Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 764532, 10 pages
http://dx.doi.org/10.1155/2014/764532

Research Article

Two Simple Numerical Methods for the Free Boundary in One-Phase Stefan Problem

Seung Hyun Kim

Cambridge School of Weston, 45 Georgian Road, Weston, MA 02493, USA

Correspondence should be addressed to Seung Hyun Kim; skim2014@csw.org

Received 2 January 2014; Revised 4 May 2014; Accepted 19 May 2014; Published 3 June 2014

Academic Editor: Yansheng Liu

We present two simple numerical methods to find the free boundary in one-phase Stefan problem. The work is motivated by the necessity for better understanding of the solution surface (temperatures) near the free boundary. We formulate a log-transform function with the unfixed and fixed free boundary that has Lipschitz character near free boundary. We solve the quadratic equation in order to locate the free boundary in a time-recursive way. We also present several numerical results which illustrate a comparison to other methods.

1. Introduction

The free boundary in a Stefan problem can give prediction of the boundary between two opposing materials. For a famous example, the boundary of an ice in water can be found according to time. In this case, water above zero degrees Celsius and ice at zero degrees Celsius are the two materials. Stefan [1] first gave conditions necessary to find the free boundary. Stefan problems can be used for the computation of diffusion [2], preservation of human blood [3], solidification of metals [4], and estimation of the thermal state in the mantle [5]. Many researchers have studied to figure out the effective method in order to approximate the free boundary.

Methods can be largely separated into three groups: analytic methods, mixed methods of numerical and analytic methods, and another method which is numerical methods. Analytic methods are developed by researchers, such as exact solution by using a fractional differential equation of Liu and Xu [6] and an analytical solution with variable latent heat of Voller et al. [7]. Mixed methods are developed by researchers, such as the approximate analytical and a numerical solution method by using a linear perturbation method of Yigit [8] and the analytical and numerical solution method with moving boundary of Lorenzo-Trueba and Voller [9]. Numerical methods are developed by researchers, such as the numerical method with automatic differentiation of Asaithambi [10] and the finite element approximation method of Nedoma [11], Słota [12], and Hinze and Ziegenbalg [5].

Among many methods, a boundary immobilization method (BIM), suggested by Landau [13], fixes the free boundary by using a transform method. The benefit of this function is its simple domain. Whatever the shape of the free boundary is, this method changes the original domain to a rectangular domain. After the BIM method, Kutluay et al. [14] suggested another method called variable space grid (VSG) method.

The main contribution of this paper is the development of two simple numerical methods to find the free boundary in a time-recursive way. Our result is motivated by the necessity for better understanding of the solution surface near free boundary. We adopt the boundary immobilization [13] to change the unknown free boundary to a known and fixed boundary. On the other hand, we adopt the constant number of space intervals between two boundaries [14].

We exploit a log-transform function with the unfixed and fixed free boundary that has Lipschitz character which reduces the accumulating error of the solution surface near the free boundary.

We apply extrapolation near the free boundary. Thus, we can determine the free boundary by solving a quadratic equation in a time-recursive way. Our method also provides fast and accurate results for calculating the free boundary and determining solutions.

The structure of the paper is as follows. Section 2 presents the model formulation. The log-transform function with the unfixed and fixed free boundary to calculate the free boundary is presented in Section 3. Numerical results and comparative studies are presented in Section 4. Section 5 summarizes the paper.

2. Problem Formulation

In this section, we present a mathematical formula for determining solution of one-phase Stefan problem.

The temperature of liquid is denoted by $u(x, t)$, where x is the liquid/solid space for $x \in [0, \infty)$ and t is the time to expiration for $t \in [0, T]$.

As seen in the previous article by Stefan [1], the temperature of liquid is considered the solution to a free boundary problem with a parabolic PDE. We suppose that the free boundary $S(t)$ is continuously increasing with $S(0) = 1$. The region where it is free boundary to hold, generally called the liquid region, is defined as $\Omega_L = [0, S(t)] \times [0, T]$, and the region where it is free boundary, generally called the solid region, is defined as $\Omega_S = (S(t), \infty) \times [0, T]$. Then, $u(\cdot, \cdot)$ and $S(\cdot)$ uniquely solve

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \text{in } \Omega_L, \tag{1}$$

being subject to the boundary conditions

$$-u_x(0, t) = f(t), \tag{2}$$

$$u(S(t), t) = 0. \tag{3}$$

The location of the liquid/solid interface is given by the heat balance equation known as Stefan condition

$$-u_x(S(t), t) = \frac{dS(t)}{dt}, \tag{4}$$

$$u(x, t) = 0, \quad \text{in } \Omega_S. \tag{5}$$

Initially there is liquid region which implies the condition

$$\begin{gathered} S(0) = 1, \\ u(x, 0) = 0. \end{gathered} \tag{6}$$

Referring to [14], this problem has the following exact solution for the liquid temperature distribution $u(x, t)$ and the interface location $S(t)$, respectively:

$$u(x, t) = \exp(t - x + 1) - 1, \quad \text{in } \Omega_L, \tag{7}$$

$$S(t) = t + 1, \tag{8}$$

where $T = 1$ and plug $f(t) = \exp(t + 1)$ into (2). The exact solution (7) can be used to compare the numerical solutions and can also be used to initialize the numerical schemes.

A VSG method, proposed in Kutluay et al. [14], which uses a fixed node number between a fixed boundary ($x = 0$) and a moving boundary ($x = s(t)$), was kept constant and equal to N so that the moving boundary always lays on the Nth grid. They derive the equation and the boundary conditions with respect to $u(x, t)$ as follows. First, the process finds the difference between the $S(t_m)$ and $S(t_{m+1})$ and divides it by N, the number of nodes. In the case below, they assign $N + 1$ as the number of x and $M + 1$ as the number of t. Therefore, $\Delta t = T/M$ so that $t_m = m\Delta t$ where $T = 1$ is the maximum of t. Similarly, $\Delta x = S(t)/N$ so that $x_n = n\Delta x$. We also define $u_n^m = u(x_n, t_m)$, $S_m = S(t_m)$, $S' = dS/dt$, and $x_n^m = x_n(t_m)$. Consider

$$\begin{gathered} dS = S_{m+1} - S_m, \qquad dx_n = x_n^{m+1} - x_n^m \\ n = 0, 1, 2, \ldots, N, \; m = 0, 1, 2, \ldots, M - 1. \end{gathered} \tag{9}$$

The ratio $dS : dx_n$ equals the ratio $S_m : x_n^m$. Thus,

$$\begin{gathered} \frac{dx_n}{dt} = \frac{x_n^m}{S_m}\frac{dS}{dt} \\ u_t\big|_n = u_x\big|_t x'(t)\big|_n + u_t\big|_x \\ \therefore u_t = u_x \frac{x_n}{S}\frac{dS}{dt} + u_{xx}. \end{gathered} \tag{10}$$

Since $-u_x(0, t) = f(t)$,

$$u_{-1}^m = u_1^m + 2\Delta x^m f(t). \tag{11}$$

When $f(t) = \exp(t + 1)$,

$$u_{-1}^m = u_1^m + 2\Delta x^m \exp(t_m + 1). \tag{12}$$

By discretizing (10), they get

$$\begin{aligned} u_n^{m+1} = u_n^m &+ \frac{x_n^m S' \Delta t}{2 S_m \Delta x^m}\left(u_{n+1}^m - u_{n-1}^m\right) \\ &+ \frac{\Delta t}{(\Delta x^m)^2}\left(u_{n+1}^m - 2u_n^m + u_{n-1}^m\right). \end{aligned} \tag{13}$$

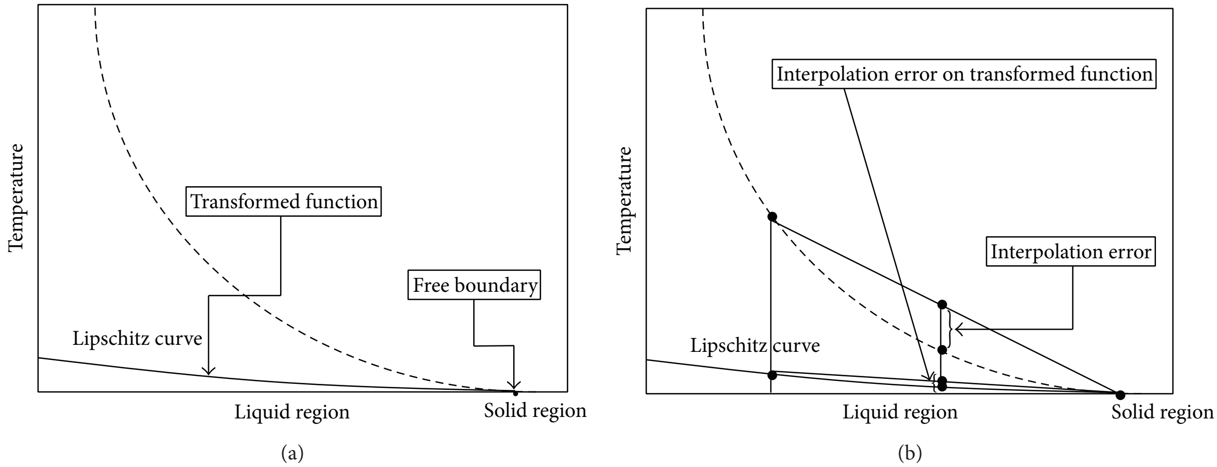
Plugging (12) into (13), they obtain the following:

$$\begin{aligned} u_0^{m+1} = \left(1 - \frac{2\Delta t}{(\Delta x^m)^2}\right) u_0^m &+ \frac{2\Delta t}{(\Delta x^m)^2} u_1^m \\ &+ \left(\frac{2\Delta t}{\Delta x^m} - \frac{x_i^m S' \Delta t}{S_m}\right)\exp(t_m + 1). \end{aligned} \tag{14}$$

In order to get $S(t)$, they use

$$\begin{gathered} S_{m+1} = S_m - \frac{\Delta t}{2\Delta x^m}\left(3u_N^m - 4u_{N-1}^m + u_{N-2}^m\right) \\ \text{where} \quad m = 0, 1, 2, \ldots. \end{gathered} \tag{15}$$

A boundary immobilization (BIM) method, proposed in Landau [13], uses a change in variables to transform the free boundary problem into a nonlinear problem on a fixed domain. The following transformation of state variable serves for such a purpose. The main point of BIM method is transformation. The transform function $\xi = x/S(t)$ changes



(a)

(b)

Figure 1: Log-transformation from $u(x,t)$ to $w(x,t)$.

any shape of domain to a rectangular domain. $\xi = 0$ when $x = 0$, and $\xi = 1$ when $x = S(t)$. This fixes the free boundary to $\xi = 1$. In the case below, he assigns $N + 1$ as the number of ξ $(0 \le \xi \le 1)$ and $M + 1$ as the number of t. Therefore, $\Delta t = T/M$ so that $t_m = m\Delta t$ where T is the maximum of t. Similarly, $\Delta\xi = 1/N$ so that $\xi_n = n\Delta\xi$. He also define $u_n^m = u(\xi_n, t_m)$, $S_m = S(t_m)$, and $S' = dS/dt$

$$u_t\big|_x = u_\xi \xi_t + u_t\big|_\xi = -\frac{x}{S^2} S'(t) + u_t\big|_\xi. \qquad (16)$$

Thus,

$$u_t = \frac{\xi}{S} S' u_\xi + \frac{1}{S^2} u_{\xi\xi}, \quad 0 < \xi < 1, \ t > 0. \qquad (17)$$

When $\xi = 0$,

$$u_\xi = -S \exp(t + 1). \qquad (18)$$

By using central scheme, he gets

$$u_{-1}^m = u_1^m + 2\Delta\xi S_m \exp(t_m + 1). \qquad (19)$$

Discretizing, he obtains

$$\begin{aligned} u_n^{m+1} = u_n^m &+ \frac{\xi_n S' \Delta t}{2S_m \Delta\xi} \left(u_{n+1}^m - u_{n-1}^m\right) \\ &+ \frac{\Delta t}{S_m^2 \Delta\xi^2} \left(u_{n+1}^m - 2u_n^m + u_{n-1}^m\right). \end{aligned} \qquad (20)$$

Plugging (19) into (20) when $n = 0$, he gets

$$\begin{aligned} u_0^{m+1} = &\left(1 - \frac{2\Delta t}{S_m^2 \Delta\xi^2}\right) u_0^m + \frac{2\Delta t}{S_m^2 \Delta\xi^2} u_1^m \\ &+ \left(\frac{2\Delta t}{S_m \Delta\xi} - S' \xi_0 \Delta t\right) \exp(t_m + 1). \end{aligned} \qquad (21)$$

He uses the following equation in order to trace S_m:

$$\begin{aligned} S_{m+1} = S_m - &\frac{\Delta t}{2\Delta\xi} \left(3u_N^m - 4u_{N-1}^m + u_{N-2}^m\right) \\ &\text{where } m = 0, 1, 2, \ldots, M - 1. \end{aligned} \qquad (22)$$

3. Log-Transform Function

In this section, we present a log-transform function with the unfixed free boundary and fixed free boundary that can determine the free boundary by solving a quadratic equation in a time-recursive way. Under the assumption of the Stefan problem, the time for free boundary can be shown to be the first hitting time of a boundary, the free boundary, in the plane consisting of pairs of the space and the time to expiration. Namely, the temperature curve of Stefan problem touches the line representing the intrinsic value tangentially. With a careful examination of the solution surface near the free boundary, we find a Lipschitz surface which reduces the accumulating error of the solution surface near free boundary. To find the free boundary, we present a log-transform function with the unfixed and fixed free boundary that has Lipschitz character near the free boundary as follows:

$$w(x,t) := \ln(u(x,t) + 1). \qquad (23)$$

The transformed function $w(x,t)$ provides that the solution surface in Ω_S is a horizontal plane, and it is an inclined plain in Ω_L. Namely, this function forms a gradual slope with the hyperplane corresponding to the solid region, thereby making the interpolation error more easily reducible when finding the free boundary (see Figure 1). $w(x,t)$ also has a Lipschitz character with nonsingularity in $(0, S(t))$. More precisely, we have $|S'(t)|/(f(t) + 1) < |w_x(S(t),t)| = |w_x(S(t),t)/(w(S(t),t) + 1)| = |-S'(t)/(w(S(t),t) + 1)| \le |S'(t)| < \infty$. Then, $w(x,t)$ is Lipschitz continuous and a natural candidate function for computation in $[0, S(t)]$. We also calculate the partial derivative with respect to x in (10), (17).

Figure 1(a) shows that $u(x,t)$ is transformed to $w(x,t)$ and Figure 1(b) shows reduced interpolation error near the free boundary.

We find the log-transform function with the unfixed and fixed free boundary to decide the free boundary by the Taylor series.

Table 1: Values in the solution.

Safety parameter ρ	RMSE	$S(t = 0.5)$	$u(0.5, T)$	$u(1.0, T)$	$u(1.5, T)$
0.0060	0.000407836	1.4998795	3.4807649	1.7177973	0.6484301
0.0062	0.000407824	1.4998786	3.4807643	1.7177965	0.6484289
0.0064	0.000407814	1.4998777	3.4807638	1.7177957	0.6484277
0.0066	0.000407806	1.4998767	3.4807632	1.7177949	0.6484264
0.0068	0.000407802	1.4998758	3.4807627	1.7177941	0.6484252
0.0070	0.000407799	1.4998748	3.4807621	1.7177933	0.6484240
0.0072	0.000407800	1.4998739	3.4807616	1.7177925	0.6484227
0.0074	0.000407803	1.4998729	3.4807610	1.7177917	0.6484215
0.0076	0.000407809	1.4998720	3.4807604	1.7177909	0.6484203
0.0078	0.000407817	1.4998711	3.4807599	1.7177901	0.6484190
0.0080	0.000407828	1.4998702	3.4807594	1.7177893	0.6484178
Maximum difference	0.000000037	0.0000093	0.0000055	0.0000080	0.0000123

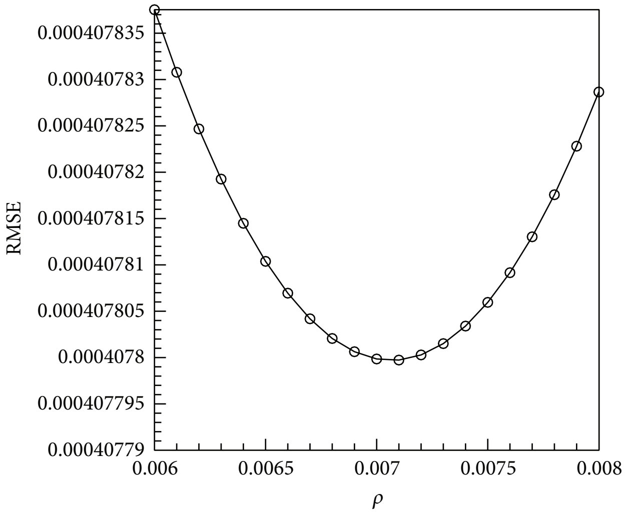


Figure 2: RMSE of the safety parameter ρ in the solution.

3.1. Log-Transform Function with Unfixed Free Boundary (LTUF). From (1), (3), and (4), we obtain the following relations near free boundary:

$$\begin{aligned} u(S(t), t) &= 0 \\ u_x(S(t), t) &= -S'(t) \\ u_{xx}(S(t), t) &= \left(1 + \frac{x_N}{S}\right) S'(t)^2. \end{aligned} \tag{24}$$

From $u(x,t) = e^{w(x,t)} - 1$, we obtain the following relations near free boundary ($w(S(t), t) = 0$):

$$\begin{aligned} u_t &= w_t, \\ u_x &= w_x, \\ u_{xx} &= w_x^2 + w_{xx}. \end{aligned} \tag{25}$$

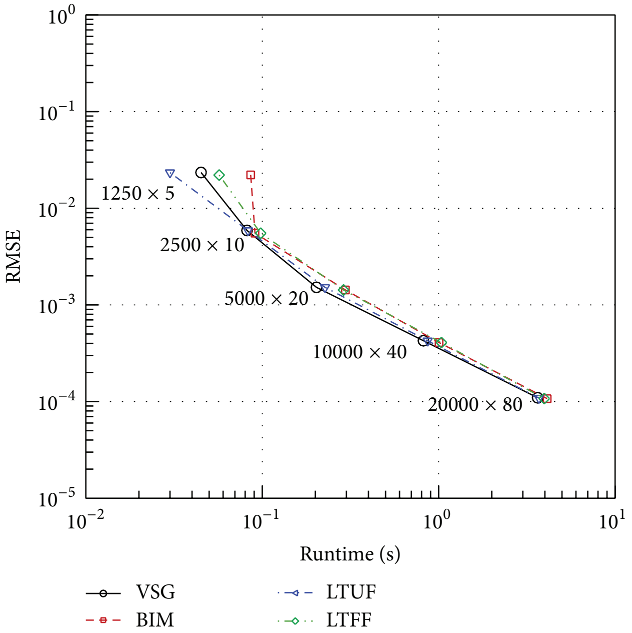


Figure 3: Comparison of RMSE and runtime in the solution.

Plugging (25) into (10), we obtain

$$w_t = \frac{x_i}{S(t)} S'(t) w_x + w_x^2 + w_{xx}. \tag{26}$$

Using $u(x,t) = e^{w(x,t)} - 1$, (24), and (25), we obtain the following:

$$\begin{aligned} w(S(t), t) &= 0 \\ w_x(S(t), t) &= -S'(t) \\ w_{xx}(S(t), t) &= \frac{x_N}{S} S'(t)^2. \end{aligned} \tag{27}$$

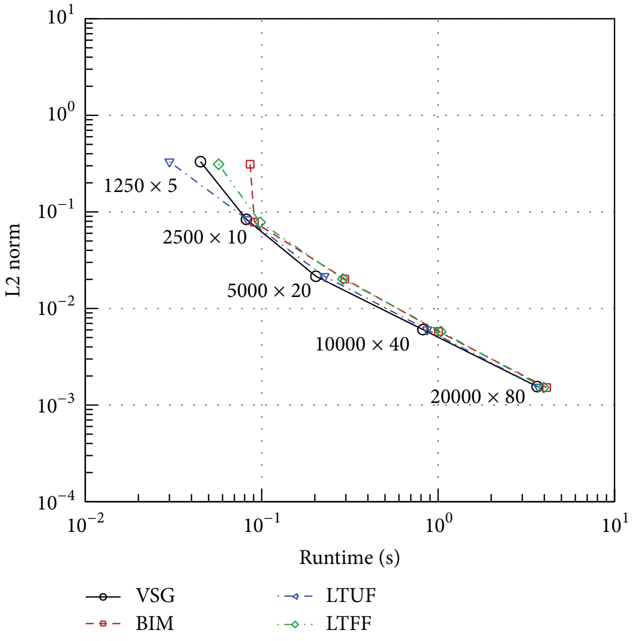


FIGURE 4: Comparison of L2 norm and runtime in the solution.

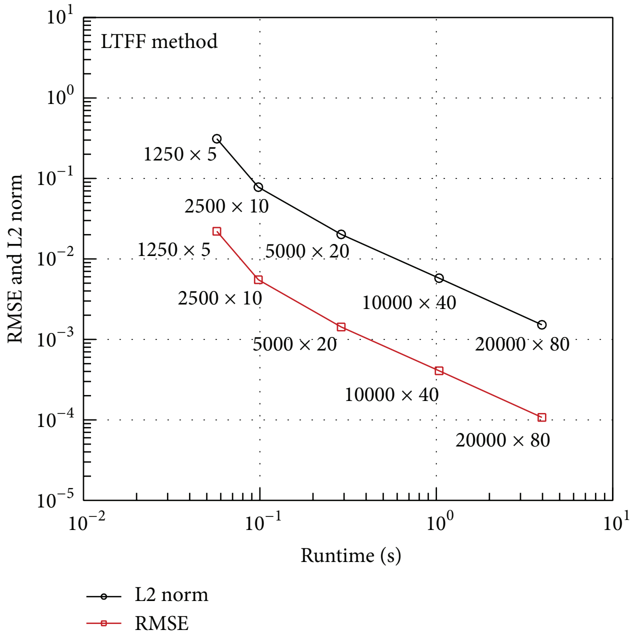


FIGURE 6: Comparison of RMSE and L2 norm with LTFF method in the solution.

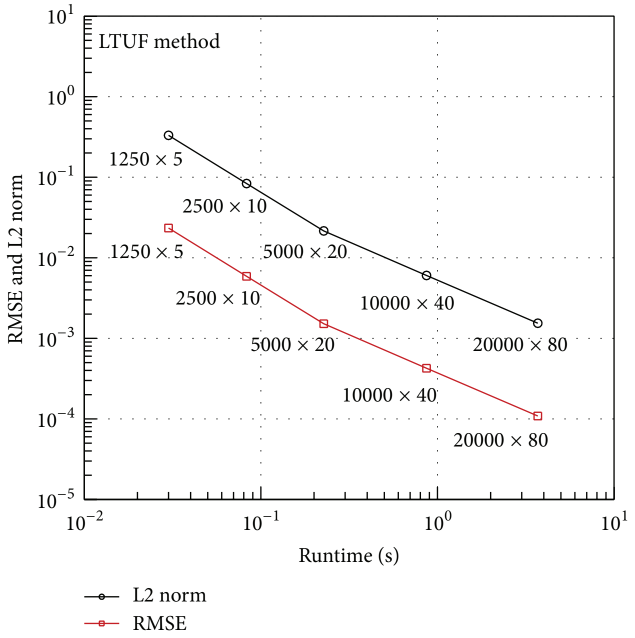


FIGURE 5: Comparison of RMSE and L2 norm with LTUF method in the solution.

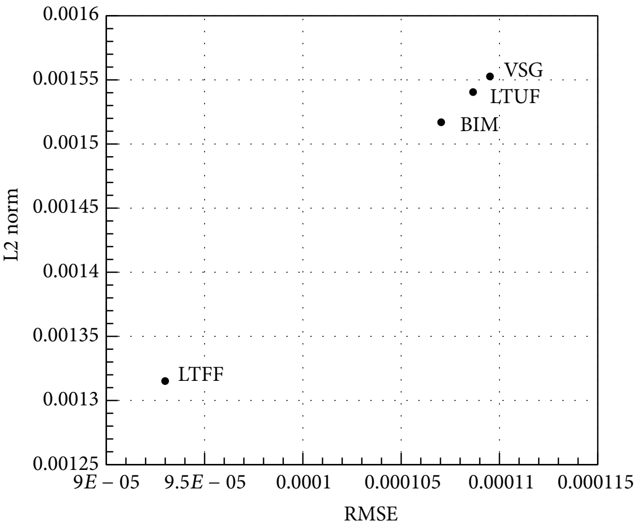


FIGURE 7: Comparison of RMSE and L2 Norm in the solution.

We lower the error by using the optimized safety parameter ρ for a Taylor series with moving node, a method suggested by Kim et al. [15], in order to control the error:

$$w(x,t) = w(S(t),t) + \frac{w_x(S(t),t)}{1!}(x - S(t)) + \frac{w_{xx}(S(t),t)}{2!}(x - S(t))^2 + O\left((x - S(t))^3\right), \tag{28}$$

$$w\left(\widetilde{S}(t),t\right) = w(S(t),t) + \frac{w_x(S(t),t)}{1!}\left(\widetilde{S}(t) - S(t)\right) + \frac{w_{xx}(S(t),t)}{2!}\left(\widetilde{S}(t) - S(t)\right)^2. \tag{29}$$

Using $\widetilde{S}(t) = S(t) - \rho\Delta x$, (27), and (29), we get

$$\frac{x_N}{2S}(-\rho\Delta x)^2 S'(t)^2 + \rho\Delta x S'(t) - \widetilde{w} = 0, \qquad \widetilde{w} = w\left(\widetilde{S}(t),t\right). \tag{30}$$

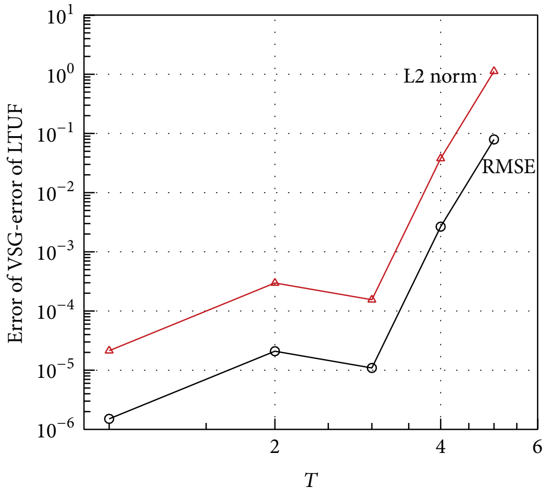


Figure 8: Increasing size of differential error of VSG and LTUF according to final time T in the solution.

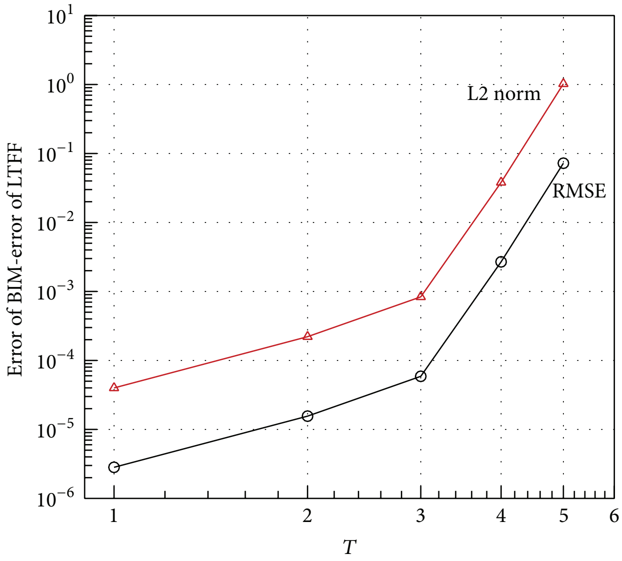


Figure 9: Increasing size of differential error of BIM and LTFF according to final time T in the solution.

Combining (30) with $u_x(S(t),t) < 0$, $S'(t) > 0$, we have

$$S'(t) = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \tag{31}$$

where

$$a = \frac{x_N}{2S}(-\rho\Delta x)^2, \quad b = \rho\Delta x, \quad c = -\widetilde{w}. \tag{32}$$

For discretization $(\Delta x, \Delta t)$, we introduce a two-dimensional mesh in the first quadrant of the $x - t$ plane. From (31) we have

$$S_{m+1} = S_m + \frac{-\widetilde{b} + \sqrt{\widetilde{b}^2 - 4\widetilde{a}\,\widetilde{c}}}{2\widetilde{a}}\Delta t, \tag{33}$$

where

$$\begin{gathered} \widetilde{a} = \frac{x_N^m}{2S_m}(-\rho\Delta x^m)^2, \\ \widetilde{b} = \rho\Delta x^m, \\ \widetilde{c} = -w\left(N\Delta x - \rho\Delta x, m\Delta t\right), \\ m = 0, 1, 2, \ldots, M-1. \end{gathered} \tag{34}$$

When the initial values are given by $w(N\Delta x - \rho\Delta x, t_0)$ (transformed value of the temperature) and S_0 (free boundary) at $t = 0$, we can determine S_1 (free boundary) at Δt using (33) and find w from (26). More importantly, for updating the free boundary our method does not include any iteration until the numerical solution is obtained. So, we repeat the previously mentioned process until $M\Delta t$ and obtain the free boundary in a time-recursive way.

3.2. Log-Transform Function with Fixed Free Boundary (LTFF).

In the case below, we assign $N+1$ as the number of y $(0 \le y \le 1)$ and $M+1$ as the number of t. Therefore, $\Delta t = T/M$ so that $t_m = m\Delta t$, where T is the maximum of t. Similarly, $\Delta y = 1/N$ so that $y_n = n\Delta y$. We also define $\widehat{u}_n^m = \widehat{u}(y_n, t_m)$, $S_m = S(t_m)$, and $S' = dS/dt$. The method starts with

$$y = \frac{x}{S(t)}, \quad \widehat{u}(y,t) = u(x,t). \tag{35}$$

We obtain from (17) and (35)

$$\begin{gathered} \widehat{u}(1,t) = 0, \\ \widehat{u}_y(1,t) = -S(t)S'(t), \\ \widehat{u}_{yy}(1,t) = 2S(t)^2 S'(t)^2. \end{gathered} \tag{36}$$

From $\widehat{u}(y,t) = \exp(w(y,t)) - 1$, we obtain the following relations near free boundary $(w(1,t) = 0)$:

$$\begin{gathered} \widehat{u}_t = w_t, \\ \widehat{u}_y = w_y, \\ \widehat{u}_{yy} = w_y^2 + w_{yy}. \end{gathered} \tag{37}$$

Plugging (35) and (37) into (17), we obtain

$$w_t = \frac{\xi}{S}S'w_\xi + \frac{1}{S^2}w_\xi^2 + \frac{1}{S^2}w_{\xi\xi}. \tag{38}$$

Table 2: Comparison of RMSE and runtime in the solution.

Mesh size ($N \times M$)	(Runtime, RMSE) VSG	 BIM	 LTUF	 LTFF
1250 × 5	(0.045, 0.02346430)	(0.086, 0.02214381)	(0.030, 0.02340634)	(0.057, 0.02206451)
2500 × 10	(0.082, 0.00589595)	(0.091, 0.00552885)	(0.083, 0.00588832)	(0.098, 0.00551475)
5000 × 20	(0.203, 0.00152029)	(0.296, 0.00142859)	(0.227, 0.00151975)	(0.289, 0.00142755)
10000 × 40	(0.820, 0.00042633)	(1.005, 0.00040835)	(0.867, 0.00042561)	(1.037, 0.00040779)
20000 × 80	(3.640, 0.00010951)	(4.122, 0.00010726)	(3.700, 0.00010865)	(3.969, 0.00009299)

Table 3: Comparison of L2 norm and runtime in the solution.

Mesh size ($N \times M$)	(Runtime, L2 norm) VSG	 BIM	 LTUF	 LTFF
1250 × 5	(0.045, 0.33100481)	(0.086, 0.31159113)	(0.030, 0.33101571)	(0.057, 0.31125825)
2500 × 10	(0.082, 0.08358954)	(0.091, 0.07818977)	(0.083, 0.08348142)	(0.098, 0.07799352)
5000 × 20	(0.203, 0.02155389)	(0.296, 0.02020339)	(0.227, 0.02154625)	(0.289, 0.02018861)
10000 × 40	(0.820, 0.00604438)	(1.005, 0.00577497)	(0.867, 0.00603415)	(1.037, 0.00576715)
20000 × 80	(3.640, 0.00155269)	(4.122, 0.00151694)	(3.700, 0.00154049)	(3.969, 0.00131512)

Table 4: Comparison of RMSE and L2 norm with LTUF method in the solution.

Mesh size ($N \times M$)	LTUF (Runtime, RMSE)	 (Runtime, L2 norm)
1250 × 5	(0.030, 0.02340634)	(0.030, 0.33101571)
2500 × 10	(0.083, 0.00588832)	(0.083, 0.08348142)
5000 × 20	(0.227, 0.00151975)	(0.227, 0.02154625)
10000 × 40	(0.867, 0.00042561)	(0.867, 0.00603415)
20000 × 80	(3.700, 0.00010865)	(3.700, 0.00154049)

Table 5: Comparison of RMSE and L2 norm with LTFF method in the solution.

Mesh size ($N \times M$)	LTFF (Runtime, RMSE)	 (Runtime, L2 norm)
1250 × 5	(0.057, 0.02206451)	(0.057, 0.31125825)
2500 × 10	(0.098, 0.00551475)	(0.098, 0.07799352)
5000 × 20	(0.289, 0.00142755)	(0.289, 0.02018861)
10000 × 40	(1.037, 0.00040779)	(1.037, 0.00576715)
20000 × 80	(3.969, 0.00009299)	(3.969, 0.00131512)

Using $\widehat{u}(x,t) = \exp(w(x,t)) - 1$, (36), and (37), we obtain the following:

$$\begin{aligned} w(1,t) &= 0, \\ w_y(1,t) &= -S(t)S'(t), \\ w_{yy}(1,t) &= S(t)^2 S'(t)^2. \end{aligned} \tag{39}$$

We lower the error by using the optimized safety parameter ρ for a Taylor series with fixed free boundary, a method suggested by Kim et al. [16], in order to control the error. We use Taylor series so that we can manage the error by employing the safety parameter ρ:

$$\begin{aligned} w(y,t) &= w(1,t) + \frac{w_y(1,t)}{1!}(y-1) \\ &\quad + \frac{w_{yy}(1,t)}{2!}(y-1)^2 + O\left((y-1)^3\right), \\ w(\widetilde{y},t) &= w(1,t) + \frac{w_y(1,t)}{1!}(\widetilde{y}-1) + \frac{w_{yy}(1,t)}{2!}(\widetilde{y}-1)^2. \end{aligned} \tag{40}$$

Using $\widetilde{y} = 1 - \rho\Delta y$, (39), and (40), we get

$$\frac{1}{2}S(t)^2(-\rho\Delta y)^2 S'(t)^2 + S(t)\rho\Delta y S'(t) - \widetilde{w} = 0. \tag{41}$$

Combining (41) with $u_x(S(t),t) < 0$, $S'(t) > 0$, we have

$$S'(t) = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \tag{42}$$

where

$$\begin{aligned} a &= \frac{1}{2}S(t)^2(-\rho\Delta y)^2, \\ b &= S(t)\rho\Delta y, \\ c &= -\widetilde{w}. \end{aligned} \tag{43}$$

For discretization $(\Delta y, \Delta t)$, we introduce a two-dimensional mesh in the first quadrant of the $y - t$ plane. From (42) we have

$$S_{m+1} = S_m + \frac{-\widetilde{b} + \sqrt{\widetilde{b}^2 - 4\widetilde{a}\,\widetilde{c}}}{2\widetilde{a}}\Delta t, \tag{44}$$

Table 6: Comparison of VSG, BIM, LTUF, LTFF method, and exact solution when $f(t) = \exp(t+1)$.

Position x	Numerical solution				Exact solution
	VSG	BIM	LTUF	LTFF	
0.00	6.389056	6.389056	6.389056	6.389056	6.389056
0.10	5.685461	5.685448	5.685467	5.685440	5.685894
0.20	5.049275	5.049261	5.049281	5.049254	5.049647
0.30	4.473626	4.473613	4.473632	4.473605	4.473947
0.40	3.952754	3.952741	3.952761	3.952734	3.953032
0.50	3.481448	3.481434	3.481454	3.481428	3.481689
0.60	3.054989	3.054975	3.054997	3.054978	3.055199
0.70	2.669112	2.669098	2.669120	2.669094	2.669296
0.80	2.319954	2.319939	2.319962	2.319937	2.320117
0.90	2.004020	2.004006	2.004030	2.004006	2.004116
1.00	1.718151	1.718137	1.718162	1.718139	1.718281
1.10	1.459484	1.459470	1.459498	1.459474	1.459603
1.20	1.225432	1.225418	1.225446	1.225425	1.225540
1.30	1.013652	1.013638	1.013667	1.013648	1.013752
1.40	0.822024	0.822010	0.822042	0.822024	0.822119
1.50	0.648631	0.648618	0.648651	0.648635	0.648721
1.60	0.491738	0.491725	0.491760	0.491746	0.491824
1.70	0.349775	0.349762	0.349799	0.349787	0.349859
1.80	0.221320	0.221308	0.221347	0.221338	0.221403
1.90	0.105089	0.105077	0.105119	0.105112	0.105171
2.00	0.000000	0.000000	0.000000	0.000000	0.000000
RMSE	0.000109	0.000107	0.000108	0.000092	
L2 norm	0.001552	0.001516	0.001540	0.001315	
Runtime (secs)	3.640	4.122	3.700	3.969	

where

$$\begin{gathered}\widetilde{a} = \frac{1}{2} S_m^2 (-\rho \Delta y)^2, \\ \widetilde{b} = S_m \rho \Delta y, \\ \widetilde{c} = -w(1 - \rho \Delta y, m \Delta t), \\ m = 0, 1, 2, \ldots, M-1.\end{gathered} \tag{45}$$

When the initial values are given by $w(1 - \rho\Delta y, t_0)$ (transformed value of the temperature) and S_0 (free boundary) at $t = 0$, we can determine S_1 (free boundary) at Δt using (44) and find w from (38). More importantly, for updating the free boundary, our method does not include any iteration until the numerical solution is obtained. So, we repeat the previously mentioned process until $M\Delta t$ and obtain the free boundary in a time-recursive way.

4. Numerical Result

In this section, we provide two numerical examples to illustrate our method. We also provide runtimes and computation errors used to compare with the results obtained by other numerical methods such as the VSG method developed by Kutluay et al. [14] and the BIM method developed by Landau [13]. In this section, we compare the four methods based on runtime, L2 norm, and root mean square error (RMSE). All calculations are carried out using a C++ implementation with the a 2.66 Ghz Intel 4 Core CPU with 4 GB RAM. A finite difference method with an explicit finite difference approximation is proposed for our methods. The benchmark results are obtained using the exact solution.

The parameter values used to calculate the free boundary and the values of temperature are $T = 1.0$, $f(t) = \exp(t + 1)$, $S(0) = 1$, and a discrete mesh of 10000×40 nodes.

In Figure 2, we find a numerical optimization $\rho = 0.00706$. Table 1 also shows the results of the free boundary and the values of temperature with the safety parameters. One can see from Table 1 that the free boundary monotonically increases as the value of ρ increases, but the values of temperature monotonically decrease when ρ increases. They are so gradual that they are not very susceptible to change in the value of ρ.

In Figure 3, Table 2, Figure 4, and Table 3, we take the parameter values used in Figure 2 except for the discrete mesh, plot runtimes, and computational errors compared with the various methods.

Figure 5 and Table 4 show the RMSE and L2 norm error of the LTUF method and compare it with sufficiently inclined slope.

Figure 6 and Table 5 show the RMSE and L2 norm error of the LTFF method and compare it with sufficiently inclined

Table 7: Comparison of VSG, BIM, LTUF, LTFF method, and exact solution when $f(t) = 1$.

position x	Numerical solution				Exact solution
	VSG	BIM	LTUF	LTFF	
0.00	1.128379	1.128379	1.128379	1.128379	1.128379
0.10	1.030558	1.030846	1.031035	1.031171	1.031198
0.20	0.939323	0.939562	0.939461	0.938557	0.939644
0.30	0.853239	0.853118	0.852186	0.853340	0.853672
0.40	0.772252	0.772710	0.773105	0.772877	0.773215
0.50	0.696855	0.697913	0.698005	0.697264	0.698177
0.60	0.627581	0.628210	0.627142	0.628511	0.628436
0.70	0.563013	0.562986	0.563804	0.562138	0.563851
0.80	0.503018	0.503362	0.504110	0.504430	0.504255
0.90	0.447454	0.449016	0.448385	0.448748	0.448465
1.00	0.397071	0.398605	0.394381	0.399291	0.399282
1.10	0.351200	0.352020	0.353327	0.353470	0.353492
1.20	0.309050	0.310587	0.310942	0.311511	0.311870
1.30	0.270431	0.272901	0.274082	0.274519	0.274184
1.40	0.235147	0.238341	0.239877	0.239340	0.240196
1.50	0.203922	0.206799	0.208701	0.210078	0.209664
1.60	0.175475	0.179454	0.181921	0.182140	0.182347
1.70	0.149508	0.154482	0.157249	0.158290	0.158005
1.80	0.125812	0.131752	0.135028	0.136610	0.136403
1.90	0.104323	0.111398	0.116116	0.117326	0.117311
2.00	0.085031	0.093533	0.098785	0.100843	0.100509
RMSE	0.0055537	0.0024662	0.0007687	0.0006550	
L2 norm	0.0789335	0.0350518	0.0109265	0.0093102	
Runtime (secs)	0.795	0.860	0.783	0.821	

slope. Especially, Figures 5 and 6 and Tables 4 and 5 show the numerical convergence of our methods. Note that the discrete meshes of $1250 \times 5, 2500 \times 10, 5000 \times 20, 10000 \times 40$, and 20000×80 nodes are plotted in Figures 3, 4, 5, 6, 7, 8, and 9, and Tables 2, 3, 4, 5, 6, and 7.

For the sake of clarity, the pairs of (RMSE, L2 norm) of all four methods are graphed in Figure 7. Figure 7 illustrates the pairs in the solution. The figure shows that, in terms of errors (RMSE, L2 norm), LTUF approximates more accurately than VSG does, and LTFF approximates more accurately than BIM does. In Figures 8 and 9, we investigate the functional behaviors of the difference between RMSE (VSG) and RMSE (LTUF), L2 norm (VSG) and L2 norm (LTUF), RMSE (BIM) and RMSE (LTFF), and L2 norm (BIM) and L2 norm (LTFF) with respect to final time change.

Table 6 reports the values of the temperatures for the specific parameter set associated with the discrete mesh of 20000×80.

In Figure 10 and Table 7, we investigate the free boundary behavior with respect to time change when $T = 1.0$, $f(t) = 1$, and $S(0) = 0$. From (1)~(5) this problem has the following exact solution $u(x,t)$:

$$u(x,t) = 2\sqrt{\frac{t}{\pi}} \exp\left(-\frac{x^2}{4t}\right) - x \operatorname{erfc}\left(\frac{x}{2\sqrt{t}}\right). \qquad (46)$$

5. Final Remarks

We suggested LTUF and LTFF methods. There are approximated solutions that show reasonable agreement.

VSG method suggested by Kutluay et al. [14] shows that accumulating error occurred during calculation by the three-term backward difference near the free boundary, while LTUF method adopts a log-transform function to reduce the accumulating error in solution surface that causes gradual slope. Our method employing a log-transform function with the fixed free boundary solves a nonlinear problem on a fixed domain derived from a free boundary problem. Since the computation process depends on Lipschitz surface, we need to focus on the motion of the solution surface which would be simple to see the minute behavior of solution surface.

Kutluay et al. [14] use three-term backward difference scheme during calculating the free boundary with BIM method suggested by Landau [13]. However, we emphasize that there is no three-term backward difference scheme in LTFF method. Instead, we solve the quadratic equation and directly find the free boundary. In an environment with the rapidly changing free boundary, our straight forward method is a very powerful tool to understand Stefan problem.

Numerical study also shows that overall speed and accuracy comparisons have demonstrated some superiority of our methods over other methods. Further investigations on the extending of the various models could be interesting and

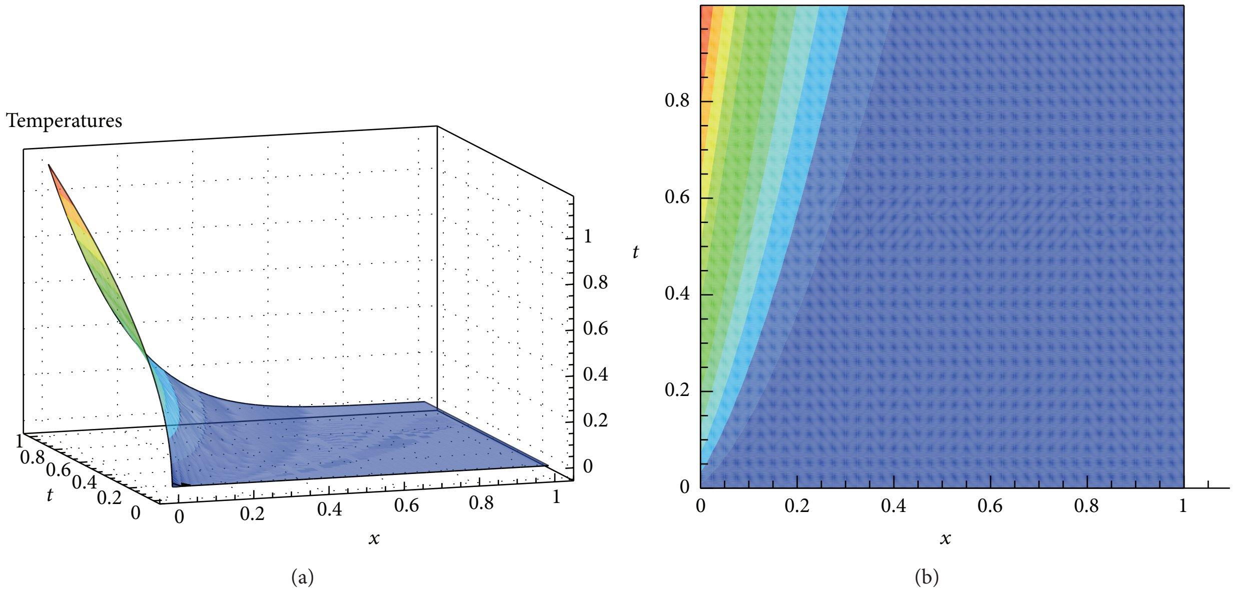


Figure 10: Numerical solution and contours when $f(t) = 1$.

useful. Also, more comparative and extensive studies on our method should be helpful for completing the mission of the present work. This remains as our future research work.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Stefan, "Uber die theorie der eisbildung inbesondee uber die eisbindung im polarmeere," *Annalen der Physik und Chemie*, vol. 42, pp. 269–286, 1891.

[2] D. Blanchard and A. Porretta, "Stefan problems with nonlinear diffusion and convection," *Journal of Differential Equations*, vol. 210, no. 2, pp. 383–428, 2005.

[3] T.-F. Cheng, "Numerical analysis of nonlinear multiphase Stefan problems," *Computers and Structures*, vol. 75, no. 2, pp. 225–233, 2000.

[4] D. Juric and G. Tryggvason, "A front-tracking method for dendritic solidification," *Journal of Computational Physics*, vol. 123, no. 1, pp. 127–148, 1996.

[5] M. Hinze and S. Ziegenbalg, "Optimal control of the free boundary in a two-phase Stefan problem," *Journal of Computational Physics*, vol. 223, no. 2, pp. 657–684, 2007.

[6] J. Liu and M. Xu, "Some exact solutions to Stefan problems with fractional differential equations," *Journal of Mathematical Analysis and Applications*, vol. 351, no. 2, pp. 536–542, 2009.

[7] V. R. Voller, J. B. Swenson, and C. Paola, "An analytical solution for a Stefan problem with variable latent heat," *International Journal of Heat and Mass Transfer*, vol. 47, no. 24, pp. 5387–5390, 2004.

[8] F. Yigit, "Approximate analytical and numerical solutions for a two-dimensional Stefan problem," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 857–869, 2008.

[9] J. Lorenzo-Trueba and V. R. Voller, "Analytical and numerical solution of a generalized Stefan problem exhibiting two moving boundaries with application to ocean delta formation," *Journal of Mathematical Analysis and Applications*, vol. 366, no. 2, pp. 538–549, 2010.

[10] A. Asaithambi, "Numerical solution of Stefan problems using automatic differentiation," *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 943–948, 2007.

[11] J. Nedoma, "Numerical solution of a Stefan-like problem in Bingham rheology," *Mathematics and Computers in Simulation*, vol. 61, no. 3–6, pp. 271–281, 2003.

[12] D. Słota, "Using genetic algorithms for the determination of an heat transfer coefficient in three-phase inverse Stefan problem," *International Communications in Heat and Mass Transfer*, vol. 35, no. 2, pp. 149–156, 2008.

[13] H. G. Landau, "Heat conduction in a melting solid," *Quarterly Journal of Applied Mathematics*, vol. 8, pp. 81–94, 1950.

[14] S. Kutluay, A. R. Bahadir, and A. Özdeş, "The numerical solution of one-phase classical Stefan problem," *Journal of Computational and Applied Mathematics*, vol. 81, no. 1, pp. 135–144, 1997.

[15] B. J. Kim, C. Ahn, and H. J. Choe, "Direct computation for American put option and free boundary using finite difference method," *Japan Journal of Industrial and Applied Mathematics*, vol. 30, no. 1, pp. 21–37, 2013.

[16] B. J. Kim, Y.-K. Ma, and H. J. Choe, "A simple numerical method for pricing an American put option," *Journal of Applied Mathematics*, vol. 2013, Article ID 128025, 7 pages, 2013.