Journal of Computational Physics **174,** 695–711 (2001)
doi:10.1006/jcph.2001.6933, available online at http://www.idealibrary.com on IDEAL®

Nonlinear Preconditioning for Diffuse Interfaces

Karl Glasner[1]

Department of Mathematics, University of Utah, Salt Lake City, Utah 84112-0090
E-mail: glasner@math.duke.edu

Received April 11, 2000; revised August 23, 2001

A method of transforming problems with diffuse interfaces is presented which leads to equations that are easier to compute accurately. Information obtained by internal layer asymptotic analysis is utilized to motivate transformations of the dependent variables. The new evolution equations which result from this change of variables can be solved numerically in a straightforward manner. Numerical experiments indicate that truncation errors can be significantly reduced in such problems, allowing a coarser grid to be used. Applications to several well-known models are presented.

Key Words: diffuse interface methods; preconditioning.

1. INTRODUCTION

Numerous physical models have solutions which develop diffuse interfaces, narrow spatial structures where the solution or its derivatives vary rapidly, but continuously, between two (or more) values. This rapid variation naturally leads to difficulty in numerical computation. In particular, very fine grids must be employed to adequately resolve the structure of solutions around the diffuse interface.

Associated with diffuse interfaces is a small scale ϵ, proportional to the width of the interface. This scale frequently appears explicitly as a parameter in the model equations. The limit as $\epsilon \to 0$ can usually be analyzed systematically by asymptotic expansions [10]. The output of this analysis is a description of the motion of the limiting interface, which is typically some free boundary problem. The fine structure of the solution is also produced as a by-product of the analysis; it is this information we will utilize to improve computational results.

Most approaches to solving such equations numerically lead to the conclusion that the grid refinement near the interface needs to be much less than the width ϵ if solutions are to be computed accurately. This leads to very large grid sizes, especially in higher dimensions, and makes computing interesting problems prohibitive.

[1] Present address: Department of Mathematics, Duke University, Durham, NC 27708-0320.

0021-9991/01 $35.00

One way of ameliorating this problem is to adaptively refine the grid only near the transition layer (refs. [4, 5, 18, 19] are examples of this). This can lead to more efficient computation, but at the expense of complicating the implementation of the underlying numerical method.

Another approach is solve the sharp interface free boundary problem derived from the asymptotic analysis as an alternative to the diffuse interface model (some examples are [14, 22]). Difficulties arise, however, during topological changes, when interfaces break apart or merge together. Frequently ad hoc approaches have been needed to handle these cases. As a result, numerical algorithms can be exceedingly complicated.

This paper explores yet another way of overcoming the problem of requiring very fine grids near interfaces. Very specific information about solutions near diffuse interfaces is available from the asymptotic analysis for small interface widths. This information may be used in turn to make intuitively motivated transformations of the equations. In this paper the transformations take the form of simple changes of variables, but nonlocal transformations could also be advantageous. On the basis of numerical experiments, we conclude that a substantial gain in accuracy is obtained from computing the equations in terms of the transformed variables instead. An important point is that this technique is not sensitive to the type of numerical discretization employed, nor is it mutually exclusive of efficiency-improving algorithms such as adaptive mesh refinement.

The idea of transforming problems to make them easier to compute is not new, of course. Applying a Fourier transform to the usual heat equation, for example, may yield a trivial set of uncoupled ordinary differential equations. Preconditioning of linear systems is another situation where a transformation can make problems significantly easier to solve; it can, for example, greatly accelerate the convergence of iterative methods. These examples serve as motivation for finding transformations for any problem which makes it easier to solve. Fundamental to this process is incorporating analytical information already known about the problem. This information need not be complete, but only needs to, in some sense, get us "closer" to knowing the solution. In the case of diffuse interface phenomena, asymptotic information for small interface width is typically available.

This paper considers several examples of diffuse interface models, and shows how simple, well-motivated transformations may improve the outcome of numerical simulation. Section 2 reviews a simple example which motivates the transformations considered in the rest of the paper. Section 3 reviews a prototypical diffuse interface problem, the Cahn–Allen equation, and shows how a simple change of variables may substantially reduce error in numerical computation of this equation. A further application to phase field models is made in Section 4. We conclude by commenting on some directions for future research.

2. A SIMPLE EXAMPLE

A simple equation possessing diffuse interface solutions is the bistable reaction diffusion equation

$$\phi_t = \phi_{xx} + f(\phi), \quad x \in \mathbb{R}, \tag{1}$$

which arises in many contexts (ecology, flame front propagation, and dynamical phase transitions, to name a few). For the sake of this example, we take

$$f(\phi) = 2\phi(1-\phi^2) + \mu(1-\phi^2), \tag{2}$$

where μ is some parameter with $|\mu| < 1$. Unlike other examples considered in this paper, the interface width does not appear explicitly, but we will show how a simple transformation can lead to huge gains in computational accuracy.

A remarkable fact about (1) is that it admits the explicit traveling wave solution

$$\phi(x, t) = \tanh(x + \mu t).$$

Furthermore, it has been shown that a large class of initial data converge exponentially in time to a translate of this solution [11]. The difficulty in computing this type of equation is now somewhat apparent: because of the exponential decay of the tanh function, if the computational grid is too coarse, ϕ will only have values very close to ± 1. This simply does not provide enough information for the interface to propagate correctly. In fact, on suitably coarse grids, solutions will get stuck in artificial steady states, a phenomenon known as "pinning" (see [17] for a proof of this).

Instead of computing ϕ, we can make an obvious change of variables

$$\psi = \tanh^{-1}(\phi).$$

Then, ψ solves the evolution equation

$$\psi_t = \psi_{xx} + \mu + 2\tanh(\psi)\left(1 - \psi_x^2\right). \tag{3}$$

Suppose now that $\{x_i\}$ is some infinite grid with spacing $\Delta x_i = x_{i+1} - x_i$, and let Δt be a discrete time step. Using second-order accurate difference formulas in space and explicit time differencing, Eq. (3) may be discretized as

$$\begin{aligned}
\frac{\psi_{i,j+1} - \psi_{i,j}}{\Delta t} &= L + \mu + 2\tanh(\psi_{i,j})(1 - G^2), \\
G &= \frac{\Delta x_{i-1}^2 \psi_{i+1,j} - \Delta x_i^2 \psi_{i-1,j} + \left(\Delta x_i^2 + \Delta x_{i-1}^2\right)\psi_{i,j}}{(\Delta x_{i-1} + \Delta x_i)\Delta x_{i-1}\Delta x_i} \\
L &= \frac{\Delta x_{i-1}\psi_{i+1,j} + \Delta x_i \psi_{i-1,j} - (\Delta x_{i-1} + \Delta x_i)\psi_{i,j}}{(\Delta x_{i-1} + \Delta x_i)\Delta x_{i-1}\Delta x_i/2},
\end{aligned}$$

where $\psi_{i,j} \approx \psi(x_i, j\Delta t)$ is a discrete representation of ψ.

Now suppose "traveling wave" initial data $\psi_{i,0} = x_i - x_0$. The discrete solution may be written down explicitly in this case,

$$\psi_{i,j} = \psi_{i,0} + \mu j \Delta t,$$

or in terms of the original functions

$$\phi_{i,j} = \tanh(x_i + \mu j \Delta t).$$

In other words, there is no error at all from discretizing the equation! This was somewhat lucky since the original equation was chosen to have a nice, explicit solution, and we chose very specific initial data. Nevertheless, the rest of the paper shows that this sort of transformation can have advantages in more elaborate settings.

3. THE ALLEN–CAHN MODEL

A well-known diffuse interface description for isothermal phase transitions is the Allen–Cahn model [2],

$$\phi_t = \Delta\phi + \epsilon^{-2} f(\phi), \tag{4}$$

where

$$f(\phi) = 2\phi(1 - \phi^2),$$

which is a multidimensional version of the reaction diffusion equation above with $\mu = 0$. A Neumann boundary condition will be imposed when necessary.

Many key ingredients of asymptotic analysis of diffuse interfaces can be seen in this example. The behavior of solutions in the limit $\epsilon \to 0$ is well known [10, 20], and we shall review it here. After a small transient time interval, the solution develops narrow layers connecting the two stable states ± 1. This suggests using a matched asymptotic approach, where an outer, slowly varying solution is sought away from the layer which must smoothly connect to a rapidly varying inner solution.

For the outer solution, one assumes a regular expansion in powers of ϵ of the form

$$\phi = \phi_0 + \epsilon\phi_1 + \epsilon^2\phi_2 + \cdots.$$

Inserting this into (4) and equating terms with like powers of ϵ, we find that $\phi_0 = \pm 1$ and $\phi_1 = 0$ (in fact, all further orders of ϕ are zero; only transcendental corrections are present). The values of ϕ_0 must match the inner solution, which we describe next.

The inner solution corresponding to the interfacial layer is then found by a similar asymptotic procedure. All quantities which depend on ϵ are expanded in a series as before, such as

$$\phi = \Phi_0 + \epsilon\Phi_1 + \epsilon^2\Phi_2 + \cdots.$$

Quantities which depend on the zero level set $\Gamma = \{x \mid \phi(x) = 0\}$, such as normal velocity and curvature, also require the same regular expansion.

To account for the possibility of a curved interface, a moving, orthogonal coordinate system (r, s) is used, where r is the signed distance from the level set $\Gamma = \{x \mid \phi(x) = 0\}$ and s is the distance along the interface, or in higher dimensions a parameterization of the level set surface which preserves distance. The coordinate system is oriented so that $r > 0$ corresponds to $\phi > 0$. The normal coordinate r is rescaled using $z = \epsilon^{-1}r$ reflecting the fact that ϕ varies rapidly near the interface.

In the new, scaled coordinate system, Eq. (4) takes the form

$$\epsilon^{-1} r_t \phi_r = \epsilon^{-2}\phi_{zz} + \epsilon^{-1}\Delta r \phi_z + \epsilon^{-2} f(\phi) + O(1). \tag{5}$$

To leading order, $\Delta r = \kappa_0$, the curvature of Γ, where the convention is used that a convex region of $\phi < 0$ has positive curvature. Also, to leading order $r_t = -V_0$, the normal velocity of Γ. Note that V is defined so that it is positive when the interface moves from left to right in the r coordinate system.

To lowest order in ϵ, we obtain

$$(\Phi_0)_{zz} + f(\phi) = 0,$$

which has the monotone increasing solution

$$\Phi_0 = \tanh(z) = \tanh\left(\frac{r}{\epsilon}\right), \tag{6}$$

which is the profile of the transition layer. This is the information we will utilize in making a beneficial change of variables in Section 3.1. For completeness, we describe the remainder of the analysis.

By equating terms of order ϵ^{-1} in (5), we get the equation

$$\left[\frac{d^2}{dx^2} + f'(\Phi_0)\right] \Phi_1 = (\kappa_0 + V_0)(\phi_0)_z.$$

This linear equation only has solutions when the right-hand side is orthogonal to functions in the null-space of the self-adjoint operator which appears on the left. Noting that $(\phi_0)_z$ is in the kernel, we multiply the equation by it, and after integration by parts we derive the so-called solvability condition

$$V_0 = -\kappa_0.$$

The leading order motion of Γ is therefore a well-known interface problem, namely motion by mean curvature [8, 9, 12].

3.1. Nonlinear Transformation

Suppose that (ν, μ) is a fixed, orthogonal coordinate system which preserves lengths, where ν is normal and μ tangential to the interface in two dimensions. Because of (6), local in time and space solutions will have the approximate traveling wave behavior

$$\phi \sim \tanh\left(\frac{\nu - V(\mu, t)t}{\epsilon}\right), \tag{7}$$

where $\nu = Vt$ locates the center of the diffuse interface.

The error due to discretizations of differential operators are typically proportional in magnitude to higher order derivatives of the actual solutions. For example, the usual second-order discretization of the Laplacian with a grid spacing of size Δx would lead to an error proportional to the fourth derivative of ϕ. Thus, the error in computing a solution such as (7) potentially has a magnitude of $(\Delta x)^2\epsilon^{-4}$, requiring that $\Delta x \ll \epsilon^2$ for solutions to be computed accurately. We will see that in practice the error is not this bad, but is still a major problem unless $\Delta x \ll \epsilon$. It is easy to see that similar problems can occur when the time derivative is discretized.

As an important note, higher order methods would likely be of little help for the following reason. The function tanh is not analytic on the whole real line and has higher order derivatives which grow factorially, canceling much of the gain from higher powers of Δx in the error term (this is in fact true of any exponentially decaying interface-like solution).

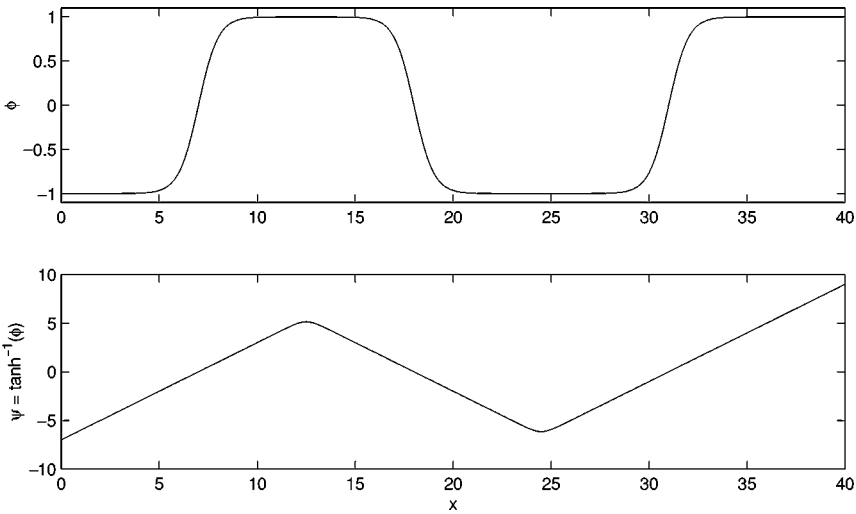

FIG. 1. The upper graph is a typical cross section of ϕ showing diffuse interfaces at $x = 7, 18, 31$ for width $\epsilon = 1$. The transformed function ψ is shown below. The only places ψ has large higher order derivatives is along the "ridges" between the diffuse interface positions.

Similar to the transformation above, we introduce the following $1-1$ change of variables:

$$\psi = \epsilon \tanh^{-1}(\phi). \tag{8}$$

From the asymptotic analysis, notice that $\psi \approx r$. In other words, ψ is a good approximation to a "signed distance function" to the zero level set and $|\nabla\psi| \approx 1$. Then by (7), ψ is approximately

$$\psi \sim \nu - V(\mu, t)t.$$

The advantage is that higher order derivatives of ψ are now completely independent of ϵ, and therefore one can expect discretization errors to be smaller.

When multiple interfaces or domain boundaries are present, ψ may posses "ridges" where two signed distance functions collide and first derivatives change rapidly (see Fig. 1). This may introduce large truncation errors near these points, but if $|\psi|$ is large along these ridges, these errors would result only in transcendentally small changes in ϕ itself, which should have virtually no consequence.

The evolution of the transformed variable ψ is now

$$\psi_t = \Delta\psi + 2\epsilon^{-1} \tanh\left(\frac{\psi}{\epsilon}\right)(1 - |\nabla\psi|^2). \tag{9}$$

Without the Laplacian term, the equation is hyperbolic and versions of it have appeared in the level set method literature [21] as a way of "reinitializing" a function whose zero level set designates some interface. The reinitialization equation, when computed to a steady state, ensures that ψ is exactly a signed distance function. In our case, since the time scale of this term is $\mathcal{O}(\epsilon)$, we are assured that to a good approximation, ψ is a signed distance function.

As a remark, it is interesting that the interface motion is readily apparent from Eq. (9). It is known that the Laplacian of a signed distance function is the mean curvature of each

of its corresponding level sets at each point. Thus, where $\psi = 0$, the normal velocity of the zero level set Γ is

$$V = -\frac{\psi_t}{|\nabla \psi|} \approx -\psi_t = -\Delta \psi = -\kappa,$$

which is motion by mean curvature as expected.

Finally, it should be mentioned that Eq. (9) is still strictly parabolic just as the original Eq. (4). Consequently, computation of this equation can be effectively handled with standard finite difference techniques, for example. Additionally, issues concerning numerical stability are the usual ones for parabolic equations.

3.2. Experimental Comparison

We now test the ideas of the preceding section to show that the proposed transformation is indeed beneficial. We discretize the two-dimensional version of equation (9) in the most obvious manner, by replacing spatial derivatives with their usual second-order accurate representations on a spatially uniform grid indexed by the pair (i, j),

$$\frac{d}{dt}\psi_{i,j} = \frac{\psi_{i-1,j} + \psi_{i+1,j} + \psi_{i,j-1} + \psi_{i,j+1} - 4\psi_{i,j}}{\Delta x^2} + 2\epsilon^{-1} \tanh\left(\frac{\psi_{i,j}}{\epsilon}\right) \times \left(1 - \frac{(\psi_{i+1,j} - \psi_{i-1,j})^2 + (\psi_{i,j+1} - \psi_{i,j-1})^2}{(2\Delta x)^2}\right),$$

where Δx is the grid spacing. The Neumann boundary condition is handled by reflecting the grid across the boundary. Since the resulting system is stiff, an implicit time stepping scheme is desirable. For most of the experiments shown here, however, the time step is small to enough to permit an explicit step. When an implicit method was used, the equations were solved by a straightforward Newton's method. The linear algebra was handled by a GMRES algorithm using a SOR preconditioner. For comparison, the original Eq. (4) was also discretized in the same manner.

3.3. Spatial Truncation Error

To compare the errors introduced by the spatial discretizations, identical initial data was propagated for a short time and the final answers were compared. The spatial domain was taken to be the box $[0, .5] \times [0, .5]$ with reflective (Neumann) boundary conditions, and the initial data was a circular "seed"

$$\phi(x, y) = \tanh\left(\frac{.25 - \sqrt{x^2 + y^2}}{\epsilon}\right),$$

which was propagated forward up to time $t = 5 \times 10^{-3}$, in all cases using an explicit time step of $\Delta t = 5 \times 10^{-6}$. For each value of ϵ, an essentially converged solution ϕ^* was computed for the untransformed Eq. (4) using a grid spacing 20 times smaller than ϵ. This solution was compared to computations with coarser grids by computing a discrete approximation to the L_2 error,

$$Error = \left(\sum_{i,j} (\phi_{i,j} - \phi^*_{i,j})^2 (\Delta x)^2\right)^{1/2},$$

TABLE I
Comparison of Spatial Truncation Errors for the Original and Transformed Equations for $\epsilon = .04$

Δx	Original	Transformed
0.08	*	2.1×10^{-3}
0.04	*	5.7×10^{-4}
0.02	2.6×10^{-3}	1.3×10^{-4}
0.01	5.9×10^{-4}	2.2×10^{-5}
0.005	1.3×10^{-4}	2.1×10^{-5}

Note. $(*)$ denotes that pinning occurred.

or for the transformed equation

$$Error = \left(\sum_{i,j} \left[\tanh\left(\frac{\psi_{i,j}}{\epsilon} \right) - \phi^*_{i,j} \right]^2 (\Delta x)^2 \right)^{1/2}.$$

Here, $\phi^*_{i,j}$ is a cubic spline interpolation of ϕ^* evaluated at the corresponding grid points.

Tables I and II show the results for both equations. Second-order convergence is seen in all cases as the grid is refined, but for equal resolutions, the transformed equation yielded smaller errors, sometimes by two orders of magnitude. The only discrepancy occurred when computing the finest resolutions of the transformed equation, likely because the true spatial truncation error was comparable to that of the nearly converged solution ϕ^*.

When Δx was too large, solutions of the original untransformed equation became pinned, that is they did not move after a short period of time. We found that this occurs in our transformed equation also, but only when Δx is many times larger than ϵ.

3.4. Approximation to Mean Curvature Flow

Frequently, equations with diffuse interfaces are used only to approximate their corresponding sharp-interface motion. It is therefore interesting to compare numerical solutions of the intended dynamics of the interface.

The computational experiment we used was identical to the previous one, except that the equation was propagated out to $t = 10^{-2}$. Since the initial data is a circle, the radius R of

TABLE II
Spatial Truncation Errors for $\epsilon = .02$

Δx	Original	Transformed
0.04	*	3.2×10^{-3}
0.02	*	1.4×10^{-4}
0.01	3.2×10^{-3}	2.3×10^{-5}
0.005	7.4×10^{-4}	2.5×10^{-5}
0.0025	1.5×10^{-4}	—

the circular interface should shrink according to

$$\frac{dR}{dt} = -\frac{1}{R}; \quad R(0) = \frac{1}{4},$$

which has the solution

$$R(t) = \sqrt{\left(\frac{1}{4}\right)^2 - 2t}.$$

The radius of the computed interface $r(t)$ was calculated by finding the zero of ψ or ϕ along the x-axis using cubic interpolation. It was checked that this only resulted in a negligible, oscillating error when the interface velocity $r'(t)$ was computed by finite

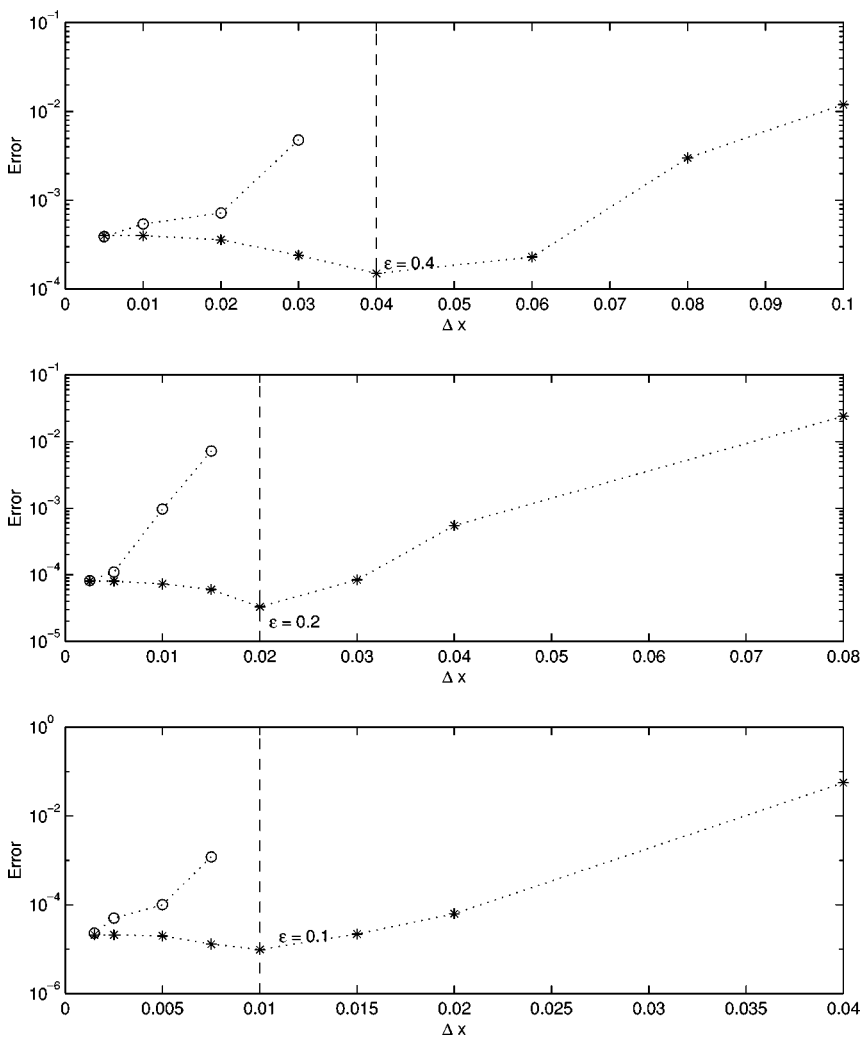

FIG. 2. Error between exact dynamics for mean curvature flow of a circular seed and computation of the original equation (circles) and the transformed equation (∗). Note that pinning ($V \to 0$) occurred in the original version when the grid refinement approached ϵ.

differences. An error defined by

$$\text{Error} = \int_0^{10^{-2}} |r'(t) + 1/r(t)|\, dt$$

was used to compare the computed solution to mean curvature flow.

Figure 2 displays the results. The transformed equation is again more accurate, but grid refinement beyond $\Delta x \approx \epsilon$ did not improve the results. This suggests that the numerical error at this point is comparable to the asymptotic error, and as a consequence further refinement is not useful. Surprisingly, the transformed version had less error when $\Delta x = \epsilon$ than it did with greater refinements, probably due to a cancellation between asymptotic and numerical error.

4. PHASE FIELD EQUATIONS

Various nonisothermal versions of the Allen–Cahn model known as phase field equations have become a popular way of simulating crystalline interface motion as well as other phase transition processes. The version we will examine is similar to that proposed by Karma and Rappel [15] and Almgren [3],

$$\frac{\epsilon\mu + \beta}{d_0}\phi_t = \Delta\phi + \epsilon^{-2} f(\phi, u) \tag{10}$$

$$(u - \phi/2)_t = \Delta u, \tag{11}$$

where we take f to have the form

$$f(\phi, u) = 2\phi(1 - \phi^2) - \frac{5}{4}\frac{\epsilon}{d_0} u(1 - \phi^2)^2.$$

The variable u typically represents temperature, and its derivatives may vary significantly across the diffuse interface. Therefore, it will be important to consider transformations involving both variables.

4.1. Asymptotic Analysis

The analysis for this equation proceeds along similar lines as the Allen–Cahn equation (see refs. [6, 15] for more detail). The variable u is also expanded in powers of ϵ for both inner and outer regions. In the outer regions, each order of u just solves a heat equation since ϕ is constant there:

$$(u_0)_t = (u_0)_{xx}, \quad (u_1)_t = (u_1)_{xx}, \ldots$$

It is necessary to match the inner and outer expansions for u by equating them on some intermediate scale. These matching conditions take the form of boundary conditions for the inner solution. We shall not need them to describe our transformation method, and therefore their description will be omitted.

To leading order, the inner solution for ϕ has the same form as before $\Phi_0(z) = \tanh(z)$. The inner solution for u, which is designated by the capital U_0, is simply

$$U_0 = \text{const.}$$

The value of this constant is determined by a solvability condition at a higher asymptotic order.

To next order, we get the linear equation

$$\left\{\frac{d^2}{dz^2} + f_\phi(\Phi_0, 0)\right\}\Phi_1 = \left(\kappa_0 - \frac{\beta}{d_0}V_0\right)(\Phi_0)_z + \frac{5}{4}\frac{U_0}{d_0}\left(1 - \Phi_0^2\right)^2.$$

The solvability condition is obtained the same way as it was for the Allen–Cahn model. By virtue of $\int (\Phi_0)_z^2 dz = \frac{4}{3}$, we obtain

$$U_0 = -\beta V_0 - d_0\kappa_0, \tag{12}$$

which is the well-known Gibbs–Thomson condition with a kinetic undercooling term. To next order in u, we obtain the equation

$$(U_1)_{zz} = \frac{V_0}{2}(\Phi_0).$$

Integrating twice gives

$$u_1 = \frac{V_0}{2}\log\cosh z + Az + B,$$

where the constants of integration A and B can be found by using the appropriate matching conditions. This gives a relationship for the jump in the normal derivative of u_0 across the interface, namely the "Stefan" condition

$$\left[\frac{du_0}{dn}\right]_\Gamma = V_0.$$

This is apparent from the form of u_1, since its derivative with respect to the unscaled normal coordinate r is $\epsilon^{-1}V_0\tanh(z)/2$.

An important observation due to Karma and Rappel [15] is that the analysis may be continued to a further order, and with an appropriate choice of μ, the condition (12) may be enforced to even a further order. Thus, the asymptotic error is only $\mathcal{O}(\epsilon^2)$, comparable to the error of a second-order numerical scheme when $\Delta x \approx \epsilon$. For this particular model, the correct choice is $\mu = 5/12$.

The main observation to be made is that near the interface, u has form

$$u = \frac{\epsilon}{2}V\log\cosh(z) + \text{a linear function}. \tag{13}$$

This motivates the change of variables presented next.

4.2. Transformation of the Phase Field Equations

For the variable ϕ, we change to the new variable ψ as before (8). Note that to a good approximation, $\psi \approx \epsilon z$, so that a good candidate for a transformation of u near the interface is

$$w = u - \frac{\epsilon}{2}V(x,t)\log\cosh\left(\frac{\psi}{\epsilon}\right).$$

This means, according to (13), that w will be approximately linear in z, unlike u whose derivative suffers a sharp transition across the interface. The function $V(x,t)$ may be chosen in any number of ways to approximate the interface velocity; one natural choice is $V(x,t) = -\psi_t(x,t)$. In the numerical experiments below, V is taken at each time step to be a function of space *only*; that is, for some particular time t, $V(x) = -\psi_t(x,t)$. This eliminates time derivatives of V which appear in the transformed equations.

The evolution equations for the new variables are

$$\frac{\epsilon\mu+\beta}{d_0}\psi_t = \Delta\psi - \frac{5}{4}\frac{u}{d_0}\operatorname{sech}^2\left(\frac{\psi}{\epsilon}\right) + 2\epsilon^{-1}\tanh\left(\frac{\psi}{\epsilon}\right)(1-|\nabla\psi|^2) \tag{14}$$

$$w_t = \Delta w + \frac{\epsilon^{-1}}{2}\operatorname{sech}^2\left(\frac{\psi}{\epsilon}\right)(\psi_t + V|\nabla\psi|^2) + \frac{1}{2}\tanh\left(\frac{\psi}{\epsilon}\right) \times (2\nabla V\cdot\nabla\psi + V(\Delta\psi-\psi_t)) + \frac{\epsilon}{2}\log\cosh\left(\frac{\psi}{\epsilon}\right)(\Delta V - V_t). \tag{15}$$

At first sight, this system may seem formidable. Equation (14) is, with exception of the term containing u, the same as (9). The second equation contains a number of terms, but each is straightforward to compute. The ϵ^{-1} term in (15) not actually that large: since $V = -\psi_t$, this term actually contains a factor of $1-|\nabla\psi|^2$, which is small because Eq. (14) forces ψ to be close to a signed distance function.

4.3. A 1-D Experiment

One test of the proposed transformation is the computation of a one-dimensional traveling wave solution. Here the system is supplemented with the far field conditions

$$\lim_{x\to\pm_\infty}\phi(x) = \mp 1, \quad \lim_{x\to-\infty}u'(x) = 0, \quad \lim_{x\to-\infty}u(x) = -\Delta.$$

When $\Delta > 1$, there is an exact traveling wave solution to the sharp interface limit free boundary problem, namely

$$u(x,t) = \begin{cases} -\Delta + \exp(-cx + c^2t) & x - ct \geq 0 \\ 0 & x - ct < 0, \end{cases}$$

which has a speed of propagation

$$c = \frac{\Delta - 1}{\beta}.$$

The goal of the experiment was to compute these velocities for original and transformed equations and compare them.

The spatial grid used was uniform, and as the simulation progressed, the grid adaptively moved keep the transition layer in the center of the computational domain. The discretization of the equations was much the same as before, with space derivatives approximated to second order. An explicit time stepping procedure was used, and the Laplacian and gradient terms were replaced by their usual second-order, finite difference approximations.

At each step, ψ_t was computed by finite differences, and the function V was regarded as a function of space only, setting

$$V(x) = \psi_t(x, t).$$

Before the variable u was updated, it was transformed into w, and the w-equation was updated in the interfacial region. Away from the interface, the u-equation was computed normally, however, since the contribution of the ϕ term is negligible there.

The specific problem had the parameters $\epsilon = 0.1$, $d_0 = 0.5$, $\beta = 0.1$, $\Delta = 1.1$, and μ chosen as before so that the sharp interface limit is approximated to order $\mathcal{O}(\epsilon^2)$. The sharp interface steady state velocity is therefore $= 1$.

In each experiment, the time step was $\Delta t = 5 \times 10^{-6}$, and the simulation was run until the average velocity stabilized within 10^{-4}. The velocity was measured over a time interval of 10 units, so that small oscillations in the interface position were averaged out. Simulations were run for the original equations, the fully transformed equations, and the case where only ϕ was transformed.

The results of the simulation are presented in Fig. 3. The conclusions are very similar to what was presented for the Allen–Cahn model, that computing the transformed equations

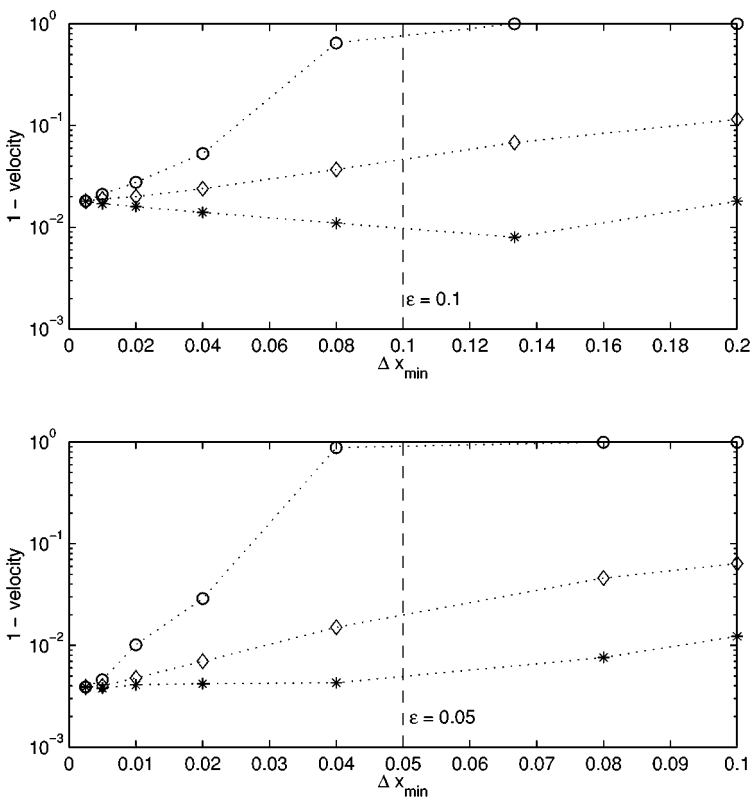

FIG. 3. Steady velocities for the phase field model as a function of the minimum grid refinement for two values of ϵ. Shown is the difference between the asymptotic velocity (=1) and the velocities obtained by computing the untransformed equation (circles), transforming ϕ only (diamonds) and transforming both variables ($*$). Note that pinning ($V \to 0$) occurred in the original version when the grid refinement approached ϵ.

gives substantially less numerical error, especially as $\Delta x_{\min}$ approaches ϵ. The necessity of the u-transformation is also apparent.

Notice that the asymptotic error is reduced roughly by a factor of four when ϵ is halved, confirming that the asymptotic error is $\mathcal{O}(\epsilon^2)$. Furthermore, the numerical error is comparable to the asymptotic error for the transformed equation even when the grid refinement is comparable to ϵ. In other words, there is no advantage in refining the grid beyond this point if the only goal is to compute the sharp interfacial motion.

4.4. 2-D Dendritic Growth

The second experiment involved computing a steadily growing, parabolic shaped dendrite on a two-dimensional domain. This growth mode requires anisotropy of surface tension, represented here by the parameter d_0, which has been included in phase field models previously in a number of ways [23, 24]. For the example here, the d_0 terms in (10) and (14) were replaced by

$$d_0(1 - 15\gamma \cos(4\theta)),$$

where γ is a measure of the anisotropy strength. The angle θ is the orientation of the interface, which satisfies

$$\tan(\theta) = \frac{\phi_y}{\phi_x} = \frac{\psi_y}{\psi_x}.$$

With this change, the essential features of the asymptotics are unchanged, and thus the proposed transformation should be equally beneficial.

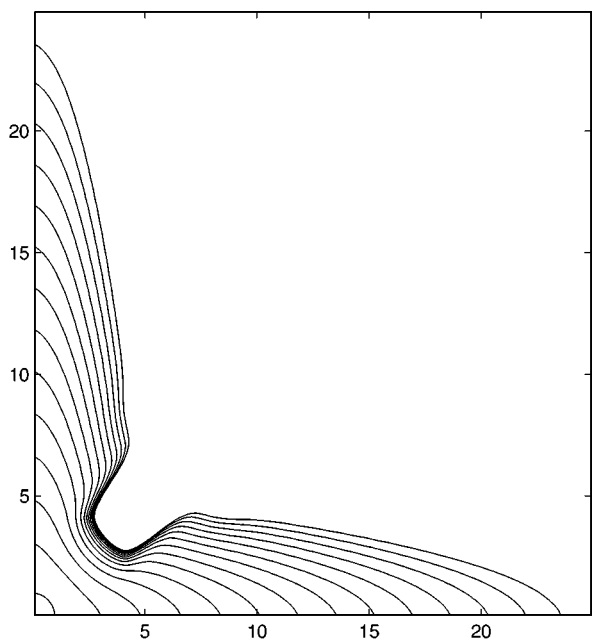

FIG. 4. Evolution of a crystalline seed. Shown are the interfaces (zero level sets) at times $t = 0, 2, 4, \ldots$.

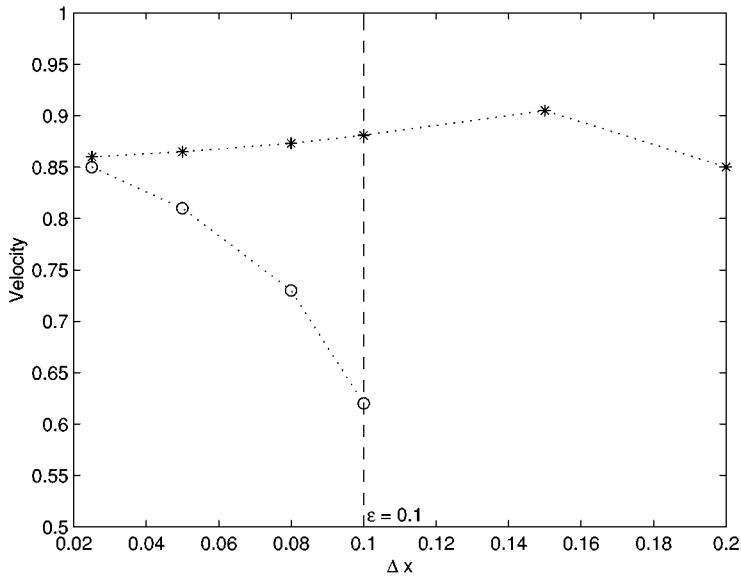

FIG. 5. Steady velocities for the two-dimensional phase field dendrite. (○) denote the results of computing with the usual equations, and (∗) are the results using the transformed equations. Again, when Δx was larger than ϵ, no steady growth was observed using the original equations.

The simulations were run assuming four-fold symmetry on a uniform spatial grid with physical dimensions $[0, 25] \times [0, 25]$. The initial data

$$\phi = \tanh\left(\frac{1 - |x|}{\epsilon}\right)$$

$$u = \begin{cases} -d_0 & |x| < 1 \\ -\Delta + (\Delta - d_0)\exp(1 - |x|) & |x| > 1 \end{cases}$$

were used for all computations, representing a small circular seed which gradually develops into a four-fold symmetric dendrite whose arms grow steadily (see Fig. 4).

The numerical experiment measured the steady growth rates of the dendrite tip for both the original and transformed equations. As with the 1-D experiment, Eq. (15) was only used near the interface; otherwise the untransformed version was used. All simulations used the parameters $\epsilon = 0.1$, $d_0 = .02$, $\beta = 0$, $\Delta = .55$, $\mu = 5/12$, $\gamma = .05$.

The results are similar to the preceding experiments (Fig. 5). As the mesh spacing approached the characteristic interface width ϵ, the normally computed equations became very inaccurate, whereas the preconditioned approach yielded reasonable results even when the mesh spacing was 2ϵ.

5. CONCLUSIONS

We have shown that simple, well-motivated transformations can significantly reduce errors associated with typical discretization schemes for diffuse interface problems. Consequently, coarser grids may be utilized, expediting computation.

The experimental results here indicate that grid resolutions may be coarsened by a factor between 4 and 10. In higher dimensions, this means that grid sizes may smaller by several

orders of magnitude, giving at least that much advantage in speed. Furthermore, there should be improvements in time truncation errors, allowing for larger time steps.

For the purpose of computing approximations to free boundary problems, we have given an empirical criterion for grid refinement: that the grid spacing Δx needn't be smaller than the interface width ϵ. Beyond this point, asymptotic error is larger than numerical error anyhow, so refinement would be pointless. The net result is that it is the size of the grid and not the interface width which limits the accuracy of the computation. This suggests that it is possible to make diffuse interface models competitive with other methods for simulating interface motion.

Regardless of the method used, there is usually no reason to keep track of phase variables like ϕ away from interfaces where they are approximately constant. Because the transformed ψ equations have much in common with level-set representations of interfaces, some techniques from this field can be borrowed. In particular, simple adaptive grid procedures for level set methods exist [1], which may provide an even more efficient method of computation.

There are many diffuse interface problems which could benefit from the techniques described here. Among them are Cahn–Hilliard models [7] for phase separation, diffuse interface models for binary fluids [13, 16], as well as the vast wealth of problems possessing viscous shock layers or reaction diffusion fronts. Applications to some of these are the subject of current investigation.

ACKNOWLEDGMENTS

The author thanks Paul Fife, David Eyre, and Robert Almgren for insightful discussions of diffuse interface models and their computation. The author also thanks Nelson Beebe for use of his fast FORTRAN implementation of the hyperbolic tangent and related functions.

REFERENCES

1. D. Adelsteinsson and J. A. Sethian, A fast level set method for propagating interfaces, *J. Comput. Phys.* **118**, 269 (1995).
2. S. Allen and J. Cahn, A microscopic theory for antiphase boundary motion and its application to domain coarsening, *Acta. Metall*. **27**, 1084 (1979).
3. R. Almgren, Second-order phase field asymptotics for unequal conductivities, *SIAM J. Appl. Math.* **59**, 2086 (1999).
4. R. Almgren and A. Almgren, *Phase Field Instabilities and Adaptive Mesh Refinement* (TMS/Soc. for Industr. & Appl. Math., Philadelphia, 1996), pp. 205–214.
5. R. J. Braun and B. T. Murray, Adaptive phase-field computations of dendritic crystal growth, *J. Crystl. Growth* **174**, 41 (1997).
6. G. Caginalp and P. Fife, Dynamics of layered interfaces arising from phase boundaries, *SIAM J. Appl. Math.* **48**, 506 (1988).
7. J. W. Cahn and J. E. Hilliard, Free energy of a nonuniform system I. Interfacial free energy, *J. Chem. Phys.* **28**, 258 (1957).
8. Y. G. Chen, Y. Giga, and S. Goto, Uniqueness and existence of viscosity solutions of generalized mean curvature flow equation, *J. Diff. Geom.* **33**, 749 (1991).
9. L. C. Evans and J. Spruck, Motion of level sets by mean curvature, *J. Diff. Geom.* **33**, 635 (1991).
10. P. C. Fife, *Dynamics of Internal Layers and Diffuse interfaces* (Soc. for Industr. & Appl. Math.) Philadelphia, 1988.

11. P. C. Fife and J. B. McLeod, The approach of solutions to nonlinear diffusion equations to traveling front solutions, *Arch. Rat. Mech. Anal.* **65**, 335 (1977).
12. M. Gage and R. S. Hamilton, The heat equation shrinking convex plane curves, *J. Diff. Geom.* **23**, 69 (1986).
13. M. E. Gurtin, D. Polignone, and J. Vinals, Two phase fluids and immiscible fluids described by and order parameter, *Math. Models Meth. Appl. Sci.* **6**, 815 (1996).
14. D. Juric and G. Tryggvason, A front tracking method for dendritic solidification, *J. Comput. Phys.* **123**, 127 (1996).
15. A. Karma and W.-J. Rappel, Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics, *Phys. Rev. E.* **53**, 3017 (1996).
16. J. Lowengrub and L. Truskinovsky, Quasi-incompressible Cahn–Hilliard fluids and topological transitions, *Proc. Royal Soc. A* **454**, 2617 (1998).
17. B. Merriman, J. K. Bence, and S. J. Osher, Motion of multiple junctions: A level set approach, *J. Comput. Phys.* **112**, 334 (1994).
18. R. H. Nochetto, M. Paolini, and C. Verdi, A dynamic mesh algorithm for curvature dependent evolving interfaces, *J. Comp. Phys.* **123**, 296 (1996).
19. N. Provatas, N. Goldenfeld, and J. Dantzig, Efficient computation of dendritic microstructure using adaptive mesh refinement, *Phys. Rev. Lett.* **80**, 3308 (1998).
20. J. Rubinstein, P. Sternberg, and J. B. Keller, Fast reaction, slow diffusion, and curve shortening, *SIAM J. Appl. Math.* **49**, 116 (1989).
21. J. A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge Univ. Press, Cambridge, UK, 1999).
22. S. Unverdi and G. Tryggvason, A front tracking method for viscous, incompressible multifluid flows, *J. Comput. Phys.* **100**, 25 (1992).
23. A. A. Wheeler and G. B. McFadden, A ξ-vector formulation of anisotropic phase-field models: 3d asymptotics, Euro. *J. Appl. Math.* **7**, 367 (1996).
24. A. A. Wheeler, B. Murray, and R. Schaefer, Computation of dendrites using a phase-field model, *Physica D* **66**, 243 (1993).