1-1-2003

A Fixed-Grid Front-Tracking Algorithm for Solidification Problems. Part I - Method and Validation

C. Y. Li

S V. Garimella
Purdue University, sureshg@purdue.edu

J E. Simpson

Li, C. Y.; Garimella, S V.; and Simpson, J E., "A Fixed-Grid Front-Tracking Algorithm for Solidification Problems. Part I - Method and Validation" (2003). *CTRC Research Publications.* Paper 79.
http://dx.doi.org/10.1080/10407790390121961

A Fixed-Grid Front-Tracking Algorithm for Solidification Problems. Part I – Method and Validation[§]

Chin-Yuan Li[¶], **Suresh V. Garimella**[†] and **James E. Simpson**[‡]
School of Mechanical Engineering
Purdue University
West Lafayette, Indiana 47907-1288

Phone: (765) 494-5621 Fax: (765) 494-0539
sureshg@ecn.purdue.edu

ABSTRACT

This paper presents a method for solving moving boundary problems associated with phase-change. This method is an explicit interface-tracking scheme that involves the reconstruction and advection of the moving interface on a fixed grid. Three distinct steps are undertaken to handle the movement of the interface: advection and reconstruction (tracking); calculation of normal velocities; and the solution of the governing equations for different phases. Details of each step and its implementation is provided. The transient heat diffusion equation in two space dimensions is the governing equation for energy transport. In order to validate the approach, results obtained from the front tracking scheme are compared with exact analytical solutions for melting problems in Cartesian and cylindrical coordinates. It is shown that the numerical and analytical results are in excellent agreement. Finally, problems involving the multi-dimensional solidification of a pure aluminum ingot subject to bi-directional heat extraction are presented. The results indicate that the algorithm was able to accurately track the front under these aggressive conditions which caused large curvature. In Part II, the front-tracking approach described herein is applied to directional solidification problems influenced by melt convection.

§ Submitted for publication in *Numerical Heat Transfer*, August 2002
¶ Research Assistant
† Professor, ***person to whom correspondence should be addressed***
‡ Postdoctoral Research Associate

INTRODUCTION

Moving liquid-vapor or solid-liquid interfaces encountered in phase change and materials processing problems may in general be highly distorted. The shape and position of such an interface is not known *a priori*, but instead must be obtained as part of the problem solution. The problem in such cases consists not only of a careful solution of the governing equations in the two phases but also of an accurate tracking of the interface in response to the thermal, compositional, or flow fields computed at each time step.

The interface between phases has been variously treated in the literature with interface-capturing and interface-tracking methods [1]. In the interface-capturing category, an attempt is made to resolve the details of the structure of the interface. The interface is modeled only to the extent that the actual physical discontinuity is known to be someplace near the middle of the (temperature) gradient; no specific modeling is needed for the interface, except perhaps for finer meshing in its vicinity. The enthalpy method is an example of this one-domain method. The different phases are considered together in the solution, and the properties are either changed discretely at the interface or continuously over a range near the interface. The interface itself is generally not tracked.

Methods of interface tracking, as distinct from capturing, may be divided into front-tracking and volume-tracking approaches. Interface-tracking studies in the literature were reviewed by Chen et al. [2] with reference to the simulation of bubble rise and distortion. In front tracking [3] the interface is identified by an ordered set of marker points located on the interface, and is represented by the distance between the points and some reference surface. A line connecting the marker points, usually a piecewise polynomial, represents the front. An irregular grid is used in the vicinity of the interface. In an extension of this method Unverdi and

Tryggvason [4] represented the interface by an indicator function, which takes a value of unity in one phase and zero in the other. An additional Eulerian grid is generated on the front, which explicitly tracks the interface and moves through the stationary grid. More recent efforts by Udaykumar et al. [5] to improve the accuracy of interface tracking may also be considered to fall under the front-tracking category. They proposed a method in two dimensions to track the position of the interface explicitly by a Lagrangian translation of marker particles, while the field equations are solved on an underlying fixed grid as in Eulerian methods.

An increasing number of approaches for treating interfaces have become available, such as Lagrangian approaches including boundary-fitted grids [3, 6] and boundary-integral methods [7], Eulerian approaches including volume-of-fluid [8], level-set [9], phase-field [10], and enthalpy [11, 12] methods, and Mixed approaches [4, 5]. Shyy et al. [13] provide a good review of interface tracking methods.

Most of the interface-tracking studies in the literature are implemented in two dimensions. The extension of these methods to three dimensions introduces a number of serious complications. Some studies [2, 14, 15] have successfully adapted the front-tracking approaches discussed above to the modeling of three-dimensional bubble dynamics. However, the solution methodologies are quite complex and computationally intensive. More importantly, in both the two- and three-dimensional implementations, the interface has not been exactly tracked. In order to properly represent the relevant physics at the interface in the simulations, the interface position must be accurately known. The accuracy of the tracking is particularly important in properly accounting for surface-tension effects.

A sharp resolution of the interface may be accomplished by an adaptive grid approach, where the grid is adjusted locally to coincide with the interface. However, it is difficult to

adjust the grids to track highly deformed interfaces. Another approach that has been successful in maintaining a sharp resolution of the interface is the use of body-fitted coordinates (see for instance [16]). This mapping method transforms the deformed physical domain into a regular, fixed-grid computational domain. In theory, this method provides a true tracking of the interface. However, in practice, the switching between the original and transformed coordinate systems required at each time step can make the computations in this approach extremely expensive. Moreover, only geometries that do not lead to singular mappings can be handled by this method. Jayaraman et al. [17] presented a promising method for tracking in three dimensions using adaptive unstructured grids to represent the three-dimensional interfaces.

The new approach for tracking the interface in Labonia et. al. [18] is motivated by the fact that a fixed-grid approach lends itself to simplified computations, in addition to being able to handle large deformations and multiple interfaces. The main disadvantage in the use of fixed grids has been the difficulty in precisely tracking the interface [1]. In their work, Labonia et al. presented a scheme for explicitly tracking the interface on a fixed grid in three dimensions that eliminated this disadvantage.

When applied to problems such as the solidification of liquid melts, interface tracking requires the solution of a system of equations for the conservation of mass, momentum, energy, and perhaps, species. The movement of the interface in such a problem is governed by an energy balance, and by additional conditions if an alloy is considered. For the present study, the governing equation is the energy equation, with conduction as the only transport mechanism. In order to demonstrate that the algorithm is capable of solving solidification problems, a series of benchmark two-dimensional test problems are investigated. Pure aluminum is used as the phase-change material. In Part II of this work [19], the method developed in the present study is

applied to solve directional solidification problems with melt convection.

NUMERICAL METHOD

In solving a solid-liquid phase-change problem, at each time step during the solidification or melting process, the governing conservation equations must be solved, and it is then necessary to compute the movement of the interface. There are three distinct steps involved: *interface reconstruction and advection* (tracking); *calculation of normal velocities*; and *solution of the governing equations*. The second and third steps involve the imposition of boundary conditions and matching conditions at the interface, and hence follow and depend upon the first step. In this section, we describe methods used for performing each of these tasks.

1. INTERFACE TRACKING

The front-tracking scheme described in the present work is an adaptation of that presented by Labonia et al. [18]. Previously, the method was used to calculate the propagation of a front in an artificial problem wherein the temperature field did not change with time [18]. In order to analyze actual solidification problems, several refinements to the scheme were required as described in this section. Additional detail, particularly regarding the identification of *intersection points*, can be found in [18].

The actual process of interface tracking is itself accomplished in three steps: identification of marker points; calculation of surface normals; and identification of intersection points. A procedure for the reconstruction of the interface is also described. The reconstructed interface, however, is not directly used in the interface tracking, and its utility is primarily graphical in nature. The concepts involved are first described with respect to

unidirectional phase change, and subsequently, the additional considerations involved in dealing with multidirectional problems are discussed.

***Identification of Marker Points*:**

Our first task is to find the points where the interface intersects the Cartesian grid lines. We do this in three parts. First, we identify the points where the interface cuts the vertical lines in the grid (see Fig. 1a), i.e., intersection of the lines ($x = x_i$, $y = y_j$, z) with the interface. We term these points of intersection as *marker points*, since these are the points used in advecting the interface according to the imposed boundary conditions. It is to be noted that there will always be points of intersection between the vertical grid lines and the interface. Next, the points of intersection of the interface with the other two sets of (horizontal) grid lines, (x_i, y, z_k) and (x, y_j, z_k), will simply be referred to as *intersection points*. It is recognized that unlike the marker points, there may or may not be intersection points between the interface and the horizontal grid lines. These intersection points are required in order to apply the matching conditions at the interface.

Once the interface is advected over time step Δt, new marker points (and intersection points) corresponding to the new interface location need to be determined. The procedure for determining the new marker points at time step ($t + \Delta t$) is represented in Fig. 1(a) in two dimensions. The marker points (denoted by a solid square symbol in Fig. 1a) at time step t are shown. For the next time step ($t + \Delta t$), these are advected in the direction of the interface normal at the normal velocity as determined by the interface condition. The resulting new locations of these points, known as *advected points* (denoted by an open circle in Fig. 1a), would in general no longer lie on the vertical grid lines. New marker points at this time step ($t + \Delta t$)

need to be obtained from these advected points as shown in Fig. 1(a). For the two-dimensional cases highlighted in this study, this is accomplished simply by fitting cubic-spline segments to each set of advected points; the intersection of the cubic-spline fits with the vertical grid lines yields the new marker points. For this cubic-spline fitting process, the front is represented as segments of the following polynomial:

$$z = ax^3 + bx^2 + cx + d \tag{1}$$

The first and second derivatives calculated from both shared segments on each point should be equal. Location information from all the advected points (x_i, y_j) and the derivatives at the two end points is used to find a set of first derivatives at each advected point. Next, the new marker points are found at the locations where the line segments between two adjacent advected points intersect the vertical gridlines, as indicated in Fig. 1(a). In the more general three-dimensional case, bilinear interpolation between the advected points is used. A local coordinate system at each surface element defined by the four advected points A, B, C and D (as shown in Fig. 2a) is used to perform this interpolation in transformed space. This is much more accurate than simply interpolating using Cartesian coordinates, since the quadrilateral interface segment in question (as shown in Fig. 2a) could in general be quite distorted. The transformation proposed here ensures that the bilinear interpolation is carried out on a local rectangular grid, thus improving the accuracy of determination of the marker points

Special handling is required for the *boundary points*. These are marker points that lie on the boundaries of the domain (calculation of the normal velocities for these points is addressed in the next subsection). Should the interface merge with the top wall, the calculated location for the boundary point falls outside the computational domain (Fig. 1a). The location of the boundary point is then corrected by finding the intersection between a line joining the projected

boundary point location and the first interior advected point, and the top boundary. The new, corrected position for the boundary point (solid circle) then lies along the top boundary (again this is shown in Fig. 1a).

***Calculation of Surface Normals*:**

Surface normals are calculated from cubic spline fits [20] to the marker points in the two orthogonal directions, that is, in the *x*-*z* and *y*-*z* planes. At the boundaries, a cubic spline is used at the faces of the domain (or edges for the 2D case). The cubic spline requires only two points and two second derivatives, and hence, yields a unique solution for the intersection point. Also, the choice of a cubic-spline fit to the marker points on the domain face is superior to simple straight-line segments for the following reason. If the advected point does not lie exactly on a grid line (and thus, does not coincide with the new marker point), a straight-line fit between advected points would yield a unique normal at the marker point. However, if the advected point does fall on a grid line, only a cubic spline may be used to get a unique normal due to the continuity of its first derivative. A cubic-spline fit also better represents the curvature of the interface.

***Identification of Intersection Points*:**

At this stage, the marker points and the interface normals are all available. The intersection points of the interface with the other, horizontal coordinate lines (these are *not* marker points) need to be found next, by the procedure described in [18]. A knowledge of the intersection points is necessary for the solution of the governing equations (which is discussed in

Section 3). It is emphasized that the intersection points are obtained in this study using the information about the marker points, and not from the advected points.

At this stage, given a normal velocity, the marker points representing the interface can be advected to the next time step, and the entire reconstruction procedure repeated. However, the normal velocities are not known *a priori* in most problems of practical interest. Instead, they need to be calculated using the physically imposed interface boundary conditions as detailed in the next important stage of the calculation, discussed in Section 2.

***Multidirectional Phase-Change*:**

For clarity, the preceding explanation is for the simplest case of a *single* set of marker points. Using a single set of marker points is recommended for situations where the interface does not take on an acute angle with respect to the axis along which the marker point locations are measured, i.e. its slope remains small. In general, however, the interface is more arbitrarily oriented and may take on any shape.

For such cases, *multiple* sets of marker points may be used, for increased accuracy. Details of this approach are as follows.

1. *Marker Points*. The procedure used to recover two sets of marker points for the 2D case is shown in Fig. 1(b). In regions where the absolute value of the slope of the interface is greater than 1 (i.e., the interface is more vertical than horizontal), *horizontal marker points* are recovered. In regions where the absolute value of the slope is less than 1, *vertical marker points*, based on locations where the interface intersects vertical grid lines are found. Both sets of marker points are advected and reconstructed in an identical manner to that described previously.

2. *Intersection Points*. With the possible existence of two sets of marker points representing the interface, there are also two different types of intesection points in multi-directional phase change problems. All of the points (solid circles in Fig. 1b) where the interface intersects with grid lines are intersection points if they are not included in the set of marker points; for example, point A (*horizontal* intersection point where the absolute value of the slope is less than 1) and point B (*vertical* intersection point where the absolute value of the slope is greater than 1) in Fig. 1(b). Information on the intersection points is used to satisfy the boundary conditions on the interface.

2. CALCULATION OF NORMAL VELOCITIES

In order to apply the boundary conditions appropriately at the interface, it is necessary to determine the exact location (marker points) of the interface within each control volume that houses a part of the interface. In fact, it is this desire to apply the physical interface conditions exactly at the tracked interface which led to the formulation of the present approach over existing interface-capturing methods and volume-of-fluid (VOF) methods.

In the solidification example we are considering, the normal velocity V_n of the interface is obtained from the difference between the normal gradients of temperature in the liquid and solid as latent heat is liberated via the following equation (Stefan condition):

$$k_S \frac{\partial T_S}{\partial n} - k_L \frac{\partial T_L}{\partial n} = \Delta H V_n \tag{2}$$

We apply this condition in transformed space (ξ, η, ζ) centered at the marker point P of interest as illustrated in Fig. 2(b). Following the definition of a gradient in general curvilinear coordinates, the discrete approximations for the temperature derivatives are:

$$\frac{\partial T}{\partial \xi} = \frac{1}{2}(T_E - T_W)$$
$$\frac{\partial T}{\partial \eta} = \frac{1}{2}(T_N - T_S) \tag{3}$$
$$\frac{\partial T}{\partial \zeta} = (-3T_P + 4T_{P_{L1}} - T_{P_{L2}}) \quad or \quad (3T_P - 4T_{P_{S1}} + T_{P_{S2}})$$

The temperatures at the interface points E, W, N, S, and P are specified by the boundary condition on the interface ($T = T_m$). The normal derivative at the interface then is

$$\frac{\partial T}{\partial n} = \left(\frac{\partial T}{\partial \xi} g^1 \cdot n + \frac{\partial T}{\partial \eta} g^2 \cdot n + \frac{\partial T}{\partial \zeta} g^3 \cdot n \right) \tag{4}$$

in which g^k are the contravariants found from the coordinate transformation [18]. Once the temperature gradients in the liquid and solid are obtained, the normal velocity can be evaluated simply with an equation of the form (depending on the exact nature of the interface boundary condition) of Eq. (2).

An additional constraint needs to be taken into account at the physical boundaries of the domain. Since the marker points at the boundaries are constrained to advect along the boundaries, only two of the three components of the velocity are needed. On a vertical side wall (*0, y, z*), for example, the interface cannot move in the *x* direction. Therefore, due to the equal and opposite reaction of the wall, the *x* component of V_n must be ignored, and the marker point on the interface is advected only in the *y* and *z* directions. This procedure, adopted due to the constraint posed by the walls, enables the movement of the interface at the wall to be correctly represented.

3. SOLUTION OF THE GOVERNING EQUATIONS

In order to solve the governing equations in the two phases, a methodology needs to be

devised for applying the equations for each of the phases in the interfacial cells. The interfacial cell is partitioned according to the actual position of the interface in the cell, and the equations are solved separately in each phase. Although the front tracking method is developed for three-dimensional work, we will only consider one- and two-dimensional benchmark problems in the present study, and so the solution scheme is only presented in two space dimensions. More general three-dimensional problems will be considered in future work.

With the interface location defined by using a set of marker points, we partition the domain into solid and liquid regions. The governing equations for conservation of energy, assuming negligible convection can be written as

$$\frac{\partial T}{\partial t} = \alpha_i \cdot \nabla^2 T \tag{5}$$

in which subscript *i* refers to either the liquid or solid phase. The finite difference analog of Eq (5) is

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha_i \cdot \left[\frac{T_{i+1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i-1,j}^{n+1}}{2\Delta x} + \frac{T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{2\Delta y} \right] \tag{6}$$

in which Forward Time Centered Space (FTCS) discretization is used. The discrete approximation to the derivatives gives a five-point stencil in 2D; the algebraic form of Eq. (6) is:

$$A_P T_{i,j}^{n+1} + A_E T_{i+1,j}^{n+1} + A_W T_{i-1,j}^{n+1} + A_N T_{i,j+1}^{n+1} + A_S T_{i,j-1}^{n+1} = A_P^0 T_{i,j}^n \tag{7}$$

where $A_P = A_P^0 + A_E + A_W + A_N + A_S$

The truncation error due to the discrete operators is expected to be of second order in space and first order in time for the bulk of the domain. The finite-difference stencil used to calculate the gradients at the interface may not be second-order accurate. Since these points are few in number, the deterioration in global accuracy below $O(\Delta x^2)$ is expected to be minimal

[21].

The discretization scheme employed at the interface is highlighted in Fig. 3. Point y_j lies in the solid phase, while y_{j+1} lies in the liquid phase. The two are separated at y_m by the interface. If we are interested in discretizing the diffusion term at y_{j+1}, i.e. in the liquid phase, such that the truncation error is $O(\Delta x)$, the following procedure is employed (with a similar result for the solid phase):

$$\left(\frac{\partial^2 T}{\partial y^2}\right)_{j+1} = \frac{(\frac{\partial T}{\partial y})_n - (\frac{\partial T}{\partial y})_s}{y_n - y_s} \tag{8}$$

where

$$(\frac{\partial T}{\partial y})_n = \frac{T_{j+2} - T_{j+1}}{y_{j+2} - y_{j+1}} \tag{9}$$

$$(\frac{\partial T}{\partial y})_s = \frac{T_{j+1} - T_m}{y_{j+1} - y_m} \tag{10}$$

and T_m is the value of the variable at the interface location, and

$$y_n - y_s = \frac{y_{j+2} + y_{j+1}}{2} - \frac{y_{j+1} + y_m}{2} \tag{11}$$

Therefore

$$\left(\frac{\partial^2 T}{\partial y^2}\right)_{j+1} = \frac{2}{y_{j+2} - y_m}\left(\frac{T_{j+2} - T_{j+1}}{y_{j+2} - y_{j+1}} - \frac{T_{j+1} - T_m}{y_{j+1} - y_m}\right) \tag{12}$$

It can be shown that the truncation error for the above expression is $O(\Delta x)$.

If the grid points are too close to the interface (taken to be a distance of $0.1\Delta x$ in this study), we use another approach to calculate the temperature at such points. For example, if y_j is too close to the interface as shown in Fig. 3, we discretize the diffusion term at y_{j-1} in terms of

y_j using considerations similar to those shown above. Once the temperatures of y_{j-1} and y_{j-2} have been found, a second-order polynomial fit of the form

$$T(y) = a_t y^2 + b_t y + c_t \tag{13}$$

is used to calculate the temperature at y_j.

After constructing the discretization equations, we cast them into linear form but do not assume that a particular method would be used for their solution. Therefore, any suitable solution method can be employed at this stage. In this work, we choose Gauss-Seidel iteration with successive over-relaxation:

$$T_{i,j}^{P+1} = T_{i,j}^{P} + \omega\{[-A_E T_{i+1,j}^{P} - A_W T_{i-1,j}^{P+1} - A_N T_{i,j+1}^{P} - A_S T_{i,j-1}^{P+1} + A_P^0 T_{i,j}^{n}]/A_P - T_{i,j}^{P}\} \tag{14}$$

The accuracy of the calculated temperature field has a strong influence on the accuracy of the predicted front locations. In order to obtain highly accurate front locations, the tolerance on the convergence criteria used for the present study is 10^{-6}.

SUMMARY OF SOLUTION SCHEME

Three distinct processes are involved in the solution scheme: interface tracking, calculation of normal velocities and the solution of governing equations over the entire domain. A summary of the steps involved is as follows:

1. Read inputs: grid space, step size, thermal properties, and initial condition data.
2. Define *marker points* and *intersection points* and calculate the normal vector.
3. Calculate the normal velocities at each marker point.
4. Move the marker points; the new location of these points need not lie on the grid. These points are then denoted *advected points.*
5. Obtain new marker points from the advected points, and calculate the surface normals.

6. Solve the energy equation in both phases.

7. Go to step 3, and repeat the process for the next time step.

RESULTS

1D Melting Problem

In order to evaluate the ability of the algorithm to account for phase-change, a simple moving boundary problem was solved first. A moving boundary problem that has an exact solution is the melting of a flat interface, the so-called Neumann problem [22]. The domain and boundary conditions are shown in Fig. 4. This problem is often used to validate numerical methods for phase-change (for example, see Refs. [13, 21]).

The energy equation (5) for this 1D case may be rewritten in non-dimensional form as

$$\frac{\partial \phi}{\partial t'} = \frac{\partial^2 \phi}{\partial y'^2} \tag{15}$$

in which the following non-dimensional quantities are used

$$\phi = \frac{T - T_m}{T_W - T_m}, \quad y' = \frac{y}{y_L}, \quad t' = \frac{t}{y_L^2 / \alpha} \tag{16}$$

and the thermal properties of the liquid and solid are assumed equal. The non-dimensional form of the Stefan condition (2) is

$$\frac{ds(t')}{dt} = St\left(\left.\frac{\partial \phi}{\partial y'}\right|_s - \left.\frac{\partial \phi}{\partial y'}\right|_l \right) \tag{17}$$

in which $s(t')$ is the (non-dimensional) front location and *St* is the *Stefan number*, an important non-dimensional parameter for this problem:

$$\mathrm{S}t = c_p (T_H - T_m) / \Delta H \tag{18}$$

Initially, the domain is all solid at the melting temperature ($\phi = 0$). The wall at $x = 0$ (Fig. 4) is maintained at a constant hot temperature of $\phi = St$, causing melting to proceed from the wall with the front progressing in the direction of increasing x. An exact solution may be derived for this problem [22, 23]. The temperature distribution is given by

$$\phi(y',t') = erf\left(\frac{y'}{2\sqrt{t'}}\right) \Big/ erf(\gamma) \tag{19}$$

and the location of the melting front is found from

$$s(t') = 2\gamma\sqrt{t'} \tag{20}$$

in which γ is obtained from the root of the transcendental equation:

$$\gamma \exp(\gamma^2)\, erf(\gamma) = \frac{St}{\sqrt{\pi}} \tag{21}$$

Simulations were performed with initial and boundary conditions for this benchmark problem set at Stefan numbers of 0.1, 2.85 and 5. The corresponding values for γ are 0.22, 0.9 and 1.06, respectively. Higher Stefan numbers correspond to higher wall superheat and result in more rapid melting. A total of 31 mesh points were used in the y-direction. One of the limitations of the front-tracking approach is that the algorithm requires a solid/liquid interface to be present in the domain at all times. To circumvent this limitation, the initial condition used for the simulation was that found from the exact solution after some small amount of time, t_0', has elapsed. At this time, a thin sliver of liquid has formed next to the heated wall, such that the interface is located at $s(t_0') = 0.325$. The corresponding temperature distribution in the solid is therefore given by:

$$\phi(y',t_0') = erf\left(\frac{y'}{2\sqrt{t_0'}}\right) \Big/ erf(\gamma), \qquad 0 < y' < s(t_0') \tag{22}$$

Results for the interface location at a Stefan number of 0.1 are shown in Fig. 5(a). The exact interface position found from Eq. (19) is also shown on this plot. The time step used for this case was $\Delta t = 1.0\times10^{-4}$. Results from the simulation are shown every 1000 time steps. As can be readily observed from Fig. 5(a), the agreement between the numerical and analytical results is excellent. The maximum error is less than 0.14%, while the error is less than 0.08% for the bulk of the results. It is important to note that as melting proceeds, the error does not increase, indicating that the algorithm is able to reliably and accurately resolve the evolution of the front position.

Results obtained at higher Stefan numbers of 2.85 and 5 are shown in Figs. 5(b) and (c), respectively. The time step used was decreased to $\Delta t = 1.0\times10^{-6}$ for the cases with $St = 2.85$ and 5 since the higher Stefan numbers cause more rapid solidification. The results exhibit similar good agreement as for $St = 0.1$.

Directional Solidification Problem

Next, a 2D melting problem in cylindrical coordinates was modeled, in order to test the ability of the algorithm to track a circular-shaped interface using two sets of marker points. The energy equation in cylindrical coordinates is

$$\frac{1}{\alpha}\frac{\partial \phi}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi}{\partial r}\right) \tag{23}$$

in which $\phi = T - T_m$. The corresponding boundary conditions in the center and the matching condition at the interface are:

$$\lim_{r\to 0}\left(2\pi r k_l \frac{\partial \phi}{\partial r}\right) = Q' \qquad\qquad r = 0 \tag{24}$$

$$k_s \frac{\partial \phi}{\partial r} - k_l \frac{\partial \phi}{\partial r} + \rho_l \cdot \Delta H \frac{dS(t)}{dt} = 0 \qquad r = S \tag{25}$$

The initial condition is:

$$\phi = 0 \qquad (T = T_m) \qquad t = 0 \tag{26}$$

The similarity variable selected to solve this problem is:

$$\xi = \frac{r^2}{4\alpha t} \tag{27}$$

A schematic of this melting problem is shown in Fig. 6. Initially, the domain is all solid at the melting temperature ($\phi = 0$). The center is maintained as a constant line heat source of Q', causing melting to proceed from the center with the front progressing in the direction of increasing *r*. The exact solution of this problem is available in Poulikakos [24]. The temperature distribution is given by using the exponential intergral function, $Ei(\xi) = \int_{\xi}^{\infty} \frac{e^{-\lambda}}{\lambda} d\lambda$, in the following form

$$\phi(\xi) = \frac{Q'}{4\pi k_l} \left[Ei(\xi_S) - Ei(\xi) \right] \tag{28}$$

and the location of the melting front at any given time ξ_S is not known. It is obtained as the root of the following transcendental equation:

$$\frac{Q'}{4\pi} e^{-\xi_S} - \alpha \rho_l \Delta H \cdot \xi_S = 0 \tag{29}$$

Then the location of the melting interface at a given time *s(t)* can be obtained by using the definition of similarity variables as follows:

$$s(t) = \sqrt{4\alpha t \cdot \xi_S} \tag{30}$$

Simulations were performed with the thermophysical properties of aluminum as in Table 1, for line heat source values of Q' = 500, 1000 and 2000 W/cm. The corresponding values for ξ_S are 0.1305, 0.1899 and 0.3301, respectively. Higher line heat source values correspond to higher temperatures at the center and result in more rapid melting. A total of 101 × 101 mesh points were used in the quarter domain (7.5 cm × 7.5 cm) of this problem. Since the algorithm requires a solid/liquid interface to be present in the domain at all times, the initial condition used for the simulation was that found from the exact solution after some small amount of solid ($s(t)$ = 1.45 cm) was already melted. The corresponding temperature distribution in the melt was obtained from Eq. (28).

Figure 7 shows computed results for the front location as well as the exact interface position found from Eq. (30) for the different line heat source strengths considered. As can be readily seen in Fig. 7(a), the agreement between the numerical and analytical results is good, with the maximum error being less than 0.8%. Results for faster melting at Q' = 1000 W/cm and Q' = 2000 W/cm are shown in Fig. 7(b) and 7(c), respectively. Under these conditions, agreement between computation and the exact solution remains good. The maximum error in these results is 1.3 and 3.1%. As for the 1D melting problem, the error does not increase with time.

Temperature traces in the melt region are shown in Fig. 8(a) for the medium heat flux case (Q' = 1000 W/cm). As for the front locations, the computed temperature profiles can be seen to agree well with the analytical result, although the discrepancy between the computed and exact results increases slightly with increasing time. Temperature contours for the entire domain from the simulated results are shown in Fig. 8(b). As can be seen, the contours are uniform and circular in shape as expected, indicating that a circular front, reconstructed using the

front-tracking scheme on a fixed, Cartesian grid, is not causing any undesirable spatial perturbations in the temperature field.

Finally, numerically simulated and analytical (exact) front locations are compared in Fig. 8(c) (for $Q' = 1000$ W/cm). The good agreement seen between the two sets of results in Fig. 7(b) is again evident here. It can be readily seen that the algorithm has captured the circular front shape on a fixed Cartesian grid with minimal error.

Ingot Casting Problem

The final test problem solved in this study is the classic problem of phase change in a box [27]. An initially superheated melt is subjected to a uniform and constant heat flux (200 W/cm^2 as a case study here) from all four walls. A schematic of this problem is shown in Fig. 9. Again, pure aluminum is taken to be the phase-change material. Planes of symmetry about the center of the domain are used to reduce the problem to one-quarter of the total domain size. This problem is a more demanding test of the algorithm than the previous cases since the expected solution is strongly two-dimensional. This has important implications for the front-tracking algorithm that has been developed, which, by the nature of the definitions of the marker points, will tend to resolve interfaces more accurately in one direction. Therefore, as for the melting in cylindrical coordinates problem considered above, the algorithm with two sets of marker points is applied for this problem, the large interface deflections of which severely test the algorithm.

Contour plots for the temperature fields at two different times at (t = 24 and 40 s) are shown as isotherms in Figs. 10(a) and 10(b) respectively. In these two figures, the bold line represents the front location, and coincides with the isotherm for the melting temperature. The

temperature field in the solid is seen to exhibit a linear variation with distance from the wall when the entire domain has nearly solidified. The front locations in the ingot casting problem were compared with the results obtained using the enthalpy method [28] under the same conditions. The comparison of these front locations every 4 seconds is shown in Fig. 10(c) with dashed lines representing the enthalpy method, and the front tracking method with one and two sets of marker points represented using dash dot and solid lines respectively. The results from these methods show very good agreement at the beginning, with the difference increasing slightly as the front propagates with time. With only one set of vertical marker points, the prediction of front locations in the left and top is less accurate than towards the right where the front is more horizontal.

CONCLUSIONS

A three-dimensional front tracking algorithm, previously developed by Labonia et al. [18] has been extended so that a solution scheme for the governing equations for temperature is included. The algorithm has also been modified to use cubic-spline reconstruction for determining the marker points; this has been found to be superior to the linear interpolation technique adopted in [18]. Additionally, improved accounting systems for handling the addition and deletion of marker points at the boundary have been added. These critical additions enable the algorithm to solve realistic problems; previously, interfaces were only tracked for (unrealistic) situations where the temperature field did not change with time.

The performance of this extended algorithm was then examined by considering a series of one- and two-dimensional test problems. First, a benchmark 1D melting problem was solved, and the results were compared to an analytical solution. The agreement between analytical and

numerical results was excellent, giving confidence in the ability of the algorithm to adequately model phase-change processes. Next, a benchmark 2D cylindrical melting problem was solved. The numerically predicted temperature profiles and front locations exhibited good agreement with the exact analytical solution. Hence, the front-tracking scheme was seen to be able to solve the difficult challenge of tracking a circular front on a Cartesian grid with minimal error.

Finally, the classic ingot solidification problem [27] was solved. The results from this test problem indicate that the algorithm is able to solve practical solidification problems with strongly multidimensional features.

The results shown in this paper demonstrate that this front-tracking scheme is accurate yet highly computationally efficient. The scheme is able to explicitly track the interface, yet is comparable in computational expense to less accurate front-capturing techniques. The fully transient simulations reported in this work required less than 20 minutes of CPU time on a Unix workstation.

In Part II of this work [19], the front-tracking scheme described in the present study is applied to real-world phase change problems including the directional solidification of pure tin and the Bridgman crystal growth of succinonitrile subject to convection in the melt region.

REFERENCES

1. J. M. Floryan and H. Rasmussen, Numerical methods for viscous flows with moving boundaries, *Appl. Mech. Rev.* **42**, 323 (1989).
2. L. Chen, S. V. Garimella, J. A. Reizes, and E. Leonardi, Motion of interacting gas bubbles in a viscous liquid including wall effects and evaporation, *Num. Heat Transfer* **31**, 629 (1997).
3. J. Glimm, D. Marchesin, and O. McBryan, A numerical method for two phase flow with an unstable interface, *J. Comput. Phys*. **39**, 179 (1981).
4. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).
5. H. S. Udaykumar, W. Shyy, and M. M. Rao, ELAFINT: a mixed Eulerian-Lagrangian method for fluid flow with complex and moving boundaries, AIAA Paper No. 94-1996 (1994).
6. G. Ryskin and L. G. Leal, Numerical solution of free-boundary problems in fluid mechanics. Parts 1, 2, 3, *J. Fluid Mech.* **148**, 1; 19; 37 (1984).
7. H. A. Stone and L. G. Leal, Relaxation and breakup of an initially extended drop in an otherwise quiescent fluid, *J. Fluid Mech.* **198**, 399 (1989).
8. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
9. J. A. Sethian, *Level Set Methods* (Cambridge University Press, New York, 1996).
10. B. T. Murray, A. A. Wheeler, and M. E. Glicksman, Simulations of experimentally observed dendritic growth behavior using a phase-field model, *J. Crystal Growth* **154**, 386 (1995).
11. C. R. Swaminathan and V. R. Voller, On the enthalpy method, *Int. J. Numer. Methods Heat Fluid Flow* **3**, 233 (1993).

12. J. E. Simpson and S. V. Garimella, An investigation of the solutal, thermal and flow fields in unidirectional alloy solidification, *Int. J. Heat Mass Transfer* **41**, 2485 (1998).

13. W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith, *Computational Fluid Dynamics with Moving Boundaries* (Taylor and Francis, 1996).

14. B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, *J. Comput. Phys.* **113**, 134 (1994).

15. M. R. H. Nobari and G. Tryggvason, Numerical simulations of three-dimensional drop collisions, *AIAA J.* **34**, 750 (1996).

16. G. J. Yeoh, G. de Vahl Davis, E. Leonardi, H. C. de Groh, and M. Yao, A numerical and experimental study of natural convection and interface shape in crystal growth, *First Int. Conf. Transport Phenomena in Processing*, Honolulu, Hawaii (March 1992).

17. V. Jayaraman, H. S. Udaykumar, and W. S. Shyy, Adaptive unstructured grid for three-dimensional interface representation, *Num. Heat Transfer* **32**, 247 (1997).

18. G. Labonia, V. Timchenko, J. E. Simpson, S. V. Garimella, E. Leonardi, and G. de Vahl Davis, Reconstruction and advection of a moving interface in three dimensions on a fixed grid, *Num. Heat Transfer* **B34**, 121 (1998).

19. C. Y. Li, J. E. Simpson, and S. V. Garimella, A Fixed-Grid Front-Tracking Algorithm for Solidification Problems. Part II – Directional Solidification with Melt Convection, *Num. Mass Transfer* (in press).

20. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes - The Art of Scientific Computing* (Cambridge University Press, New York, 1986).

21. H. S. Udaykumar, R. Mittal, and W. Shyy, Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids, *J. Comp. Phys.* **153**, 535 (1999).

22. M. N. Özişik, *Heat Conduction* (John Wiley, 1980).

23. V. Alexiades and A. D. Solomon, *Mathematical Modeling of Melting and Freezing Processes* (Hemisphere, Washington D.C., 1993).

24. D. Poulikakos, *Conduction Heat Transfer* (Prentice Hall, Englewood Cliffs, New Jersey, 1994).

25. H. S. Carslaw and J. C. Jaeger, *Conduction of Heat in Solid*, 2nd edn. (Clarendon Press, Oxford, 1959).

26. W. H. Cubberly, *Metals Handbook, Properties and Selection: Nonferrous Alloys and Pure Metals.* 9th Edn. (ASM. Metal Park, Ohio, 1979).

27. N. Shamsundar and E. M. Sparrow, Analysis of multidimensional conduction phase-change via the enthalpy method, *J. Heat Transfer*, 333 (1975).

28. J. E. Simpson and S. V. Garimella, The influence of gravity levels on the horizontal Bridgman crystal growth of an alloy, *Int. J. Heat Mass Transfer* **43**, 1905 (2000).

NOMENCLATURE

Roman Symbols

A_K^{ϕ}	coefficients in discrete equations
A	area
a, b, c, d	coefficients for the polynomial equation
c_p	specific heat capacity at constant pressure
D_K^{ϕ}	diffusive coefficients in discrete equations
D	species diffusion coefficient
f	volume fraction
g	contravariants from the coordinate transformation
H	height of computational domain
K	slope of interface
k	thermal conductivity
L	length
n	ordinate in normal direction
m	slope of liquidus line
r	radial coordinate
s(t)	solidification front location
St	Stefan number, $c_p(T_H - T_m)/\Delta H$
t	time
T	temperature
U	velocity
V	front moving velocity

V	volume
w	relaxation factor
x, y, z	Cartesian coordinates

Greek Symbols

α	thermal diffusivity
ε	tolerance
ΔH	Enthalpy of freezing
Δt	time step
$\Delta x, \Delta y$	spatial mesh sizes
ρ	density
τ	ordinate in tangential direction
ξ, η, ζ	transformed coordinates

Subscripts

0	initial condition
1, 2, 3	*x, y* and *z* directions
e	east face
E	east cell
i, j	mesh point index
l	liquid
m	at solidification front
n	north face

N	north cell
P	grid node
s	south face; solid
S	south cell
w	west face
W	west cell

Superscripts

*	reference or dimensional quantity
n	time step

FIGURE CAPTIONS

Fig. 1. Schematic diagrams outlining the procedure for recovering new marker points at time t + Δt both in the bulk of the domain and at corners. (a) Single set of marker points and (b) *Two* sets of marker points in both vertical and horizontal directions.

Fig. 2. (a) The local coordinate system at each surface element transformed from Cartesian coordinates (x,y,z) to transformed coordinates (ξ,η,ζ). (b) Nomenclature used for interface temperature gradient equations.

Fig. 3. Schematic indicating the nomenclature used for the finite difference stencil around the interface for a 1D case.

Fig. 4. Schematic of the problem geometry and boundary conditions for the benchmark Neumann melting problem.

Fig. 5. Numerically simulated and analytically determined melting front locations for the 1D melting problem: (a) St = 0.1, (b) St = 2.85 and (c) St = 5.0.

Fig. 6. Schematic of the 2D cylindrical melting problem.

Fig. 7. Numerically simulated and analytically determined melting front locations for the 2D cylindrical melting problem: (a) Q' = 500 W/cm, (b) Q' = 1000 W/cm, and (c) Q' = 2000 W/cm.

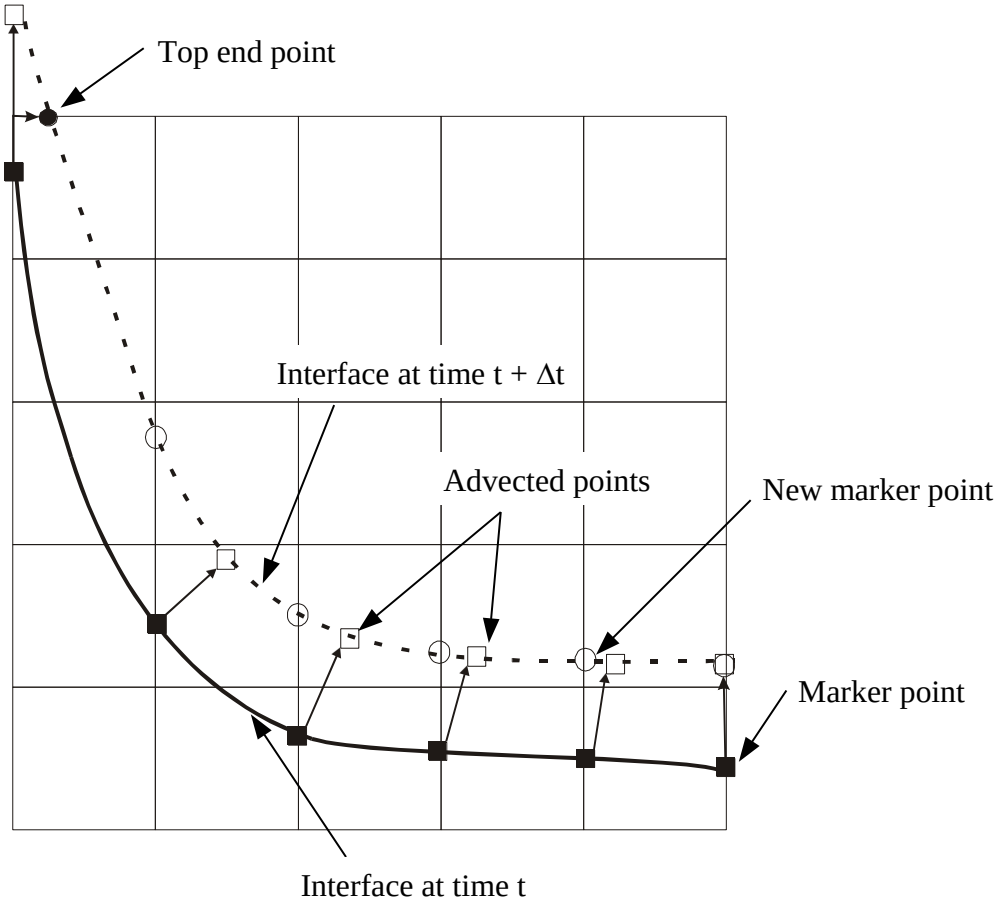
Fig. 8. Numerically simulated and analytically determined results for the 2D cylindrical melting problem. The line heat source for these results is Q' = 1000 W/cm. (a) Melt temperatures at three different times, (b) Contour plot of temperatures at t = 107.9 s, and (c) Melting front locations.

Fig. 9. Schematic of the problem geometry and boundary conditions for the ingot casting problem.

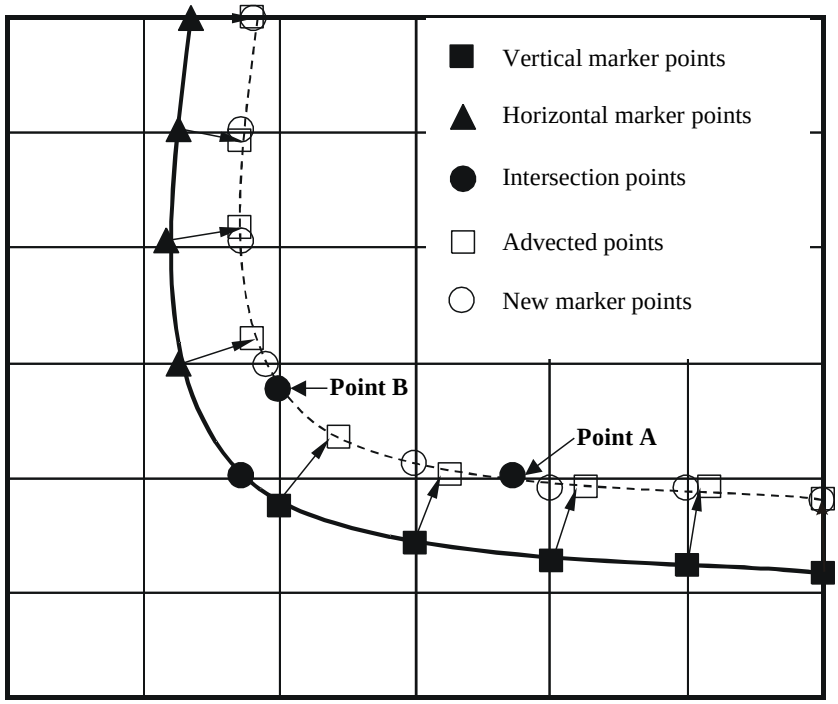
Fig. 10. Results for the ingot casting problem: (a) Isotherms at t = 24 s, (b) Isotherms at t = 40 s, and (c) Comparison of front locations at every 4 s for the ingot casting problem; enthalpy method represented by dashed lines and front tracking method from the present work with one set and two sets of marker points represented by dash dot and solid lines.

Table 1. Thermophysical properties of pure aluminum [25, 26].

Property	Value	Units
k_s	237.65	W/mK
k_l	94.14	W/mK
$C_{p,s}$	903.74	J/kgK
$C_{p,l}$	1079.47	J/kgK
ρ	2700	kg/m^3
ΔH	397.48	kJ/kg

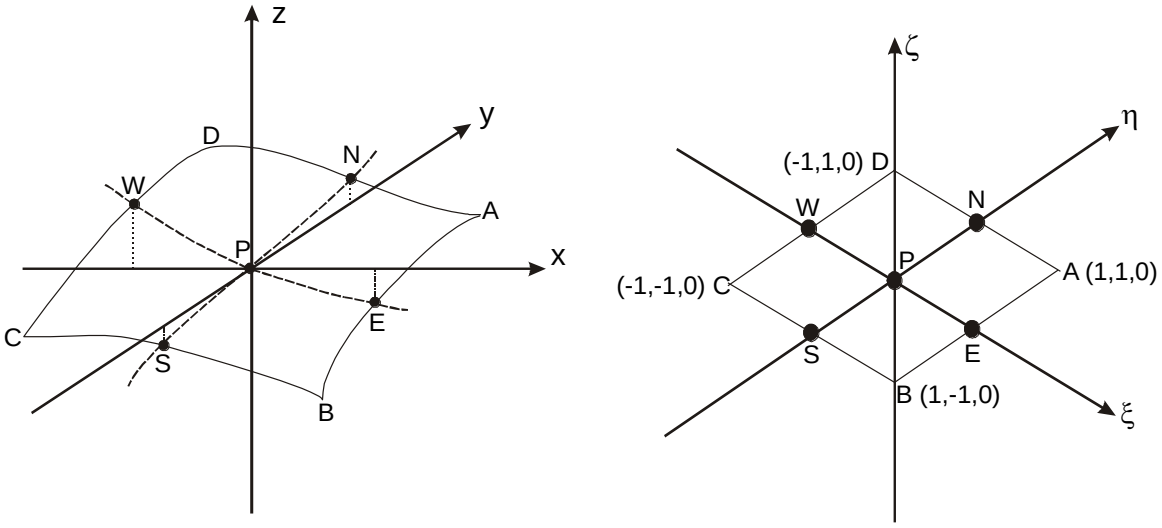


(a)

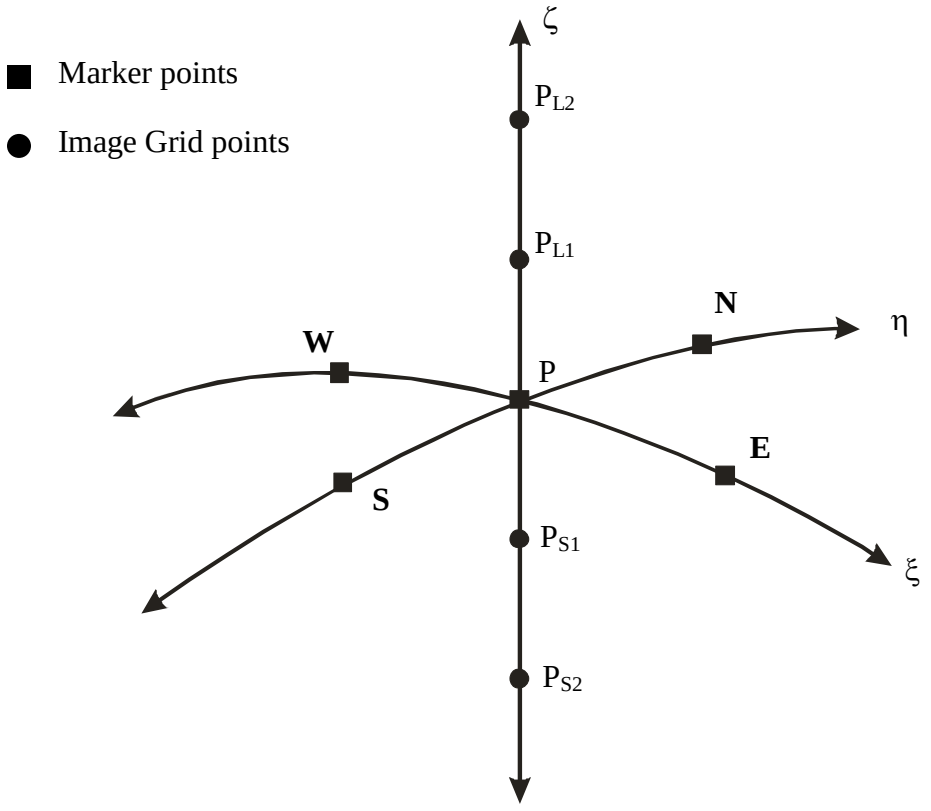


(b)

Figure 1, Li et al.



(a)



(b)

Figure 2, Li et al.

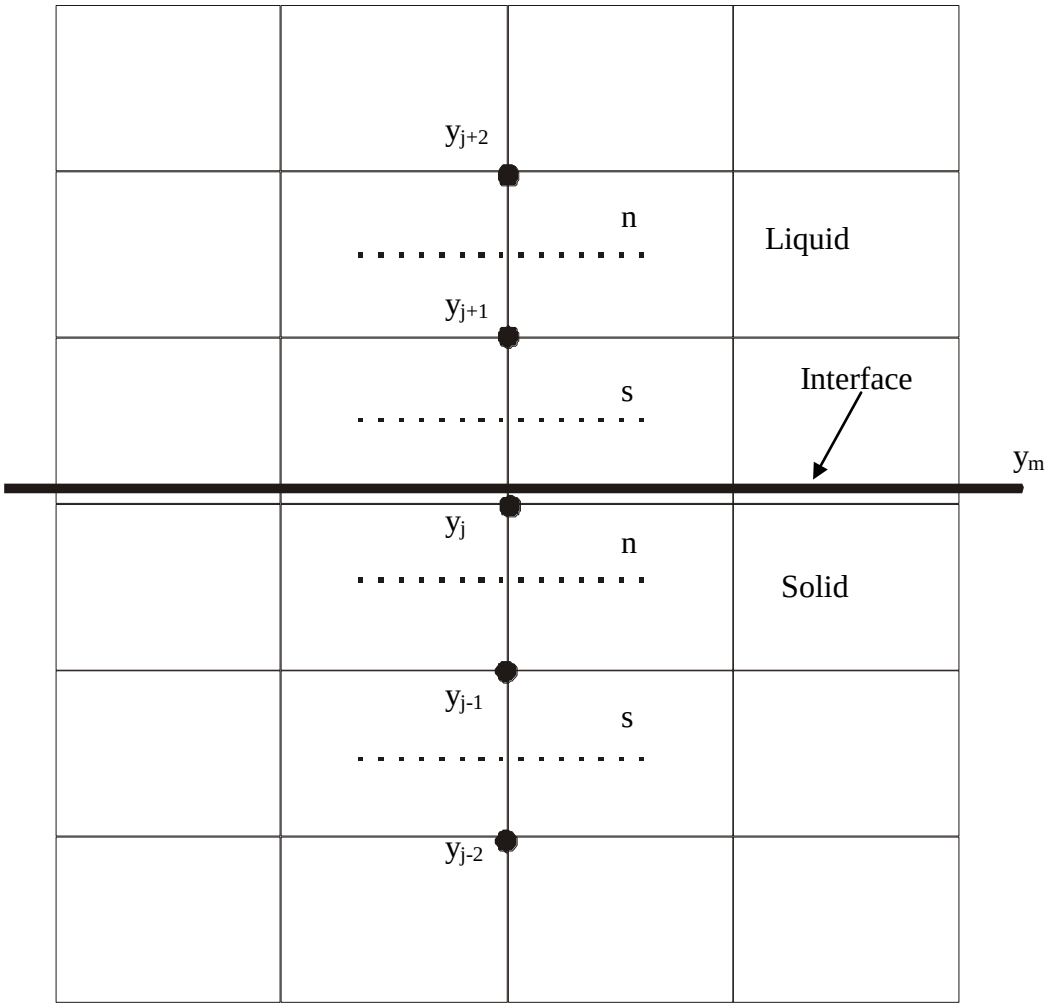


Figure 3, Li et al.

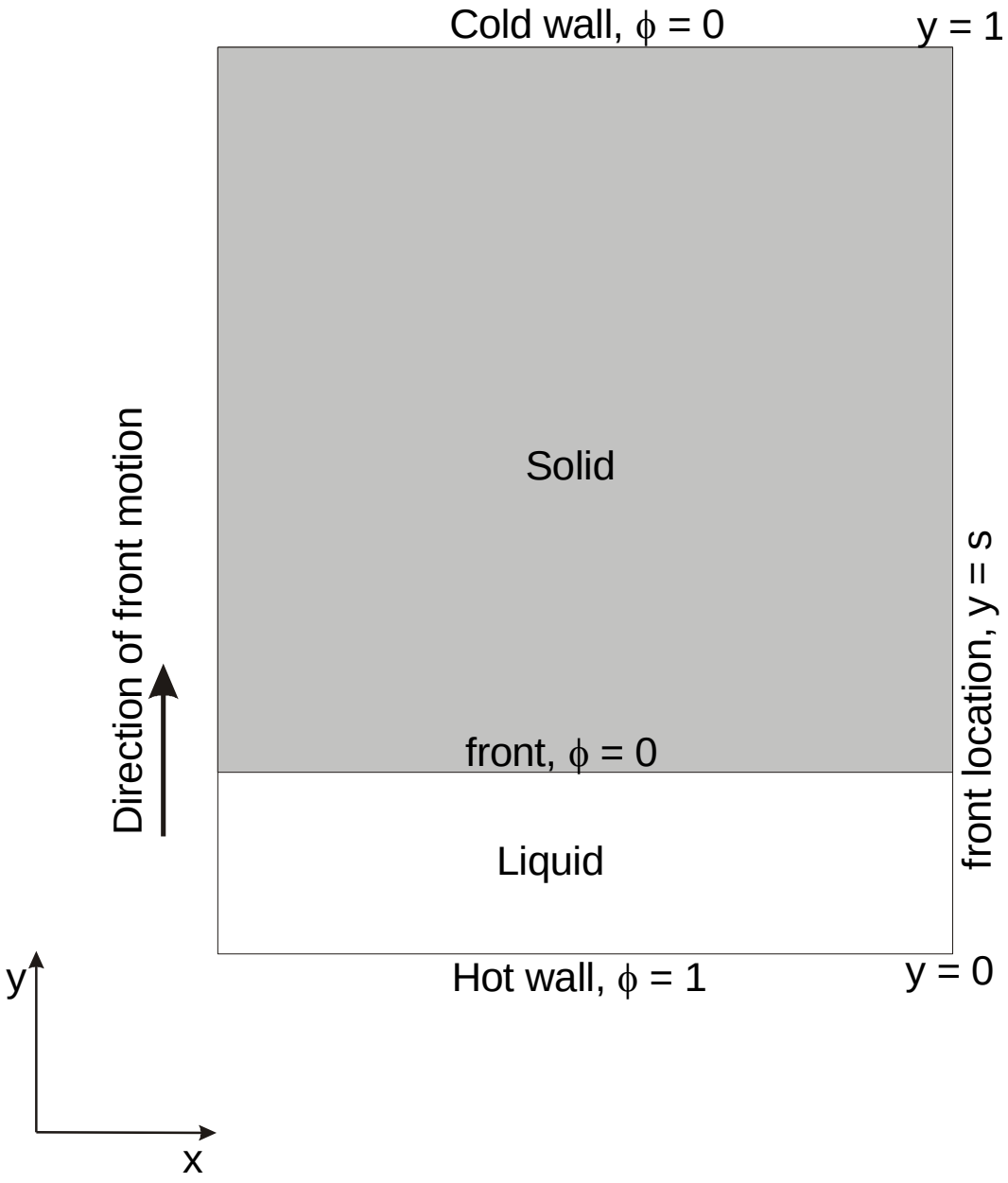
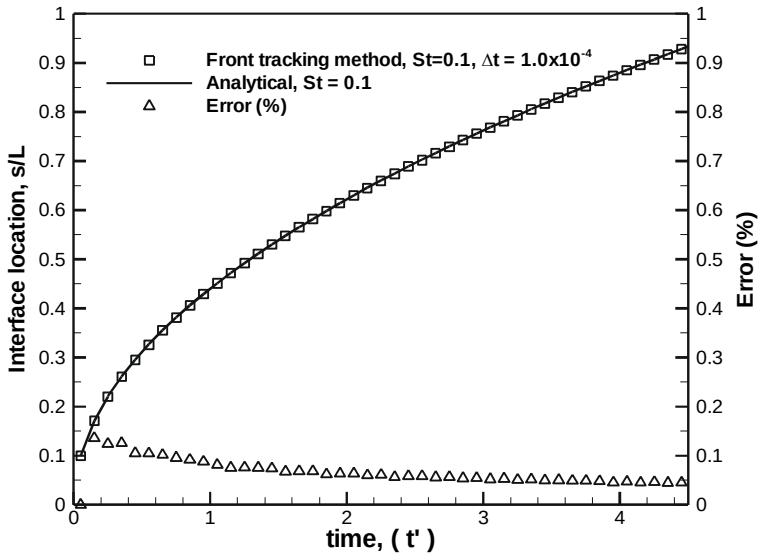
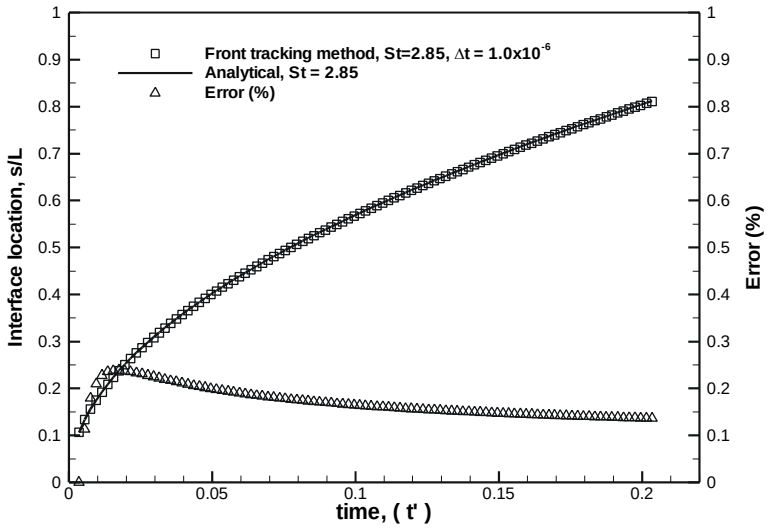


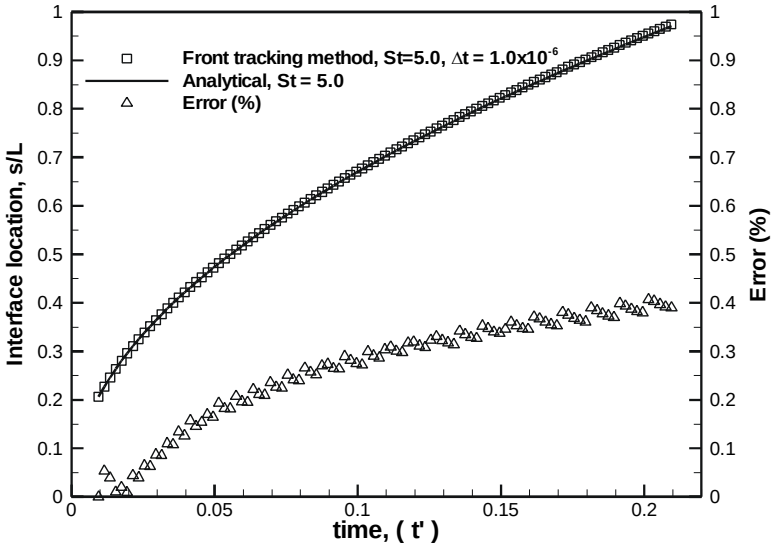
Figure 4, Li et al.



(a)



(b)



(c)

Figure 5, Li et al.

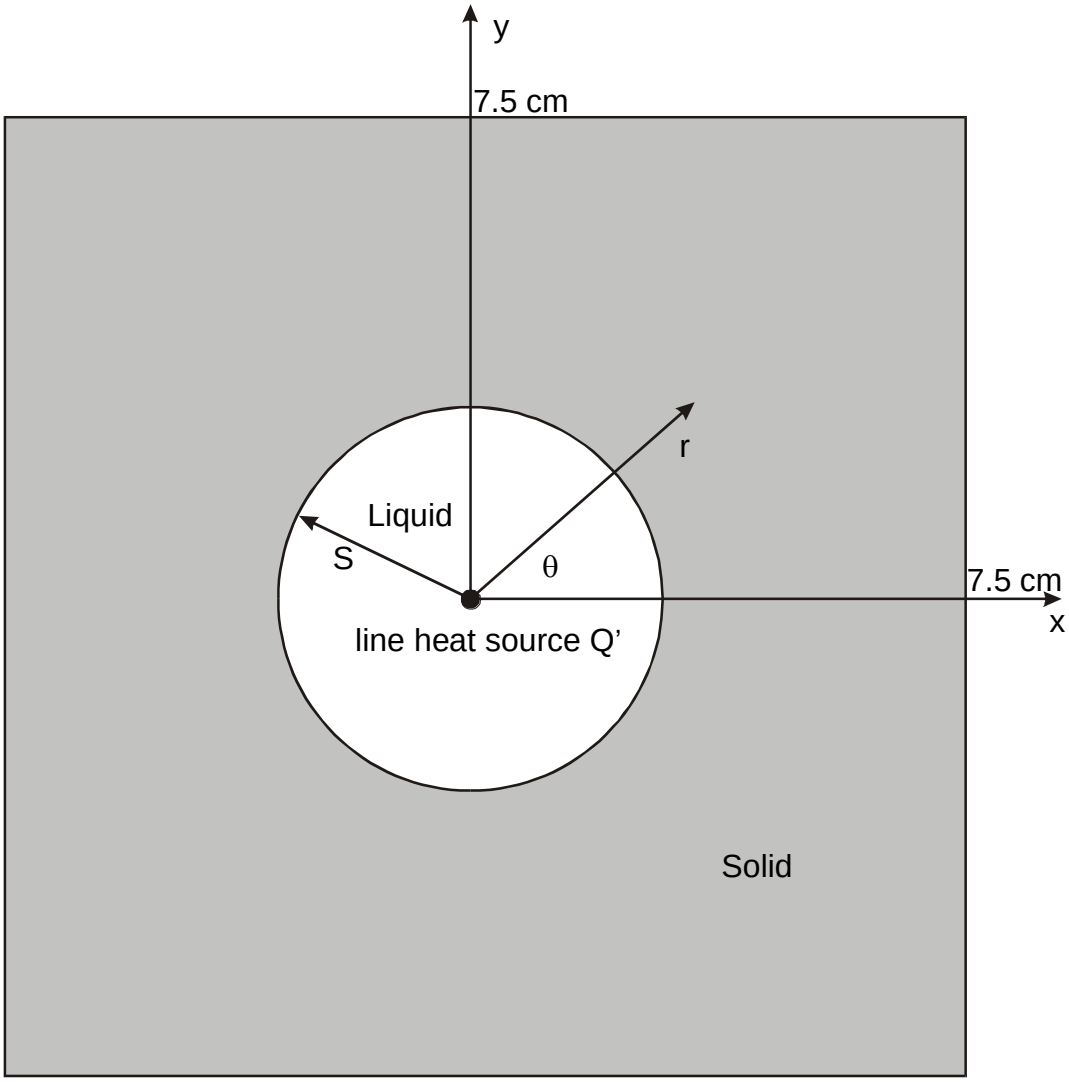
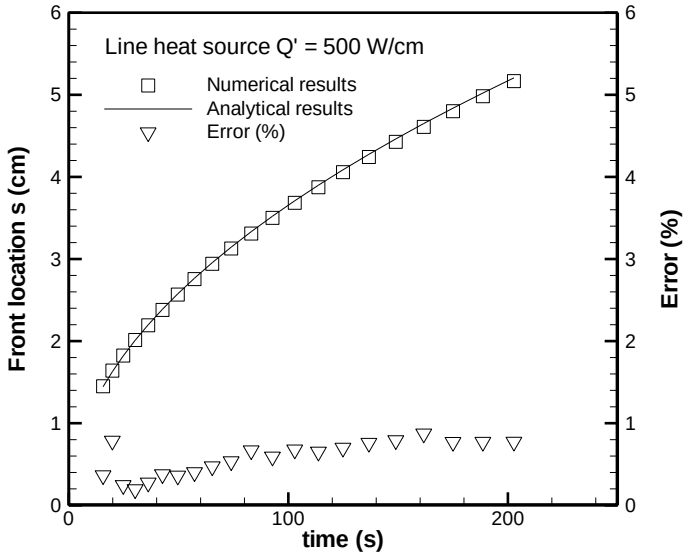
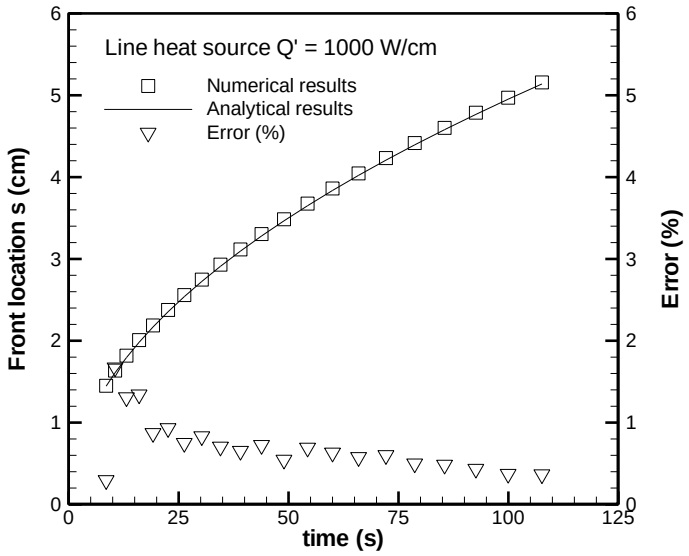


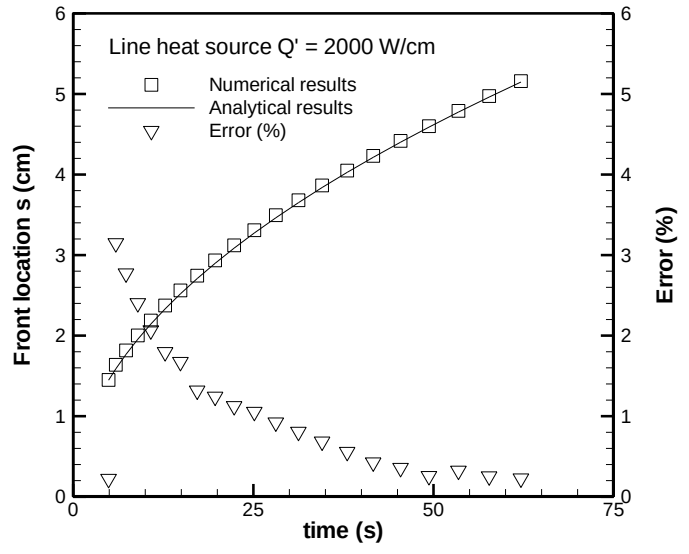
Figure 6, Li et al.



(a)

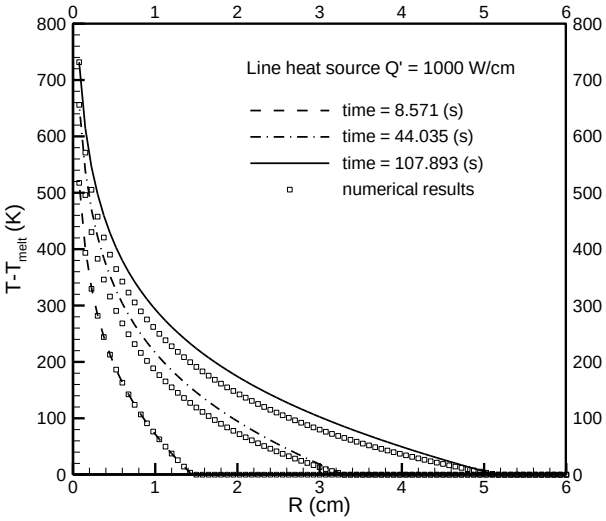


(b)

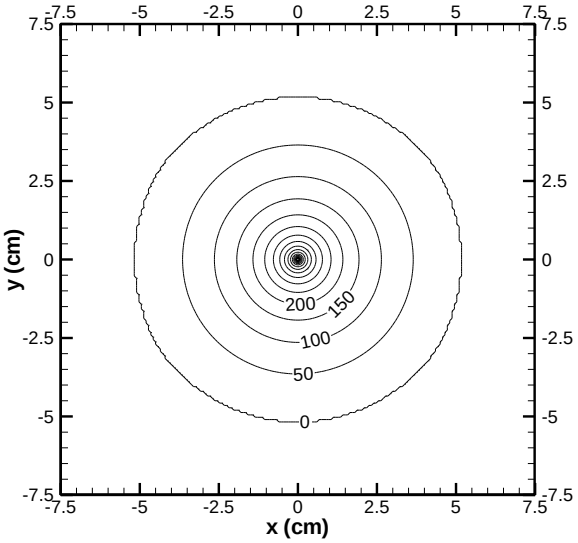


(c)

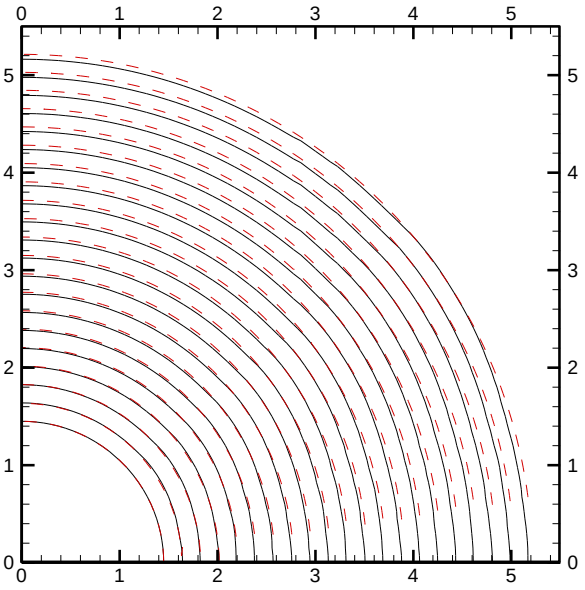
Figure 7, Li et al.



(a)

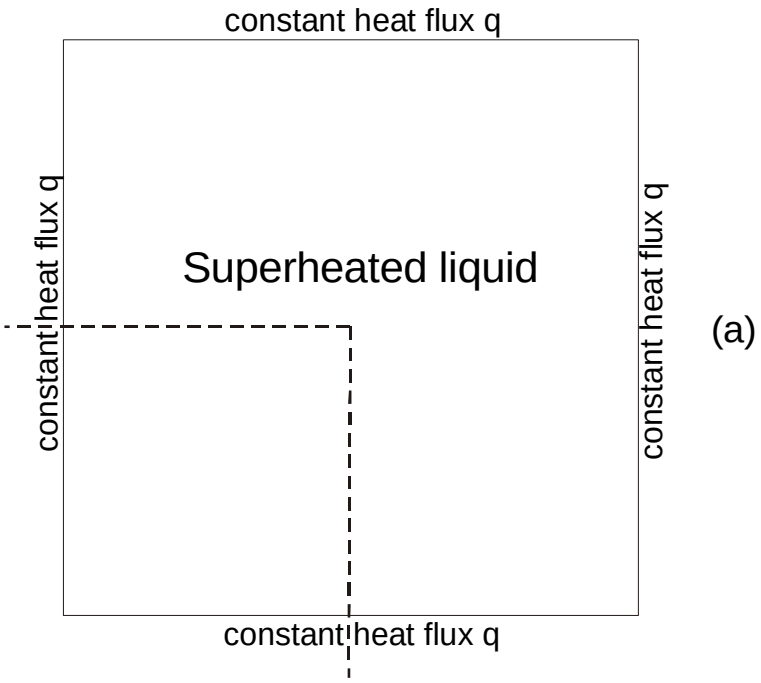


(b)



(c)

Figure 8, Li et al.



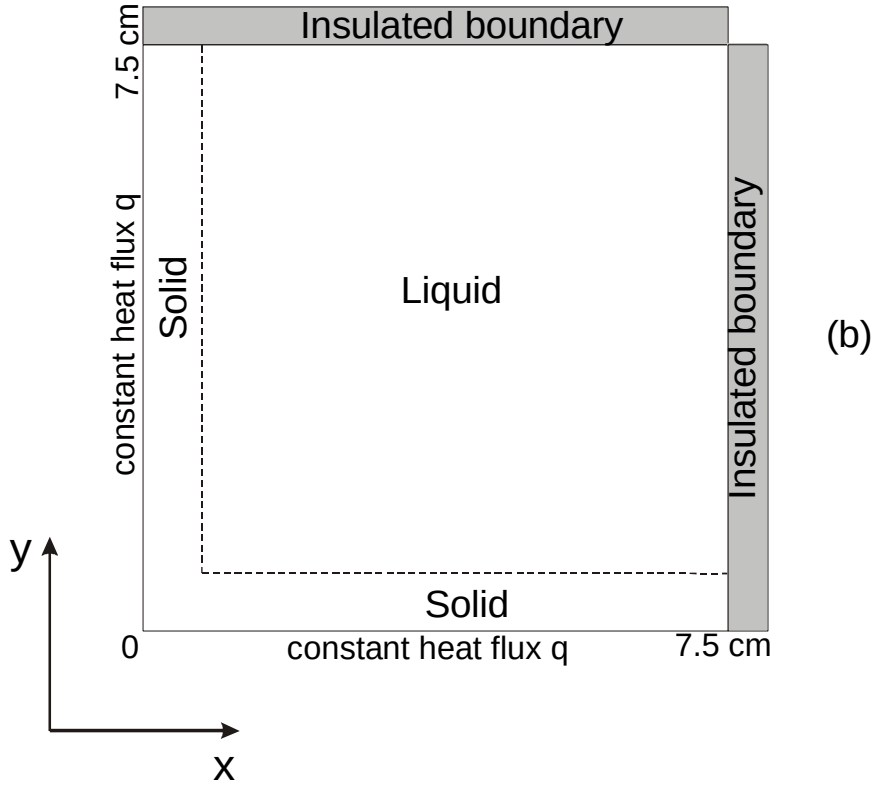
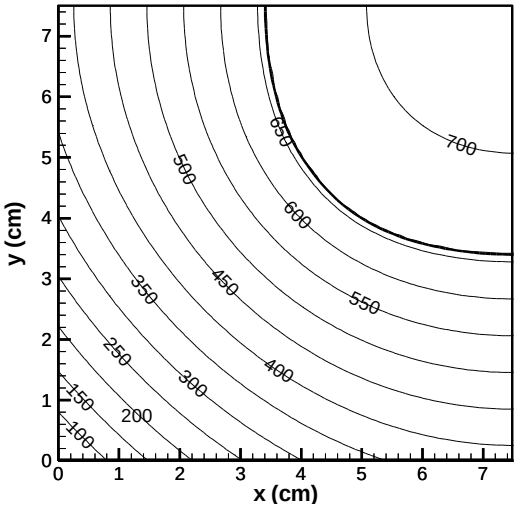
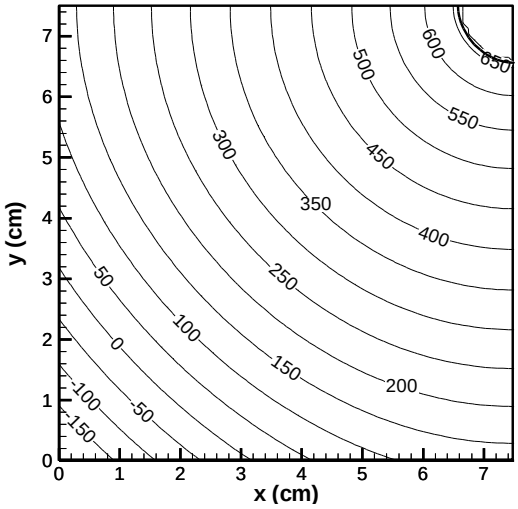


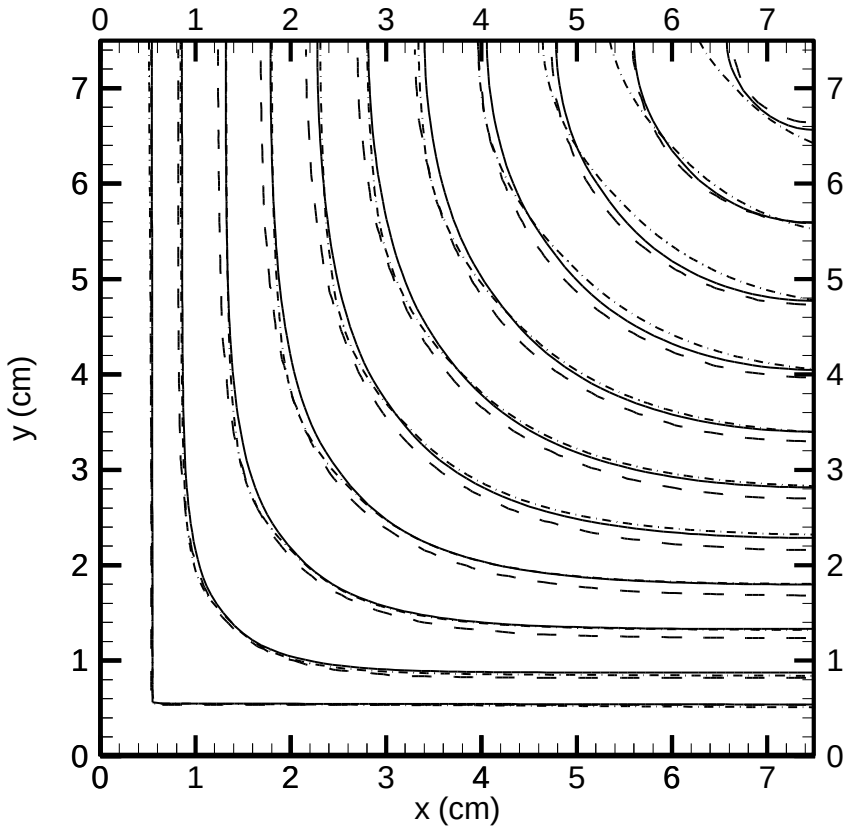
Figure 9, Li et al.



(a)



(b)



(c)

Results using enthalpy method [28]

Results using two sets of marker points

Results using one set of marker points

Figure 10, Li et al.