A Galerkin Method for Stefan Problems

N. S. Asaithambi

Department of Mathematics and Computer Science
Lincoln University
Lincoln University, Pennsylvania 19352

Transmitted by Melvin Scott

ABSTRACT

Stefan problems are described by a parabolic partial-differential equation, along with two boundary conditions on a moving boundary, which is to be determined as part of the solution. The purpose of this paper is to develop a finite-element method for the solution of a one-dimensional Stefan problem. First, a coordinate transformation is used to transform the changing physical domain into a fixed computational domain. Then, the weak or Galerkin formulation of the initial boundary value problem is used to reduce it to a system of initial-value problems in ordinary differential equations. Several finite-difference marching methods are used to solve the resulting initial-value problems. The method is developed and illustrated using the Stefan problem concerning the heat transfer in an ice–water medium. The computational results are in very good agreement with the results produced earlier by several authors.

1. INTRODUCTION

A variety of transient problems arising in engineering and science involve a domain whose boundary changes its shape and size in time. For example, see Ockendon and Hodgkins [15], Furzerland [8], and Crank [4]. These problems are commonly referred to as moving boundary problems (MBPs). In particular, the MBPs arising in heat conduction and diffusion are called Stefan problems. The existence and uniqueness of the solution of Stefan problems have been studied in detail by Rubinstein [17]. The presence of the moving boundary makes these problems nonlinear (see Carslaw and Jaeger [3]), and hence, their analytical solution is difficult. Approximate analytical methods,

655 Avenue of the Americas, New York, NY 10010 0096-3003/92/$5.00

which yield solutions to Stefan problems in simple closed forms, have been investigated by Goodman [9], Reynolds and Dolton [16], and Gupta and Banik [10]. However, there is neither a check on the accuracy of these solutions, nor a systematic approach to improving the accuracy, if desired. The most commonly used numerical methods are based on finite-differences. Several of these methods may be found in Landau [13], Crank [5], Murray and Landis [14], Crank and Gupta [6], Gupta and Kumar [11], and Asaithambi [1, 2]. A finite-element solution was proposed by Finn and Voroǵlu [7], who solved a sequence of two-point boundary-value problems for each time level. The basic approach of their method was to *guess* the position of the moving boundary, and impose the boundary conditions at the approximate location. In this paper, we present a method which will eliminate this uncertainty by first mapping the changing physical domain into a fixed computational domain. The direct weak or Galerkin formulation of the resulting problem yields a system of initial-value problems in ordinary differential equations, which may be solved by finite-difference marching procedures.

2. STATEMENT OF PROBLEM

We consider the heat transfer in an ice–water medium in the region $0 \leqslant x \leqslant 1$. The portion $0 \leqslant s_1(t) < x < s_2(t) \leqslant 1$ of this region is occupied by the water that undergoes phase change, and the remaining portion by ice. The initial temperature distribution in ice is linear in x, and symmetric about $x = 0.5$. The temperature of the water is assumed to be zero, which is the critical temperature of phase change. The temperature at the fixed surfaces $x = 0$ and $x = 1$ is assumed to be -1 for all time. The problem is to determine $s_1(t)$ and $s_2(t)$ along with the temperature distribution, starting with $s_1(0) = 0.25$, and $s_2(0) = 0.75$. Since the problem is entirely symmetric about $x = 0.5$, it suffices to consider only the region $0 \leqslant x \leqslant 0.5$. We denote by $u(x, t)$ the temperature at a distance x from the fixed surface $x = 0$ at time t, and $s(t)$ the position of the moving boundary as time progresses. In terms of these variables, the mathematical problem in the dimensionless form corresponding to the above physical process may be described by the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \text{ on } 0 < x < s(t), \ t > 0, \tag{2.1}$$

subject to the boundary conditions

$$u = -1, \ x = 0, \ t > 0, \tag{2.2}$$

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

$$\frac{\partial u}{\partial x} = \frac{ds}{dt}, \; x = s(t), t > 0, \tag{2.4}$$

along with the initial conditions

$$u(x,0) = \begin{cases} 4x - 1, & \text{for } 0 \leqslant x \leqslant 0.25, \\ 0, & \text{for } 0.25 \leqslant x \leqslant 0.5, \end{cases} \tag{2.5}$$

$$s(0) = 0.25. \tag{2.6}$$

We wish to determine $s(t)$ for $t > 0$, and $u(x, t)$ for $0 < x < s(t)$ and $t > 0$.

3. NUMERICAL SOLUTION

The development of the numerical method consists mainly of three major steps. They are, (i) coordinate transformation, (ii) weak or Galerkin formulation, and (iii) solution of the resulting initial-value problems.

Coordinate transformation

In order to facilitate easy implementation of finite-difference and finite-element methods for free and moving boundary problems, it has become common practice to first transform the changing physical domain to a fixed computational domain. For the present problem, the simple transformation

$$\xi = x/s(t), \qquad \tau = t, \tag{3.1}$$

will transform the changing domain $0 \leqslant x \leqslant s(t)$, $t > 0$ into the fixed domain $0 \leqslant \xi \leqslant 1$, $\tau > 0$. We will perform all computations in the $\xi - \tau$ domain, still considering $s(\tau)$ as an unknown. For convenience, we will henceforth abbreviate $s(\tau)$ by s. Under the transformation (3.1), the governing mathematical problem (2.1)–(2.6) takes the form

$$\frac{\partial u}{\partial \tau} = \frac{\xi}{s}\frac{ds}{d\tau}\frac{\partial u}{\partial \xi} + \frac{1}{s^2}\frac{\partial^2 u}{\partial \xi^2} \text{ on } 0 < \xi < 1, \qquad \tau > 0, \tag{3.2}$$

subject to the boundary conditions

$$u = -1, \xi = 0, \tau > 0, \tag{3.3}$$

$$u = 0, \xi = 1, \tau > 0, \tag{3.4}$$

$$\frac{1}{s}\frac{\partial u}{\partial \xi} = \frac{ds}{d\tau}, \; \xi = 1, \tau > 0, \tag{3.5}$$

along with the initial conditions

$$u(\xi,0) = \begin{cases} \xi - 1, & \text{for } 0 \leqslant \xi \leqslant 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3.6}$$

$$s(0) = 0.25. \tag{3.7}$$

We wish to determine $s(\tau)$, and $u(\xi, \tau)$ on $0 < \xi < 1$ for $\tau > 0$. Next, we consider the weak or Galerkin formulation of the problem described by (3.2)–(3.7).

Weak or Galerkin Formulation

If u satisfies (3.2), then it also satisfies

$$\int_0^1 \left(\frac{\partial u}{\partial \tau} - \frac{\xi}{s} \frac{ds}{d\tau} \frac{\partial u}{\partial \xi} - \frac{1}{s^2} \frac{\partial^2 u}{\partial \xi^2} \right) w(\xi)\, d\xi = 0, \tag{3.8}$$

for any continuous $w(\xi)$, called a weight function. Thus, (3.8) is a necessary condition for u to be a solution of (3.1). It is well known that, under suitable smoothness assumptions on u, (3.8) will also be a sufficient condition. If the weight function $w(\xi)$ also satisfies $w(0) = w(1) = 0$, then (3.8) is equivalent to

$$\int_0^1 \left(\frac{\partial u}{\partial \tau} w(\xi) - \frac{\xi}{s} \frac{ds}{d\tau} \frac{\partial u}{\partial \xi} w(\xi) + \frac{1}{s^2} \frac{\partial u}{\partial \xi} \frac{\partial w}{\partial \xi} \right) d\xi = 0, \tag{3.9}$$

which is obtained by using integration by parts on the last term in the integrands in (3.8). Thus, we will solve (3.2) by solving its weak or Galerkin form given by (3.9).

For obtaining a finite-element approximation to the solution u of (3.9), we begin by subdividing the region $0 \leqslant \xi \leqslant 1$, using the uniformly spaced grid points $\xi_j = jh$, where $h = 1/N$ is called the mesh size, and N the number of subintervals. Then, we proceed to determine approximations of the form

$$u^h(\xi, \tau) = \sum_{j=0}^{N} u_j(\tau) \phi_j(\xi), \tag{3.10}$$

where $\{\phi_j(\xi)\}_{j=0}^{N}$ are the hat functions defined by

$$\phi_j(\xi) = \begin{cases} \dfrac{\xi - \xi_{j-1}}{h}, & \text{for } \xi_{j-1} \leqslant \xi \leqslant \xi_j, \\ \dfrac{\xi_{j+1} - \xi}{h}, & \text{for } \xi_j \leqslant \xi \leqslant \xi_{j+1}, \\ 0, & \text{otherwise}. \end{cases} \tag{3.11}$$

Letting $u_0(\tau) \equiv -1$ and $u_N(\tau) \equiv 0$ in (3.10) yields an approximation $u^h(\xi, \tau)$ that satisfies the boundary conditions (3.3) and (3.4). The computational problem is to obtain the time-dependent coefficients $u_j(\tau)$ for $j = 1, 2, \cdots, N-1$. This may be accomplished by using $N-1$ appropriate weight functions $w(\xi)$. The usual practice is to take $w(\xi) = \phi_i(\xi)$ for $i = 1, 2, \cdots, N-1$. Thus, the use of (3.10) and (3.11) in (3.9) yields the system of equations given by

$$a_i \frac{du_{i-1}}{d\tau} + b_i \frac{du_i}{d\tau} + c_i \frac{du_{i+1}}{d\tau} = p_i u_{i-1} + q_i u_i + r_i u_{i+1},\ i = 1, 2, \cdots, N-1, \tag{3.12}$$

where

$$\begin{aligned} a_i &= \frac{1}{6}, & p_i &= \frac{1}{s}\left[\frac{(1-3i)}{6}\frac{ds}{d\tau} + \frac{1}{sh^2}\right], \\ b_i &= \frac{2}{3}, & q_i &= \frac{1}{s}\left[-\frac{1}{3}\frac{ds}{d\tau} - \frac{2}{sh^2}\right], \\ c_i &= \frac{1}{6}, & r_i &= \frac{1}{s}\left[\frac{(1+3i)}{6}\frac{ds}{d\tau} + \frac{1}{sh^2}\right], \end{aligned} \tag{3.13}$$

in which we have abbreviated $u_i(\tau)$ as u_i. Since by (3.10) it follows that

$$u^h(\xi, \tau) = u_{N-1}\phi_{N-1}(\xi) + u_N\phi_N(\xi) \text{ for } \xi \in [1-h, 1],$$

and $u_N \equiv 0$, the boundary condition (3.5) becomes

$$\frac{ds}{d\tau} = \frac{1}{s} u_{N-1} \phi'_{N-1}(1) = -\frac{1}{sh} u_{N-1}. \tag{3.14}$$

Thus the initial-boundary problem described by (3.2)–(3.7) has been changed to the system of initial-value problems described by (3.12)–(3.14), subject to the initial conditions

$$u_i(0) = ih - 1, \quad i = 1, 2, \cdots, N - 1, \quad s(0) = 0.25. \tag{3.15}$$

Next, we consider the solution of (3.12)–(3.15).

Solution of the Initial-Value Problems

Starting with the initial values given in (3.15), we wish to obtain $u_j(\tau_k)$ where $\tau_k = k\Delta\tau)$ for $k = 1, 2, \cdots$ for a chosen step size $\Delta\tau$. By adding the differential equations $du_0/d\tau = 0$ and $du_N/d\tau = 0$ to (3.12), we obtain the matrix form

$$M\frac{d\mathbf{u}}{d\tau} = A(s)\mathbf{u}, \tag{3.16}$$

where M and $A(s)$ are the $(N + 1) \times (N + 1)$ matrices and $\mathbf{u}$ is the $(N + 1)$-vector given by

$$M = \text{tridiag}\left[\frac{1}{6}\,\frac{2}{3}\,\frac{1}{6}\right],$$

$$A(s) = \text{tridiag}\,[\,p_i \; q_i \; r_i\,],$$

$$\mathbf{u}^T = [u_0 \cdots u_N]. \tag{3.17}$$

Note that the first and the last row of M consist of the only nonzero entry 1 on the main diagonal, and the first and the last rows of $A(s)$ are zero. In order to keep the notation uniform, we write (3.14) as

$$\frac{ds}{d\tau} = -\frac{1}{sh} u_{N-1} = g(\tau, s, \mathbf{u}). \tag{3.18}$$

Traditional finite-difference methods may be used to solve (3.16) and (3.18).

For example, if we let $\mathbf{u}^k$ denote an approximation to $\mathbf{u}(k\Delta\tau)$ and s^k denote an approximation to $s(k\Delta\tau)$, the forward-difference (or Euler's) method for the solution of (3.16) and (3.18) amounts to solving the equations

$$M\mathbf{u}^{k+1} = M\mathbf{u}^k + \Delta\tau A(s^k)\mathbf{u}^k, \tag{3.19a}$$

$$s^{k+1} = s^k + \Delta\tau g\left(\tau_k, s^k, \mathbf{u}^k\right). \tag{3.19b}$$

We had difficulty obtaining reasonable solutions for moderate values of the mesh size h and the step size $\Delta\tau$ when (3.19a) and (3.19b) were used. The numerical results exhibited an oscillatory behavior even for moderate mesh sizes h. This is not surprising since (3.2) is actually a problem of the convection-diffusion type, and numerical methods are known to produce oscillatory solutions if the mesh size h is not chosen carefully. An accepted alternative in finite-difference methods is to use what are known as *upwind difference schemes*. For the finite-element method, the corresponding alternative is to choose the weight functions $w(\xi)$, which are different from the $\phi_i(\xi)$, defined in (3.11). An accepted set of weight functions (for example, see Hall and Porsching [12]) $\{\varphi_i(\xi)\}$ is obtained by adding quadratic corrections to $\phi_i(\xi)$. We set $\varphi_i(\xi) = \phi_i(\xi) + \alpha\beta_i(\xi)$, where the quadratic corrections $\beta_i(\xi)$ are given by

$$\beta_i(\xi)\begin{cases} \dfrac{3(\xi - \xi_{i-1})(\xi_i - \xi)}{h^2}, & \text{for } \xi_{i-1} \leqslant \xi \leqslant \xi_i, \\ \dfrac{3(\xi - \xi_i)(\xi_{i+1} - \xi)}{h^2}, & \text{for } \xi_i \leqslant \xi \leqslant \xi_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \tag{3.20}$$

and α is some constant. Generally, $\alpha > 0$ if the convective term has a positive coefficient, and $\alpha < 0$ otherwise. With the use of these new weight functions, the initial-value problems are still of the same form as (3.12), with

$$a_i = \frac{1}{6} + \frac{\alpha}{4}, \quad p_i = \frac{1}{s}\left[\left(\frac{(1-3i)}{6} + \alpha\frac{(1-2i)}{4}\right)\frac{ds}{d\tau}\right.$$
$$\left. + \left(\frac{(1-6\alpha(i-1))}{sh^2}\right)\right],$$

$$b_i = \frac{2}{3}, \qquad q_i = \frac{1}{s}\left[-\frac{1}{3}\frac{ds}{d\tau} - \left(\frac{(2+6\alpha(1-2i))}{sh^2}\right)\right],$$

$$c_i = \frac{1}{6} - \frac{\alpha}{4}, \quad r_i = \frac{1}{s}\left[\left(\frac{(1+3i)}{6} - \alpha\frac{(1+2i)}{4}\right)\frac{ds}{d\tau} + \left(\frac{(1-6\alpha i)}{sh^2}\right)\right], \tag{3.21}$$

which may be used to define the matrices M and $A(s)$ of (3.16), correspondingly. Note that putting $\alpha = 0$ in (3.21) results in (3.13). In addition to this modification of the weight functions, we used a 2-stage Runge-Kutta marching procedure instead of the Euler scheme defined by (3.19a)–(3.19b) for the solution of (3.16) and (3.18). In order to describe the 2-stage Runge-Kutta procedure, let us rewrite (3.16) as $d\mathbf{u}/d\tau = \boldsymbol{f}(\tau, s, \mathbf{u})$, where

$$\mathbf{f}(\tau, s, \mathbf{u}) = M^{-1}A(s)\mathbf{u}. \tag{3.22}$$

The 2-stage Runge-Kutta procedure sets

$$\begin{aligned} \mathbf{f}_1 &= \Delta\tau\mathbf{f}(\tau_k, s^k, \mathbf{u}^k), & \mathbf{f}_2 &= \Delta\tau\mathbf{f}(\tau_k + \Delta\tau, s^k + g_1, \mathbf{u}^k + \mathbf{f}_1), \\ g_1 &= \Delta\tau g(\tau_k, s^k, \mathbf{u}^k), & g_2 &= \Delta\tau g(\tau_k + \Delta\tau, s^k + g_1, \mathbf{u}^k + \mathbf{f}_1), \end{aligned} \tag{3.23}$$

and obtains $\boldsymbol{u}^{k+1}$ and s^{k+1} using

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \frac{1}{2}(\mathbf{f}_1 + \mathbf{f}_2),$$

$$s^{k+1} = s^k + \frac{1}{2}(g_1 + g_2). \tag{3.24}$$

Note that we do not have to invert the matrix M explicitly, since $\mathbf{f}$ may be obtained by solving the linear system $M\mathbf{f} = A(s)\boldsymbol{u}$.

Even though the oscillations in the numerical solutions were suppressed using (3.12), (3.21), and (3.24), the step size $\Delta\tau$ had to be restricted for stability purposes. This is not surprising either, since we know that even for linear parabolic problems, explicit methods in general restrict the step size $\Delta\tau$ according to the relation $\Delta\tau/h^2 \leqslant c$ for some constant c. In other words, any decrease in the mesh size h should be accompanied by a corresponding decrease in $\Delta\tau$. This happens to be a severe restriction. Thus, it is reasonable to use an implicit marching procedure (such as the backward-difference scheme or the Crank-Nicolson scheme). We have used the backward-difference scheme for the present work, which easily extends to the Crank-Nicolson scheme. For the derivation of the backward-difference method, we discretize (3.16) and (3.18) in the form

$$M\mathbf{u}^{k+1} = M\mathbf{u}^k + \Delta\tau A(s^{k+1})\mathbf{u}^{k+1}, \tag{3.25a}$$

$$s^{k+1} = s^k + \Delta\tau g\left(\tau_{k+1}, s^{k+1}, \mathbf{u}^{k+1}\right). \tag{3.25b}$$

Compare this with (3.19a) and (3.19b). It is apparent that while (3.19a)–(3.19b) involve only the solution of the linear system of equation (3.19a) for $\boldsymbol{u}^{k+1}$, and a direct evaluation of (3.19b) for s^{k+1}, the backward-difference method requires the numerical solution of the nonlinear system of equations (3.25a)–(3.25b), which we solve iteratively. If we denote by $\boldsymbol{u}^{k+1,l}$ and $s^{k+1,l}$ iterates for $\boldsymbol{u}^{k+1}$ and s^{k+1}, respectively, then we begin the iterative process by setting

$$\mathbf{u}^{k+1,0} = \mathbf{u}^k, \; s^{k+1,0} = s^k. \tag{3.26}$$

Then we solve

$$\left(M - \Delta\tau A(s^{k+1,l})\right)\mathbf{u}^{k+1,l+1} = M\mathbf{u}^k, \tag{3.27}$$

and set

$$s^{k+1,l+1} = s^k + \Delta\tau g\left(\tau_{k+1}, s^{k+1,l}, \mathbf{u}^{k+1,l+1}\right), \tag{3.28}$$

for $l = 0, 1, \cdots$ until a prescribed error tolerance is met. The $ds/d\tau$ term in $A(s^{k+1,l})$ is discretized as $(s^{k+1,l} - s^k)/\Delta\tau$. Note that the use of (3.27) amounts to an increase in the computational effort needed. However, this extra computational effort is justified since, in general, there is no restriction on the step size $\Delta\tau$ any more. We summarize the numerical results in the next section.

4. NUMERICAL RESULTS AND DISCUSSION

An exact solution for the problem described by (2.1)–(2.6), provided by Rubinstein [17], has been used for comparison by the many authors who have computed the solution numerically. We will compare our results with the exact solution of Rubinstein [17] and the finite-element solution of Finn & Varoğlu [7].

As we mentioned earlier, for the present method, an explicit marching procedure imposes severe restriction on the size of $\Delta\tau$. For the 2-stage Runge-Kutta procedure, we obtained reasonable solutions for $\Delta\tau = 0.0001$, and $N = 10$. There were no obvious oscillations in the computed solutions. However, the accuracy of the solution was better when the quadratic corrections were used for the weight functions. We obtained excellent results for $\alpha = -0.002$. However, the choice of the parameter α was very difficult. Table 1 shows these results, along with the numerically evaluated exact solution of Rubinstein [17].

Shown in Table 2 are the results of the present method using the backward-difference or implicit marching the iterative solution of the result-

TABLE 1
EXPLICIT MARCHING

t	Rubinstein [4]	Present ($\alpha = 0$)	Present ($\alpha = -0.002$)
0.01	0.2813	0.2823	0.2817
0.02	0.3079	0.3094	0.3082
0.03	0.3321	0.3342	0.3324
0.04	0.3545	0.3571	0.3548
0.06	0.3955	0.3991	0.3958
0.08	0.4326	0.4370	0.4329
0.10	0.4668	0.4719	0.4671

[1] $\Delta\tau = 0.0001$ $N = 10$ ($h = 0.1$).

TABLE 2
IMPLICIT MARCHING

t	Finn & Varoğlu [16]	Present $N = 10$, $\Delta\tau = 0.01$	Present $N = 20$, $\Delta\tau = 0.005$
0.01	0.2806	0.2803	0.2808
0.02	0.3072	0.3069	0.3074
0.03	0.3315	0.3312	0.3316
0.04	0.3541	0.3537	0.3541
0.06	0.3956	0.3951	0.3952
0.08	0.4333	0.4326	0.4326
0.10	0.4682	0.4671	0.4670

ing nonlinear system of algebraic equations, along with the finite-element solution of Finn and Varoğlu [7] were produced using 20 elements ($N = 20$) with a step size of $\Delta t = 0.005$. The accuracy of the present results, obtained with explicit marching a quadratic corrections, (especially those corresponding to $\alpha = -0.002$) is certainly superior to those of Finn and Varoğlu [7]. However, the step size $\Delta\tau = 0.0001$ is very small and hence requires more number of steps to march to a given time.

On the other hand, as is evident from Table 2, the present method has produced more accurate results using an implicit marching process even with fewer number of elements ($N = 10$) and larger step size ($\Delta\tau = 0.01$) than those required by Finn and Varoğlu [7]. In addition, the present results (obtained using implicit marching) are also superior to the results of many other authors reported by Gupta and Banik [10].

REFERENCES

1 N. S. Asaithambi, Numerical modeling of Stefan problems, *Mathematical Computer Modelling* 14:139–144 (1990).

2 N. S. Asaithambi, On a variable time step method for the one-dimensional Stefan problem, *Comput. Methods. Appl. Mech. Engrg.* 71:1–13 (1988).

3 H. S. Carslaw and J. C. Jaeger, *Conduction of Heat in Solids*, 2nd ed., Oxford University Press, London, 1959.

4 J. Crank, *Free and Moving Boundary Problems,* Clarendon Press, Oxford, England, 1984.

5 J. Crank, Two methods for the numerical solution of moving boundary problems in diffusion and heat flow, *Journal of Mechanical Applied Mathematics*, 10:220–231 (1957).

6 J. Crank and R. S. Gupta, A moving boundary problem arising from the diffusion in absorbing tissue, *Journal of the Institute of Mathematical Applications* 10:19–33 (1972).

7 W. D. L. Finn and E. Varoğlu, Finite element solution of the Stefan problem, in *The Mathematics of Finite Elements and Applications*, MAFELAP 1978, (J. R. Whiteman, Ed.), Academic Press, New York, 1979.

8 R. M. Furzeland, A survey of the formulation and solution of free and moving boundary (Stefan) Problems, *Technical Report* TR/76, Brunel University, Uxbridge, United Kingdom.

9 T. R. Goodman, The heat-balance integral and its application to problems involving a change of phase, *Trans. ASME Ser. EJ. Appl. Mech.*, 80:335–342, (1959).

10 R. S. Gupta and N. C. Banik, Constrained integral method for solving moving boundary problems, *Comput. Methods Appl. Mech. Engrg.* 67:211–221 (1988).

11 R. S. Gupta and D. Kumar, A modified variable time step method for the one-dimensional Stefan problem, *Comput. Methods Appl. Mech. Engrg.* 23:101–109 (1980).

12 C. A. Hall and T. A. Porsching, *Numerical Analysis of Partial Differential Equations*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

13 H. G. Landau, Heat conduction in melting solids, *Quart. Appl. Math.* 8:81–94 (1950).

14 W. D. Murray and F. Landis, Numerical and machine solutions of transient heat conduction problems involving melting of freezing, *Journal of Heat Transfer* 81:106–112, (1959).

15 J. R. Ockendon and W. R. Hodgkins, Eds., *Moving Boundary Problems in Heat Flow and Diffusion*, Clarendon Press, Oxford, England, 1975.

16 W. C. Reynolds and T. A. Dolton, Use of integral methods in transient heat transfer analysis, *American Society of Mechanical Engineers* paper 58-A-248, 1958.

17 L. I. Rubinstein, The Stefan problem, *Transactions of the Mathematical Monographs* 27 (1971).