number r for the different methods. Problem C; r_f=0.35, α=0.01 and n_x=n_t=20

The variation of *MPD* with r, for problem C, is shown in Figure 3.5. It shows that *MPD* for implicit methods increases with r, whereas for ExVSS and EuVSS it remains approximately steady over a large range of r. Figure 3.6 shows that, as in problems A and B, the computational performance factor of EuVSS and ExVSS methods is higher than that of the FI and CN methods for all values of r used in the computations.

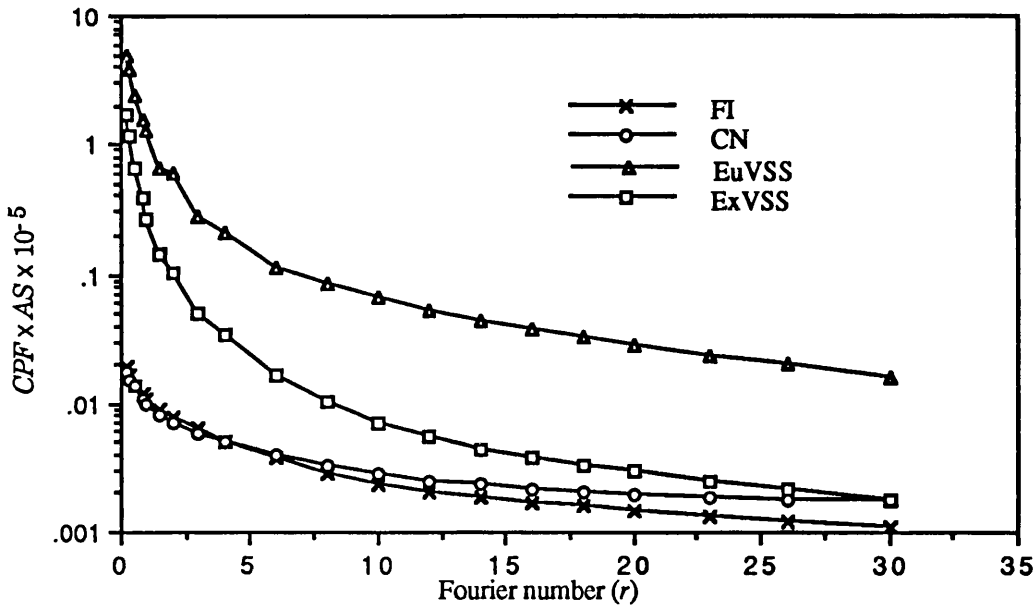

Figure 3.6: Comparison of computational performance factor *CPF* for the different methods. Problem C; r_f=0.35, α=0.01 and n_x=n_t=20

3.5. Conclusion

The numerical results for all three test problems show that EuVSS and ExVSS are more accurate than implicit schemes for any choice of Fourier number. Taking into consideration both the run time and accuracy, the explicit methods combined with the VSIET achieve higher computational performances than that of implicit methods for all the computations performed. The comparison of the different numerical schemes in terms of their computational performance factor, which measures unit accuracy per unit cpu-time per array element, seems to be a better tool for comparing competing numerical schemes than the usual comparison in terms of accuracy or cpu-time separately.

The numerical results for all three test problems show that the EuVSS method has the highest computational performance throughout. Despite the good accuracy of ExVSS, its *CPF* sometimes approaches that of the implicit methods, this is due mainly to the time spent in evaluating the exponential functions. However, the performance of ExVSS can be considerably increased if the following equation is used instead of (3.10):

$$T_{i,j}^{m} = T_{i,j}^{m-1} - r_c\left(2T_{i,j}^{m-1} - T_{i-1,j}^{m-1} - T_{i+1,j}^{m-1}\right) + \frac{(r_c)^2}{2T_{i,j}^{m-1}}\left(2T_{i,j}^{m-1} - T_{i-1,j}^{m-1} - T_{i+1,j}^{m-1}\right)^2 \qquad (3.28)$$

Equation (3.28) is obtained by approximating the exponential in (3.10) by the first three terms of the series expansion $\left(\exp(y) = 1 + y + \frac{1}{2}y^2\right)$. This permits considerable reduction in cpu-time, without loss of accuracy, this results in a higher *CPF*.

This study has shown that a numerical scheme based on an improved algorithm and using simple finite difference equations, may achieve superior computational performance to one based on more complex equations. Frequently, increasing the complexity results in much higher cpu-time for only a small improvement in accuracy.

3.6. References

1. J.M. Bettencourt, O.C. Zienkiewicz and G. Cantin, *Int. J. Numer. Meth. Eng.* **17**, 931-938 (1981).
2. O.C. Zienkiewicz and Parakeh, *Int. J. Numer. Meth. Eng.* **2**, 66-71 (1970).
3. D.W. Nicolson, *Acta Mechanica* **38**, 191-198 (1981).
4. R.W. Lewis and K. Morgan (eds.), *Numerical Methods in Thermal Problems*, Pineridge Press, Swansea (1985).
5. Z. Zlatev and G.P. Thomson, *Int. J. Numer. Meth. Eng.* **14**, 1051-1061 (1979).
6. H.C. Carslaw and J.C. Jaeger (eds.), *Conduction of Heat in Solids* , Clarendon press, Oxford (1959).
7. M. Zerroukat and C.R. Chatwin, *Int. J. Numer. Meth. Eng.* **35**, 1503-1520 (1992).

8. D. Poirier and M. Salcudean, *Trans. ASME J. Heat Transfer* **110**, 562-570 (1988).
9. R.M. Furzeland, *J. Inst. Maths. Appl.* **26**, 411-429 (1980).
10. C. Haberland and A. Lahrmann, *Int. J. Numer. Meth. Eng.* **25**, 593-609 (1988).
11. M.C. Bhattacharya and M.G. Davies, *Int. J. Numer. Meth. Eng.* **24**, 1317-1331 (1987).
12. M.C. Bhattacharya, *Int. J. Numer. Meth. Eng.* **21**, 239-265 (1985).
13. C. Haberland and A. Lahrmann, in: R.W. Lewis and K. Morgan (eds.), *Numerical Methods in Thermal Problems Vol. V*, Pineridge press, Swansea, (1987).
14. J. Crank and P. Nicolson, *Proc. Camb. Phil. Soci.* **43**, 50-67 (1947).
15. G.D. Smith (ed.), *Numerical Solution of Partial Differential Equations*, Clarendon press, Oxford (1978).

4 The Explicit Variable Time Step Method for Moving Boundary Problems

Summary

Numerical solution of heat conduction problems with phase changes benefits from the application of variable time step methods when the behaviour of the moving boundary is not known *a priori*. Due to convergence and stability constraints only implicit finite difference equations have been used with these methods. Implicit methods show a significant loss of accuracy and exhibit convergence difficulties when used for relatively slow or rapid moving–boundary problems. To overcome these problems an improved explicit variable time-step method, which combines the explicit exponential finite difference equation, a variable time-step grid network and virtual unequal space-increments around the moving boundary, is presented and tested for both a solidification and a melting problem. A virtual sub-interval time-step elimination technique is incorporated to insure that stability is automatically maintained for any mesh size. Unlike the implicit variable time step methods, the accuracy of the resulting method is not affected by the velocity of the moving boundary. For both test problems numerical results are in better agreement with known analytical solutions than results predicted by other numerical methods.

Nomenclature

a	Slab thickness	α	Diffusivity
F	Heat flux input	ε_{min}	Time-step error allowed to stop
h	Heat transfer coefficient		iteration
K	Thermal conductivity	λ	Latent heat of fusion/unit volume
M	Liquid/solid interface position		*Subscripts*
N	Total number of space elements	i, j	Space/time indices
PD_t	Percentage deviation for time	i_b	The node index holding the
PD_{ST}	Percentage deviation for surface		moving boundary
	temperature	l	Liquid
r_{cl}	Corrected Fourier number in the	m	Virtual sub-time step index in the
	liquid region		liquid region
r_{cs}	Corrected Fourier number in the	n	Virtual sub-time step index in the
	solid region		solid region
r_f	Fixed Fourier number (≤ 0.5)	p	Number of virtual sub-time steps
T	Temperature		in the liquid region
t	Time variable	q	Number of virtual sub-time steps
T_b	Temperature at the moving		in the solid region
	boundary	s	Solid
T_0	Temperature of the fluid in		*Superscript*
	contact the slab	k	Iteration index
x	Space variable		

4.1. Introduction

Several numerical methods are available [1] for solving moving boundary problems. According to Gupta [2] these may be classified into two categories:

(i) fixed grid methods,

(ii) variable grid methods.

In the first category the space and time dimensions are divided into step lengths of equal magnitude. For these methods the Euler explicit and implicit finite difference equations can be utilised; when the explicit finite difference equation is used the mesh size must be carefully chosen to avoid problems with convergence and stability, this method requires considerable memory size if computations are performed for large periods of time. Although the implicit finite difference equations have less demanding stability requirements, fixed grid methods either using the explicit or the implicit difference equations, break down when the moving boundary jumps a distance larger than a single space increment during a single time step [3]; in other words: these methods

are inadequate for heat transfer processes with phase changes where the moving boundary may have an instantaneously high velocity during the process.

Using variable grid methods only one dimension, either time or space, is divided into equal sized increments – the other being variable. In the Murray and Landis method [4]: the time increment is kept constant and the liquid and solid regions are at any instant in time divided into a fixed number of equally spaced intervals, however, the size of the intervals will increase or decrease as the proportions of solid/liquid alters; this kind of subdivision presents difficulties in commencing the computation, especially when the liquid or solidified depth is very small (for example laser surface melting). Douglas and Gallie [5] suggest in their methods that the thickness is to be divided into N equally spaced intervals of length Δx; the time step Δt, which is variable, is the time taken for the moving boundary to move Δx. Due to stability problems only the implicit difference equations have been used so far in these methods, where at each time step a set of $(N+1)$ simultaneous linear equations must be solved. When the time step is relatively large, the Euler explicit and the implicit equations suffer from either: significant oscillations, or loss of accuracy [6]; in other words, the implicit variable time step method may not be sufficiently accurate to solve heat transfer problems where the liquid/solid interface has a relatively slow velocity.

It is clear that the choice of the numerical method for a specific problem depends not only on the nature of the heat transfer process itself, where *a priori* view is necessary, but also on the available computational platform. This chapter presents a numerical method which can be used to solve any one–dimensional moving boundary problem without stability problems or loss of accuracy irrespective of the velocity of the moving boundary. An additional benefit of the method is that it does not require a large amount of computer memory, this is due to the incorporation of the virtual sub-interval time step elimination technique, it may also be less costly in terms of cpu-time due to its high suitability for vectorisation.

The numerical method is described for a general heat transfer problem where a solid material melts under the effect of a variable heat flux prescribed at one boundary and loses heat by convection through the other. It is convenient to test the method by adapting this general case to solve two specific problems which have been dealt with by other authors:

A) a liquid maintained at its fusion temperature which solidifies by losing heat through its surface by convection [5,7],

B) a solid material at constant temperature which melts due to a variable heat flux input at one boundary [8,9].

These results are compared with known analytical solutions which are used as a reference standard.

One-dimensional heat transfer in a homogeneous material of diffusivity α is described by the Fourier conduction equation:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le a \quad , \quad 0 \le t \le \tau \tag{4.1}$$

Difference solutions of equation (4.1), which give improved accuracy, have been reported recently [10-16]; Bhattacharya [10-14] gives two improved explicit finite difference equations which are:

$$T_{i,j+1} = T_{i,j} \exp\left\{-r\left(\frac{2T_{i,j} - T_{i-1,j} - T_{i+1,j}}{T_{i,j}}\right)\right\} \tag{4.2}$$

$$T_{i,j+1} = T_{i,j} \exp\left\{-r\left(\arccos\left[\frac{T_{i-1,j} + T_{i+1,j}}{2T_{i,j}}\right]\right)^2\right\} \tag{4.3}$$

where $r = \alpha \Delta t / \Delta x^2$.

Equation (4.3) is an efficient approach when used for relatively slow heat transfer processes. But when the cooling rate is relatively high, which results in steep temperature gradients, hence, $\left|(T_{i-1,j} + T_{i+1,j})/(2T_{i,j})\right|$ may exceed unity. Numerical experiments shows that a scheme based on (4.3) have a higher probability of break down, than that based on (4.2), if used for heat transfer processes with higher cooling rates such as subsequent cooling after laser surface heating. This work concentrates on the linear exponential equation, given by (4.2), as it has less stability constraints than (4.3).

4.2. Mathematical formulation of two phase Stefan problems

Consider the one-dimensional general melting process where a piece of solid material of infinite transverse dimensions and thickness a, is subjected to a source of heat $F(t)$ at $x = 0$, and is in contact with a fluid at temperature T_0 at $x = a$ (Figure 4.1).

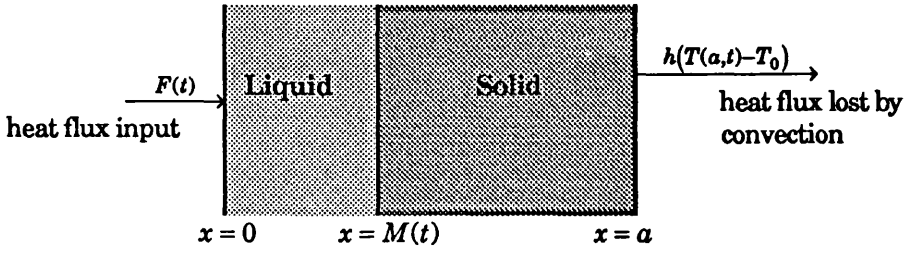

Figure 4.1: One-dimensional melting problem

Assuming that the thermal properties are constant but different for the solid and liquid regions, the process is described by the following equations:

$$\frac{\partial T_l}{\partial t} = \alpha_l \frac{\partial^2 T_l}{\partial x^2} \quad , \quad 0 \le x \le M(t) \quad , \quad t > 0 \tag{4.4}$$

$$\frac{\partial T_s}{\partial t} = \alpha_s \frac{\partial^2 T_s}{\partial x^2} \quad , \quad M(t) \le x \le a \quad , \quad t > 0 \tag{4.5}$$

$$\left.\begin{aligned} \lambda \frac{dM}{dt} &= K_s\left(\frac{\partial T_s}{\partial x}\right) - K_l\left(\frac{\partial T_l}{\partial x}\right) \\ T_l(x,t) &= T_s(x,t) = T_b \end{aligned}\right\} \quad , \quad x = M(t) \quad , \quad t > 0 \tag{4.6}$$

with the boundary conditions:

$$K_l\left(\frac{\partial T_l}{\partial x}\right) = -F(t) \quad , \quad x = 0 \quad , \quad t > 0 \tag{4.7}$$

$$-K_s\left(\frac{\partial T_s}{\partial x}\right) = h\left(T_s(x,t) - T_0\right) \quad , \quad x = a \quad , \quad t > 0 \tag{4.8}$$

and the initial conditions:

$$T_s(x,0) = T(x) \quad \text{and} \quad M(0) = 0 \tag{4.9}$$

4.3. Numerical computation scheme
(the Explicit Variable Time–Step, EVTS, method)

At the beginning of computation an arbitrary fixed value of Fourier number $r_f \le \frac{1}{2}$ is chosen. The material thickness a is divided into N intervals of equal length $\Delta x = a/N$, where $x_i = i\Delta x, i = 0,N$. The position of the moving boundary is indicated by i_b; hence, in the liquid the spatial index ranges from $i = 0$ to $i = i_b$ and in the solid $i = i_b + 1,N$. The time step Δt is of variable length and is determined as the time for the moving boundary to move a single space increment Δx; consequently, the moving boundary always coincides with a spatial grid line for each time step. The time step index varies from $j = 0$ to $j = N$ and increments at the same rate as i_b. Figure 4.2 illustrates the grid network.

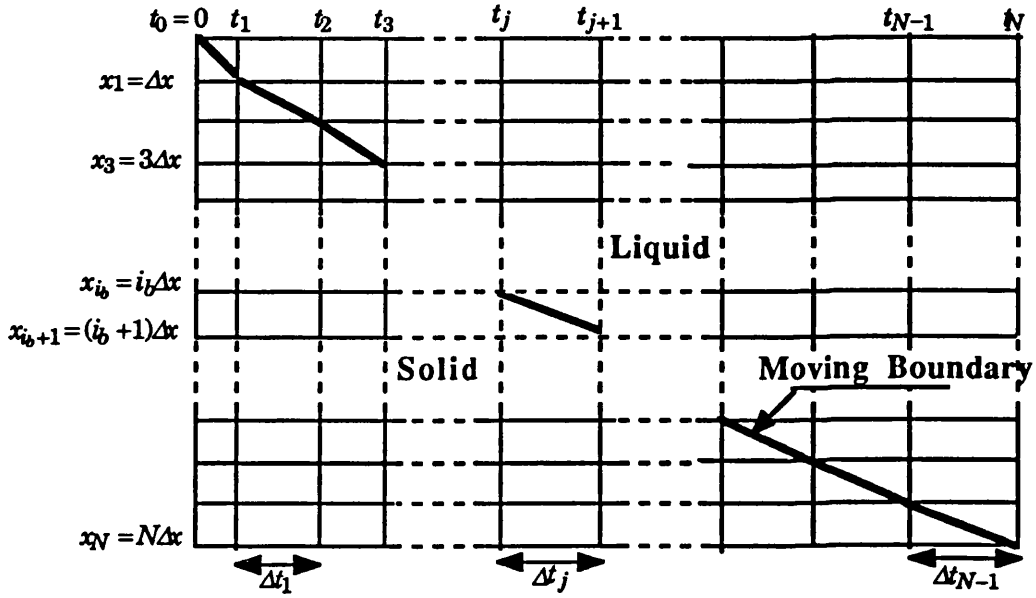

Figure 4.2: Variable time step grid network

A Virtual Sub-Interval time step Elimination Technique (VSIET) was incorporated to insure that the stability is automatically maintained for any mesh size without increasing the storage requirements. In moving boundary problems where two different phases exist simultaneously, the VSIET sub-divides the main time step increment into a number of Virtual Sub-time Steps (VSSs); namely, p in the liquid and q in the solid. In the liquid the VSS index m varies from $m=1$ to $m=p$; in the solid the VSS index n varies from $n=1$ to $n=q$. This technique is automatically applied to preserve the stability of the scheme for any situation which may occur during the computation process. These VSSs are referred to as *virtual*, as the temperature distribution at the most recent VSS is discarded (*eliminated*) each time the new VSS is completed. Only the temperature distribution at the main time step (*real*) is retained.

When the temperature distribution at each real time step is determined, the energy balance at the moving boundary (4.6) is checked. Each time the equality between the right hand and the left hand sides of (4.6) is not verified, Δt must be re-evaluated (4.30) as must p and q (4.14). This iterative process, where k is the iteration index, is repeated until the desired accuracy in Δt is reached (4.31).

The numerical procedure is illustrated for passing from a real time step to another (i.e. from t_j to t_{j+1}). The complete solution is achieved by repeating the procedure for each time step until the end of the heat transfer process. In particular the computation process, for the test problems in section 4.4, is stopped when the moving boundary has reached $x=a$, i.e. complete melting or complete solidification.

The notation $T_{i,j}^k$ denotes the temperature at the grid point of coordinates (x_i,t_j) given by:

$$\left(x_i = i\Delta x \quad , \quad t_j = \sum_{q=0}^{j-1} \Delta t_q \right) \tag{4.10}$$

$T^k_{i,j,m,p}$ denotes the variable T at x_i at the VSS (j,m,p) located at the virtual time $t^k_{j,m,p}$ given by:

$$t^k_{j,m,p} = \sum_{q=0}^{j-1} \Delta t_q + m\frac{\Delta t^k_j}{p^k_j} \tag{4.11}$$

where Δt_q is the time taken by the moving boundary to move from x_q to x_{q+1} and the superscript k is the iteration index.

Suppose the moving boundary is at node (i_b, j) and the time necessary to reach the node $(i_b + 1, j + 1)$ is Δt^k_j (as shown in Figure 4.3), where

$$r^k_j = \min(\alpha_l, \alpha_s) \times \left(\frac{\Delta t^k_j}{\Delta x^2} \right) > \frac{1}{2}; \tag{4.12}$$

then the problem is to compute temperature distribution at $t = t_{j+1}$ from that of $t = t_j$, as well as the time step Δt_j necessary for the moving boundary to move from the node (i_b, j) to $(i_b + 1, j + 1)$.

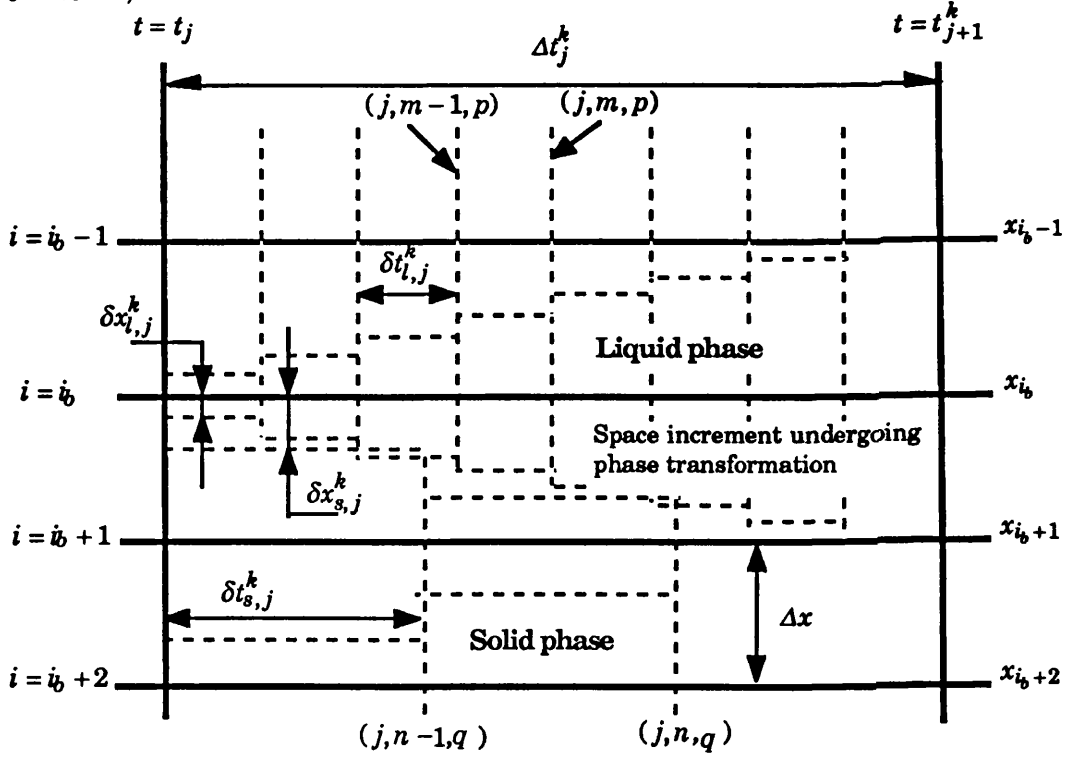

Figure 4.3: Descretisation around the moving boundary. Dashed and solid lines represent, virtual and real grid networks respectively

The VSIET is automatically applied to insure the stability of the scheme, when the Fourier number is greater than 0.5 (4.12), which is the limit of stability for explicit finite difference equations. Therefore Δt_j^k is divided in the liquid region into p_j^k virtual sub-intervals of length $\delta t_{l,j}^k$ and into q_j^k virtual sub-intervals of length $\delta t_{s,j}^k$ in the solid region (as shown in Figure 4.3). Defining $r_{l,j}^k$ and $r_{s,j}^k$ as:

$$r_{l,j}^k = \alpha_l \frac{\Delta t_j^k}{\Delta x^2} \quad , \quad r_{s,j}^k = \alpha_s \frac{\Delta t_j^k}{\Delta x^2} \tag{4.13}$$

and p_j^k and q_j^k must satisfy the following conditions:

$$p_j^k = \text{int}\left(\frac{r_{l,j}^k}{r_f}\right) + 1 \quad , \quad q_j^k = \text{int}\left(\frac{r_{s,j}^k}{r_f}\right) + 1 \tag{4.14}$$

where "int(y)" stipulates the smallest integer less than y.

The corrected Fourier number in the liquid $r_{cl,j}^k$ and in the solid region $r_{cs,j}^k$ are defined as:

$$r_{cl,j}^k = \frac{r_{l,j}^k}{p_j^k} \quad , \quad r_{cs,j}^k = \frac{\alpha_s}{\alpha_l}\frac{p_j^k}{q_j^k} r_{cl,j}^k \tag{4.15}$$

4.3.1. Computation of temperature distribution in the liquid region

In the liquid region, the temperatures distribution, $T_{i,j+1}^k, i = 0, i_b$, at time $t = t_{j+1}^k$ is computed from that at time $t = t_j$ in p_j^k stages. The temperature distribution in the liquid region at any VSS (j,m,p), where $\left\{j = 0, N-1; m = 1, p_j^k\right\}$, is calculated explicitly from that of previous VSS $(j,m-1,p)$ using:

$$T_{i,j,m,p}^k = T_{i,j,m-1,p}^k \exp\left(-r_{cl,j}^k \psi_{i,j,m-1,p}^k\right) \quad , \quad i = 0, i_b \tag{4.16}$$

where

$$\psi_{i,j,m,p}^k = \frac{2}{T_{i,j,m,p}^k}\left(T_{i,j,m,p}^k - T_{i+1,j,m,p}^k - \left(\frac{\Delta x}{K_l}\right)\overline{F}_{j,m,p}\right) \quad \text{for} \quad i = 0 \tag{4.17}$$

$$\psi_{i,j,m,p}^k = \frac{1}{T_{i,j,m,p}^k}\left(2T_{i,j,m,p}^k - T_{i+1,j,m,p}^k - T_{i-1,j,m,p}^k\right) \quad \text{for} \quad i = 1, i_b - 1 \tag{4.18}$$

and

$$\overline{F}_{j,m,p}=\frac{1}{2}\left\{F\left(t_{j,m,p}\right)+F\left(t_{j,m+1,p}\right)\right\} \tag{4.19}$$

In order to calculate the temperature at x_{i_b} at (j,m,p), the movement of the moving boundary, between the two real nodes (i_b,j) and $(i_b+1,j+1)$, is approximated by a step shaped function of p_j^k steps of equal length $\delta x_{l,j}^k=\Delta x/p_j^k$ (see Figure 4.3). Therefore, the spatial grid line i_b can be considered to be in the middle of the two virtual grid lines located, respectively, at $x_u=x_{i_b}-m\delta x_{l,j}^k$ and $x_d=x_{i_b}+m\delta x_{l,j}^k$. The temperature at x_u can be calculated by interpolation from temperatures at x_{i_b} and x_{i_b-1}, the temperature at x_d is always T_b. In order not to violate the stability criteria the temperature $T_{i_b,j,m,p}^k$ should be computed from temperatures at x_u,x_{i_b},x_d at $(j,m-1,p)$ in $\left(p_j^k/m\right)^2$ stages. However, in order to minimise the computation time and complexity, the temperature change across each virtual sub-interval is approximated by a step change:

$$T_l(x,t)=\begin{cases}T_l\left(x,t_{j,m-1,p}^k\right) & , \quad 0\le x\le x_{i_b} \quad , \quad t_{j,m-1,p}^k\le t<t_{j,m,p}^k \\ T_l\left(x,t_{j,m,p}^k\right) & , \quad 0\le x\le x_{i_b} \quad , \quad t_{j,m,p}^k\le t<t_{j,m+1,p}^k\end{cases} \tag{4.20}$$

Thus, the temperature $T_{i_b,j,m,p}^k$ will be computed, from temperatures at $(j,m-1,p)$, in one stage (for details see appendix B) using (4.16) where:

$$\psi_{i_b,j,m,p}^k=\frac{1}{T_{i_b,j,m,p}^k}\left\{\left(1+\frac{m+1}{p_j^k}\right)T_{i_b,j,m,p}^k-\left(\frac{m+1}{p_j^k}\right)T_{i_b-1,j,m,p}^k-T_b\right\} \tag{4.21}$$

Hence, the temperature profile between $t=t_j$ and $t=t_{j+1}^k$ at any depth x in the liquid region including x_{i_b} is approximated by a step wise function of p_j^k stages.

Since the VSS $\left(j,m=p_j^k,p\right)$ always coincides with the time $t=t_{j+1}^k$, the temperature distribution at $\left(j,p_j^k,p\right)$ is assigned to the real grid as the temperature distribution in the liquid region at the time $t=t_{j+1}^k$

$$T_{i,j+1}^k=T_{i,j,p_j^k,p}^k \quad , \quad i=0,i_b \tag{4.22}$$

When $j=0$ ($i_b=0$, i.e. the moving boundary is moving from (0,0) to (1,1)), instead of equations (4.17) and (4.21), the following equation is used:

$$\psi_{i_b,j,m,p}^k=\frac{2}{T_{i_b,j,m,p}^k}\left(T_{i_b,j,m,p}^k-\left(\frac{m+1}{p_j^k}\right)\left(\frac{\Delta x}{K_l}\right)\overline{F}_{0,m,p}-T_b\right) \tag{4.23}$$

4.3.2. Computation of temperature distribution in the solid region

In the solid region, the temperature distribution ($T^k_{i,j+1}, i = i_b + 1, N$) at the time $t = t^k_{j+1}$ is computed from that of time $t = t_j$ in q^k_j stages. Similarly, the temperature distribution at each VSS (j,n,q), where $\left\{j = 0, N-1; n = 1, q^k_j\right\}$, in the solid region is computed explicitly from that of the previous VSS $(j,n-1,q)$:

$$T^k_{i,j,n,q} = T^k_{i,j,n-1,q} \exp\left(-r^k_{cs,j}\, \psi^k_{i,j,n-1,q}\right) \quad , \quad i = i_b + 1, N \tag{4.24}$$

where

$$\psi^k_{i,j,n,q} = \frac{2}{T^k_{i,j,n,q}} \left\{ \left(1 + \frac{h\Delta x}{K_s}\right) T^k_{i,j,n,q} - T^k_{i-1,j,n,q} - \left(\frac{h\Delta x}{K_s}\right) T_0 \right\} \quad \text{for} \quad i = N \tag{4.25}$$

and

$$\psi^k_{i,j,n,q} = \frac{1}{T^k_{i,j,n,q}} \left(2T^k_{i,j,n,q} - T^k_{i+1,j,n,q} - T^k_{i-1,j,n,q}\right) \quad \text{for} \quad i = i_b + 2, N-1 \tag{4.26}$$

To calculate the temperature $T^k_{i_b+1,j,n,q}$, the same approach as for $T^k_{i_b,j,m,p}$ is used. The movement of the moving boundary between the two real nodes (i_b, j) and $(i_b + 1, j+1)$ is approximated by a step shaped function of q^k_j steps of equal length $\delta x^k_{s,j} = \Delta x / q^k_j$ (see Figure 4.3). Similarly the temperature $T^k_{i_b+1,j,n,q}$ should be computed in $\left(\left(q^k_j - n\right)/q^k_j\right)^2$ stages in order to maintain the stability criteria. In the same way as for the grid line x_{i_b}, to minimise the computation time and avoid further complications, the temperature evolution between $t = t_j$ and $t = t^k_{j+1}$ at any depth in the solid region (including x_{i_b+1}) is approximated by a step shaped function of q^k_j steps:

$$T_s(x,t) = \begin{cases} T_s\left(x, t^k_{j,n-1,q}\right) & , \quad x_{i_b+1} \le x \le a \quad , \quad t^k_{j,n-1,q} \le t < t^k_{j,n,q} \\ T_s\left(x, t^k_{j,n,q}\right) & , \quad x_{i_b+1} \le x \le a \quad , \quad t^k_{j,n,q} \le t < t^k_{j,n+1,q} \end{cases} \tag{4.27}$$

this allows the temperature $T^k_{i_b+1,j,n,q}$ to be computed explicitly in one stage using (4.24), where:

$$\psi^k_{i_b+1,j,n,q} = \frac{1}{T^k_{i_b+1,j,n,q}} \left\{ \left(2 - \frac{n+1}{q^k_j}\right) T^k_{i_b+1,j,n,q} - \left(1 - \frac{n+1}{q^k_j}\right) T^k_{i_b+2,j,n,q} - T_b \right\} \tag{4.28}$$

As the VSS $\left(j,n=q_j^k,q\right)$ also coincides with the time $t=t_{j+1}^k$, the temperature distribution at $\left(j,q_j^k,q\right)$ is assigned as the temperature distribution in the solid region at the time $t=t_{j+1}^k$:

$$T_{i,j+1}^k = T_{i,j,q_j^k,q}^k \quad , \quad i=i_b+1,N \tag{4.29}$$

When the temperature distributions in both liquid (4.22) and solid regions (4.29) are estimated at the time $t=t_{j+1}^k$ (the intermediate steps are virtual and are progressively eliminated) the Δt_j^{k+1} is recalculated from the finite difference form of (4.6) giving:

$$\Delta t_j^{k+1} = \left(\frac{\lambda \Delta x^2}{K_s}\right)\left[T_{i_b+2,j+1}^k + \mu T_{i_b,j+1}^k - (1+\mu)T_b\right]^{-1} \tag{4.30}$$

where $\mu = K_l/K_s$.

The iteration process at each time step is stopped when the following test is verified:

$$\left|100\times\frac{\Delta t_j^{k+1}-\Delta t_j^k}{\Delta t_j^k}\right| \le \varepsilon_{min} \tag{4.31}$$

The computing procedure at each time step j, begins by choosing $\Delta t_j^{k=0}$ equal to Δt_{j-1}, which is already known from the previous time step. The temperature distribution $T_{i,j+1}^{k=0}, i=0,N$ is determined from (4.22) and (4.29). These values are in turn used to find the new estimate $\Delta t_j^{k=1}$ from (4.30). $\Delta t_j^{k=1}$ is used in the numerical analysis (from (4.13) to (4.29)) to determine the new temperature distribution $T_{i,j+1}^{k=1}, i=0,N$. This process is repeated until the value of Δt_j, for two successive iterations, satisfies (4.31).

The numerical analysis presented (from (4.13) to (4.29)) shows the procedure for computing the temperature distribution (throughout the liquid and solid) at a discrete time $t_{j+1}=t_j+\Delta t_j$ from that of the previous time t_j. The time step interval necessary for the moving boundary to move a single space increment is also determined iteratively. For the complete solution, the procedure is repeated at each real time step until the end of the heat transfer process. In particular, for the test problems in section 4.4, the computation process stops when the moving boundary has reached $x=a$ (i.e. complete melting or complete solidification of the slab).

4.3.3. The special case of zero temperature at the moving boundary

When $T_b = 0$, the temperature at the node (i_b, j) which always represents the position of the moving boundary will be equal to zero also (i.e. $T_{i_b,j,0,p} = T_{i_b,j} = T_b = 0$). Hence,, equations (4.21) and (4.23) becomes indeterminate when $m = 0$.

To calculate $T^k_{i_b,j,m,p}$ when $m = 1$, equations (4.16) and (4.21) must be replaced by:

$$T^k_{i_b,j,1,p} = r^k_{cl,j}\left(\frac{1}{p^k_j}\right)T^k_{i_b-1,j} \tag{4.32}$$

Equation (4.32) is obtained by using the previously stated approximations, i.e. that the moving boundary moves according to an equal-step shaped function during time steps, and applying equation (23) in reference [10].

Similarly, when the moving boundary is moving from (0,0) to (1,1), instead of calculating $T^k_{i_b,j,m,p}$ when $m = 1$ using equations (4.16) and (4.23), the following is used:

$$T^k_{i_b,j,1,p} = r^k_{cl,j}\left(\frac{1}{p^k_j}\right)\left(\frac{\Delta x}{K_l}\right)\overline{F}_{j,0,p} \tag{4.33}$$

4.4. Test problems

Problem A

The numerical scheme, illustrated in section 4.3 which will be referred to as the explicit variable time-step (EVTS) method, is used to solve the heat transfer problem which has been dealt with by: Goodling and Khader [17,18], Gupta and Kumar [7]. As previously stated the process consists of solidification of liquid initially at its fusion temperature throughout by losing heat through its surface by convection; the governing equations are:

$$\frac{\partial T}{\partial t} = \alpha\frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le M(t) \quad , \quad t > 0 \tag{4.34}$$

$$\frac{\partial T}{\partial x} = hT \quad , \quad x = 0 \quad , \quad t > 0 \tag{4.35}$$

$$T(x,t) = 1 \quad , \quad M(t) \le x \le 1.0 \quad , \quad t > 0 \tag{4.36}$$

$$\left.\begin{aligned}\frac{dM}{dt} &= \frac{\partial T}{\partial x}\\ T(x,t) &= 1\end{aligned}\right\} \quad , \quad x = M(t) \quad , \quad t > 0 \tag{4.37}$$

$$M(0) = 0 \tag{4.38}$$

Solutions of the problem A, when $h = 10.0$ and $\alpha = 1.0$, using: the method proposed by Goodling and Khader [17,18] which will be referred to as the GK method, the Extended Douglas and Gallies's method (EDG) proposed by Gupta and Kumar [19], the Modified Variable Time–Step (MVTS) method given also by the same authors [7], and the present method are given in Table 4.1 for increasingly fine spatial resolution.

For accuracy assessment a recourse is made to an analytical solution based on the heat-balance integral method of Goodman [20] (details can be found in [7]). An identical time step error $\varepsilon_{min} = 0.05\%$ has been allowed for all the methods. For relatively large values of N the GK method fails to converge with a small time step error; hence, an $\varepsilon_{min} = 0.5\%$ is used when $\Delta x = 0.01$. For EVTS an $\varepsilon_{min} = 0.05\%$ and $r_f = 0.35$ were maintained in all computations.

Problem B

A solid material melts due to a variable heat input at the fixed boundary, $x = 0$, the temperature throughout the solid is assumed to remain at the melting point; the governing equations in non-dimensional form are:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 \le x \le M(t) \quad , \quad t > 0 \tag{4.39}$$

$$\frac{\partial T}{\partial x} = -\exp(\alpha t) \quad , \quad x = 0 \quad , \quad t > 0 \tag{4.40}$$

$$T(x,t) = 1 \quad , \quad M(t) \le x \le 1.0 \quad , \quad t > 0 \tag{4.41}$$

$$\left.\begin{aligned} \frac{1}{\alpha}\frac{dM}{dt} &= -\frac{\partial T}{\partial x} \\ T(x,t) &= 1.0 \end{aligned}\right\} \quad , \quad x = M(t) \quad , \quad t > 0 \tag{4.42}$$

$$M(0) = 0 \tag{4.43}$$

Exact solution:

$$T(x,t) = \exp(\alpha t - x) \qquad , \qquad M(t) = \alpha t \tag{4.44}$$

Problem B has been solved using MVTS and EVTS, for both methods an error of $\varepsilon_{min} = 0.05\%$ is allowed for time steps and results are computed with $r_f = 0.40$ for the EVTS method. Results for $\alpha = 0.1$ and $\alpha = 10.0$ are given in Tables 4.3 and 4.4 respectively. For $N > 40$ and a time step error of $\varepsilon_{min} = 0.05\%$, the MVTS method does not converge; hence, for $N = 50$ and $N = 100$ the time step error is increased to 0.1% and 0.2% respectively.

The accuracy of the numerical results is assessed by comparison with the analytical solution and expressed as Percentage Deviation (PD) defined as:

$$PD = \frac{\text{(Numerical solution) - (Analytical solution)}}{\text{(Analytical solution)}} \times 100$$

The Percentage Deviation for Surface Temperature $PD_{ST}(M)$), when the moving boundary is at $x=M(t)$, is defined as:

$$PD_{ST}(M) = 100 \times \frac{T_{Nu}(0,t) - T_{An}(0,t)}{T_{An}(0,t)} \tag{4.45}$$

The Percentage Deviation for the time $PD_t(M)$, when the moving boundary is located at $x=M(t)$, is defined as:

$$PD_t(M) = 100 \times \frac{t_{Nu} - t_{An}}{t_{An}} \tag{4.46}$$

where the subscripts An and Nu refer to the value calculated Analytically and Numerically, respectively.

4.5. Numerical results and discussion

Table 4.1 (Problem A): Comparison of time Δt required for the moving interface $M(t)$ to move a distance Δx, Surface temperature $T(0,t)$ is also tabulated.

	$\Delta x = 0.1$		$\Delta x = 0.025$		$\Delta x = 0.01$	
$M(t)$	Δt	$T(0,t)$	Δt	$T(0,t)$	Δt	$T(0,t)$
	0.0200	0.5000	0.0061	0.5280	0.0025	0.5299
0.1	0.0200	0.5000	0.0061	0.5280	0.0025	0.5299
	0.0200	0.5000	0.0061	0.5280	0.0025	0.5299
	0.0196	0.4905	0.0059	0.5315	0.0025	0.5309
	0.0627	0.2242	0.0161	0.2233	0.0065	0.2231
0.4	0.0627	0.2242	0.0161	0.2233	0.0065	0.2231
	0.0627	0.2242	0.0161	0.2234	0.0065	0.2233
	0.0596	0.2235	0.0158	0.2229	0.0064	0.2229
	0.1021	0.1416	0.0259	0.1407	0.0104	0.1407
0.7	0.1021	0.1416	0.0259	0.1407	0.0104	0.1406
	0.1021	0.1417	0.0259	0.1407	0.0104	0.1406
	0.0983	0.1406	0.0256	0.1404	0.0103	0.1404
	0.1413	0.1032	0.0356	0.1026	0.0142	0.1026
1.0	0.1413	0.1032	0.0357	0.1026	0.0143	0.1026
	0.1413	0.1032	0.0356	0.1026	0.0143	0.1026
	0.1370	0.1024	0.0353	0.1024	0.0142	0.1023

Note– For each value of $M(t)$, the first, second, third and fourth row of data corresponds to the EDG, MVTS, GK and EVTS method respectively.

Table 4.1 shows that the results predicted by the EVTS method for test problem A compare very well with the other variable time step methods. It can also be seen that results of the EDG, GK and MVTS methods are identical; the only difference is the time of convergence or number of iterations (MVTS and EDG converge much faster than the GK method [7]). For EVTS, time steps are evaluated within the specified accuracy in the same number of iterations as for MVTS for all but the initial time step, whereas for MVTS it is calculated directly from the boundary conditions and by an iterative process for EVTS. As the GK, EDG and MVTS methods give identical solutions, the EVTS method will be compared, in terms of accuracy, to MVTS only (this comparison being valid for the GK and EDG methods also).

Table 4.2 shows that for test problem A, the position of the moving boundary is calculated with higher accuracy using EVTS than with MVTS for any spatial resolution. For instance the time for complete solidification is calculated with an error of 13.58% using MVTS when $N = 10$, whereas using EVTS it is calculated with an error of 9.21% only.

Unlike problem A, the numerical results for problem B, when $\alpha = 0.1$ which is a relatively slow process, show that the EVTS method is far more accurate than MVTS. From Table 4.3 it can be seen that for example when $N = 10$ the time for complete melting

Table 4.2 (Problem A): Comparison of time t required for the moving boundary to move $M(t)$ for different space increment resolution. Percentage deviation from the analytical solution is also tabulated.

	$N = 10$		$N = 50$		$N = 100$		Analytic
$M(t)$	t	$PD_t(M)$	t	$PD_t(M)$	t	$PD_t(M)$	t
MVTS method							
0.1	0.0200	12.359	0.0185	3.932	0.0182	2.247	0.0178
0.2	0.0556	14.168	0.0512	5.133	0.0505	3.696	0.0487
0.3	0.1050	14.379	0.0972	5.882	0.0961	4.684	0.0918
0.4	0.1677	14.237	0.1565	6.607	0.1549	5.517	0.1468
0.5	0.2436	14.044	0.2288	7.116	0.2267	6.132	0.2136
0.6	0.3327	13.938	0.3141	7.568	0.3116	6.712	0.2920
0.7	0.4348	13.821	0.4125	7.984	0.4095	7.198	0.3820
0.8	0.5500	13.730	0.5239	8.333	0.5204	7.609	0.4836
0.9	0.6782	13.658	0.6483	8.647	0.6444	7.993	0.5967
1.0	0.8194	13.584	0.7857	8.913	0.7816	8.344	0.7214
EVTS method							
0.1	0.0196	10.112	0.0182	2.247	0.0181	1.685	0.0178
0.2	0.0532	09.240	0.0503	3.285	0.0500	2.669	0.0487
0.3	0.0998	08.714	0.0955	4.030	0.0952	3.703	0.0918
0.4	0.1594	08.583	0.1539	4.836	0.1535	4.564	0.1468
0.5	0.2319	08.567	0.2253	5.471	0.2248	5.243	0.2136
0.6	0.3172	08.630	0.3097	6.061	0.3091	5.856	0.2920
0.7	0.4156	08.795	0.4071	6.570	0.4065	6.413	0.3820
0.8	0.5256	08.912	0.5174	6.989	0.5169	6.885	0.4836
0.9	0.6509	09.083	0.6408	7.390	0.6403	7.306	0.5967
1.0	0.7879	09.218	0.7772	7.734	0.7766	7.651	0.7214

Table 4.3 (Problem B): Comparison of percentage deviation for time PD_t and for surface temperature PD_{ST} when the moving boundary is located at $M(t)$ for different space increment resolutions. $\alpha = 0.1$

	$N = 10$		$N = 50$		$N = 100$	
$M(t)$	PD_t	PD_{ST}	PD_t	PD_{ST}	PD_t	PD_{ST}
MVTS method						
0.1	-8.730	0.398	22.519	-1.302	26.729	-1.429
0.2	14.370	-1.874	43.330	-3.103	47.254	-3.225
0.3	31.376	-3.266	59.353	-4.111	63.200	-4.215
0.4	45.040	-3.566	72.452	- 3.901	76.297	-3.934
0.5	56.445	-2.589	83.469	- 2.195	87.321	-2.122
0.6	66.224	-0.120	92.923	1.289	96.796	1.531
0.7	74.744	-4.057	101.145	6.872	105.042	7.359
0.8	82.256	10.231	108.382	14.935	112.278	15.762
0.9	88.946	18.681	114.816	25.943	118.666	27.195
1.0	94.915	29.783	120.566	40.455	124.385	42.294
EVTS method						
0.1	0.959	-0.551	0.129	-0.162	0.059	-0.081
0.2	0.880	-1.375	-0.070	-0.262	-0.019	-0.131
0.3	0.526	-1.948	-0.176	-0.362	-0.036	-0.192
0.4	0.140	-2.359	-0.219	-0.449	-0.027	-0.234
0.5	-0.160	-2.729	-0.231	-0.533	-0.012	-0.278
0.6	-0.369	-3.078	-0.235	-0.614	-0.001	-0.318
0.7	-0.505	-3.406	-0.230	-0.690	-0.017	-0.352
0.8	-0.590	-3.698	-0.223	-0.750	-0.031	-0.372
0.9	-0.638	-3.972	-0.215	-0.813	-0.042	-0.394
1.0	-0.666	-4.226	-0.209	-0.817	-0.053	-0.415

using MVTS is calculated with an error of 94.915% but only 0.666% using EVTS. It can also be seen that the accuracy obtained by EVTS with only 10 space elements is not achievable with MVTS even using more than 100 space elements. Similarly the accuracy of the EVTS method remains outstanding even when the velocity of the moving boundary is relatively high. This is illustrated in Table 4.4 where numerical results are presented for problem B when $\alpha = 10.0$.

Table 4.4 (Problem B): Comparison of time t required for the moving boundary to move $M(t)$ along with surface temperature $T(0,t)$. Percentage deviation for both time (PD_t) and surface temperature (PD_{ST}) are also tabulated. ($N = 10$ and $\alpha = 10$)

	MVTS method				EVTS method				Analytic	
$M(t)$	t	PD_t	$T(0,t)$	PD_{ST}	t	PD_t	$T(0,t)$	PD_{ST}	t	$T(0,t)$
0.1	0.0091	-9.000	1.1096	0.398	0.0101	0.999	1.0991	-0.551	0.0100	1.1052
0.2	0.0176	-11.999	1.2371	1.285	0.0202	1.000	1.2046	-1.375	0.0200	1.2214
0.3	0.0256	-14.666	1.3824	2.407	0.0302	0.666	1.3236	-1.948	0.0300	1.3499
0.4	0.0331	-17.249	1.5450	3.566	0.0401	0.250	1.4566	-2.359	0.0400	1.4918
0.5	0.0402	-19.599	1.7246	4.603	0.0499	-0.200	1.6037	-2.729	0.0500	1.6487
0.6	0.0469	-21.833	1.9211	5.433	0.0598	-0.333	1.7660	-3.078	0.0600	1.8221
0.7	0.0534	-23.714	2.1341	5.973	0.0696	-0.571	1.9452	-3.406	0.0700	2.2255
0.8	0.0597	-25.374	2.3635	6.200	0.0795	-0.624	2.1432	-3.698	0.0800	2.4596
0.9	0.0657	-26.999	2.6093	6.086	0.0894	-0.666	2.3619	-3.972	0.0900	2.4596
1.0	0.0715	-28.500	2.8714	5.632	0.0993	-0.700	2.6035	-4.223	0.1000	2.7183

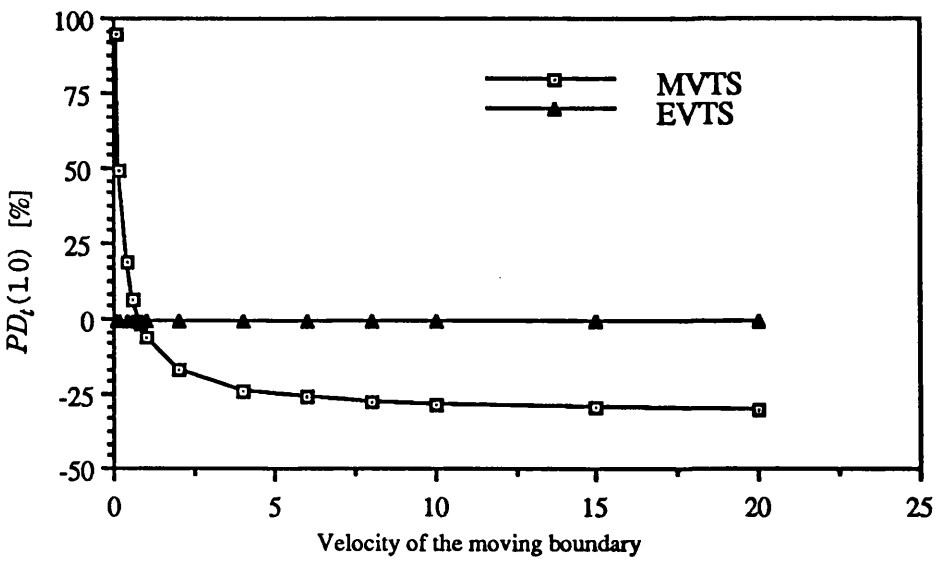

Figure 4.4: Variation of the $PD_t(1.0)$ with the velocity of the moving boundary

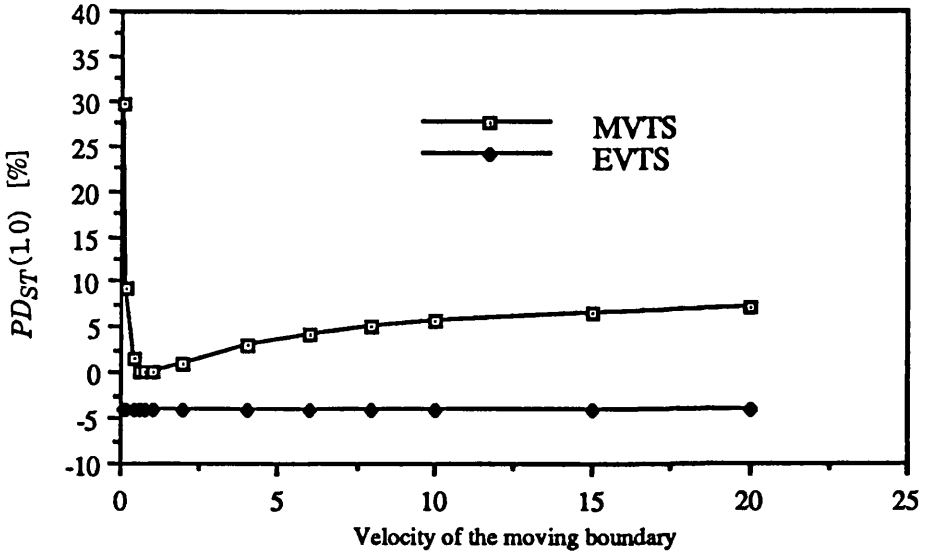

Figure 4.5: Variation of the $PD_{ST}(1.0)$ with the velocity of the moving boundary

When problem B is solved with different values of α, which represents also the velocity of the moving boundary, $dM/dt = \alpha$, the MVTS method is very inaccurate for small and large values of α, whereas the EVTS method achieves good accuracy for all values of α. This is illustrated in Figure 4.4 and 4.5. These figures shows that for EVTS both the $PD_t(1.0)$ and $PD_{ST}(1.0)$ remain constant with the variation of the speed of the moving boundary (this holds true for all $PD_t(M)$ and $PD_{ST}(M)$). It can also be seen from these figures that MVTS cannot accommodate either slow or fast moving boundaries.

When the moving boundary is either slow or fast, the number of space elements is relatively large and the specified time step error (ε_{min}) is small; the MVTS method fails to converge. Unlike MVTS, using EVTS the same time step error was maintained for all computations without exhibiting any convergence difficulties.

4.6. Conclusion

The results for both test problems show that the method presented herein outperforms the implicit methods and exhibits exceptional accuracy in the case of rapid or slow Stefan problems. The EVTS method avoids the solution of a system of linear equations at each time step, it requires less memory resources and is highly stable. Since its accuracy is not affected by the velocity of the moving boundary, the approach is highly efficient compared to the implicit methods for modelling rapid or slow heat transfer processes. Due to its explicit procedure the EVTS method, where the unknown variables are determined with no recurrence (i.e. statements that form a cycle of variable dependencies), is more suitable for vectorisation than the implicit finite difference and

finite element methods. Therefore the EVTS method would be more beneficial than implicit and finite element methods in terms of computational efficiency using vectorisation facilities.

The EVTS method combines the advantages of variable time step methods and explicit procedures whilst excluding their disadvantages. The variable time step approach avoids the use of unnecessary real time steps if the moving boundary is moving with relatively slow velocity; this results in computational efficiency and storage economy. In contrast, if the moving boundary is rapid, the time step is chosen in such a way as to avoid the break-down of the scheme that would occur for a fixed grid network. The VSIET automatically preserves the stability of the scheme, irrespective of the velocity of the moving boundary (without increasing the storage requirements), and reduces the loss of accuracy associated with relatively large time steps.

4.7. References

1. J. Crank (ed.), *Free moving boundary problems*, Clarendon Press, Oxford (1984).
2. R.S. Gupta, Ph.D. Thesis, Brunel University, Uxbridge (1972).
3. D.R. Atthey, *J. Inst. Math. Appl.* **13**, 353-366 (1974).
4. W.D. Murray and F. Landis, *J. Heat Transfer* **81**, 106-112 (1959).
5. J. Douglas and T.M. Gallie, *Duke Math. Jl* **22**, 557-570 (1955).
6. C. Haberland and A. Lahrmann, *Int. J. Numer. Meth. Eng.* **25**, 593-609 (1988).
7. R. S. Gupta and D. Kumar, *Int. J. Heat Transfer* **24**, 251-259 (1981).
8. K.H. Hoffman (ed.), Freie Universitat (Berlin) Fachbereich Mathmatik Vols I-III, Berlin (1977).
9. R.M. Furzeland, *J. Inst. Math. Appl.* **26**, 411-429 (1980).
10. M.C. Bhattacharya, *Int. J. Numer. Meth. Eng.* **21**, 239-265 (1985).
11. M.C. Bhattacharya, *Appl. Math. Modelling* **10**, 68-70 (1986).
12. M.C. Bhattacharya, *J. Sound Vibration* **107**, 507-521 (1986).
13. M.C. Bhattacharya and M.G. Davies, *Int. J. Numer. Meth. Eng.* **24**, 1317-1331 (1987).
14. M.C. Bhattacharya, *Communications Appl. Numer. Meth.* **6**, 173-184 (1990).
15. R.B. Potts, *J. Aust. Math. Soc.* **B27**, 488-501 (1986).
16. A.R. Gourlay, *Proc. Roy. Soc. London* **A323**, 219-235 (1971).
17. J.S. Goodling and M.S. Khader, *AFS Cast Metals Res. Jl* **10**, 26-29 (1974).
18. J.S. Goodling and M.S. Khader, *J. Heat Transfer* **96**, 114-115 (1974).
19. R.S. Gupta and D. Kumar, *Comp. Meth. Appl. Eng.* **23**, 101-109 (1980).
20. T.R. Goodman, *Trans. Am. Soc. Mech. Engrs* **80**, 335-342 (1959).

5 Implicit Moving Boundary Problems (Oxygen Diffusion Problem)

Summary

The diffusion of oxygen into an absorbing medium, as an example of implicit moving boundary problems, has been dealt with by a number of authors using various numerical techniques and, where appropriate, approximate analytical expressions. To evaluate the time for complete absorption, extrapolation is usually employed. An unconditionally stable explicit numerical scheme that avoids the limitations of such methods is presented and tested herein. Unlike existing schemes this method is fully numerical; it avoids the large array size, generally required for existing methods, by using a variable length time step. The time for complete absorption emerges from the final step in the normal computing procedure with no recourse to extrapolation. Furthermore, due to the implicit condition prevailing at the moving boundary, no iterations are needed to evaluate the time step required for the moving boundary to move a single space increment. The numerical results obtained compare very favourably with those due to earlier authors.

Nomenclature

D	Diffusion coefficient	X, x	Space, dimensionless space
M	Dimensionless moving boundary	τ	Time variable
n	Total number of space intervals	Δt	Dimensionless time step
R	Rate of consumption of oxygen per unit volume of the medium	Δx	Dimensionless space interval
			Subscripts
r_f	A fixed value of Fourier number	i	Space node index
U, u	Oxygen concentration, dimensionless concentration	i_b	Space node index containing the moving boundary
U_0	Initial concentration at the surface	j	Time step index
S	Position of the moving boundary		*Superscript*
t	Dimensionless time variable	m	Virtual time step index

5.1. Introduction

Under certain conditions, the phenomenon of oxygen diffusion from blood into oxygen-consuming tissue, gives rise to a moving boundary. A typical example arises when studying tumour tissue-oxygen interaction. Generally, in moving boundary problems, the velocity of the Moving Boundary (MB) is determined by the physical requirement that the latent heat required for the change of phase must be supplied or removed by conduction. In the oxygen diffusion problem, a moving boundary is an essential feature of the problem, but the conditions which determine its movement are different. Not only the concentration of oxygen is always zero at the MB, no oxygen diffuses across it at any time. Unlike general moving boundary problems, an explicit relationship containing the velocity of the MB is absent.

The problem was firstly reported by Crank and Gupta [1], who studied the moving boundary problem arising from the diffusion of oxygen into absorbing tissue. First the oxygen is allowed to diffuse into a medium, some of the oxygen is absorbed by the medium, thereby being removed from the diffusion process. The concentration of oxygen at the surface of the medium is maintained constant. This phase of the problem continues until a steady state is reached in which the oxygen does not penetrate any further into the medium. The supply of oxygen is then cut off and the surface is sealed so that no oxygen passes in or out, the medium continues to absorb the available oxygen already in it and as a consequence the boundary marking the furthest depth of penetration in the steady state, starts to recede towards the sealed surface. The major problem is that of tracking the movement of the MB during this stage of the problem as well as determining the distribution of oxygen throughout the medium at any instant in time.

This type of problem (also known as implicit moving boundary problems) has been addressed by Crank and Gupta [1]; initially when the MB is moving slowly they used an approximate analytical solution and a numerical scheme once the velocity of MB has increased. Their numerical scheme consists of using a fixed grid network and calculating the concentration at each node using the Euler equation. For the grid near the MB, a Lagrange type formula is used and the location of the MB is determined using a Taylor series expansion. In a later paper [2] the same authors used a uniform grid system which moves with the velocity of the MB. This has the effect of transferring the unequal interval from the neighbourhood of the MB to the surface of the medium and therefore gives an improvement in the smoothness of the calculated motion of the MB. They evaluated the concentration distribution at the new grid points by interpolation using either: cubic splines or ordinary polynomials.

Hansen and Hougaard [3] used an integral equation for the function defining the position of the MB and an integral formula for the concentration distribution. The integral equation is solved asymptotically during the entire motion of the MB whereas the concentration integral is solved asymptotically for small times, and computed by numerical quadrature at later times. Apart from the lengthy computational procedure, their results are acknowledged to be the most accurate.

Many other authors have dealt with the problem by various methods; for instance, Ferris and Hill [4] used an appropriate variable transformation to fix the MB, Berger *et al.* [5] used a truncation method, whilst Miller *et al.* [6] used finite elements. More references relating to the oxygen diffusion problem involving a moving boundary can be found in [7] and [8].

The common feature of the numerical methods [1-8] is that they adopt a fixed grid network in space and time, and utilise both numerical computation and analytical approximations. In order to prevent the oxygen concentration going negative, which causes instability problems, these methods [1-8] often resort to small time steps which not only increase the cpu-time but the array size requirements also. Furthermore, the numerical procedures can not be used up to the end of the absorption process due to the lack of necessary mesh points, when the MB is too close to the sealed surface; therefore, in all these methods extrapolation is used to evaluate the time at which the MB reaches the sealed surface.

Gupta and Kumar [9], by using a variable time step grid network, avoid the large number of time steps generally required for the methods reported in [1-8]. Their method computes the concentration distribution using the Crank–Nicolson implicit finite difference equation. Due to the implicit boundary condition, an integral equation is used to determine the time step; this gives rise to convergence difficulties because it is very sensitive to a change in concentration. Furthermore, the first estimate of the time step

length at each step is a major difficulty, as it is crucial for the convergence and the stability of the scheme.

In this chapter the EVTS method [10] is modified to solve the oxygen diffusion as an example of implicit moving boundary problems. The significance of the method is that it combines the advantages of the variable time step grid and explicit equations, whilst excluding their disadvantages. The variable time step approach avoids unnecessary time steps at the beginning of the process when the MB is moving with a relatively slow velocity, this results in a high computational efficiency and storage economy. In contrast, by the end of the diffusion process the MB is moving relatively quickly and the time step is automatically adjusted to a smaller value; this avoids the break down that would occur for a fixed grid network which would fail unless a very small time step was used throughout. The stability of the algorithm is automatically maintained for any mesh size, without increasing the minimum array size requirement, by using virtual time steps which are progressively eliminated as the computation moves forward. Furthermore, the iterative process, generally required to determine the time step necessary for the MB to be located at a mesh point, is unnecessary.

5.2. Mathematical formulation of the problem

The oxygen diffusion problem defined in the introduction, is formulated in two stages; firstly: the steady-state, and secondly: the moving boundary stage. The solution of the steady-state, which is formulated as a boundary value problem, is readily available in [1]. This chapter concentrates on the moving boundary problem, which requires evaluation of the oxygen concentration distribution $U(X,\tau)$, between the outer surface X=0, and the position of the MB, $X = S(\tau)$, at any instant in time τ (τ=0 is the time when the surface is sealed). The functional relationship between these variables is given by the diffusion equation:

$$\frac{\partial U}{\partial \tau} = D\frac{\partial^2 U}{\partial X^2} - R \quad , \quad 0 \le X \le S(\tau) \quad , \quad \tau > 0 \tag{5.1}$$

with the following boundary conditions:

$$\frac{\partial U}{\partial X} = 0 \quad , \quad X = 0 \quad , \quad \tau \ge 0 \tag{5.2}$$

$$U = \frac{\partial U}{\partial X} = 0 \quad , \quad X = S(\tau) \quad , \quad \tau > 0 \tag{5.3}$$

$$U = \frac{R}{2D}(X - S_0)^2 \quad , \quad 0 \le X \le S_0 \quad , \quad \tau = 0 \tag{5.4}$$

$$S_0 = S(0) = \sqrt{\frac{2DU_0}{R}} \tag{5.5}$$

where $U(X,\tau)$ denotes the concentration of oxygen free to diffuse at depth X, time τ; D and R are the diffusion coefficient and the rate of oxygen consumption per unit volume of the medium, respectively. Equations (5.4) and (5.5) are the solution of the steady state [1], where U_0 is the concentration at which the surface was maintained until a steady state is reached, and S_0 is the maximum depth of oxygen penetration. To simplify the problem the variables are made dimensionless:

$$x = \frac{X}{S_0} \quad , \quad t = \frac{D}{S_0^2}\tau \quad , \quad u = \frac{D}{RS_0^2}U = \frac{1}{2U_0}U \quad , \quad M = \frac{S}{S_0}. \tag{5.6}$$

The system of equations (5.1) to (5.5) is thus reduced to the following non-dimensional formulation:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - 1 \quad , \quad 0 \le x \le M(t) \quad , \quad t > 0 \tag{5.7}$$

$$\frac{\partial u}{\partial x} = 0 \quad , \quad x = 0 \quad , \quad t \ge 0 \tag{5.8}$$

$$u = \frac{\partial u}{\partial x} = 0 \quad , \quad x = M(t) \quad , \quad t > 0 \tag{5.9}$$

$$u = \tfrac{1}{2}(1-x)^2 \quad , \quad 0 \le x \le 1 \quad , \quad t = 0 \tag{5.10}$$

$$M(t) = 1 \quad , \quad t = 0 \tag{5.11}$$

The absence of (dM/dt) from (5.9) makes this an implicit moving boundary problem.

5.3. Numerical computation scheme

The space region $0 \le x \le 1$ is divided into n equal sections of length $\Delta x = 1/n$. Let $u_{i,j}$ be the oxygen concentration at the nodal point (x_i, t_j) in the space-time domain, given by the coordinates

$$\left(x_i = i\Delta x \quad , \quad t_j = \sum_{q=0}^{j-1} \Delta t_q \right) \tag{5.12}$$

where i=0,n; j=0,n and Δt_q is the time necessary for the MB to move from x_{n-q} to x_{n-q-1}. The notation $u_{i,j}^m$ denotes the value of u at x_i at the virtual time t_j^m which is given by:

$$t_j^m = m r_f \Delta x^2 + \sum_{q=0}^{j-1} \Delta t_q \tag{5.13}$$

where r_f is a fixed value of Fourier number less than or equal to $\frac{1}{2}$ and m is a virtual time step index.

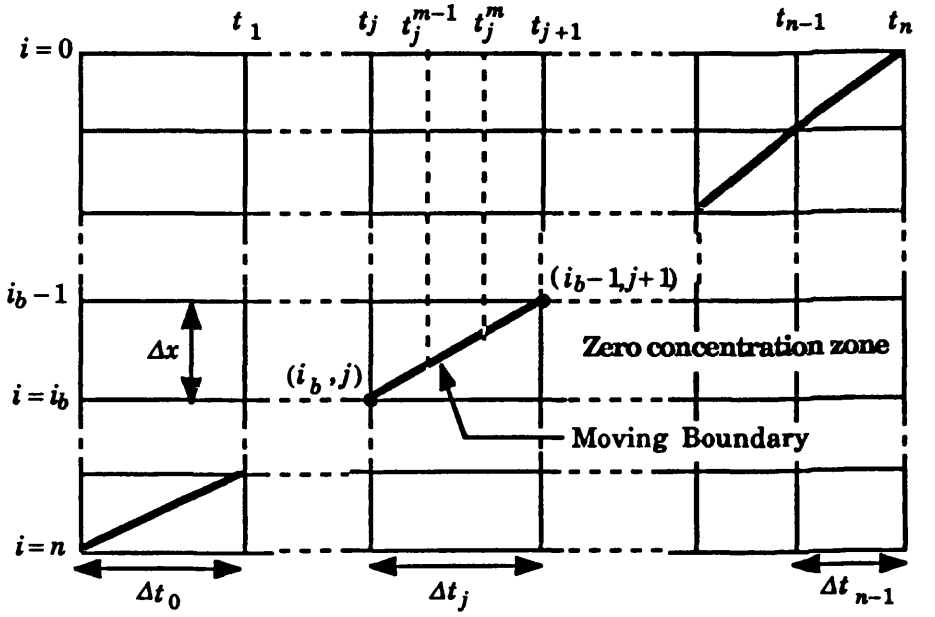

Figure 5.1: Discretisation of the space-time domain

Suppose that the moving boundary is located at the node (i_b, j) at the time $t = t_j$, then the problem is to compute the concentration distribution at the time $t = t_{j+1}$ – from that of $t = t_j$ – as well as the magnitude of time step necessary for the MB to move from the node (i_b, j) to $(i_b - 1, j + 1)$, where i_b is the grid line containing the MB ($i_b = n - j$) (Figure 5.1). Therefore, the numerical scheme proceeds as follows:

The concentration distribution, $u_{i,j}^m; i = 0, i_b - 1$, at any virtual time $t = t_j^m$ ($t_j < t_j^m \leq t_{j+1}$) is calculated (see details in appendix C.1) from that of the previous virtual time $t = t_j^{m-1}$ using:

$$u_{i,j}^m = u_{i,j}^{m-1} - r_f \theta_{i,j}^{m-1} \left(1 - \frac{r_f}{2u_{i,j}^{m-1}} \theta_{i,j}^{m-1} \right) - r_f \Delta x^2 \quad , \quad i = 0, i_b - 1 \tag{5.14}$$

where

$$\theta_{i,j}^{m}=\begin{cases}2\left(u_{i,j}^{m}-u_{i+1,j}^{m}\right) & , \; i=0\\ 2u_{i,j}^{m}-u_{i+1,j}^{m}-u_{i-1,j}^{m} & , \; i=1,i_b-2\\ 2u_{i,j}^{m}-u_{i-1,j}^{m} & , \; i=i_b-1\end{cases} \tag{5.15}$$

Suppose at some intermediate virtual time step $m=k$, so that $u_{i_b-1,j}^{k}\le 0$, then the time step, Δt_j, necessary for the MB to move from (i_b,j) to $(i_b-1,j+1)$, is given by:

$$\Delta t_j=\left[(k-1)r_f+\bar{r}_j\right]\Delta x^2 \quad \text{or} \quad t_{j+1}=\left[(k-1)r_f+\bar{r}_j\right]\Delta x^2+\sum_{q=0}^{j-1}\Delta t_q \tag{5.16}$$

where $\bar{r}_j$ is the positive root of the quadratic equation:

$$\bar{r}_j\theta_{i_b-1,j}^{k-1}\left(1-\frac{\bar{r}_j}{2u_{i_b-1,j}^{k-1}}\theta_{i_b-1,j}^{k-1}\right)+\bar{r}_j\Delta x^2-u_{i_b-1,j}^{k-1}=0 \tag{5.17}$$

in order to satisfy $u_{i_b-1,j+1}=0$.

After determination of $\bar{r}_j$, the concentration distribution at the time $t=t_{j+1}$ is calculated using:

$$u_{i,j+1}=u_{i,j}^{k-1}-\bar{r}_j\theta_{i,j}^{k-1}\left(1-\frac{\bar{r}_j}{2u_{i,j}^{k-1}}\theta_{i,j}^{k-1}\right)-\bar{r}_j\Delta x^2 \quad , \quad i=0,i_b-1 \tag{5.18}$$

During the last time step Δt_{n-1} (i.e. the MB is moving from $x=\Delta x$ to $x=0$), $\theta_{i,j}^{m}$, given by (5.15), becomes:

$$\theta_{i,j}^{m}=2u_{i,j}^{m} \quad , \quad i=0 \quad , \quad j=n-1 \tag{5.19}$$

since the virtual nodal point $\left(x_1,t_{n-1}^{m}\right)$ is at the zero concentration zone and therefore $u_{1,n-1}^{m}=0$.

Remark: Numerically, it was found that $\Delta t_{n-1}=u_{0,n-1}$, the same equation given by Gupta and Kumar (Eq. (4.1) in [9]) using an integral equation.

For each real time step, the numerical procedure starts by computing the oxygen distribution for each virtual time step, which are progressively eliminated, until there is a change in the sign of the concentration at the grid line $i=i_b-1$; then the time at which

the MB is located at x_{i_b-1} is corrected using equations (5.16) and (5.17), and the concentration distribution at the corrected time is evaluated from equation (5.18) (the full numerical procedure is shown more clearly by the algorithm in appendix C.2).

5.4. Numerical results and discussion

Table 5.1 shows the time corresponding to the position of the moving boundary $x = M(t)$ and the concentration distribution at different discrete depths x, using both the present algorithm and the method of Gupta and Kumar [9]. It shows that both results are in good agreement. However, the time step – in the method of Gupta and Kumar [9] – is calculated iteratively using an integral equation, which gives rise to convergence difficulties which develop as n increases causing the program to go into an infinite loop as it becomes very sensitive to rounding errors [11]. Furthermore, the first estimation of Δt_j, for each time step, is a major difficulty in their method; the value Δt_j must be carefully chosen to avoid a negative concentration which causes instability problems.

In order to make comparison with other numerical methods, in which predominantly a fixed grid network is used, a linear interpolation using the two closest points has been made. In Table 5.2, the differences in the boundary position – relative to the results of Hansen and Hougaard [3] – for different methods are shown. The results of Crank and Gupta [1] are initially in good agreement with those of Hansen and Hougaard [3], they then start to deviate; this is due to the fact that the position of the MB is calculated using the first terms of the Taylor series which exclude the velocity of the MB. Provided that the MB is not moving fast such an approximation is reasonable; however, it loses accuracy as the velocity of MB increases with time. The results of Miller *et al.* [6] behave in similar manner as those of Crank and Gupta [1], where the difference increases as time increases; whereas, the interpolated values of Gupta and Kumar [9] oscillate around those of Hansen and Hougaard [3]. The results due to the present algorithm give fairly good agreement, despite the fact that these values are interpolated to make a comparison at similar times. The deviation evident is mainly due to interpolation error, which decreases as time increases because the time steps becomes smaller with a consequent decrease in interpolation error. Furthermore, Hansen and Hougaard [3], by using very small time steps (10^{-4}), predict the time for complete absorption to be somewhere between 0.1972 and 0.1977; whilst Miller *et al.* [6] predict it to be in the range of 0.196 and 0.198. Using the present algorithm with only 20 space elements (n=20), the time for total absorption is predicted to be 0.19744, when n=40 it is 0.19742, which are closer to the accurate value of 0.19743 obtained by Dahmardah and Mayers [12].

Table 5.1: Comparison of values of concentration $u(x,t)\times10^6$ at different depths x at time t when the moving boundary is located at $M(t)$. $n = 100$

$M(t)\backslash x$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	t
0.9	12660	12317	11327	9791	7869	5764	3698	1898	591	0	0.10912
	12281	11941	10955	9432	7537	5471	3458	1724	495	0	0.11159
0.8	7985	7734	7008	5887	4495	2995	1585	591	0	0	0.13845
	7966	7712	6980	5851	4452	2949	1543	466	0	0	0.13861
0.7	5311	5107	4519	3618	2522	1392	448	0	0	0	0.15679
	5318	5113	4520	3613	2511	1379	438	0	0	0	0.15669
0.6	3663	3487	2983	2223	1330	492	0	0	0	0	0.16870
	3501	3327	2829	2081	1208	407	0	0	0	0	0.16983
0.5	2440	2286	1850	1209	506	0	0	0	0	0	0.17787
	2210	2060	1636	1020	370	0	0	0	0	0	0.17954
0.4	1259	1130	772	296	0	0	0	0	0	0	0.18704
	1297	1166	805	323	0	0	0	0	0	0	0.18665
0.3	687	573	275	0	0	0	0	0	0	0	0.19162
	671	558	262	0	0	0	0	0	0	0	0.19165
0.2	295	198	0	0	0	0	0	0	0	0	0.19483
	275	178	0	0	0	0	0	0	0	0	0.19490
0.1	56	0	0	0	0	0	0	0	0	0	0.19683
	63	0	0	0	0	0	0	0	0	0	0.19668
0.0	0	0	0	0	0	0	0	0	0	0	0.19732
	0	0	0	0	0	0	0	0	0	0	0.19722

Note: For each value of $M(t)$, the first and the second row of data corresponds to the method of Gupta and Kumar [9] and the present method respectively. The numerical method of Gupta and Kumar starts from t=0.02 (Approximate solution), whereas the present method starts from t=0.

Table 5.2: Differences ($\times10^4$) in the position of the moving boundary M(t) calculated from those obtained by Hansen and Hougaard [3].

time	Hansen and Hougaard [3] $M(t)$	Crank and Gupta [1]	Miller, Morton & Baines [6]	Gupta and Kumar [9] (i)	Gupta and Kumar [9] (ii)	present method $n = 50$
0.060	0.99180	4.0	2.0	-18.7	-18.8	-107.2
0.080	0.97155	3.5	1.7	24.4	23.1	-97.1
0.100	0.93501	1.7	5.8	-101.3	-105.5	-93.7
0.120	0.87916	-3.1	4.1	-88.3	-132.7	-90.9
0.140	0.79891	-3.5	2.5	-73.4	41.6	-88.6
0.150	0.74668	-18.1	0.1	-72.4	-84.2	-86.2
0.160	0.68337	-20.9	-2.0	-9.0	-10.1	-86.2
0.180	0.50109	-50.2	-26.0	-234.4	-210.7	-83.3
0.190	0.34537	-66.4	-56.6	-100.5	-61.7	-77.4
0.195	0.20652	-452.4	–	-140.7	3.8	-65.2
0.196	0.16266	–	-165.6	-138.4	-52.9	-60.4

Note: (i) corresponds to values calculated using 100 equal intervals of length $\Delta x = 0.01$.
(ii) corresponds to values calculated using 58 unequal intervals of length $\Delta x_i = x_{i+1} - x_i$
where $x_i = \sqrt{i-8}/50,\ 0 \le i \le 16$; $x_i = i/16\sqrt{8/50},\ 16 \le i \le 58$.

Table 5.3: Comparison of the concentration u(0,t) at the sealed surface (x=0) at different times

time	Hansen and Hougaard [3]	Crank and Gupta [1]	Miller, Morton and Baines [6]	Gupta and Kumar [9] (i)*	(ii)*	present method $n = 50$
0.10	0.14318	0.14329	0.14315	0.14234	0.14236	0.14326
0.15	0.06308	0.06316	0.06302	0.06281	0.06282	0.06306
0.18	0.02178	0.02182	-	0.02162	0.02163	0.02175
0.19	0.00902	0.00904	0.00893	0.00887	0.00888	0.00899

* see footnote of Table 5.2

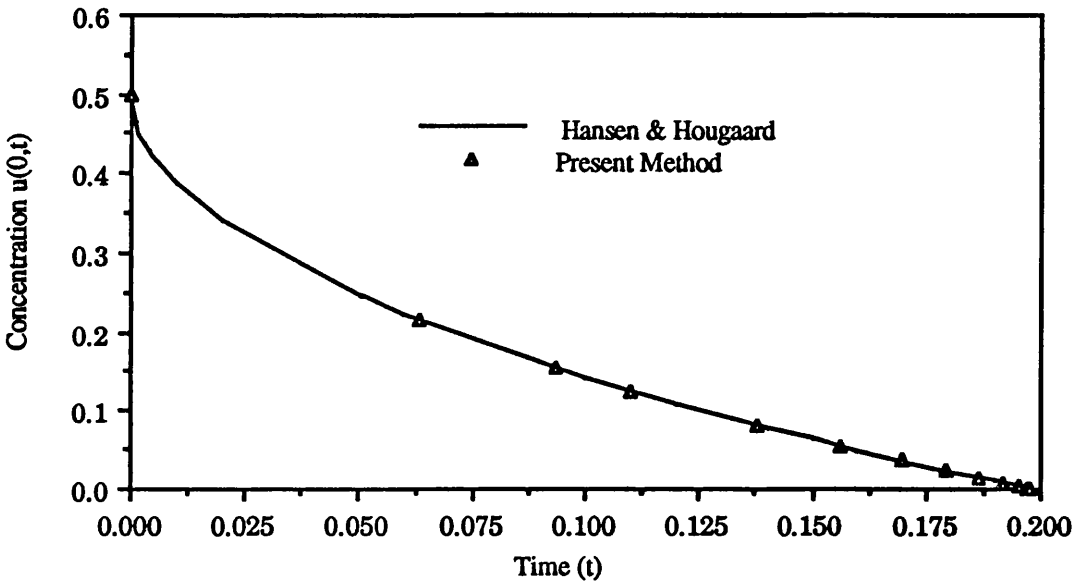

Figure 5.2: Variation of the concentration $u(0,t)$ at the sealed surface $x=0$ with time t

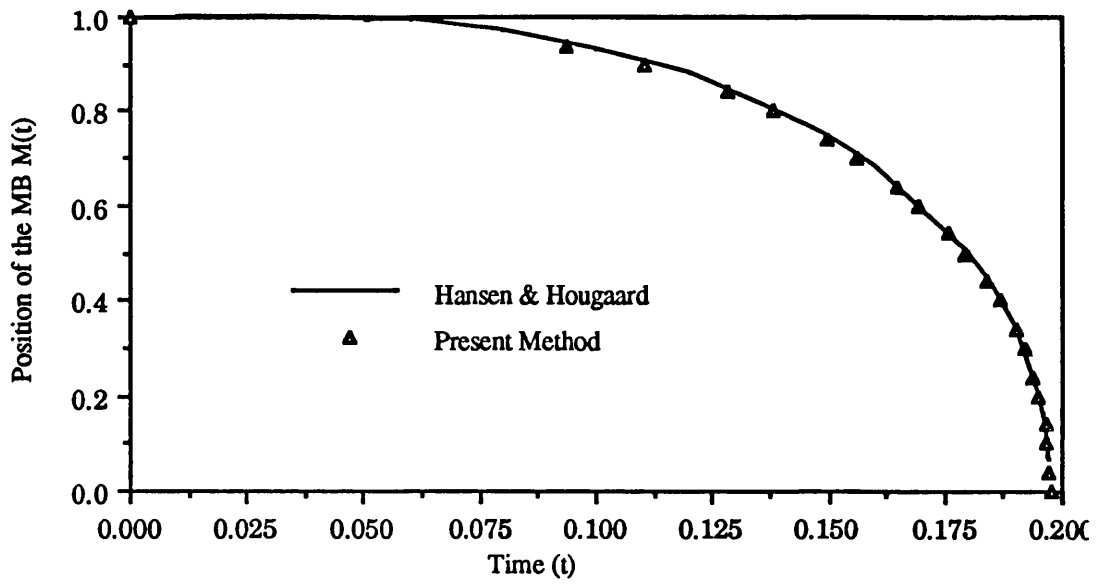

Figure 5.3: The position of the moving boundary $M(t)$ as a function of time t

Table 5.3 shows the concentration at the surface (x=0) at different times. It shows that the results of Crank & Gupta [1] and Miller *et al.* [6] diverge from those of Hansen and Hougaard as time increases. However, the interpolated results due to the present algorithm agree very well with those of Hansen and Hougaard [3]; the difference decreases as time increases due to decrease in the interpolation error as the velocity of the MB increases with time.

In Figures 5.2 and 5.3 a comparison of both concentration and the movement of the MB is shown using Hansen and Hougaard's method and the present algorithm. Both figures show that the agreement between the two methods is good.

5.5. Conclusion

Numerical results show that the present algorithm achieves similar results to other more lengthy and time consuming procedures. The algorithm avoids: numerical integration, solution of systems of linear equations and the iterative process to track the MB at a mesh point. Furthermore, the computation procedure is fully numerical and carried out up to the end of the absorption process without any computational difficulties. The time for complete absorption emerges from the normal computing process without recourse to extrapolation. It also uses a smaller array size by using virtual sub-intervals and is very suitable for vectorisation due to its explicit computation scheme.

5.6. References

1. J. Crank and R.S. Gupta, *J. Inst. Math. Appl.* **10**, 19-33 (1972).
2. J. Crank and R.S. Gupta, *J. Inst. Math. Appl.* **10**, 269-304 (1972).
3. E. Hansen and P. Hougaard, *J. Inst. Math. Appl.* **13**, 385-398 (1974).
4. D.H. Ferris and S. Hill, NPL rept. NAC 45 (1974).
5. A.E. Berger, M. Ciment and J.C.W. Rogers, *SIAM J. Numer. Anal.* **12**, 646-672 (1975).
6. J.V. Miller, K.W. Morton and M.J. Baines, *J. Inst. Math. Appl.* **22**, 467-477 (1978).
7. R.M. Furzeland, *J. Inst. Math. Appl.* **26**, 411-429 (1980).
8. J. Crank (ed.), *Free and Moving Boundary Problems*, Clarendon Press, (1984).
9. R.S. Gupta and D. Kumar, *Comput. Meth. Appl. Mech. Eng.* **29**, 233-239 (1981).
10. M. Zerroukat and C.R. Chatwin, *Int. J. Numer. Meth. Eng.* 35, 1503-1520 (1992).
11. R.S. Gupta and D. Kumar, *Comput. Meth. Appl. Mech. Eng.* **4**, 143-152 (1974).

12. H.O. Dahmadah and D.F. Mayers, *IMA. J. Numer. Anal.* **3**, 81-85 (1983).

13. M.C. Bhattacharya, *Int. J. Numer. Meth. Eng.* **21**, 239-265 (1985).

14. M.C. Bhattacharya, *Commu. Appl. Numer. Meth.* **6**, 173-184 (1990).

6 Multiple Moving Boundary Problems

Summary

Most numerical methods developed for moving boundary or Stefan problems deal with the case of a single Moving Boundary (MB) separating two different media. Although this is applicable to a large number of engineering problems, there are many problems where more than one MB exists simultaneously during the process. A heat transfer process involving heating of a solid, melting and partial vaporisation of liquid can be considered as a three–phase Stefan or a two MB problem, where the time of appearance and disappearance of phases are to be determined as a part of the solution. An explicit unconditionally stable numerical scheme for such problems is presented and tested herein. The approach originates from the Explicit Variable Time Step (EVTS) method, developed in chapter 4, for single MB problems. During the vaporisation stage, where two MBs exist simultaneously, the method uses a virtual distorted grid network moving in parallel to the vapour/liquid interface in order to determine its position vis–a–vis the real grid network. The method has been tested by solving both the collapse of an adiabatic wall and a normalised two MB problem whose exact solution is known.

Nomenclature

a Slab thickness

C Heat capacity/unit volume

F Heat flux input

h Heat transfer coefficient

K Thermal conductivity

M Liquid/solid interface position

N Total number of space elements

t Time variable

t_m Time of melting

t_v Time of vaporisation

t_e Time of end of the process

T Temperature

T_m Melting temperature

T_v Vaporisation temperature

T_0 Temperature of the fluid in contact the slab

r_c Corrected Fourier number for the heating stage

r_{cl} Corrected Fourier number in the liquid region

r_f Fixed Fourier number (≤ 0.5)

x Space variable

V Vapour/liquid interface position

α Diffusivity

θ Temperature at the first virtual distorted line

φ Temperature at the second virtual distorted line

ϵ_{min} Time-step error allowed to stop iteration

λ_m Latent heat of fusion/unit volume

λ_v Latent heat of vaporisation/unit volume

Subscripts

i, j Space/time indices

i_m The node index holding the moving boundary

i_s The first node in the non-distorted grid just below the liquid/vapour interface

l Liquid

n Sub-interval time step index in the liquid region

p Number of sub-intervals in the liquid region for a given time step

q Number of sub-intervals in the solid region for a given time step

s Solid

Superscript

k Iteration index

6.1. Introduction

Most numerical methods dealing with moving boundary problems are generally conceived for one MB separating two phases [1,2]. Whilst single MB problems are of considerable utility in modelling a whole class of engineering problems, many engineering and physical problems must be modelled with two or more MBs. Heat conduction problems with heating of solid, melting and partial vaporisation of liquid is one such problem.

Bonnerot and Jamet [3] have introduced a conservative finite element method which can be used to solve such problems. They applied a modified and extended form of their third-order-accuracy discontinuous finite element method [4]. It adopts a curved triangular element for each appearing or disappearing phase, whilst a curved trapezoidal element is employed elsewhere (this method will be referred to as the BJ method). However, finite element techniques are time consuming and less amenable to vectorisation than finite difference methods, which continue to be widely used due to their simplicity in formulation and ease of programming.

Due to its excellent performance in solving moving boundary problems, the EVTS method [5] is extended to deal with multi-phase Stefan problems. The present scheme adopts a fixed time-space grid network during the pre–melting stage, a variable time step grid network during the melting and vaporisation stages and uses explicit finite difference replacements for the partial and ordinary differential equations. During the vaporisation period, which is the most difficult period to compute, the method uses the approach of fixing the Liquid/Solid (L/S) interface at a real space grid line and determines the time step iteratively. Having arranged the solution scheme, so that the L/S interface always coincides with a spatial grid line, a further problem is to determine the corresponding position of the Vapour/Liquid (V/L) interface. Therefore the method uses a virtual distorted grid network moving in parallel to the second MB; this permits the explicit determination of the position of the second MB as well as the temperature throughout the liquid region between the two moving boundaries.

As in EVTS, the Virtual Sub-Interval Elimination Technique (VSIET) [5] is incorporated throughout all the different stages of the computation – to ensure that stability is automatically maintained for any mesh size. As a result of incorporating the VSIET the accuracy of the present method is unaffected by the speed of the MBs.

The numerical method is presented for a general heat transfer problem which will be described in section 6.2. For validation, the method was used to solve the collapse of a solid wall due to a heat flux input at one boundary with an adiabatic condition at the other [3]. The problem involves heating of the solid, melting, vaporisation and finally complete collapse of the wall when the solid phase disappears. Numerical results are compared with those of the BJ finite element method and show a good agreement.

Furthermore, a normalised two MB problem – which has an exact analytical solution – was considered and numerical results show that the method achieves a good accuracy relative to the exact solution which is used as a reference standard.

6.2. Description of multiphase Stefan problems

Consider a solid material of infinite transverse dimensions and thickness a to be subjected to a variable source of heat $F(t)$, at one extremity ($x = 0$) and in contact with a fluid at a constant temperature T_0 at the other ($x = a$). Assuming that the heat transfer is one-dimensional and the material thermal properties are constant within each phase but differ from one to another; depending on the process duration, the following stages will occur:

6.2.1. Heating of the solid stage ($0 \le t \le t_m$; Figure 6.1a)

Due to the positive heat flux input, the temperature throughout the solid increases with no change of phase. The governing equations for this stage are:

$$C_s \frac{\partial T_s}{\partial t} = K_s \frac{\partial^2 T_s}{\partial x^2} \quad , \quad M(t) \le x \le a \quad , \quad 0 \le t \le t_e \tag{6.1}$$

$$-K_s \left(\frac{\partial T_s}{\partial x} \right) = F(t) \quad , \quad x = 0 \quad , \quad 0 \le t \le t_m \tag{6.2}$$

$$-K_s \left(\frac{\partial T_s}{\partial x} \right) = h\left(T_s(x,t) - T_0\right) \quad , \quad x = a \quad , \quad 0 \le t \le t_e \tag{6.3}$$

where $T(x,t)$ is the temperature at depth x time t; C, K, $\alpha = K/C$ and h are heat capacity per unit volume, thermal conductivity, diffusivity and convective heat transfer coefficient, respectively; $M(t)$, t_m and t_e denote the liquid/solid interface position, the time at which melting starts, and the time at which the heat transfer process ends, respectively.

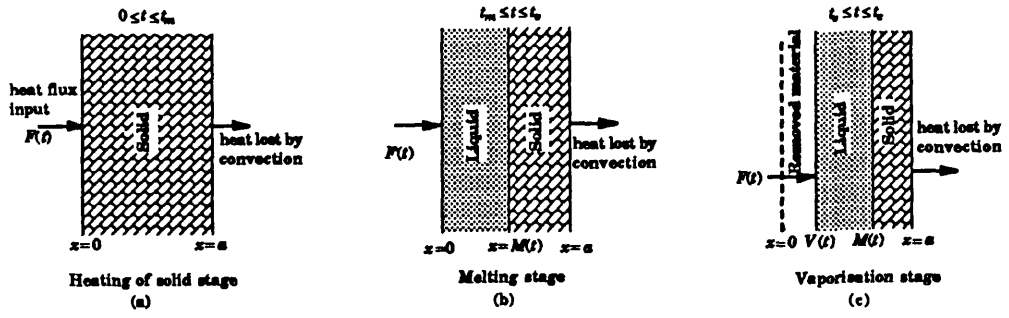

Figure 6.1: Three phase Stefan problem with mixed boundary conditions

6.2.2. Melting stage ($t_m \le t \le t_v$; Figure 6.1b)

When the surface temperature $T(0,t)$ reaches the melting point of the material (T_m) a MB, $x = M(t)$, appears which separates the liquid from the solid. It is assumed that the liquid and the solid have the same density and consequently there is no displacement of the liquid surface. The liquid occupies the region $0 \le x \le M(t)$ and the solid occupies the region $M(t) \le x \le a$; the liquid/solid interface $M(t)$ moves in the x direction and temperature increases throughout the solid and liquid regions. In addition to (6.1) and (6.3), the following equations are necessary to describe this stage:

$$C_l \frac{\partial T_l}{\partial t} = K_l \frac{\partial^2 T_l}{\partial x^2} \quad , \quad V(t) \le x \le M(t) \quad , \quad t_m \le t \le t_e \tag{6.4}$$

$$-K_l \left(\frac{\partial T_l}{\partial x} \right) = F(t) \quad , \quad x = 0 \quad , \quad t_m \le t \le t_v \tag{6.5}$$

$$\lambda_m \frac{dM}{dt} = K_s \left(\frac{\partial T_s}{\partial x} \right) - K_l \left(\frac{\partial T_l}{\partial x} \right) \quad , \quad x = M(t) \quad , \quad t_m \le t \le t_e \tag{6.6}$$

where λ_m, $V(t)$ and t_v are latent heat of fusion per unit volume, the vapour/liquid interface position and the time at which vaporisation starts, respectively. The subscripts l and s distinguish liquid and solid phases respectively.

6.2.3. Vaporisation stage ($t_v \le t \le t_e$; Figure 6.1c)

When the surface temperature $T(0,t)$ reaches the vaporisation point (T_v) of the material, a second MB, $x = V(t)$, appears which separates the liquid from the vapour. Assuming that the vapour is removed as soon as it appears (i.e. only two phases remain);

therefore, the liquid occupies the region $V(t) \le x \le M(t)$ and the solid occupies $M(t) \le x \le a$ (the ablation case, where the whole liquid region is instantaneously evaporated and the problem becomes a one moving boundary problem, is not considered in this work). The two MBs continue to propagate in the x direction and the temperature increases throughout the solid and liquid regions until the time (t_e) of the end of the heat transfer process. The final time for the test problems in section 6.4 is the time for complete melting (i.e. when the liquid/solid interface $M(t)$ has reached $x = a$). This stage is governed by (6.1), (6.3), (6.4), (6.6) and the following condition at the second MB, $x = V(t)$:

$$\lambda_v \frac{dV}{dt} = K_l \left(\frac{\partial T_l}{\partial x} \right) + F(t) \quad , \quad x = V(t) \quad , \quad t_v \le t \le t_e \tag{6.7}$$

where λ_v is the latent heat of vaporisation per unit volume.
The following conditions apply for all three stages:

$$\left.\begin{array}{lll} V(t) = 0 & , & t \le t_v \\ M(t) = 0 & , & t \le t_m \\ T(x,t) = T_m & , & x = M(t) \ , \ t \ge t_m \\ T(x,t) = T_v & , & x = V(t) \ , \ t \ge t_v \\ T(x,0) = T_0 & , & \forall x \in [0,a] \end{array}\right\} \tag{6.8}$$

6.3. Numerical computation scheme

In this section, a description of the numerical scheme for the problem described in section 6.2 is given in the order of the computational sequences, assuming that all the stages occurred during the process. As previously mentioned, the numerical method adopts a fixed time-space grid network during the heating of solid stage, and variable time step grid network in the melting and vaporisation stages. The total thickness a is divided into a fixed number of space intervals N, of length Δx, during the whole computation process.

Let j_m and j_v be the time step indices so that $t_{j_m} \le t_m \le t_{j_m+1}$ and $t_{j_v} \le t_v \le t_{j_v+1}$. The notation $T_{i,j}$ stipulates the value of T at the real grid point in the $(x-t)$ domain given by the coordinates:

$$\left(x_i = i\Delta x \quad , \quad t_j = \begin{cases} j\Delta t_0 & , \quad j < j_m \\ t_m + \displaystyle\sum_{q=j_m}^{j-1} \Delta t_q & , \quad j \ge j_m \end{cases} \right) \tag{6.9}$$

and V_j the value of V at the time t_j

6.3.1. Heating of solid stage

In this stage, where there are no phase changes, a fixed time step Δt_0 is chosen arbitrarily. The notation $\psi_{i,j,n,p}$ is used to indicate the value of the variable ψ at the virtual grid point in the $(x-t)$ domain given by the coordinates:

$$\left(x_i = i\Delta x \quad , \quad t_{j,n,p} = \left(j + \frac{n}{p}\right)\Delta t_0\right) \tag{6.10}$$

and the notation (j,n,p) represents the Virtual Sub-Step (VSS) at the time $t_{j,n,p}$, where n is the VSS index.

The temperature distribution at each VSS (j,n,p_0), where $\{1 \le j \le j_m\ , 1 \le n \le p_0\}$, is calculated from that of the previous VSS $(j,n-1,p_0)$ using:

$$T_{i,j,n,p_0} = T_{i,j,n-1,p_0} \exp\left(-r_c\ \psi_{i,j,n-1,p_0}\right) \quad , \quad i = 0,N \tag{6.11}$$

where

$$\psi_{i,j,n,p} = \begin{cases} \dfrac{2}{T_{i,j,n,p}}\left(T_{i,j,n,p} - T_{i+1,j,n,p} - \left(\dfrac{\Delta x}{K_s}\right)\overline{F}_{j,n,p}\right) & , \ i = 0 \\ \dfrac{1}{T_{i,j,n,p}}\left(2T_{i,j,n,p} - T_{i-1,j,n,p} - T_{i+1,j,n,p}\right) & , \ 1 \le i \le N-1 \\ \dfrac{2}{T_{i,j,n,p}}\left(\left(1 + \dfrac{h\Delta x}{K_s}\right)T_{i,j,n,p} - T_{i-1,j,n,p} - \left(\dfrac{h\Delta x}{K_s}\right)T_0\right) & , \ i = N \end{cases} \tag{6.12}$$

where

$$\overline{F}_{j,n,p} = \frac{1}{2}\left\{F\left(t_{j,n,p}\right) + F\left(t_{j,n+1,p}\right)\right\} \tag{6.13}$$

$$p_0 = \text{int}\left(\frac{\alpha_s}{r_f}\frac{\Delta t_0}{\Delta x^2}\right) + 1 \qquad , \qquad r_c = \frac{\alpha_s}{p_0}\frac{\Delta t_0}{\Delta x^2} \tag{6.14}$$

where "int(y)" defines the smallest integer less than y; r_f is an arbitrary fixed value of Fourier number ($r_f \le \frac{1}{2}$).

The temperature distribution at the last VSS $(j,n = p_0,p_0)$ of each time step is assigned to the real grid; the temperature distribution at the time $t = t_{j+1}$ is given by:

$$T_{i,j+1} = T_{i,j,p_0,p_0} \quad , \quad i = 0,N \tag{6.15}$$

In order to avoid stability problems when the rate of the heat flux input prescribed at $x = 0$ is relatively high, it is necessary to calculate the temperature distribution at the first time step $j = 1$ analytically (see appendix D.1).

Suppose at some intermediate VSS (j_m, n_m, p_0), the surface temperature exceeds the melting point T_m, then the time at which the melting starts (t_m) is given by:

$$t_m = j_m \Delta t_0 + (n_m - 1)\left(\frac{\Delta t_0}{p_0}\right) + \left(\frac{-\Delta x^2}{\alpha_s \psi_{0,j_m,n_m-1,p_0}}\right) \mathrm{Log_e}\left(\frac{T_m}{T_{0,j_m,n_m-1,p_0}}\right) \tag{6.16}$$

6.3.2. Melting stage

For sections 6.3.2 and 6.3.3 we use the notation $\psi^k_{i,j,n,p}$ to indicate the k^{th} iterated value of the variable ψ at the virtual grid point in the $(x-t)$ domain given by the coordinates

$$\left(x_i = i\Delta x \quad , \quad t^k_{j,n,p} = t_m + \sum_{q=j_m}^{j-1} \Delta t_q + n\frac{\Delta t^k_j}{p^k_j} \right) \tag{6.17}$$

where Δt_q is the time necessary for the moving boundary $M(t)$ to move from the position $x = x_{q-j_m}$ to $x = x_{q+1-j_m}$.

For this stage, the temperature throughout the liquid and solid regions as well as the position of the moving boundary $M(t)$ are computed using the EVTS method [5](explained fully in chapter 4). At some VSS (j_v, n_v, p), the surface temperature will exceed the vaporisation temperature (T_v), once this occurs the vaporisation time (t_v) is calculated using:

$$t^k_v = t_m + \sum_{q=j_m}^{j_v-1} \Delta t_q + \left(\frac{n_v - 1}{p^k_{j_v}}\right) \Delta t^k_{j_v} + \left(\frac{-\Delta x^2}{\alpha_l \psi^k_{0,j_v,n_v-1,p}}\right) \mathrm{Log_e}\left(\frac{T_v}{T^k_{0,j_v,n_v-1,p}}\right) \tag{6.18}$$

After determination of the time at which the vaporisation starts, computation is carried out similarly to melting until the condition given by (6.30) is verified. Consequently a liquid region of thickness $x = x_1$ near the surface will be superheated (i.e. temperature is above the vaporisation point).

6.3.3. Vaporisation stage

At the beginning of computation of this stage, the liquid region of thickness $x = x_1$ is superheated, equilibrium is established by vaporisation of the superheated liquid to a liquid depth of $x = x_2$. A mathematical expression similar to that suggested by Heitz and Westwater [6] and later developed by Hsu *et al.* [7] for a superheated solid is used:

$$x_2 = \frac{C_l}{\lambda_v} \int_0^{x_1} \left(T_l\left(x, t_{j_v+1}\right) - T_v \right) dx \tag{6.19}$$

Generally the superheated region extends less than one space increment, therefore $T_l\left(x, t_{j_v+1}\right)$ can be approximated by a polynomial of degree one in the region $0 \le x \le x_1$; hence, the depth of the initially vaporised material is determined directly without recourse to numerical integration.

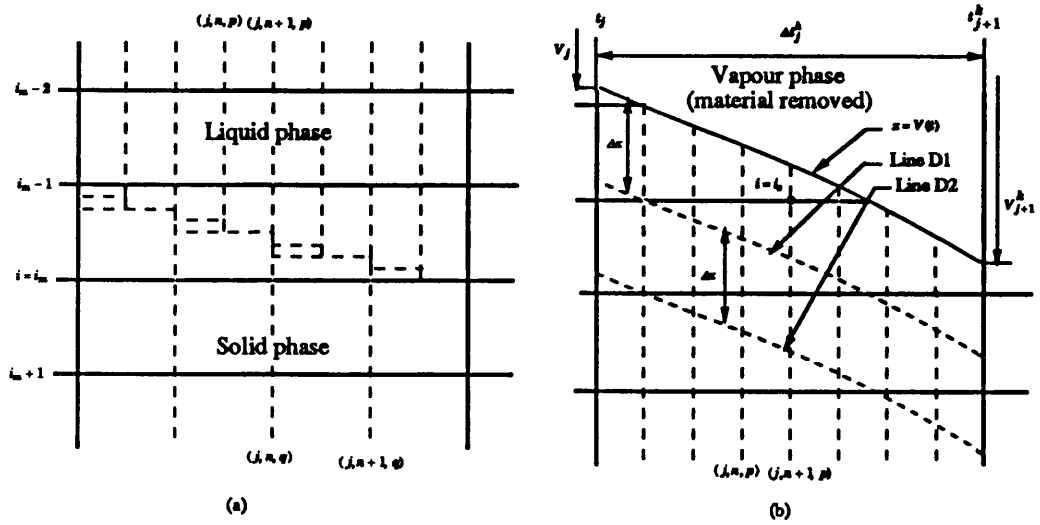

Figure 6.2: Discretisation around the moving boundaries. (a) Around the liquid/solid interface $M(t)$. (b) Around the vapour/liquid interface $V(t)$, together with both real (solid lines) and virtual (dashed lines) grid networks.

At any time during the vaporisation stage, where two MBs exist simultaneously, both the energy balance conditions given by (6.6) and (6.7) must be satisfied at each time step. The method insures that $M(t)$ moves a space increment Δx and determines iteratively the time step Δt necessary for that move. During each VSS at each iteration, the position of $V(t)$ is adjusted to satisfy the condition given by (6.7); hence, the problem is simplified, as in section 6.2.2, where the test for halting the iterations is carried out at the liquid/solid interface only. In order to achieve this, it is assumed that there is a virtual distorted grid

network $(y-t)$ formed by $V(t)$, and two lines (one at Δx and the other at $2\Delta x$ relative to $V(t)$) moving in parallel with $V(t)$ (see Figure 6.2b). To transform the equations from the original grid $(x-t)$ to $(y-t)$ the transformation $y=x+V(t)$ was used, applying the chain rule, (6.4) becomes:

$$\frac{\partial T_l}{\partial t}=\frac{K_l}{c_l}\frac{\partial^2 T_l}{\partial y^2}-\frac{dV}{dt}\frac{\partial T_l}{\partial y} \tag{6.20}$$

Writing equation (6.20) between two VSSs (j,n,p) and $(j,n+1,p)$, and assuming a linear propagation of $V(t)$ between each two VSSs; hence, dV/dt can be approximated by $\chi_{j,n,p}$ which represents the variation of $V(t)$ during the course of (j,n,p) and $(j,n+1,p)$, (6.20) becomes:

$$\frac{\partial T_l}{\partial t}=\alpha_l\frac{\partial^2 T_l}{\partial y^2}-\chi_{j,n,p}\frac{\partial T_l}{\partial y} \tag{6.21}$$

Following the procedure of Bhattacharya [8,9], (6.21) has a difference solution of the form of (6.11). The temperature $\theta^k_{j,n+1,p}$ at the first virtual distorted grid line D1, which is situated Δx from the position of the MB $x=V(t)$, at $(j,n+1,p)$ is given by:

$$\theta^k_{j,n+1,p}=\theta^k_{j,n,p}\exp\left(-r^k_{cl,j}\,\phi^k_{j,n,p}\right) \tag{6.22}$$

Where

$$\begin{aligned}\phi^k_{j,n,p}=&\left(\frac{\Delta x\,\chi^k_{j,n,p}}{2\alpha_l}\right)^2+\left(\frac{1}{\theta^k_{j,n,p}}\right)\left\{\left(2\theta^k_{j,n,p}-\varphi^k_{j,n,p}-T_v\right)\right.\\&\left.-\left(\frac{\Delta x\,\chi^k_{j,n,p}}{2\alpha_l}\right)\left(T_v-\varphi^k_{j,n,p}\right)-\left(\frac{\Delta x\,\chi^k_{j,n,p}}{2\sqrt{2}\alpha_l}\right)^2\left(T_v+\varphi^k_{j,n,p}\right)\right\}\end{aligned} \tag{6.23}$$

where $r^k_{cl,j}$ and p^k_j are defined in chapter 4.

The position of $V(t)$ is conditioned to satisfy the energy balance at $x=V(t)$; replacing (6.7) by a finite difference gives:

$$\chi^k_{j,n,p}=\left(\frac{\Delta t^k_j}{\lambda_v p^k_j}\right)\left\{\overline{F}_{j,n,p}-\frac{K_l}{\Delta x}\left(T_v-\theta^k_{j,n+1,p}\right)\right\} \tag{6.24}$$

Since $\chi^k_{j,n,p}$ is not known, to solve (6.22), (6.23) and (6.24), an iterative procedure is used starting with $\chi^{k,0}_{j,n,p} = \chi^k_{j,n-1,p}$. Due to the small length of sub–intervals, $\chi^k_{j,n,p}$ is generally determined in two iterations only.

After determination of $\chi^k_{j,n,p}$, the temperature at the surface node, $i = i_s$, which is the first node below $x = V(t)$ in the original (non-distorted) grid (see Figure 6.2b), at the VSS $(j, n+1, p)$ can be interpolated by using simple interpolation from the temperature at the virtual distorted line D1 and the temperature at the moving boundary $x = V(t)$:

$$T^k_{i_s,j,n+1,p} = \theta^k_{j,n+1,p}\left[1 - \mathrm{mod}\left(\frac{V^k_{j,n+1,p}}{\Delta x}, 1\right)\right] + T_v\left[\mathrm{mod}\left(\frac{V^k_{j,n+1,p}}{\Delta x}, 1\right)\right] \tag{6.25}$$

where $\mathrm{mod}(y,x) = y - x\,\mathrm{int}\left(\frac{y}{x}\right)$ and $V^k_{j,n+1,p}$ is given by:

$$V^k_{j,n+1,p} = V_{j-1} + \sum_{n=0}^{n} \chi^k_{j,n,p} \tag{6.26}$$

Now the computation of temperatures at the virtual grid points, $i > i_s$, throughout the liquid are carried out in a similar manner to section 6.2.2.

It can be seen from (6.23) that for the next VSS, the temperature $\varphi^k_{j,n+1,p}$, at the second virtual distorted line D2 at $(j, n+1, p)$, must be known. In order to minimise the cpu-time, it is calculated by a simple interpolation from temperatures at the original (non-distorted) grid and it is given by:

$$\varphi^k_{j,n+1,p} = T^k_{i_s+2,j,n+1,p}\left[1 - \mathrm{mod}\left(\frac{V^k_{j,n+1,p}}{\Delta x}, 1\right)\right] + T^k_{i_s+1,j,n+1,p}\left[\mathrm{mod}\left(\frac{V^k_{j,n+1,p}}{\Delta x}, 1\right)\right] \tag{6.27}$$

The position of $V(t)$ and the temperature distribution throughout the liquid and solid regions (computation of temperature distribution throughout the solid region is similar to section 6.2.2) at the time $t = t^k_{j+1}$ are given by:

$$V^k_{j+1} = V_j + \sum_{n=0}^{p^k_j - 1} \chi^k_{j,n,p} \tag{6.28}$$

$$T^k_{i,j+1} = \begin{cases} T^k_{i,j,p^k_j,p} & , \quad i = i_s, i_m - 1 \\ T_m & , \quad i = i_m \\ T^k_{i,j,q^k_j,q} & , \quad i = i_m + 1, N \end{cases} \tag{6.29}$$

Once the temperature distribution throughout the solid and liquid regions at the time $t = t_{j+1}^k$, has been estimated from (6.29), all the intermediate time steps are virtual and are progressively eliminated (see the algorithm in appendix D.2); the criterion used for convergence of the temperature field was:

$$\left|\varepsilon_j^k\right| \leq \varepsilon_{min} \tag{6.30}$$

where ε_{min} is the time step error allowed to stop iterations and ε_j^k is given by:

$$\varepsilon_j^k = 100 \times \frac{\Delta t_j^k - \xi_j^k}{\Delta t_j^k} \tag{6.31}$$

where

$$\xi_j^k = \left\{ \left(\frac{1}{\lambda_m \Delta x^2} \right) \left[K_l \left(T_{i_m - 1, j+1}^k - T_m \right) - K_s \left(T_m - T_{i_m + 1, j+1}^k \right) \right] \right\}^{-1} \tag{6.32}$$

If equation (6.30) is not satisfied, a relaxation procedure is applied to determine the time step for the $(k+1)^{th}$ iteration:

$$\Delta t_j^{k+1} = \Delta t_j^k \left(1 - \frac{\varepsilon_j^k}{100\,\omega} \right) \tag{6.33}$$

where ω is a relaxation factor and $\Delta t_j^0 = \Delta t_{j-1}$.

***Remark* 1**: When the energy balance at $x = M(t)$ takes into account the temperature gradient in one phase only as in test problem 2 of section 6.4, the time step at the next iteration Δt_j^{k+1} can be determined by a similar procedure to that suggested by Gupta and Kumar [10] (i.e. $\Delta t_j^{k+1} = \xi_j^k$). For the case of two MBs such as problem 2, the procedure may converge in certain cases and may not in others, however, equation (6.33) is general and has been satisfactory for all the computations performed.

***Remark* 2**: Experiments show that a given value of ω may give satisfactory convergence for some time steps and may exhibit non-convergence oscillation for others. In these circumstances such oscillatory behaviour can be avoided by evaluating the time step for the $(k+1)^{th}$ iteration using: $\Delta t_j^{k+1} = \frac{1}{2}\left(\Delta t_j^k + \xi_j^k \right)$ [11].

6.4. Test problems

Problem 1: Collapse of a solid wall

The method described in section 6.3 has been used to solve the collapse of a solid wall due to a constant heat flux input F at $x = 0$, whilst the other extremity $x = a$ is thermally insulated (i.e. $\partial T / \partial x = 0$ at $x = a$)[3]. Solving this problem is similar to solving the problem previously described in section 6.2 but with a convective heat transfer coefficient $h = 0$. Bonnerot and Jamet [3] demonstrated the solution of the problem with two different sets of data.

Problem 1a:

$$\begin{gathered} a = 1 \ , \ T(x,0) = 27 \ , \ F = 2500 \ , \ C_l = C_s = 4.944, \\ K_l = K_s = 0.259 \ , \ \lambda_m = 2160 \ , \ \lambda_v = 37200, \\ T_m = 1454 \ , \ T_v = 3000 \end{gathered} \tag{6.34}$$

This data set permits all three stages (i.e. heating of solid, melting and vaporisation) to occur. For the present method a choice of $\Delta x = 0.02$ was used so as to have approximately the same total number of grid points as the BJ method, where $\Delta t / \Delta x = 1/16$. For the present method, which will be referred to as the ZC method, a relaxation factor $\omega = 3.0$, $\varepsilon_{min} = 0.05\%$ and $r_f = 0.30$ have been used. A comparison of results using the BJ method and the ZC method are shown in Table 6.1, and Figures 6.1, 6.2, and 6.3, these are discussed in section 6.5.

Remark 3: Since the time step during the melting and vaporisation stages for the ZC method is variable, the Δt quoted in Table 6.1 and 6.2 is the Δt_0 defined in section 6.3.1.

Problem 1b:

$$\begin{gathered} a = 1 \ , \ T(x,0) = 27 \ , \ F = 2500 \ , \ c_l = c_s = 1.041\rho, \\ K_l = 1.73 \ , \ K_s = 0.865 \ , \ \lambda_m = 400\rho \ , \ \lambda_v = 10700\rho, \\ T_m = 638 \ , \ T_v = 2480 \ , \ \rho = 2.77 \end{gathered} \tag{6.35}$$

where ρ is the material density.

With this data set, the melting interface reaches the adiabatic boundary before vaporisation occurs; the process is therefore reduced to a single MB problem. The same parameters (ZC $\{\Delta x, r_f, \omega, \varepsilon_{min}\}$, BJ $\{\Delta t / \Delta x, \varepsilon\}$) have been used as in problem 1a. the comparison of results of the ZC and BJ methods are shown in Table 6.2 and Figures 6.4, 6.5 and 6.6, these are discussed in section 6.5.

Problem 2

In order to further validate the accuracy of the method, the following normalised two MB problem, whose exact analytical solution is known, was considered. A solid material melts due to a variable heat flux input applied at the liquid surface, the temperature throughout the solid region is always assumed to remain at the melting point. The governing equations are:

$$\frac{\partial T_l}{\partial t} = \alpha_l \frac{\partial^2 T_l}{\partial x^2} \quad , \quad V(t) \le x \le M(t) \quad , \quad 0 \le t \le t_e \tag{6.36}$$

$$\frac{\partial T_l}{\partial x} = -\exp(\alpha_l t) \quad , \quad x = 0 \quad , \quad 0 \le t \le t_v \tag{6.37}$$

$$\frac{1}{\alpha_l}\frac{dM}{dt} = -\frac{\partial T_l}{\partial x} \quad , \quad x = M(t) \quad , \quad 0 \le t \le t_e \tag{6.38}$$

$$\frac{T_v}{\alpha_l}\frac{dV}{dt} = \exp(\alpha_l t_v) \quad , \quad t_v \le t \le t_e \tag{6.39}$$

$$T_l(x,t) = \begin{cases} T_m = 1.0 \ , M(t) \le x \le 1.0 \ , \ t \ge 0 \\ T_v \qquad\quad , \ x = V(t) \ , \ t \ge t_v \end{cases} \tag{6.40}$$

$$M(0) = 0 \quad , \quad T_l(x,0) = 1.0 \quad \text{for} \quad 0 \le x \le 1.0 \tag{6.41}$$

The exact analytical solution is given by:

$$T_l(x,t) = \exp(\alpha_l t - x) \quad , \quad M(t) = \alpha_l t \quad , \quad V(t) = \alpha_l (t - t_v)$$
$$t_v = \frac{1}{\alpha_l}\mathrm{Log_e}(T_v) \quad \text{and} \quad t_e = \frac{1}{\alpha_l} \tag{6.42}$$

Since in this particular problem the condition at the second moving boundary (vapour/liquid) is not coupled to the temperature gradient in the liquid region; the virtual distorted grid network is omitted in order to reduce the cpu-time; therefore the temperature near the vapour/liquid interface is computed in a similar manner to that near the liquid/solid interface. Numerical results when $T_v = 1.5 T_m$, $\omega = 2.0$, $r_f = 0.35$ and $\varepsilon_{min} = 0.01\%$ are shown in Table 6.3 and Figures 6.9 and 6.10.

For accuracy assessment, the Maximum Percentage Deviation for Temperature (*MPDT*) is defined as:

$$MPDT = \max\left\{\left(\left(\left|PDT_{i,j}\right|\right), i = 0, N\right), j = 0, j_e\right\} \tag{6.43}$$

where $PDT_{i,j}$ is the percentage deviation from the analytical temperature calculated at the nodal point (x_i, t_j) which is given by:

$$PDT_{i,j} = 100 \times \frac{T_{Nu}(x_i,t_j) - T_{An}(x_i,t_j)}{T_{An}(x_i,t_j)} \tag{6.44}$$

The Maximum Percentage Deviation for time (*MPDt*) is given by:

$$MPDt = \max\left\{\left(\left|PDt_j\right|\right), j = 0, j_e\right\} \tag{6.45}$$

where

$$PDt_j = 100 \times \frac{(t_j)_{Nu} - (t_j)_{An}}{(t_j)_{An}} \tag{6.46}$$

The Percentage Deviation for Vaporisation (*PDV*) interface position is given by:

$$PDV_j = 100 \times \frac{(V_j)_{Nu} - (V_j)_{An}}{(V_j)_{An}} \tag{6.47}$$

where the subscripts An and Nu refer to the value calculated Analytically and Numerically, respectively.

6.5. Numerical results and discussion

Numerical results show that for problem 1a, which corresponds to a low conductivity material compared to that used in problem 1b, the vaporisation occurs before the liquid/solid interface reaches the adiabatic boundary. On the contrary, in problem 1b, the wall collapses before the appearance of a vapour/liquid interface and therefore the problem is one of melting only.

Table 6.1, figures 6.3, 6.4 and 6.5 show that for problem 1a, both the BJ and ZC solutions are in good agreement; a maximum relative error of only 2% occurs between the two solutions. For problem 1a, the BJ method has a cpu-time of 40 seconds, whereas the ZC solution takes only 9 seconds (non-vectorised program) on an IBM3090/150VF (Computer Centre, University of Glasgow). The ZC scheme is more than 4 times faster than the BJ method. Using vectorisation, the speed up factor due to vectorisation varies from 2.5 to 3.5 for the BJ method, whereas it varies from 5 to 6 for the ZC method. This however, is due to the fact that explicit methods are more suitable for vectorisation than implicit finite

Table 6.1 (Problem 1a): Comparison of t_m, t_v, t_e and $V(t_e)$ for different time increments Δt.

Δt	t_m	t_v	t_e	$V(t_e)$
BJ method [3]				
1/4	0.36150	1.63034	9.75464	0.28351
1/8	0.32897	1.63611	9.40210	0.26703
1/16	0.32767	1.63446	9.38719	0.26633
1/32	0.32768	1.63439	9.38708	0.26632
ZC method				
1/4	0.32740	1.58887	9.15907	0.26318
1/8	0.32660	1.58783	9.15828	0.26319
1/16	0.32580	1.58705	9.15745	0.26318
1/32	0.32494	1.58620	9.15670	0.26319

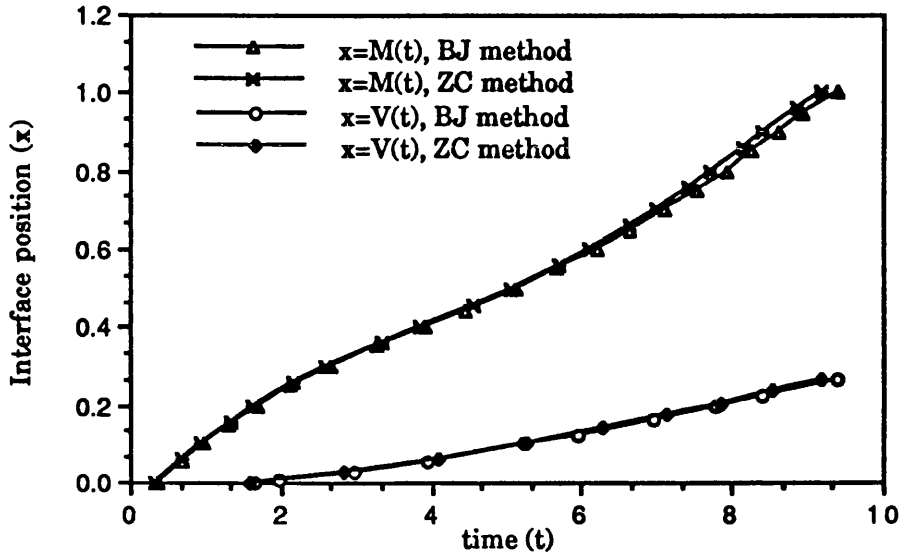

Figure 6.3 (Problem 1a): Position of the liquid/solid and vapour/liquid interfaces versus time

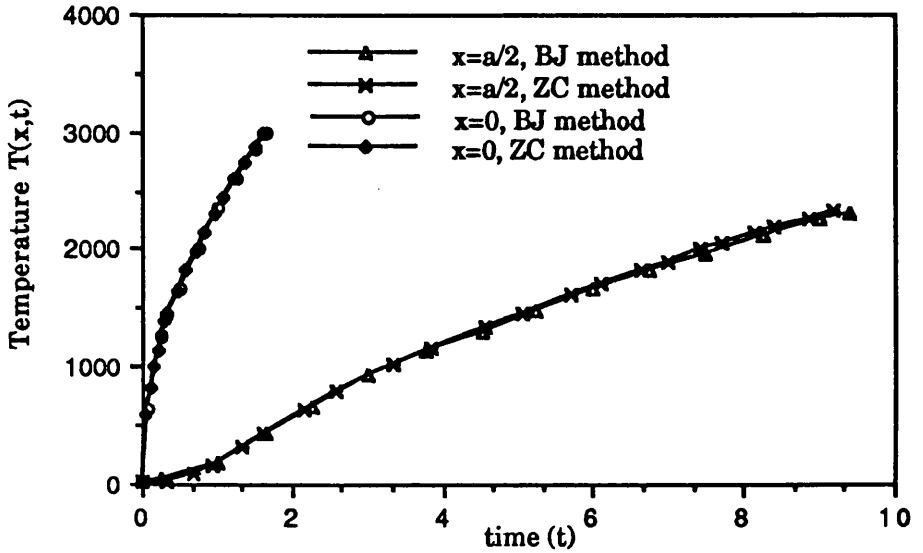

Figure 6.4 (Problem 1a): Temperature versus time at different depths

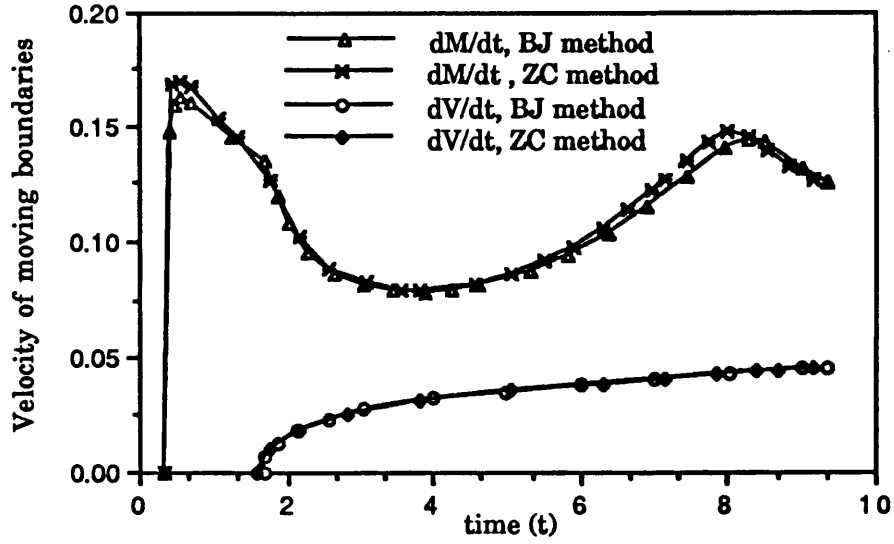

Figure 6.5 (Problem 1a): Velocity of the moving boundaries versus time

difference and finite element methods [12] where the unknown variables are determined with recurrent do–loops (i.e. do–loops that form a cycle of variable dependencies).

Figure 6.5 shows the velocity history of the two MBs for problem 1a, where the L/S interface starts with zero velocity and reaches a maximum speed just after its appearance; its propagation speed then starts to decrease due the greater thermal-resistance of the liquid. When the V/L interface appears, the L/S interface slows sharply due the heat loss from vaporisation. The V/L interface accelerates sharply in the initial instance, then continues to accelerate at a lower rate throughout the process. The L/S interface velocity reaches a minimum then starts to increase again, this is due to the fact that the heat flux entering the solid region decreases sharply when approaching the adiabatic extremity (i.e. there is less energy absorbed as the whole solid approaches the melting temperature). The heat flux at the L/S interface decreases with a smaller gradient than the heat flux entering the solid region resulting in a greater proportion of the energy being available to melt the solid material; meanwhile, the V/L interface converges towards a constant acceleration due to the constant heat flux input. Before the total melting or the collapse of the wall, the L/S interface velocity starts to decrease due to the decreasing heat flux through the liquid and the absence of heat flux in the solid region.

As with problem 1a there is good agreement between the two solutions for problem 1b, this is shown in Table 6.2, Figures 6.6, 6.7 and 6.8. Unlike problem 1a, the material used in problem 1b is a good conductor of heat, therefore, the melting process is faster than problem 1a. This can be seen from Figure 8 where the velocity of the L/S interface is much higher than that predicted by Figure 6.3; the wall collapses at t=1.8 rather than t=9.2 for problem 1a.

Table 6.2 (Problem 1b): Comparison of t_m, t_e and $T(0,t_e)$ for different time increments Δt.

Δt	t_m	t_e	$T(0,t_e)$
BJ method [3]			
1/4	0.23485	1.92536	2349.18
1/8	0.23401	1.92383	2347.98
1/16	0.23400	1.92381	2347.96
ZC method			
1/4	0.23352	1.89063	2341.45
1/8	0.23398	1.88747	2341.40
1/16	0.23390	1.88719	2341.40

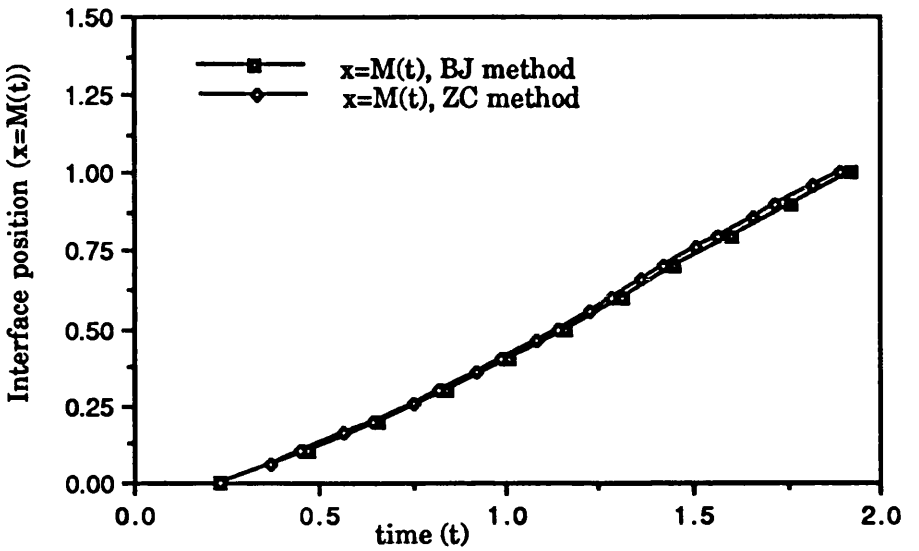

Figure 6.6 (Problem 1b): The liquid/solid interface position versus time

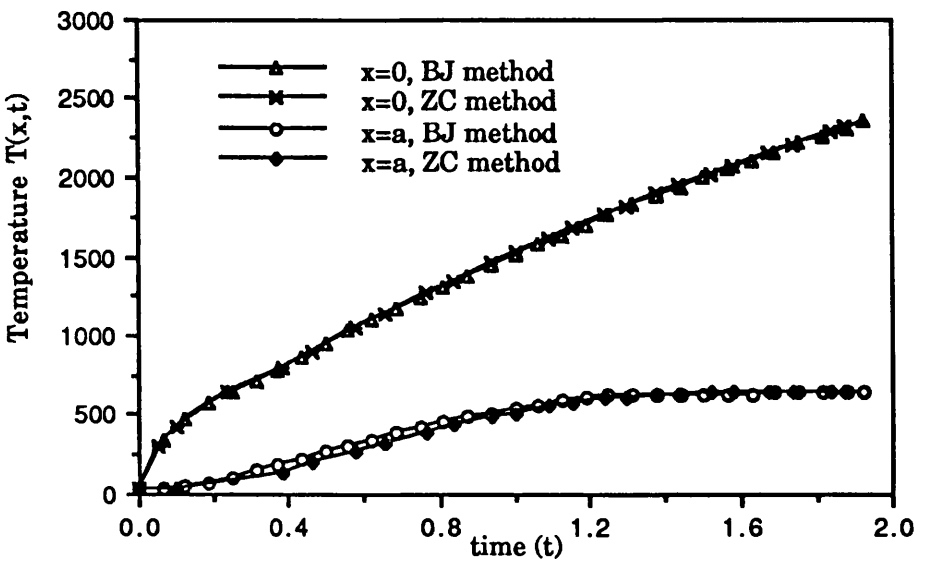

Figure 6.7 (Problem 1b): Temperature versus time at different depths

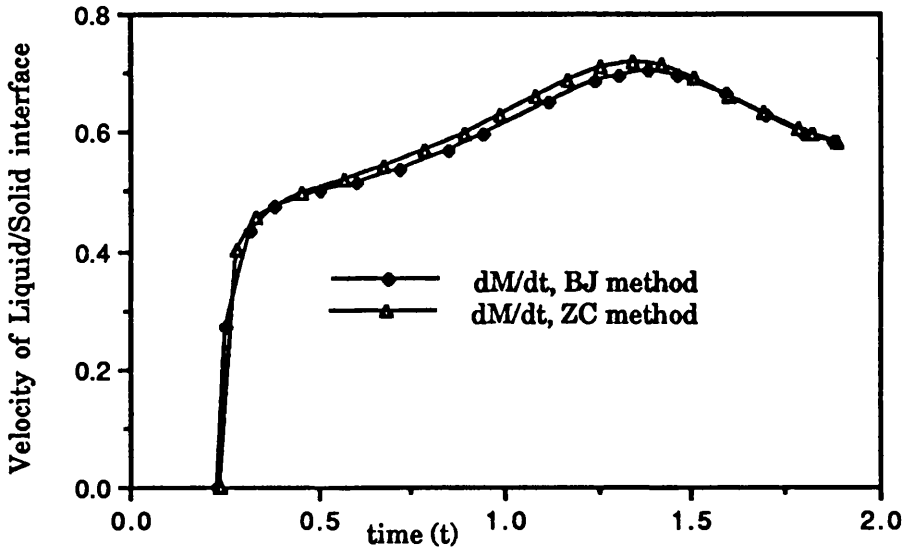

Figure 6.8 (Problem 1b): Velocity of the liquid/solid interface versus time

Unlike the behaviour of L/S interface in Figure 6.3, the melting interface for problem 1b always accelerates due to the higher conductivity of the liquid compared to that of the solid. At the end of the process the L/S interface has a similar behaviour to that of problem 1a due to the adiabatic extremity.

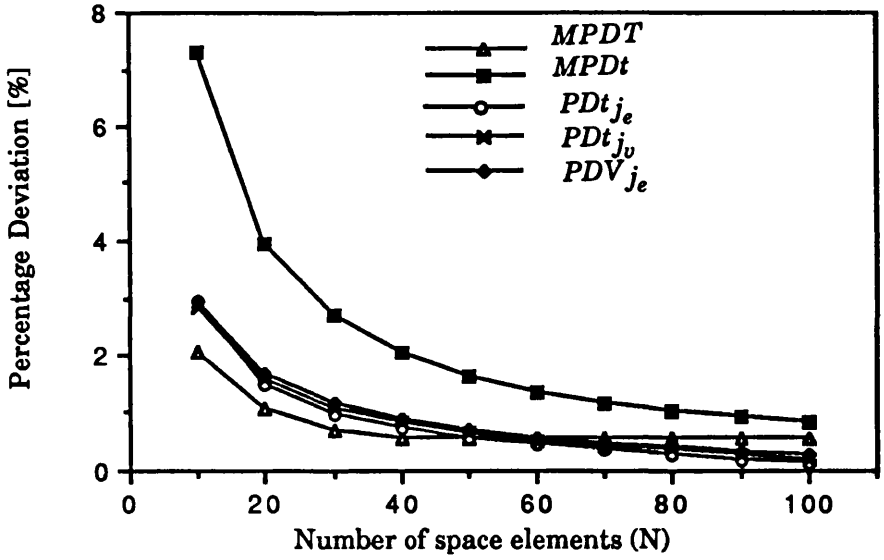

Figure 6.9 (Problem 2, α=0.1): Variation of percentage deviations from the analytical solution with the total number of space elements N

Figure 6.9 shows that $MPDT$, $MPDt$, PDt_{j_v}, PDt_{j_e} and PDV_{j_e} all decrease as N increases for problem 2. In other words, the accuracy of the ZC method increases by increasing the total number of space elements which results in greater cpu-time (as shown in Figure 6.10).

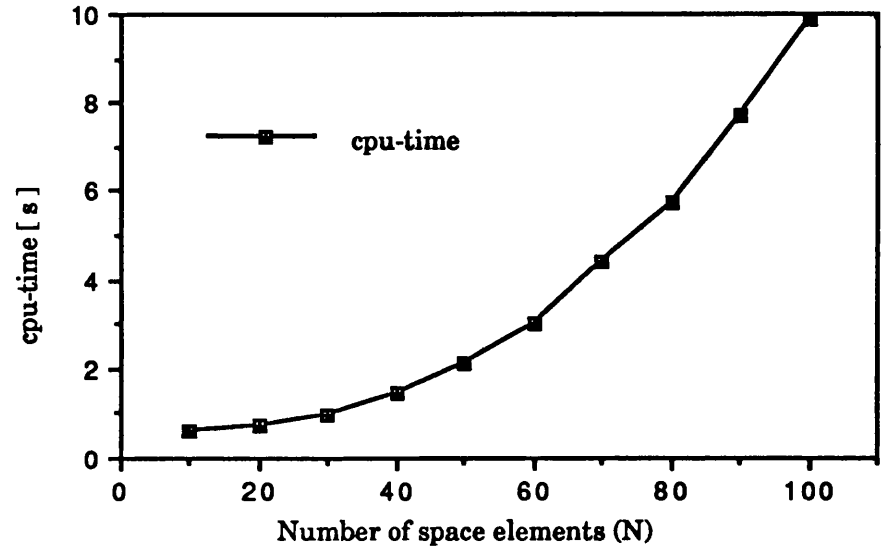

Figure 6.10 (Problem 2, α=0.1): Variation of cpu-time with the number of space elements N

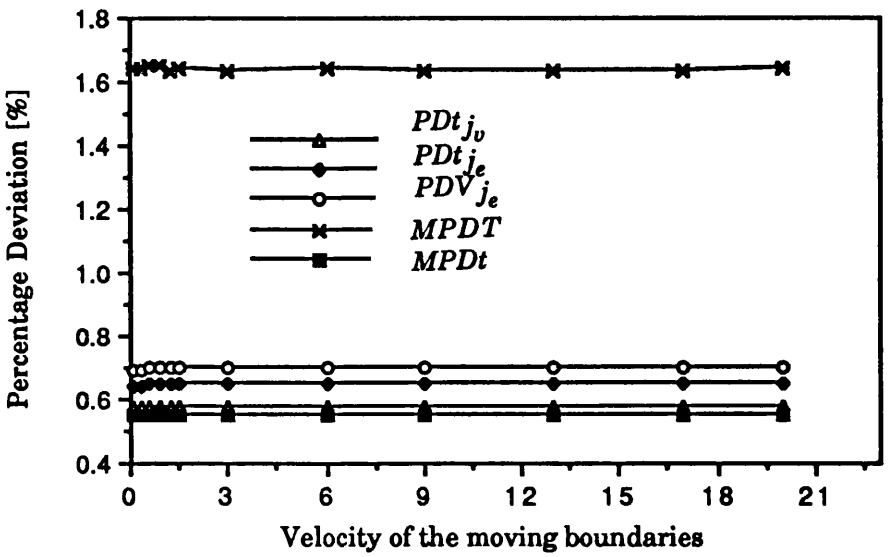

Figure 6.11 (Problem 2, Δx=0.02): Variation of percentage deviations from the analytical solution with the velocity of moving boundaries

Figure 6.9 also shows that despite the singularities that appear with each moving boundary, the accuracy is very satisfactory. Maximum errors for the different variables of the solution are always generated at the appearance of a MB, these errors decrease with time as shown in Table 6.3.

Table 6.3 (Problem 2): A comparison of computed history of the moving boundaries $M(t)$ and $V(t)$ with the analytical solution. Percentage deviations are also tabulated.

$M(t)$	t_{Nu}	t_{An}	PDt	V_{Nu}	V_{An}	PDV
0.1	0.492868	0.5000	-1.426458			
0.2	0.987501	1.0000	-1.249921			
0.3	1.482948	1.5000	-1.136717			
0.4	1.979041	2.0000	-1.047945			
0.5	2.475761	2.5000	-0.969506	0.092003	0.094535	-2.677884
0.6	2.973228	3.0000	-0.892385	0.191497	0.194535	-1.561637
0.7	3.471273	3.5000	-0.820732	0.291106	0.294535	-1.163964
0.8	3.969764	4.0000	-0.755906	0.390805	0.394535	-0.945447
0.9	4.468581	4.5000	-0.698196	0.490569	0.494535	-0.801985
1.0	4.967664	5.0000	-0.646725	0.590385	0.594535	-0.697940

Due to the incorporation of the VSIET, the accuracy of the present scheme is unaffected by the nature of the heat transfer problem; this can be seen from Figure 6.11 where the percentage deviation for the different variables in the solution remain constant with the variation of moving boundary velocity. Therefore the method presented can be adapted to any heat transfer problem irrespective of its nature and good accuracy can be achieved by tuning the parameters r_f, Δx and ε_{min}.

6.6. Conclusion

Vectorisation is an essential tool to increase performance and reduce the cpu–time of large scale computation; explicit methods, where the unknown variables are determined with no recurrence (i.e. group of statements that form a cycle of variable dependencies), are highly suitable for vectorisation. However, if computations are to be performed for extended times, explicit methods do have instability constraints which make higher demands on computer memory requirements due to the large amount of data that must be stored.

The ZC method combines the variable time step approach and explicit procedures. The variable time-step approach requires less memory and storage requirements than a fixed grid network; this results in highly efficient use of the computational platform. Additionally, the explicit computation procedure permits to maximise the percentage of vectorisation of the scheme. By incorporating the VSIET the stability of the schemes is automatically maintained for any mesh size without increasing the storage requirements.

Other benefits of the method are simplicity and ease of programming, in the ZC method only simple linear equations must be solved compared to many numerical integrations and the solution of systems of linear equations for the BJ method.

6.7. References

1. J.R. Ockendon and W.R. Hodgkins (eds.), ***Moving Boundary Problems in Heat Flow and Diffusion,*** Clarendon Press, Oxford (1979).
2. J. Crank (ed.), ***Free and moving boundary problems,*** Clarendon Press, Oxford (1984).
3. R. Bonnerot and P. Jamet, *J. Comput. Phys.* **41**, 357-388 (1981).
4. R. Bonnerot and P. Jamet, *J. Comput. Phys.* **32**, 145-167 (1979).
5. M. Zerroukat and C.R. Chatwin, *Int. J. Numer. Meth. Eng.* **35**, 1503-1520 (1992).
6. W.L. Heitz and J.W. Westwater, *Int. J. Heat Mass Transfer* **13**, 1371-1375 (1970).
7. S.C. Hsu, S. Chakravorty and R. Mehrabian, *Metall. Trans.* **9B**, 221-229 (1978).
8. M.C. Bhattacharya, *Communications Appl. Numer. meth.* **6**, 173-184 (1990).
9. M. C. Bhattacharya, *Int. J. Numer. Meth. Eng.* **21**, 239-265 (1985).
10. R.S. Gupta and D. Kumar, *Int. J. Heat Transfer* **24**, 251-259 (1981).
11. M. Zerroukat, Technical Report TR/GLU/MZ920820, Dept. Mech. Eng., University of Glasgow, (1992).
12. M. Zerroukat, Technical Report TR/GLU/MZ920830, Dept. Mech. Eng., University of Glasgow, (1992).

7 Heat Flow During Laser Heat Treatment of Materials

Summary

A new explicit finite difference equation for solution of the non–linear heat conduction equation is developed and validated by solving a normalised problem which has an analytical solution. A virtual sub–interval elimination technique is incorporated into the algorithm to insure the stability of the scheme irrespective of the mesh size; hence, avoiding the necessity for large array sizes. The finite difference scheme is then utilised to simulate the melting and re–crystallisation of silicon when subjected to radiation by a scanning cw–laser beam. The model incorporates temperature dependence of material properties and the surface reflectivity; the variation of the incident power density with time is also taken into account. The numerical scheme is also applied to the laser heating and melting of nodular cast iron; the results are compared with published experimental results.

Nomenclature

a, b, c, d	Constants
C	Heat capacity/unit volume of material
c_p	Heat capacity of air
F	Laser power density
g	Gravity acceleration
h	Heat transfer coefficient
K	Thermal conductivity of material
M	Liquid/solid interface position
MPD	Maximum percentage deviation
N	Total number of space elements
t_m	Time of melting
t_e	Time of end of the process
$T(x,t)$	Temperature at depth x, time t
T_m	Melting temperature
T_0	Temperature of air
r	Fourier number
R	Reflectivity of the surface material
r_f	Fixed Fourier number (≤ 0.5)
P	Total incident power of the laser beam
PD	Percentage deviation
X	Thickness of the work-piece
v	Scanning speed
κ	Thermal conductivity of air
σ	Stefan–Boltzman constant
μ	Dynamic viscosity of air
ρ	Density of air
α	Diffusivity
Δx	Space element
Δt	Time step length
λ	Latent heat of fusion/unit volume
ω	Laser beam diameter
Subscripts	
i, j	Space / time indices
l, s	Liquid / Solid
Superscript	
n	Virtual time step index
k	Last virtual time step index for a given real time step

7.1. Introduction

Laser processing of surfaces, which involves rapid localised heating of materials to modify its physical properties by altering the surface structure or distribution of impurities, has found many applications in manufacturing and particularly in semi–conductor technology [1–3]. Typically a high–power laser beam is utilised to rapidly heat a thin surface region, with the underlying bulk substrate providing self quenching.

To achieve a desired effect on a specific material it is necessary to make the optimum choice of process parameters such as: power, scan velocity and the beam diameter. To avoid a costly time–consuming experimental trial–and–error search to obtain the optimal combination; and in order to assess the influence of the laser parameters on the process, it is necessary to construct a mathematical model describing the laser–material interaction process.

Laser materials processing technology allows beam–workpiece interaction times ranging from a few picoseconds duration to several seconds; the concomitant incident energy densities, which induce observably reproducible effects, range from several mJ/cm^2 to some thousands of J/cm^2. For processes with high power and picoseconds interaction times, the heat diffusion length becomes comparable both to the radiation absorption length – which is of the order of 20 nm in most metals – and to the mean free path l of the heat carriers (i.e. the conduction electrons for which l=20 nm at room temperature). Under these extreme conditions, the linearity between the heat flux and temperature gradient breaks down; for these circumstances the energy transport into the material can be described using the kinetic theory rather than the Fourier conduction theory [4]. However, for interaction times greater than some picoseconds, the heat conduction process is correctly described by the Fourier conduction equation [5].

A new explicit finite difference equation for the heat conduction equation – where thermal properties are a function of temperature – is developed, and validated by solving a normalised problem which has an analytical solution. In order to eliminate the constraint of using a small mesh size, a virtual sub–interval elimination technique is incorporated to insure the stability of the scheme irrespective of the mesh size without loss of accuracy. The finite difference scheme is then utilised to simulate the melting and re–crystallisation of silicon when subjected to radiation by a scanning cw–laser beam. The process is physically described as follows: a laser beam moving with a uniform velocity scans a silicon substrate of infinite length and width but definite depth. The incident power is partly reflected and partly absorbed, depending on the reflectivity of the surface. Some of the absorbed energy is lost by re–radiation and convection from both the upper and lower surfaces into the surrounding environment, whilst the rest is conducted throughout the substrate. When the beam has completely passed, the melted zone is self quenched by the solid substrate and the heat lost from the surfaces. The numerical

scheme is also applied to the laser heating and melting of nodular cast iron; the results are compared with published experimental results.

7.2. Finite difference solution for non-linear heat conduction problems

The one-dimensional heat conduction equation in homogenous material with variable thermal properties is given by:

$$C(T)\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}\left(K(T)\frac{\partial T}{\partial x}\right) \tag{7.1}$$

where $T(x,t)$, $C(T)$ and $K(T)$ are temperature at the coordinates (x,t), heat capacity/unit volume and thermal conductivity respectively. Equation (7.1) can also be written as:

$$C(T)\frac{\partial T}{\partial t}=K(T)\frac{\partial^2 T}{\partial x^2}+\frac{\partial K}{\partial x}\left(\frac{\partial T}{\partial x}\right) \tag{7.2}$$

Writing (7.2) for the grid line i, situated at $x_i = i\Delta x$ relative to $x=0$, gives:

$$\left(\frac{\partial T}{\partial t}\right)_i=\frac{K(T_i)}{C(T_i)}\left(\frac{\partial^2 T}{\partial x^2}\right)_i+\frac{1}{C(T_i)}\left(\frac{\partial K}{\partial x}\right)_i\left(\frac{\partial T}{\partial x}\right)_i \tag{7.3}$$

Using a three point Lagrange formula, for the grid lines $(i\text{-}1)$, i and $(i+1)$; $(\partial K / \partial x)_i$ can be approximated by:

$$\left(\frac{\partial K}{\partial x}\right)_i=\frac{K_{i+1}-K_{i-1}}{2\Delta x} \tag{7.4}$$

Making use of the following notations:

$$\frac{1}{C(T_i)}\left(\frac{\partial K}{\partial x}\right)_i=-\frac{K_{i-1}-K_{i+1}}{2\Delta x C_i}=-\gamma_i \quad , \quad \frac{K(T_i)}{C(T_i)}=\frac{K_i}{C_i}=\alpha_i \tag{7.5}$$

equation (7.3) is written as:

$$\left(\frac{\partial T}{\partial t}\right)_i=\alpha_i\left(\frac{\partial^2 T}{\partial x^2}\right)_i-\gamma_i\left(\frac{\partial T}{\partial x}\right)_i \tag{7.6}$$

Following the procedure of Bhattacharya [6]; the finite difference solution of (7.6) – expressing the temperature at the node ($x_i = i\Delta x, t_{j+1} = t_j + \Delta t$), as a function of those at nodes (x_{i-1}, t_j), (x_i, t_j) and (x_{i+1}, t_j)– is given by:

$$T_{i,j+1} = T_{i,j} \exp\left(-r_{i,j}\, \psi_{i,j}\right) \tag{7.7}$$

where

$$\psi_{i,j} = \left(\frac{1}{T_{i,j}}\right)\left\{2T_{i,j} - \left(1+\beta_{i,j}\right)T_{i-1,j} - \left(1-\beta_{i,j}\right)T_{i+1,j}\right\} \tag{7.8}$$

where

$$\beta_{i,j} = \frac{K_{i-1,j} - K_{i+1,j}}{4K_{i,j}} \quad , \quad \alpha_{i,j} = \frac{K_{i,j}}{C_{i,j}} \quad , \quad r_{i,j} = \alpha_{i,j}\frac{\Delta t}{\Delta x^2} \tag{7.9}$$

$\psi_{i,j}$ can also be evaluated using the approach in [7] yielding to a more lengthy equation:

$$\psi_{i,j} = \beta_{i,j}^2 + \left(\frac{1}{T_{i,j}}\right)\left\{\begin{array}{l} 2T_{i,j} - \left(1+\beta_{i,j}\right)T_{i-1,j} - \\ \left(1-\beta_{i,j}\right)T_{i+1,j} - \dfrac{\beta_{i,j}^2}{2}\left(T_{i-1,j} + T_{i+1,j}\right) \end{array}\right\} \tag{7.10}$$

When $T_{i,j} \to 0$, $T_{i,j}\exp(-r_{i,j}\psi_{i,j}) \to T_{i,j} - r_{i,j}\psi_{i,j}T_{i,j}$; this eliminates $T_{i,j}$ from denominator of $\psi_{i,j}$, given by either (7.8) or (7.10); replacing $T_{i,j} = 0$ in what remains of the equation yields the finite difference equation for the special case of $T_{i,j} = 0$.

The classical finite difference equation of (7.1) is given by [8]:

$$C_{i,j}\frac{T_{i,j+1} - T_{i,j}}{\Delta t} = \frac{1}{\Delta x}\left(K_{i-1/2,j}\frac{T_{i-1,j} - T_{i,j}}{\Delta x} - K_{i+1/2,j}\frac{T_{i,j} - T_{i+1,j}}{\Delta x}\right) \tag{7.11}$$

where

$$K_{i-1/2,j} = \frac{K_{i-1,j} + K_{i,j}}{2} \quad , \quad K_{i+1/2,j} = \frac{K_{i,j} + K_{i+1,j}}{2} \tag{7.12}$$

In equation (7.12), a linear interpolation is used for the thermal conductivity between each grid point giving K at the midpoint of the space interval separating the grid points. $K_{i-1/2,j}$ and $K_{i+1/2,j}$ can also be evaluated using the approach of summing the thermal resistance as suggested by Patankar [9]. This gives:

$$K_{i-1/2,j} = \frac{2K_{i-1,j}K_{i,j}}{K_{i-1,j} + K_{i,j}} \quad , \quad K_{i+1/2,j} = \frac{2K_{i,j}K_{i+1,j}}{K_{i,j} + K_{i+1,j}} \tag{7.13}$$

7.2.1 Virtual Sub-Interval Elimination Technique (VSIET)

This technique permits the use of large time steps for any of the explicit equations (Eq.(7.7) or (7.11)) without loss of accuracy or stability. It consists of automatic generation of Virtual Sub-time-Steps (VSSs), which are progressively eliminated as the computation moves forward [10]. For illustration, let us consider that the temperature distribution $T_{i,j}, i=0,N$ is known and we wish to compute the temperature distribution at the time $t_{j+1}=t_j+\Delta t$, using (7.7) and (7.8). The temperature distribution $-T^n_{i,j}, i=0,N$ – at any VSS of index n=1, k is calculated from that of previous VSS using:

$$T^n_{i,j}=T^{n-1}_{i,j}\exp\left\{\frac{-r^{n-1}_{i,j}}{T^{n-1}_{i,j}}\left[2T^{n-1}_{i,j}-\left(1+\beta^{n-1}_{i,j}\right)T^{n-1}_{i-1,j}-\left(1-\beta^{n-1}_{i,j}\right)T^{n-1}_{i+1,j}\right]\right\} \tag{7.14}$$

where

$$r^n_{i,j}=\frac{\alpha^n_{i,j}}{\max\left(\alpha^n_{i,j}, i=0,N\right)}r_f \quad , \quad \alpha^n_{i,j}=\frac{K^n_{i,j}}{C^n_{i,j}} \quad , \quad \beta^n_{i,j}=\frac{K^n_{i-1,j}-K^n_{i+1,j}}{4K^n_{i,j}} \tag{7.15}$$

where $r_f \le 0.5$ is a fixed Fourier number throughout the computation process. At some intermediate VSS $n=k$, so that $t^k_j \ge t_{j+1}$, the temperature distribution at $t=t_{j+1}$, is calculated from that of $t=t^{k-1}_j$, using equation (7.14) (i.e. $T_{i,j+1}=T^k_{i,j}, i=0,N$) with $r^{k-1}_{i,j}=\alpha^{k-1}_{i,j}(t_{j+1}-t^{k-1}_j)/\Delta x^2$.

In order to validate the finite difference equation and compare its performance with the classical equations, let us consider the following problem:

$$C(T)\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}\left(K(T)\frac{\partial T}{\partial x}\right) \quad , \quad 0\le x\le 1 \quad , \quad t>0 \tag{7.16}$$

$$T(x,t)=\phi(t) \quad , \quad x=0 \quad , \quad t\ge 0 \tag{7.17}$$

$$T(x,t)=\varphi(t) \quad , \quad x=1 \quad , \quad t\ge 0 \tag{7.18}$$

$$T(x,t)=\eta(x) \quad , \quad 0\le x\le 1 \quad , \quad t=0 \tag{7.19}$$

Let us assume the following dependencies:

$$K(T)=d\,\mathrm{Log}_e(T) \quad , \quad C(T)=\frac{dc^2}{b}\left(1+\mathrm{Log}_e(T)\right) \tag{7.20}$$

If we choose the following boundary conditions:

$$\eta(x) = \exp(-cx) \quad , \quad \phi(t) = a\exp(bt) \quad , \quad \varphi(t) = a\exp(bt - c) \tag{7.21}$$

The following is an exact solution for equations (7.16) to (7.19):

$$T(x,t) = a\exp(bt - cx) \tag{7.22}$$

where a, b, c and d are constants.

The problem defined by (7.16) to (7.19) is solved using: the classical equation (i.e. (7.11)&(7.12)), the equation of Patankar (i.e. (7.11)&(7.13)) and the new finite difference equations (i.e. (7.7)&(7.8) and (7.7)&(7.10)). The VSIET is incorporated with all the equations. For accuracy assessment, the numerical results are compared to the analytical solution and expressed in terms of percentage deviations defined as:

$$MPD_j = \max\left(\left|PD_{i,j}\right| i = 0,N\right) \quad , \quad PD_{i,j} = 100 \times \frac{\left(T_{i,j}\right)_{Nu} - \left(T_{i,j}\right)_{An}}{\left(T_{i,j}\right)_{An}} \tag{7.23}$$

where An and Nu refer to temperature calculated Analytically and Numerically respectively. Numerical results, omitting the units of variables, (when a=20, b=4, c=1/2, d=2.0, Δx=0.1, Δt=0.01 and $r_f = 0.35$), are shown in Tables 7.1 and 7.2. These are discussed in section 7.5.

Table 7.1: Comparison of percentage deviations (x10²) at the centre of the slab.

time step		Classical (7.11)&(7.12)	Patankar (7.11)&(7.13)	Present equations (7.7)&(7.10)	(7.7)&(7.8)	Analytic
j	t_j	$PD_{N/2,j}$	$PD_{N/2,j}$	$PD_{N/2,j}$	$PD_{N/2,j}$	$T\left(\frac{1}{2},t_j\right)$
1	0.01	–0.1694	–0.1788	0.05647	+0.03765	16.2117
2	0.02	–0.2080	–0.2261	0.05426	+0.03617	16.8733
3	0.03	–0.2346	–0.2520	0.01738	+0.01738	17.5619
4	0.04	–0.2421	–0.2588	0.00000	–0.00835	18.2786
5	0.05	–0.2567	–0.2647	0.00802	–0.00802	19.0246
6	0.06	–0.2620	–0.2697	0.00000	–0.03082	19.8010
7	0.07	–0.2369	–0.2517	0.00740	+0.00000	20.6091
8	0.08	–0.2276	–0.2419	0.00711	+0.00711	21.4501
9	0.09	–0.2119	–0.2255	0.02050	+0.02050	22.3255
10	0.10	–0.2036	–0.2101	0.02627	+0.02627	23.2367

Table 7.2: Comparison of maximum percentage deviation ($x10^2$).

time step	Classical (7.11)&(7.12)	Patankar (7.11)&(7.13)	Present equations (7.7)&(7.10)	Present equations (7.7)&(7.8)
j	MPD_j	MPD_j	MPD_j	MPD_j
1	0.1694	0.1788	0.07297	0.07021
2	0.2080	0.2261	0.07683	0.05844
3	0.2346	0.2520	0.04801	0.05581
4	0.2457	0.2633	0.02667	0.02728
5	0.2614	0.2698	0.01864	0.02071
6	0.2620	0.2697	0.00852	0.03240
7	0.2369	0.2517	0.01557	0.00818
8	0.2276	0.2468	0.00786	0.00869
9	0.2227	0.2299	0.02601	0.02050
10	0.2071	0.2140	0.03123	0.02761

7.3. Mathematical formulation of laser beam–material interaction

As the laser–beam diameter is very small compared to the lateral dimensions of the material, and the power distribution is symmetric with respect to the beam centre (i.e. Gaussian); the approximation of the heat flow by a one-dimensional model is not expected to introduce significant errors [11], since the maximum heat effect occurs when the surface is irradiated by the centre part of the beam. The process can be adequately modelled as a 1–D moving boundary problem with variable thermal properties. The governing equations, describing the heat conduction into a material of thickness X with change of phase, are written as:

$$C_l(T_l)\frac{\partial T_l}{\partial t} = \frac{\partial}{\partial x}\left(K_l(T_l)\frac{\partial T_l}{\partial x}\right) \quad , \quad 0 \le x \le M(t) \quad , \quad t_m \le t \le t_e \tag{7.24}$$

$$C_s(T_s)\frac{\partial T_s}{\partial t} = \frac{\partial}{\partial x}\left(K_s(T_s)\frac{\partial T_s}{\partial x}\right) \quad , \quad M(t) \le x \le X \quad , \quad t > 0 \tag{7.25}$$

$$\left.\begin{aligned} \lambda\frac{dM}{dt} &= K_s(T_m)\left(\frac{\partial T_s}{\partial x}\right) - K_l(T_m)\left(\frac{\partial T_l}{\partial x}\right) \\ T_s(x,t) &= T_l(x,t) = T_m \end{aligned}\right\} \quad , \quad x = M(t) \quad , \quad t_m \le t \le t_e \tag{7.26}$$

where: M, λ, T_m, t_m and t_e refer to the liquid/solid interface position, latent heat of melting/solidification per unit volume, melting temperature, time at which melting starts and time at which solidification ends, respectively. The subscripts s and l refer to solid and liquid regions respectively. Equations (7.24)–(7.26) may also be written with a heat generation term, due to penetration of the radiation. However, calculations show that

energy absorption is predominantly a surface phenomenon [12]; hence, the heat generation term is neglected in (7.24)–(7.26). The boundary conditions are written as:

$$-K_l(T_l)\left(\frac{\partial T_l}{\partial x}\right) = F(T_l,t) - h(T_l,T_0)(T_l - T_0) \quad , \quad x = 0 \quad , \quad t_m \le t \le t_e \tag{7.27}$$

$$-K_s(T_s)\left(\frac{\partial T_s}{\partial x}\right) = h(T_s,T_0)(T_s - T_0) \quad , \quad x = X \quad , \quad 0 \le t \le t_e \tag{7.28}$$

where h is the heat transfer coefficient and $F(T,t)$ is the absorbed power density – at the centre of the beam – by the surface material, which is at temperature T, at time t. Assuming that the laser beam has a Gaussian distribution of radius ω at the $1/e$ point; if the laser beam has a total incident power P and is moving with a constant velocity v, then $F(T,t)$ is given by:

$$F(T,t) = \begin{cases} [1 - R(T)]\dfrac{P}{\pi\omega^2}\exp\left\{-\left(\dfrac{vt - \omega}{\omega}\right)^2\right\} & , \quad 0 \le t \le \tau \\ 0 & , \quad t > \tau \end{cases} \tag{7.29}$$

where $\tau = \omega / v$ and R are the beam–workpiece interaction time and the reflectivity of the surface, respectively. From both surfaces (x=0 and x=X) the heat is lost by a combination of free–convection and radiation. The function $h(T_1,T_2)$– expressing the heat transfer coefficient, due to both convection and radiation, of a plate of thickness X at temperature T_1 sub–merged into a fluid at temperature T_2 – is given by:

$$h(T_1,T_2) = \sigma[1 - R(T_1)](T_1^2 + T_2^2)(T_1 + T_2) + z\frac{\kappa(T_f)}{X}\left(\frac{c_p(T_f)}{\kappa(T_f)}\frac{2g\rho^2(T_f)(T_1 - T_2)X^3}{(T_1 + T_2)\mu(T_f)}\right)^y \tag{7.30}$$

where $T_f = \frac{1}{2}(T_1 + T_2)$, σ and g: are temperature of the air film, Stefan-Boltzman constant and gravity respectively; c_p, μ, ρ, and κ denotes heat capacity, dynamic viscosity, density and thermal conductivity of air respectively; these variables are temperature dependent. The correlation factors z and y are equal to: z=0.14 and y=1/3, for the surface facing upwards; z=0.58 and y=1/5 for the surface facing downwards [13].

7.3.1. Application to silicon

The variation of the thermal conductivity and the heat capacity/unit volume of silicon with temperature is given by [14]:

$$K(T) = 299 \times 10^2 \left(\frac{1}{T - 99} \right), \quad C(T) = 2.336 \times 10^6 \left(\frac{T - 159}{T - 99} \right) \tag{7.31}$$

The units in (7.31) are: T [°K], K[W /m°K] and C[J / m^3°K]. The melting temperature of silicon is $T_m = 1690$ °K and the latent heat of fusion per unit volume $\lambda = 3282 \times 10^6$ J / m^3. For liquid silicon the thermal conductivity increases to 0.64 [W/mK], while $C(T)$ is taken to be constant [11]. Experimental evidence [15,16] shows that the reflectivity of silicon varies linearly with temperature for temperatures less than the melting point. However, a more accurate function, for a large range of temperatures including melting, is given by [14]:

$$R(T) = 0.372 + 2.693 \times 10^{-5} T + 2.691 \times 10^{-15} T^4, \quad T \leq 3130 \text{ °K} \tag{7.32}$$

7.3.2. Application to nodular cast-iron

The dependence of thermal conductivity and the heat capacity/unit volume of nodular cast iron with temperature are approximated, to fit the experimental data in [17], by:

$$K(T) = \begin{cases} 8.257 \times 10^{-3}(T - 273) + 12.0 & , \quad 273 \leq T \leq 973 \\ -4.450 \times 10^{-3}(T - 973) + 17.78 & , \quad 973 \leq T \leq 1573 \\ 0.360(T - 1573) + 15.11 & , \quad 1573 \leq T \leq 1623 \\ 33.11 & , \quad T > 1623 \end{cases} \tag{7.33}$$

$$C(T) = \begin{cases} 4.670 \times 10^{-3}(T - 273) + 1.94 & , \quad 273 \leq T \leq 1573 \\ 0.123(T - 1573) + 8.01 & , \quad 1573 \leq T \leq 1623 \\ -0.123(T - 1623) + 14.17 & , \quad 1623 \leq T \leq 1673 \\ 8.01 & , \quad T > 1673 \end{cases} \tag{7.34}$$

The units in equation (7.33) and (7.34) are: T [°K], C [10^6 J / m^3] and K [W /m°K]. The melting temperature of nodular cast iron is $T_m = 1573$ °K and the latent heat of fusion per unit volume $\lambda = 791.134 \times 10^6$ J / m^3. The reflectivity as a function of temperature is approximated with a linear function [18], which is given by:

$$R(T) = 0.72 - 25 \times 10^{-5}(T - 293) \quad , \quad T \text{ [°K]} \tag{7.35}$$

For both processes (silicon and nodular cast-iron), the functions describing the variation of thermal properties of air with temperature are obtained by approximating the experimental data in [19] by a polynomial of degree 5.

7.4. Numerical computation scheme

Prior to melting ($T(0,t)<T_m$) when only one phase is present, the numerical computation is similar to that illustrated in section 7.2. When the temperature of the surface, exposed to laser radiation, exceeds the melting point, the heat in the superheated region $0 \le x \le x_1$ (i.e. $T(x,t_m)>T_m$) is converted to heat of fusion, giving an initial melt depth x_2 at a uniform melting temperature [20]. This can be expressed mathematically as:

$$x_2 = \frac{C(T_m)}{\lambda}\int_0^{x_1}(T(x,t_m)-T_m)dx \tag{7.36}$$

Generally the superheated region extends less than a space increment ($x_1<\Delta x$); therefore, $T(x,t_m)$ in (7.36) is approximated by a linear function of x; hence, x_2 is directly obtained without recourse to numerical integration.

During the melting and solidification stages, the problem is to compute the temperature distribution $T_{i,j+1}, i=0,N$ as well as the melting front position M_{j+1} from those at the previous time t_j. The computing procedure is the same as that in section 7.2, except for the nodes near the moving interface. Consider that the moving boundary is located between i_m and i_m+1. For grid lines $i \in \{0 \le i \le i_{m-1}\} \cup \{i_{m+2} \le i \le N\}$ equation (7.14) is used. For the grid lines i_m and i_m+1, instead of (7.14) the following is used:

$$T_{i,j}^n = T_{i,j}^{n-1}\exp\left(-\frac{r_{i,j}^{n-1}}{T_{i,j}^{n-1}}\Omega_{i,j}^{n-1}\right) \quad , \quad i=i_m, i_m+1 \tag{7.37}$$

where

$$\Omega_{i,j}^{n-1} = \begin{cases} \left[1+\xi+(\xi-1)\beta_{i,j}^{n-1}\right]T_{i,j}^{n-1} - \left(\xi+\xi\beta_{i,j}^{n-1}\right)T_{i-1,j}^{n-1} - \left(1-\beta_{i,j}^{n-1}\right)T_m, & i=i_m \\ \left[2-\xi\left(1-\beta_{i,j}^{n-1}\right)\right]T_{i,j}^{n-1} - \left[(1-\xi)\left(1-\beta_{i,j}^{n-1}\right)\right]T_{i+1,j}^{n-1} - \left(1+\beta_{i,j}^{n-1}\right)T_m, & i=i_m+1 \end{cases} \tag{7.38}$$

where

$$\beta_{i,j}^{n-1} = \begin{cases} \left[(1-\xi)K_{i,j}^{n-1} + \xi K_{i-1,j}^{n-1} - K(T_m)\right] / 4K_{i,j}^{n-1} & , \quad i=i_m \\ \left[-\xi K_{i,j}^{n-1} + (\xi-1)K_{i+1,j}^{n-1} + K(T_m)\right] / 4K_{i,j}^{n-1} & , \quad i=i_m+1 \end{cases} \tag{7.39}$$

where $\xi = \mathrm{mod}\left(M_j^{n-1}, \Delta x\right)$. Equations (7.38) and (7.39) using the same approximations, for the nodes around the moving boundary, used in [10] (also in chapter 4).

The temperature distribution at the time t_{j+1} is calculated from that of the last VSS (i.e. $T_{i,j+1} = T^k_{i,j}, i = 0,N$). Equation (7.14) is used for grid lines $i \in \{0, i_m - 1\} \cup \{i_m + 2, N\}$ and (7.37) for $i = i_m, i_m + 1$. The position of the melting front is given by:

$$M_{j+1} = M_j + \sum_{n=1}^{k} \chi^n_j \tag{7.40}$$

where χ^n_j, which is the variation of the melting front during each virtual time step $\delta t^n_j = t^n_j - t^{n-1}_j$, is given in appendix E.

7.5. Numerical results and discussion

Table 7.1 shows that the relative errors using the classical equations are greater than that of the new equations. It also shows that both the classical finite difference equations under-estimate the solution; whereas, (7.7)&(7.10) give an over-estimated solution; however, the numerical results due to (7.7)&(7.8) oscillates around the analytical solution. Table 7.2 shows the maximum percentage deviation at each time step; overall the new finite difference equations are 4 times more accurate than the classical ones. Equations (7.8) and (7.10) achieve similar accuracy; however, (7.8) requires less operations.

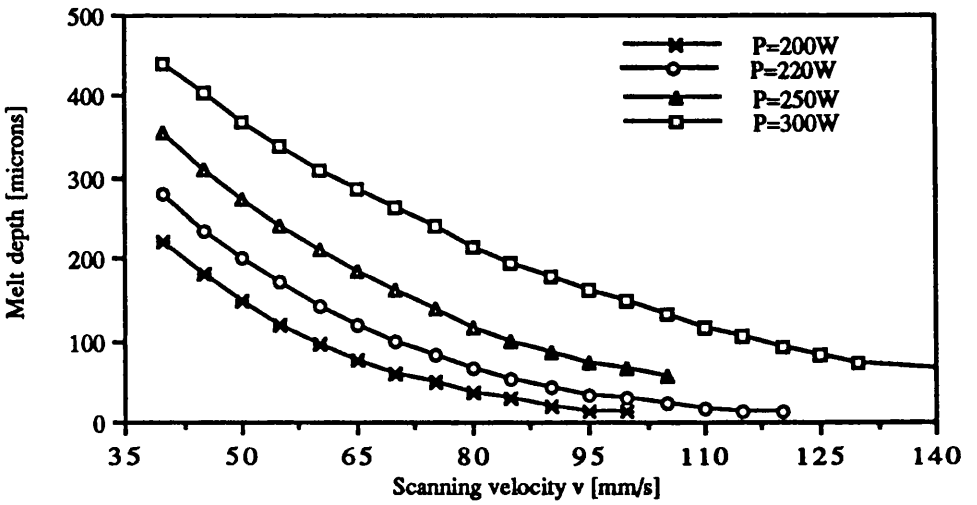

Figure 7.1: Variation of melt depth with the scanning speed at different incident powers – Silicon.

Figure 7.1 and 7.2 shows the variation of the melt depth with the scanning velocity and the incident power, for a silicon substrate of thickness X=0.5mm irradiated with a

laser beam of ω=1mm. Figure 7.1 shows that as the velocity of the beam increases the melt depth decreases, whereas it increases when increasing the total incident power, as shown in Figure 7.2. Figure 7.2 also shows that the melt depth is sensitive to the laser power at low scanning speeds, this sensitivity is less apparent at high scanning speeds. Figure 7.3 shows that the melt depth depends strongly on the beam diameter. Numerical results shows that the melt depth is generally very sensitive to the change in either the total incident power or the beam diameter; therefore, the scanning speed is the most useful laser parameter for high precision control of the process.

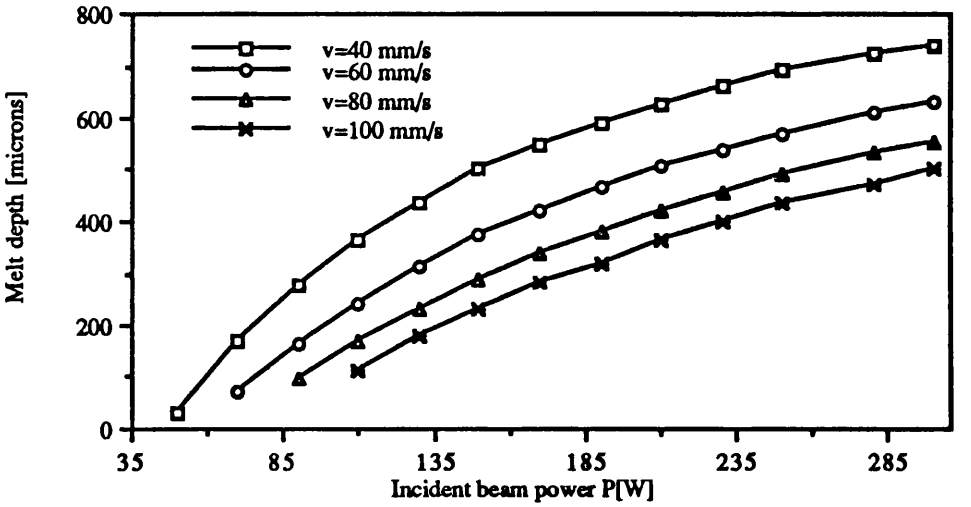

Figure 7.2: Variation of melt depth with the incident power at different scanning velocities – Silicon.

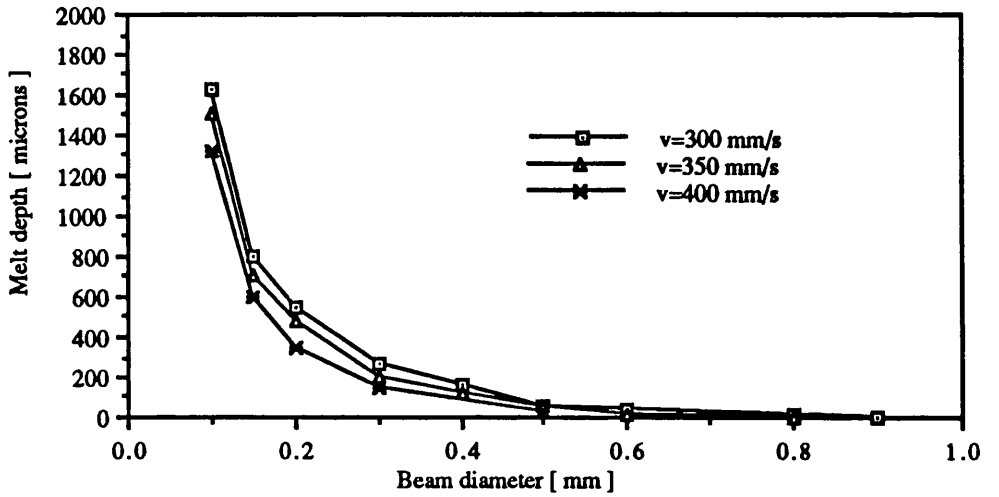

Figure 7.3: Variation of melt depth with the beam diameter at different scanning velocities – Silicon.

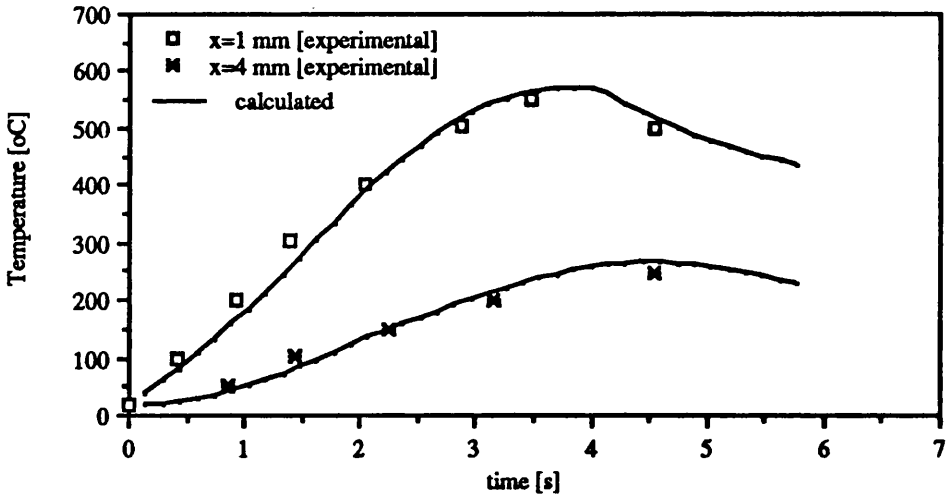

Figure 7.4: Comparison of measured and calculated temperature histories at different depths for nodular cast-iron. P=950W, ω=16 mm

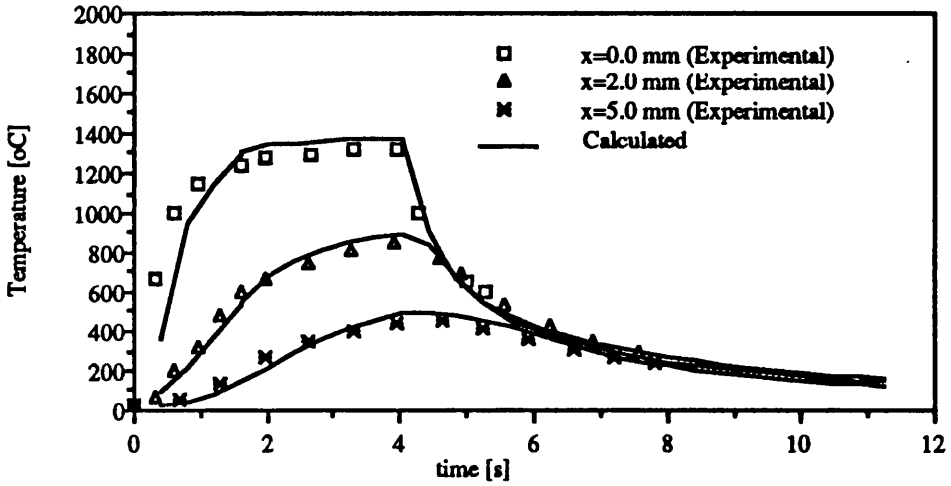

Figure 7.5: Comparison of measured and calculated temperature histories at different depths for nodular cast-iron. P=3.3KW, ω=16 mm

In order to measure the temperature evolution in a substrate material when irradiated by a laser beam, Gay [17] used large interaction times (2-4s) and large beam diameters (14-16mm) to achieve lower heating rates. The same experimental environment was used in the calculation of temperature profile during the radiation of a nodular-cast-iron substrate. Figure 7.4 shows a comparison between the calculated and measured [17] temperature evolution at different depths, when a 20 mm thick substrate is heated (no change of phase occurs) by a 16 mm diameter, 950 W laser beam, for 4 seconds. The agreement is fairly good. To obtain melting the incident power is increased to 3.3 KW and the thickness of the material reduced to 15 mm. Figure 7.5 shows the calculated and measured temperature for such conditions. In order to see the variation of the melt

depth with the interaction time, a 10 mm diameter, 3 KW laser beam was used. This is illustrated in Figure 7.6, where a comparison between the measured and predicted melt depth for different interaction times is shown. The agreement is satisfactory.

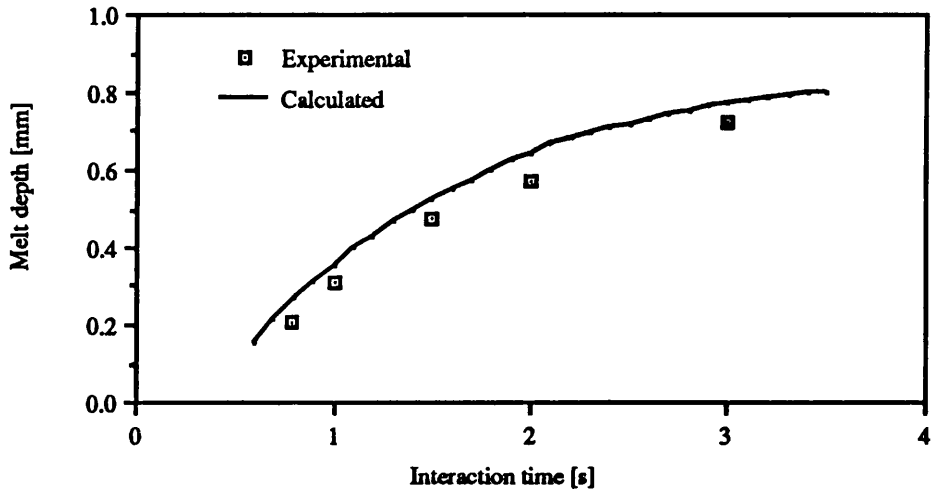

Figure 7.6: Comparison of measured and calculated melt depths for different interaction times for nodular cast-iron. P=3KW, ω=10 mm.

7.6. Conclusion

A finite difference equation for non-linear heat conduction problems has been presented and proved to be superior in terms of accuracy, when compared to the classical equations. The equation is also incorporated into a model for simulation of heat flow during melting and re–crystallisation of material induced by a scanning laser beam. The calculation includes temperature dependence of thermal properties and surface reflectivity. The numerical results compare very favourably with experimental results. Silicon was chosen due to its importance to the electronics industry, and nodular cast iron for comparison with experimental results. The large array sizes and consequently large computer memory, generally required for such calculations, are avoided by using the virtual time step elimination technique.

7.7. References

1. I.N. Miaoulis and B.B. Mikic, *J. Appl. Phys.* **59**, 1658-1662 (1986).
2. I.N. Miaoulis and B.B. Mikic, *J. Appl. Phys.* **59**, 1663-1666 (1986).
3. D. Waechter, P. Schvan, R.E. Thomas and N.G. Tarr, *J. Appl. Phys.* **59**, 3371-3374 (1986).

4. A.A. Alwan, *Surface Temperature Transients in Laser Machining Metals*, M.Sc. Thesis, University of Glasgow, 1989.
5. L.G.D.D. Rose, in *Laser Surface Treatment of Metals*, C.W. Draper and P. Mazzoldi (eds.), Martinus Nijhoff, Dordrecht, 1986.
6. M.C. Bhattacharya, *Int. J. Numer. Meth. Eng.* **21**, 239-265 (1985).
7. M.C. Bhattacharya, *Commu. Appl. Numer. Meth.* **6**, 173-184 (1990).
8. Y. Jaluria and K.E.Torrance (eds.), *Computational Heat Transfer*, Hemisphere P.C., Washington D.C., 1986.
9. S.V. Patankar (ed.), *Numerical Heat Transfer and Fluid Flow*, Hemisphere P.C., Washington D.C., 1980.
10. M. Zerroukat and C.R. Chatwin, *Int. J. Numer. Meth. Eng.* **35**, 1503-1520 (1992).
11. P. Schvan and R.E. Thomas, *J. Appl. Phys.* **57**, 4738-4741 (1985).
12. K. Kubota,C.E. Hunt and J. Frey, *Appl. Phys. Lett.* **46**, 1153-1155 (1985).
13. J. Fujii and H. Imura, *Int. J. Heat Mass Transfer* **15**, 755-767 (1973).
14. J.E. Moody and R.H. Hendel, *J. Appl. Phys.* **53**, 4364–4371 (1982).
15. G.E. Jellison Jr. and F.A. Modine, *Phys. Review* **B27**, 7466-7472 (1983).
16. G.E. Jellison Jr. and H.H. Burke, *J. Appl. Phys.* **60**, 841–843 (1986).
17. P. Gay, in *Laser Surface Treatment of Metals*, C.W. Draper and P. Mazzoldi (eds.), Martinus Nijhoff, Dordrecht, 1986.
18. W.W. Duley, in *Laser Surface Treatment of Metals*, C.W. Draper and P. Mazzoldi (eds.), Martinus Nijhoff, Dordrecht, 1986.
19. G.F.C. Rogers and Y.R. Mayhew (eds.), *Thermodynamic and Transport Properties of Fluids*, Basil Balackwell, Oxford, 1980.
20. S.C. Hsu, S. Chakravorty and R. Mehrabian, *Metall. Trans.* **B9**, 221-229 (1978).

General conclusions and suggestions for further work

The accuracy of numerical schemes depends largely on the recurrence formulae such as Euler, Crank–Nicolson and the fully implicit equations. On the other hand, the refinement of the mesh size also increases the accuracy of the solution, but at the expense of increasing the memory size requirements and cpu-time. When computations have to be performed for extended times, the use of large time steps is desirable to reduce the memory size requirements and the run time to the minimum possible (computational efficiency). Under these circumstances the implicit equations, which are stable for any mesh size, have been used extensively. Since the implicit equations are very inaccurate when the time step is large (chapter 3), and in order to achieve a good accuracy without recourse to large memory size requirements, the combination of explicit equations with the VSIET is the best choice. This permits calculations for large problems even using a computer with a very small RAM (Random Access Memory).

There is wide-spread use of Personal Computers (PC's) and work-stations which have limited RAM (Random Access Memory), therefore in order to compute any type of problem the efficient use of memory is of major concern. The VSIET can be used to accomplish such tasks; using the VSIET, large computations can be performed with a minimum array size. Furthermore, the VSIET automatically preserves the stability of the scheme, irrespective of the mesh size, without increasing the storage requirements, and reduces the loss of accuracy associated with implicit finite difference equations, when used with large time steps. The use of explicit computation procedures permits the maximum percentage vectorisation of the scheme, since they are more suitable for vectorisation than implicit finite difference and finite element methods, due to the fact that the unknown variables are determined with no recurrence (statements that form a cycle of variable dependencies). This results in efficient use of the computational platform.

The idea of combining explicit equations with the VSIET, has been successfully applied to different problems, and proved to be superior in terms of accuracy and computational efficiency. The numerical schemes presented in different chapters are characterised by the simplicity in formulation and programming.

Furthermore, non-linear heat conduction problems have also been dealt with. A typical application of these numerical schemes was presented in the last chapter, it consists of using numerical computation to study the heat flow during melting and re–crystallisation of material subjected to radiation from a scanning laser beam. The numerical results compare very favourably with experimental results.

Further research should be undertaken to extend these schemes to higher dimensions (2-D and 3-D). The extension of these schemes to higher dimensions is possible, since the virtual time steps concern the time variable only. To achieve this, a combination of a fixed grid network and the VSIET can be exploited. The computation of the moving boundary can be performed using a similar scheme to that of chapter 7; however, incorporating the enthalpy–temperature relations would also be very useful. The finite difference equations for the solution of the 2-D and 3-D non-linear heat conduction equation, using the approach in chapter 7, have already been developed and tested; they are not reported herein but will be the subject of a future publication.

The new finite difference equation can be used for both equal and unequal nodal spacing as well as for both Dirichlet and Newmann type boundary conditions. This facilitates the computation around the boundaries of complex shapes, such as corners, by using meshes of arbitrary geometry near the boundaries. As seen for 1–D problems, the VSIET uses two vectors, **A** and **B**, and transposes them as the computation moves one virtual step. For higher dimensions, the same procedure is used, the only difference is that **A** and **B** are arrays of 2 dimensions or 3 dimensions for 2–D and 3–D problems, respectively.

The numerical scheme may be used in many industrial applications involving transient heat transfer without or with phase transformations. Many industrial systems undergo transient processes at various stages of operation: the start–up and shut–down of systems, such as power plant, chemical manufacture, furnaces and ovens. All these systems involve a consideration of transient heat conduction as well as the variation of the thermal properties with temperature.

Appendix A: Mathematical formulation of moving boundary problems (Chapter 2)

A.1. One-phase ice melting problem

The simplest formulation of Moving Boundary Problems (MBPs) is the melting of a semi-infinite sheet of ice (J. Stefan [1]), maintained at the melting point (zero temperature). An interface $x = M(t)$ on which the melting occurs, moves from the surface, which is kept at a constant temperature $T_0 > 0$, into the ice-sheet. If $T(x,t)$ denotes the temperature at the liquid depth x measured from the surface of the sheet, at time t, the problem is to find the pair of functions $T(x,t)$ and $M(t)$ which verify the following set of equations:

$$\rho c \frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 < x < M(t) \quad , \quad t > 0 \tag{A.1}$$

with the boundary condition

$$T(x,t) = T_0 \quad , \quad x = 0 \quad , \quad t = 0 \tag{A.2}$$

and initial conditions

$$T(x,t) = 0 \quad , \quad x > 0 \quad , \quad t = 0 \tag{A.3}$$

$$M(t) = 0 \quad , \quad t = 0 \tag{A.4}$$

Two further conditions are needed on the moving boundary, these are:

$$\left.\begin{aligned} T(x,t) &= 0 \\ -K \frac{\partial T}{\partial x} &= \rho L \frac{dM}{dt} \end{aligned}\right\} \quad , \quad x = M(t) \quad , \quad t > 0 \tag{A.5}$$

where ρ, c, K and L are density, heat capacity, thermal conductivity and latent heat of fusion respectively.

A.2. Two-phase ice melting problem

A more general formulation can expressed by the fact the heat flow occurs in both phases. The heat parameters may all be functions of T, x and t, i.e. $K = K(T,x,t)$ and

$c = c(T,x,t)$. There may be also a heat source $Q = Q(T_m,x,t)$ (or sink) at the moving boundary. The heat conduction equation (A.1) becomes

$$\rho_i c_i(T_i,x,t)\frac{\partial T_i}{\partial t} = \frac{\partial}{\partial x}\left(K_i(T_i,x,t)\frac{\partial T_i}{\partial x}\right) \quad , \quad x \in R_i \quad , \quad i = 1,2 \tag{A.6}$$

where R_1 and R_2 are the regions defined by:

$$\begin{aligned} R_1 &= \{(x,t)\,;\,0 < x < M(t)\,,\,t > 0\} \\ R_2 &= \{(x,t)\,;\,M(t) < x < a\,,\,t > 0\} \end{aligned} \tag{A.7}$$

and the condition at the moving boundary (A.5) becomes:

$$K_2(T_m,x,t)\frac{\partial T_2}{\partial x} - K_1(T_m,x,t)\frac{\partial T_1}{\partial x} = \rho L\frac{dM}{dt} - Q(T_m,x,t) \tag{A.8}$$

It can also be emphasised that the change of phase may not be explained by a sharp isothermal boundary. An alternative formulation may be used to explain the fact that the phase change occurs over a range of temperature "*Mushy region*". Atthey [2] studied the welding of two sheets of metal as an example of mushy region formulation. Several authors [3-7] have presented formulations for different problems and discussed the physical significance of such formulations.

A.3. General formulation of a moving boundary problem

The general melting or solidification process of a material is described as follows: At time $t = t_0$, the two regions R_1 and R_2 which are bounded by the fixed boundary surfaces ∂R_1 and ∂R_2 respectively and the moving boundary $M = M(X,t_0)$ (Fig. A.1), with X the space coordinate vector.

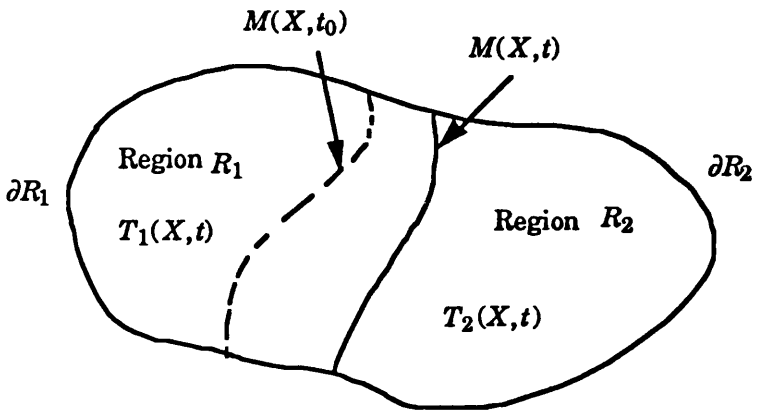

Figure A.1: Two different regions separated by a moving boundary $M(X,t)$

At later time t the moving boundary M moves into a new position $M(X,t)$, in the domain $R = R_2 \cup R_2$. The problem is to locate M in space at any time and subsequently the temperature distributions $T_1(X,t)$ and $T_2(X,t)$ in the regions R_1 and R_2 respectively. The formulation of a such problem, taking into considerations, dependence of thermal properties on temperature, heat generation terms E and Q within the material and at the moving boundary respectively, can be expressed as:

$$\rho_i c_i \frac{\partial T_i}{\partial t} = \nabla(K_i \nabla T_i) + E \quad , \quad x \in R_i \quad , \quad i = 1,2 \tag{A.9}$$

with the boundary conditions

$$\frac{\partial T_i}{\partial n} - h_i T_i = g_i \quad , \quad x \in \partial R_i \quad , \quad i = 1,2 \tag{A.10}$$

and on the moving boundary $M(X,t)$

$$\left[\frac{\partial}{\partial n}(KT)\right]_1^2 = \frac{\partial}{\partial n}(K_2 T_2) - \frac{\partial}{\partial n}(K_1 T_1) = \rho L V_n + Q$$
$$T_1(X,t) = T_1(X,t) = T_m \tag{A.11}$$

and the initial conditions

$$T_1(X,0) = T_1(X) \quad , \quad T_2(X,0) = T_2(X) \quad , \quad M(X,0) = M_0(X) \tag{A.12}$$

where n is the outward normal to the moving boundary and V_n is the velocity of this boundary along the normal. The functions E, Q and g can be also functions of X and t.

Equation (A.11) can also be written in Patel's expressions for each space direction x,y and z, which are more practical forms. Details of the Patel's expression can be found in [8] and [9].

A.4. Implicit moving boundary problems (oxygen diffusion problem)

In the problems formulated in A.1, A.2 and A.3, the moving boundary condition connect the variable T or its derivatives with the velocity of the moving boundary (e.g. Eq. (A.8)). Some problems exist, in which such an explicit relationship does not exists on the moving boundary, these are called *implicit moving boundary problems* [10], in other words they correspond to the special case of L=0 in (A.8).

The problem, arising from the diffusion of oxygen in a medium which simultaneously consume the oxygen [11,12], is an example of implicit moving boundary

problems, due to the nature of the boundary condition prevailing at the moving boundary. The diffusion process is described by the differential equation

$$\frac{\partial U}{\partial t} = D\frac{\partial^2 U}{\partial x^2} - R \tag{A.13}$$

where $U(x,t)$ denotes the concentration of the oxygen free to diffuse at distance x from the outer surface of the medium at time t, R is the rate of consumption of oxygen per unit volume of the medium and D is the diffusion coefficient.

The oxygen diffusion problem is formulated in two stages, first as a steady state solution (details can be found in [11]) and secondly as a moving boundary problem. After the surface x=0 has been sealed, the oxygen which is already in the region, continues to be consumed; consequently, the point of zero concentration moves towards x=0. Let the position of zero concentration be denoted by $M(t)$, the problem is expressed by the following equations:

$$\frac{\partial U}{\partial t} = D\frac{\partial^2 U}{\partial x^2} - R \quad , \quad 0 < x < M(t) \quad , \quad t > 0 \tag{A.14}$$

$$\frac{\partial U}{\partial x} = 0 \quad , \quad x = 0 \quad , \quad t > 0 \tag{A.15}$$

$$U = \frac{\partial U}{\partial x} = 0 \quad , \quad x = M(t) \quad , \quad t \geq 0 \tag{A.16}$$

$$U(x,t) = \frac{R}{2D}(x - M_0)^2 \quad , \quad 0 \leq x \leq M_0 \quad , \quad t = 0 \tag{A.17}$$

where $M_0 = M(0)$, which is given by the solution of the steady state [11] and t=0 is the moment when the surface is sealed. It can be noticed that the velocity of the moving boundary is absent in the boundary condition given by equation (A.16).

Appendix B: Calculation of temperatures around the moving boundary (Chapter 4)

To calculate the temperature $T^k_{i_b,j,m,p}$ the following approximation are made:

(i) The moving boundary between $t = t_j$ and $t = t^k_{j+1}$ is approximated by a step wise function of p^k_j steps.

(ii) The temperature profile between $t = t_j$ and $t = t^k_{j+1}$ is approximated by a step wise function of p^k_j steps (i.e. The temperature remains constant within a virtual sub-time step).

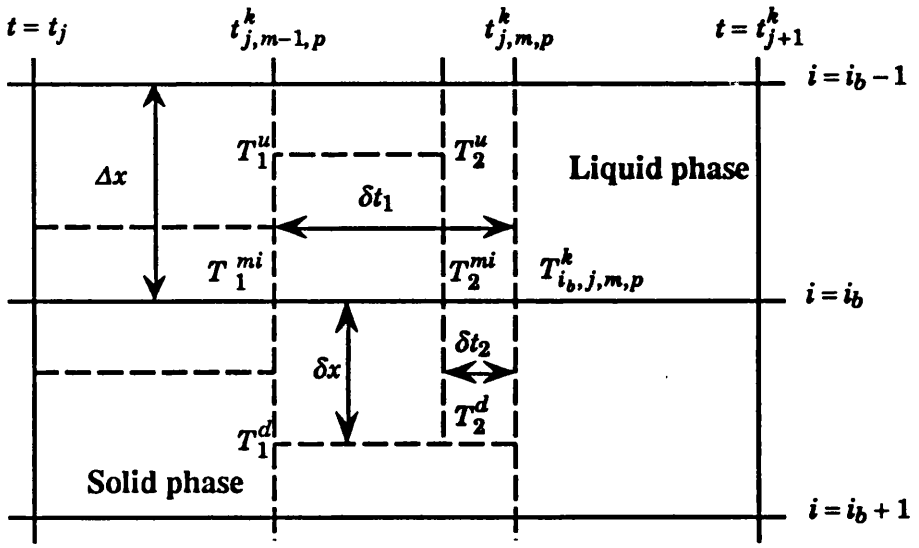

Figure B.1: Descretisation around the moving boundary

It follows from Figure B.1:

$$T^k_{i_b,j,m,p} = T^{mi}_1 \exp\left(-r_1 \frac{2T^{mi}_1 - T^u_1 - T^d_1}{T^{mi}_1}\right) \tag{B.1}$$

where $r_1 = \alpha_l \, \delta t_1 / \delta x^2$.

From (i), $\delta x = m\,\Delta x / p^k_j$, therefore $r_1 = \left(p^k_j / m\right)^2 r^k_{cl,j} > \frac{1}{2}$. To preserve stability, δt_1 must in turn be divided into $\left(p^k_j / m\right)^2$ increments of length δt_1 (Figure B.1); therefore $T^k_{i_b,j,m,p}$ must now be calculated from:

$$T^k_{i_b,j,m,p} = T^{mi}_2 \exp\left(-r_2 \frac{2T^{mi}_2 - T^u_2 - T^d_2}{T^{mi}_2}\right) \tag{B.2}$$

where

$$r_2 = \alpha_l \frac{\delta t_2}{\delta x^2} = \alpha_l \frac{\delta t_1 \Big/ \left(p^k_j / m\right)^2}{\left(m\Delta x / p^k_j\right)^2} = r^k_{cl,j} \tag{B.3}$$

Using the approximation stated in (ii):

$$T^{mi}_2 = T^{mi}_1 = T^k_{i_b,j,m-1,p} \quad , \quad T^u_2 = T^u_1 \quad , \quad T^d_2 = T^d_1 = T_b \tag{B.4}$$

T^u_1 is interpolated from $T^k_{i_b-1,j,m-1,p}$ and $T^k_{i_b,j,m-1,p}$ already known from the previous VSS $(j, m-1, p)$ giving:

$$T_1^u = \left(1 - \frac{m}{p_j^k}\right) T^k_{i_b,j,m-1,p} + \left(\frac{m}{p_j^k}\right) T^k_{i_b-1,j,m-1,p} \tag{B.5}$$

Rewriting (B.2) and making use of (B.3), (B.4) and (B.5):

$$T^k_{i_b,j,m,p} = T^k_{i_b,j,m-1,p} \exp\left(-r^k_{cl,j}\, \psi^k_{i_b,j,m-1,p}\right) \tag{B.6}$$

where

$$\psi^k_{i_b,j,m-1,p} = \frac{1}{T^k_{i_b,j,m-1,p}} \left\{ \left(1 + \frac{m}{p_j^k}\right) T^k_{i_b,j,m-1,p} - \left(\frac{m}{p_j^k}\right) T^k_{i_b-1,j,m-1,p} - T_b \right\} \tag{B.7}$$

and in general for any value of m, $\psi^k_{i_b,j,m,p}$ is given by:

$$\psi^k_{i_b,j,m,p} = \frac{1}{T^k_{i_b,j,m,p}} \left\{ \left(1 + \frac{m+1}{p_j^k}\right) T^k_{i_b,j,m,p} - \left(\frac{m+1}{p_j^k}\right) T^k_{i_b-1,j,m,p} - T_b \right\} \tag{B.8}$$

The same approach can be used to derive equation (4.28).

Appendix C: The finite difference scheme of the oxygen diffusion problem (Chapter 5)

C.1. Finite difference of the diffusion equation

Using the change variable $v=u+t$, equation (5.7) becomes:

$$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} \tag{C.1}$$

where

$$v(x,0) = u(x,0) \quad , \qquad \frac{\partial^2 v}{\partial x^2} = \frac{\partial^2 u}{\partial x^2} \tag{C.2}$$

Following the procedure of Bhattacharya [13], the finite difference approximation to (C.1), expressing $v\left(x_i, t_j^m\right)$ as a function of $v\left(x_i, t_j^{m-1}\right)$, $v\left(x_{i-1}, t_j^{m-1}\right)$ and $v\left(x_{i+1}, t_j^{m-1}\right)$, is given by:

$$v^m_{i,j} = v^{m-1}_{i,j} \exp\left\{ \left(\frac{t_j^m - t_j^{m-1}}{\Delta x^2} \right) \frac{v^{m-1}_{i+1,j} - 2v^{m-1}_{i,j} + v^{m-1}_{i-1,j}}{v^{m-1}_{i,j}} \right\} \tag{C.3}$$

Similarly (5.7), taking into account (C.2), has the following finite difference solution:

$$u_{i,j}^{m} = u_{i,j}^{m-1} \exp\left\{-r_f \frac{2u_{i,j}^{m-1} - u_{i+1,j}^{m-1} - u_{i-1,j}^{m-1}}{u_{i,j}^{m-1}}\right\} - \delta t \quad , \quad i = 0, i_b - 1 \tag{C.4}$$

where $\delta t = t_j^m - t_j^{m-1}$ and $r_f = \dfrac{\delta t}{\Delta x^2} \le \dfrac{1}{2}$.

Since the concentration is small everywhere and to avoid overflow and underflow, the exponential in equation (C.4) is approximated by the first three terms of the Taylor expansion ($\exp(y) = 1 + y + \frac{1}{2}y^2$), equation (C.4) therefore becomes:

$$u_{i,j}^{m} = u_{i,j}^{m-1} - r_f \theta_{i,j}^{m-1}\left(1 - \frac{r_f}{2u_{i,j}^{m-1}}\theta_{i,j}^{m-1}\right) - r_f \Delta x^2 \quad , \quad i = 0, i_b - 1 \tag{C.5}$$

where

$$\theta_{i,j}^{m} = 2u_{i,j}^{m} - u_{i+1,j}^{m} - u_{i-1,j}^{m} \tag{C.6}$$

When $i = 0$, $u_{i-1,j}^{m}$ is a fictitious concentration at $x = -\Delta x$. To satisfy the boundary condition (5.8), $u_{i-1,j}^{m}$ is replaced by $u_{i+1,j}^{m}$ giving:

$$\theta_{i,j}^{m} = 2\left(u_{i,j}^{m} - u_{i+1,j}^{m}\right) \quad , \quad i = 0 \tag{C.7}$$

As $\left(x_{i_b}, t_j^m\right)$ is located in the zero concentration zone and therefore $u_{i_b,j}^{m} = 0$, when $i = i_b - 1$, equation (C.6) becomes:

$$\theta_{i,j}^{m} = 2u_{i,j}^{m} - u_{i-1,j}^{m} \quad , \quad i = i_b - 1 \tag{C.8}$$

C.2. Algorithm for numerical computation

Let us define the real functions F and G as:

$$F(a,b,c,i,i_b,r) = b - r\theta\left(1 - \frac{r}{2b}\theta\right) - r\Delta x^2 \text{ where } \theta = \begin{cases} 2(b-c) & , i = 0 \\ 2b - a - c & , i = 1, i_b - 1 \\ 2b - a & , i = i_b \end{cases} \tag{C.9}$$

$$G(a,b,r) = r(2b-a)\left(1 - \frac{r(2b-a)}{2b}\right) + r\Delta x^2 - b \tag{C.10}$$

The notation A_i and B_i represents the i^{th} element of the vector **A** and **B** respectively.

The problem defined in section 5.2 can be computed following an algorithm of the following type:

$u_{i,0} = B_i = u(i\Delta x, 0); i = 0, n$ using Eq. (5.10)

do $j = 1, n$

$i_b = n - j$; $m = 0$; $s_j = i_b \Delta x$

do while $(B_{i_b} > 0)$

$m = m + 1$

$A_i = B_i$; $i = 0, i_b$

do $i = 0, i_b$

$B_i = F(A_{i-1}, A_i, A_{i+1}, i, i_b, r_f)$

end do

end while

calculate $\bar{r}_j > 0$ so that $G(A_{i_b-1}, A_{i_b}, \bar{r}_j) = 0$

$t_j = t_{j-1} + \{(m-1)r_f + \bar{r}_j\}\Delta x^2$

$u_{i,j} = F(A_{i-1}, A_i, A_{i+1}, i, i_b, \bar{r}_j)$; $i = 0, i_b$

end do

Appendix D: Numerical scheme of multiple moving boundary problems (Chapter 6)

D.1. Computation of temperatures at the first time step

The dummy point temperature approach, which is generally used to convert a heat flux to a fictitious temperature outside the domain, is not a sufficiently accurate approach when the time step is relatively large, especially at the beginning of the computation where the solid is at constant temperature throughout. Therefore, in order to avoid instability of the scheme, and also the need for extremely small increments, it is necessary to approximate the temperature at the first time step analytically. Since the temperature at $x = a$ remains unchanged during the time $0 \le t \le \Delta t_0$, the integral transform technique [14] was used to solve equations (6.1) and (6.2) for a semi–infinite slab subjected to constant heat flux F_c at $x = 0$, yielding the following solution for the temperature distribution:

$$T_s(x,t) = \frac{2F_c}{K_s}(\alpha_s t)^{1/2}\,\text{ierfc}\left(\frac{x}{2(\alpha_s t)^{1/2}}\right) + T_0 \quad , \quad 0 \le t \le \Delta t_0 \qquad \text{(D.1)}$$

The heat flux during the time $0 \le t \le \Delta t_0$ is approximated by the average of $F(0)$ and $F(\Delta t_0)$, hence the temperature distribution at the time $t_1 = \Delta t_0$ is given by:

$$T_{i,1} = \frac{F(0)+F(\Delta t_0)}{K_s}(\alpha_s \Delta t_0)^{1/2} \operatorname{ierfc}\left(\frac{i\Delta x}{2(\alpha_s \Delta t_0)^{1/2}}\right) + T_0 \quad , \quad 0 \le i \le N \tag{D.2}$$

where

$$\operatorname{ierfc}(x) = \frac{\exp(-x^2)}{\pi^{1/2}} - x \operatorname{erfc}(x) \tag{D.3}$$

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) \tag{D.4}$$

and

$$\operatorname{erf}(x) = \frac{2}{\pi^{1/2}} \int_0^x \exp(-y^2)\, dy \tag{D.5}$$

Combination of (D.3) and (D.4) give:

$$\operatorname{ierfc}(x) = \frac{\exp(-x^2)}{\pi^{1/2}} - x\,(1 - \operatorname{erf}(x)) \tag{D.6}$$

In order to reduce the computational time to the minimum possible, $\operatorname{erf}(x)$ is approximated within an error of $|\varepsilon(x)| \le 1.5 \times 10^{-7}$ by [15]:

$$\operatorname{erf}(x) = 1 - \exp(-x^2)\left[\sum_{i=1}^{5} a_i \left(\frac{1}{1+bx}\right)^i\right] + \varepsilon(x) \tag{D.7}$$

where

$$\begin{aligned} &b = 0.3275911 \qquad a_1 = 0.254829592 \qquad a_2 = -0.284496736 \\ &a_3 = 1.421413741 \quad a_4 = -1.453152027 \quad a_5 = 1.061405429 \end{aligned} \tag{D.8}$$

D.2. Algorithm for numerical computation

For simplification we use the notation VO(D) to represent a Vector Output of the temperatures of D discrete points, at any instant in time, calculated from the Vector Input VI(D) from the previous time step. Subscripts l and s are used to distinguish liquid and solid respectively the definition of all other variables are identical to the main text). The

computation of the problem such as the one described in section 6.2 can be computed using the method described in section 6.3 following an algorithm of the following type:

Heating of solid stage

calculate temperature distribution at the first time step $j = 1$ (analytical approximation)

calculate p_0 and r_c

do while $(T(0,t) < T_m)$

$j = j + 1$

do $n = 1, p_0$

calculate $\mathbf{VO}_s(N+1) = \mathbf{VO}_s(\mathbf{VI}_s(N+1))$

make the transposition $\mathbf{VI}_s(N+1) = \mathbf{VO}_s(N+1)$

for the next VSS $(n+1)$

end do

assign the final $\mathbf{VO}_s(N+1)$ as temperature distribution at the time t_{j+1}

end while

correct t_m and recalculate the vector $\mathbf{VO}_s(N+1)$ for the corrected time t_m

Melting stage

do while $(T(0,t) < T_v)$

$j = j + 1$

$i_m = j - j_m + 1$

do while $\left(\left|\varepsilon_j^k\right| > \varepsilon_{min}\right)$, $\Delta t_j^0 = \Delta t_{j-1}$

calculate p_j^k and q_j^k and the corrected values of $r_{cl,j}^k$ and $r_{cs,j}^k$

do $n = 1, p_j^k$

$\mathbf{VO}_l^k(i_m) = \mathbf{VO}_l^k\left(\mathbf{VI}_l^k(i_m)\right)$

$\mathbf{VI}_l^k(i_m) = \mathbf{VO}_l^k(i_m)$

end do

do $n = 1, q_j^k$

$\mathbf{VO}_s^k(N+1-i_m) = \mathbf{VO}_s^k\left(\mathbf{VI}_s^k(N+1-i_m)\right)$

$\mathbf{VI}_s^k(N+1-i_m) = \mathbf{VO}_s^k(N+1-i_m)$

end do

calculate ε_j^k and the time step Δt_j^{k+1} for the next iteration $k = k+1$

end while

assign $\mathbf{VO}_l(i_m)$ and $\mathbf{VO}_s(N+1-i_m)$ as temperature distribution in the liquid and solid, respectively, at the time t_{j+1}

end while

Vaporisation stage

calculate t_v and the initially removed material x_2

do $j = j_v, j_e$

$i_m = j - j_m + 1$

do while $\left(\left|\varepsilon_j^k\right| > \varepsilon_{min}\right)$; $\Delta t_j^0 = \Delta t_{j-1}$

calculate p_j^k and q_j^k ; the values of $r_{cl,j}^k$ and $r_{cs,j}^k$

do $n = 1, p_j^k$

calculate $\chi_{j,n-1,p}^k$ and locate the surface node i_s

$\mathrm{VO}_l^k(i_m - i_s) = \mathrm{VO}_l^k\left(\mathrm{VI}_l^k(i_m - i_s)\right)$

$\mathrm{VI}_l^k(i_m - i_s) = \mathrm{VO}_l^k(i_m - i_s)$

end do

do $n = 1, q_j^k$

$\mathrm{VO}_s^k(N + 1 - i_m) = \mathrm{VO}_s^k\left(\mathrm{VI}_s^k(N + 1 - i_m)\right)$

$\mathrm{VI}_s^k(N + 1 - i_m) = \mathrm{VO}_s^k(N + 1 - i_m)$

end do

calculate ε_j^k and the time step Δt_j^{k+1} for the next iteration $k = k + 1$

end while

assign $\mathrm{VO}_l(i_m)$ and $\mathrm{VO}_s(N + 1 - i_m)$ as temperature distribution in the liquid and solid, respectively, at the time t_{j+1}

end do

stop

end

Appendix E: Computation of liquid/solid motion (Chapter 7)

Integrating (7.24) with respect to x, from x=0 to x=M, and (7.25) from x=M to x=X, we obtain:

$$\int_0^M C_l \frac{\partial T_l}{\partial t} dx + \int_M^X C_s \frac{\partial T_s}{\partial t} dx = -\lambda \frac{dM}{dt} + Q_1(t) - Q_2(t) \qquad \text{(E.1)}$$

where $Q_1(t)$ and $Q_2(t)$ are the heat flux absorbed by the top surface (i.e. right side of (7.27)) and the heat flux lost by the bottom surface (i.e. right side of (7.28)), respectively. Further integration of (E.1) with respect of t, from $t = t_j^{n-1}$ to $t = t_j^n = t_j^{n-1} + \delta t_j^n$, gives:

$$\int_{t_j^{n-1}}^{t_j^n}\int_0^M C_l \frac{\partial T_l}{\partial t}\,dx\,dt + \int_{t_j^{n-1}}^{t_j^n}\int_M^X C_s \frac{\partial T_s}{\partial t}\,dx\,dt = -\lambda\chi_j^n + \int_{t_j^{n-1}}^{t_j^n} Q_1(t)\,dt - \int_{t_j^{n-1}}^{t_j^n} Q_2(t)\,dt \qquad \text{(E.2)}$$

where χ_j^n is the variation of the position of the liquid/solid interface during the time step $\delta t_j^n = t_j^n - t_j^{n-1}$. Using the trapezoidal rule to approximate the integrals in (E.2), χ_j^n is given by:

$$\chi_j^n = \frac{1}{\lambda}\left\{\left(\frac{Q_{1,j}^n + Q_{1,j}^{n-1}}{2}\right)\delta t_j^n - \left(\frac{Q_{2,j}^n + Q_{2,j}^{n-1}}{2}\right)\delta t_j^n - \theta_j^n\right\} \qquad \text{(E.3)}$$

where $Q_{1,j}^n = Q_1(t_j^n)$, $Q_{2,j}^n = Q_2(t_j^n)$ and θ_j^n is given by:

$$\theta_j^n = \frac{\Delta x}{2}\sum_{i=0}^{N-1} C\left(\frac{T_{i,j}^n + T_{i+1,j}^n + T_{i,j}^{n-1} + T_{i+1,j}^{n-1}}{4}\right)\left(T_{i,j}^n + T_{i+1,j}^n - T_{i,j}^{n-1} - T_{i+1,j}^{n-1}\right) \qquad \text{(E.4)}$$

References

1. J. Stefan, *Sber. Akad. Wiss. Wien.* **98**, 473-484 (1889).
2. D.R. Atthey, *J. Inst. Math. Appl.* **13**, 353-366 (1974).
3. L.I. Rubinstein, *Bull. Math. Biophys.* **36**, 365-377 (1974).
4. L.I. Rubinstein, *J. Inst. Math. Appl.* **24**, 259-277 (1979).
5. L.I. Rubinstein, *IMA J. Appl. Math.* **28**, 287-299 (1979).
6. E. Magenes (ed.), *Free boundary Problems vols. I and II*, Instituo Nazionale di Alta Matematica Francescon Severi, Rome, (1980).
7. C. Elliott and J.R. Ockendon, Weak and Variational methods for moving boundary problems, *Research Notes in Mathematics* **59**, Pitman, London (1982).
8. P.D. Patel, *Am. Inst. Aeronautics Astro. J.* **6**, 2454-2464 (1968).
9. A. Lazaridis, *Int. J. Heat Mass Transfer* **13**, 1459-1477 (1970).
10. G.G. Sackett, *SIAM J. Numer. Anal.* **8**, 80-96 (1971).
11. J. Crank and R.S. Gupta, *J. Inst. Math. Appl.* **10**, 19-33 (1972).
12. J. Crank and R.S. Gupta, *J. Inst. Math. Appl.* **10**, 296-304 (1972).
13. M.C. Bhattacharya, *Int. J. Numer. Meth. Eng.* **21**, 239-265 (1985).
14. M.N. Ozisk (ed.), *Boundary value problems of heat conduction*, Int. textbook company, Scranton, Pennsylvania (1968).
15. M. Abramowitz and I.A. Stegun (eds.), *Handbook of mathematical functions*, Dover Publications INC, New York (1965).

Publications due to this work

M. Zerroukat and C.R. Chatwin, An explicit unconditionally stable variable time step method for one-dimensional Stefan problems, *International Journal for Numerical Methods in Engineering*, Vol. 35, pp. 1503-1520 (1992).

M. Zerroukat and C.R. Chatwin, An explicit variable time step method for implicit moving boundary problems, Accepted for publication in:
Communications in Applied Numerical Methods.

M. Zerroukat and C.R. Chatwin, A finite difference algorithm for multiple moving boundary problems using real and virtual grid networks,
Journal of Computational Physics (revised).

M. Zerroukat and C.R. Chatwin, Heat flow in materials during melting and resolidification induced by a scanning laser beam, Accepted for publication in:
Proceedings of the 8th International Conference on Numerical Methods in Thermal Problems.

M. Zerroukat and C.R. Chatwin, Computational performance of finite difference methods for transient heat conduction problems, Submitted to:
Computer Methods in Applied Mechanics and Engineering.