A Second Order Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains *

Frederic Gibou †
Ronald Fedkiw ‡
Li-Tien Cheng §
Myungjoo Kang ¶

November 27, 2001

Abstract

In this paper, we consider the variable coefficient Poisson equation with Dirichlet boundary conditions on an irregular domain and show that one can obtain second order accuracy with a rather simple discretization. Moreover, since our discretization matrix is symmetric, it can be inverted rather quickly as opposed to the more complicated nonsymmetric discretization matrices found in other second order accurate discretizations of this problem. Multidimensional computational results are presented to demonstrate the second order accuracy of this numerical method. In addition, we use our approach to formulate a second order accurate symmetric implicit time discretization of the heat equation on irregular domains. Then, we briefly consider Stefan problems.

*Research supported in part by ONR N00014-97-1-0027
†Mathematics Department & Computer Science Department, Stanford University, Stanford, CA 94305
‡Computer Science Department, Stanford University, Stanford, CA 94305
§Department of Mathematics, University of California San Diego, La Jolla, CA 92093
¶Department of Mathematics, University of California Los Angeles, Los Angeles, California 90095

1 Introduction

In [20] the Ghost Fluid Method [7] was used as a guide to develop a first order accurate symmetric discretization of the variable coefficient Poisson equation in the presence of an irregular interface across which the variable coefficients, the solution and the derivatives of the solution may have jumps. This new numerical method was applied to two phase incompressible flow in [15] and to incompressible flame front discontinuities in [21]. In this paper, we consider a similar Poisson equation where Dirichlet boundary conditions (instead of jump conditions) are imposed on the irregular interface. In this case, the solution is not coupled across the interface, and we are able to design a second order accurate symmetric discretization as opposed to the first order accurate discretization proposed in [20] for the jump condition case. Both the discretization proposed here and that in [20] yield symmetric matrices that can readily be inverted with a number of fast methods, e.g. we use a Preconditioned Conjugate Gradient (PCG) method (see e.g. [10]) in both instances. We note that [18] designed a second order accurate method for the jump condition case as well, but their discretization matrix is not symmetric.

This new second order accurate symmetric discretization is also quite useful for solving Stefan problems. We use a level set formulation [22] to represent the interface location and a finite difference discretization of the heat equation on a Cartesian grid to solve for the temperature. In order to avoid the stringent $O(\triangle x^2)$, or even worse $O(\theta^2 \triangle x^2)$ with $0 < \theta \leq 1$ for cells cut by the interface, time step restriction imposed by explicit time discretization of the heat equation, we use an implicit time discretization. This requires a matrix inversion that can be rather time consuming especially if one uses a nonsymmetric discretization of the spatial terms. This was the case in [5] where the nonsymmetric matrix was inverted with the (very slow) Gauss-Seidel method (see e.g. [10]). Our alternative symmetric discretization allows a (relatively fast) PCG method to be used for the matrix inversion.

The earliest level set method for solidification type problems was presented in [27] where the authors recast the equations of motion into a boundary integral equation and used the level set method to update the location of the interface. In [5] the boundary integral equations were avoided by using a finite difference method to solve the heat equation on a Cartesian grid with Dirichlet boundary conditions imposed on the interface. The jump in the first derivatives of the temperature was used to compute an interface velocity which was extended to a band about the interface and used to evolve a level set function in time. The velocity calculation in [5] is rather awkward and both the standard grid and a 45^o rotated grid are used to aid in the removal of nonphysical grid anisotropy effects. This velocity computation was improved upon in [17] where the authors show good agreement between the level set method and phase field methods for the case where the thermal conductivities are the same in both materials. In addition, [17] showed that the level set method continues to perform well for the case where the thermal conductivities are different in the two materials. For more details on phase field methods for the Stefan problem, see [17] and the references therein.

In [30], the authors discretized the heat equation on a Cartesian grid in a manner very similar to that proposed in [5] resulting in a nonsymmetric matrix when applying an implicit time discretization. [30] used front tracking to update the location of the interface improving upon the front tracking approach proposed in [14] which used the smeared out immersed boundary method from [23] and explicit time stepping. The interested reader is also referred to [4] for an interesting analysis of the immersed boundary method in conjunction with heat transport.

In [12], the authors solved a variable coefficient Poisson equation in the presence of an irregular interface where Dirichlet boundary conditions were imposed. They used a finite volume method that results in a nonsymmetric discretization matrix. Both multigrid methods and adaptive mesh refinement were used in [12], and in [11] this nonsymmetric finite volume discretization was coupled to a volume of fluid front tracking method in order to solve the Stefan problem.

The interested reader is referred to [14], [5] and the references therein for an extensive summary of computational results for the Stefan problem. Most notably, [25] uses adaptive finite element methods for both the heat equation and for the interface evolution producing spectacular (and rare) three dimensional results. Other impressive three dimensional results can be found in the phase field model of [16] and the finite difference diffusion Monte Carlo method of [24].

2 Equations

2.1 Poisson Equation

Consider a Cartesian computational domain, Ω, with exterior boundary, $\partial\Omega$, and a lower dimensional interface, Γ, that divides the computational domain into disjoint pieces, Ω^- and Ω^+. The variable coefficient Poisson equation is given by

$$\nabla \cdot (\beta(\vec{x})\nabla u(\vec{x})) = f(\vec{x}), \qquad \vec{x} \in \Omega \tag{1}$$

where $\vec{x} = (x, y, z)$ are the spatial dimensions, $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ is the divergence operator, and $\beta(\vec{x})$ is assumed continuous on each disjoint subdomain, Ω^- and Ω^+, but may be discontinuous across the interface Γ. Furthermore, $\beta(\vec{x})$ is assumed to be positive and bounded below by some $\epsilon > 0$. On $\partial\Omega$, either Dirichlet boundary conditions of $u(\vec{x}) = g(\vec{x})$ or Neumann boundary conditions of $u_n(\vec{x}) = h(\vec{x})$ are specified. Note that $u_n = \nabla u \cdot \vec{N}$ is the normal derivative of u with normal $\vec{N}$.

In [20], equation 1 was solved with a first order numerical method when the jump conditions $[u]_\Gamma = a(\vec{x})$ and $[\beta u_n]_\Gamma = b(\vec{x})$ were specified across the interface. If instead, a Dirichlet boundary condition of $u_\Gamma = c(\vec{x})$ is specified on the interface, then equation 1 decouples into two separate equations, one on Ω^- and one on Ω^+. Any jumps of u, βu_n or β across the interface can be ignored allowing equation 1 to be considered separately and independently on Ω^- and on Ω^+.

2.2 Heat Equation

Starting from conservation of mass, momentum and energy we can obtain

$$\rho e_t + \rho\vec{V} \cdot \nabla e + p\nabla \cdot \vec{V} = \nabla \cdot (k\nabla T) \tag{2}$$

where ρ is the density, e is the internal energy per unit mass, $\vec{V} =< v_1, v_2, v_3 >$ are the velocities, p is the pressure, k is the thermal conductivity and T is the temperature. Assuming e depends on at most temperature, and that the specific heat at constant volume, c_v, is constant leads to $e = e_o + c_v(T - T_o)$ where e_o is the internal energy per unit mass at some reference temperature T_o [3]. This and the incompressibility assumption, $\nabla \cdot \vec{V} = 0$, simplify equation 2 to

$$\rho c_v T_t + \rho c_v \vec{V} \cdot \nabla T = \nabla \cdot (k\nabla T) \tag{3}$$

which can be further simplified to the standard heat equation

$$\rho c_v T_t = \nabla \cdot (k \nabla T) \tag{4}$$

ignoring the effects of convection, i.e. setting $\vec{V} = 0$.

2.3 Interface Velocity and Jump Conditions

Unreacted and *reacted* incompressible flows are separated by an interface across which the unreacted material is converted to the reacted material, and we use "u" and "r" subscripts to refer to the unreacted and reacted materials, respectively. The interface velocity is denoted by $\vec{W} = D\vec{N}$ where D is the normal component of the interface velocity and $\vec{N} =< n_1, n_2, n_3 >$ is the local unit normal to the interface. The normal component of the interface velocity is calculated by adding the unreacted materials normal velocity to the reaction speed, S. That is, $D = (V_N)_u + S$ where $V_N = \vec{V} \cdot \vec{N}$ is the normal velocity.

Conservation of mass, momentum and energy implies the standard Rankine-Hugoniot jump conditions across the interface

$$[\rho(V_N - D)] = 0 \tag{5}$$

$$\left[\rho(V_N - D)^2 + p\right] = 0 \tag{6}$$

$$\left[\left(\rho e + \frac{\rho(V_N - D)^2}{2} + p\right)(V_N - D)\right] = \left[k\nabla T \cdot \vec{N}\right] \tag{7}$$

where $[A] = A_r - A_u$ defines "$[\cdot]$" as the jump in a quantity across the interface. Equation 7 was derived assuming that the tangential velocities are continuous across the interface which is true when $D \neq V_N$, i.e. $S \neq 0$ (see e.g. [8]). This equation can be rewritten as

$$-\rho_u S\left(\left[e + \frac{p}{\rho}\right] + \frac{\rho_u^2 S^2}{2}\left[\frac{1}{\rho^2}\right]\right) = \left[k\nabla T \cdot \vec{N}\right] \tag{8}$$

using equation 5 and $D = (V_N)_u + S$. Assuming the enthalpy per unit mass, $h = e + \frac{p}{\rho}$ depends on at most temperature, and that the specific heat at constant pressure, c_p, is constant leads to $h = h_o + c_p\,(T - T_o)$ where h_o

is the enthalpy per unit mass at some reference temperature T_o [3]. This allows us to rewrite equation 8 as

$$-\rho_u S\left([h_o]+[c_p](T_I-T_o)+\frac{\rho_u^2 S^2}{2}\left[\frac{1}{\rho^2}\right]\right)=\left[k\nabla T\cdot\vec{N}\right] \tag{9}$$

where we have used the fact that the temperature is continuous across the interface, $[T]=0$, and labeled the interface temperature T_I. It is convenient to choose the reference temperature T_o equal to the standard temperature at which the reaction takes place, e.g. in the case of freezing water $T_o=273K$.

For the Stefan problem, we assume that there is no expansion across the front (i.e. $[\rho]=0$) reducing equation 5 to $[V_N]=0$, equation 6 to $[p]=0$, and equation 9 to

$$-\rho S\left([h_o]+[c_p](T_I-T_o)\right)=\left[k\nabla T\cdot\vec{N}\right] \tag{10}$$

where $\rho=\rho_u=\rho_r$. Furthermore, the standard interface boundary condition of $T_I=T_o$ reduces this last equation to

$$-\rho S[h_o]=\left[k\nabla T\cdot\vec{N}\right] \tag{11}$$

where $[h_o]$ is calculated at the reaction temperature of $T_I=T_o$.

2.4 Level Set Equation

The level set equation

$$\phi_t+\vec{W}\cdot\nabla\phi=0 \tag{12}$$

is used to keep track of the interface location as the set of points where $\phi=0$. The unreacted and reacted materials are then designated by the points where $\phi>0$ and $\phi\le 0$ respectively. Using $\phi\le 0$ instead of $\phi=0$ for the reacted points removes the measure zero ambiguity of points that happen to lie on the interface. In this sense, the numerical interface lies in between $\phi=0$ and the positive values of ϕ and can be located numerically by finding the zero level of ϕ. To keep the values of ϕ close to those of a signed distance function, i.e. $|\nabla\phi|=1$, the reinitialization equation

$$\phi_\tau+S(\phi_o)\left(|\nabla\phi|-1\right)=0 \tag{13}$$

is iterated for a few steps in ficticious time, τ. Here $S(\phi_o)$ is a smoothed out sign function. The level set function is used to compute the normal $\vec{N}=\frac{\nabla\phi}{|\nabla\phi|}$ and the curvature $\kappa=-\nabla\cdot\vec{N}$ in a standard fashion. For more details on the level set function see [7, 22, 29, 15].

3 Numerical Method

3.1 Poisson Equation

Consider the variable coefficient Poisson equation in one spatial dimension

$$(\beta u_x)_x = f \tag{14}$$

with Dirichlet boundary conditions of $u = g$ on the interface where $\phi = 0$. One can consider each simply connected portion of the domain separately, i.e. equation 14 can be solved on the subdomain where $\phi \leq 0$ independent of the solution procedure for the subdomain where $\phi > 0$. Although in practice, it is usually simpler and more efficient to solve for both subdomains at the same time.

The computational domain is discretized into cells of size $\triangle x$ where the cell centers are referred to as grid points or grid nodes with the i-th grid node located at x_i. The cell edges are referred to as fluxes so that the two fluxes bounding the i-th computational cell are located at $x_{i\pm\frac{1}{2}}$. The solution to the Poisson equation is computed at the grid nodes and is written as $u_i = u(x_i)$. An analogous definition holds for f_i, β_i, and ϕ_i. Since β and ϕ are known only at the grid nodes x_i, their values at the fluxes is defined by the linear average of the nodal values, e.g. $\phi_{i+\frac{1}{2}} = \frac{\phi_i + \phi_{i+1}}{2}$ is a second order accurate approximation to ϕ at the flux located between the i-th and $(i+1)$-st cells.

In the absence of an irregular interface, the standard discretization for equation 14

$$\frac{\beta_{i+\frac{1}{2}}\left(\frac{u_{i+1}-u_i}{\triangle x}\right) - \beta_{i-\frac{1}{2}}\left(\frac{u_i - u_{i-1}}{\triangle x}\right)}{\triangle x} = f_i \tag{15}$$

can be used to solve this problem with Dirichlet $u = g$ boundary conditions on $\partial\Omega$ enforced by setting $u_i = g_i$ when x_i is a boundary point. For each unknown, u_i, equation 15 is used to fill in one row of a matrix creating a linear system of equations. Since the resulting matrix is symmetric, a wide number of fast linear solvers can be used. In the examples section, the symmetric linear system is solved with a PCG method using an Incomplete Choleski preconditioner [10].

Next, suppose that an interface point, x_I, is located in between two grid points x_i and x_{i+1} with a Dirichlet $u = u_I$ boundary condition applied at x_I. Here we will only address computing the numerical solution to the left of

x_I noting that it is independent of the solution to the right of x_I. Equation 15 is still valid for all the unknowns to the left and including u_{i-1}, but can no longer be applied at x_i to solve for u_i since the subdomain to the left of x_I does not contain a valid value of u_{i+1}. This can be remedied by defining a ghost value of u^G_{i+1} at x_{i+1} and rewriting equation 15 as

$$\frac{\beta_{i+\frac{1}{2}}\left(\frac{u^G_{i+1}-u_i}{\triangle x}\right)-\beta_{i-\frac{1}{2}}\left(\frac{u_i-u_{i-1}}{\triangle x}\right)}{\triangle x}=f_i \tag{16}$$

in order to solve for u_i. Possible candidates for u^G_{i+1} include

$$u^G_{i+1}=u_I \tag{17}$$

$$u^G_{i+1}=\frac{u_I+(\theta-1)\,u_i}{\theta} \tag{18}$$

and

$$u^G_{i+1}=\frac{2u_I+(2\theta^2-2)\,u_i+(-\theta^2+1)\,u_{i-1}}{\theta^2+\theta} \tag{19}$$

using constant, linear and quadratic extrapolation respectively. Here $\theta\in[0,1]$ is defined by $\theta=\frac{x_I-x_i}{\triangle x}$, and can be calculated as $\theta=\frac{|\phi|}{\triangle x}$ since $\phi=0$ at x_I and is signed distance away from x_I. Since equations 18 and 19 are poorly behaved for small θ, they are not used when $\theta\le\triangle x$. Instead, u_i is set equal to u_I which effectively moves the interface from x_I to x_i. This second order accurate perturbation of the interface location does not degrade the overall second order accuracy of the solution obtained using equation 15 to solve for the remaining unknowns. Furthermore, $u_i=u_I$ is second order accurate as long as the desired solution has bounded first derivatives.

Plugging equation 19 into equation 15 gives a nonsymmetric discretization of

$$\frac{\left(\frac{u_I-u_i}{\theta\triangle x}\right)-\left(\frac{u_i-u_{i-1}}{\triangle x}\right)}{.5\,(\theta\triangle x+\triangle x)}=f_i \tag{20}$$

in the case of $\beta=1$. Equation 20 is the nonsymmetric discretization used in [5] and [30] to obtain second order accurate numerical methods. That is, both [5] and [30] use the quadratic extrapolation given in equation 19 to obtain second order accuracy. Alternatively, plugging equation 18 into equation 15 gives a symmetric discretization of

$$\frac{\beta_{i+\frac{1}{2}}\left(\frac{u_I-u_i}{\theta\triangle x}\right)-\beta_{i-\frac{1}{2}}\left(\frac{u_i-u_{i-1}}{\triangle x}\right)}{\triangle x}=f_i \tag{21}$$

based on linear extrapolation in the partial cell. Surprisingly, this symmetric discretization is second order accurate as well. This was first pointed out in [6] and is elaborated on here.

Assume that the standard second order accurate discretization in equation 15 is used to obtain the standard linear system of equations for u at every grid point except for x_i, and equation 16 is used to write a linear equation for u_i introducing a new unknown u_{i+1}^G. The system is closed with equation 18 for u_{i+1}^G. In practice, equations 18 and 16 are combined to obtain equation 21 and a symmetric linear system. Solving this linear system of equations leads to well determined values (up to some prescribed tolerance near roundoff error levels) of u at each grid node in the subdomain as well as a well determined value of u_{i+1}^G (from equation 18). Designate $\vec{u}$ as the solution vector containing all these values of u.

Next, consider a modified problem where a Dirichlet boundary condition of $u_{i+1} = u_{i+1}^G$ is specified at x_{i+1} where u_{i+1}^G is chosen to be the value of u_{i+1}^G from $\vec{u}$ (defined above). This modified problem can be discretized to second order accuracy everywhere using the standard discretization in equation 15 at every node except at x_i where equation 16 is used. Note that equation 16 *is* the standard second order accurate discretization when a Dirichlet boundary condition of $u_{i+1} = u_{i+1}^G$ is applied at x_{i+1}. Thus, this new linear system of equations can be solved in standard fashion to obtain a second order accurate solution at each grid node. The realization that $\vec{u}$ (defined above) is an exact solution to *this* new linear system implies that $\vec{u}$ is a valid second order accurate solution to this modified problem.

Since $\vec{u}$ is a second order accurate solution to the modified problem, $\vec{u}$ can be used to obtain the interface location for the modified problem to second order accuracy. The linear interpolant that uses u_i at x_i and u_{i+1}^G at x_{i+1} predicts an interface location of *exactly* x_I. Since higher order accurate interpolants (higher than linear) can contribute at most an $O(\Delta x^2)$ perturbation of the interface location, the interface location dictated by the modified problem is at most an $O(\Delta x^2)$ perturbation of the true interface location, x_I. Thus, $\vec{u}$ is a second order accurate solution to a modified problem where the interface location has been perturbed by $O(\Delta x^2)$. This makes $\vec{u}$ a second order accurate solution to the original problem as well. (The interested reader is referred to a more detailed proof of a related discretization to a similar problem, see Jones and Menzies [13].)

Similarly, note that plugging equation 17 into equation 15 effectively perturbs the interface by an $O(\triangle x)$ amount resulting in a first order accurate algorithm.

In certain situations, β may only be known at the grid nodes and the interface in which case $\beta_{i+\frac{1}{2}}$ in equation 21 can be determined from a ghost value, β_{i+1}^G, and the usual averaging

$$\beta_{i+\frac{1}{2}} = \frac{\beta_i + \beta_{i+1}^G}{2} \tag{22}$$

noting that the ghost value is easily defined using linear extrapolation

$$\beta_{i+1}^G = \frac{\beta_I + (\theta - 1)\,\beta_i}{\theta} \tag{23}$$

according to equation 18.

In multiple spatial dimensions, the equations are discretized in a dimension by dimension manner using the one dimensional discretization outlined above. That is, the $(\beta u_x)_x$, $(\beta u_y)_y$ and $(\beta u_z)_z$ terms in equation 1 are each discretized independently in the same manner that $(\beta u_x)_x$ was discretized in equation 14 above.

3.2 Heat Equation

Consider the heat equation 4 with an explicit Euler time discretization

$$\frac{T^{n+1} - T^n}{\triangle t} = \frac{1}{\rho c_v} \nabla \cdot (k \nabla T^n) \tag{24}$$

and Dirichlet boundary conditions of $T = g$ on the interface where $\phi = 0$. Assuming that ρ and c_v are constant in each subdomain allows equation 24 to be rewritten as

$$\frac{T^{n+1} - T^n}{\triangle t} = \nabla \cdot \left(\hat{k} \nabla T^n \right) \tag{25}$$

where $\hat{k} = \frac{k}{\rho c_v}$. For stability, a time step restriction of

$$\triangle t \hat{k} \left(\frac{2}{(\theta_1 \triangle x)^2} + \frac{2}{(\theta_2 \triangle y)^2} + \frac{2}{(\theta_3 \triangle z)^2} \right) \leq 1 \tag{26}$$

is needed where θ_1, θ_2 and θ_3 are the cell fractions in each spatial dimension for cells cut by the interface with $0 < \theta_1, \theta_2, \theta_3 \leq 1$.

Implicit Euler time discretization

$$\frac{T^{n+1} - T^n}{\triangle t} = \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) \tag{27}$$

avoids the time step stability restriction in equation 26. Equation 27 can be rewritten as

$$T^{n+1} - \triangle t \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) = T^n \tag{28}$$

where the $\nabla \cdot \left(\hat{k} \nabla T^{n+1} \right)$ term is discretized in the same fashion as the variable coefficient Poisson equation (above) noting that each subdomain can be considered independently. For each unknown, T_i^{n+1}, equation 28 is used to fill in one row of a matrix creating a linear system of equations. Since the resulting matrix is symmetric, a wide number of fast linear solvers can be used. In the examples section, the symmetric linear system is solved with a PCG method using an Incomplete Choleski preconditioner [10]. Note that equation 27 is first order in time and second order in space, and one needs to choose $\triangle t$ proportional to $\triangle x^2$ in order to obtain an overall asymptotic accuracy of $O(\triangle x^2)$. In the numerical examples section, we chose the time step for the heat equation as either $\triangle t^H = .5\triangle x$ or $\triangle t^H = .5\triangle x^2$ depending on whether we are trying to obtain first or second order overall accuracy, respectively.

The Crank-Nicolson scheme

$$\frac{T^{n+1} - T^n}{\triangle t} = \frac{1}{2} \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) + \frac{1}{2} \nabla \cdot \left(\hat{k} \nabla T^n \right) \tag{29}$$

can be used to achieve second order accuracy in both space and time with $\triangle t$ proportional to $\triangle x$. In the numerical examples section, we choose $\triangle t^H = .5\triangle x$. For the Crank-Nicolson scheme,

$$T^{n+1} - \frac{\triangle t}{2} \nabla \cdot \left(\hat{k} \nabla T^{n+1} \right) = T^n + \frac{\triangle t}{2} \nabla \cdot \left(\hat{k} \nabla T^n \right) \tag{30}$$

is used to create a symmetric linear system of equations for the unknowns T_i^{n+1}.

3.3 Stefan Problem

3.3.1 Interface Velocity

Equation 11 is used to find the reaction speed S using the jump in $k \nabla T \cdot \vec{N}$ across the interface. Thus, accurate values of the temperature gradient are needed at grid nodes near the interface. The local unit normal $\vec{N} = \frac{\nabla \phi}{|\nabla \phi|}$ is computed using the level set method (as described in [15]).

Once T_N is defined in a band about the interface, we extrapolate the values of $(T_N)_r$ from the reacted side of the interface to the unreacted side and extrapolate the values of $(T_N)_u$ from the unreacted side to the reacted side so that both $(T_N)_r$ and $(T_N)_u$ are defined at every grid point in a band about the interface. This is accomplished with constant extrapolation in the direction normal to the interface and implemented by solving

$$I_\tau \pm \vec{N} \cdot \vec{\nabla} I = 0 \tag{31}$$

to steady state where $I_\tau = 0$. This is done separately to advect $I = (T_N)_r$ in one direction and to advect $I = (T_N)_u$ in the other direction. Instead of time marching equation 31 in fictitious time τ, a first order accurate solution to the steady state of equation 31 is obtained using the fast (velocity) extension method in [1] (which is based on the Fast Marching Method, see e.g. [26]).

Once values of both $(T_N)_r$ and $(T_N)_u$ have been defined at grid nodes near the interface, equation 11 is used to find the reaction speed S calculating $\left[k\nabla T \cdot \vec{N}\right]$ in a node by node fashion using the nodal values of $(T_N)_r$ and $(T_N)_u$.

We use the following procedure to calculate ∇T at grid points near the interface. There are four cases to consider when computing $(T_x)_{i,j,k}$: Case 1 - If $T_{i,j,k}$, $T_{i-1,j,k}$ and $T_{i+1,j,k}$ all lie on the same side of the interface, then $(T_x)_{i,j,k}$ is calculated with $T_{i,j,k}$ and either $T_{i-1,j,k}$ or $T_{i+1,j,k}$ depending on which of these two are closer to the interface as determined by the local absolute value of ϕ. Case 2 - If $T_{i,j,k}$ and $T_{i-1,j,k}$ lie on one side of the interface, and $T_{i+1,j,k}$ lies on the other side of the interface, then $(T_x)_{i,j,k}$ is calculated using $T_{i,j,k}$ and the local interface boundary condition for T as long as the distance from $\vec{x}_{i,j,k}$ to the interface is greater than $\triangle x^2$. Otherwise, $T_{i-1,j,k}$ is used in place of $T_{i,j,k}$. Case 3 - If $T_{i,j,k}$ and $T_{i+1,j,k}$ lie on one side of the interface, and $T_{i-1,j,k}$ lies on the other side of the interface, then $(T_x)_{i,j,k}$ is calculated using $T_{i,j,k}$ and the local interface boundary condition for T as long as the distance from $\vec{x}_{i,j,k}$ to the interface is greater than $\triangle x^2$. Otherwise, $T_{i+1,j,k}$ is used in place of $T_{i,j,k}$. Case 4 - If $T_{i,j,k}$ lies on one side of the interface and both $T_{i-1,j,k}$ and $T_{i+1,j,k}$ lie on the opposite side of the interface, then the two local interface boundary conditions for T are used to calculate $(T_x)_{i,j,k}$ as long as the distance between the two interface locations is greater than $\triangle x^2$. Otherwise, the problem is under-resolved and we set $(T_x)_{i,j,k} = 0$ under the assumption that there is little variance in T in between these two very close points on the interface. T_y and T_z are calculated in a similar fashion.

The level set function is evolved in time from ϕ^n to ϕ^{n+1} using nodal velocities, $\vec{W} = S\vec{N}$, and a 3rd order accurate TVD Runge-Kutta (see [15] and

[28]) time stepping method. Detailed discretizations for equation 12 and for equation 13 are given in [7]. Note that the 5th order WENO discretization from [7] is used to discretize the spatial terms in equations 12 and 13 for the numerical examples in this paper. For stability, a time step restriction of

$$\triangle t^L \left(\frac{w_1}{\triangle x} + \frac{w_2}{\triangle y} + \frac{w_3}{\triangle z} \right) \leq .5 \tag{32}$$

is used when solving equation 12 with $\vec{W} =< w_1, w_2, w_3 >$. The overall time step is chosen as $\min(\triangle t^H, \triangle t^L)$ where $\triangle t^H = .5\triangle x$ or $\triangle t^H = .5\triangle x^2$ as outlined above.

3.3.2 Ghost Cells

Equations 28 and 30 require a valid value of T^n at each grid point. As the interface moves, the grid points that cross the interface may no longer have valid values of T^n. For example, consider the solidification of water where a grid point in the water with $T^n > 273.15K$ crosses over the interface into the ice. Now that grid point is associated with the ice, but still has $T^n > 273.15K$ as opposed to a correct value of $T^n < 273.15K$. These errors seem to have been ignored by most authors and are probably negligible when the temperature is continuous across the interface. However, when the temperature (or more likely its equivalent in a related problem) is discontinuous across the interface, using this invalid value of T^n can cause significant errors.

In order to determine a ghost cell value of $T^n_{i,j,k}$ at a grid point adjacent to the interface, we use the interface boundary condition $T_I = g(\vec{x_I})$ at the closest interface location $\vec{x}_I = \vec{x}_{i,j,k} - \phi_{i,j,k}\vec{N}$. Then assuming the temperature profile is locally linear, the ghost value is defined as $T^n_{i,j,k} = T_I + \phi_{i,j,k}(T_N)_{i,j,k}$ where $(T_N)_{i,j,k}$ is the value of T_N that has already been extrapolated from the other side of the interface. That is, on the reacted side of the interface the extrapolated value of $(T_N)_u$ is used, and on the unreacted side of the interface the extrapolated value of $(T_N)_r$ is used.

Besides a valid value of T^n, equation 30 requires a valid value of $\frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^n\right)$ at each grid point implying that ghost cell values of $\frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^n\right)$ need to be defined in grid cells adjacent to the interface in case they are uncovered as the interface moves across the grid. Since a second order accurate quadratic extension of a function does not change the values of its second derivative, ghost cell values of $\frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^n\right)$ are calculated by extrapolating this term across the interface according to equation 31 with $I = \frac{\triangle t}{2} \nabla \cdot \left(\hat{k} \nabla T^n \right)$. Once again the fast extension method from [1] is used. Here, in order to get smooth values of I, an Isobaric Fix technique (see [9] and [7]) is used to extrapolate the values of I across the interface that are at least one grid cell away from the interface, as opposed to the usual procedure of extrapolating the values that are adjacent to the interface.

4 Examples

In each example, we use the level set function ϕ in order to decompose the domain into separate regions. The interior region Ω^- is defined by $\phi \leq 0$ while the exterior region Ω^+ is defined by $\phi > 0$. In each example, the L^1 and L^∞ errors are computed at the final time.

4.1 Poisson Equation

Here we consider equation 1 for cases where β piecewise constant on each subdomain or spatially varying on each subdomain. When β is constant on a subdomain, it can be moved to the right hand side rewriting equation 1 as $\triangle u = \hat{f}$ where $\hat{f} = \frac{f}{\beta}$. In this case, β can be ignored. Since Ω^- is completely decoupled from Ω^+, we only compute solutions for Ω^- here.

4.1.1 Example 1

Consider $u_{xx} = f$ on $\Omega = [-.5, .5]$ with an exact solution of $u = 4x^2 \sin(2\pi x)$ on Ω^- where $\phi = |x| - .313$ so that the interface does not lie on a grid point in our test cases. Table I shows the results of the numerical accuracy tests.

Table I - 1D Laplace Equation

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	4.422×10^{-4}	$--$	9.236×10^{-4}	$--$
81	1.132×10^{-4}	1.97	2.654×10^{-4}	1.79
161	2.736×10^{-5}	2.04	7.306×10^{-5}	1.86

4.1.2 Example 2

Consider $(\beta u_x)_x = f$ on $\Omega = [-.5, .5]$ with an exact solution of $u = e^{4x} \sin(2\pi x)$ and $\beta = \cos(x)$ on Ω^- where $\phi = |x| - .313$. Figure 1 shows the numerical solution with 61 grid points, and Table II shows the results of the numerical accuracy tests.

Table II - 1D Poisson Equation

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	1.939×10^{-2}	$--$	8.072×10^{-2}	$--$
81	5.015×10^{-3}	1.95	2.010×10^{-2}	2.01
161	1.198×10^{-3}	2.06	5.532×10^{-3}	1.86

4.1.3 Example 3

This example was taken from [19]. Consider $\triangle u = f$ on $\Omega = [-1,1]\times[-1,1]$ with an exact solution of $u = x^2 + y^2$ on Ω^-. The interface is parameterized by $(x(\theta), y(\theta))$ where $x(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\cos(\theta)$ and $y(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\sin(\theta)$ with $\theta \in [0, 2\pi]$. In order to compute ϕ, the interface was divided into 2000 equally spaced points. At each grid node, the magnitude of the signed distance function ϕ was computed using the closest point, and the sign of ϕ was computed by using the cross-product between the normal and tangent vectors to the interface so that ϕ is negative inside the closed contour. Table III shows the results of the numerical accuracy tests.

Table III - 2D Laplace Equation

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
101×101	7.329×10^{-5}	$--$	9.777×10^{-5}	$--$
201×201	1.776×10^{-5}	2.04	2.427×10^{-5}	2.01
401×401	4.714×10^{-6}	1.92	6.178×10^{-6}	1.97

4.1.4 Example 4

This example was taken from [19]. Consider $\nabla \cdot (\beta \nabla u) = f$ on $\Omega = [-1,1] \times [0,3]$ with an exact solution of $u = e^x(x^2\sin(y) + y^2)$ and $\beta = 2 + \sin(xy)$ on Ω^-. The interface is parameterized by $(x(\theta), y(\theta))$ where $x(\theta) = .6\cos(\theta) - .3\cos(3\theta)$ and $y(\theta) = 1.5 + .7\sin(\theta) - .07\sin(3\theta) + .2\sin(7\theta)$ with $\theta \in [0, 2\pi]$. ϕ is computed as in example 3. Figure 2 shows the numerical solution with 61 grid points in the x-direction and 121 grid points in the y-direction, and Table IV shows the results of the numerical accuracy tests.

Table IV - 2D Poisson Equation

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
81×121	2.414×10^{-4}	$--$	1.129×10^{-3}	$--$
161×241	6.291×10^{-5}	1.93	3.043×10^{-4}	1.87
321×481	1.707×10^{-5}	1.88	7.804×10^{-5}	1.94

4.1.5 Example 5

Consider $\triangle u = f$ on $\Omega = [0,1] \times [0,1] \times [0,1]$ with an exact solution of $u(x,y,z) = e^{-x^2-y^2-z^2}$ on Ω^- where $\phi = \sqrt{(x-.5)^2 + (y-.5)^2 + (z-.5)^2} - .3$. Table V shows the results of the numerical accuracy tests.

Table V - 3D Laplace Equation

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
$26 \times 26 \times 26$	6.394×10^{-5}	$--$	2.272×10^{-4}	$--$
$51 \times 51 \times 51$	1.635×10^{-5}	1.96	5.198×10^{-5}	2.12
$101 \times 101 \times 101$	3.997×10^{-6}	2.03	1.306×10^{-5}	1.99

4.1.6 Example 6

Consider $\nabla \cdot (\beta \nabla u) = f$ on $\Omega = [0,1] \times [0,1] \times [0,1]$ with an exact solution of $u = \sin(4\pi x)\sin(4\pi y)\sin(4\pi z)$, and $\beta = xyz$ on Ω^- where $\phi = \sqrt{(x-.5)^2 + (y-.5)^2 + (z-.5)^2} - .3$. Table VI shows the results of the numerical accuracy tests.

Table VI - 3D Poisson Equation

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
$21 \times 21 \times 21$	1.059×10^{-2}	$--$	3.690×10^{-2}	$--$
$41 \times 41 \times 41$	2.370×10^{-3}	2.16	8.989×10^{-3}	2.03
$81 \times 81 \times 81$	5.619×10^{-4}	2.03	2.170×10^{-3}	2.08

4.2 Heat Equation

Here we consider equation 4 where k is a (possibly different) constant on each subdomain. In this case, equation 4 can be rewritten as $T_t = \hat{k} \triangle T$ where $\hat{k} = \frac{k}{\rho c_v}$. In the examples below, we take $\hat{k} = 1$.

4.2.1 Example 7

Consider $T_t = T_{xx}$ on $\Omega = [-1, 1]$ with an exact solution of $T = e^{-\pi^2 t}\cos(\pi x)$ on Ω^- where $\phi = |x| - .313$. Tables VII, VIII and IX show the results of the numerical accuracy tests.

Table VII - 1D Heat Equation - Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	1.443×10^{-2}	$--$	2.222×10^{-2}	$--$
81	7.240×10^{-3}	0.99	1.118×10^{-2}	1
161	3.634×10^{-3}	0.99	5.609×10^{-3}	0.97

Table VIII - 1D Heat Equation - Backward Euler - $\triangle t \approx \triangle x^2$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	6.198×10^{-4}	$--$	8.866×10^{-4}	$--$
81	1.540×10^{-4}	1.98	2.194×10^{-4}	2.00
161	3.839×10^{-5}	2.01	5.458×10^{-5}	2.00

Table IX - 1D Heat Equation - Crank-Nicholson - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	4.084×10^{-4}	$--$	6.811×10^{-4}	$--$
81	9.907×10^{-5}	2.01	1.623×10^{-4}	2.08
161	2.424×10^{-5}	2.03	3.993×10^{-5}	2.00

4.2.2 Example 8

Consider $T_t = \triangle T$ on $\Omega = [-1,1] \times [-1,1]$ with an exact solution of $T = e^{-2t}\sin(x)\sin(y)$ on Ω^-. The interface is parameterized by $(x(\theta), y(\theta))$ where $x(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\cos(\theta)$ and $y(\theta) = .02\sqrt{5} + (.5 + .2\sin(5\theta))\sin(\theta)$ with $\theta \in [0, 2\pi]$. ϕ is computed as in example 3. Figure 3 shows the numerical solution computed with the Crank-Nicolson scheme at $t = .1$ with 81 grid points in each spatial direction. Tables X, XI and XII show the results of the numerical accuracy tests.

Table X - 2D Heat Equation - Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
81×81	1.282×10^{-5}	$--$	2.340×10^{-4}	$--$
161×161	5.618×10^{-6}	1.19	4.131×10^{-5}	2.50
321×321	2.539×10^{-6}	1.14	7.966×10^{-6}	2.37

Table XI - 2D Heat Equation - Backward Euler - $\triangle t \approx \triangle x^2$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
81×81	4.886×10^{-6}	$--$	2.340×10^{-4}	$--$
161×161	9.307×10^{-7}	2.39	4.131×10^{-5}	2.50
321×321	1.687×10^{-7}	2.46	7.569×10^{-7}	5.77

Table XII - 2D Heat Equation - Crank-Nicholson - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
81×81	5.440×10^{-6}	$--$	2.340×10^{-4}	$--$
161×161	7.888×10^{-7}	2.78	4.131×10^{-5}	2.50
321×321	1.424×10^{-7}	2.46	6.207×10^{-7}	6.05

4.2.3 Example 9

Consider $T_t = \triangle T$ on $\Omega = [0, .5] \times [0, .5] \times [0, .5]$ with an exact solution of $T = e^{-3t} \sin(x) \sin(y) \sin(z)$ on Ω^- where $\phi = \sqrt{(x-.5)^2 + (y-.5)^2 + (z-.5)^2} - .15$. Figure 4 shows the $z = .25$ cross section of the numerical solution computed with the Crank-Nicolson scheme at $t = .1$ with 41 grid points in each spatial direction. Tables XIII, XIV and XV show the results of the numerical accuracy tests.

Table XIII - 3D Heat Equation- Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
$26 \times 26 \times 26$	1.727×10^{-6}	$--$	4.129×10^{-6}	$--$
$51 \times 51 \times 51$	7.591×10^{-7}	1.19	1.937×10^{-6}	1.09
$101 \times 101 \times 101$	3.596×10^{-7}	1.08	9.524×10^{-7}	1.03

Table XIV - 3D Heat Equation- Backward Euler - $\triangle t \approx \triangle x^2$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
$26 \times 26 \times 26$	4.139×10^{-7}	$--$	1.294×10^{-6}	$--$
$51 \times 51 \times 51$	1.049×10^{-7}	1.98	2.958×10^{-7}	2.12
$101 \times 101 \times 101$	2.559×10^{-8}	2.03	7.536×10^{-8}	1.97

Table XV - 3D Heat Equation - Crank-Nicholson - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
$26 \times 26 \times 26$	5.607×10^{-7}	$--$	2.805×10^{-6}	$--$
$51 \times 51 \times 51$	7.620×10^{-8}	2.87	2.079×10^{-7}	3.75
$101 \times 101 \times 101$	2.094×10^{-8}	1.86	5.617×10^{-8}	1.88

4.3 Stefan Problem

Here we consider the Stefan problem with $k = \rho = c_v = 1$ in each subdomain. Then the temperature in each subdomain is governed by the heat equation $T_t = \triangle T$, and the interface velocity is given by

$$S = -\frac{1}{[h_o]} \left[\nabla T \cdot \vec{N} \right] \tag{33}$$

from equation 11. A Dirichlet $T = 0$ interface boundary condition is used at the interface separating the two subdomains.

4.3.1 Example 10

Let $\Omega = [0,1]$ with an exact solution of $T = e^{t-x+.5} - 1$ on Ω^- and $T = 0$ on Ω^+ where $\phi = x - .5$ at $t = 0$. Here, $[h_o] = -1$ so that the interface velocity is $S = \left[\nabla T \cdot \vec{N}\right]$. Dirichlet boundary conditions are enforced on the $\partial\Omega$ using the exact solutions. Figure 5 shows the numerical solution computed with the Crank-Nicolson scheme at $t = .25$ with 81 grid points. Tables XVI and XVII show the results of the numerical accuracy tests. Note that the Crank-Nicolson scheme is no longer second order accurate overall, due to the loss of one order of accuracy in the computed interface velocity. However, as shown in Table XVIII, the exact interface velocity can be used to obtain the expected second order accurate results (in this case the problem reduces to solving the heat equation on a time varying domain). In general, the exact interface velocity is not known, so we use the simpler backward Euler method for the two and three dimensional examples that follow. Even though the backward Euler method is only first order accurate in time, we still derive significant benefits from the symmetric second order accurate spatial discretization.

Table XVI - 1D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	7.065×10^{-4}	$--$	8.949×10^{-4}	$--$
81	3.542×10^{-4}	0.99	4.527×10^{-4}	1.01
161	1.769×10^{-4}	1.01	2.272×10^{-4}	0.98

Table XVII - 1D Stefan Problem - Crank Nicolson - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	2.372×10^{-4}	$--$	4.501×10^{-4}	$--$
81	1.129×10^{-4}	1.07	2.125×10^{-4}	1.08
161	5.388×10^{-5}	1.06	9.975×10^{-5}	1.09

Table XVIII - 1D Stefan Problem (exact interface location) - Crank Nicolson - $\triangle t \approx \triangle x$

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	2.716×10^{-5}	$--$	6.621×10^{-5}	$--$
81	6.789×10^{-6}	2.00	1.479×10^{-5}	2.16
161	1.681×10^{-6}	2.01	4.055×10^{-6}	1.87

4.3.2 Example 11

Let $\Omega = [-5,5] \times [-5,5]$ and consider the Frank sphere which is an exact solution of the Stefan problem, see for example [2]. In two spatial dimensions, the exact interface location is a disk of radius $R = s_o\sqrt{t}$ with an exact solution of $T = 0$ inside the disk and

$$T = T_\infty \left(1 - \frac{F(s)}{F(s_o)}\right) \tag{34}$$

outside the disk where $s = \frac{r}{\sqrt{t}}$, $r = \sqrt{x^2 + y^2}$, $F(s) = E_1(s^2/4)$,

$$E_1(z) = \int_z^\infty \frac{\exp(z)}{z} dz \tag{35}$$

and the value of s_o depends on the choice of T_∞, e.g. we take $T_\infty = -.5$ implying $s_o = 1.56$. Initially, $t = 1$ so that $R = 1.56$ and $\phi = \sqrt{x^2 + y^2} - 1.56$. In this example, $[h_o] = 1$ and the interface velocity is given by $S = -\left[\nabla T \cdot \vec{N}\right] = \frac{s_o}{2\sqrt{t}}$. Dirichlet boundary conditions are enforced on the $\partial\Omega$ using the exact solution. Figure 6 shows the numerical solution computed with the backward Euler scheme at $t = 1.5$ with 60 grid points in each spatial dimension plotted on top of the exact solution. Figure 7 shows the convergence of the Frank sphere solution's radius as the grid size is refined. Table XXIX and XXX show the results of the numerical accuracy tests on the temperature field and the radius, respectively. The numerical estimates for the radius were calculated using the grid points adjacent to the interface.

Table XXIX - 2D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x$
Error in the Temperature Field

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
41	1.912×10^{-3}	$--$	2.706×10^{-2}	$--$
81	9.587×10^{-4}	0.99	1.600×10^{-2}	0.75
161	4.195×10^{-4}	1.19	1.148×10^{-2}	0.47

Table XXX - 2D Stefan Problem - Backward Euler - $\triangle t \approx \triangle x$
Error in the Frank Sphere's Radius

Number of Points	$L^1 -$ error	order	$L^\infty -$ error	order
21	1.812×10^{0}	$--$	1.674×10^{-1}	$--$
41	3.105×10^{-1}	2.54	6.350×10^{-2}	1.39
81	6.013×10^{-2}	2.30	2.847×10^{-2}	1.15
161	1.396×10^{-2}	2.17	1.159×10^{-2}	1.29

4.3.3 Example 12

The following two examples illustrate our method's potential for modeling crystal growth in two and three spatial dimensions. We obtain a fair amount of detail in the dendrite structures with little computational effort, since our method yields a symmetric linear system that can be inverted efficiently. (Both $2D$ and $3D$ numerical experiments were performed on a Pentium III laptop).

Let $\Omega = [-1.5, 1.5]^2$ for the $2D$ case and $\Omega = [-1.5, 1.5]^3$ for the $3D$ case, with an initially circular (spherical) interface of radius .1. Initially, $T = 0$ inside Ω^-, and $T = -.5$ outside Ω^-. Here, $[h_o] = 1$ so the interface velocity is given by $S = -\left[\nabla T \cdot \vec{N}\right]$. Dirichlet boundary conditions of $T = -.5$ are enforced on $\partial\Omega$. The $T = -.5$ material is supercooled and the interface grows outward in an unstable fashion as shown in figures 8 and 9. Of course, we expect anisotropic effects due to the grid, since the interface is unstable in this supercooled example.

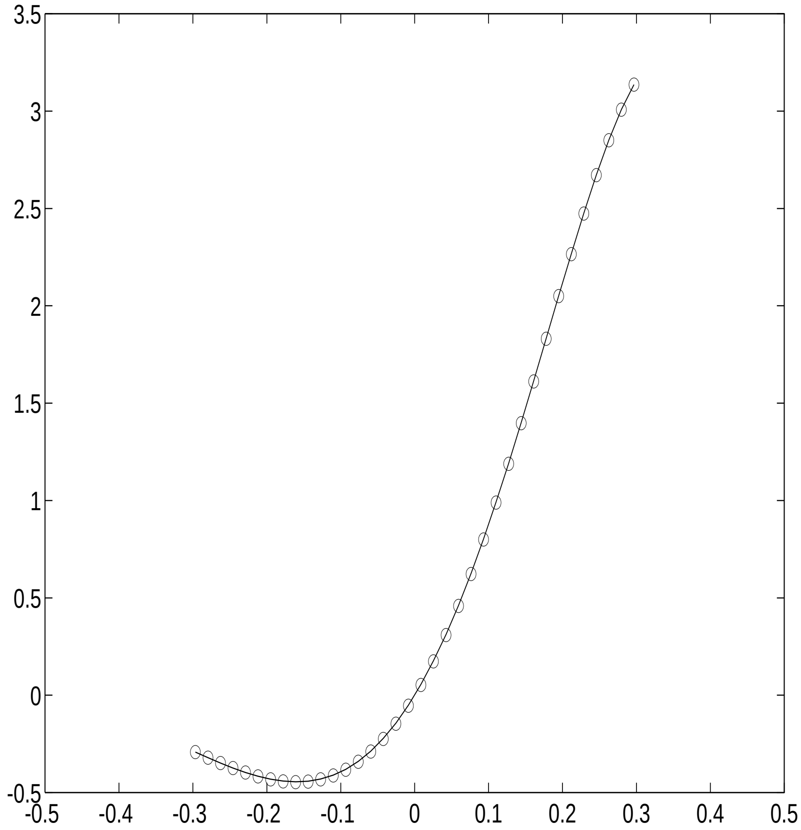

Figure 1: 1D Poisson equation, $(\beta u_x) = f$, on $[-.313, .313]$ with Dirichlet boundary conditions. The circles are the computed solution, and the solid line is the exact solution.

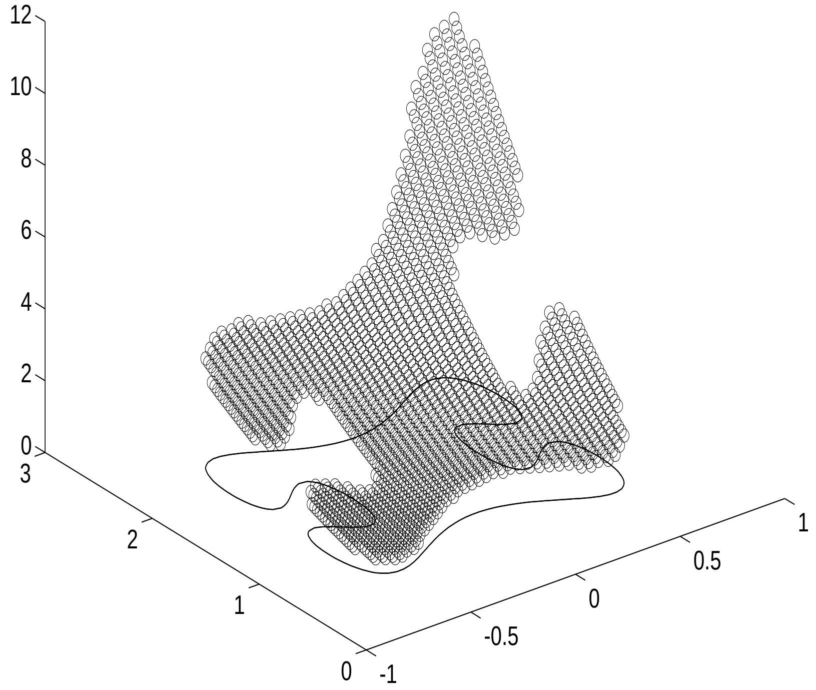

Figure 2: 2D Poisson equation, $\nabla \cdot (\beta \nabla u) = f$, with Dirichlet boundary conditions. The circles are the computed solution, and the solid line contour outlines the irregularly shaped computational domain.

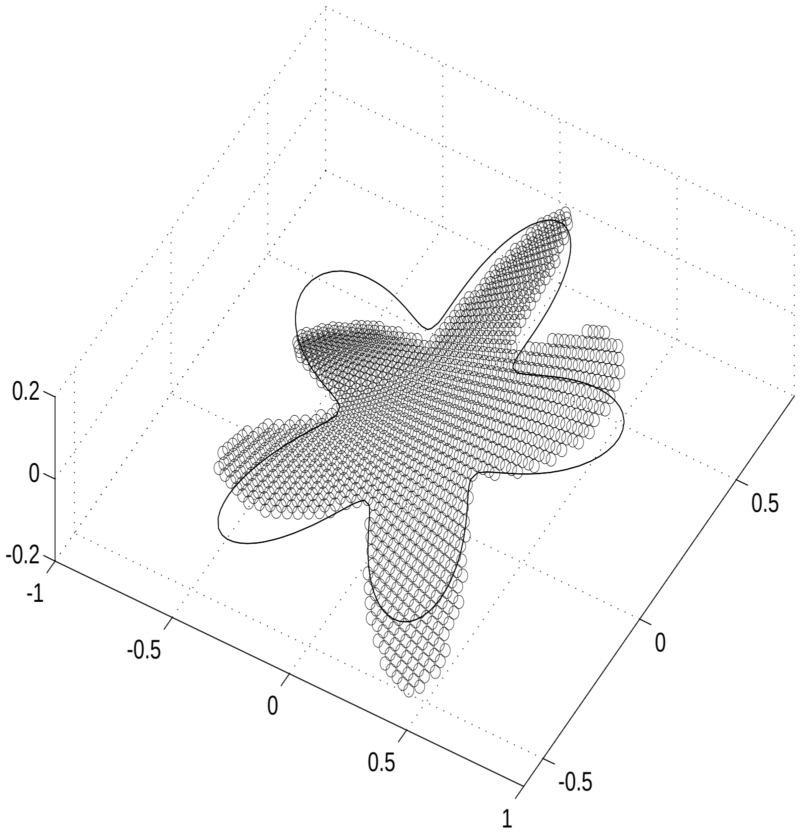

Figure 3: 2D heat equation, $T_t = \triangle T$, with Dirichlet boundary conditions. The circles are the computed solution, and the solid line contour outlines the irregularly shaped computational domain.

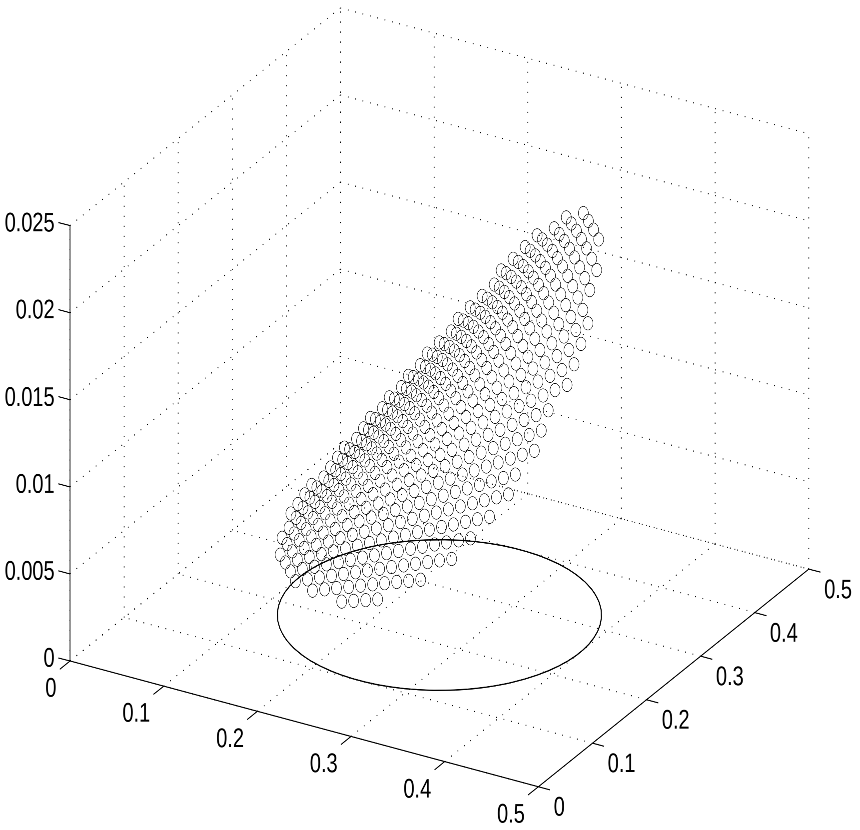

Figure 4: 3D heat equation, $T_t = \triangle T$, with Dirichlet boundary conditions. The lower dimensional $z = .25$ cross section of the solution is shown. The circles are the computed solution, and the solid line contour outlines the computational domain.

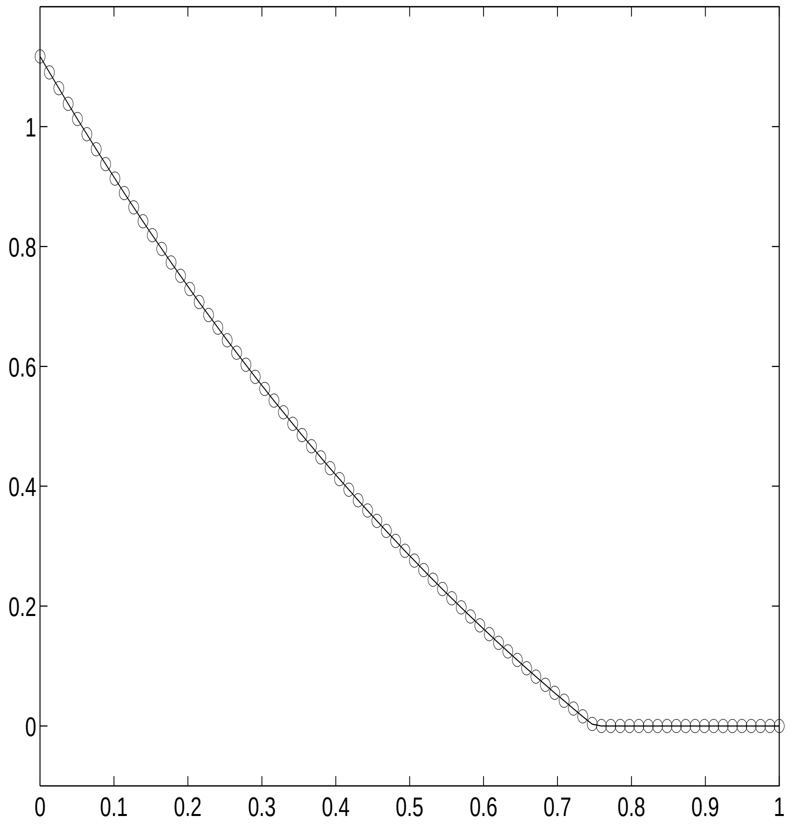

Figure 5: 1D Stefan problem. The circles are the computed solution, and the solid line is the exact solution. The interface is currently located near $x = .75$ where the temperature profile has a kink.

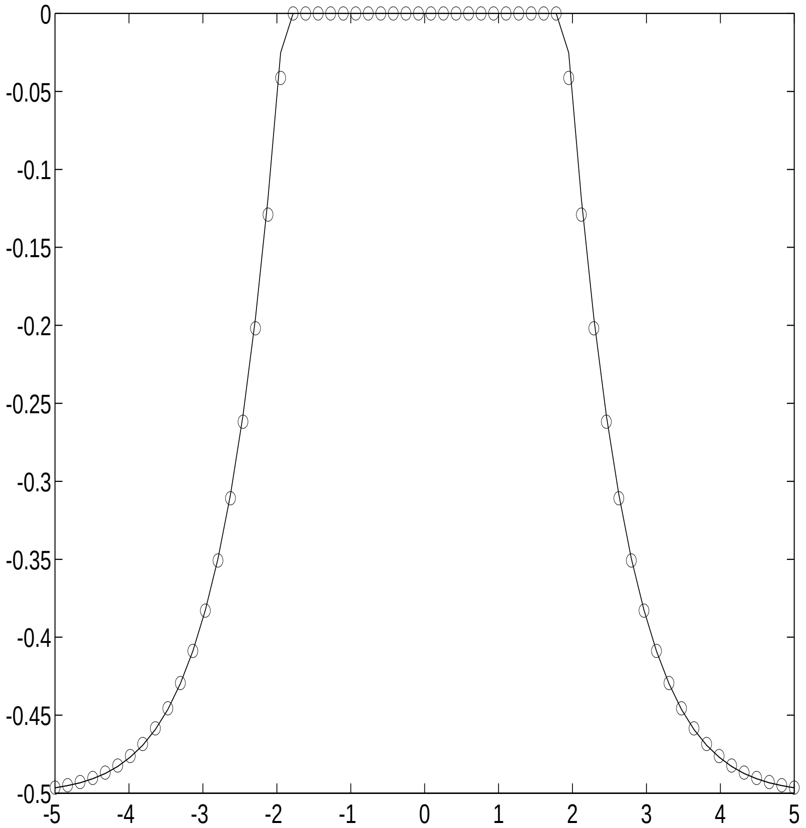

Figure 6: 2D Stefan problem. The lower dimensional $y = 0$ cross section of the solution is shown. The circles are the computed solution, and the solid line is the exact Frank sphere solution. The two interface points in this cross section are located near $x = \pm 2$ where the temperature profile is kinked.

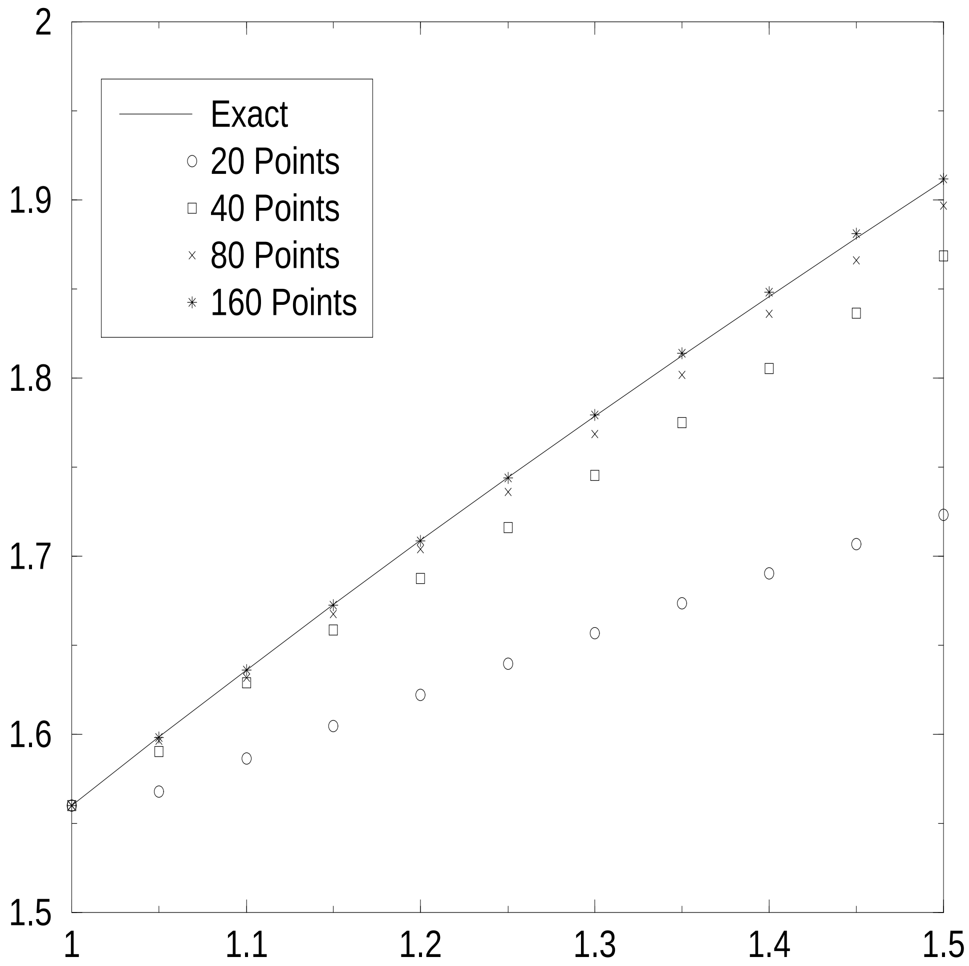

Figure 7: 2D Stefan problem. The graph shows the growing interface radius as a function of time. The exact Frank sphere solution is plotted as a solid line, and the computed solutions are shown for four different grids. The computed results clearly converge to the exact solution.

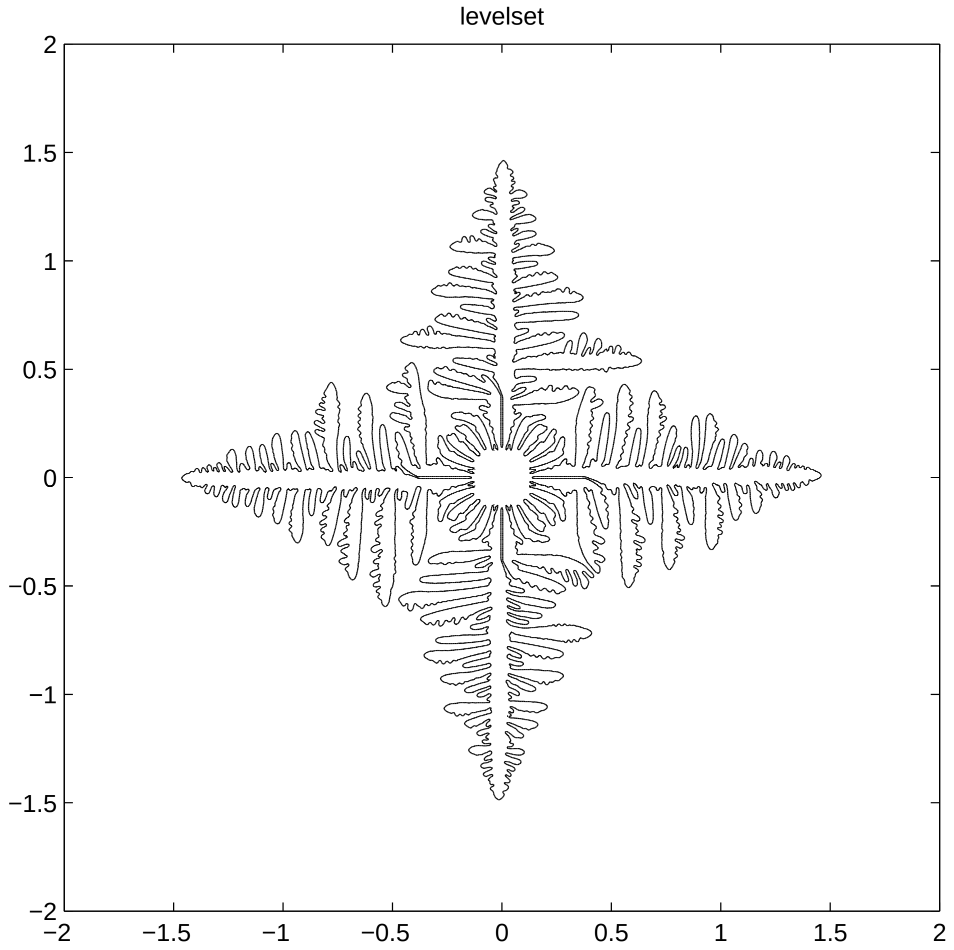

Figure 8: 2D Stefan problem. The contour shows the interface location at $t = .15$. This computation uses 1000 grid points in each spatial direction. The supercooled material in the exterior region promotes unstable growth.

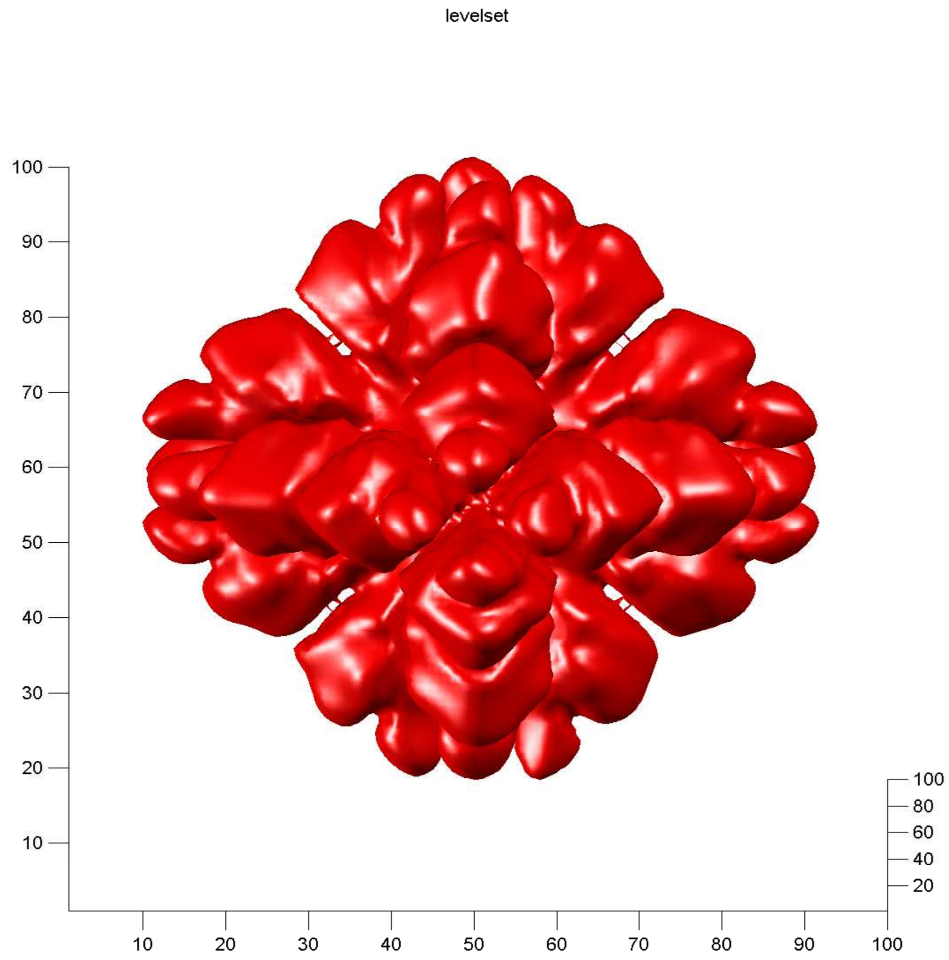

Figure 9: 3D Stefan problem. The contour shows the interface location at $t = .14$. This computation uses 100 grid points in each spatial direction. The supercooled material in the exterior region promotes unstable growth.

5 Conclusions

We have proposed a very simple algorithm for obtaining second order accurate solutions to the variable coefficient Poisson equation with Dirichlet boundary conditions on irregular domains. Our discretization of the problem produces a symmetric matrix which is straightforward and computationally inexpensive to invert. Current second order accurate state of the art algorithms are more complicated to apply *and* produce nonsymmetric discretization matrices which are more costly to invert. While some authors claim that they can efficiently invert nonsymmetric matrices, it makes little sense to use a discretization that is both more complicated *and* produces a nonsymmetric matrix when one can use a simpler discretization *and* obtain a symmetric matrix.

With a number of numerical examples, we have demonstrated that second order accurate numerical results are readily obtained on reasonable grids. In addition, we have shown how the proposed symmetric discretization can be applied to an implicit time discretization of the heat equation on an irregular domain to obtain second order accurate results there as well. Lastly, we addressed the Stefan problem where second order accurate results can only be obtained if the interface velocity is known to second order accuracy. Unfortunately, the interface velocity depends on derivatives of the temperature which are one order less accurate than the temperature itself producing overall first order accurate results. However, our method still lowers the truncation error in this case as the decoupled heat equation step can be solved with second order accuracy. Our symmetric discretization enabled us to carry out numerical experiments in three spatial dimensions in a reasonable amount of time without the need for a parallel implementation of the code.

6 Acknowledgment

We would like to thank Chris Anderson and Susan Chen for fruitful discussions.

References

[1] Adalsteinsson, D. and Sethian, J., *The Fast Construction of Extension Velocities in Level Set Methods*, J. Comput. Phys. 148, 2-22 (1999).

[2] Almgren, R., *Variational Algorithms and Pattern Formation in Dendritic Solidification*, J. Comput. Phys. 106, 337-354 (1993).

[3] Atkins, P., *Physical Chemistry*, 5th edition, Freeman, 1994.

[4] Beyer, R. and LeVeque, R., *Analysis of a One-Dimensional Model for the Immersed Boundary Method*, SIAM J. Num. Anal. 29, 332-364 (1992).

[5] Chen, S., Merriman, B., Osher, S. and Smereka, P., *A Simple Level Set Method for Solving Stefan Problems*, J. Comput. Phys. 135, 8-29 (1997).

[6] Fedkiw. R., *A Symmetric Spatial Discretization for Implicit Time Discretization of Stefan Type Problems*, (unpublished) June 1998.

[7] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., *A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)*, J. Comput. Phys. 152, 457-492 (1999).

[8] Fedkiw, R., Aslam, T., and Xu, S., *The Ghost Fluid Method for Deflagration and Detonation Discontinuities*, J. Comput. Phys. 154, 393-427 (1999).

[9] Fedkiw, R., Marquina, A., and Merriman, B., *An Isobaric Fix for the Overheating Problem in Multimaterial Compressible Flows*, J. Comput. Phys. 148, 545-578 (1999).

[10] Golub, G. and Van Loan, C., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.

[11] Johansen, H., *Cartesian Grid Embedded Boundary Finite Difference Methods for Elliptic and Parabolic Differential Equations on Irregular Domains*, Ph.D. Thesis, University of California, Berkeley, CA 1997.

[12] Johansen, H. and Colella, P., *A Cartesian Grid Embedded Boundary Method for Poisson's Equation on Irregular Domains*, J. Comput. Phys. 147, 60-85 (1998).

[13] Jones, W. and Menzies, K., *Analysis of the Cell-Centered Finite Volume Method for the Diffusion Equation*, J. Comput. Phys. 165, 45-68 (2000).

[14] Juric, D. and Tryggvason, G., *A Front-Tracking Method for Dendritic Solidification*, J. Comput. Phys. 123, 127-148 (1996).

[15] Kang, M., Fedkiw, R., and Liu, X.-D., *A Boundary Condition Capturing Method for Multiphase Incompressible Flow*, J. Sci. Comput. 15, 323-360 (2000).

[16] Karma, A. and Rappel, W.-J., *Quantitative Phase Field Modeling of Dendritic Growth in Two and Three Dimensions*, Physical Review E 57, 4323-4349 (1998).

[17] Kim, Y.-T., Goldenfeld, N. and Dantzig, J., *Computation of Dendritic Microstructures using a Level Set Method*, Physical Review E 62, 2471-2474 (2000).

[18] LeVeque, R. and Li, Z., *The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources*, SIAM J. Num. Anal. 31, 1019-1044 (1994).

[19] Li, Z., *A Fast Iterative Algorithm for Elliptic Interface Problems*, SIAM J. Numer. Anal. 35, 230-254 (1998).

[20] Liu, X.-D., Fedkiw, R. and Kang, M., *A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains*, J. Comput. Phys. 154, 151-178 (2000).

[21] Nguyen, D., Fedkiw, R. and Kang, M., "A Boundary Condition Capturing Method for Incompressible Flame Discontinuities", J. Comput. Phys. 172, 71-98 (2001).

[22] Osher, S. and Sethian, J., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, J. of Comput. Phys. 79, 12-49 (1988).

[23] Peskin, C., *Numerical Analysis of Blood Flow in the Heart*, J. Comput. Phys. 25, 220-252 (1977).

[24] Plapp, M. and Karma, A., *Multiscale Finite-Difference-Diffusion-Monte-Carlo Method for Simulating Dendritic Solidification*, J. Comput. Phys. 165, 592-619 (2000).

[25] Schmidt, A., *Computation of Three Dimensional Dendrites with Finite Elements*, J. Comput. Phys. 125, 293-312 (1996).

[26] Sethian, J., *Fast Marching Methods*, SIAM Review 41, 199-235 (1999).

[27] Sethian, J. and Strain, J., *Crystal Growth and Dendritic Solidification*, J. Comput. Phys. 98, 231-253 (1992).

[28] Shu, C.W. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes*, J. Comput. Phys. 77, 439-471 (1988).

[29] Sussman, M., Smereka, P. and Osher, S., *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow*, J. Comput. Phys. 114, 146-154 (1994).

[30] Udaykumar, H., Mittal, R. and Shyy, W., *Computation of Solid-Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids*, J. Comput. Phys. 153, 535-574 (1999).