Journal of Computational Physics **151,** 498–533 (1999)
Article ID jcph.1999.6194, available online at http://www.idealibrary.com on IDEAL®

Semi-Lagrangian Methods for Level Set Equations

John Strain[1]

Department of Mathematics, University of California, 970 Evans Hall,
Number 3840, Berkeley, California 94720-3840
E-mail: strain@math.berkeley.edu

Received August 19, 1998; revised January 5, 1999

A new numerical method for solving geometric moving interface problems is presented. The method combines a level set approach and a semi-Lagrangian time stepping scheme which is explicit yet unconditionally stable. The combination decouples each mesh point from the others and the time step from the CFL stability condition, permitting the construction of methods which are efficient, adaptive, and modular. Analysis of a linear one-dimensional model problem suggests a surprising convergence criterion which is supported by heuristic arguments and confirmed by an extensive collection of two-dimensional numerical results. The new method computes correct viscosity solutions to problems involving geometry, anisotropy, curvature, and complex topological events. © 1999 Academic Press

Key Words: moving interfaces; level sets; backward characteristics; semi-Lagrangian schemes; CIR scheme; motion by curvature.

CONTENTS

1. INTRODUCTION

We present a new numerical method for moving interface problems. The method merges and breaks interfaces naturally and generally via the level set approach, while decoupling

[1] Research supported by NSF Young Investigator and SCREMS Awards and by Air Force Office of Scientific Research Grant FDF49620-93-1-0053.

time step restrictions from the Courant–Friedrichs-Lewy (CFL) stability condition by using an explicit yet unconditionally stable semi-Lagrangian time stepping scheme with velocity smoothing and frequent redistancing. The time stepping scheme also decouples each mesh point from the others, potentially simplifying both adaptive mesh refinement and parallel implementation.

Section 2 of this paper contains standard background material: moving interface problems and examples, level set and semi-Lagrangian methods. Section 3 presents our method and explains why it works. Section 4 validates it by solving an extensive collection of numerical examples including geometric motions with corners, anisotropy, curvature, and complex topology. Section 5 draws conclusions and discusses future extensions and applications.

2. BACKGROUND

This section summarizes standard background material on moving interface problems and numerical methods. Subsection 2.1 classifies moving interface problems commonly found in applications, by the degree of locality of the velocity as a functional of the interface. Subsection 2.2 describes how to convert these problems into level set equations on a fixed domain, eliminating the moving interface. Subsection 2.3 introduces the level set method for moving interfaces, Subsection 2.4 relates moving interfaces and CFL conditions for some important model problems, and Subsection 2.5 reviews and analyzes the simplest semi-Lagrangian scheme for hyperbolic partial differential equations (PDEs). Subsection 2.6 discusses the derivation of higher-order accurate semi-Lagrangian schemes.

2.1. Moving Interface Problems

A moving interface $\Gamma(t)$ is a collection of nonintersecting oriented closed curves in $\mathbf{R}^2$ or surfaces in $\mathbf{R}^3$ for each time t, a set-valued function of time. Since each component of $\Gamma(t)$ is closed, $\Gamma(t)$ has an interior and an exterior. Assume $\Gamma(t)$ is sufficiently smooth in space and time. Then for each time t and each $x \in \Gamma(t)$ there is

- ○ an outward unit normal vector $N(x, t)$,
- ○ a signed curvature $C(x, t)$, chosen positive for a circle or sphere, and
- ○ a normal velocity $V(x, t)$, chosen positive where the interior of $\Gamma(t)$ is growing.

Given a parametrization of $\Gamma(t)$, these quantities can be calculated by standard geometric formulas found in [41].

A *moving interface problem* is a closed system of equations which specifies the normal velocity V as a functional of Γ and the other unknowns in the problem. Such problems can be divided into three broad classes involving passive transport, geometry, and/or PDEs or integral relations off the interface. All occur frequently in applications.

2.1.1. *Passive Transport*

Passive transport moves an interface in some external flow, which may be given a priori or computed on the fly but does not depend on the interface itself. Thus $F(x, t)$ is a given velocity field on $\mathbf{R}^d$ and the normal velocity of $\Gamma(t)$ is $V(x, t) = N(x, t) \cdot F(x, t)$ which is independent of $\Gamma(t)$. This type of problem occurs when modeling common and important physical situations such as rotation, shearing, and stretching in an ambient flow, and is conceptually the simplest to solve because the motion of each point on the interface obeys an ordinary differential equation with a known right-hand side.

2.1.2. *Geometry*

More complex problems allow the local interfacial geometry to interact with the motion, so the interface satisfies a partial rather than ordinary differential equation. The normal velocity is a given function

$$V = V(x, t, N, C, \dots) \tag{1}$$

of the interfacial position, normal, curvature, and other local geometric quantities.

EXAMPLE 1. The simplest geometric motion propagates $\Gamma(t)$ along its normal vector with constant uniform velocity. Corners form and merging occurs if $\Gamma(0)$ is not convex, so $\Gamma(t)$ does not remain smooth, yielding the simplest example of a "viscosity solution" to a Hamilton–Jacobi equation [7, 13].

Specialized methods for motion with unit normal velocity can be built from Huygens' principle: $\Gamma(t)$ is an envelope of the set of radius-$|t|$ circles centered on each point of $\Gamma(0)$. Consider, for example, the inverted "V" $y = -|x|$ shown in Fig. 1. If $\Gamma(t)$ is given by $y = \psi(x, t)$, Huygens' principle gives

$$\psi(x, t) = \begin{cases} x + \sqrt{2}t, & x < -t/\sqrt{2} \\ \sqrt{t^2 - x^2}, & |x| < t/\sqrt{2} \\ -x + \sqrt{2}t, & x > t/\sqrt{2} \end{cases} \tag{2}$$

for $t \geq 0$. As $t \leq 0$ decreases, the inner envelope remains sharp: $\psi(x, t) = -|x - \sqrt{2}t|$ for $t < 0$. Time-reversal symmetry is broken, as for shocks in hyperbolic conservation laws [17].

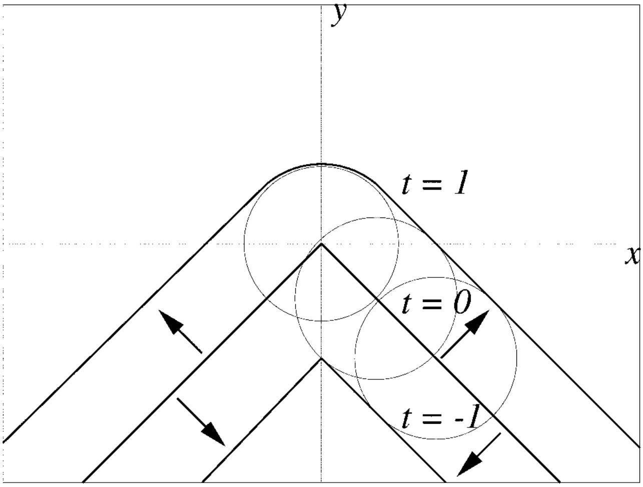

FIG. 1. Corners moving outward with unit velocity round off into circular arcs, while corners moving inward remain sharp by Huygens' principle.

EXAMPLE 2. A common two-dimensional geometric problem has a curve $\Gamma(t)$ evolving under a K-fold symmetric anisotropic normal velocity

$$V(x,t) = R + \epsilon\cos(K\theta + \theta_0) + (R' + \epsilon'\cos(K'\theta + \theta_0'))C, \tag{3}$$

where $\cos\theta = N \cdot e_1$ is the cosine of the angle between the normal vector and the positive x-axis.

Anisotropic velocity fields grow or shrink interfaces along their normals with speed depending on local orientation, easily producing complex merging shapes and making these models popular in materials science [40]. With sufficient anisotropy, such velocity fields produce faceted interfaces via the Wulff construction [8, 21, 23, 43, 42]. At the corners of facets, the viscosity solution behaves differently from Example 1, because the velocity is anisotropic. Rather than rounding off, the corner remains sharp even when the velocity is a smooth function of the normal direction. See Subsections 4.2.3 and 4.2.7 for numerical examples.

2.1.3. *PDE*

In moving interface problems for PDEs, the interfacial velocity depends on additional fields satisfying algebraic, ordinary differential, partial differential, or integral equations on or off the interface. These fields can mediate long-distance nonlocal interactions, so the evolution equation for the interface is no longer a local PDE.

EXAMPLE 3. In volume diffusion [9, 20],

$$V(x,t) = \frac{\partial u(x,t)}{\partial N}, \tag{4}$$

where $u(x,t)$ solves the Laplace equation

$$\Delta u = 0 \qquad \text{outside } \Gamma(t) \tag{5}$$

and the boundary condition

$$u = C \qquad \text{on } \Gamma(t), \tag{6}$$

with boundary conditions at ∞. Using the Dirichlet–Neumann operator Λ_Γ which maps boundary values for the Laplace equation outside Γ to the normal derivative of the solution on Γ, Eqs. (4)–(6) become a single nonlinear nonlocal pseudodifferential equation

$$V(t) = \Lambda_{\Gamma(t)}C(t). \tag{7}$$

Equation (7) gives the velocity V, and a curve movement equation which moves the interface with given velocity V completes the moving interface problem. Several curve movement equations exist [34].

EXAMPLE 4. A model for crystal growth is treated in [2, 4, 5, 26, 28, 34]. Here V is the jump across the interface of the normal derivative $\frac{\partial u(x,t)}{\partial N}$, where u satisfies the Stefan problem

$$u_t = \Delta u \qquad \text{off } \Gamma(t) \tag{8}$$

$$u = -\epsilon_c(N)C - \epsilon_v(N)V \qquad \text{on } \Gamma(t) \tag{9}$$

with boundary conditions on outer boundaries. Here ϵ_c and ϵ_v are given functions of the outward normal N, as in Example 2.

Problems close to engineering practice often involve complex systems of PDEs and integral equations modeling physical effects such as heat flow, convection, elasticity, radiation, chemical and biological reactions, and fields satisfying integrodifferential conditions on the interface itself. Such problems can be extremely difficult to solve numerically, even without moving interfaces.

2.2. Level Set Equations

Moving interface problems can be reformulated as "level set equations" on a fixed domain, using the *zero set*

$$\Gamma = \{x \in \mathbf{R}^d : \varphi(x) = 0\} \tag{10}$$

of a function $\varphi : \mathbf{R}^d \to \mathbf{R}$. Given an interface Γ, there are many functions φ for which Γ is the zero set: for example, the distance and the signed distance to Γ,

$$\varphi(x) = \min_{y\in\Gamma} \|x - y\|, \qquad \varphi(x) = \pm\min_{y\in\Gamma} \|x - y\|, \tag{11}$$

where the plus sign is chosen for x in the interior of Γ. However, not every zero set is admissible as an interface. Zero sets may be flat where φ is equal to zero on a region and may cross at isolated points. These pathologies are excluded if $\nabla\varphi$ never vanishes on Γ. Then φ crosses zero cleanly and we can recover Γ from φ by contouring. Thus the signed distance represents Γ more stably than the distance. Figure 2 shows a hexagon in the plane and the corresponding signed distance function φ.

Many geometric properties of Γ have simple expressions in terms of φ, because φ contains local information which allows implicit differentiation of Γ. For example, the normal velocity, outward unit normal, and curvature are given by

$$V = \varphi_t / \|\nabla\varphi\|, \tag{12}$$

$$N = \nabla\varphi / \|\nabla\varphi\|, \tag{13}$$

$$C = -\nabla \cdot N, \tag{14}$$

if φ is chosen to be positive inside the zero set [41]. These formulas can be evaluated everywhere φ is known, as well as on Γ. At a point x away from Γ, they give the geometry of the level set passing through x.

Thus if we have the interface then we can compute its velocity from φ. Conversely, given an extension of the normal velocity V to a function of t and $x \in \mathbf{R}^d$, Eq. (12) can be viewed

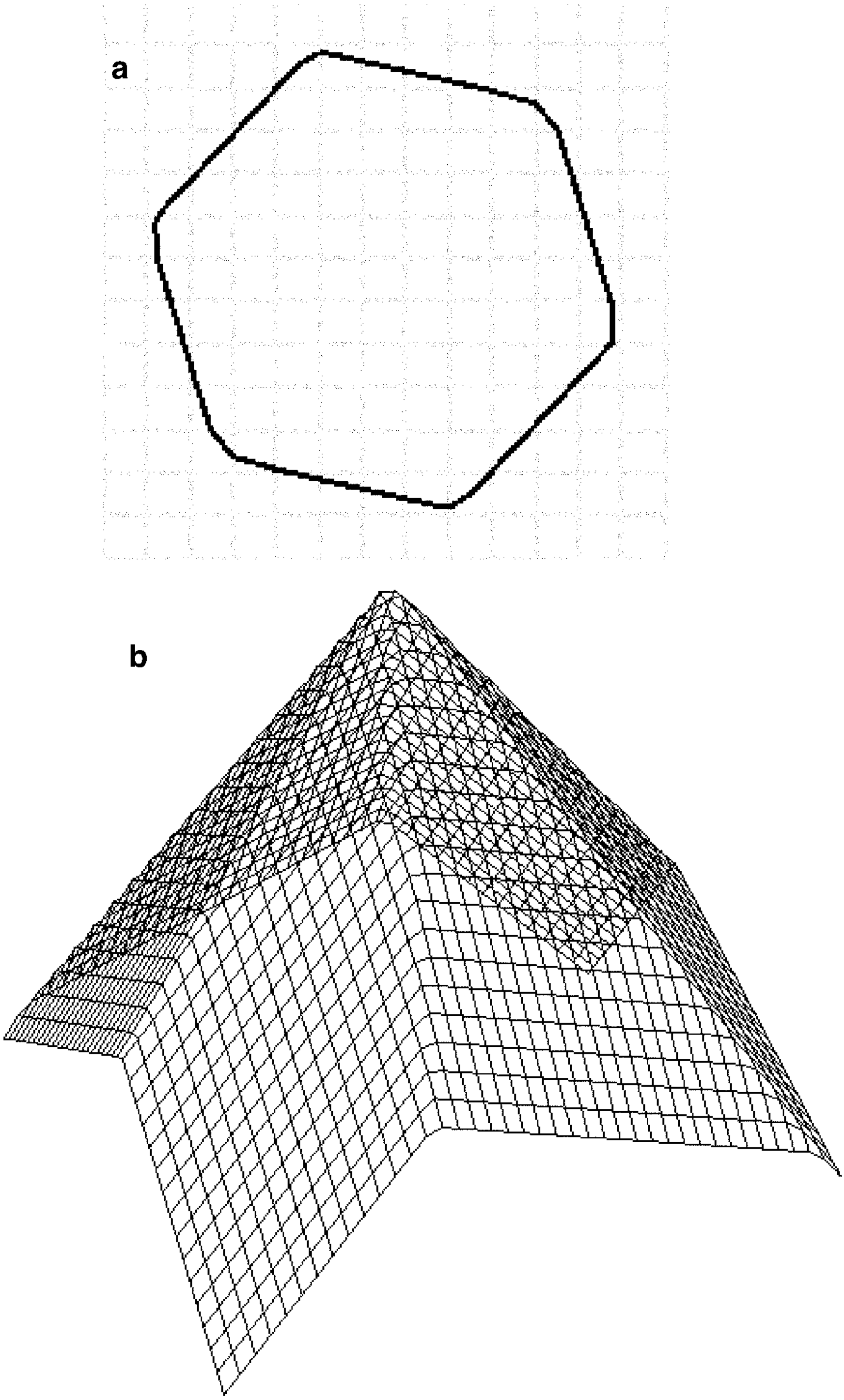

FIG. 2. The correspondence between (a) a hexagonal interface and (b) the signed distance φ to the interface, plotted over a 20^2 grid.

as a PDE—the “level set equation”—which moves Γ by evolving φ:

$$\varphi_t - V\|\nabla\varphi\| = 0. \tag{15}$$

Alternatively, we can construct a vector velocity field F on $\mathbf{R}^d$ with $F = VN$ on Γ, and solve the “linear level set equation”

$$\varphi_t - F \cdot \nabla\varphi = 0. \tag{16}$$

Equations (15) and (16) move every level set of φ with the extended velocity V or F, and in particular move the zero set with the correct velocity. This approach to moving interfaces embeds the topology in φ rather than $\Gamma(t)$, allowing merging, breaking, and other topological changes to be handled automatically. We pay the price of going up one dimension. Either V or VN must be extended to a function on the whole space, but the extension can be almost completely arbitrary away from $\Gamma(t)$.

The moving interface problems from Subsection 2.1 can be put in level set form as follows.

2.2.1. *Passive Transport*

For passive transport, F is already defined on $\mathbf{R}^d$ and is a natural extension of VN. Since N can be extended by Eq. (13), a natural extension of V is $N \cdot F$. The resulting level set equation is a hyperbolic PDE, nonlinear if V is extended,

$$\varphi_t - V(x,t)\|\nabla\varphi\| = 0 \tag{17}$$

and linear if F is extended,

$$\varphi_t - F(x,t) \cdot \nabla\varphi = 0. \tag{18}$$

N is singular where $\nabla\varphi$ vanishes or is singular. For example, in Fig. 2, $\nabla\varphi$ does not exist at the center and the corners of each hexagonal level set, where φ is not differentiable. Even if $\nabla\varphi$ exists everywhere, it must vanish at maximum points interior to Γ, so N is never globally smooth. This suggests that we should extend F rather than V, solve Eq. (16) instead of Eq. (15), and avoid using N off Γ.

2.2.2. *Geometry*

With geometric quantities extended naturally by Eqs. (12)–(14), the level set equation for Example 2 reads

$$\varphi_t - (R + \epsilon\cos(K\theta + \theta_0))\|\nabla\varphi\| = (R' + \epsilon'\cos(K'\theta + \theta_0'))\nabla \cdot (\nabla\varphi/\|\nabla\varphi\|)\|\nabla\varphi\|, \tag{19}$$

where $\cos\theta = \varphi_x/\|\nabla\varphi\|$. This is a mixed hyperbolic-parabolic PDE containing both first-order and second-order spatial derivatives of φ, and becoming singular where $\nabla\varphi$ vanishes.

2.2.3. *PDE*

For fluid problems with moving interfaces, the fluid velocity provides a natural extension of VN off the interface. But in many other PDE-type models, the normal velocity is built from quantities such as boundary values and jump conditions, whose natural habitat is the interface. Then an extension of V is not obvious. One could set $V = N \cdot \nabla u$ in Example 3 (volume diffusion), but ∇u is discontinuous across $\Gamma(t)$. In Example 4 (crystal growth), V is defined as a jump across $\Gamma(t)$ and an extension of V is even less obvious. Thus various extensions have been developed: In [28], for example, our Eq. (9) was solved for V under the assumption $\epsilon_v(N) \neq 0$ to get

$$V = \frac{-1}{\epsilon_v(N)}(u + \epsilon_c(N)C), \tag{20}$$

where N and C are extended naturally and the jump condition is built into the solution of the heat equation via classical potential theory. General schemes which extend any velocity field off any interface were presented in [1, 5, 38, 39] and applied to this crystal growth model in [5].

2.3. The Level Set Method

The level set method moves $\Gamma(t)$ from $t = 0$ by constructing an initial level set function $\varphi(x, 0)$ for $\Gamma(0)$ and an extended velocity field V or F for $t \geq 0$, solving one of the level set equations Eq. (15) or Eq. (16) numerically, then finding $\Gamma(t)$ from $\varphi(x, t)$ when required. The method was introduced in [22], and an extensive recent survey is [27]. It has undergone much development and been applied to many moving interface problems.

The main advantage of the level set method over other numerical methods for moving interfaces is its natural treatment of topological changes such as merging and breaking. These changes can be difficult to handle with methods based on parametrization, but solving the level set equation merges interfaces naturally and automatically as shown in Fig. 3.

There are some potential difficulties with the level set method. It can be more expensive since it goes up a dimension, particularly if uniform meshes are used. Extending the velocity off $\Gamma(t)$ can be difficult. One must be careful to obtain the correct "viscosity solution" of Eq. (15) or Eq. (16), by using an appropriate solver for the level set equation [27]. The method is not sufficiently *modular*; a new code must be written for each new problem to be solved, since the velocity evaluation is intertwined with the moving interface code by velocity extension and CFL conditions.

We present a level set solver on a uniform mesh in Section 3, which is shown experimentally to obtain the correct viscosity solution for passive transport and geometric problems where velocity extension is straightforward. This solver is designed for easy adaptive mesh refinement with large time steps, yielding optimal efficiency. An adaptive version is developed in [37]. On this foundation, an efficient, general, and robust velocity extension is built in [38] and yields a completely modular level set method.

2.4. CFL Conditions

Almost all explicit schemes for PDEs such as the level set equation encounter time step restrictions due to the famous Courant–Friedrichs–Lewy (CFL) condition [17]. This necessary condition for convergence requires that in the limit as the time and space mesh sizes go

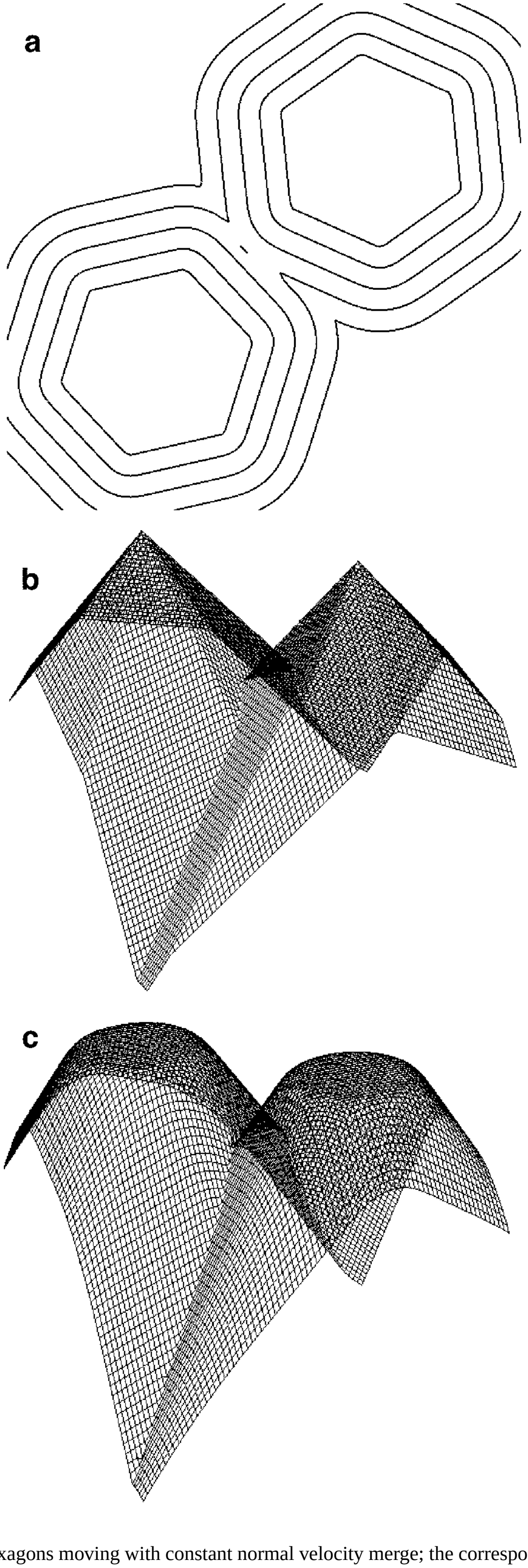

FIG. 3. (a) Two hexagons moving with constant normal velocity merge; the corresponding level set function is shown at (b) initial and (c) final times.

to zero, the domain of dependence of the numerical solution at each spacetime point must include that of the exact solution. For explicit schemes with bounded stencils for first-order hyperbolic PDEs, the CFL condition imposes a time step restriction of the dimensionally natural form $|Uk| \leq O(h)$, where k is the time step, h is the spatial mesh size, and U is proportional to a characteristic velocity of the PDE. For higher-order PDEs these time step restrictions often become $k \leq O(h^2)$ or $O(h^3)$ and make explicit schemes prohibitively expensive. The usual remedy—implicit time-stepping schemes—is often unavailable for level set equations because the complex and problem-dependent relation between V and $\Gamma(t)$ frustrates most nonlinear equation solvers.

In passive transport and unit normal velocity, the level set equation is first-order hyperbolic, so most explicit schemes encounter a time step restriction $k \leq O(h)$. This restriction is inconvenient if a fine or adaptive mesh is used. In the curvature-dependent geometric motion of Example 2, explicit treatment of the second-order parabolic term requires an asymptotically smaller time step $k \leq O(h^2)$. Volume diffusion (Example 3) involves the theory of the “Dirichlet–Neumann operator” Λ which maps boundary values to normal derivatives. Λ is a first-order pseudodifferential operator, and C is a second derivative of φ, so $V = \Lambda C$ resembles a third-order derivative of φ. Therefore $k \leq O(h^3)$, and similarly in Example 4 (crystal growth). This condition requires extremely small time steps. If higher-order PDEs such as elasticity are involved, these small time steps can make most schemes prohibitively expensive.

These time step restrictions can be eliminated by allowing unbounded stencils. For example, we can build a trivial explicit method for the heat equation which is stable and convergent with large time steps $k = O(h)$, if we allow stencil size to grow as the mesh is refined. Take the standard explicit finite difference method on a sequence of meshes with mesh sizes $h = 1/n$ and time step $\Delta t = h^2/2$, so the usual CFL condition is satisfied. Define a new finite difference method with step size $k = h = 2n\Delta t = 1/n$ by taking $2n$ tiny steps of the standard method to pass from t to $t + k$. The new method is stable and convergent with $k = h$, hence satisfies the CFL condition.

Our moving interface method decouples time steps from CFL conditions by using the explicit unconditionally stable time stepping scheme reviewed in Subsection 2.5. More general schemes of this “semi-Lagrangian” type are presented in Subsection 2.6. For first-order hyperbolic problems, these schemes satisfy the CFL condition with large time steps by shifting the stencil. For higher-order level set equations, heuristic reasons for our methods to satisfy CFL conditions are discussed in Subsection 3.3.

2.5. The CIR Scheme

Consider the simplest linear hyperbolic PDE

$$\varphi_t - F(x,t) \cdot \nabla\varphi = 0. \tag{21}$$

Equation (21) propagates φ values along the characteristic curves $s(t)$ defined by

$$\dot{x}(t) = -F(s(t),t), \qquad s(0) = x_0, \tag{22}$$

because

$$\frac{d}{dt}\varphi(s(t),t) = \varphi_t + \dot{x} \cdot \varphi_x = \varphi_t - F \cdot \nabla\varphi = 0 \tag{23}$$

if φ solves Eq. (21). Thus we can find φ values at any time t by finding the characteristic curve passing through (x, t) and following it backwards to some previous point (x_0, t_0) where the value of φ is known: then $\varphi(x, t) = \varphi(x_0, t_0)$. This observation forms the basis of the “backward characteristic” or “CIR” scheme due to Courant, Isaacson, and Rees [6], which is the simplest semi-Lagrangian scheme. Given φ at time t_n, CIR approximates $\varphi(x, t_{n+1})$ at any point x at time $t_{n+1} = t_n + k$ by evaluating the previous velocity $F(x, t_n)$, approximating the backward characteristic through x by a straight line

$$x + (t_{n+1} - t)F(x, t_n) \approx s(t) \tag{24}$$

and interpolating φ at time t_n to the point

$$x + kF(x, t_n) \approx s(t_n). \tag{25}$$

Then $\varphi(x, t_{n+1})$ is set equal to the interpolated value.

For linear PDEs, the Lax–Richtmyer equivalence theorem [17] guarantees that CIR will converge to the exact solution as $k, h \to 0$ if it is stable and consistent. For nonlinear PDEs, stability and consistency are necessary but not sufficient.

2.5.1. *Stability*

The stability properties of the CIR scheme are excellent. Each new value $\varphi(x, t_{n+1})$ is a single interpolated value of φ at time t_n, so unconditional stability is guaranteed in any norm where the interpolation does not increase norms. For example, CIR with linear interpolation is unconditionally stable in the maximum norm. In general, semi-Lagrangian schemes satisfy the CFL condition by shifting the stencil, rather than restricting the time step. Thus information propagates over long distances in one step.

2.5.2. *Consistency*

Explicit unconditionally stable schemes like CIR or the Dufort–Frankel scheme [17] usually require some consistency condition, in place of the time step restriction $k \leq O(h)$ required by other explicit schemes. The consistency condition for CIR can be illustrated with the simplest one-dimensional linear hyperbolic PDE

$$\varphi_t - V\varphi_x = 0, \qquad \varphi(x, 0) = f(x), \tag{26}$$

whose solution is $\varphi(x, t) = f(x + Vt)$. The CIR scheme on a uniform mesh $x = jh,\ t = nk$ produces numerical approximations u_j^n to $\varphi(jh, nk)$ by the formula

$$u_j^{n+1} = qu_{m+1}^n + (1 - q)u_m^n, \tag{27}$$

where

$$m = j - \lfloor Vk/h \rfloor, \qquad q = \frac{(j - m)h - Vk}{h}, \tag{28}$$

as in Fig. 4. The scheme is unconditionally stable because the projected point s need not lie in the same computational cell as x; the stencil shifts to satisfy the CFL condition discussed in Subsection 2.4.

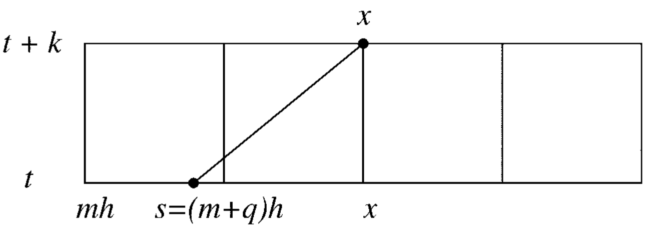

FIG. 4. One-dimensional semi-Lagrangian CIR scheme: move x backward with velocity V, then interpolate φ at time t to point s.

To check consistency, we plug the exact solution φ into the numerical formula and bound the truncation error $\tau(x, t)$ defined by

$$\varphi(jh, (n+1)k) = q\varphi((m+1)h, nk) + (1-q)\varphi(mh, nk) + k\tau(jh, (n+1)k). \quad (29)$$

The scheme is consistent to first order if $\tau = O(h) + O(k)$ on a fixed time interval as $h, k \to O$. Taylor expansion gives

$$\tau = O\left(\frac{h^2}{k}\right) + O(k) \quad (30)$$

if the initial data f have two continuous derivatives. The first term comes from the $O(h^2)$ error in linear interpolation, repeated at $O(1/k)$ time steps, while the second term is due to approximating the characteristics by straight lines with first-order accurate slopes $F(x, t_n)$. Thus CIR is first-order accurate if the following condition is satisfied,

$$k \geq Ch \quad (31)$$

for some arbitrary constant C. This consistency condition differs from the usual time step restriction $|Vk| \leq h$ in two important ways: the inequality is reversed so h is bounded rather than k, and the constant C is completely independent of V and need only be fixed as $k, h \to 0$.

A similar calculation shows that with higher-order accurate interpolation this lower bound becomes even less restrictive. For an interpolation method with error $O(h^q)$ per interpolation, a consistency condition $k \geq O(h^p)$ produces a semi-Lagrangian scheme with formal error $O(h^{q-p}) + O(k)$. However, stability becomes an issue since higher-order interpolation may allow the maximum norm of the solution to increase.

2.5.3. *Nonlinear Hyperbolic Problems*

To apply the CIR scheme to nonlinear hyperbolic PDEs of the form

$$\varphi_t - F(x, t, \varphi) \cdot \nabla\varphi = 0, \quad (32)$$

Courant, Isaacson, and Rees use a standard approach: Freeze φ at time t_n in the argument list of F, then apply the linear CIR scheme to move forward one step from t_n to t_{n+1}. The scheme remains unconditionally stable, and if the solution remains smooth, Taylor expansion shows that consistency is unaffected. However, solutions to a general nonlinear hyperbolic PDE do not remain smooth. Instead, they develop shock discontinuities and degenerate to weak solutions. Uniqueness then fails and an entropy condition is required to select the correct weak solution.

When shocks occur, both theory and numerics become more difficult. If the PDE is a conservation law and the numerical scheme is in conservation form, then the Lax–Wendroff theorem [17] guarantees that any limit of the scheme is a weak solution. Equation (32) is not in general a conservation law, and CIR is not in conservation form, so the Lax–Wendroff theorem does not apply. In fact, CIR moves shocks at the wrong speed even in simple conservation laws [17] and thus cannot be convergent.

Thus the CIR scheme—while explicit and unconditionally stable—has never been popular for solving nonlinear conservation laws. It has been used mainly for linear problems, where stability plus consistency guarantee convergence. In Subsection 3.3, we explain the special features of nonlinear level set equations which permit the convergence of methods based on the CIR scheme.

2.6. Semi-Lagrangian Schemes

Semi-Lagrangian schemes which preserve the unconditional stability of CIR but enjoy higher-order accuracy have been widely used for modeling linear advection in atmospheric science [3, 24, 31, 33]. Their unconditional stability is particularly useful on the sphere [18, 32], where it eliminates the stringent time step restriction encountered by Eulerian schemes on small mesh cells near the poles. In moving interface problems, semi-Lagrangian schemes permit local mesh refinement with large time steps and overcome the inefficiency of level set methods on a uniform mesh. Semi-Lagrangian schemes for special level set equations have been constructed in [11].

An effective viewpoint for the derivation of higher-order accurate semi-Lagrangian schemes is presented by Smolarkiewicz and Pudykiewicz in [31], and involves three steps: spacetime integration, interpolation or advection, and discretization.

2.6.1. *Spacetime Integration*

Consider the linear hyperbolic PDE

$$\varphi_t - F \cdot \nabla\varphi = 0. \tag{33}$$

Suppose we know φ on a regular grid at time s and we seek the values $\varphi(x, t)$ at some time $t > s$. The fundamental theorem of calculus and Eq. (33) give

$$\varphi(x, t) = \varphi(y, s) + \int_C \nabla\varphi \cdot (dx + F dt), \tag{34}$$

where C is any path in spacetime connecting (y, s) to (x, t).

Several well-known classes of schemes for Eq. (33) are distinguished by their choices of C. Eulerian schemes take $x = y$ and C a straight line segment parallel to the t-axis as in Fig. 5a. Pure Lagrangian schemes take C to be the Lagrangian trajectory T defined by

$$\dot{x}(\sigma) = -F(x(\sigma), \sigma) \tag{35}$$

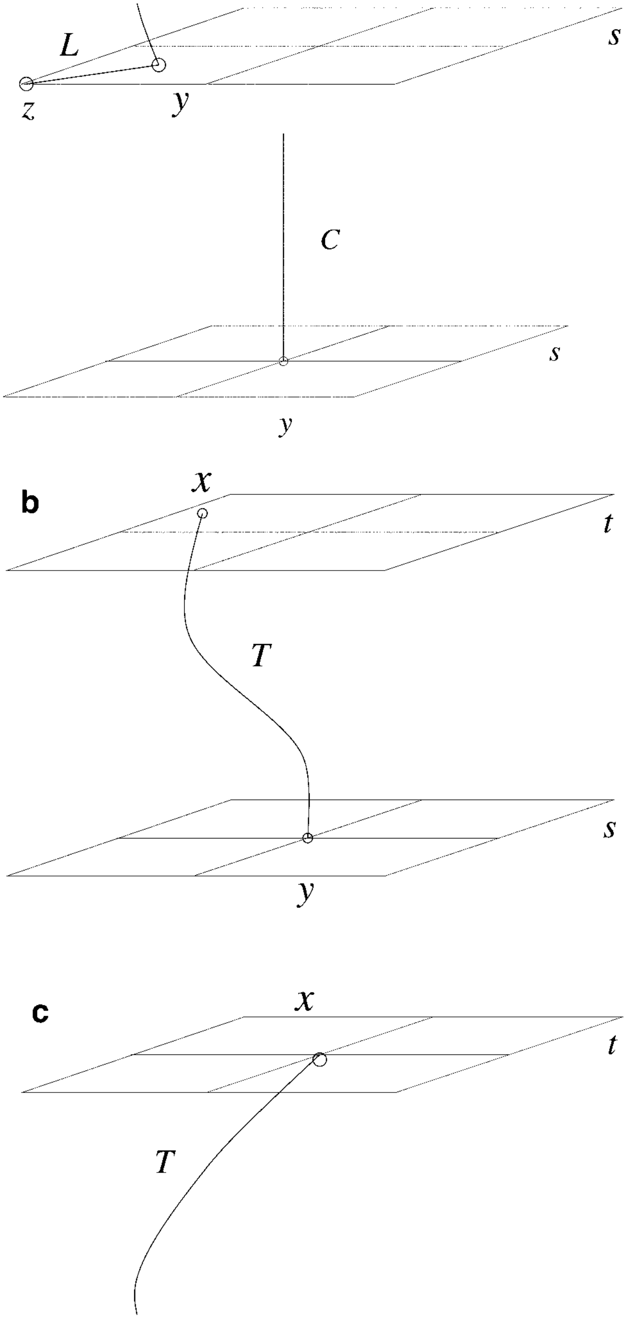

FIG. 5. Spacetime integration paths C for (a) Eulerian, (b) Lagrangian, and (c) semi-Lagrangian schemes.

starting at a grid point y, as in Fig. 5b. Since $dx + F dt = 0$ on T, Lagrangian schemes propagate φ values unchanged along T, assuring unconditional stability:

$$\varphi(x, t) = \varphi(x(t), t) = \varphi(y, s). \tag{36}$$

The main drawback of Lagrangian schemes is that a regular mesh rapidly distorts, losing discretization accuracy. This mesh distortion has been a long-standing problem in 2-D vortex methods, solved in [35].

Semi-Lagrangian schemes combine the regular mesh of an Eulerian scheme with the unconditional stability of a Lagrangian scheme. They build values of φ at regular mesh points x at time t by running a Lagrangian trajectory T backwards from (x, t) to some point (y, s), then a simple path L from the nearest grid point z at time s to y, as in Fig. 5c. Since $dt = 0$ on L and $dx + F dt = 0$ on T, we have

$$\varphi(x, t) = \varphi(z, s) + \int_L \nabla\varphi \cdot dx = \varphi(y, s). \tag{37}$$

Thus semi-Lagrangian schemes need only transport the φ evaluation from x to y, either by interpolation or advection.

2.6.2. *Interpolation or Advection*

Many semi-Lagrangian schemes can be derived by *interpolating* $\varphi(y, s)$ from known grid values, as in the CIR scheme of Subsection 2.5. Linear interpolation gives unconditional stability with first-order accuracy, while higher-order accurate polynomial interpolation can be unstable. Shape-preserving interpolation methods have been compared in [25], and some of these methods yield stable schemes for advection.

Stability issues are eliminated in [31] by re-examining the integral expression

$$\varphi(y, s) = \varphi(z, s) + \int_L \nabla\varphi \cdot dx. \tag{38}$$

This integral transports the evaluation point of φ from z to y and can therefore be viewed as *linear advection* with constant velocity parallel to $y - z$. The advantage of this viewpoint is that monotone Eulerian advection schemes generate stable semi-Lagrangian schemes: there is no CFL time step restriction since y and z are less than half a mesh size apart. Alternatively, L can be built from line segments parallel to coordinate axes, giving naturally split semi-Lagrangian schemes from one-dimensional Eulerian advection schemes. Viewing interpolation as advection can also be reversed, yielding shape-preserving interpolation from Eulerian advection schemes [30].

2.6.3. *Discretization*

Specific semi-Lagrangian schemes usually approximate trajectories by a second-order accurate ordinary differential equation solver such as the implicit mid-point rule

$$y = x + (t - s) F\left(\frac{1}{2}(x + y), \frac{1}{2}(t + s)\right), \tag{39}$$

with F values interpolated—or advected—from the grid points. Equation (39) is nonlinear, but fixed point iteration is proved convergent in [12, 24] if the weak non-intersection condition

$$(t - s)\|DF\| < 1 \tag{40}$$

is satisfied. Semi-Lagrangian schemes are intended for computing smooth solutions without shocks, but it is shown in [12] that—even for Lipschitz solutions—the accuracy of these schemes is limited only by trajectory smoothness, not by solution smoothness.

Given second-order accurate trajectories, a second-order semi-Lagrangian scheme can be built on third-order interpolation methods or Eulerian advection schemes [29]. Spurious oscillations are common with high-order polynomial interpolation, making shape-preserving interpolation and monotone advection preferable. In this paper, we implement first-order CIR time stepping with arbitrary-order ENO interpolation [15] to provide spatial accuracy without spurious oscillations. We plan to implement second-order trajectory calculation in future work to reduce the dissipation evident in a few of our numerical experiments.

3. A SEMI-LAGRANGIAN METHOD FOR MOVING INTERFACES

3.1. Overview of the Method

We use semi-Lagrangian time stepping schemes to solve the level set equation

$$\varphi_t - F \cdot \nabla\varphi = 0. \tag{41}$$

Here F is a velocity field on $\mathbf{R}^d$ which extends VN off $\Gamma(t)$ and may depend on anything: φ, N, C, other derivatives of φ, nonlocal terms, jump conditions, history terms, and so forth. The combination of level sets and semi-Lagrangian time stepping schemes yields a family of methods parametrized by several options. After an overview of these methods, we discuss each option in detail and explain how it contributes to convergence.

3.1.1. *Algorithm*

Given the level set function $\varphi(x, t_n)$ for every point x in a uniform grid at time t_n, our methods compute $\varphi(x, t_{n+1} = t + k)$ at each grid point x by the CIR scheme:

- ○ Evaluate the extended velocity $F(x, t_n)$ at x.
- ○ Optionally postprocess F with truncation and smoothing.
- ○ Move x backwards with velocity $-F(x, t_n)$ to get the point

$$s = x + kF(x, t_n). \tag{42}$$

- ○ Interpolate or advect $\varphi(x, t_n)$ to the point s to get $\varphi(x, t_{n+1}) = \varphi(s, t_n)$.
- ○ Redistance φ if desired, by replacing φ by the signed distance to its zero set.

3.1.2. *Features*

Methods of this family have several unique features:

○ Each new mesh value is a completely independent computation. This allows easy parallel implementation and—more importantly—simplifies construction of adaptive meshes which concentrate computational effort near the interface. Thus the cost of going up a dimension is eliminated.

○ Adaptive mesh refinement does not globally restrict the time step because the time step is decoupled from the CFL stability condition by the unconditional stability of CIR.

○ Adaptive mesh refinement criteria are easy to formulate because we are computing an approximate distance to the interface, which naturally determines refinement. No derivative estimates are necessary.

These methods are implemented on a tree mesh in [37] and combined with fast tree-based redistancing and extension techniques in [38] to yield a general, efficient, and modular method for moving interfaces.

3.2. Options

This family of methods can be parametrized by choosing the following options:

○ The φ interpolation or advection technique which obtains $\varphi(s, t_n)$ at off-grid points s.

○ The velocity evaluation technique which builds $F(x, t_n)$. This may require differentiation and interpolation in the geometric case, or solution of a PDE or integral equation in the general case. A general extension technique may be used, or a problem-dependent extension may be built.

○ Postprocessing of F and φ for stability and accuracy: for some problems such as curvature flows, the optional postprocessing consisting of velocity truncation and smoothing and redistancing at every step appears to be mandatory for convergence.

○ Boundary conditions required when the projected point s falls outside the domain where φ is known.

3.2.1. *Interpolation of φ*

Each evaluation of $\varphi(x, t_{n+1})$ requires interpolation or advection to obtain φ values off the grid. There are infinitely many interpolation techniques, but our choice is restricted by two requirements. First, the level set function φ is only Lipschitz continuous in general since faceting may occur. Thus high-order polynomial interpolation requiring smooth φ should be avoided. Second, stability of the semi-Lagrangian approach in any given norm is guaranteed only for interpolation techniques which do not increase the norm too much. For example, linear interpolation, shape-preserving interpolation [25], and monotone advection [31] guarantee unconditional max-norm stability.

Given these two requirements, essentially non-oscillatory (ENO) interpolation [15] provides sufficient stability and arbitrary-order accuracy. ENO does not guarantee unconditional stability as linear interpolation would, but gives excellent results in practice. Thus we use ENO interpolation and differentiation throughout this paper.

In one dimension, ENO is designed to reduce the variation of the interpolant by sliding the usual polynomial interpolation stencil to minimize differences. In two dimensions, one coordinate direction is chosen first and the stencil slides in that direction. Each stencil value is computed by one-dimensional ENO in the other direction. See Fig. 6 for an example. This choice breaks $x - y$ symmetry, giving a useful error indicator: inaccurate computations become unsymmetric.

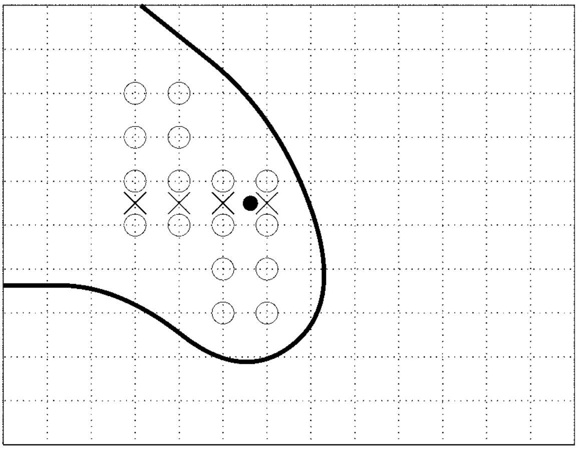

FIG. 6. A possible stencil for third-order ENO interpolation to the solid point s: Open dots indicate mesh points in the stencil, crosses fictitious points for interpolation in the x variable, and the curve is avoided by the ENO stencil because across it differences of the interpolated function are large.

3.2.2. *Velocity Evaluation*

Velocity evaluation may require various problem-dependent computations involving φ, derivatives of φ, and possibly other data. For extending the velocity in PDE problems, we plan to use the general velocity extension of [38]. It redistances efficiently at every step and requires the velocity only on the interface, decoupling the level set method from the velocity computation on $\Gamma(t)$ and permitting the modular solution of moving interface problems for PDEs. For the passive transport and geometric flows computed in Section 4, we use the natural velocity extensions of Subsection 2.2, truncated and smoothed away from the interface for numerical convenience. The following additional procedures are required for geometric flows.

Differentiation of φ. We compute derivatives of φ by optionally smoothing φ once, then differentiating the ENO interpolant to φ. Smoothing is helpful when the interface is faceted or highly complex, because φ is Lipschitz continuous with corners at the facets (as in Fig. 2 above) and unsmoothed ENO differentiation can be inaccurate at corners.

Truncation. The curvature and normal have singularities when $\nabla\varphi = 0$, so we truncate geometric velocity fields away from $\Gamma(t)$. We scale the velocity vector F away from $\Gamma(t)$ so that its maximum norm over the set $\{|\varphi| > 2h\}$ is equal to its maximum norm over the set $\{|\varphi| \leq 2h\}$. Thus large F values near singularities cannot corrupt the solution.

Velocity smoothing. Differentiating the smoothed ENO interpolant to φ produces accurate normal vectors but noisy curvature, because φ is only Lipschitz continuous; hence we smooth curvature-dependent velocities. Each smoothing pass replaces each velocity value by the arithmetic mean of the 3^d nearest values. This commits $O(h^2)$ error in each step, so the total error due to smoothing at any fixed time is $O(h^2/k) = O(h)$ if the consistency condition $k \geq O(h)$ is satisfied. Thus this smoothing technique matches well with the first-order CIR scheme. Higher-order smoothing can be used with a higher-order time stepping scheme. Figure 7 shows smoothing of an anisotropic velocity field for moving a faceted interface, with and without φ and velocity smoothing.

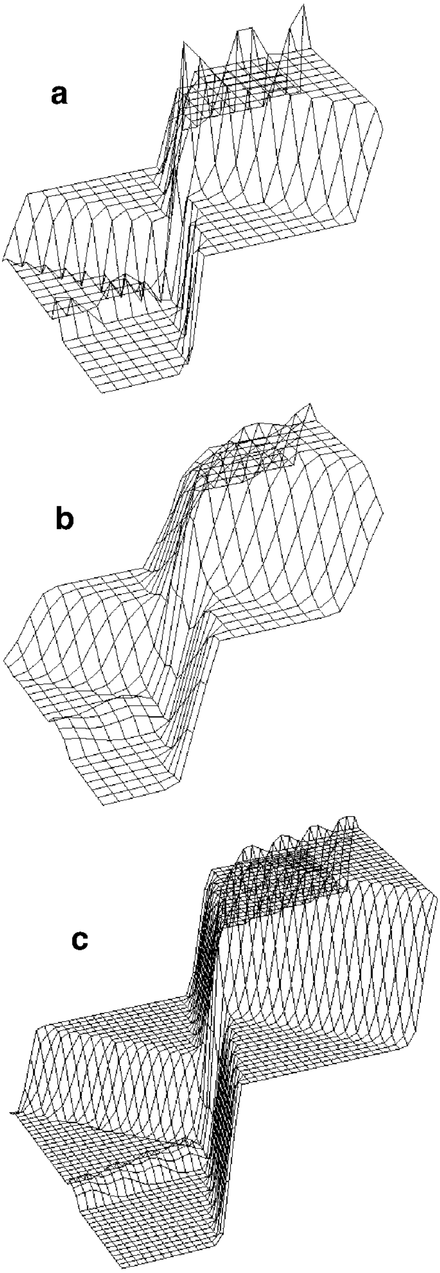

FIG. 7. The x-component F_x of the triangular velocity field $F = (1 + \cos(3\theta + 0.3)/2)N$ where $\theta = \varphi_x/\|\nabla\varphi\|$ is the angle between the normal vector and the x-axis and φ is the hexagonal signed distance function of Fig. 2. Here F_x is computed with degree-1 ENO interpolation and differentiation, and plotted (a) unsmoothed on a 20^2 mesh, (b) after one smoothing pass on a 20^2 mesh, and (c) after one smoothing pass on a 40^2 mesh.

3.2.3. *Redistancing*

The level set equation for moving interfaces—unlike a general PDE—is relevant only near the zero set of the solution. As a consequence, we can re-initialize or "redistance" the solution at any time, by replacing it with the exact signed distance function to its zero

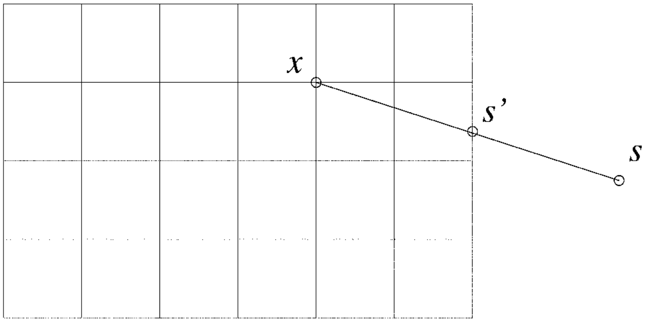

FIG. 8. Boundary conditions implemented by projection.

set. Redistancing is expensive if done naively, but several fast schemes are available [1, 5, 36, 39]. After piecewise-linear contouring of φ, for example, the Voronoi diagram of the resulting polygonal interface can be built in theoretically optimal time [44] and yields almost instantaneous redistancing by standard optimal search techniques [16]. A simplified Voronoi diagram [19] can yield the same result with considerably lower conceptual complexity; however, implementations are not yet available.

Redistancing can be viewed as a form of filtering which eliminates many numerical issues while preserving the interface. For example, boundary conditions far from the interface become much less important because their effect is discarded after redistancing. Redistancing also simplifies geometric velocities: when φ is a signed distance function, $\|\nabla\varphi\| = 1$ near $\Gamma(t)$, so N and C simplify to $\nabla\varphi$ and $\Delta\varphi$.

3.2.4. *Boundary Conditions*

Semi-Lagrangian schemes require numerical boundary conditions to specify values for $\varphi(s, t_n)$ when s lies outside the domain D covered by the grid. There are two simple boundary conditions: extension and projection. In extension, we extend φ as a constant or linear function along lines normal to the boundary ∂D and apply our standard interpolation scheme to interpolate the extended values to s. In projection, we arrest s as it leaves the domain and use one-sided interpolation to the point where s crosses ∂D. Figure 8 shows projection in action: if the point s from Eq. (42) falls outside the domain, then the value of φ is interpolated to s' and $\varphi(x, t_n + k) = \varphi(s', t_n)$.

Our method uses projection because it is simple, effective, and it combines well with ENO schemes which adapt automatically to one-sided interpolation. The combination of projection with truncation, smoothing, and redistancing proved highly effective in our numerical examples. Further research into boundary conditions might be useful in solving parabolic problems like curvature flow where information enters the domain at high speed.

3.3. Convergence

Semi-Lagrangian time-stepping schemes are ideal for solving level set equations, because they promise optimal efficiency via easy adaptive mesh refinement and unrestricted time steps. To fulfill this promise, they must converge to the correct solution near the interface. The following heuristics—and the experiments of Section 4—suggest that these schemes should converge.

3.3.1. *Absence of shocks*

Semi-Lagrangian schemes converge for Lipschitz continuous solutions of advection equations [12], but diverge when shock discontinuities are present [17]. This poses no problem for level set equations, which—like advection equations in atmospheric science—have no shocks. Indeed, the solution φ must remain Lipschitz continuous at all times, or we cannot extract the zero set $\Gamma(t)$. Lipschitz continuity can be rigorously proven for passive transport and some geometric problems [10] and guaranteed in general by redistancing φ at every step.

Given that φ remains Lipschitz continuous, it is easy to see why semi-Lagrangian schemes should work: At a shock, φ would be discontinuous, so a tiny error in velocity would make the trajectory look the wrong way and commit an $O(1)$ error in φ, followed by F; hence shocks would move at the wrong speed. A Lipschitz continuous φ has "kinks" or corners at worst rather than discontinuities, so a small velocity error causes a small solution error.

3.3.2. *The CFL Condition*

The CFL condition requires that a convergent numerical scheme must propagate information about solution values at approximately the right speed, and usually restricts the time step. Our goal in applying semi-Lagrangian schemes to moving interface problems is to satisfy the CFL condition *without* restricting the time step. For interfaces undergoing passive transport, we have linear advection where semi-Lagrangian schemes converge [12], so the CFL condition is satisfied. For geometric problems involving curvature, the level set equation becomes parabolic and information propagates along the interface with infinite speed. Even so, our methods can satisfy the CFL condition as $k = O(h) \to 0$ for the following heuristic reasons.

Nonlocal velocity computation. The domain of dependence of the CIR solution $\varphi(x, t_{n+1})$ obviously includes the single interpolation point $s = x + kF(x, t_n)$ and its stencil, but the point s in turn depends on the φ values used to compute the extended velocity $F(x, t_n)$. Thus the CFL condition can be satisfied in principle by computing F nonlocally with arbitrarily large time steps. For PDE-type moving interface problems F is almost always a global functional of φ, so the CFL condition is satisfied.

From a theoretical point of view, if the solution is continuous and the problem has a maximum principle, each new solution value is exactly equal to some old solution value: define a velocity field F to point to that old value. This highly nonlocal velocity satisfies the CFL condition with any time step.

Velocity smoothing. A specific nonlocal technique which satisfies the CFL condition is to postprocess the velocity field by smoothing or averaging it over a sufficiently large stencil. Accuracy can be maintained by increasing stencil size only logarithmically as $h \to 0$. In practice, a few passes of smoothing produces convergent solutions even though curvature flow velocities give parabolic level set equations, for which explicit schemes usually require $k = O(h^2)$.

Redistancing. Replacing φ by the signed distance to its zero set $\Gamma(t)$ also implements long-distance information transfer and helps satisfy the CFL condition. While redistancing propagates information primarily normal to the interface, its influence is enhanced in regions of high curvature such as corners where normal vectors cross near the interface: these are also the regions where propagation speeds are highest. Frequent redistancing also removes many of the other inconvenient numerical artifacts of the level set method, such as boundary conditions and treatment of singularities.

Velocity extension. For general moving interface problems, the velocity F is known only on $\Gamma(t)$ and must be extended to $\mathbf{R}^d$. Typically F is extended as a constant normal to $\Gamma(t)$ [1, 5, 38, 39], propagating information along the same paths as redistancing and satisfying the CFL condition in the same way.

Modularity. Since a major design goal of our method is modularity—the moving interface code should have minimal information about the velocity-interface relationship—these postprocessing techniques should maintain modularity while satisfying the CFL condition. Nonlocal velocity computation and smoothing inhibit modularity, while the combination of redistancing and velocity extension respects it.

4. NUMERICAL RESULTS

We study the accuracy of our semi-Lagrangian level set method on several interfaces moving under passive transport and geometric motion with corners, anisotropy, nontrivial topology, and curvature. Some PDE-type examples with a general velocity extension [38] will be treated in future work.

Unless otherwise noted, all the examples were computed with the following numerical parameters.

○ Third-order ENO was used for both the φ interpolation and the velocity computation (in geometric moving interface problems where V requires derivatives of φ).

○ Three runs were made with 40, 80, and 160 time steps on a 40^2, 80^2, and 160^2 mesh. Most plots superimpose the three runs to demonstrate convergence to graphical accuracy.

○ For curvature-dependent problems, the velocity was truncated and smoothed once per step, and φ was redistanced at every step to ensure the CFL condition was satisfied.

The method was implemented for two-dimensional level set equations in Standard C, compiled with the SunSoft C compiler using the -fast flag, and run on one CPU of a 2-CPU 200 MHz Sun Ultra-2 under Solaris 2.6.

4.1. Passive Transport

Passive transport problems form convenient test cases for level set methods, because complex exact solutions can easily be evaluated. Thus we can measure the error and rate of convergence. We carry out convergence studies for three passive transport problems and verify the accuracy, robustness, and conservation properties of the CIR scheme with ENO interpolation of degrees 1, 2, and 3.

4.1.1. *Bubbles in a Shear Flow*

We begin our study of passive transport by measuring the accuracy of the method on the collection of circular bubbles shown in Fig. 9, moving with a divergence-free linear shearing velocity

$$F(x, y) = \frac{1}{2}\left(x - 3y + 1, -y - \frac{1}{2}\right). \tag{43}$$

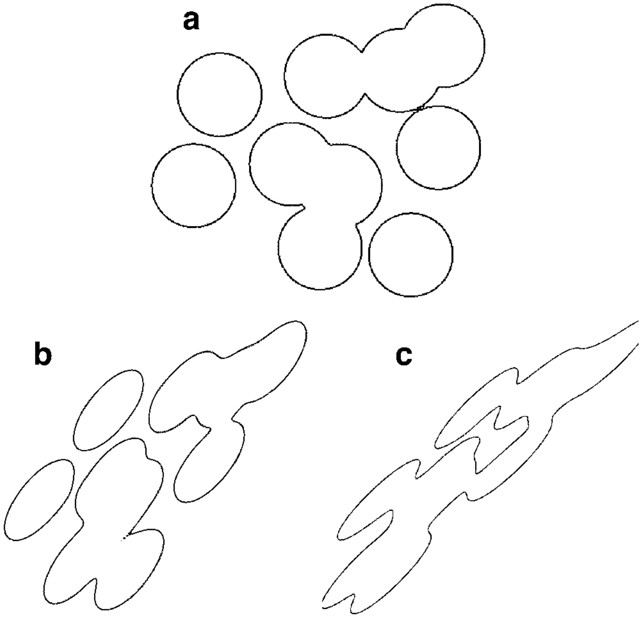

FIG. 9. A collection of bubbles moving with linear shearing velocity.

We used 20, 40, 80, and 160 time steps on $0 \le t \le 1$ and $40^2, 80^2, 160^2$, and 320^2 grids on $[-6, 6] \times [-6, 6]$ (see Table I). ENO interpolation of degrees 1, 2, and 3 was used to interpolate φ. Table I reports the maximum of the exact distance function on the computed contour at time $t = 1$. First-order accuracy is clearly evident along diagonals, where

TABLE I

Grid	$N_T = 20$	40	80	160
		ENO degree 1		
40^2	0.342	0.551	0.756	0.55
80^2	0.0428	0.15	0.235	0.353
160^2	0.00628	0.00868	0.0677	0.231
320^2	0.019	0.00351	0.00467	0.0294
		ENO degree 2		
40^2	0.0938	0.13	0.102	0.0911
80^2	0.0126	0.0389	0.104	0.145
160^2	0.022	0.00967	0.00183	0.0272
320^2	0.0238	0.0116	0.00536	0.00163
		ENO degree 3		
40^2	0.00708	0.122	0.188	0.193
80^2	0.018	0.00562	0.0431	0.0519
160^2	0.0226	0.0103	0.00408	0.00189
320^2	0.0239	0.0117	0.00555	0.00249

Note. Maximum error at $t = 1$ in the interface shown in Fig. 9, moving with divergence-free linear shearing velocity $F(x, y) = \frac{1}{2}(x - 3y + 1, -y - \frac{1}{2})$, computed with N_T time steps of the CIR scheme with ENO interpolation of degrees 1, 2, and 3. The domain is $[-6, 6]^2$.

$h \le O(k)$. This agrees with the one-dimensional model theory of Section 2. The error decreases dramatically when we change from ENO degree 1 to degree 2, but degree 3 makes no further improvement.

4.1.2. *Grid Effects on Triangles*

A common problem in moving interfaces is sensitive dependence on numerical artifacts such as grid orientation. We check for grid effects in a sharply faceted interface by revolving, shrinking, and expanding a triangle with a linear velocity field. In all cases, the interface moves with the appropriate speed independently of its orientation relative to the grid. Figure 10 plots the results with both second and third-order degree ENO on the domain $[-2, 2]^2$ and shows that grid effects are minimal. The dissipation exhibited in Fig. 10a could be considerably reduced by second-order trajectory computation.

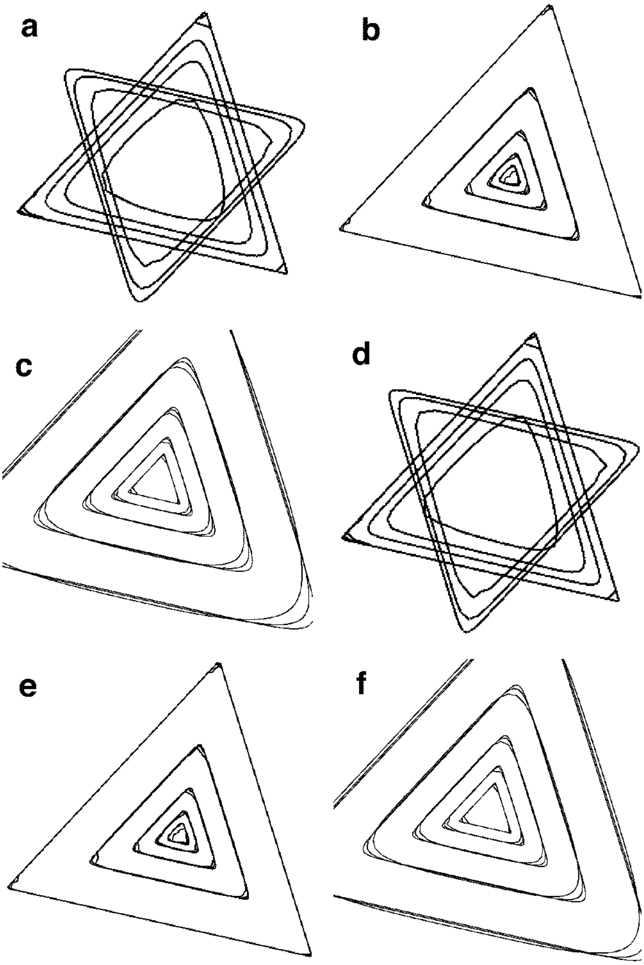

FIG. 10. Tests of grid effects in sharp corners with linear velocity field. (a) A rotating triangle at a half period and a full period, computed with degree-2 ENO. (b) A triangle shrinking with $V(x, y) = \frac{-5}{2}(x, y)$ from $t = 0$ to $t = 1$. (c) A triangle expanding with $V(x, y) = 2(x, y)$ from $t = 0$ to $t = 1$. Plots (d) through (f) show the same calculation with degree-3 ENO.

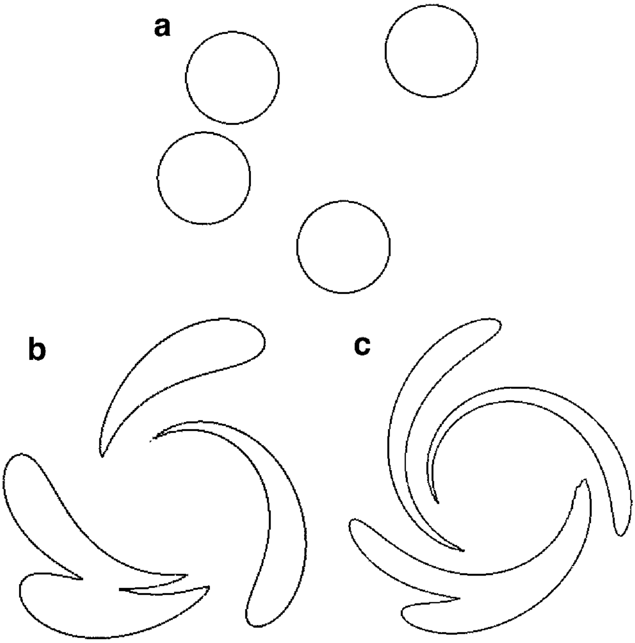

FIG. 11. A collection of circular bubbles under a divergence-free shearing velocity.

4.1.3. *Mass Conservation in a Shear Flow*

We conclude our study of passive transport by measuring mass conservation in a collection of bubbles moving in the divergence-free shearing flow given by

$$F(x, y) = \frac{\max(1 - (1 - x^2 - y^2)_+^4, 0)}{8(x^2 + y^2)}(-y, x). \tag{44}$$

Figure 11 shows the extreme distortion produced by this flow, computed with 160 time steps on $0 \le t \le 100$ and a 160^2 mesh on the domain $[-6, 6]^2$. Despite this distortion, mass is well conserved; the final area inside the computed interface is 12.4669, close to the exact value of $4\pi = 12.5664$.

In the exact solution interfaces cannot touch, because of standard uniqueness theorems for ordinary differential equations. Thus merging of computational interfaces can happen even when it is impossible in theory, and must be allowed for in any robust moving interface method. Automatic handling of unexpected topological changes is one of the strengths of the level set approach.

4.2. Geometry

We validate our semi-Lagrangian moving interface method by computing converged solutions to a variety of geometric moving interface problems including viscosity solutions to corners moving with unit normal velocity, the faceted Wulff limit for anisotropic normal velocity fields, complex topological changes under anisotropic curvature-dependent flows, and nonconvex shapes shrinking to round points under flow by curvature. Moving interface problems for PDEs require a general velocity extension but display little additional complexity and will be solved in future.

TABLE II

Grid	$N_T = 20$	40	80	160
	ENO degree 1			
20^2	0.0589	0.0634	0.0657	0.0668
40^2	0.0159	0.0179	0.019	0.0197
80^2	0.00557	0.00647	0.00696	0.00722
160^2	0.00119	0.00128	0.00137	0.00142
	ENO degree 2			
20^2	0.0077	0.00801	0.00814	0.0082
40^2	0.0014	0.00146	0.00148	0.00148
80^2	0.000456	0.000481	0.000488	0.000489
160^2	0.0000768	0.000078	0.0000792	0.0000795
	ENO degree 3			
20^2	0.00185	0.00194	0.00198	0.00199
40^2	0.000658	0.000669	0.000672	0.000674
80^2	0.000346	0.000349	0.00035	0.000351
160^2	0.0000724	0.0000725	0.0000726	0.0000726

Note. Maximum of exact distance function at $t = 1$ on a circle of radius $R(t) = 1 + t$ and center $(1/2\pi, 1/2\pi)$, moving with constant normal velocity $V = 1$, computed with N_T time steps of the CIR scheme with ENO interpolation of degrees 1, 2, and 3.

4.2.1. *Unit Normal Velocity*

We verify first-order accuracy on a unit circle centered at $(1/2\pi, 1/2\pi)$, expanding with unit normal velocity $F = N$, extended naturally via Eq. (13) with singularities truncated;

$$F = N = \frac{\nabla\varphi}{\max(10^{-8}, \|\nabla\varphi\|)}. \tag{45}$$

Table II reports the maximum of the exact distance function on the computed contour at time $t = 1$, with 20, 40, 80, and 160 time steps on $0 \le t \le 1$ and 20^2, 40^2, 80^2, and 160^2 grids on $[-3, 3]^2$. ENO interpolation of degrees 1, 2, and 3 was used both in the φ interpolation and in the evaluation of N. Considerably better than first-order accuracy is evident along diagonals, where $h \le O(k)$, because the exact interface is a linear function of t.

4.2.2. *Viscosity Solutions with Corners*

One of the most important issues in level set equations is the correct computation of "viscosity solutions" for faceted interfaces in geometric and PDE problems [27]. A key ingredient in this computation is a corner moving in or out with unit normal velocity. Inward motion should keep corners sharp (the "shock" case), while outward motion should produce rounded corners due to Huygens' principle (the "rarefaction" case), as discussed in Subsection 2.1.2. Figure 12 shows a triangle moving with positive and negative unit normal velocity, both aligned with the mesh and at an angle to check for grid effects, and demonstrates that our semi-Lagrangian method computes the correct viscosity solution in each case.

This agrees with theory: any reasonable computed normal has unit length, so our method propagates information at unit speed. An incorrect solution typically preserves a sharp corner moving outward, rather than rounding it off as prescribed by Huygens' Principle: Figure 13 illustrates the difference. CIR produces the correct solution because zero φ values

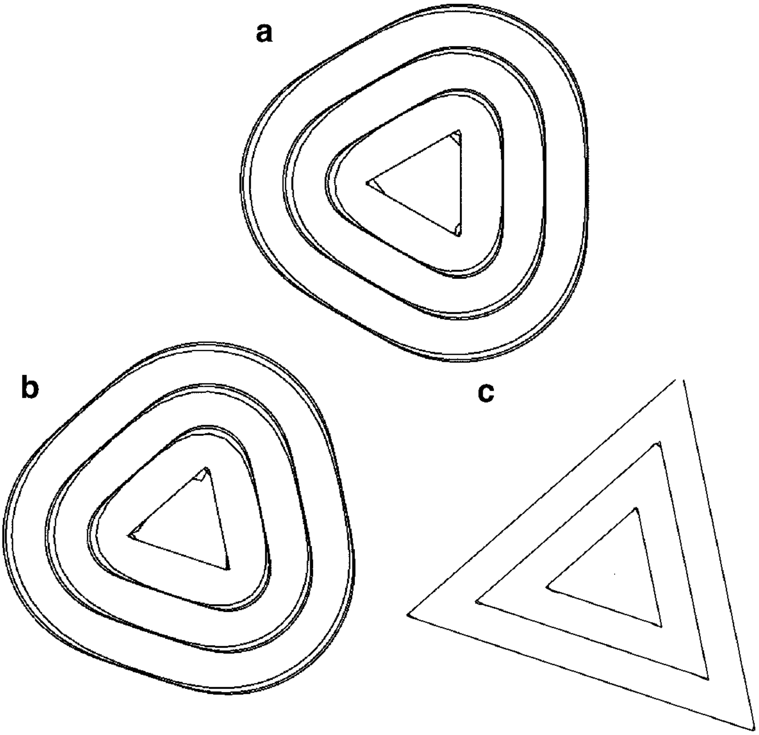

FIG. 12. Viscosity solutions for triangles moving with positive or negative unit normal velocity. (a) An expanding triangle at zero angle to the mesh, with round corners. (b) An expanding triangle at angle 0.2 radians to the mesh, with round corners. (c) A shrinking triangle at angle 0.2 radians to the mesh, with sharp corners.

delineating $\Gamma(1)$ near the corner must be located on a unit circle centered somewhere on $\Gamma(0)$, rather than $\sqrt{2}$ from $\Gamma(0)$, as they are in the incorrect solution.

4.2.3. *Anisotropic Normal Velocity and the Wulff Limit*

Another key issue for level set methods is anisotropic motion along the normal. Most numerical methods for level set equations are connected to the theory of Hamilton–Jacobi

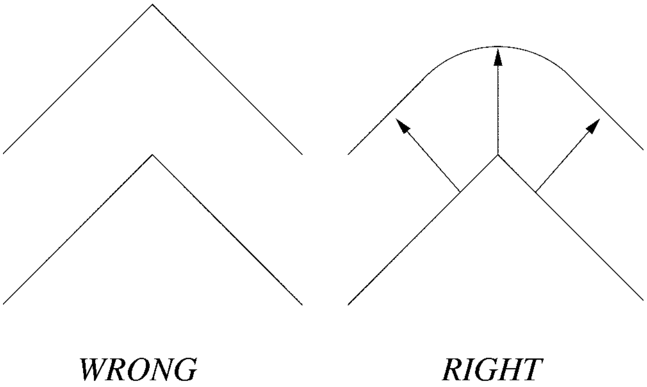

FIG. 13. Right and wrong propagation of corners under unit normal velocity.

equations

$$\varphi_t + H(\nabla\varphi) = 0, \tag{46}$$

which encounters difficulties when the Hamiltonian H is nonconvex. For anisotropic normal velocities

$$V = R + \epsilon\cos(k\theta), \qquad \cos\theta = \varphi_x/\|\nabla\varphi\|, \tag{47}$$

the Hamiltonian is nonconvex if

$$R + \epsilon(1 - k^2) < 0 < R - |\epsilon|, \tag{48}$$

causing some Hamilton–Jacobi methods to break down.

In Fig. 14, we evolve an initially circular interface under several anisotropic normal velocities, producing nonconvex Hamiltonians. The interface converges rapidly to the "Wulff shape" [23, 42, 43] corresponding to each given anisotropy, as predicted by rigorous theory [21]. The faceted Wulff shape is a natural limit, since portions of the interface with normal vectors not aligned along minima of the velocity will grow faster, causing facets to develop. In Fig. 15, we begin from a highly nonconvex initial interface, producing a severe test of the method. The asymptotic Wulff shape is still computed accurately. The small grid-dependence which remains could likely be removed with a second-order accurate trajectory computation.

These computations were smoothed and their convergence improved by applying one pass of smoothing to φ before ENO differentiation, one pass to F after differentiation, and redistancing φ at every step. This emphasizes an essential reasons why the CIR scheme

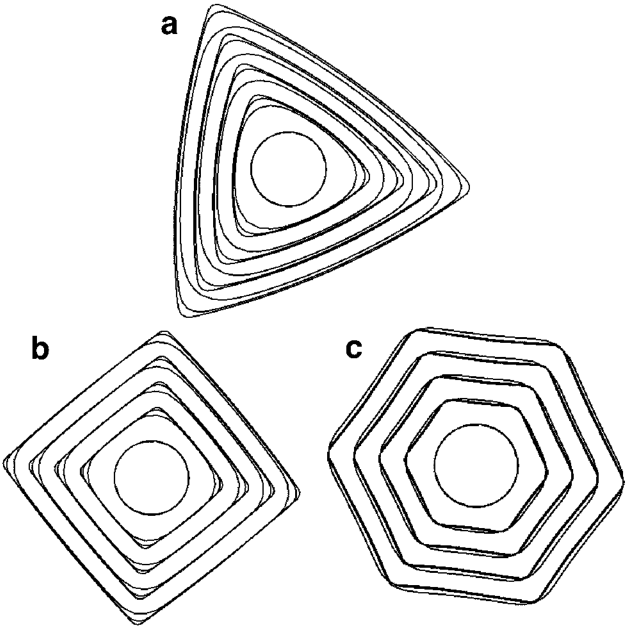

FIG. 14. Wulff shapes growing from circular initial interfaces (with radius 1/2 and center at $(1/2\pi, 1/2\pi)$). Here $0 \le t \le 1$ and the domain is $[-3, 3]^2$.

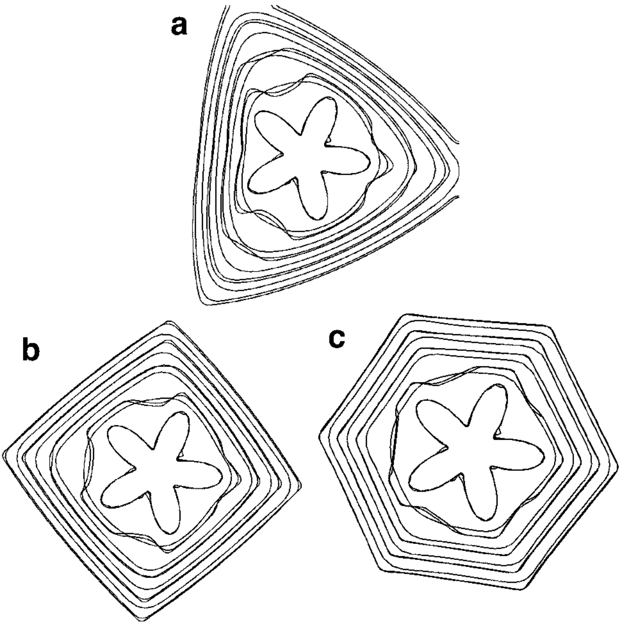

FIG. 15. Wulff shapes developing from nonconvex initial interfaces (given by $\sqrt{(2\pi x - 1)^2 + (2\pi y - 1)^2} = 2\pi(0.8 + 0.4\cos(5\xi))$ where $\tan\xi = (2\pi y - 1)/(2\pi x - 1)$). Here $0 \le t \le 1$ and the domain is $[-3, 3]^2$.

works for level set equations: We are free to modify φ and F away from $\Gamma(t)$ to suit numerical convenience—or to satisfy the CFL condition.

4.2.4. *Merging under Anisotropy*

Starting from a collection of randomly placed, sized, and oriented trefoil shapes, we move the interface along its normal with a threefold anisotropic speed $V = 2 + \cos(3\theta + 0.3)$, where θ is the angle between the normal vector and the positive x-axis. This motion involves considerable topological complexity, which is correctly computed by the level set approach. Figure 16 shows that even this highly nonconvex initial interface is also aproaching the asymptotic triangular Wulff shape as $t \to \infty$.

4.2.5. *Circles under Curvature*

A circle shrinking with normal velocity equal to its curvature has exact radius $R(t) = \sqrt{R(0)^2 - 2t}$, so with $R(0) = 2$ a circle should collapse to a point in time $0 \le t \le 2$. A smaller circle with $R(0) = 1$ vanishes in time $t = 0.5$. Figure 17 shows convergence to graphical accuracy, computed with 20, 40, 80, 160 time steps on 20^2, 40^2, 80^2, 160^2 grids and plotted every 0.2 time units.

A convenient measure of convergence is the extinction time—the first time when the interface completely vanishes. For the four runs shown, the extinction time is 1.1, 1.5, 1.73, and 1.85, displaying slow but smoothly monotone first-order convergence to the correct value 2. The extinction time is difficult to resolve because it depends sensitively on the movement of the interface as it vanishes. Even with smoothing, our computed velocity always moves the interface faster than the exact velocity.

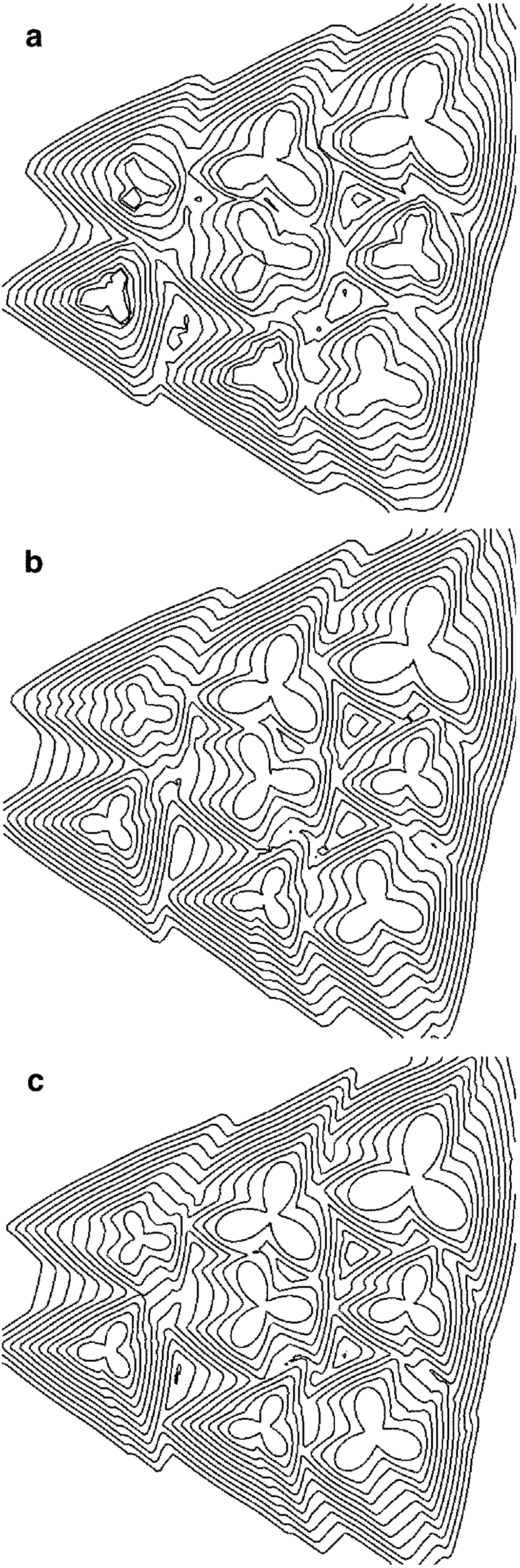

FIG. 16. A collection of randomly located, sized, and oriented trefoils growing and merging under anisotropic normal velocity $V = 2 + \cos(3\theta + 0.3)$. Here our method used third-order ENO with (a) 40 time steps on a 40^2 mesh, (b) 80 steps on an 80^2 mesh, and (c) 160 steps on an 160^2 mesh to achieve convergence to graphical accuracy.

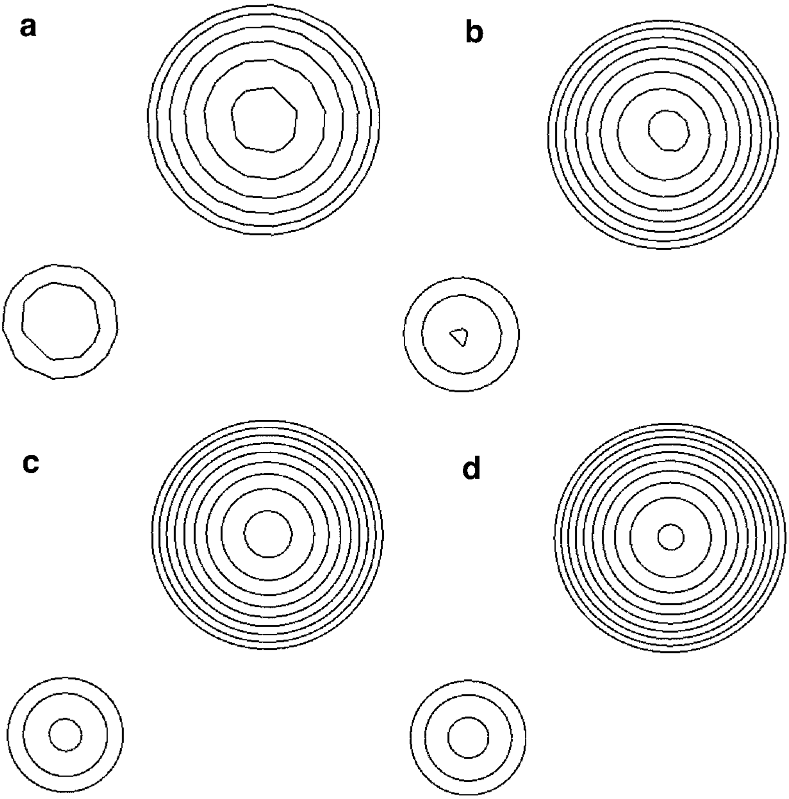

FIG. 17. Convergence of two circles collapsing under curvature flow $V = C$, computed from $t = 0$ to the extinction times $t = 1/2$ and $t = 2$ with third-order ENO on (a) 20 time steps on 20^2 grid covering $[-4, 4]^2$ with 1 velocity smoothing pass per step, (b) 40 time steps on 40^2 grid covering $[-4, 4]^2$ with 2 passes per step, (c) 80 steps on 80^2 grid with 3 passes, (d) 160 steps on 160^2 grid with 4 passes.

For this parabolic problem, velocity smoothing and truncation, φ smoothing, and frequent redistancing all contribute to convergence of the CIR scheme as $k \to 0$ with $k = O(h)$. As discussed in Subsection 3.3.2, they all play a role in satisfying the parabolic CFL condition with these unusually large time steps. We truncated the velocity away from the interface at each step, smoothed the truncated velocity once per step on the 20^2 mesh, twice per step on the 40^2 mesh, and so forth. The resulting logarithmic increase in stencil width as the mesh size goes to zero satisfies the CFL condition. We smoothed φ once before ENO differentiation, to compute derivatives of nonsmooth φ values. We also redistanced φ from the interface at the end of every step, a highly nonlocal information transfer which also helps satisfy the CFL condition. These smoothing and redistancing options were chosen after some experimentation and constitute the minimum postprocessing required to achieve convergence.

4.2.6. *Nonconvex Interfaces under Curvature*

We verified that randomly placed, sized, and oriented nonconvex trefoil shapes collapse under curvature flow to round points, as predicted by a geometric theorem [14]. Figure 18 shows results for $0 \leq t \leq 1/2$ on $[-4, 4]^2$, with one velocity smoothing pass, one φ smoothing pass, and one redistancing per step. Experiments showed that this rather small amount of smoothing sufficed for convergence to graphical accuracy.

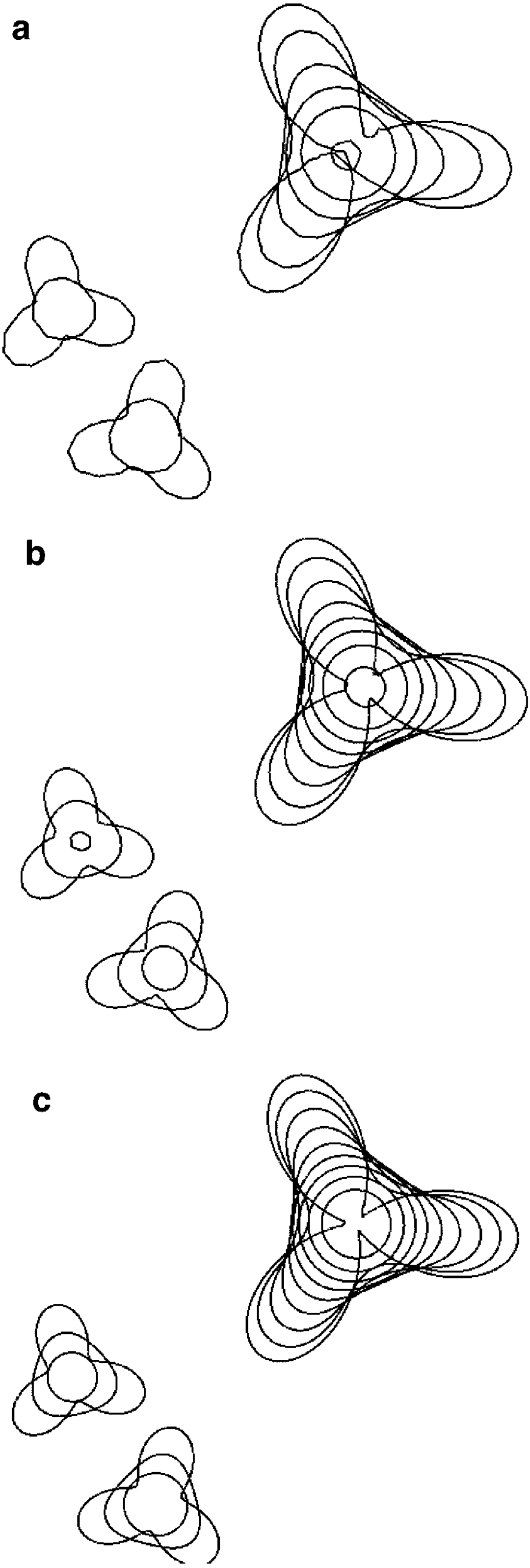

FIG. 18. Convergence of a collection of trefoils to round points under curvature flow $V = C$, computed from $t = 0$ to $t = 1/2$ with third-order ENO on (a) 40 time steps on 40^2 grid covering $[-4, 4]^2$, (b) 80 steps on 80^2 grid, (c) 160 steps on 160^2 grid.

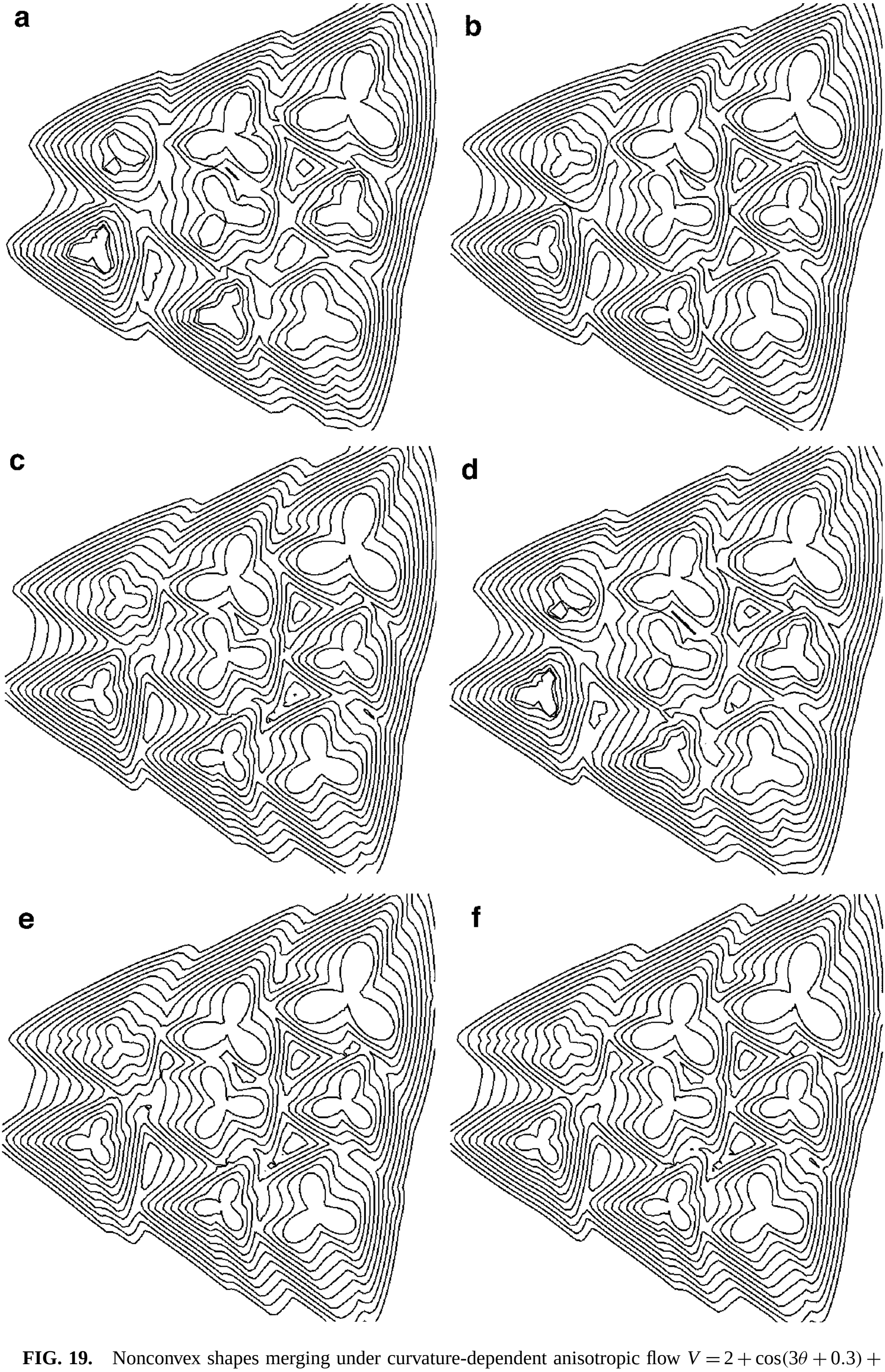

FIG. 19. Nonconvex shapes merging under curvature-dependent anisotropic flow $V = 2 + \cos(3\theta + 0.3) + \epsilon C$. Convergence to the viscosity solution as $\epsilon \to 0$ is demonstrated with $\epsilon = 0.1$ (a–c) and $\epsilon = 0.01$ (d–f); cf. Fig. 16 for the limit case $\epsilon = 0$.

4.2.7. *Nonconvex Interfaces Merging under Anisotropy Plus Curvature*

Finally, we demonstrate topological complexity in the viscosity limit, with a curvature-smoothed velocity

$$V = 2 + \cos(3\theta + 0.3) + \epsilon C. \tag{49}$$

We illustrate the limit $\epsilon \to 0$ computationally with $\epsilon = 0.1$ and 0.01, carrying out a convergence study for each value of ϵ separately. Figure 19 shows the results, which converge rapidly to the results shown in Fig. 16. We used one velocity smoothing pass, one φ smoothing pass, and one redistancing per step.

4.3. Convergence

These numerical experiments have consistently demonstrated that our semi-Lagrangian methods converge with appropriate problem-dependent truncation, smoothing, and redistancing options. Our methods converge without options for passive transport and constant normal velocity. When anisotropy or curvature is present, redistancing plus one to four passes of velocity smoothing must be applied at each step to ensure convergence. These conclusions agree with the heuristics of Subsection 3.3 and show that CFL timestep restrictions can be eliminated—even for curvature-dependent parabolic problems!

5. CONCLUSION

We have described and validated new numerical methods for moving interfaces, based on semi-Lagrangian time stepping schemes for level set equations. We presented heuristic arguments and experimental evidence showing these methods work well for difficult moving interface problems involving merging, faceting, transport, and anisotropic curvature-dependent geometry.

These methods have unique capabilities

- to move interfaces with appropriate time steps unconstrained by numerical stability issues,
- to decouple each mesh point from the others, allowing easy adaptive mesh refinement, and
- potentially to decouple the velocity computation from the moving interface, allowing convenient modular solution of a vast spectrum of moving interface problems.

Our ultimate goal is a “black-box” method for moving interfaces, which can accept the interface and its velocity at time t and return the evolved interface one time step later. Such a method can simplify the solution of moving interface problems, because the moving interface code need not change when the physical problem off the interface is modified.

Planned future research on these methods includes

- further analysis of CFL conditions for parabolic problems with infinite propagation speed,
- adaptive modular methods [38],
- second-order accurate time stepping,
- CAD geometry input and contouring with NURBS, and
- applications to industrial crystal growth problems, where the moving interface is coupled to complex materials science.

ACKNOWLEDGMENT

The author thanks the reviewers for many helpful suggestions.

REFERENCES

1. D. A. Adalsteinsson and J. A. Sethian, *The Fast Construction of Extension Velocities in Level Set Methods*, Technical Report PAM-738, UC Berkeley Center for Pure and Applied Mathematics, 1997.
2. R. J. Almgren, Variational algorithms and pattern formation in dendritic solidification, *J. Comput. Phys.* **106**, 337 (1993).
3. P. Bartello and S. J. Thomas, The cost-effectiveness of semi-Lagrangian advection, *Monthly Weather Rev.* **124**, 2883 (1996).
4. L. N. Brush and R. F. Sekerka, A numerical study of two-dimensional growth forms in the presence of anisotropic growth kinetics, *J. Crystal Growth* **96**, 419 (1989).
5. S. Chen, B. Merriman, S. Osher, and P. Smereka, A simple level set method for solving Stefan problems, *J. Comput. Phys.* **135**, 8 (1997).
6. R. Courant, E. Isaacson, and M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, *Comm. Pure Appl. Math.* **5**, 243 (1952).
7. M. G. Crandall, H. Ishii, and P. L. Lions, User's guide to viscosity solutions of Hamilton–Jacobi equations, *Bull. Am. Math. Soc.* **27**, 1 (1992).
8. R. Dobrushin, R. Kotecky, and S. Shlosman, Wulff construction: A global shape from local interaction, in *Am. Math. Soc. Translations of Mathematical Monographs* (Am. Math. Soc., Providence, RI, 1992), Vol. 104.
9. J. Duchon and R. Robert, Evolution d'une interface par capillarite et diffusion de volume I. Existence locale en temps, *Ann. Inst. H. Poincare* **1**, 361 (1984).
10. L. C. Evans and J. Spruck, Motion of level sets by mean curvature, 1, *J. Differential Geom.* **33**, 635 (1991).
11. M. Falcone, The minimum time problem and its applications to front propagation, in *Motion by Mean Curvature and Related Topics*, edited by A. Visintin and G. Buttazzo (De Gruyter, Berlin, 1994).
12. M. Falcone and R. Ferretti, Convergence analysis for a class of high-order semi-Lagrangian schemes, *SIAM J. Numer. Anal.* **35**, 909 (1998).
13. M. Falcone, T. Giorgi, and P. Loreti, Level sets of viscosity solutions and applications, *SIAM J. Appl. Math.* **54**, 1335 (1994).
14. M. A. Grayson, The heat equation shrinks embedded plane curves to round points, *J. Differential Geom.* **26**, 285 (1987).
15. A. Harten, S. Osher, B. Engquist, and S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, *J. Comput. Phys.* **71**, 231 (1987).
16. D. Kirkpatrick, Optimal search in planar subdivisions, *SIAM J. Comput.* **12**, 28 (1983).
17. R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser, Basel, 1990).
18. P. A. Makar and S. R. Karpik, Basis-spline interpolation on the sphere: Application to semi-Lagrangian advection, *Monthly Weather Rev.* **124**, 182 (1996).
19. M. McAllister, D. Kirkpatrick, and J. Snoeyink, A compact piecewise-linear Voronoi diagram for convex sites in the plane, *Discrete Comput. Geom.* **15**, 73 (1996).
20. W. W. Mullins and R. F. Sekerka, Morphological stability of a particle growing by diffusion or heat flow, *J. Appl. Phys.* **34**, 323 (1963).
21. S. Osher and B. Merriman, The Wulff shape as the asymptotic limit of a growing crystal interface, *Asian J. Math.* **1**, 560 (1997).
22. S. J. Osher and J. A. Sethian, Front propagation with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
23. D. Peng, S. Osher, B. Merriman, and H. Zhao, *The Geometry of Wulff Crystal Shapes and Its Relations with Riemann Problems*, CAM Report 98-51, University of California, Los Angeles, Program in Computational and Applied Mathematics, 1998.

24. J. Pudykiewicz and A. Staniforth, Some properties and comparative performance of the semi-Lagrangian method of Robert in the solution of the advection-diffusion equation, *Atmos. Ocean.* **22**, 283 (1984).

25. P. J. Rasch and D. L. Williamson, On shape-preserving interpolation and semi-Lagrangian transport, *SIAM J. Sci. Stat. Comput.* **11**, 656 (1990).

26. A. Schmidt, Computation of three dimensional dendrites with finite elements, *J. Comput. Phys.* **125**, 293 (1996).

27. J. Sethian, *Level Set Methods* (Cambridge Univ. Press, Cambridge, UK, 1996).

28. J. A. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98**, 231 (1992).

29. P. K. Smolarkiewicz and W. W. Grabowski, The multidimensional positive definite advection transport algorithm: Nonoscillatory option, *J. Comput. Phys.* **86**, 355 (1990).

30. P. K. Smolarkiewicz and G. A. Grell, A class of monotone interpolation schemes, *J. Comput. Phys.* **101**, 431 (1992).

31. P. K. Smolarkiewicz and J. Pudykiewicz, A class of semi-Lagrangian approximations for fluids, *J. Atmos. Sci.* **49**, 2082 (1992).

32. P. K. Smolarkiewicz and P. J. Rasch, Monotone advection on the sphere—An Eulerian versus semi-Lagrangian approach, *J. Atmos. Sci.* **48**, 793 (1991).

33. A. Staniforth and J. Côté, Semi-Lagrangian schemes for atmospheric models—A review, *Monthly Weather Rev.* **119**, 2206 (1991).

34. J. Strain, A boundary integral approach to unstable solidification, *J. Comput. Phys.* **85**, 342 (1989).

35. J. Strain, Fast adaptive 2D vortex methods, *J. Comput. Phys.* **132**, 108 (1997).

36. J. Strain, Efficient redistancing in level set methods, *J. Comput. Phys.*, in press.

37. J. Strain, Tree methods for moving interfaces, *J. Comput. Phys.*, in press.

38. J. Strain, Modular methods for moving interfaces, *J. Comput. Phys.*, in press.

39. M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).

40. J. Taylor, J. W. Cahn, and C. A. Handwerker, Geometric models of crystal growth, *Acta Met. Mat.* **40**, 1443 (1992).

41. C. Truesdell and R. A. Toupin, The classical field theories, in *Handbuch der Physik III/1*, edited by S. Flügge (Springer-Verlag, Berlin, 1960).

42. D. P. Woodruff, *The Solid-Liquid Interface* (Cambridge Univ. Press, Cambridge, UK, 1973).

43. G. Wulff, Zur Frage der Geschwindigkeit des Wachstums und der Auflösung der Krystallflachen, *Zeit. Krystall. Min.* **34**, 449 (1901).

44. C. K. Yap, An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments, *Discrete Comput. Geom.* **2**, 365 (1987).