Acta Numerica (1996), *pp.* 309–395 © Cambridge University Press, 1996

Theory, algorithms, and applications of level set methods for propagating interfaces

James A. Sethian *
Department of Mathematics
University of California
Berkeley, CA 94720, USA
E-mail: sethian@math.berkeley.edu

We review recent work on level set methods for following the evolution of complex interfaces. These techniques are based on solving initial value partial differential equations for level set functions, using techniques borrowed from hyperbolic conservation laws. Topological changes, corner and cusp development, and accurate determination of geometric properties such as curvature and normal direction are naturally obtained in this setting. The methodology results in robust, accurate, and efficient numerical algorithms for propagating interfaces in highly complex settings. We review the basic theory and approximations, describe a hierarchy of fast methods, including an extremely fast marching level set scheme for monotonically advancing fronts, based on a stationary formulation of the problem, and discuss extensions to multiple interfaces and triple points. Finally, we demonstrate the technique applied to a series of examples from geometry, material science and computer vision, including mean curvature flow, minimal surfaces, grid generation, fluid mechanics, combustion, image processing, computer vision, and etching, deposition and lithography in the microfabrication of electronic components.

* Supported in part by the Applied Mathematics Subprogram of the Office of Energy Research under DE-AC03-76SF00098, and the National Science Foundation DARPA under grant DMS-8919074.

CONTENTS

1. Introduction

Propagating interfaces occur in a wide variety of settings. As physical entities, they include ocean waves, burning flames, and material boundaries. Less obvious boundaries are equally important, and include shapes against backgrounds, hand-written characters, and iso-intensity contours in images.

The goal of this paper is to describe some recent work on level set methods, which attempts to unify these problems and provide a general framework for modelling the evolution of boundaries. Our aim is to review a collection of state-of-the-art details of computational techniques for tracking moving interfaces, and to give some sense of the flavour and breadth of applications.

Level set methods are numerical techniques that offer remarkably powerful tools for understanding, analysing, and computing interface motion in a host of settings. At their core, they rely on a fundamental shift in how one views moving boundaries; rethinking the Lagrangian geometric perspective and exchanging it for an Eulerian, initial value partial differential equation perspective. Five clear advantages result from this new view of propagating interfaces.

1 From a theoretical/mathematical point of view, the real complexities of front motion are illuminated, in particular, the role of singularities, weak solutions, shock formation, entropy conditions and topological change in the evolving interface.

2 From a numerical perspective, natural and accurate ways of computing delicate quantities emerge, including the ability to build high-order advection schemes, compute local curvature in two and three dimensions, track sharp corners and cusps, and handle subtle topological changes of merger and breakage.

3 From an implementation point of view, since the approach is based on an initial value partial differential equation, robust schemes result from numerical parameters set at the beginning of the computation. The error is thus controlled by

- the order of the numerical method
- the grid spacing Δh
- the time step Δt.

4 Computational adaptivity, both in meshing and in computational labour, is possible, as is a clear path to parallelism.

5 For monotonically advancing fronts obeying certain speed laws, we introduce exceptionally fast methods based on narrow band techniques and sorting algorithms.

In this paper, we survey an illustrative subset of past and current applications of level set methods. By no means is this an exhaustive review. A large body of work has been reluctantly skipped in the effort to keep this paper of reasonable length. The interested reader is referred to the many references given throughout the text.

PART I: LEVEL SET FORMULATIONS

2. Theory of front evolution

Consider a boundary, either a curve in two dimensions or a surface in three dimensions, separating one region from another. Imagine that this curve/surface moves in its normal direction with a known speed function F. Our goal is to track the motion of this interface as it evolves. We are only concerned with the motion of the interface in its normal direction: throughout, we shall ignore tangential motions of the interface.

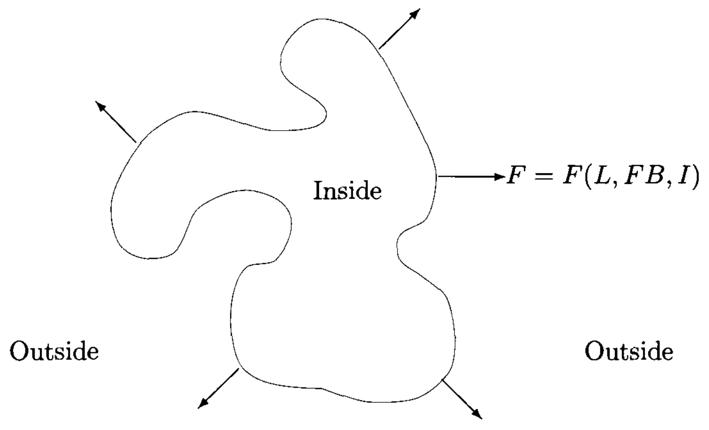


Fig. 1. Curve propagating with speed F in normal direction.

The speed function F can be thought of as depending on three types of arguments, namely

$$F = F(L, G, I), \tag{2.1}$$

where

L = Local properties of the front are those determined by local geometric information, such as curvature and normal direction.

G = Global properties of the front such as integrals along the front and associated differential equations, are those whose solutions depend on the shape and position of the front. For example, suppose the interface is a source of heat affecting diffusion on either side of the interface, which in turn influences the motion of the interface. This would be characterized as global argument.

I = Independent properties are those that are independent of the shape of the front, such as an underlying fluid velocity which passively transports the front.

Much of the challenge in interface problems comes from producing an adequate model for the speed function F; this is a separate issue independent of the goal of an accurate scheme for advancing the interface based on the model for F. In this section, we assume that the speed function F is known. In Part V, we discuss the development of models for a collection of applications.

Our first goal is to develop the necessary theory to understand the interplay between the speed function F and the shape of the interface. For ease of discussion, we now turn to the simplest case of a closed curve propagating in the plane.

2.1. Fundamental formulation

Let γ be a smooth, closed initial curve in R^2, and let $\gamma(t)$ be the one-parameter family of curves generated by moving $\gamma = \gamma(t = 0)$ along its normal vector field with speed F. Here, F is the given scalar function. Thus, we have that $\vec{n} \cdot \vec{x}_t = F$, where $\vec{x}$ is the position vector of the curve, t is time, and $\vec{n}$ is the unit normal to the curve.

As a first attempt, a natural approach is to consider a parametrized description of the motion. We further restrict ourselves and imagine that the speed function F depends only on the local curvature κ of the curve, that is, $F = F(\kappa)$. Thus, we let the position vector $\vec{x}(s,t)$ parametrize γ at time t. Here, $0 \le s \le S$, and we assume periodic boundary conditions $\vec{x}(0,t) = \vec{x}(S,t)$. The curve is parametrized so that the interior is on the left in the direction of increasing s (see Fig. 2). Let $\vec{n}(s,t)$ be the parametrization of the outward normal and $\kappa(s,t)$ be the parametrization of the curvature. The equations of motion can then be written in terms of individual components $\vec{x} = (x,y)$ as

$$x_t = \frac{y_s F\left(\frac{y_{ss}x_s - x_{ss}y_s}{(x_s^2+y_s^2)^{3/2}}\right)}{(x_s^2+y_s^2)^{1/2}}, \qquad y_t = -\frac{x_s F\left(\frac{y_{ss}x_s - x_{ss}y_s}{(x_s^2+y_s^2)^{3/2}}\right)}{(x_s^2+y_s^2)^{1/2}}, \tag{2.2}$$

where we have used the parametrized expression for the curvature $\kappa = (y_{ss}x_s - x_{ss}y_s)(x_s^2+y_s^2)^{-3/2}$ inside the speed function $F(\kappa)$. This is a 'Lagrangian' representation because the range of $(x(s,t), y(s,t))$ describes the moving front.

2.2. Total variation: stability and the growth of oscillations

What happens to oscillations in the initial curve as it moves? We summarize the argument in Sethian (1985) showing that the decay of oscillations depends only on the sign of F_κ at $\kappa = 0$. The metric $g(s,t)$, which measures the

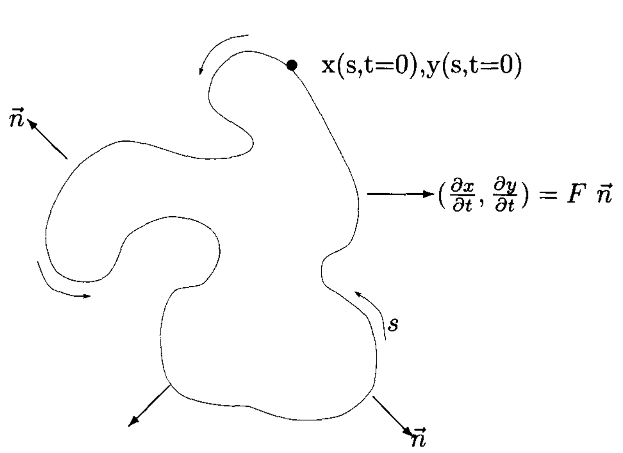

x(s,t=0),y(s,t=0)
$\vec{n}$
$(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}) = F\,\vec{n}$
s
$\vec{n}$

Fig. 2. Parametrized view of propagating curve.

'stretch' of the parametrization, is given by $g(s,t) = (x_s^2 + y_s^2)^{1/2}$. Define the total oscillation (also known as the total variation) in the front

$$Var(t) = \int_0^S |\kappa(s,t)| g(s,t)\, \mathrm{d}s. \tag{2.3}$$

This measures the amount of 'wrinkling'. Our goal is to find out if this wrinkling increases or decreases as the front evolves (see Fig. 3).

Differentiation of both the curvature and the metric with respect to time, together with substitution from equation (2.2) produces the corresponding evolution equations for the metric and curvature, namely

$$\kappa_t = -g^{-1}(F_s g^{-1})_s - \kappa^2 F \tag{2.4}$$

$$g_t = g\kappa F \tag{2.5}$$

(Here, g^{-1} is $1/g$, not the inverse). Now, suppose we have a non-convex initial curve moving with speed $F(\kappa)$, and suppose the moving curve stays smooth. By evaluating the time change of the total variation in the solution, we have the following (see Sethian (1985)).

Theorem Consider a front moving along its normal vector field with speed $F(\kappa)$, as in equation (2.2). Assume that the initial curve $\gamma(0)$ is smooth and non-convex, so that $\kappa(s,0)$ changes sign. Assume that F is twice differentiable, and that $\kappa(s,t)$ is twice differentiable for $0 \le s \le S$ and $0 \le t \le T$. Then, for $0 \le t \le T$,

- if $F_\kappa \le 0$ ($F_\kappa \ge 0$) wherever $\kappa = 0$, then

$$\frac{d\mathrm{Var}(t)}{dt} \le 0 \qquad (\frac{d\mathrm{Var}(t)}{dt} \ge 0); \tag{2.6}$$

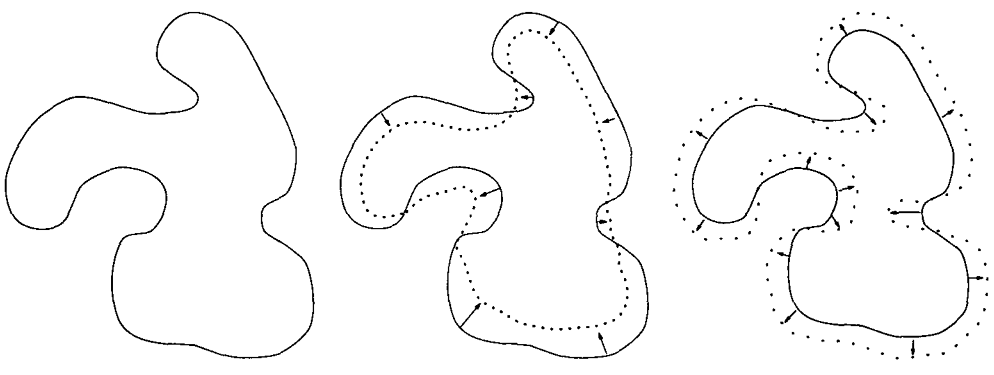

(a) Original curve (b) Decrease in variation (c) Increase in variation

Fig. 3. Change in variation.

- if $F_\kappa < 0$ ($F_\kappa > 0$) and $\kappa_s \neq 0$ wherever $\kappa = 0$, then

$$\frac{d\mathrm{Var}(t)}{dt} < 0 \qquad \left(\frac{d\mathrm{Var}(t)}{dt} > 0\right). \tag{2.7}$$

Remarks The theorem states that if $F_\kappa < 0$ wherever $\kappa = 0$, then the total variation decreases as the front moves and the front 'smooths out', that is, the energy of the front dissipates. The front is assumed to remain smooth in the interval $0 \leq t \leq T$ (the curvature is assumed to be twice differentiable). (In the next section, we discuss what happens if the front ceases to be smooth and develops a corner.) In the special case when $\gamma(t)$ is convex for all t, the theorem is trivial, since $\mathrm{Var}(t) = \int_0^S \kappa g \, ds = 2\pi$. The proof may be found in Sethian (1985).

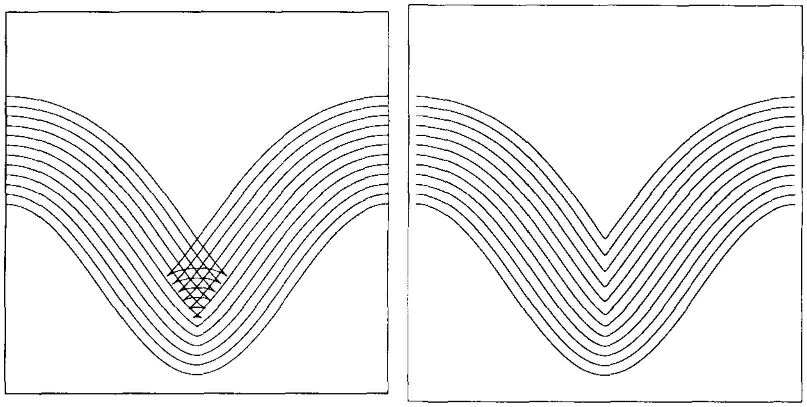
Two important cases can be easily checked. A speed function $F(\kappa) = 1 - \epsilon\kappa$ for ϵ positive has derivative $F_\kappa = -\epsilon$, and hence the total variation decays. Conversely, a speed function of the form $F(\kappa) = 1 + \epsilon\kappa$ yields a positive speed derivative, and hence oscillations grow. We shall see that the sign of the curvature term in this case corresponds to the backwards heat equation, and hence must be unstable.

2.3. The role of entropy conditions and weak solutions

The above theorem assumes that the front stays smooth and differentiable. In many cases of evolving fronts, differentiability is soon lost. For example, consider the periodic initial cosine curve

$$\gamma(0) = (-s, [1 + \cos 2\pi s]/2) \tag{2.8}$$

propagating with speed $F(\kappa) = 1$. The exact solution to this problem at time t may be easily constructed by advancing each point of the front in its normal



(a) Swallowtail ($F = 1.0$) (b) Entropy solution ($F = 1.0$)

Fig. 4. Cosine curve propagating with unit speed.

direction a distance t. In fact, in terms of our parametrization of the front, the solution is given by

$$x(s,t) = \frac{y_s(s,t=0)}{(x_s^2(s,t=0) + y_s^2(s,t=0))^{1/2}}\, t + x(s,t=0), \qquad (2.9)$$

$$y(s,t) = -\frac{x_s(s,t=0)}{(x_s^2(s,t=0) + y_s^2(s,t=0))^{1/2}}\, t + y(s,t=0). \qquad (2.10)$$

In Fig. 4, the solution is given for this propagating cosine curve.

It is clear that the front develops a sharp corner, known as a *shock*, in finite time; however, once this corner develops, it is not at all clear how to construct the normal at the corner and continue the evolution, since the derivative is not defined there. Thus, beyond the formation of the discontinuity in the derivative, we need a *weak solution*, so-called because the solution weakly satisfies the definition of differentiability.

How shall we continue the solution beyond the formation of the singularity in the curvature corresponding to the corner in the front? The correct answer depends on the nature of the interface under discussion. If we regard the interface as a geometric curve evolving under the prescribed speed function, then one possible weak solution is the 'swallowtail' solution formed by letting the front pass through itself; this is the solution shown in Fig. 4a. We note that this solution is in fact the one given by equations (2.9), (2.10); the lack of differentiability at the centre point does not destroy the solution, since we have written the solution in terms of the initial data.

However, if we regard the moving curve as an interface separating two regions, the front at time t should consist of only the set of all points located a distance t from the initial curve. (This is known as the Huygens' principle

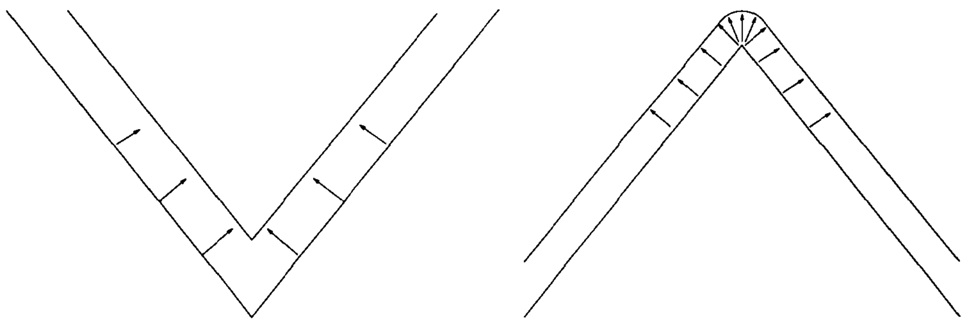

(a) Shock (b) Rarefaction fan

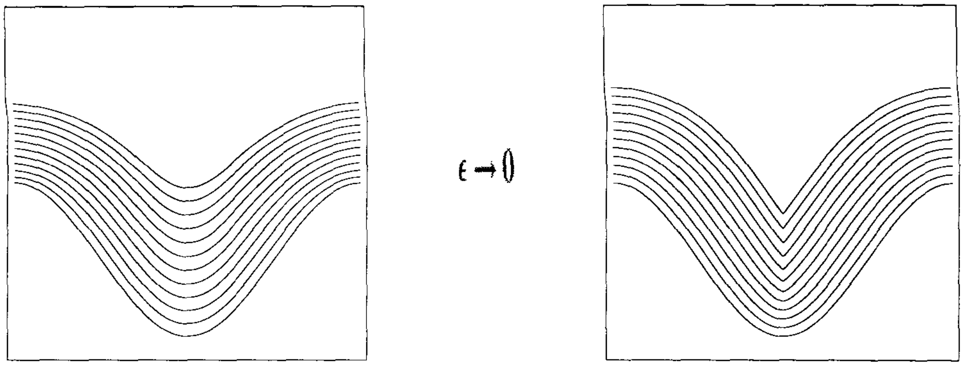
Fig. 5. Front propagating with unit normal speed.

construction.) Roughly speaking, we want to remove the 'tail' from the 'swallowtail'. In Fig. 4b, we show this alternate weak solution. Another way to characterize this weak solution is via the following 'entropy condition' posed by Sethian (1982, 1985): if the front is viewed as a burning flame, then *once a particle is burnt it stays burnt.* Careful adherence to this stipulation produces the Huygens' principle construction.

What does this 'entropy condition' have to do with the notion of 'entropy'? An intuitive answer is that an entropy condition stipulates that no new information can be created during the evolution of the problem. Once an entropy condition is invoked, some information about the initial data is lost. Indeed, our entropy condition says that once a particle is burnt, it stays burnt, that is, once a corner has developed, the solution is no longer reversible. The problem cannot be run 'backwards' in time; if we try to do so, we will not retrieve the initial data. Thus, some information about the solution is forever lost.

As further illustration, we consider the case of a V-shaped front propagating normal to itself with unit speed ($F = 1$). In Fig. 5a, the point of the front is downwards: as the front moves inwards with unit speed, a shock develops as the front pinches off, and an entropy condition is required to select the correct solution to stop the solution from being multiple-valued. Conversely, in Fig. 5b, the point of the front is upwards: in this case the unit normal speed results in a circular fan, which connects the left state with slope $+1$ to the right state, which has slope -1.

It is important to summarize a key point in the above discussion. Our choice of weak solution given by our entropy condition rests on the perspective that the front separates two regions, and the assumption that we are interested in tracking the progress of one region into the other.



(a) $F = 1. - 0.25\kappa$ $\qquad F = 1. - \epsilon\kappa \qquad$ (b) Entropy solution ($F = 1.0$)

Fig. 6. Entropy solution is the limit of viscous solution.

2.4. Effects of curvature: the viscous limit and the link to hyperbolic conservation laws

Consider now a speed function of the form $F = 1 - \epsilon\kappa$, where ϵ is a constant. The modifying effects of curvature on the former constant speed law are profound, and in fact pave the way towards constructing accurate numerical schemes that adhere to the correct entropy condition.

Following Sethian (1985), the curvature evolution equation given by (2.4) can be rewritten in terms of arclength, namely

$$\kappa_t = \epsilon\kappa_{\alpha\alpha} + \epsilon\kappa^3 - \kappa^2, \tag{2.11}$$

where the second derivative of the curvature κ is taken with respect to arclength α. This is a reaction-diffusion equation; the drive toward singularities due to the reaction term $(\epsilon\kappa^3 - \kappa^2)$ is balanced by the smoothing effect of the diffusion term $(\epsilon\kappa_{\alpha\alpha})$. Indeed, with $\epsilon = 0$, we have a pure reaction equation $\kappa_t = -\kappa^2$, and the developing corner can be seen in the exact solution $\kappa(s,t) = \kappa(s,0)/(1 + t\kappa(s,0))$, which is singular in finite t if the initial curvature is anywhere negative. Thus, as shown, corners can form in the moving curve when $\epsilon = 0$.

Consider again the cosine front given in equation (2.8) and the speed function $F(\kappa) = 1 - \epsilon\kappa$, $\epsilon > 0$. As the front moves, the troughs at $s = n + 1/2, n = 0, \pm1, \pm2, \ldots$ are sharpened by the negative reaction term (because $\kappa < 0$ at such points) and smoothed by the positive diffusion term (see Fig. 6a). For $\epsilon > 0$, it can be shown (see Sethian (1985) and Osher and Sethian (1988)) that the moving front stays C^∞. The entropy-satisfying solution to this problem when $F = 1$ from Fig. 4b is shown in Fig. 6b.

The central observation, and key to the level set approach, is the following link.

Consider the above propagating cosine curve and the two solutions:

- $X^{\epsilon}_{\text{curvature}}(t)$, obtained by evolving the initial front with $F_{\epsilon} = 1 - \epsilon\kappa$
- $X_{\text{constant}}(t)$, obtained with speed function $F = 1$ and the entropy condition.

Then, at any time T,

$$\lim_{\epsilon \to 0} X^{\epsilon}_{\text{curvature}}(T) = X_{\text{constant}}(T). \tag{2.12}$$

Thus, the limit of motion with curvature, known as the 'viscous limit', is the entropy solution for the constant speed case, see Sethian (1985).

Why is this known as the 'viscous limit', or, more accurately, what does this have to do with viscosity? In order to see why viscosity is an appropriate name, we turn to the link between propagating fronts and hyperbolic conservation laws.

An equation of the form

$$u_t + [G(u)]_x = 0 \tag{2.13}$$

is known as a hyperbolic conservation law. A simple example is Burger's equation, given by

$$u_t + uu_x = 0, \tag{2.14}$$

which describes the motion of a compressible fluid in one dimension. The solution to this equation can develop discontinuities, known as 'shocks', where the fluid undergoes a sudden expansion or compression. These shocks (for example, a sonic boom) can arise from arbitrarily smooth initial data; they are a function of the equation itself. However, if one includes the effects of fluid viscosity, the equation includes a right-hand side, namely

$$u_t + uu_x = \epsilon u_{xx}. \tag{2.15}$$

This second derivative on the right-hand side acts like a smoothing term and stops the development of such shocks; it can be shown that the solutions must remain smooth for all time.

What does this have to do with our propagating front equation? Consider the initial front given by the graph of $f(x)$, with f periodic on $[0,1]$, and suppose that the propagating front remains a function for all time. Let ψ be the height of the propagating function at time t, thus $\psi(x,0) = f(x)$. The tangent at (x,ψ) is $(1,\psi_x)$. Referring to Fig. 7, the change in height V in a unit time is related to the speed F in the normal direction by

$$\frac{V}{F} = \frac{(1+\psi_x^2)^{1/2}}{1}, \tag{2.16}$$

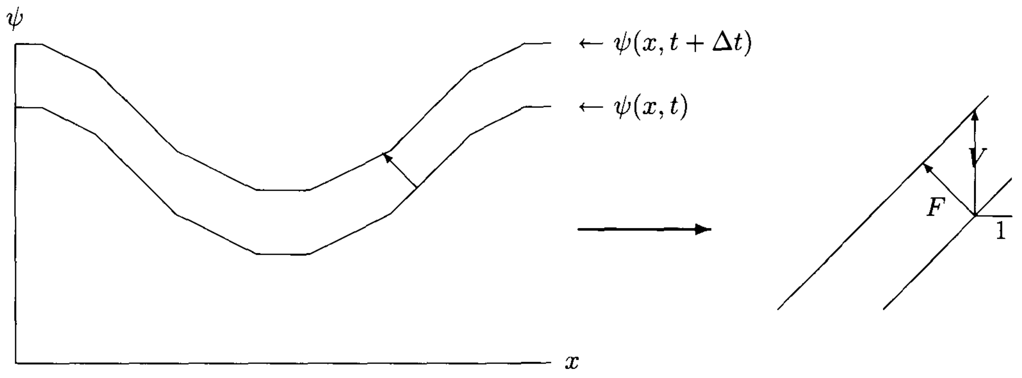


Fig. 7. Variables for propagating graph.

and thus the equation of motion becomes

$$\psi_t = F(1+\psi_x^2)^{1/2}. \tag{2.17}$$

Using the speed function $F(\kappa) = 1-\epsilon\kappa$ and the formula $\kappa = -\psi_{xx}/(1+\psi_x^2)^{3/2}$, we get

$$\psi_t \; - \; (1+\psi_x^2)^{1/2} = \epsilon \frac{\psi_{xx}}{1+\psi_x^2}. \tag{2.18}$$

We first note that this is a partial differential equation with a first-order time and space derivative on the left-hand side, and a parabolic second-order term on the right. Differentiating both sides of this equation yields an evolution equation for the slope $u = d\psi/dx$ of the propagating front, namely

$$u_t + [-(1+u^2)^{1/2}]_x = \epsilon[\frac{u_x}{1+u^2}]_x. \tag{2.19}$$

Thus, the derivative of our equation with parabolic right-hand side for the changing height ψ looks like a viscous hyperbolic conservation law with $G(u) = (1+u^2)^{1/2}$ for the propagating slope u; see Sethian (1987). Hyperbolic conservation laws of the above form have a long history, in fact, our entropy condition is equivalent to the one for propagating shocks in hyperbolic conservation laws. Our goal will be to exploit the theory and technology of numerical solutions of hyperbolic conservation laws to devise accurate numerical schemes to solve the equation of motion for propagating fronts.

Before doing so, however, we have a technical problem. The equation of motion given by equation (2.17) only refers to fronts that remain the graph of a function as they move. The above ideas must be extended to include propagating fronts that are not easily written as functions. This is the time-dependent level set idea introduced by Osher and Sethian (1988).

3. The time-dependent level set formulation

3.1. Formulation

Given a closed $N-1$ dimensional hypersurface $\Gamma(t)$, we now produce an *Eulerian* formulation for the motion of the hypersurface propagating along its normal direction with speed F, where F can be a function of various arguments, including the curvature, normal direction, *etc.* The main idea of the level set methodology is to embed this propagating interface as the zero level set of a higher-dimensional function ϕ. Let $\phi(x, t=0)$, where x is a point in R^N, be defined by

$$\phi(x,t=0) = \pm d, \tag{3.1}$$

where d is the distance from x to $\Gamma(t=0)$, and the plus (minus) sign is chosen if the point x is outside (inside) the initial hypersurface $\Gamma(t=0)$. Thus, we have an initial function $\phi(x,t=0) : R^N \to R$ with the property that

$$\Gamma(t=0) = (x|\phi(x,t=0)=0)\,. \tag{3.2}$$

Our goal is to produce an equation for the evolving function $\phi(x,t)$ that contains the embedded motion of $\Gamma(t)$ as the level set $\phi = 0$. Let $x(t)$ be the path of a point on the propagating front. That is, $x(t=0)$ is a point on the initial front $\Gamma(t=0)$, and $x_t \cdot n = F(x(t))$ where n is the normal to the front at $x(t)$. Since we want the zero level set of the evolving function ϕ to always match the propagating hypersurface, we must have

$$\phi(x(t),t) = 0. \tag{3.3}$$

By the chain rule,

$$\phi_t + \nabla\phi(x(t),t)\cdot x'(t) = 0. \tag{3.4}$$

Since $n = \nabla\phi/|\nabla\phi|$, we have the evolution equation for ϕ, namely

$$\phi_t + F|\nabla\phi| = 0 \tag{3.5}$$

$$\phi(x,t=0) \quad \text{given}. \tag{3.6}$$

This is our time-dependent level set equation. For certain forms of the speed function F, we obtain a standard Hamilton–Jacobi equation.

In Fig. 8, taken from Sethian (1994), we show the outward propagation of an initial curve and the accompanying motion of the level set function ϕ.

In Fig. 8a, we show the initial circle, and in Fig. 8c, we show the circle at a later time. In Fig. 8b, we show the initial position of the level set function ϕ, and in Fig. 8d, we show this function at a later time. We refer to this as an Eulerian formulation because the underlying coordinate system remains fixed.

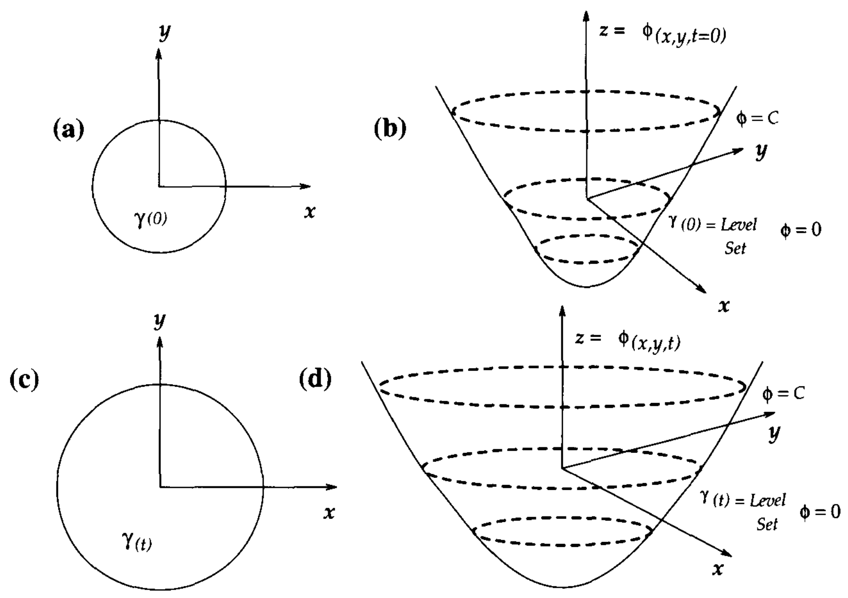


Fig. 8. Propagating circle.

3.2. Advantages

There are four major advantages to this Eulerian level set formulation.

1 The evolving function $\phi(x,t)$ always remains a function as long as F is smooth. However, the level surface $\phi = 0$, and hence the propagating hypersurface $\Gamma(t)$ may change topology, break, merge, and form sharp corners as the function ϕ evolves.

2 The second major advantage concerns numerical approximation. Because $\phi(x,t)$ remains a function as it evolves, we may use a discrete grid in the domain of x and substitute finite difference approximations for the spatial and temporal derivatives. For example, using a uniform mesh of spacing h, with grid nodes (i,j), and employing the standard notation that ϕ_{ij}^n is the approximation to the solution $\phi(ih, jh, n\Delta t)$, where Δt is the time step, we might write

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} + (F)(\nabla_{ij}\phi_{ij}^n) = 0 \tag{3.7}$$

Here, we have used forward differences in time, and let $\nabla_{ij}\phi_{ij}^n$ *be some* appropriate finite difference operator for the spatial derivative. Thus, an explicit finite difference approach is possible.

3 Intrinsic geometric properties of the front may be easily determined from the level set function ϕ. For example, at any point of the front,

the normal vector is given by

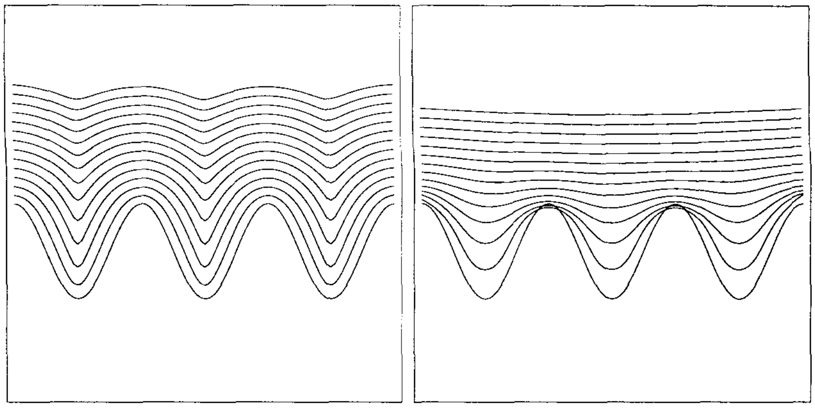
$$\vec{n} = \frac{\nabla \phi}{|\nabla \phi|}, \tag{3.8}$$

and the curvature of each level set is easily obtained from the divergence of the gradient of the unit normal vector to the front, that is,

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}}. \tag{3.9}$$

4 Finally, there are no significant changes required to follow fronts in three dimensions. By simply extending the array structures and gradient operators, propagating surfaces are easily handled.

As an example of the application of level set methods, consider once again the problem of a front propagating with speed $F(\kappa) = 1 - \epsilon\kappa$. In Fig. 9, we show two cases of a propagating initial triple sine curve. For ϵ small (Fig. 9a), the troughs sharpen up and result in transverse lines that come too close together. For ϵ large (Fig. 9b), parts of the boundary with high values of positive curvature can initially move downwards, and concave parts of the front can move quickly upwards.



(a) $F = 1. - 0.025\kappa$ (b) $F = 1. - 0.25\kappa$

Fig. 9. Propagating triple sine curve.

3.3. *Theoretical aspects of the level set formulation*

Numerical techniques for approximating moving fronts have been the focus of much effort in computational physics. At the same time, the theoretical analysis of moving curves and surfaces has been a subject of considerable importance in its own right. The work of Gage (1984), Gage and Hamilton

(1986) and Grayson (1987), discussed later, provided some groundbreaking analysis of flow of curves under the curvature, and the seminal result that a closed curve shrinking under its curvature collapses smoothly to a point.

Along a very different approach, Brakke (1978) applied varifold theory to the problem of a hypersurface moving under its curvature, and in doing so provided a wide-ranging perspective for these problems. His was a general approach, and included problems in which the results were not necessarily smooth. His analysis provided a detailed look at surface curvature evolution problems.

There has been considerable theoretical analysis of the level set approach, its formulation, and its relation to other perspectives on front propagation. The model posed in Sethian (1982), which considered flame propagation as a function of curvature, and introduced an entropy condition for evolving fronts, served as the basis for theoretical analysis by Barles (1985). The full level set methodology of Osher and Sethian (1988) provides a view of surface evolution different from the one provided in the work of Gage, Grayson, and Brakke. First, the embedding of the front as a higher-dimensional function naturally accounts for some of the issues of topological change and corner formation. Second, and more importantly, the transformation of a geometry problem into an initial value partial differential equation means that available technology, including regularity of solutions, viscous solutions of Hamilton–Jacobi equations, and questions of existence and uniqueness, can be applied in this geometrical setting.

Using the above level set approach, Evans and Spruck (1991, 1992*a*, 1992*b*, 1995), and Chen, Giga and Goto (1991), Giga and Goto (1992) and Giga, Goto and Ishii (1992) performed detailed analysis of curvature flow in a series of papers. They exploited much of the work on viscosity solutions of partial differential equations developed over the past 15 years (see Lions (1982)), which itself was inspired by the corresponding work applied to hyperbolic conservation laws. These papers examined the regularity of curvature flow equations, pathological cases, and the link between the level set perspective and the varifold approach of Brakke. These papers opened up a series of investigations into further issues; we also refer the interested reader to Ilmanen (1992, 1994) and Evans, Soner and Souganidis (1992).

3.4. Summary

The discussion in Part I may be summarized as follows:

1 A front propagating at a constant speed can form corners as it evolves; at such points, the front is no longer differentiable and a weak solution must be constructed to continue the solution.

2 The correct weak solution, motivated by viewing the front as an evolving interface separating two regions, comes by means of an entropy condition.
3 A front propagating at a speed that depends on its curvature does not form corners and stays smooth for all time. Furthermore, the limit of this motion as the dependence on curvature vanishes is the entropy-satisfying solution obtained for the constant speed case.
4 If the propagating front remains a graph as it moves, there is a direct link between the equation of motion and one-dimensional hyperbolic conservation laws. The role of curvature in a propagating front is analogous to the role of viscosity in equations of viscous compressible fluid flow.
5 By embedding the motion of a curve as the zero level set of a higher-dimensional function, an initial value partial differential equation can be obtained which extends the above to include arbitrary curves and surfaces moving in two and three space dimensions.

4. The stationary level set formulation

In the above level set equation

$$\phi_t + F|\nabla\phi| = 0 \tag{4.1}$$

the position of the front is given by the zero level set of ϕ at a time t. Suppose we now restrict ourselves to the particular case of a front propagating with a speed F that is either always positive or always negative. In this case, we can convert our level set formulation from a time-dependent partial differential equation to a stationary one, in which time has disappeared. We now describe a stationary level set formulation, which is common in control theory.

To explain this transformation, imagine the two-dimensional case in which the interface is a propagating curve, and suppose we graph the evolving zero level set above the xy plane. That is, let $T(x, y)$ be the time at which the curve crosses the point (x, y). The surface $T(x, y)$ then satisfies the equation

$$|\nabla T|F = 1. \tag{4.2}$$

Equation (4.2) simply says that the gradient of arrival time surface is inversely proportional to the speed of the front. This is a Hamilton–Jacobi equation, and the recasting of the a front motion problem into a stationary one is common in a variety of applications; see Falcone (1994) and Falcone, Giorgi and Loretti (1994). In the case where the speed function F depends only on position, we get the well-known Eikonal equation. The requirement that the speed function always be positive* is so that the crossing time surface $T(x, y)$ is single-valued.

* or conversely, always negative

To summarize,

- In the time-dependent level set equation, the position of the front Γ at time t is given by the zero level set of ϕ at time t, that is $\Gamma(t) = \{(x,y) : \phi(x,t) = 0\}$.
- In the stationary level set equation, the position of the front Γ is given by the level set $\Gamma(t) = \{(x,y) : T(x,y) = t\}$.

That is, we wish to solve

Time-dependent formulation	**Stationary formulation**
$\phi_t + F\|\nabla\phi\| = 0$	$\|\nabla T\|F = 1$
Front= $\Gamma(t) = \{(x,y) : \phi(x,t) = 0\}$	Front= $\Gamma(t) = \{(x,y) : T(x,y) = t\}$
Applies for arbitrary F	Requires $F > 0$.

(4.3)

In both cases, we require an 'entropy-satisfying' approximation to the gradient term. In the next section, we discuss appropriate approximations for this term, leading to schemes for both the time-dependent and stationary level set formulations. Our goal now is to turn to the issue of numerical approximations, and to develop the necessary theory and numerics to approximate accurately the level set initial value partial differential equation.

PART II: NUMERICAL APPROXIMATION

5. Traditional techniques for tracking interfaces

Before discussing the numerical approximation of these level set equations, it is instructive to review briefly more traditional techniques for computing the motion of interfaces.

Marker/string methods In these methods, a discrete parametrized version of the interface boundary is used. In two dimensions, marker particles are used; in three dimensions, a nodal triangularization of the interface is often developed. The positions of the nodes are then updated by determining front information about the normals and curvature from the marker representation. Such representations can be quite accurate; however, limitations exist for complex motions. Firstly, if corners and cusps develop in the evolving front, markers usually form 'swallowtail' solutions, which must be removed through delooping techniques which attempt to enforce an entropy condition inherent in such motion; see Sethian (1985). Second, topological changes are difficult to handle: when regions merge, some markers must be removed. Third, significant instabilities in the front can result, since the underlying marker particle motions represent a weakly ill-posed initial value problem; see Osher and Sethian (1988). Finally, extensions of such methods to three dimensions require additional work.

Cell-based methods In these methods, introduced by Noh and Woodward (1976), the computational domain is divided into a set of cells containing 'volume fractions'. These volume fractions are numbers between 0 and 1, and represent the fraction of each cell containing the physical material. At any time, the front can be reconstructed from these volume fractions. Since their introduction, many elaborate reconstruction techniques have been developed over the years to include pitched slopes and curved surfaces; see Chorin (1980), Hirt and Nicholls (1981) and Puckett (1991). The accompanying accuracy depends on the sophistication of the reconstruction and the so-called 'advection sweeps'. Some of the most elaborate and accurate versions of these schemes to date are due to Puckett; see Puckett (1991). Advantages of such techniques include the ability to easily handle topological changes, design adaptive mesh methods, and extend the results to three dimensions. However, determination of geometric quantities such as normals and curvature can be inaccurate.

Characteristic methods In these methods, 'ray-trace'-like techniques are used. The characteristic equations for the propagating interface are used, and the entropy condition at forming corners (see Sethian (1985)) is formally enforced by constructing the envelope of the evolving characteristics. Such methods handle the looping problems more naturally, but may be complex in three dimensions and require adaptive adding and removing rays, which can cause instabilities and/or oversmoothing.

6. A first attempt at constructing an approximation to the gradient

We now turn to our time-dependent level set equation itself, and attempt to construct a numerical method.

Recall that our goal is to solve the equation given in (3.5) by

$$\phi_t + F|\nabla\phi| = 0, \tag{6.1}$$

$$\phi(x, t = 0) \quad \text{given.} \tag{6.2}$$

The marker particle method discretizes the front. The volume-of-fluid (VOF) method divides the domain space into cells containing fractions of material. The level set method divides the domain up into grid points that discretize the values of the level set function ϕ. Thus, the grid values give the height of a surface above the domain, and if we slice this surface by the xy plane, we extract the zero level set corresponding to the front.

Another way to look at this is to say that each grid point contains the value of the level set function at that point. Thus, there is an entire family of contours, only one of which is the zero level set (see Fig. 10). Rather than move each of the contours in a Lagrangian fashion, we stand at each grid point and update its value to correspond to the motion of the surface, thus producing a new contour value at that grid point.

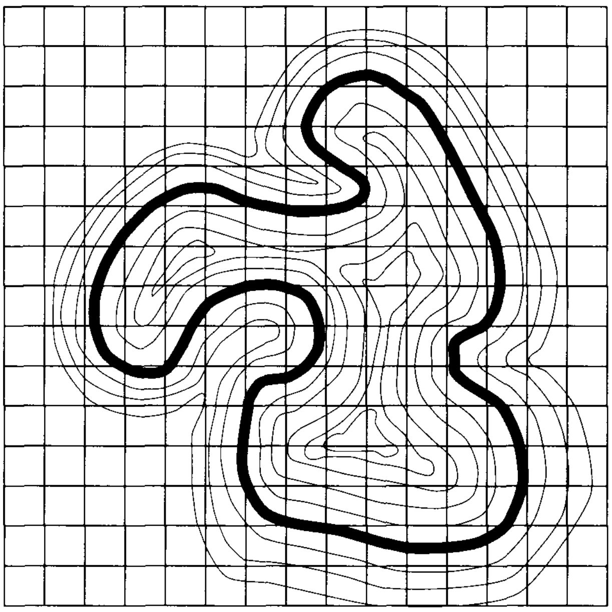


Fig. 10. Dark line is zero level set corresponding to front.

What is the right way to approximate this equation? We shall investigate the most straightforward numerical approach we can think of by studying the simpler case of an evolving curve whose position can always be described as the graph of a function. The equation for this case was given in (2.17) as shown in Fig. 7, namely

$$\psi_t = F(1 + \psi_x^2)^{1/2}. \tag{6.3}$$

Perhaps the most straightforward way of creating an algorithm to approximate the solution to this equation is to replace all spatial derivatives with central differences and the time derivative with a forward difference, just as we did in the Lagrangian case. However, it is easy to see that such an algorithm may not work. Let $F(\kappa) = 1$ and consider the initial value problem

$$\psi_t = (1 + \psi_x^2)^{1/2}, \tag{6.4}$$

$$\psi(x,0) = f(x) = \left\{ \begin{array}{ll} 1/2 - x & x \le 1/2 \\ x - 1/2 & x > 1/2 \end{array} \right\}. \tag{6.5}$$

The initial front is a 'V' formed by rays meeting at $(1/2, 0)$. By our entropy condition, the solution at any time t is the set of all points located a distance t from the initial 'V'. To construct a numerical scheme, divide the interval $[0,1]$ into $2M - 1$ points, and form the central difference approximation to the spatial derivative ψ_x in equation (6.4), namely

$$\frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} = [1 + [\frac{\psi_{i+1}^n - \psi_{i-1}^n}{2\Delta x}]^2]^{1/2} \tag{6.6}$$

Since $x_M = 1/2$, by symmetry, $\psi_{M+1} = \psi_{M-1}$, thus $\psi_t(1/2,0) = 1$. However, for all $x \neq 1/2$, ψ_t is correctly calculated to be $\sqrt{2}$, since the graph is linear on either side of the corner and thus the central difference approximation is exact. Note that this has nothing to do with the size of the space step Δx or the time step Δt. *No matter how small we take the numerical parameters, as long as we use an odd number of points, the approximation to ψ_t at $x = 1/2$ gets no better.* It is simply due to the way in which the derivative ψ_x is approximated. In Fig. 11, we show results using this scheme, with the time derivative ψ_t replaced by a forward difference scheme.

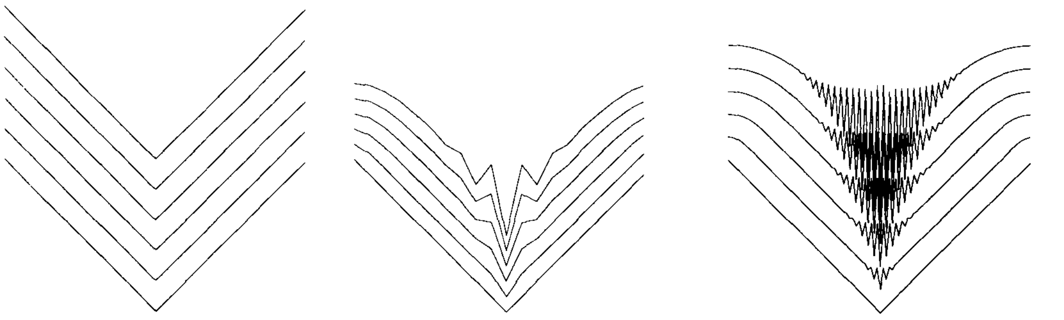

Exact solution
Central differences $\Delta t = .005$
Central differences $\Delta t = .0005$

Fig. 11. Central difference approximation to level set equation.

It is easy to see what has gone wrong. In the exact solution, $\psi_t = \sqrt{2}$ for all $x \neq 1/2$. This should also hold at $x = 1/2$ where the slope is not defined; the Huygens construction sets $\psi_t(x = 1/2, t)$ equal to $\lim_{x \to 1/2} \psi_t$. Unfortunately, the central difference approximation chooses a different (and, for our purpose, wrong) limiting solution. It sets the undefined slope ψ_x equal to the average of the left and right slopes. As the calculation progresses, this miscalculation of the slope propagates outwards from the spike as wild oscillations. Eventually, these oscillations cause blowup in the solution.

It is clear that some more care must be taken in formulating an algorithm. What we require are schemes that approximate the gradient term $|\nabla\phi|$ in a way that correctly accounts for the entropy condition. This is the topic of the next section.

7. Schemes from hyperbolic conservation laws

Our schemes are linked to those from hyperbolic conservation laws. As motivation, consider the single scalar hyperbolic conservation law

$$u_t + [G(u)]_x = 0. \tag{7.1}$$

It is well known that discontinuities known as shocks can develop in the solution, even with smooth initial data; see Lax (1970) and LeVeque (1992). These discontinuities occur because of the collision of characteristics, and an appropriate weak solution must be constructed to carry the solution beyond the collision time. The correct 'entropy solution' is obtained by considering the limit of the associated viscous conservation laws $u_t + [G(u)]_x = \epsilon u_{xx}$ as the viscous coefficient ϵ goes to zero.

From a numerical point of view, the equation can be approximated through the construction of appropriate numerical fluxes g such that

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{g(u_i^n, u_{i+1}^n) - g(u_{i-1}^n, u_i^n)}{\Delta x}, \tag{7.2}$$

where we require that $g(u, u) = G(u)$. A wide collection of numerical flux functions are available, such as the Lax–Friedrichs flux, Godunox flux, and TVD schemes; see Colella and Puckett (1994) and LeVeque (1992). The goal in the construction of such flux functions is to make sure that the conservation form of the equation is preserved, make sure the entropy condition is satisfied, and try to give smooth (highly accurate) solutions away from the discontinuities. One of the most straightforward approximate numerical fluxes is the Engquist–Osher scheme (Engquist and Osher 1980), which is given by

$$g_{EO}(u_1, u_2) = G(u_1) + \int_{u_1}^{u_2} \min(\frac{\mathrm{d}G}{\mathrm{d}u}, 0)\,\mathrm{d}u. \tag{7.3}$$

For the Burger's equation in which $G(u) = u^2$, we have the particularly compact representation of this flux function as

$$g_{EO}(u_1, u_2) = (\max(u_1, 0)^2 + \min(u_2, 0)^2). \tag{7.4}$$

This flux function will serve as our core technique for approximating the level set equation.

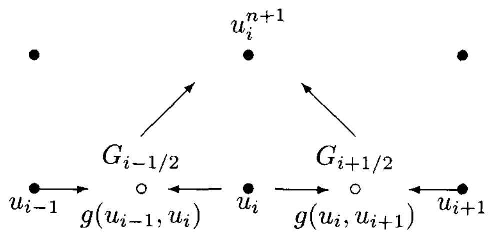


Fig. 12. Update of u through numerical flux function.

8. Approximations to the time-dependent level set equation

In this section, we develop schemes for the level set equation

$$\phi_t + F|\nabla\phi| = 0. \tag{8.1}$$

We begin by writing this equation with a general Hamiltonian H as

$$\phi_t + H(\phi_x, \phi_y, \phi_z) = 0, \tag{8.2}$$

where

$$H(u, v, w) = \sqrt{u^2 + v^2 + w^2}. \tag{8.3}$$

We begin with the one-dimensional version, that is, $\phi_t + H(\phi_x) = 0$, where $H(u) = \sqrt{u^2}$.

From the previous section, we have numerical flux functions for the conservation equation $u_t + [G(u)]_x = 0$, satisfying

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{g(u_i^n, u_{i+1}^n) - g(u_{i-1}^n, u_i^n)}{\Delta x}. \tag{8.4}$$

In terms of our computational grid shown in Fig. 12, the value of G at the point $(i - 1/2)\Delta x$ (called $G_{i-1/2}$) is approximated by the numerical flux function g as

$$G_{i-1/2} = g(u_{i-1}^n, u_i^n). \tag{8.5}$$

Similarly, at the point $i + 1/2$, we have

$$G_{i+1/2} = g(u_i^n, u_{i+1}^n). \tag{8.6}$$

Then from Fig. 12, we see that the right-hand side of equation (7.2) is just the central difference operator applied to the numerical flux function g. As the grid size goes to zero, consistency requires that $g(u, u) = G(u)$.

We are now all set to build a scheme for our level set equation. Let $u = \phi_x$. Then we can write

$$\phi_t + H(u) = 0. \tag{8.7}$$

In terms of our computational grid in Fig. 13, in order to produce ϕ_i^{n+1} we

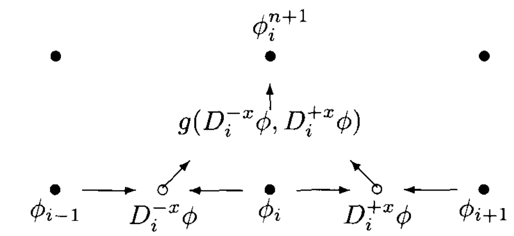


Fig. 13. Update of ϕ through numerical Hamiltonian.

need ϕ_i^n as well as a value for $H(u_i^n)$. Fortunately, an approximate value for $H(u_i^n)$ is *exactly* what is given by our numerical flux function, hence we have

$$H(u) \approx g(u_{i-1/2}, u_{i+1/2}). \tag{8.8}$$

All that remains is to construct values for u in the middle of our computational cells. Since $u = \phi_x$, we can use a central difference approximation in ϕ to construct those values. Thus (see Fig. 13), we have

$$\phi_i^{n+1} = \phi_i^n - \Delta t \, g\left(\frac{\phi_i^n - \phi_{i-1}^n}{\Delta x}, \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}\right), \tag{8.9}$$

where g is one of the numerical flux functions and, again, we have substituted forward and backward difference operators on ϕ for the values of u at the left and right states.

In the specific case of our one-dimensional level set equation with $H(u) = \sqrt{u^2}$, we can use the EO scheme given in the previous section and, for speed $F = 1$, write

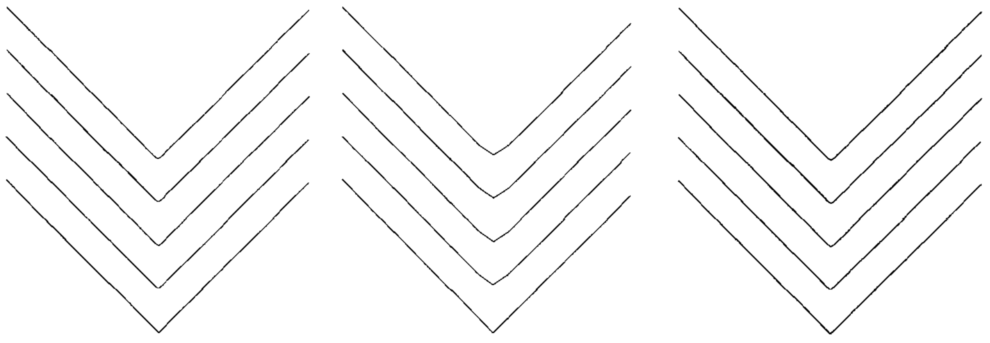
$$\phi_i^{n+1} = \phi_i^n - \Delta t \, (\max(D_i^{-x}, 0)^2 + \min(D_i^{+x}, 0)^2)^{1/2}. \tag{8.10}$$

This is the level set scheme given in Osher and Sethian (1988). As long as the Hamiltonian is symmetric in each of the space dimensions, the above can be replicated in each space variable to construct schemes for two- and three-dimensional front propagation problems.

In general we apply the following philosophy:

1 if the Hamiltonian H is convex, then we use the above scheme
2 if the Hamiltonian H is non-convex, then we use a variant on the Lax–Friedrichs scheme described below.

It is important to point out that far more sophisticated schemes exist than the ones presented here. In the applications of these schemes to hyperbolic problems and shock dynamics, high-order resolution schemes are often necessary (Colella and Puckett 1994), because of the differentiation of the numerical flux function g. However, in our case, because we are solving



(a) Exact solution (b) Scheme with 20 points (c) Scheme with 100 points

Fig. 14. Upwind, entropy-satisfying approximations to the level set equation.

$\phi_t + H(u) = 0$ rather than $u_t + [H(u)]_x$, the differentiation is not required. Thus, we have found that for almost all practical purposes, the first- and second-order schemes presented below are adequate.

Before constructing the general schemes, let's return to the example of the propagating curve. Earlier, we attempted to follow the propagation of a simple corner moving with speed $F = 1$. Our attempts to use a central difference approximation failed. In Fig. 14, we show what happens if we use the scheme given in equation (8.10). The exact answer is shown, together with two simulations. The first uses the entropy-satisfying scheme with only 20 points (Fig. 14b), the second (Fig. 14c) with 100 points. In the first approximation, the entropy condition is satisfied, but the corner is somewhat smoothed due to the small number of points used. In the more refined calculation, the corner remains sharp, and the exact solution is very closely approximated.

9. First- and second-order schemes for convex speed functions

Given a convex speed function F (that is, a speed function F such that the resulting Hamiltonian $H = F|\nabla\phi|$ is convex), we can produce the following schemes, first presented in Osher and Sethian (1988). Start with the equation

$$\phi_t + H(\phi_x, \phi_y, \phi_z) = 0, \tag{9.1}$$

and approximate it by

$$\begin{aligned}\phi_i^{n+1} = \phi_i^n - \Delta t\,\Big(&\frac{\phi_{ijk}^n - \phi_{i-1,j,k}^n}{\Delta x}, \frac{\phi_{i+1,j,k}^n - \phi_{i,j,k}^n}{\Delta x}, \\ &\frac{\phi_{ijk}^n - \phi_{i,j-1,k}^n}{\Delta y}, \frac{\phi_{i,j+1,k}^n - \phi_{i,j,k}^n}{\Delta y}, \\ &\frac{\phi_{ijk}^n - \phi_{i,j,k-1}^n}{\Delta z}, \frac{\phi_{i,j,k+1}^n - \phi_{i,j,k}^n}{\Delta z}\Big).\end{aligned} \tag{9.2}$$

A multi-dimensional version of this scheme (Osher and Sethian 1988) is then given by

$$g_{EO}(u_1,u_2,v_1,v_2,w_1,w_2) = [\max(u_1,0)^2+\min(u_2,0)^2+ \max(v_1,0)^2+\min(v_2,0)^2+ \max(w_1,0)^2+\min(w_2,0)^2]^{1/2}. \tag{9.3}$$

Thus we have

First-order space convex

$$\phi_{ijk}^{n+1} = \phi_{ijk}^n - \Delta t\left(\max(F_{ijk},0)\nabla^+ + \min(F_{ijk},0)\nabla^-\right) \tag{9.4}$$

where

$$\nabla^+ = \Big(\max(D_{ijk}^{-x},0)^2+\min(D_{ijk}^{+x},0)^2+ \max(D_{ijk}^{-y},0)^2+\min(D_{ijk}^{+y},0)^2+ \max(D_{ijk}^{-z},0)^2+\min(D_{ijk}^{+z},0)^2\Big)^{1/2}, \tag{9.5}$$

$$\nabla^- = \Big(\max(D_{ijk}^{+x},0)^2+\min(D_{ijk}^{-x},0)^2+ \max(D_{ijk}^{+y},0)^2+\min(D_{ijk}^{-y},0)^2+ \max(D_{ijk}^{+z},0)^2+\min(D_{ijk}^{-z},0)^2\Big)^{1/2}. \tag{9.6}$$

Here, we have used a short-hand notation in which $D^{+x}\phi_i^n$ is rewritten as D_i^{+x}, *etc.*

Second-order space convex

The above schemes can be extended to higher order, using technology from Harten, Engquist, Osher and Chakravarthy (1987). The basic trick is to build a switch that turns itself off whenever a shock is detected; otherwise, it will use a higher-order polynomial approximation of minimal oscillations. These details will not be presented; see Osher and Sethian (1988) for details. The scheme is the same as the above; however, this time ∇^+ and ∇^- are given by

$$\nabla^+ = \Big(\max(A,0)^2+\min(B,0)^2+ \max(C,0)^2+\min(D,0)^2+ \max(E,0)^2+\min(F,0)^2\Big)^{1/2}, \tag{9.7}$$

$$\nabla^- = \Big(\max(B,0)^2+\min(A,0)^2+ \max(D,0)^2+\min(C,0)^2+ \max(F,0)^2+\min(E,0)^2\Big)^{1/2}, \tag{9.8}$$

where

$$A = D^{-x}_{ijk} + \frac{\Delta x}{2} m(D^{-x-x}_{ijk}, D^{+x-x}_{ijk}), \quad B = D^{+x}_{ijk} - \tfrac{\Delta x}{2} m(D^{+x+x}_{ijk}, D^{+x-x}_{ijk}), \tag{9.9}$$

$$C = D^{-y}_{ijk} + \frac{\Delta y}{2} m(D^{-y-y}_{ijk}, D^{+y-y}_{ijk}), \quad D = D^{+y}_{ijk} - \tfrac{\Delta y}{2} m(D^{+y+y}_{ijk}, D^{+y-y}_{ijk}), \tag{9.10}$$

$$E = D^{-z}_{ijk} + \frac{\Delta z}{2} m(D^{-z-z}_{ijk}, D^{+z-z}_{ijk}), \quad F = D^{+z}_{ijk} - \tfrac{\Delta z}{2} m(D^{+z+z}_{ijk}, D^{+z-z}_{ijk}), \tag{9.11}$$

and the switch function is given by

$$m(x,y) = \left\{ \begin{array}{cc} \left\{ \begin{array}{ll} x & \text{if } |x| \le |y| \\ y & \text{if } |x| > |y| \end{array} \right\} & xy \ge 0 \\ 0 & xy < 0 \end{array} \right\}. \tag{9.12}$$

10. First- and second-order schemes for non-convex speed functions

Given a non-convex speed function F (that is, a speed function F for which the resulting Hamiltonian $H = F|\nabla\phi|$ is non-convex), we can follow the work in Osher and Shu (1991), replacing the Hamiltonian $F|\nabla\phi|$ with the Lax–Friedrichs numerical flux function. This produces the following schemes.

First-order space non-convex

$$\begin{aligned} \phi^{n+1}_{ijk} &= \phi^n_{ijk} - \Delta t[H(\frac{D^{-x}_{ijk} + D^{+x}_{ijk}}{2}, \frac{D^{-y}_{ijk} + D^{+y}_{ijk}}{2}, \frac{D^{-z}_{ijk} + D^{+z}_{ijk}}{2}) \\ &\quad -\frac{1}{2}\alpha_u(D^{+x}_{ijk} - D^{-x}_{ijk}) - \frac{1}{2}\alpha_v(D^{+y}_{ijk} - D^{-y}_{ijk}) - \frac{1}{2}\alpha_w(D^{+z}_{ijk} - D^{-z}_{ijk})] \end{aligned} \tag{10.1}$$

where α_u (α_v, α_w) is a bound on the partial derivative of the Hamiltonian with respect to the first (second, third) argument, and the non-convex Hamiltonian is a user-defined input function.

Second-order space non-convex

$$\begin{aligned} \phi^{n+1}_{ijk} = \quad & \phi^n_{ijk} - \Delta t[H(\frac{A+B}{2}, \frac{C+D}{2}, \frac{E+F}{2}) \\ & -\frac{1}{2}\alpha_u(B-A) - \frac{1}{2}\alpha_v(D-C) - \frac{1}{2}\alpha_w(F-E)], \end{aligned} \tag{10.2}$$

where A, B, C, D, E, and F are defined as above. For details, see Osher and Shu (1991) and Adalsteinsson and Sethian (1995*b*, 1995*c*).

11. Approximations to curvature and normals

As discussed above, one of the main advantages of level set formulations is that geometric qualities of the propagating interface, such as curvature and normal direction, are easily calculated. For example, consider the case of a curve propagating in the plane. The expression for the curvature of the zero level set assigned to the interface itself (as well as all other level sets) is given by

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_y\phi_x\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}. \tag{11.1}$$

In the case of a surface propagating in three space dimensions, one has many choices for the curvature of the front, including the mean curvature κ_M and the Gaussian curvature κ_G. Both may be conveniently expressed in terms of the level set function ϕ as

$$\kappa_M = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} = \frac{\begin{array}{c}(\phi_{yy}+\phi_{zz})\phi_x^2 + (\phi_{xx}+\phi_{zz})\phi_y^2 + (\phi_{xx}+\phi_{yy})\phi_z^2 \\ -2\phi_x\phi_y\phi_{xy} - 2\phi_x\phi_z\phi_{xz} - 2\phi_y\phi_z\phi_{yz}\end{array}}{(\phi_x^2+\phi_y^2+\phi_z^2)^{3/2}} \tag{11.2}$$

$$\kappa_G = \frac{\begin{array}{c}\phi_x^2(\phi_{yy}\phi_{zz}-\phi_{yz}^2) + \phi_y^2(\phi_{xx}\phi_{zz}-\phi_{xz}^2) + \phi_z^2(\phi_{xx}\phi_{yy}-\phi_{xy}^2) \\ + 2[\phi_x\phi_y(\phi_{xz}\phi_{yz}-\phi_{xy}\phi_{zz}) + \phi_y\phi_z(\phi_{xy}\phi_{xz}-\phi_{yz}\phi_{xx}) \\ + \phi_x\phi_z(\phi_{xy}\phi_{yz}-\phi_{xz}\phi_{yy})]\end{array}}{(\phi_z^2+\phi_y^2+\phi_x^2)^2} \tag{11.3}$$

Construction of the normal itself requires a more sophisticated scheme than simply building the difference approximation to $\nabla\phi$. This is because at corners, the direction of the normal can undergo a jump. This suggests the following technique, introduced by Sethian and Strain (1992). First, the one-sided difference approximations to the unit normal in each possible direction are formed. All four limiting normals are then averaged to produce the approximate normal at the corner. Thus, the normal n_{ij} is formed by first letting

$$n_{ij}^* \equiv \frac{\phi_x, \phi_y}{(\phi_x^2+\phi_y^2)^{1/2}} \tag{11.4}$$

$$\begin{aligned} = & \frac{(D_{ij}^{+x}, D_{ij}^{+y})}{[(D_{ij}^{+x})^2 + (D_{ij}^{+y})^2]^{1/2}} + \frac{(D_{ij}^{-x}, D_{ij}^{+y})}{[(D_{ij}^{-x})^2 + (D_{ij}^{+y})^2]^{1/2}} \\ & \frac{(D_{ij}^{+x}, D_{ij}^{-y})}{[(D_{ij}^{+x})^2 + (D_{ij}^{-y})^2]^{1/2}} + \frac{(D_{ij}^{-x}, D_{ij}^{-y})}{[(D_{ij}^{-x})^2 + (D_{ij}^{-y})^2]^{1/2}}, \end{aligned} \tag{11.5}$$

and then normalizing so that $n_{ij} \equiv n_{ij}^*/|n_{ij}^*|$. If any of the one-sided approximations to $|\nabla\phi|$ is zero, that term is not considered and the weights are adjusted accordingly.

12. Initialization and boundary conditions

The time-dependent level set approach requires an initial function $\phi(x, t = 0)$ with the property that the zero level set of that initial function corresponds to the position of the initial front. The original level set algorithm computed the signed distance from each grid point to the initial front, which is matched to the zero level set. This is an expensive technique. Many other initial functions are possible, including those that are essentially constant except in a narrow band around the front itself.

The use of a finite computational grid means that we must develop boundary conditions. If the speed function F causes the front to expand (such as in the case $F = 1$), upwind schemes will naturally default to outward-flowing one-sided differences at the boundary of the domain. However, in cases of more complex speed functions, mirror boundary conditions usually suffice.

PART III: EXTENSIONS

13. A hierarchy of fast level set methods

The above time-dependent level set method is easily programmed. However, it is not particularly fast, nor does it make efficient use of computational resources. In this section, we consider a sequence of more sophisticated versions of the basic scheme.

13.1. Parallel algorithms

The above method updates *all* the level sets, not just the zero level set corresponding to the front itself. The advantage of this approach is that the data structures and operations are extremely clear, and it is a good starting point for building level set codes.

There are a variety of circumstances in which this approach is desirable. If, in fact, all the level sets are themselves important (such as problems encountered in image processing, which are discussed below), then computation over the entire domain is required. A simple approach is to perform a parallel computation. Since each grid point is updated by a nearest neighbour stencil using only grid points on each side, this technique almost falls under the classification of 'embarrassingly parallel'. In Sethian (1989), a parallel version of the level set method was developed for the Connection Machine CM-2 and CM-5. In the CM-2, nodes are arranged in a hypercube fashion; in the CM-5, nodes are arranged in a fat-tree. The code was written in global CMFortran, and at each grid point CSHIFT operators were used to update the level set function. The operation count per time step reduces to $O(1)$, since in most cases the full grid can be placed into physical memory. A time-explicit second-order space method was used to update the level set equation.

13.2. Adaptive mesh refinement

As a first level of creating a more efficient level set method, an adaptive mesh refinement strategy can be pursued. This is the approach taken by Milne and Sethian (1995), motivated by the adaptive mesh refinement work in Berger and Colella (1989). Adaptivity may be desired in regions where level curves develop high curvature or where speed functions change rapidly. In Fig. 15a, we show mesh cells that are hierarchically refined in response to a parent–child relationship around the zero level set of ϕ. Calculations are performed on both the fine and coarse grid. Grid cell boundaries always lie along coordinate lines, and patches do not overlap; in the scheme presented in Milne and Sethian (1995), no attempt is made to align the refined cells with the front.

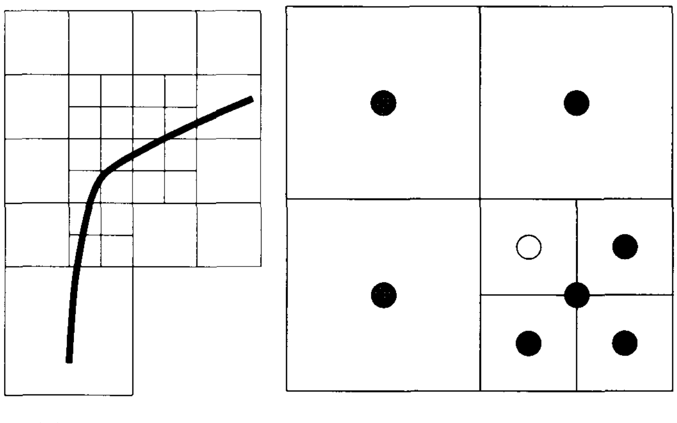

(a) Cell hierarchy (b) Hanging nodes

Fig. 15. Adaptive mesh refinement.

The data structures for the adaptive mesh refinement are fairly straightforward. However, considerable care must be taken at the interfaces between coarse and fine cells; in particular, the update strategy for ϕ at so-called 'hanging nodes' is subtle. These are nodes at the boundary between two levels of refinement which do not have a full set of nearest neighbours required to update ϕ. To illustrate, in Fig. 15b, we show a two-dimensional adaptive mesh; the goal is to determine an accurate update strategy for the hanging node marked $\circ$.

The strategy for updating ϕ at such points is as follows. Consider the archetypical speed function $F(\kappa) = 1 - \epsilon\kappa$.

- The advection term 1 leads to a hyperbolic equation; here, straightforward interpolation of the updated values of ϕ from the coarse cell grid is used to produce the new value of ϕ at $\circ$. The sophisticated technology in Berger and Colella (1989) is not required, since we are modelling the update according to the numerical flux function g, not the derivative of the numerical flux function as required for hyperbolic conservation laws.
- In the case of the curvature term $-\epsilon\kappa$, the situation is not as straightforward. The parabolic term cannot be approximated through simple interpolation. Interpolation from updated values on the coarse grid to the fine grid was shown to provide poor answers; if this procedure is employed, the boundary between the two levels of refinement acts as a source of noise, and significant error is generated at the boundary. Instead, values from both the coarse and refined grid next to the hanging node are used to construct a least squares solution for ϕ before the update. This solution surface is then formally differentiated to produce the various first and second derivatives in each component direction. These values are then used to produce the update value for ϕ similar to all other nodes. For details, see Milne and Sethian (1995).

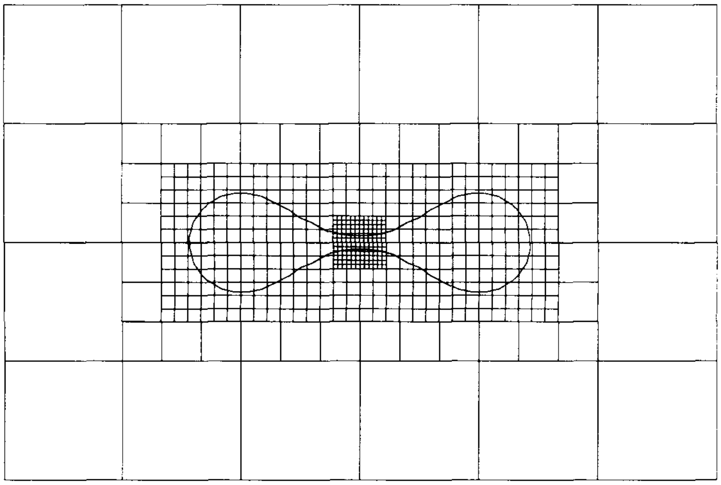


Fig. 16. Two-dimensional slice of adaptive mesh for propagating surface.

As illustration, in Fig. 16, we show a two-dimensional slice of a fully three-dimensional adaptive mesh calculation of a surface collapsing under its mean curvature. As discussed in a later section, the dumbbell neck pinches off under such a configuration, due to the high positive value of the principal axis of curvature.

13.3. Narrow banding and fast methods

An alternative to the above techniques is particularly valuable when one wants to track a specific front, namely the one associated with the zero level set. There are several disadvantages with the 'full-matrix' approach given above.

Speed Performing calculations over the entire computational domain requires $O(N^2)$ operations in two dimensions, and $O(N^3)$ operations in three dimensions, where N is the number of grid points along a side. As an alternative, an efficient modification is to perform work only in a neighbourhood of the zero level set; this is known as the *narrow band approach.* In this case, the operation count in three dimensions drops to $O(kN^2)$, where k is the number of cells in the width of the narrow band. This is a significant cost reduction. Typically, good results can be obtained with about six cells on either side of the zero level.

Calculating extension variables The time-dependent level set approach requires the extension of the speed function F in equation (3.5) to *all* of space; this then updates all of the level sets, not simply the zero level set on which the speed function is naturally defined. Recall that three types of arguments may influence the front speed F; local, global, and independent. Some of these variables may have meaning only on the front itself, and it may be both difficult and awkward to design a speed function that extrapolates the velocity away from the zero level set in a smooth fashion. Thus, another advantage of a narrow band approach is that this extension need only be done to points lying in the narrow band.

Choosing time step The full-matrix approach requires the choice of a time step that applies in response to the maximum velocity over the entire domain, not simply in response to the speed of the front itself. In a narrow band implementation, the time step can be adaptively chosen in response to the maximum velocity field only within the narrow band. This is of significance in problems in which the front speed changes substantially as it moves (for example, due to the local curvature or as determined by the underlying domain). In such problems, the CFL restriction for the velocity field for *all* the level sets may be much more stringent than the one for those sets within the narrow band.

The above 'narrow band' method was introduced by Chopp (1993), used in recovering shapes from images by Malladi, Sethian and Vemuri (1994), and analysed extensively by Adalsteinsson and Sethian (1995*a*).

In Fig. 17 we show the placement of a narrow band around an initial front. The entire two-dimensional grid of data is stored in a square array. A one-dimensional object is then used to keep track of the points in this array (dark grid points in Fig. 17 are located in a narrow band around the front of a user-

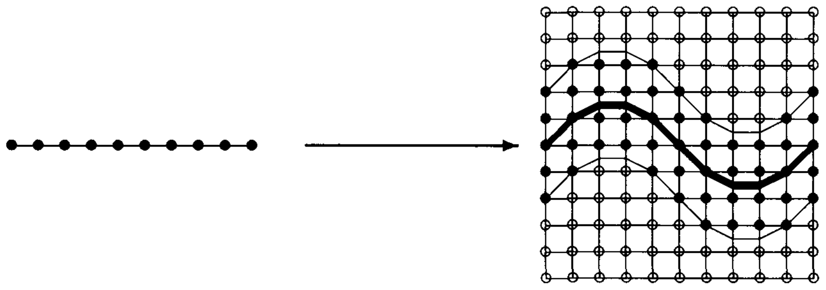


Fig. 17. Pointer array tags interior and boundary band points.

defined width). Only the values of ϕ at points within the tube are updated; values of ϕ at grid points on the boundary of the narrow band are frozen. When the front moves near the edge of the tube boundary, the calculation is stopped, and a new tube is built with the zero level set interface boundary at the centre. This rebuilding process is known as 're-initialization'.

Thus, the narrow band method consists of the following loop:

- tag 'alive' points in narrow band
- build 'land mines' to indicate nearby edge
- initialize 'far away' points outside(inside) narrow band with large positive(negative) values
- solve level set equation until a land mine is hit
- rebuild, loop.

Use of narrow bands leads to level set front advancement algorithms which are equivalent in complexity to traditional marker methods and cell techniques, while maintaining the features of topological merger, accuracy, and easy extension to multi-dimensions. Typically, the speed-up associated with the narrow band method is about ten times faster on a 160×160 grid than the full matrix method. Such a speed-up is substantial: in three-dimensional simulations, it can make the difference between computationally intensive problems and those that can be done with relative ease. Details on the accuracy, typical tube sizes, and number of times a tube must be rebuilt may be found in Adalsteinsson and Sethian (1995*a*).

This narrow banding technique requires a rebuilding and re-initialization of a new narrow band around the location of the front. There are several ways to perform this re-initialization, one of which leads to an efficient level set scheme for the particular case of a front propagating with a speed $F = F(x, y)$ where F is of one sign.

13.4. Re-initialization techniques: direct evaluation, iteration, Huygens' flowing

Direct evaluation
A straightforward approach is to find the zero level set by using a contour plotter and then recalculate the signed distance from each grid point to this zero level set to rebuild the band. This technique, first used in Chopp (1993), can be used to ensure that the level set function stays well behaved. However, this approach can be expensive, since the front must be explicitly constructed and distances must be calculated to neighbouring grid points.

Iteration
An alternative to this was given by Sussman, Smereka and Osher (1994), based on an observation of Morel. Its virtue is that one need not find the zero level set to re-initialize the level set function. Consider the partial differential equation

$$\phi_t = \text{sign}(\phi)(1 - |\nabla\phi|). \tag{13.1}$$

Given initial data for ϕ, solving the above equation to steady state provides a new value for ϕ with the property that $|\nabla\phi| = 1$, since convergence occurs when the right-hand side is zero. The sign function controls the flow of information in (13.1); if ϕ is negative, information flows one way and if ϕ is positive, then information flows the other way. The next effect is to 'straighten out' the level sets on either side of the zero level set and produce a ϕ function with $|\nabla\phi| = 1$, which in fact corresponds to the signed distance function. Thus, their approach is to periodically stop the level set calculation and solve (13.1) until convergence: if repeated sufficiently often, the initial data are often close to the signed distance function, and few iterations are required. One disadvantage of this technique is the relative crudeness of the switch function based on the sign of the level set equation; considerable motion of the zero level set can occur during the re-initialization, since the sign function does not accurately model the exact location of the front.

Huygens' principle flowing
An alternative technique is based on the idea of computing crossing times as discussed in Sethian (1994), and is related to the ideas given in Kimmel and Bruckstein (1993). Consider a particular value for the level set function $\phi_{\text{initial}}(x,t)$. With speed function $F = 1$, flow the level set function both forward and backwards in time and calculate crossing times (that is when ϕ changes sign) at each grid point. These crossing times (both positive and negative) are equal to the signed distance function by Huygens' principle. This approach has the advantage that one knows *a priori* how long to run the problem forward and backward to re-initialize grid points a given distance from the front. This calculation can be performed using a high-order scheme to produce accurate values for the crossing times.

This idea of computing crossing times is equivalent to converting the level set evolution problem into a stationary problem. This conversion can be used to develop an extremely fast marching level set scheme for the particular case of solving the level set equation for speed function $F = F(x, y)$, where F is always either positive or negative. This is discussed in the section on fast marching methods for the stationary formulation.

14. Additional complexities

Since their introduction, the capabilities and applicability of time-dependent level set methods have been considerably refined and extended. In this section, we discuss a few extensions that have proved to be useful in a variety of applications.

14.1. Masking and sources

Consider the problem of a front propagating with a speed F and subject to the constraint that the evolving interface cannot enter into a region Ω in the domain. This region Ω is referred to as a 'mask', since it inhibits all motion. There are several solutions to this problem, depending on the degree of accuracy required.

The simplest solution is to set the speed function F equal to zero for all grid points inside Ω. The location of all points inside Ω can be determined before any calculation is carried out. This technique assures that the front stops within one grid cell of the mask. In Fig. 18, we show a plane front propagating upwards with speed $F = 1$ in the upwards direction, with a rectangular block in the centre of the domain serving as a mask. In Fig. 18a, the speed function is reset to zero inside the mask region, and as the front propagates upwards it is stopped in the vicinity of the mask and is forced to bend around it.

The calculations in Fig. 18a are performed on a very crude 13×13 mesh in order to accentuate a problem with this approach, namely that the front can only be guaranteed to stop within one grid cell of the obstacle itself. This is because the level set method constructs an interpolated speed between grid points, and hence by setting the speed function to zero on and in the mask, the front slows down before it actually reaches the mask. Note that since this means one grid cell *normal* to the mask's boundary, a considerable amount of error can result.

A different fix, which eliminates much of this problem, comes from an alternate view. Given a mask area Ω, construct the signed distance function ϕ^{Ω} by taking the positive distance if inside Ω and the negative distance if outside (note that this is opposite sign choice from the one typically used). Then we limit motion into the masked region not by modifying the speed function, but instead by resetting the evolving level set function. Let $\phi^{(*)}$ be

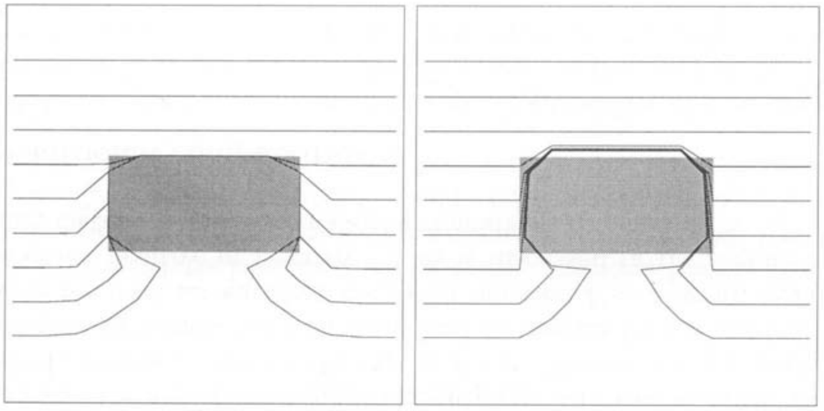

(a) Velocity = 0 inside rectangle (b) ϕ reset by mask

Fig. 18. Front propagating upwards around masking block: 13×13 grid.

the value produced by advancing the level set ϕ^n one time step. Then let

$$\phi^{n+1} = \max(\phi^*, \phi^\Omega). \tag{14.1}$$

This resets the level set function so that penetration is not possible; of course, this is only accurate to the order of the grid. Results using this scheme are shown in Fig. 18b. Again, we have used a very coarse grid to accentuate the differences.

If we consider now the opposite problem, in which a region Ω acts as a source, then the solution is equally straightforward, and we obtain $\phi^{n+1} = \min(\phi^*, -\phi^\Omega)$; this is the technique used in Rhee, Talbot and Sethian (1995).

14.2. Discontinuous speed functions and sub-grid resolution

Let us generalize the above problem somewhat, and imagine that we want to solve an interface propagation problem in which there is a clear discontinuous speed function. For example, one may want to track the propagation of an interface through materials that correspond to differing media, across which propagation rates change quite sharply. As an example, consider again the evolution of the upwards propagating front, but this time the rectangular block halves the speed (that is, $F = 1$ outside Ω and $F = 0.5$ inside and on Ω). The standard level set will interpolate between these two speeds, and the results obtained will depend on the placement of the underlying grid; substantial variation in result will occur depending on whether a grid line lies directly on, below, or above the bottom edge of the rectangular block.

In order to solve this problem accurately, we need some sub-grid information about the speed function to construct correctly the speed function for those cells that lie only partially within Ω. Such a technique can be devised, motivated by the idea of the volume-of-fluid methods discussed earlier.

Given a region Ω, before any calculation proceeds we construct the cell fraction Vol_{ij}^{Ω}, which is a number between 0 and 1 for those cells that have at least one grid point in Ω and one outside Ω. This cell fraction corresponds to the amount of Ω material in the cell. These values are stored, and a list is kept of such boundary cells. We then proceed with the level set calculation, letting F be given by its value in the corresponding region. However, we modify the speed function for those cells that are marked as boundary cells. At the beginning of the time step, compute the volume fraction Vol_{ij}^{ϕ} for the zero level set in each cell; this may be approximately done without explicitly finding the zero level set through a least squares fit. This value is then compared with the stored value $\mathrm{Vol}_{ij}^{\Omega}$ to provide an appropriate speed.

14.3. Multiple interfaces

As initially designed in Osher and Sethian (1988), the level set technique applies to problems in which there is a clear distinction between an 'inside' and 'outside'. This is because the interface is assigned the zero level value between the two regions. Extensions to multiple (more than two) interfaces have been made in some specific cases. In the case in which interfaces are passively transported and behave nicely, one may be able to use only one level set function and judiciously assign different values at the interfaces. For example, the zero level set may correspond to the boundary between two regions A and B, with the level set value 10 corresponding to the interface between two regions B and C. If A and C never touch, then this technique may be used to follow the interfaces in some cases.

However, in the more general case involving the emergence and motion of triple points, a different approach is required, because many different situations can occur; see, for example, Bronsard and Wetton (1995) and Taylor, Cahn and Handwerker (1992). Consider the following canonical example, as illustrated in Fig. 19. Regions A and B are both circular disks growing into region C with speed unity in the direction normal to the interface. At some point, the interfaces will touch and meet at a triple point, where a clear notion of 'inside' and 'outside' cannot be assigned in a consistent manner.

A level set approach to this problem has been proposed by Merriman, Bence and Osher (1994); they move each interface separately for one time step, find the interfaces of the various fronts, and then rebuild level set functions. This technique requires re-initializing the pairwise level set functions; any of the techniques described earlier can be used. Before describing this technique, we discuss a later set of techniques presented in Sethian (1995*a*, 1995*b*), applicable to many cases, and which do not require any such re-initialization. We will then return to the first algorithm in the case of motion driven by surface tension where some sort of re-initialization is required.

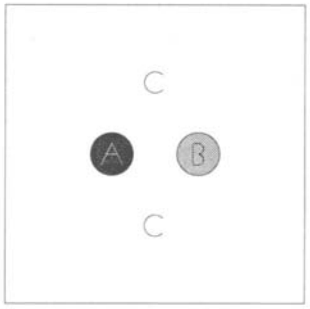


Fig. 19. Regions A and B expand into region C.

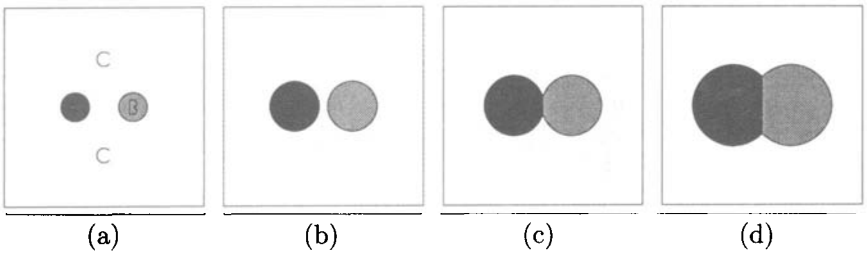
The key idea in each method lies in recasting the interface motion as the motion of one level set function for each material. In some sense, this is what was done in the re-ignition idea given in Rhee et al. (1995), where the front was a flame, which propagated downstream under a fluid flow. This front was re-ignited at each time step at a flame holder point by taking the minimum of the advancing flame and its original configuration around the flame holder, thus ensuring that the maximum burned fluid is achieved.

In general, imagine N separate regions and a full set of all possible pairwise speed functions F_{IJ} which describe the propagation speed of region I into region J: F is taken as zero if region I cannot penetrate J. The idea is to advance each interface to obtain a trial value for each interface with respect to motion into every other region, and then combine the trial values in such a way as to obtain the maximum possible motion of the interface.

In general then, proceed as follows. Given a region I, obtain $N-1$ trial level set functions ϕ^*_{IJ} by moving region I into each possible region J, J=1,N ($J \neq I$) with speed F_{IJ}. During the motion of region I into region J, assume that all other regions are impenetrable, that is, use the masking rule given by equation (14.1). We then test the penetrability of region J itself, leaving the value of ϕ^*_{IJ} unchanged if $F_{IJ} \neq 0$, otherwise modifying it with the maximum of itself and $-\phi^*_{JI}$. Finally, to allow region I to evolve as much as possible, we take the minimum over all possible motions to obtain the new position; this is the re-ignition idea described earlier. Complete details of the approach may be found in Sethian (1995*a*).

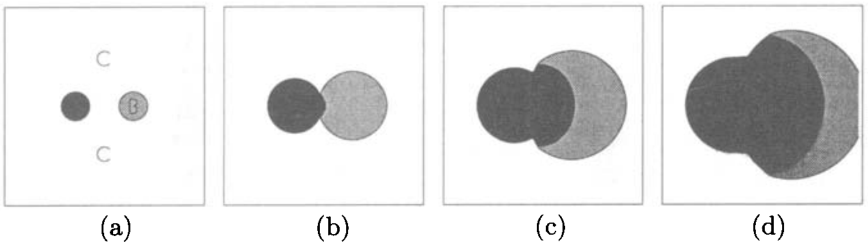
Three examples are shown to illustrate this approach. Given regions A, B, and C, the *influence matrix* describes the interaction of the various regions with each other. The interaction of each region with itself is null, hence the matrix has a dash on the diagonal. The interaction of any pair of regions is required to be zero in one of the two interactions.

In Fig. 20, regions A and B expand with unit speed into region C, but cannot penetrate each other. They advance and meet; the boundary between the two becomes a vertical straight line.



Influence matrix	on A	on B	on C
Effect of A	–	0.0	1.0
Effect of B	0.0	–	1.0
Effect of C	0.0	0.0	–

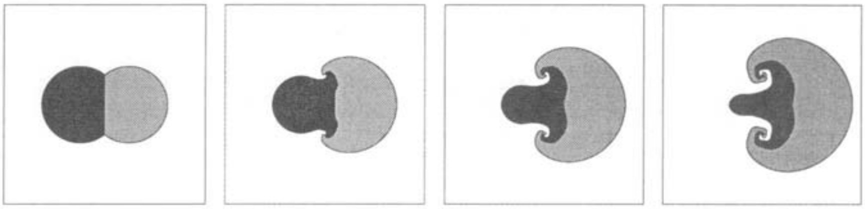
Fig. 20. A and B move into C with speed 1, stop at each other.



Influence matrix	on A	on B	on C
Effect of A	–	3.0	1.0
Effect of B	0.0	–	2.0
Effect of C	0.0	0.0	–

Fig. 21. A into C with speed 1, A into B with speed 3, B into C with speed 2.

Next, we consider a problem with different evolution rates. In Fig. 21, region A grows with speed 1 into region C (and region C grows with speed 0 into region A), and region B grows with speed 2 into region C. Once they come into contact, region A dominates region B with speed 3, thus region B grows through Fig. 21c, and then is 'eaten up' by the advancing region A. Note what happens: region A advances with speed 3 to the edge of region B, which is only advancing with speed 1 into region C. However, region A cannot pass region B, because *its* speed into region C is slower than that of region B.


(a) $T = 0.0$ (b) $T = 2.0$ (c) $T = 3.0$ (d) $T = 5.0$

Influence matrix	on A	on B	on C
Effect of A	–	1.0	0.0
Effect of B	0.0	–	1.0
Effect of C	1.0	0.0	–

Fig. 22. Spiralling triple point: 98 $\times$ 98 grid.

Finally, in Fig. 22, the motion of a triple point between regions A, B, and C is shown. Assume that region A penetrates B with speed 1, B penetrates C with speed 1, and C penetrates A with speed 1. The exact solution is given by a spiral with no limiting tangent angle as the triple point is approached. The triple point does not move; instead, the regions spiral around it. In Fig. 22, results are shown from a calculation on a 98 $\times$ 98 grid. Starting from the initial configuration, the regions spiral around each other, with the leading tip of each spiral controlled by the grid size. In other words, we are unable to resolve spirals tighter than the grid size, and hence that controls the fine-scale description of the motion. However, we note that the triple point remains fixed. A series of additional calculations using this approach may be found in Sethian (1995*b*).

14.4. Triple points

Consider now the case of a triple point motion in which the speed of each interface is driven by curvature, which may correspond to surface tension. Imagine a triple point in which each of the three regions is attempting to move according to their own curvature. In Fig. 23a we show an initial configuration, and in Fig. 23b a final state, which consists of the three lines meeting in equal angles of 120 degrees.

If one attempts to apply the level set method for multiple interfaces described in the previous sections, a difficulty occurs, because each level function attempts to move away from the others, creating a gap. In Fig. 23c, we show this gap developing when a level set technique is applied to the final state.

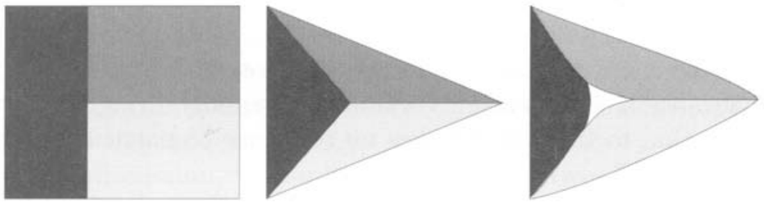

(a) Initial state (b) Final state (c) Level set

Fig. 23. Evolution of triple point under curvature.

Two different level set type algorithms were introduced in Merriman et al. (1994) for tackling this problem. The first can be viewed a 'fix' to the above problem; the second is a wholly different level set approach.

First, the problem with the above calculation is that the various level sets pull apart. A remedy is to reset the level set functions every time step to hold the triple point in place, that is,

$$\phi_i = \phi_i - \max \phi_j, \tag{14.2}$$

where the maximum is taken over all $j \neq i$. This keeps the triple point in place; however, the cost is that the level set functions can develop spontaneous zero crossings later in time. A remedy is to re-initialize all the level sets using any of the re-initialization techniques described in the previous section. With those two added steps in the algorithm, level set methods can easily handle some interesting problems concerning triple points.

A considerably different approach works by applying applying a reaction-diffusion type equation to a characteristic function assigned to each region, which is one inside the region and zero outside. This algorithm works by exploiting the link between curvature flow and a diffusion equation, along the lines of the material discussed earlier. The basic idea is that a diffusion term is applied, and then a sharpening term is executed, which sharpens up the solution. The net effect is to evolve the boundary line under curvature. This is a very clever algorithm, and can be applied to multiple interfaces. A series of fascinating calculations are presented in Merriman et al. (1994); for additional work on this topic, see Bronsard and Wetton (1995).

14.5. Building extension velocity fields

What happens when the speed of the moving front has meaning only on the front itself? This is a common occurrence in areas such as combustion, material science, and fluid mechanics, where the philosophy of embedding the front as the zero level set of a family of contours can be problematic. In fact, the most difficult part of level set methods is this 'extension' problem, and it will be a central focus of Part IV.

Recall the division of arguments in the speed function F given in equation (2.1). Front-based arguments are those that depend on geometric quantities of the front, such as curvature and normal vector, and have a clear meaning for all the level sets. Independent variables are equally straightforward, since their contribution to the speed makes no reference to particular information from the front itself.

The troublesome variables are the so-called 'global' variables, which can arise from solving differential equations on either side of the interface. Briefly, there are at least four ways to extend a velocity from the front to the grid points.

1 At each grid point, find the closest point on the front. This was the technique used in Malladi et al. (1994), and may be done efficiently in many cases by tracing backwards along the gradient given by $\nabla\phi$.
2 Evaluate the speed function off the front using an equation that only has meaning on the front itself. This is the technique used in the crystal growth/dendritic solidification calculations employed in Sethian and Strain (1992), where a boundary integral is evaluated both on and off the front.
3 Develop an evaluation technique that assigns artificial speeds to the level set going through any particular grid point. For example, in the etching/deposition simulations of Adalsteinsson and Sethian (1995*b*, 1995*c*, 1996), visibility of the zero level set must be evaluated away from the front itself.
4 Smearing the influence of the front. In the combustion calculations of Rhee et al. (1995), the influence of the front is mollified to neighbouring grid points on which an appropriate equation is solved.

PART IV: A NEW FAST METHOD

Consider the special case of a monotonically advancing front, that is, a front moving with speed F where F is always postive (or negative). Previously, we have produced a stationary level set equation, namely

$$|\nabla T|F = 1. \tag{14.3}$$

which is simply a static Hamilton–Jacobi equation; if F is only a function of position, this becomes the well-known Eikonal equation. A large body of research has been devoted to studying these types of equations; we refer the interested reader to Barles and Souganidis (1991), Lions (1982) and Souganidis (1985), to name just a few. At the same time, there are a wide collection of schemes to solve this problem; see, for example, Bardi and Falcone (1990), Falcone et al. (1994), Rouy and Tourin (1992) and Osher and Rudin (1992). Here we introduce an entirely new and extremely fast method for solving this equation. It relies on a marriage between our narrow band technique and a fast heapsort algorithm, and can be viewed as an extreme one-cell version of our narrow band technique.

For ease of discussion, we limit ourselves to a two-dimensional problem inside a square from $[0,1] \times [0,1]$ and imagine that the initial front is along the line $y = 0$; furthermore, we assume that we are given a positive speed function $F(x,y)$ that is periodic in x. Thus, the front propagates upwards off the initial line, and the speed does not depend on the orientation of the front (it depends only on *independent variables*, using our earlier terminology). Using our approximation to the gradient, we are then looking for a solution in the unit box to the equation

$$\begin{aligned} F_{ij}^{-1} = \max(D_{ij}^{-x}T, 0)^2 + \min(D_{ij}^{+x}T, 0)^2 + \\ \max(D_{ij}^{-y}T, 0)^2 + \min(D_{ij}^{+y}T, 0)^2), \end{aligned} \tag{14.4}$$

where $T(x, 0) = 0$.

Since equation (14.4) is in essence a quadratic equation for the value at each grid point (assuming the others are held fixed), we can iterate until convergence by solving the equation at each grid point, selecting the largest possible value as the solution in accordance with obtaining the correct viscosity solution. An iterative algorithm for computing the solution to this problem was introduced by Rouy and Tourin (1992). Typically, one iterates several times through the entire set of grid points until a converged solution is reached.

15. A fast marching level set method

The key to constructing a fast marching algorithm is the observation that the upwind difference structure of equation (14.4) means that information propagates 'one way', that is, from smaller values of T to larger values. Hence, our algorithm rests on 'solving' equation (14.4) by building the solution outwards from the smallest time value T. Our idea is to sweep the front ahead in an upwind fashion by considering a set of points in a narrow band around the existing front, and to march this narrow band forward, freezing the values of existing points and bringing new ones into the narrow band structure. The key is in the selection of *which* grid point in the narrow band to update. The technique is easiest to explain algorithmically; see Fig. 24. We imagine that we want to propagate a front through an N by N grid with speed F_{ij} giving the speed in the normal direction at each grid point. Here the set of grid points $j = 1$ corresponds to the y axis, and we assume that $F_{ij} > 0$.

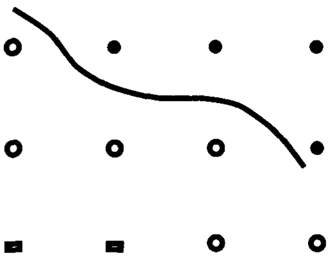


Fig. 24. Narrow band approach to marching level set method.

Algorithm

1 Initialize

(a) (Alive points: grey disks): Let A be the set of all grid points $\{i, j = 1\}$; set $T_{i,1} = 0.0$ for all points in A.
(b) (Narrow band points: black circles): Let *NarrowBand* be the set of all grid points $\{i, j = 2\}$, set $T_{i,1} = dy/F_{ij}$ for all points in *NarrowBand*.
(c) (Far away points: black rectangles): Let *FarAway* be the set of all grid points $\{i, j > 2\}$, set $T_{i,j} = \infty$ for all points in *FarAway*.

2 Marching forwards

(a) Begin loop: Let $(i_{\min}, j_{\min})$ be the point in *NarrowBand* with the smallest value for T.
(b) Add the point $(i_{\min}, j_{\min})$ to A; remove it from *NarrowBand*.
(c) Add to the narrow band list any neighbouring points $(i_{\min} - 1, j_{\min})$, $(i_{\min} + 1, j_{\min})$, $(i_{\min}, j_{\min} - 1)$, $(i_{\min}, j_{\min} + 1)$ that are not *Alive*. If the neighbour is in *FarAway*, remove it from that list.
(d) Recompute the values of T at all neighbours according to equation (14.4), selecting the largest possible solution to the quadratic equation.
(e) Return to top of loop.

We take periodic boundary conditions where required. Assuming for the moment that it takes no work to determine the member of the narrow band with the smallest value of T, the total work required to compute the solution at all grid points is $O(N^2)$, where calculation is performed on an N by N grid.

Why does the above algorithm work? Since we are always locating the smallest value in the narrow band, its value for T must be correct; other narrow band points or far away points with larger T values cannot affect it. The process of recomputing the T values at neighbouring points (that have not been previously accepted) cannot yield a value smaller than any of that

$$\begin{array}{ccc} & \mathrm{B} & \\ \mathrm{A} & ? & \mathrm{C} \\ & \mathrm{D} & \end{array}$$

Fig. 25. Matrix of neighbouring values.

at any of the accepted points, since the correct viscosity solution is obtained by selecting the *largest* possible solution to the quadratic equation. Thus, we can march the solution outwards, always selecting the narrow band grid point with minimum trial value for T, and readjusting neighbours. Another way to look at this is that each minimum trial value begins an application of Huygens' principle, and the expanding wave front touches and updates all others.

15.1. Proof that the algorithm constructs a viable solution

Here, we prove that the above algorithm produces a solution that everywhere satisfies the discrete version of our equation, which is given by

$$\begin{aligned} f_{ij}^2 = \max\left(\max(D_{ij}^{-x}T, 0), -\min(D_{ij}^{+x}T, 0)\right)^2 + \\ \max\left(\max(D_{ij}^{-y}T, 0), -\min(D_{ij}^{+y}T, 0)\right)^2, \end{aligned} \tag{15.1}$$

where $f_{ij}^2 = 1/F_{ij}^2$. We shall give a constructive proof. Since the values of $T(x, y, z)$ are built by marching forwards from the smallest value to the largest, we need only show that whenever a 'trial' value is converted into an 'alive' value, none of the recomputed neighbours obtain new values less than the accepted value. If this is true, then we will always be marching ahead in time, and thus the correct 'upwind' nature of the differencing will be respected. We shall prove our result in two dimensions; the three-dimensional proof is the same.

Thus, consider the matrix of grid values given in Fig. 25. Our argument will follow the computation of the new value of T in the centre grid point to replace the value of ?, based on the neighbouring values. We will assume, without loss of generality, that the value A at the left grid point is the smallest of all 'trial' values, and prove that when we recompute the value at the centre grid point (called $T_{\text{recomputed from } A}$), it cannot be less than A. This will prove that the upwinding is respected, and that we need not go back and readjust previously set values. We shall consider the four cases that (1) none of the neighbours B, C, or D, are 'alive', (2) one of these neighbours is 'alive',

(3) two of the neighbours are 'alive', and (4) all three of these neighbours are 'alive'. †

A, B, C and D are 'trial', A is the smallest
In this case, all of the neighbours around the centre grid point are either 'trial' or set to *FarAway.* Since A is the smallest such value, we convert that value to 'alive' and recompute the value at the centre grid point. We now show that the recomputed value $A \leq T_{\text{recomputed from } A} \leq A + f$.

1 Suppose $A + f \leq \min(B, D)$. Then $T_{\text{recomputed from } A} = (A + f)$ is a solution to the problem, since only the difference operator to the left grid point is nonzero. ‡

2 Suppose $A + f \geq \min(B, D)$. Then, without loss of generality, assume that $B \leq D$. We can solve the quadratic equation

$$(T_{\text{recomputed from } A} - A)^2 + (T_{\text{recomputed from } A} - B)^2 = f^2. \qquad (15.2)$$

The discriminant is non-negative when $f \geq \frac{(B-A)}{\sqrt{2}}$, which must be true since we assumed that $A + f \geq B$ and hence $f \geq (B - A)$. Thus, a solution exists, and it is easy to check that this solution must then be greater than or equal to B and thus falls into the required range. Furthermore, we see that $T \leq A + f$, since the second term on the left is non-negative.

Thus, we have shown that $A \leq T_{\text{recomputed from } A} \leq A + f$, and therefore $T_{\text{recomputed from } A}$ cannot be less than the just converted value A.

This case will act as template for the other cases.

B is 'alive', A, C and D are 'trial', A is the smallest of the trial values
In this case, A has just been converted, since it is the smallest of the trial values. We shall prove when we recalculate $T_{\text{recomputed from } A}$, its new value must still be greater than A. At some previous stage, when B was converted from trial to alive, the values of A, C and D were all trial values, and hence must have been larger. Then this means that when B was converted from trial to alive, we had the previous case above, and hence $B \leq T_{\text{recomputed from } B} \leq B+f$; furthermore, since the value at the centre was *not* chosen as the smallest trial value, we must have that $A \leq B + f$. By the above case, we then have that $B \leq A \leq T_{\text{recomputed from } A} \leq B + f$, and hence the recomputed value cannot be less than the just converted value of A.

† Recall that 'alive' means that their T values are less than A. Here, we are using the notation that the symbol A stands for both the grid point and its T value.

‡ We are absorbing the grid size Δx into the inverse speed function f.

C is 'alive', A, B and D are 'trial', A is the smallest of the trial values
In this case, due to the direction of the upwind differencing, the value at C is the contributor in the x direction, the acceptance of A does not affect the recomputation, and the case defaults into the first case above.

The remaining cases are all the same, since the differencing takes the smallest values in each coordinate direction. The proof in three dimensions is identical.

15.2. Finding the smallest value

The key to an efficient version of the above technique lies in a fast way of locating the grid point in the narrow band with the smallest value for T. We use a variation on a heapsort algorithm, see Press, Flannery, Teukolsky and Vetterling (1988) and Sedgewick (1988), with the additional feature of back pointers. In more detail, imagine that the list of narrow band points is initially sorted in a heapsort so that the smallest member can be easily located. We store the values of these points in the heapsort, together with their indices which give their location in the grid structure. We keep a companion array which points from the two-dimensional grid to the location of that grid point in the heapsort array. Finding the smallest value is easy. In order to find the neighbours of that point, we use the pointers from the grid array to the heapsort structure. The values of the neighbours are then recomputed, and then the results are bubbled upwards in the heapsort until they reach their correct locations, at the same time readjusting the pointers in the grid array. This results in an $O(\log N)$ algorithm for the total amount of work, where N is the number of points in the narrow band. For implementation details and further application of this technique, see Sethian (1995*c*, 1996) and Sethian, Adalsteinsson and Malladi (1996).

The above technique considered a flat initial interface for which trial values at the narrow band points could be easily initialized. Suppose we are given an arbitrary closed curve or surface as the initial location of the front. In this case, we use the original narrow band level set method to initialize the problem. First, label all grid points as 'far away' and assign them T values of ∞. Then, in a very small neighbourhood around the interface, construct the signed distance function from the initial hypersurface Γ. Propagate that surface both forwards and backwards in time until a layer of grid points is crossed in each direction, computing the signed crossing times as in Sethian (1994). Then collect the points with negative crossing times as 'alive' points with T value equal to the crossing time, and the points with positive crossing times as narrow band points with T value equal to the positive crossing times. Then begin the fast marching algorithm.

16. Other speed functions: when does this method work?

Our fast marching method as designed applied to speed functions F which depend only on position. In such cases, other forms for the gradient approximation can be used; for example, Rouy and Tourin (1992), namely

$$\begin{aligned} F_{ij}^{-1} = \max\Big(\max(D_{ij}^{-x}T, 0), -\min(D_{ij}^{+x}T, 0)\Big)^2 + \\ \max\Big(\max(D_{ij}^{-y}T, 0), -\min(D_{ij}^{+y}T, 0)\Big)^2 . \end{aligned} \tag{16.1}$$

How general is our new technique? Suppose now we consider the more general case of stationary level set equation:

$$|\nabla T| F = 1. \tag{16.2}$$

We begin by rewriting this in the standard form of a static Hamiltonian, namely

$$H(T_x, T_y, T_z) = 1. \tag{16.3}$$

We already have a scheme for the case where $H = \nabla T$. Some variations on this Hamiltonian important in computer vision, such as

$$H = \max(|T_x|, |T_y|, |T_z|) \qquad H = |T_x| + |T_y| + |T_z|, \tag{16.4}$$

may be approximated in a straightforward manner using any of the above entropy-satisfying approximations to the individual gradients. Our fast marching method will work in these cases. When the speed F depends in a subtle way on the value of ∇T (for example, in some problems in etching and deposition discussed in a later section), the situation is more delicate.

When will the technique work? Here, we present an intuitive perspective; complete details may be found in Sethian (1996) and Sethian et al. (1996). Suppose H is convex and always positive (or always negative), and suppose the approximation to H satisfies two properties:

1 the approximation scheme is consistent
2 at each grid point the scheme only makes use of smaller neighbouring values when updating the value at that point (this is the upwindness requirement), and cannot produce a new value which is less than any of the neighbours.

Then we can expect that our upwind sweeping method will work; searching for the smallest trial value will provide a consistent way of sweeping through the mesh and constructing the solution surface T. Complete details and many other schemes may be found in Sethian (1996) and Sethian et al. (1996).

17. Some clarifying comments

The time-dependent level set method and the stationary level set method each require careful construction of upwind, entropy-satisfying schemes, and make use of the dynamics and geometry of front propagation analysed in Sethian (1985). However, we note that the time-dependent level set method advances the front *simultaneously*, while the stationary method constructs 'scaffolding' to build the time solution surface T one grid point at a time. This means that the time at which the surface crosses a grid point (that is, its T value) may be found before other positions of that front at that time are determined. As such, there is *no* notion of a time step in the stationary method: one is simply constructing the stationary surface in an upwind fashion.

This means that if one is attempting to solve a problem in which the speed of a front depends on the current position of the front (such as in the case of visibility), or on subtle orientations in the front (such as in sputter yield problems), it is not clear how to use the stationary method, since the front is being constructed one grid point at a time.

The stationary method works because we were able to construct a simple approximation to the gradient. This was possible because the speed function F did *not* depend on the orientation of the front, nor on issues like visibility. Thus, returning to our earlier categorization of speed functions, our fast scheme works in cases where the speed F only depends on independent variables, such as in the case of photolithography development. Upwind entropy-satisfying schemes which can be transported to this fast stationary scheme for the case of more general speed functions F are more problematic, and discussed in detail in Sethian (1996).

To summarize,

- The stationary method is convenient for problems in which the front speed depends on independent variables, such as a photoresist rate function, and only applies if the speed function does not change sign.
- The time-dependent level set method is designed for more delicate speed functions, and can accurately evolve fronts under highly complex arguments.

In the next part, we discuss a variety of applications which employ both the time-dependent level set method and the fast marching method for monotonically advancing fronts.

PART V: APPLICATIONS

In this part, we present a series of applications of both the time-dependent level set method and the fast marching level set method to propagating interfaces. This is only a subset of the level set applications in the literature; throughout, we provide references to many other applications.

18. Geometry

In this section, we consider application of level set methods to problems in the geometric evolution of curves and surfaces. The motion will depend solely on local geometric properties such as normal direction and curvature; nonetheless, this is a fascinating and rich area.

18.1. Curvature flow

Suppose we are given a hypersurface in R^n propagating with some speed $F(\kappa)$. Previously, we have considered speed functions of the form $F(\kappa) = 1 - \epsilon\kappa$, where κ is the curvature. Let us now focus on a special speed function, namely $F = -\kappa$, where κ is the curvature. This corresponds to a geometric version of the heat equation; large oscillations are immediately smoothed out, and long-term solutions correspond to dissipation of all information about the initial state. As we shall see in later sections, curvature motion plays an important role in many applications such as a modelling term for surface tension in flexible membranes and a viscous term in physical phenomena.

The remarkable work of Gage and Grayson investigated the motion of a simple closed curve collapsing under its curvature. First, Gage showed that any convex curve moving under such a motion remains convex and must shrink to a point (Gage 1984, Gage and Hamilton 1986). Grayson (1987) followed this work with a stunning proof that *all* curves must shrink to a round point, regardless of their initial shape.

In Fig. 26, we take an odd-shaped initial curve and view this as the zero level set of a function defined in all of R^2. Here, for illustration, we have ϕ such that $\phi < 0$ as black and $\phi > 0$ as white, thus the zero level set is the boundary between the two. As the level curves flow under curvature, the ensuing motion carries each to a point, which then disappears. In the evolution of the front, one clearly sees that the large oscillations disappear quickly, and then, as the front becomes circular, motion slows, and the front eventually disappears.

In three dimensions, flow under mean curvature does not necessarily result in a collapse to a sphere. Huisken (1984) showed that convex shapes shrink to spheres as they collapse, analogous to the result of Gage. However, Grayson (1989) showed that non-convex shapes may in fact *not* shrink to a sphere,

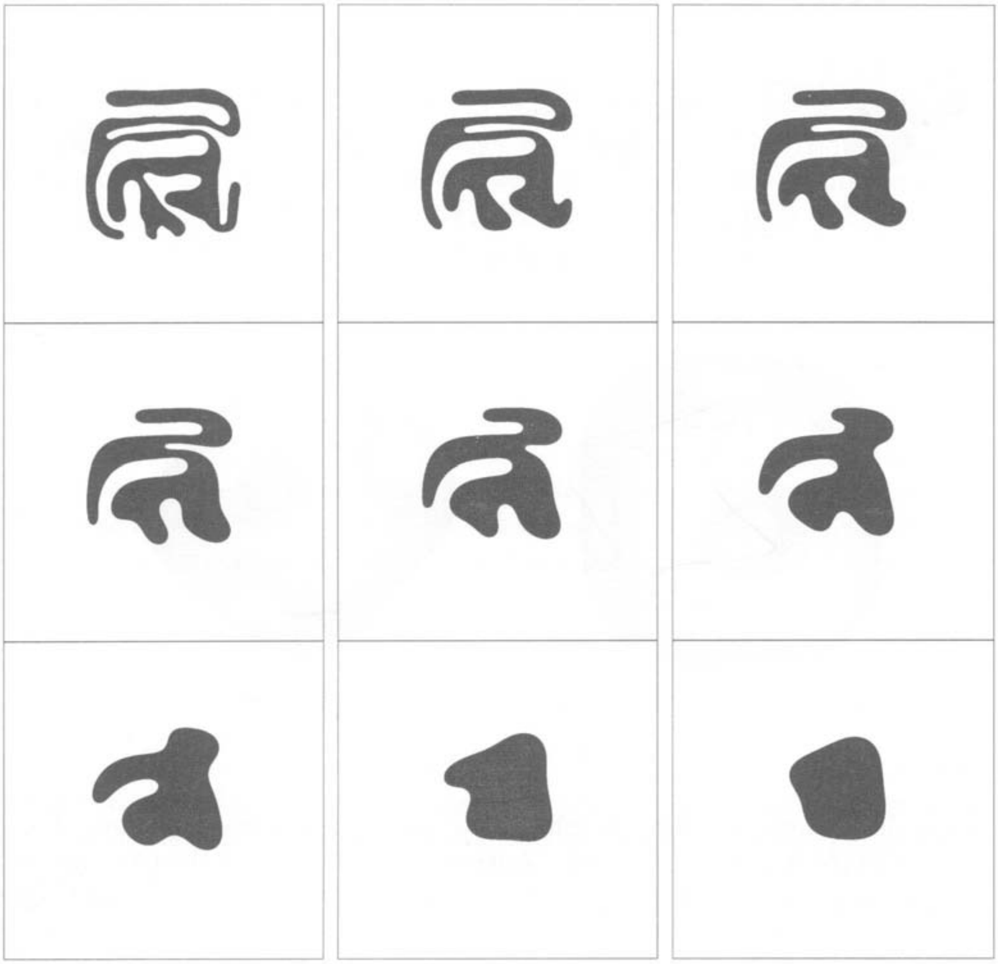


Fig. 26. $F(\kappa) = -\kappa$.

and provided the counterexample of the dumbbell. A narrow handle of a dumbbell may have such a high inner radius that the mean curvature of the saddle point at the neck may still be positive, and hence the neck will pinch off.

As illustration, in Fig. 27, taken from Chopp and Sethian (1993), we show two connected dumbbells collapsing under mean curvature. As the intersection point collapses, the necks break off and leave a remaining 'pillow' region behind. This pillow region collapses as well, and eventually all five regions disappear.

Finally, what about self-similar shapes? In two dimensions, it is clear that a circle collapsing under its own curvature remains a circle; this can be seen by integrating the ordinary differential equation for the changing radius. In

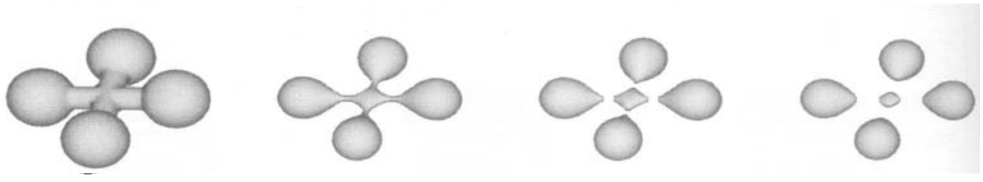
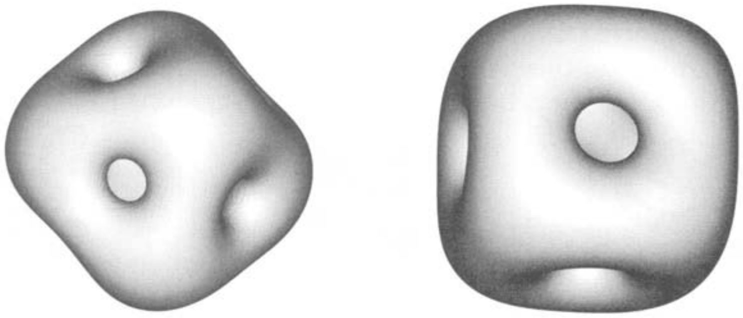


Fig. 27. Collapse of two-handled dumbbell.



(a) Self-similar cube with holes (b) Self-similar octahedron with holes

Fig. 28. Self-similar shapes.

three dimensions, a sphere is self-similar under mean curvature flow, since its curvature is always constant. Angenent (1992) proved the existence of a self-similar torus that preserves the balance between the competing pulls towards a ring and a sphere.

In order to devise an algorithm to produce self-similar shapes, two things are required. First, since hypersurfaces get smaller as they move under their curvature, a mechanism is needed to 'rescale' their motion so that the evolution can be continued towards a possible self-similar shape. Second, a way of pushing the evolving fronts back towards self-similarity is required. Chopp has accomplished both in a clever numerical algorithm that produces a family of self-similar surfaces; see Chopp (1994). His family comes from taking a regular polyhedron (for example, a cube), and drilling holes in each face. The resulting figure then evolves according to auxiliary level set equations, which contain the re-scaling as part of the equation of motion. Two such self-similar surfaces are shown in Fig. 28.

18.2. Grid generation

Imagine that one is given a closed body, either as a curve in two space dimensions, or a surface in three space dimensions. In many situations, one wishes to generate a logically rectangular, body-fitted grid around or inside this body. By logically rectangular, we mean that each node of the grid has four neighbours (in two dimensions; in three dimensions, there are six neighbours). By body-fitted, we mean that the grid aligns itself with the body so that one set of coordinate lines matches the body itself. This grid generation problem is difficult in part because of the competing desires of uniformity in cell area and mesh orthogonality; we refer the interested reader to Knupp and Steinberg (1993).

Level set techniques offer an interesting technique for generating such grids. The idea, as presented in Sethian (1994), is to exploit the geometric nature of the problem and view the body itself as the initial position of an interface that must be advanced outwards away from the body. The initial position of the interface and its position at later times forms one set of grid lines; its orthogonal set forms the other. The body is propagated outwards with speed $F = 1 - \epsilon\kappa$; by finding the zero level set at discrete times, the set of coordinate lines that encircle the body is found. Construction of transverse lines normal to the body are obtained by following trajectories of $\nabla\phi$. Additional node adjustment is possible through application of additional smoothing operators. Node placement on the boundary and the ensuing exterior/interior grid can be automatically controlled. For details, see Sethian (1994).

This technique can almost be viewed as a hyperbolic solver. However, by solving the correct evolution equation for an advancing front, we avoid the difficulties of shock formation and colliding characteristics that plague most hyperbolic techniques. User intervention is kept to a minimum; for the most part, grids are generated automatically without the need to adjust parameters.

In Fig. 29, we show a variety of grids constructed using this level set approach, starting with relatively smooth grids and ending with a three-dimensional grid around an indented dumbbell. As can be seen, interior and exterior grids can be created, with the capability of handling significant corners and cusps. The grids are automatically created; there has been no adjustment of parameters in the creation of these different grids.

18.3. Image enhancement and noise removal

The previous sections concerned geometrical motion of a particular hypersurface of interest. Next, we turn to a level set problem in which *all* the level sets have meaning, and must be evolved.

The goal in this section is to apply some of the level set methodology to image enhancement and noise removal. To do so, we first need a few definitions. Define an *image* to be an intensity map $I(x, y)$ given at each point of a two-dimensional domain. The range of the function $I(x, y)$ depends on the type of image: for monochrome images, the range is $\{0, 1\}$; for grey-scale images, $I(x, y)$ is a function mapping into $\{0, 1, \ldots, 255\}$; and for colour

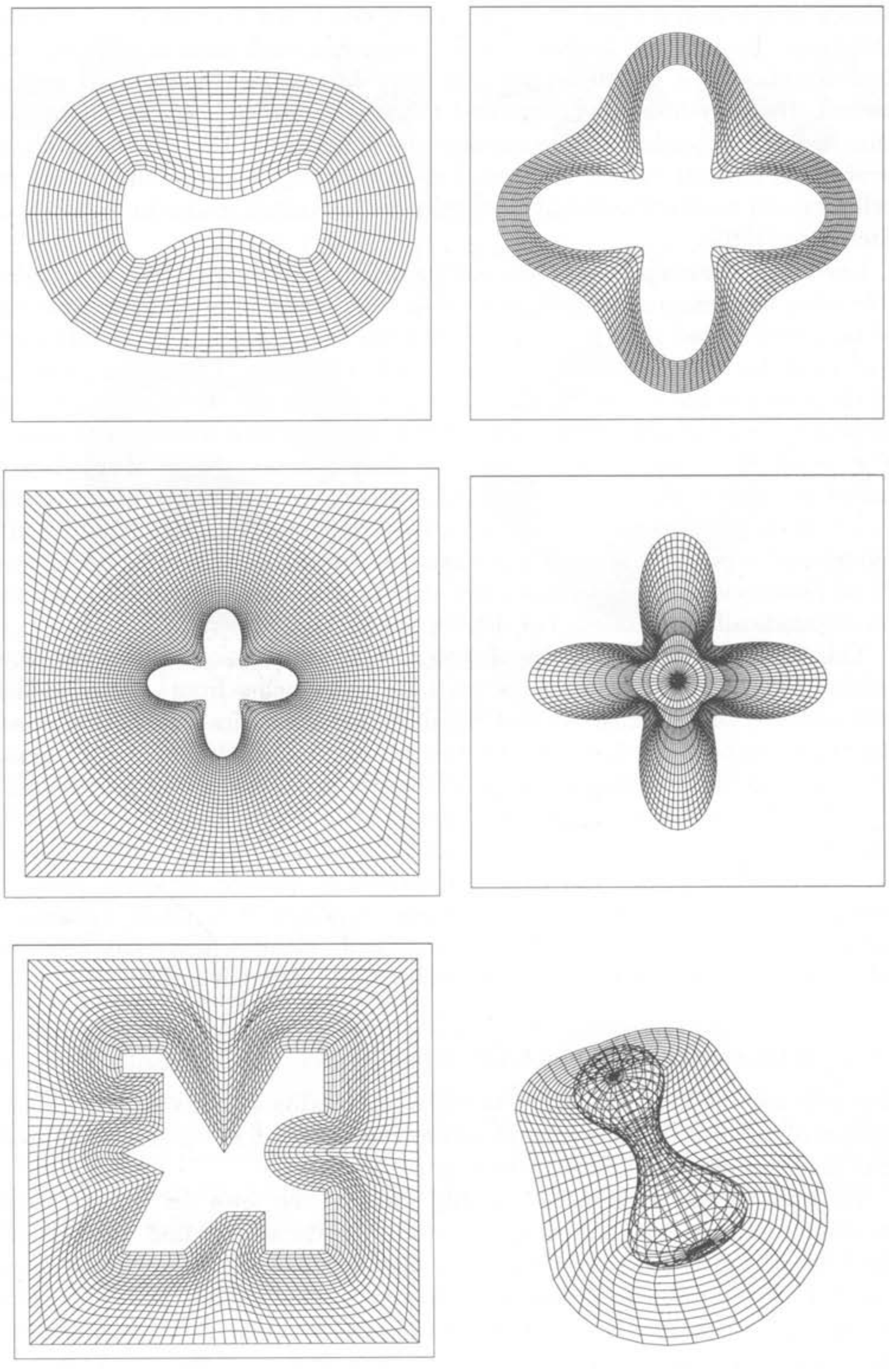


Fig. 29. Body-fitted grids generated using level set approach.

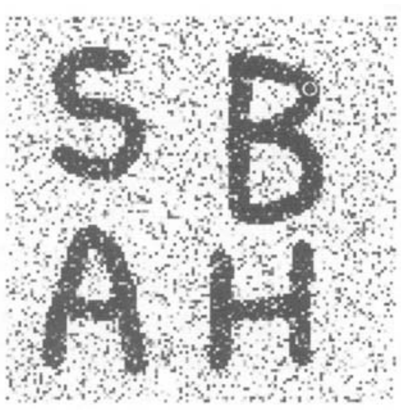


Fig. 30. An image $I(x, y)$ with noise.

images, $I(x, y)$ is a vector-valued function into some colour-space, typically either RGB or HSI.

Given such an image, two goals are to remove noise from the image (see Fig. 30) without sacrificing useful detail, and to enhance or highlight certain features. A straightforward and widely used approach is the Gaussian filter, in which both 1-D and 2-D signals are smoothed by convolving them with a Gaussian kernel; the degree of blurring is controlled by the characteristic width of the Gaussian filter. If the image is viewed as a surface, this convolution will reduce spikes as they blend into the background values. In this sense, the Gaussian removes noise. However, the Gaussian is an isotropic operator; it smoothes in all directions, and sharp boundaries will also be blurred. The goal is to improve upon this basic idea and remove noise without being forced into too much blurring. A variety of techniques has been introduced to improve upon this basic idea, including anisotropic diffusion schemes, which perform intraregion smoothing in preference to interregion smoothing (see Perona and Malik (1990)), as well as Wiener filters, and wavelet schemes.

Alvarez, Lions and Morel (1992) introduced a significant advancement in noise removal by employing, in part, some of the above ideas about curvature flow and level set equations. Consider the equation

$$I_t = F|\nabla I| \qquad \text{where} \qquad F = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|}\right) = \kappa. \tag{18.1}$$

This is our standard curvature evolution equation. An attractive quality of this motion is that sharp boundaries are preserved: smoothing takes place inside a region, but not across region boundaries. Of course, as shown by Grayson's theorem, eventually all information is removed as each contour shrinks to zero and vanishes.

An alternative approach is due to Rudin, Osher and Fatemi (1992). They take a total variation approach to the problem, which leads to a level set methodology and a very similar curvature-based speed function, namely $F(\kappa) = \kappa/|I|$; see also Osher and Rudin (1990). Following these works, a variation on these two approaches was produced by Sapiro and Tannenbaum (1993*b*). In that work, a speed function of the form $F(\kappa) = \kappa^{1/3}$ was employed. In each of these schemes, all information is eventually removed through continued application of the scheme. Thus, a 'stopping criterion' is required.

A recent level set scheme for noise removal and image enhancement was introduced by Malladi and Sethian (1995) and Malladi and Sethian (1996*a*). The scheme results from returning to the original ideas of curvature flow, and exploiting a 'min/max' function, which correctly selects the optimal motion to remove noise. It has two highly desirable features:

1 there is an intrinsic, adjustable definition of scale within the algorithm, such that all noise below that level is removed, and all features above that level are preserved

2 the algorithm stops automatically once the sub-scale noise is removed; continued application of the scheme produces no change.

These two features are quite powerful, and lead to a series of open questions about the morphology of shape and asymptotics of scale-removal; for details, see Malladi and Sethian (1996*a*).

The min/max flow
Consider the equation

$$\phi_t = \tilde{F}|\nabla\phi|. \tag{18.2}$$

A curve collapsing under its curvature will correspond to speed $\tilde{F} = \kappa$. Now, consider two variations on the basic curvature flow, namely

$$\tilde{F}(\kappa) = \min(\kappa, 0.0) \qquad \tilde{F}(\kappa) = \max(\kappa, 0.0).$$

Here, we have chosen the negative of the signed distance in the interior, and the positive sign in the exterior region. The effect of flow under $\tilde{F}(\kappa) = \min(\kappa, 0.0)$ is to allow the inward concave fingers to grow outwards, while suppressing the motion of the outward convex regions. Thus, the motion halts as soon as the convex hull is obtained. Conversely, the effect of flow under $\tilde{F}(\kappa) = \max(\kappa, 0.0)$ is to allow the outward regions to grow inwards while suppressing the motion of the inward concave regions. However, once the shape becomes fully convex, the curvature is always positive and the flow becomes the same as regular curvature flow.

Our goal is to select the correct choice of flow that smoothes out small oscillations, but maintains the essential properties of the shape. In order to do so, we discuss the idea of the min/max switch.

Consider the following speed function, introduced in Malladi and Sethian (1995) and refined considerably in Malladi and Sethian (1996*a*):

$$F^{\text{Stencil}=k}_{\text{min}\,/\,\text{max}} = \begin{cases} \min(\kappa, 0) & \text{if } \text{Ave}^{R=kh}_{\phi(x,y)} < 0 \\ \max(\kappa, 0) & \text{if } \text{Ave}^{R=kh}_{\phi(x,y)} \geq 0, \end{cases} \tag{18.3}$$

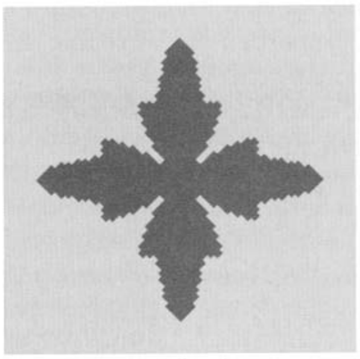
where $\text{Ave}^{R=kh}_{\phi(x,y)}$ is defined as the average value of ϕ in a disk of radius $R = kh$ centred around the point (x, y). Here, h is the step size of the grid. Thus, given a *StencilRadius* k, the above yields a speed function that depends on the value of ϕ at the point (x, y), the average value of ϕ in neighbourhood of a given size, and the value of the curvature of the level curve going through (x, y).

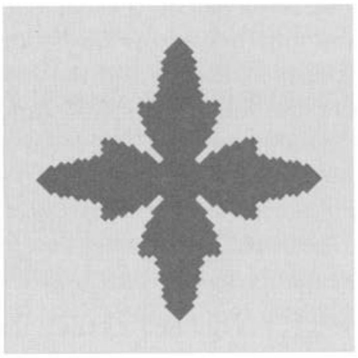
We can examine this speed function in some detail. For ease of exposition, consider a black region on a white background, chosen so that the interior has a negative value of ϕ and the exterior a positive value.

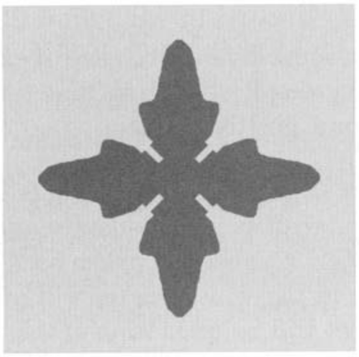
StencilRadius $k = 0$ If the radius $R = 0$ $(k = 0)$, then choice of $\min(\kappa, 0)$ or $\max(\kappa, 0)$ depends only on the value of ϕ. All the level curves in the black region will attempt to form their convex hull, when seen from the black side, and all the level curves in the white region will attempt to form *their* convex hull. The net effect will be no motion of the zero level set itself, and the boundary will not move.

StencilRadius $k = 1$ If the average is taken over a stencil of radius kh, then some movement of the zero level corresponding to the boundary is possible. If there are some oscillations in the front boundary on the order of one or two pixels, then the average value of ϕ at the point (x, y) can have a different sign from the value at (x, y) itself. In this case, the flow will act as if it were selected from the 'other side', and some motion will be allowed until these first-order oscillations are removed, and a balance between the two sides is again reached. Once this balance is reached, further motion is suppressed.

StencilRadius $k > 1$ By taking averages over a larger stencil, larger amounts of smoothing are applied to the boundary. In other words, decisions about where features belong are based on larger and larger perspectives. Once features on the order of size k are removed from the boundary, balance is reached and the flow stops automatically. As an example, let $k = \infty$. Since the average will compute to a value close to the background colour, on this scale all structures are insignificant, and the max flow will be chosen everywhere, forcing the boundary to disappear.


(a) Initial boundary 'noisy' shape


(b) Min/max flow: stencil radius $= 0; (T = \infty)$


(c) Min/max flow: stencil radius $= 1; (T = \infty)$

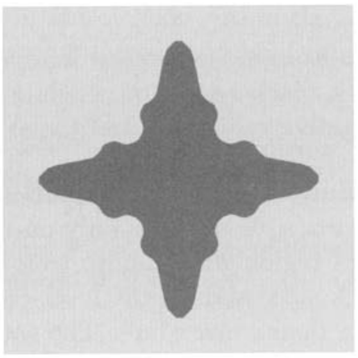

(d) Continued min/max flow: stencil radius $= 2; (T = \infty)$

Fig. 31. Motion of star-shaped region with noise under min/max flow at various stencil levels.

To show the results of this hierarchical flow, we start with an initial shape in Fig. 31a and first perform the min/max flow until steady state is reached with stencil size zero in Fig. 31b. In this case, the steady state is achieved immediately, and the final state is the same as the initial state. Min/max flow is then performed until steady state is achieved with stencil size $k = 1$ in Fig. 31c, and then min/max flow is again applied with a larger stencil until steady state is achieved in Fig. 31d.

These results can be summarized as follows:

- the min/max flow switch selects the correct motion to diffuse the small-scale pixel notches into the boundary
- the larger, global properties of the shape are maintained

- furthermore, and equally importantly, the flow stops once these notches are diffused into the main structure
- edge definition is maintained and, in some global sense, the area inside the boundary is preserved
- the noise removal capabilities of the min/max flow are scale-dependent, and can be hierarchically adjusted
- the scheme requires only a nearest neighbour stencil evaluation.

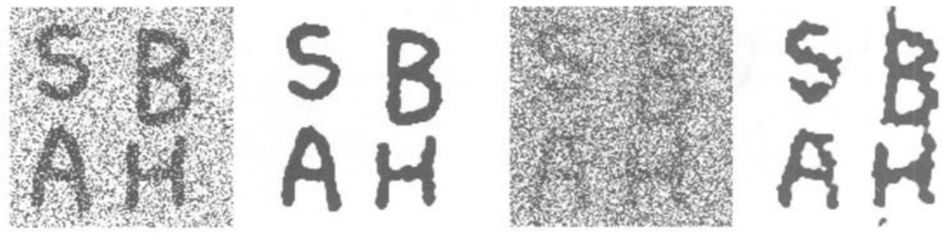
Extension of min/max scheme to grey-scale, texture, and colour images
The above technique applies to monochrome images. An extension to grey-scale images can be made by replacing the fixed threshold test value of 0 with a value that depends on the local neighbourhood. As designed in Malladi and Sethian (1995), let $T_{\text{threshold}}$ be the average value of the intensity obtained in the direction perpendicular to the gradient direction. Note that since the direction perpendicular to the gradient is tangent to the iso-intensity contour through (x, y), the two points used to compute are either in the same region, or the point (x, y) is an inflection point, in which the curvature is in fact zero and the min/max flow will always yield zero. By choosing a larger stencil we mean computing this tangential average over endpoints located further apart.
Formally then, our min/max scheme becomes:

$$\tilde{F}_{\min/\max} = \begin{cases} \max(\kappa, 0) & \text{if Average}(x, y) < T_{\text{threshold}} \\ \min(\kappa, 0) & \text{otherwise.} \end{cases} \tag{18.4}$$

Further details about this scheme may be found in Malladi and Sethian (1996*a*). In that work, these techniques are applied to a wide range of images, including salt-and-pepper, multiplicative and Gaussian noise applied to monochrome, grey-scale, textured, and colour images.

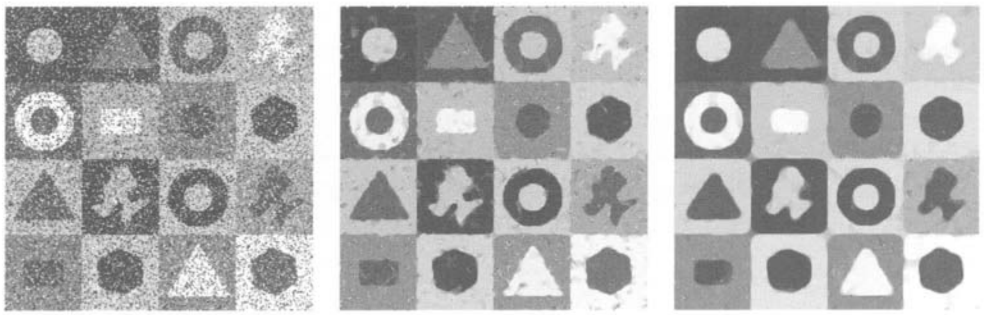
Results
In this section, we provide a few examples of this min/max flow. In Fig. 32, 50% and 80% grey-scale noise is added to a monochrome image of a handwritten character. The noise is added as follows: $X\%$ noise means that at $X\%$ of the pixels, the given value is replaced with a number chosen with uniform distribution between 0 and 255. Here, the min/max switch function is taken relative to the value 127.5 rather than zero. The restored figures are converged. Continued application of the scheme yields almost no change in the results.

Next, salt-and-pepper grey-scale noise is removed from a grey-scale image. The results are obtained as follows. Begin with 40% noise in Fig. 33a. First, the min/max flow from equation (18.4) is applied until a steady state is reached (Fig. 33b). This removes most of the noise. The scheme is then continued with a larger threshold stencil for the threshold to remove further noise (Fig. 33c). For the larger stencil, we compute the average over a larger



(a) 50% noise (b) Restored (c) 80% noise (d) Restored

Fig. 32. Image restoration of binary images with grey-scale salt-and-pepper noise using min/max flow: restored shapes are final shape obtained ($T = \infty$).



(a) 40% noise (b) Min/max flow (c) Cont: larger stencil

Fig. 33. Min/max flow applied to grey-scale image.

disk and calculate the threshold value $T_{\text{threshold}}$ using a correspondingly longer tangent vector.

Finally, we show the effect of this scheme applied to an image upon which 100% Gaussian grey-scale noise has been superimposed; a random component drawn from a Gaussian distribution with mean zero is added to each (every) pixel. Fig. 34 shows the 'noisy' original together with the reconstructed min/max flow image.

19. Combustion, crystal growth, and two-fluid flow

In this section, we consider some applications of the level set methodology to a large and challenging class of interface problems. These problems are characterized by physical phenomena in which the front acts as a boundary condition to a partial differential equation, and the solution of this equation controls the motion of the front. In combustion problems, the interface is a flame, and both exothermic expansion along the front and flame-induced vorticity drive the underlying fluid mechanics. In crystal growth and dendritic

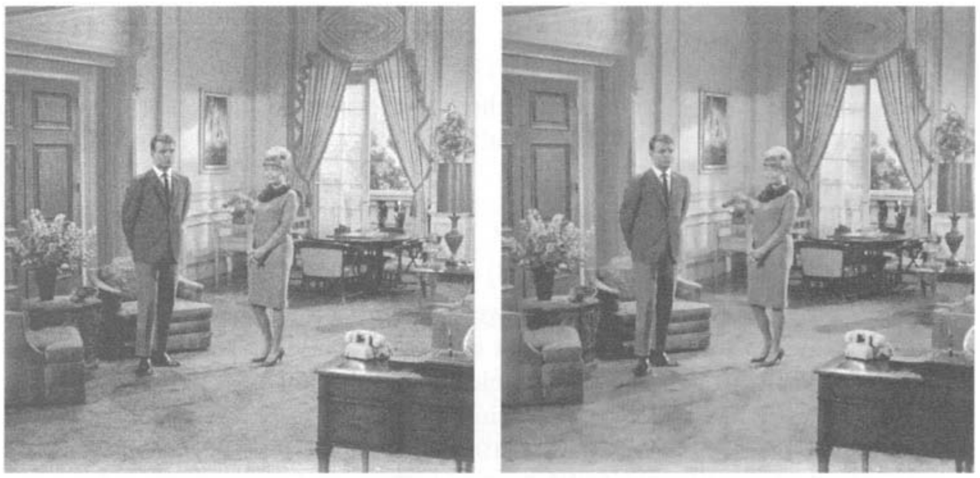


Fig. 34. Continuous Gaussian noise added to image.

formation, the interface is the solid/liquid boundary, and is driven by a jump condition related to heat release along the interface. In two-fluid problems, the interface represents the boundary between two immiscible fluids of significantly different densities and/or viscosities, and the surface tension along this interface plays a significant role in the motion of the fluids.

From an algorithmic perspective, the significant level set issue is that information about the speed of the front itself must be somehow transferred to the Eulerian framework that updates the level set function at the fixed grid points. This is a significant challenge for two reasons.

- In many situations, the interface velocity is determined by the interaction of local geometric quantities of the front itself (such as curvature) with global variables on either side of the interface (for example, jumps in velocity, heat, or concentration of species). If these global variables are calculated on a grid, it may be difficult to extend the values to the front itself where they are required to evaluate the speed function F. However, these quantities are only known at grid points, not at the front itself, where the relationship has meaning.
- It may be very difficult to extend this interface velocity back to the grid points (that is, to the other level sets). This problem, known as the *extension problem* (see Adalsteinsson and Sethian (1995*b*, 1995*c*)) must be solved in order for the level set method to work; some mechanism of updating the grid points in the neighbourhood of the zero level set is required.

In this section, we discuss three application areas where these problems have been solved. The common link in these sections is the presence of a term in the equation represented by a Dirac delta function along the interface. The interested reader is referred to the literature cited below for a detailed description of the algorithms and results.

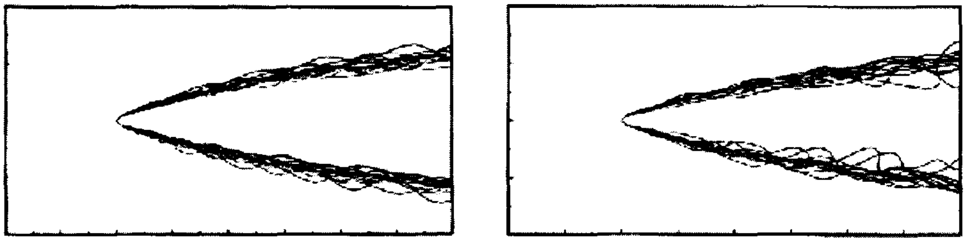
19.1. Turbulent combustion of flames and vorticity, exothermicity, flame stretch and wrinkling

In Rhee et al. (1995), a flame is viewed as an infinitely thin reaction zone, separating two regions of different but constant densities. The hydrodynamic flow field is two dimensional and inviscid, and the Mach number is vanishingly small. This corresponds to the equations of zero Mach number combustion, introduced in Majda and Sethian (1984). The flame propagates into the unburnt gas at a prescribed flame speed S_u, which depends on the local curvature, due to the focusing/de-focusing of heating effects as a function of flame shape.

As the reactants are converted to products (that is, as the material makes the transition from 'unburnt' to 'burnt'), the local fluid undergoes volume increase known as exothermic expansion, associated with the density jump. At the same time, pressure gradients tangential to the flame cause different accelerations in the light and heavy gases. This causes a production of vorticity (known as baroclinic torque) across the flame, since the pressure gradient is not always aligned with the density gradient. Together, the burning of the flame acts as source of vorticity and volume for the underlying hydrodynamic field, both of which in turn affect the evolution of the flame interface.

This model presents a significant challenge for a level set method. The flame is tracked by identifying the flame interface as the zero level set of the level set function. The curvature is determined using the expression given in equation (11.1). The vortical field is represented by a collection of vortex blobs as in vortex method; see Chorin (1973) and Sethian (1991). The exothermic field is determined by solving a Poisson's equation on the underlying grid with right-hand side given by smearing the Dirac delta function to the neighbouring grid points. The no-flow boundary is satisfied by the addition of a potential flow that exactly cancels the existing flow field. Finally, the tangential stretch component is evaluated by tracing the values of the tangential velocity from the given position backwards along the normal to the front to evaluate the change in tangential velocity. For complete details, see Rhee et al. (1995).

Fig. 35, taken from Rhee et al. (1995), shows two results from this algorithm. We compare an anchored flame, with upstream turbulence imposed by a statistical distribution of positive and negative vortices. The goal here is to understand the effect of exothermicity and flame-induced vorticity on



(a) Flame with exothermicity (b) Flame with exothermicity and vorticity

Fig. 35. Comparison of flame brush.

the flame wrinkling and stability. In Fig. 35a, we show an anchored flame in the oncoming turbulent field; here, different time snapshots are superimposed upon each other to show the flame 'brush'. In Fig. 35b, we turn on the effects of both volume expansion (a large density jump) and vorticity generation along the flame front. The resulting flow field generates a significantly wider flame brush, as the vorticity induces flame wrinkling and the exothermicity affects the surrounding flow field.

In the above application of the level method, the front acted as source of volume and vorticity. In the next application, developed in Sethian and Strain (1992), the interface motion is controlled by a complex jump condition.

19.2. Crystal growth and dendritic solidification

Imagine a container filled with a liquid such as water, which has been so smoothly and uniformly cooled below its freezing point that the liquid does not freeze. The system is now in a 'metastable' state, where a small disturbance such as dropping a tiny seed of the solid phase into the liquid will initiate a rapid and unstable process known as *dendritic solidification.* The solid phase will grow from the seed by sending out branching fingers into the distant cooler liquid nearer the undercooled wall. This growth process is *unstable* in the sense that small perturbations of the initial data can produce large changes in the time-dependent solid/liquid boundary.

Mathematically, this phenomenon can be modelled by a moving boundary problem. The temperature field satisfies a heat equation in each phase, coupled through two boundary conditions on the unknown moving solid/liquid boundary, as well as initial and boundary conditions. First, the normal velocity of the interface depends on the jump in the normal component of the heat flux across the interface. Second, the temperature at the interface itself depends on the local curvature and the velocity. Thus, the goal is to incorporate the influence of the front on the heat solvers across the interface. For further details, see Cahn and Hilliard (1958) and Mullins and Sekerka (1963).

A variety of techniques is possible to approximate numerically these equations of motion. One approach is to solve the heat equation in each phase and try to move the boundary so that the two boundary conditions are satisfied. Another approach is to recast the equations of motion as a single integral equation on the moving boundary and solve the integral equation numerically.

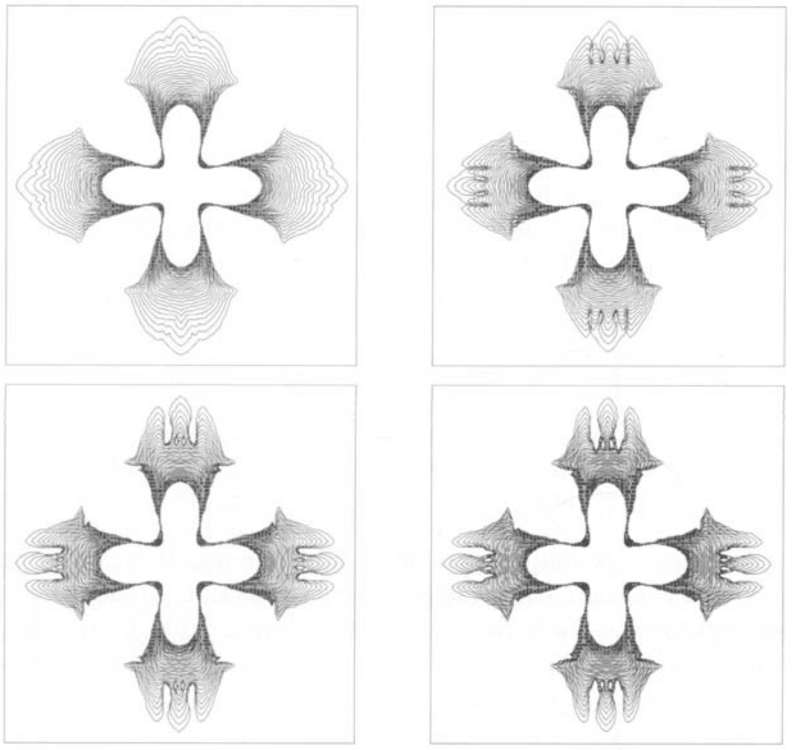
In Sethian and Strain (1992) a hybrid level set/boundary integral approach was developed, which includes the effects of undercooling, crystalline anisotropy, surface tension, molecular kinetics, and initial conditions. The central idea is to exploit a transformation due to Strain (1989), which converts the equations of motion into a single, history-dependent boundary integral equation on the solid/liquid boundary, given by

$$\epsilon_\kappa(n)\kappa + \epsilon_V(n)V + U + H\int_0^t \int_{\Gamma(t')} K(x,x',t-t')\, V(x',t')\,\mathrm{d}x'\,\mathrm{d}t' = 0 \quad (19.1)$$

for all x on the interface $\Gamma(t)$. Here, K is the heat kernel, ϵ_κ and ϵ_V are constants, U is the temperature, V is the normal velocity of the interface, and H is the latent heat of solidification. Note that the velocity V depends not only on the position of the front but also on its previous history. Thus, as shown in Strain (1989), information about the temperature off the front is stored in the previous history of the boundary. This can be evaluated by a combination of fast techniques; see Strain (1988, 1989, 1990) and Greengard and Strain (1990).

In Fig. 36, we show one example from Sethian and Strain (1992) in which the effect of changing the latent heat of solidification H is analysed. Since the latent heat controls the balance between the pure geometric effects and the solution of the history-dependent heat integral, increasing H puts more emphasis on the heat equation/jump conditions. Calculations are performed on a unit box, with a constant undercooling on the side walls of $u_B = -1$. The kinetic coefficient is $\epsilon_V = .001$; the surface tension coefficient is $\epsilon_\kappa = .001$; there is no crystalline anisotropy. The initial shape was a perturbed circle. A 96×96 mesh is used with time step $\Delta t = .00125$. The calculations are all plotted at the same time.

In the calculations shown, H is varied smoothly. In Fig. 36a, $H = .75$; the dominance of geometric motion serves to create a rapidly evolving boundary that is mostly smooth. H is increased in each successive figure, ending with $H = 1.0$ in Fig. 36d. As the latent heat of solidification is increased, the growing limbs expand outwards less smoothly, and instead develop flat ends. These flat ends are unstable and serve as precursors to tip splitting. We also note that the influence of the heat integral slows down the evolving boundary, as witnessed by the fact that all the plots are given at the same time. Presumably, increasing latent heat decreases the most unstable wavelength, as described by linear stability theory. The final shape shows side-branching, tip splitting, and the strong effects of the side walls.



Upper left: $H = .75$ Upper right: $H = .833$
Lower left: $H = .916$ Lower right: $H = 1.0$

Fig. 36. Effect of changing latent heat.

20. Two-phase flow

In this section, we discuss level set methods applied to problems of two-phase flow.

Two early applications of fluid dynamical problems using level set methods to track the interface are the projection method calculations of compressible gas dynamics of Mulder, Osher and Sethian (1992) and the combustion calculations of Zhu and Sethian (1992). Each viewed the interface as the zero level set, and tracked this interface as a method of separating the two regions.

Firstly, in Mulder et al. (1992) the evolution of rising bubbles in compressible gas dynamics was studied. The level set equation for the evolving interface separating two fluids of differing densities was incorporated inside the conservation equations for the fluid dynamics. Both the Kelvin–Helmholtz instability and the Rayleigh–Taylor instability were studied; the density ratio was about 30 to 4, and both gases were treated as perfect gases. Considerable discussion was devoted to the advantages and disadvantages of embedding the level set equation as an additional conservation law.

Next, in the combustion calculations presented in Zhu and Sethian (1992), the interface represented a flame propagating from the burnt region into the unburnt region. Unlike the above calculations concerning flame stability in flame holders, in these calculations the flame was viewed as a 'cold flame'; that is, the hydrodynamic flow field affected the position of the flame, but the advancing flame did not in turn affect the hydrodynamic field. In these calculations, the hydrodynamic field was computed using Chorin's projection method (see Chorin (1968)), coupled to the level set approach. The problem under study was the evolution of a flame inside a swirling two-dimensional chamber; and the results showed the intermixing that can occur, the creating of pockets of unburnt fuel surrounded by burnt pockets.

These two works were followed by the work of Chang, Hou, Merriman and Osher (1994) and Sussman et al. (1994), using projection methods coupled to the level set equation to study the motion of incompressible, immiscible fluids where steep gradients in density and viscosity exist across the interface, and the role of surface tension is crucial.

There are three key aspects of these fascinating calculations. First, using a formulation first developed by Brackbill, Kothe and Zemach (1992), the surface tension generated by the level set curvature expression can be smoothed using a mollified delta function to the neighbouring grid points; this smoothing denotes a 'thickness' for the interface layer, and allows the role of surface tension to be transported to the grid for inclusion in the projection method. Second, the contribution due to surface tension is converted from a delta function to the Heaviside formulation, and incorporated as such into the projection step. This eliminates the standard numerical instabilities and oscillations that plague attempts to directly difference the delta function itself. Third, the level set lines cease to correspond to the distance function due to the significant variation in fluid velocities across the interface; to redistribute the level set contours, the re-initialization idea using iteration on the sign of ϕ described in equation (13.1) was developed.

The calculations performed using these techniques show a wide range of applications concerning falling drops, colliding drops, and the role of surface tension, and open the door to a range of important fluid dynamics applications. We refer the reader to Sussman et al. (1994) and Chang et al. (1994) for further details.

21. Constrained problems: minimal surfaces and shape recovery

Another set of applications include problems in which the motion of the front is constrained by external boundary conditions. In this section, we briefly review some work on the construction of minimal surfaces and shape recovery from images.

21.1. Minimal surfaces

The basic problem may be stated as follows. Consider a closed curve $\Gamma(s)$ in R^3. The goal is to construct a membrane with boundary Γ and mean curvature zero. In some cases, this can be achieved as follows. Given the bounding wire frame Γ, consider some initial surface $S(t = 0)$ whose boundary is Γ. Let $S(t)$ be the family of surfaces parametrized by t obtained by allowing the initial surface $S(t = 0)$ to evolve under mean curvature, with boundary always given by Γ. Defining the surface S by $S = \lim_{t\to\infty} S(t)$, one expects that the surface S will be a minimal surface for the boundary Γ. Several computational approaches exist to construct such minimal surfaces based on this approach, including Brakke's Surface Evolver program (Brakke 1990).

A level set approach to this problem rests on embedding the motion of the surface towards its minimal energy as the zero level set of a higher-dimensional function. Thus, given an initial surface $S(0)$ passing through Γ, construct a family of neighbouring surfaces by viewing $S(0)$ as the zero level set of some function ϕ over all of R^3. Using the level set equation (3.5), evolve ϕ according to the speed law $F(\kappa) = -\kappa$. Then a possible minimal surface S will be given by

$$S = \lim_{t\to\infty} \{x : \phi(x,t) = 0\}. \tag{21.1}$$

The difficult challenge with the above approach is to ensure that the evolving zero level set always remains attached to the boundary Γ. This is accomplished in Chopp (1993), by creating boundary conditions of grid points closest to the wire frame linking together the neighbouring values of ϕ, to force the level set $\phi = 0$ through Γ. Thus we obtain a constrained level set problem: we track mean curvature flow requiring that the evolving zero level set remains attached to the front.

In Fig. 37, taken from Chopp (1993), the minimal surface spanning two rings each of radius 0.5 and at positions $x = \pm.277259$ is computed. A cylinder spanning the two rings is taken as the initial level set $\phi = 0$. A $27 \times 47 \times 47$ mesh with space step 0.025 is used. The final shape is shown in Fig. 37.

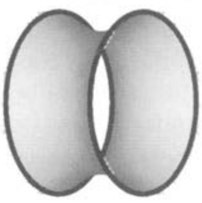
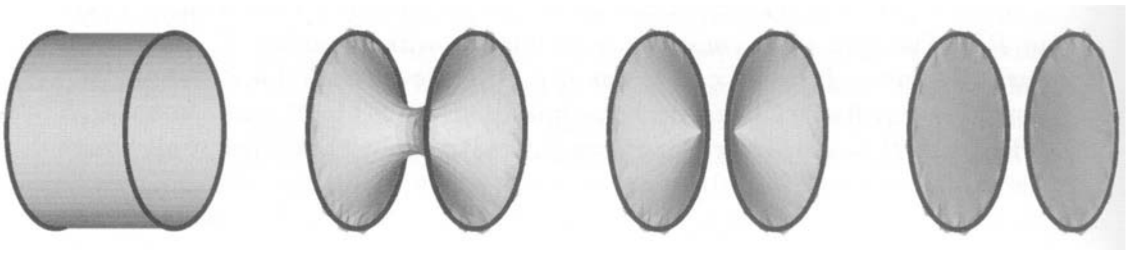


Fig. 37. Minimal surface: catenoid.



Time $= 0.0$ Time $= 0.41$ Time $= 0.42$ Time $= 0.50$

Fig. 38. Splitting of catenoid.

Next, in Fig. 38 (again taken from Chopp (1993)), this same problem is computed, but the rings are placed far enough apart so that a catenoid solution cannot exist. Starting with a cylinder as the initial surface, the evolution of this surface is computed as it collapses under mean curvature while remaining attached to the two wire frames. As the surface evolves, the middle pinches off and the surface splits into two surfaces, each of which quickly collapses into a disk. The final shape of a disk spanning each ring is indeed a minimal surface for this problem. This example illustrates one of the virtues of the level set approach. No special cutting or *ad hoc* decisions are employed to decide when to break the surface. Instead, viewing the zero level set as but one member of a family of flowing surfaces allows this smooth transition. Further results may be found in Chopp (1993).

21.2. Shape detection/recovery

Imagine that one is given an image. The goal in *shape detection/recovery* is to extract a particular shape from that image; here, 'extract' means to produce a mathematical description of the shape, which can be used in a variety of forms. The work on level set techniques applied to shape recovery described here was first presented in Malladi et al. (1994); further work using the level set scheme in the context of shape recovery may be found in Malladi, Sethian and Vemuri (1995*b*), Malladi and Sethian (1994), Malladi, Adalsteinsson and Sethian (1995*a*) and Caselles, Catte, Coll and Dibos (n.d.). We refer the interested reader to those papers for motivation, details, and a large number of examples.

Imagine that we are given an image, with the goal of isolating a shape within the image. Our approach (see Malladi et al. (1994)) is motivated by the active force contour/snake approach to shape recovery given by Kass, Witkin and Terzopoulos (1988). Consider a speed function of the form $1 - \epsilon\kappa$ $(1 + \epsilon\kappa)$, where ϵ is a constant. As discussed earlier, the constant acts as an advection term, and is independent of the moving front's geometry. The front uniformly expands (contracts) with speed 1 (-1) depending on the sign, and is analogous to the inflation force defined in Cohen (1991). The diffusive second term ϵK depends on the geometry of the front and smooths out the high curvature regions of the front. It has the same regularizing effect on the front as the internal deformation energy term in thin-plate-membrane splines (Kass et al. 1988).

Our goal now is to define a speed function from the image data that acts as a halting criterion for this speed function. We multiply the above speed function by the term

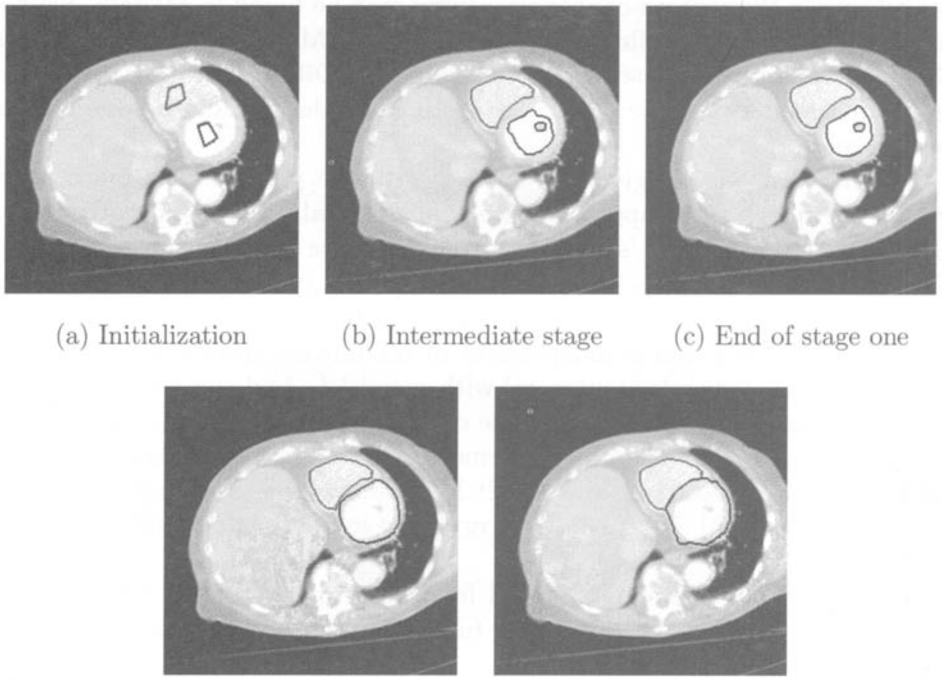
$$k_I(x,y) = \frac{1}{1+ \mid \nabla G_\sigma * I(x,y) \mid}, \tag{21.2}$$

where the expression $G_\sigma * I$ denotes the image convolved with a Gaussian smoothing filter whose characteristic width is σ. The term $|\nabla G_\sigma * I(x,y)|$ is essentially zero except where the image gradient changes rapidly, in which case the value becomes large. Thus, the filter $k_I(x,y)$ is close to unity away from boundaries, and drops to zero near sharp changes in the image gradient, which presumably corresponds to the edge of the desired shape. In other words, the filter function anticipates steep changes in the image gradient, and stops the evolving front from passing out of the desired region.

Thus the algorithm works as follows. A small front (typically a circle) is started inside the desired region. This front then grows outwards and is stopped at the shape boundary by the filter term, which drops the value of the speed function F to zero.

There are several desirable aspects of this approach:

- the initial front can consist of many fronts; due to the topological capabilities of the level set method, these fronts will merge into a single front as it grows into the particular shape
- the front can follow intricate twists and turns in the desired boundary
- use of narrow band techniques makes the algorithm very fast
- the technique can be used to extract three-dimensional shapes as well by initializing in a ball inside the desired region


(a) Initialization (b) Intermediate stage (c) End of stage one

(d) Intermediate stage two (e) End of stage two

Fig. 39. Shape extraction from heart data.

- small isolated spots of noise where the image gradient changes substantially are ignored; the front propagates *around* these points and closes back in on itself and then disappears.

As a demonstration, level set shape recovery techniques are applied to the difficult problem of extracting images of the left and right ventricles of the heart. In these calculations, taken from Malladi and Sethian (1996*b*), the problem is initialized by simultaneously tagging both the left and the right ventricle; the right ventricle is found by the evolving front, as is the left ventricle. Note that in the evolution of the right ventricle front, the papillary muscle is also found; see Fig. 39. This feature is obtained by initializing with a single contour, enclosing the papillary muscle and separating into an inner ring and outer ring. After the outer walls of the left and right ventricles are recovered, the outer wall of the right ventricle is extracted; this is done by temporarily relaxing the stopping criterion, and allowing the front to move past the inner wall of the right ventricle. Once this occurs, the stopping criterion is turned back on, and the front expands until the outer wall is found.

This technique for shape detection/recovery can be performed in three dimensions if three-dimensional data is available. For details of this and related work, see Malladi et al. (1994), Malladi et al. (1995*b*) and Malladi and Sethian (1996*b*).

22. Applications of the fast marching level set method

In the case of a monotonically advancing front whose speed in the normal direction depends only on position, we have previously seen that this can be converted into a stationary time problem. Furthermore, we have developed a fast marching algorithm for solving the Eikonal equation associated with this problem. Here, we show two applications of this technique.

22.1. Shape-from-shading

Suppose we illuminate a non-self-shadowing surface from a single point light source. At each point of the surface, one can define the brightness map I which depends on the reflectivity of the surface and the angle between the incoming light ray and the surface normal. Points of the surface where the normal is parallel to the incoming beam are brightest; those where the normal is almost orthogonal are darkest (again, we rule out surfaces that are self-shadowing). The goal of *shape-from-shading* is to reconstruct the surface from its brightness function I.

We point out right away that the problem as posed does not have a unique solution. For example, imagine a beam coming straight down; it is impossible to differentiate a surface from its mirror image from the brightness function. That is, a deep valley could also be a mountain peak. Other ambiguities can exist, we refer the reader to Rouy and Tourin (1992) and Kimmel and Bruckstein (1992). Nonetheless, in its simplest form the shape-from-shading problem provides a simple example of an Eikonal equation that can be solved using our fast marching level set method.

We begin by considering a surface $T(x, y)$; the surface normal is then given by

$$n = \frac{(-T_x, -T_y, 1)}{(|\nabla T|^2 + 1)^{1/2}}. \tag{22.1}$$

Let (α, β, γ) be the direction from the light source. In the simplest case of a Lambertian surface, the brightness map is given in a very simple form by

$$I(x, y) = (\alpha, \beta, \gamma) \cdot n. \tag{22.2}$$

Thus, the shape-from-shading problem is to reconstruct the surface $T(x, y)$ given the brightness map $I(x, y)$.

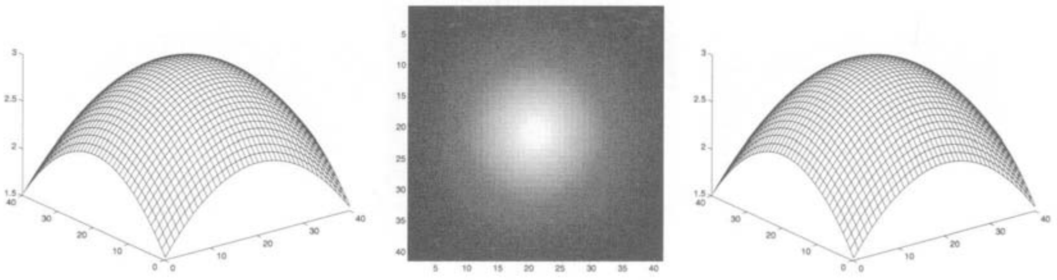

(a) Original (b) Brightness (c) Reconstructed

Fig. 40. Shape-from-shading reconstruction of paraboloid surface.

Consider the simplest case, namely that in which the light comes from straight down. Then the light source vector is $(0, 0, 1)$, and we have

$$I(x,y) = \frac{1.}{(|\nabla T|^2 + 1)^{1/2}}. \tag{22.3}$$

Rearranging terms, we then have an Eikonal equation for the surface, namely

$$|\nabla T| = \sqrt{\frac{1}{I^2} - 1}, \tag{22.4}$$

where n is the normal to the surface. We still need initial conditions for this problem. Let us imagine that at extrema of T we know the values of T. Then we can construct a viable solution surface using our fast marching method.

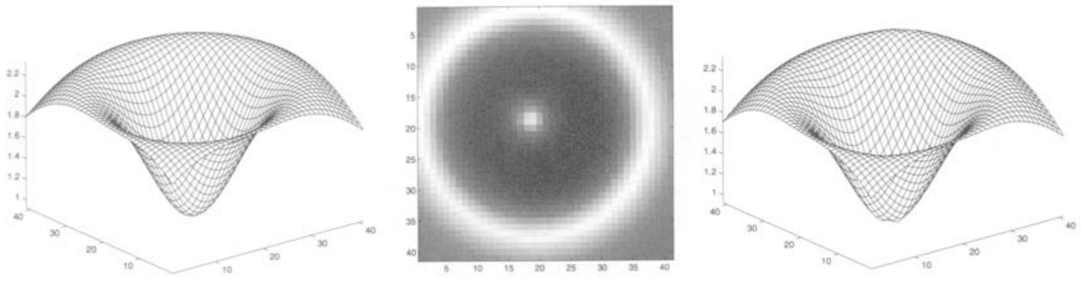
To demonstrate, we start with a given surface, first compute the brightness map I, and then reconstruct the surface by solving the above Eikonal equation. In Fig. 40, we show a paraboloid surface of the form $T = 3. - 3(x^2 + y^2)$. In Fig. 40a we show the original surface, in Fig. 40b we show the brightness map $I(x, y)$, and in Fig. 40c we show the reconstructed surface. This surface is 'built' by setting $T = 3$ at the point where the maximum is obtained, and then solving the Eikonal equation.

As a more complex example, we use a double Gaussian function of the form

$$T(x,y) = 3e^{-(x^2+y^2)} - 2e^{-20((x-.05)^2+(y-.05)^2)}. \tag{22.5}$$

Once again, we compute the brightness map and then reconstruct the surface; see Fig. 41.

We have barely touched the topic of shape-from-shading; in the case of multiple extrema and non-vertical light sources, more care must be taken, and we refer the interested reader to the above sources. Nonetheless, the fast marching level set algorithm is extremely effective for these problems.



(a) Original (b) Brightness (c) Reconstructed

Fig. 41. Shape-from-shading reconstruction of double Gaussian surface.

22.2. Photolithography development

One component process in the manufacturing of microchips is the stage of *lithography development*; see Section 17. In this process, the resist properties of a material have been altered due to exposure to a beam which has been partially blocked by a pattern mask. The material is then 'developed', which means the material with less resistivity is etched away. While the process is discussed in more detail in the next section, at this point we simply note that the problem reduces to that of following an initially plane interface propagating downwards in three dimensions, where the speed in the normal direction is given as a supplied rate function at each point. The speed $F = F(x, y, z)$ depends only on position; however, it may change extremely rapidly. The goal in lithography development is to track this evolving front. In order to develop realistic structures in three-dimensional development profiles, a grid of size $300 \times 300 \times 100$ is not unreasonable; hence a fast algorithm is required to perform the development step.

Start with a flat profile at height $z = 1$ in the unit cube centred at $(.5, .5, .5)$ and follow the evolution of the interface downwards with speed given by the model Gaussian rate function

$$F(x, y, z) = e^{-64(r^2)}(\cos^2(12z) + .01), \qquad (22.6)$$

where $r = \sqrt{(x - .5)^2 + (y - .5)^2)}$. This rate function F corresponds to the effect of standing waves which change the resist properties of the material, and causes sharp undulations and turns in the evolving profile. In Fig. 42, we show the profile etched out by such an initial state; the calculation is carried out until $T = 10$.

In Fig. 43, we give timings for a parameter study on a Sparc10 for the speed function $F = e^{-64(r^2)}(\cos^2(6z) + .01)$. We note that loading the file containing the model Gaussian rate function F is a significant proportion of the total compute time.

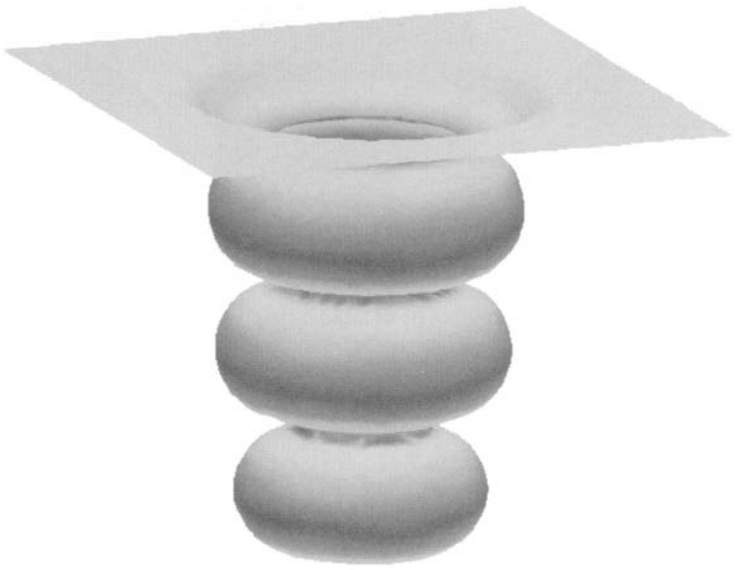


Fig. 42. Lithographic development on 50 × 50 × 50 grid.

Grid size	Time to load rate file	Time to propagate front	Total time
$50 \times 50 \times 50$	0.1 secs	0.7 secs	0.8 secs
$100 \times 100 \times 100$	1.2 secs	8.2 secs	9.4 secs
$150 \times 150 \times 150$	3.9 secs	37.8 secs	41.7 secs
$200 \times 200 \times 200$	9.0 secs	80.0 secs	89 secs

Fig. 43. Timings for development to T=10: Sparc 10.

Further details of the application of our fast marching level set method may be found in Sethian (1995*c*, 1996) and Sethian et al. (1996).

23. A final example: etching and deposition for the microfabrication of semiconductor devices

23.1. Background

A goal of numerical simulations in microfabrication of semiconductor devices is to model the process by which silicon devices are manufactured. Here, we briefly summarize the stages involved. First, a single crystal ingot of silicon is extracted from molten pure silicon. This silicon ingot is then sliced into several hundred thin wafers, each of which is then polished to a smooth finish. A thin crystalline layer is then oxidized, a light sensitive 'photoresist' is applied, and then the wafer is covered with a pattern mask that shields part of the photoresist. This pattern mask contains the layout of the circuit itself. Under exposure to a light or an electron beam, the exposed photoresist polymerizes and hardens, leaving an unexposed material which is then etched away in a dry etch process, revealing a bare silicon dioxide layer. Ionized impurity atoms such as boron, phosphorus and argon are then implanted into the pattern of the exposed silicon wafer, and silicon dioxide is deposited at reduced pressure in a plasma discharge from gas mixtures at a low temperature. Finally, thin films such as aluminium are deposited by processes such as plasma sputtering, and contacts to the electrical components and component interconnections are established. The result is a device that carries the desired electrical properties.

The above processes produce considerable change in the surface profile as it undergoes the stages of etching, deposition, and photolithography. This problem is known as the 'surface topography problem' in microfabrication, and is controlled by a large collection of physical effects, including the visibility of the etching/deposition source at each point of the evolving profile, surface diffusion along the front, non-convex sputter laws that produce faceting, shocks and rarefactions, material-dependent discontinuous etch rates, and masking profiles.

The underlying physical effects involved in etching, deposition and lithography are quite complex; excellent reviews are due to Neureuther and his group: see Helmsen (1994), Scheckler (1991), Scheckler, Toh, Hoffstetter and Neureuther (1991), Toh (1990) and Toh and Neureuther (1991), as well as Cale and Raupp (1990*a*, 1990*b*, 1990*c*), McVittie, Rey, Bariya et al. (1991) and Rey, Cheng, McVittie and Saraswat (1991). The effects may be summarized briefly as follows.

Deposition Particles are deposited on the surface, which causes build-up in the profile. The particles may either isotropically condense from the surroundings (known as chemical or 'wet' deposition), or be deposited from a source. In the latter case, we envision particles leaving the source and depositing on the surface; the main advantage of this approach is increased control over the directionality of surface deposition. The rate of deposition, and hence growth of the layer, may depend on source masking, visibility effects between the source and surface point, angle-dependent flux distribution of source particles, the angle of incidence of the particles relative to the surface normal direction, reflection of deposited particles, and surface diffusion effects.

Etching Particles remove material from the evolving profile boundary. The material may be isotropically removed, as in chemical or 'wet' etching, or chipped away through reactive ion etching, also known as 'ion milling'. Similar to deposition, the main advantage of reactive ion etching is enhanced directionality, which becomes increasingly important as device sizes decrease substantially and etching must proceed in vertical directions without affecting adjacent features. The total etch rate consists of an ion-assisted rate and a purely chemical etch rate due to etching by neutral radicals, which may still have a directional component. As in the above, the total etch rate due to wet and directional milling effects can depend on source masking, visibility effects between the source and surface point, angle-dependent flux distributions of source particles, the angle of incidence of the particles relative to the surface, reflection/re-emission of particles, and surface diffusion effects.

Lithography As discussed earlier, the underlying material is treated by an electromagnetic wave that alters the resist property of the material. The aerial image is found, which then determines the amount of crosslinking at each point in the material, which then produces the etch/resist rate at each point of the material. A profile is then etched into the material, where the speed of the profile in its normal direction at any point is given by the underlying etch rate. The key factors that determine the evolving shape are the etch/resist profile and masking effects.

In the final analysis, the above reduces to our familiar problem of tracking the boundary of a moving interface moving under a speed function F. Abstractly, we may write

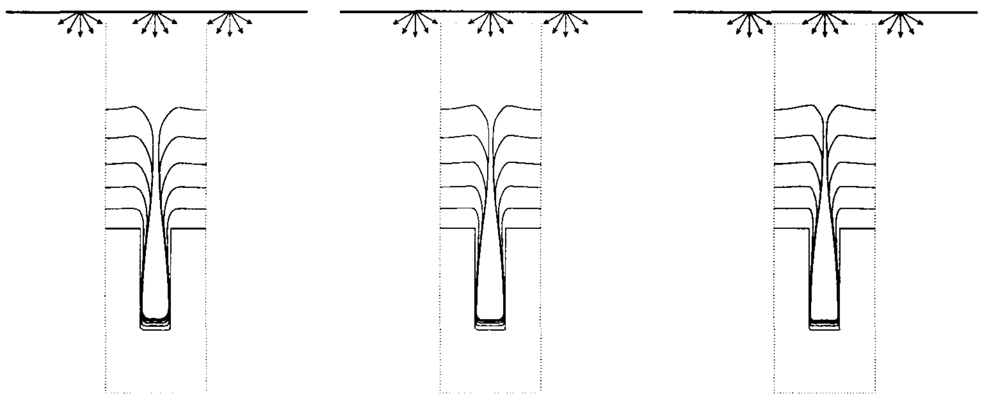
$$F = F_{\text{Deposition/Etching}} + F_{\text{Lithography}}. \tag{23.1}$$

Of course, all effects do not take place at once; however, the design of the numerical algorithm allows various combinations of terms to be 'turned on' during any time step of the surface advancement. For details and additional calculations of level set methods applied to microfabrication, see Adalsteinsson and Sethian (1995*b*, 1995*c*, 1996).

23.2. Results

Etching/deposition

We begin in Fig. 44 with a deposition source above a trench, where deposition material is emitted from a line source from the solid line above the trench. In this experiment, the deposition rate is the same in all directions. The effects of shadowing are considered. Fig. 44a shows results for 40 computational cells along the width of the computed region (between the two vertical dashed lines); Fig. 44b has 80 cells, and Fig. 44c has 160 cells. The time step for all three calculations is $\Delta t = .00625$. The calculations are performed with a narrow band tube width of 6 cells on either side of the front. There is little change between the calculation with 80 cells and the one with 160 cells, indicating that convergence has been achieved. As the walls pinch toward each other, the seen visible angle decreases and the speed diminishes.



(a) 40 cells (b) 80 cells (c) 160 cells

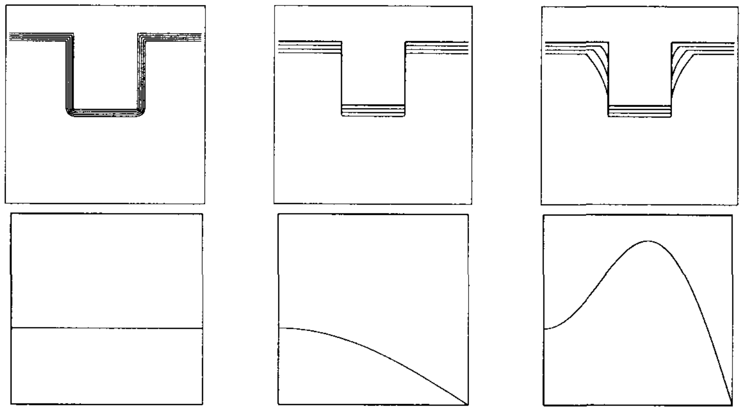
Fig. 44. Source deposition into trench.

Ion-milling: non-convex sputter laws

A more sophisticated set of examples arises in simulations (for example, of ion milling) in which the normal speed of the profile depends on the angle of incidence between the surface normal and the incoming beam. This yield function is often empirically fit from experiment, and has been observed to cause such effects as faceting at corners; see Leon, Tazawa, Saito, Yoshi and Scharfetter (1993) and Katardjiev, Carter and Nobes (1988). As shown in Adalsteinsson and Sethian (1995*b* and 1995*c*), such yield functions can often give rise to non-convex Hamiltonians, in which case alternative schemes must be used. To study this phenomenon, in Fig. 45 we consider several front motions and their effects on corners. We envision an etching beam coming down in the vertical direction. In the cases under study here, the angle θ refers to the angle between the surface normal and the positive vertical. For this set of calculations, in order to focus on the geometry of sputter effects on shocks/rarefaction fan development, visibility effects are ignored. The calculations are made using the schemes for non-convex Hamiltonians described earlier. Following our usual notation, let $F(\theta)$ be the speed of the front in direction normal to the surface.

In column A, the effects of purely isotropic motion are shown; thus the yield function is $F = 1$. Located above the yield graph are the motions of a downwards square wave. In column B, the effects of directional motion are shown; thus the yield function is $F = \cos(\theta)$. In this case, the horizontal components on the profile do not move, and vertical components move with unit speed. In column C, the effects of a yield function of the form $F = [1 + 4\sin^2(\theta)]\cos(\theta)$ are shown.

The results of these calculations are given in Fig. 45. The results show that the effects of angle dependent yield functions are pronounced. In column A



$F(\theta) = 1$ column A $\quad F = \cos(\theta)$ column B $\quad F = [1 + 4\sin^2(\theta)]\cos(\theta)$ column C

Fig. 45. Effect of different yield functions: non-convex scheme.

the isotropic rate produces smooth corners, correctly building the necessary rarefaction fans in outward corners and entropy satisfying shocks in inward corners, as discussed and analysed in Sethian (1982 and 1985). In column B, the directional rate causes the front to be essentially translated upwards, with minimal rounding of the corners. In column C, the yield function results in faceting of inward corners where shocks form and sharp corners in the construction of rarefaction fans.

Discontinuous etch rates

Next, we study the effects of etching through different materials. In this example, the etch rates are discontinuous, and hence sharp corners develop in the propagating profile. The results of these calculations are shown in Fig. 46. A top material masks a lower material, and the profile etches through the lower material first and underneath the upper material. The profile depends on the ratio of the etch rates. In Fig. 46a, the two materials have the same etch rate, and hence the front simply propagates in its normal direction with unit speed, regardless of which material it is passing through. In Fig. 46b, the bottom material etches four times faster than the top; in Fig. 46c, the ratio is ten to one. Finally, in Fig. 46d, the ratio is forty to one, in which case the top material almost acts like a mask.

Simultaneous etching and deposition

Next, a parameter study of simultaneous etching and deposition is taken from Adalsteinsson and Sethian (1996). The speed function is

$$F = (1 - \alpha)F_{\text{etch}} + \alpha F_{\text{Deposition}}, \qquad (23.2)$$

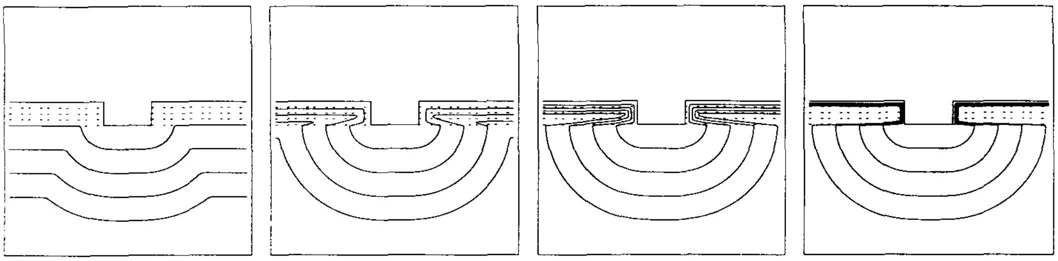


Fig. 46. Etch ratio = bottom material rate to top material rate.

where

$$\begin{aligned} F_{\text{etch}} &= (5.2249 \cos\theta - 5.5914 \cos^2\theta + 1.3665 \cos^4\theta), \\ F_{\text{Deposition}} &= \beta F_{\text{Isotropic}} + (1-\beta) F_{\text{Source}}. \end{aligned} \tag{23.3}$$

Visibility effects are considered in all terms except isotropic deposition. Fig. 47 shows the results of varying α and β between 0 and 1.

23.3. Three-dimensional simulations

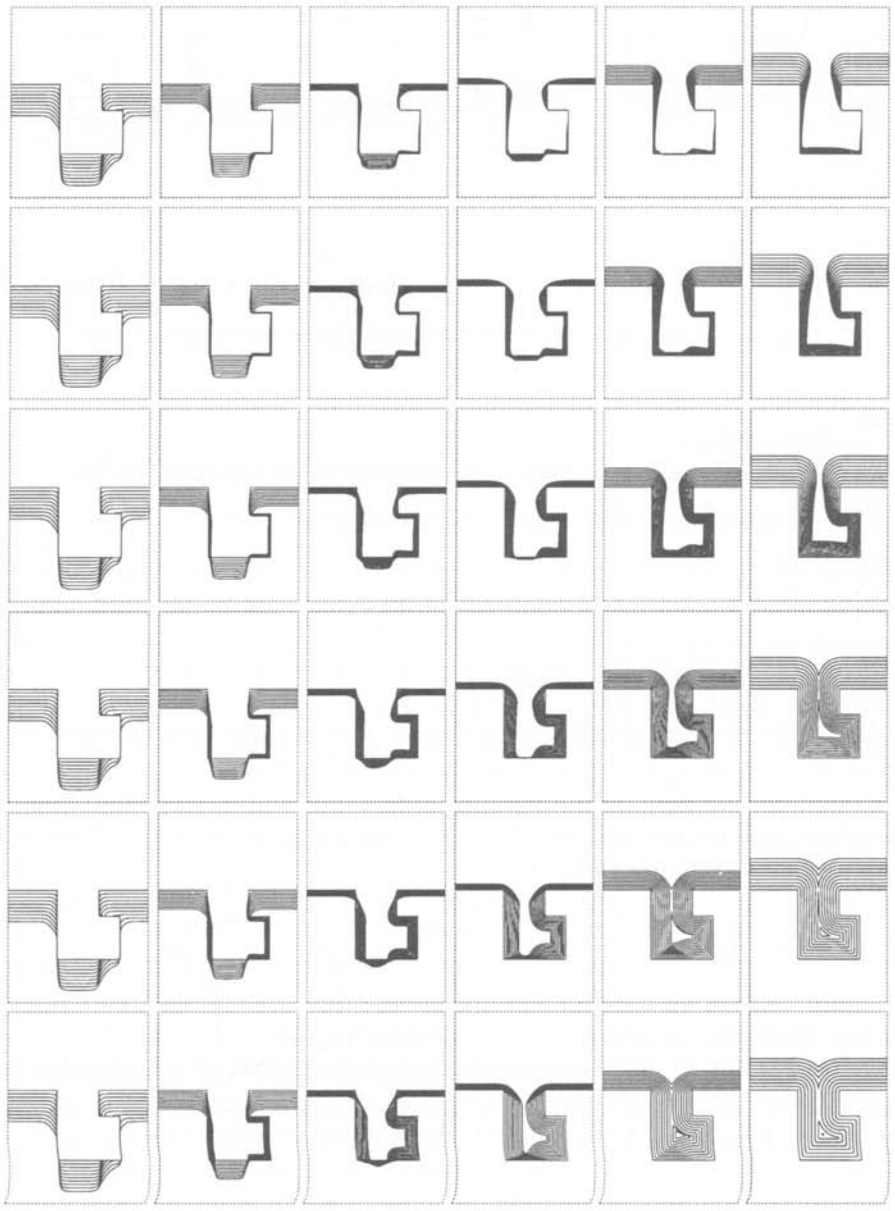
Finally, a three-dimensional example of a non-convex sputter yield law is applied to an indented saddle, which gives rise to faceting as shown in Fig. 48. Complete details of the above and a large variety of simulations of etching, deposition, and lithography development may be found in Adalsteinsson and Sethian (1995*b*, 1995*c*, 1996).

24. Other work

The range of level set techniques extends far beyond the work covered here. Here, we point the reader to some additional topics.

On the theoretical side, considerable analysis of level set methods has been performed in recent years; see, for example, Brakke (1978), Ecker and Huisman (1991), Evans and Spruck (1991, 1992*a*, 1992*b*, 1995), Chen et al. (1991), Giga and Goto (1992), Giga et al. (1992) and Ambrosio and Soner (1994). This work has concentrated on many aspects, including questions of existence and uniqueness, pathological cases, extensions of these ideas to fronts of co-dimension greater than one (such as evolving curves in three dimensions), coupling with diffusion equations, links between the level set technique and Brakke's groundbreaking original varifold approach.

On the theoretical/numerical analysis side, level set techniques exploit the considerable technology developed in the area of viscous solutions to Hamilton–Jacobi equations; see the work in Barles (1993), Crandall, Evans



$$F = (1-\alpha)F_{\text{etch}} + \alpha F_{\text{Deposition}}$$

$$F_{\text{etch}} = (5.2249\cos\theta - 5.5914\cos^2\theta + 1.3665\cos^4\theta)\cos\theta$$

$$F_{\text{Deposition}} = \beta F_{\text{Isotropic}} + (1-\beta)F_{\text{Source}}$$

α increases from left to right

β increases from bottom to top

Fig. 47. Simultaneous etching and deposition.

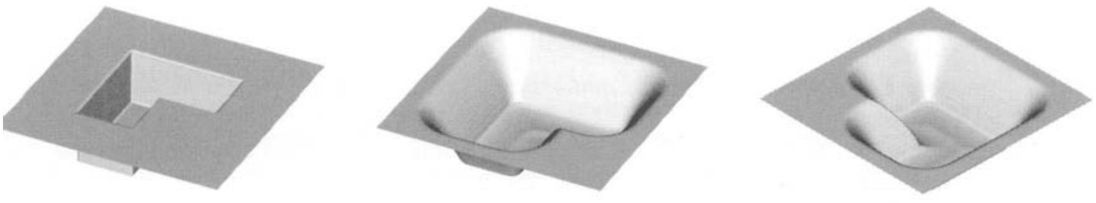

Initial shape: $T = 0$ $F = [1 + 4\sin^2(\theta)]\cos(\theta),\ T = 8$ Final rotated

Fig. 48. Downward saddle under sputter etch.

and Lions (1984), Crandall, Ishii and Lions (1992), Crandall and Lions (1983) and Lions (1982).

A wide range of applications relates to level set methods, including work on minimal arrival times by Falcone (1994), flame propagation work by Zhu and Ronney (1995), a wide collection of applications from computer vision by Kimmel (1995), gradient flows applied to geometric active contour models (Kichenassamy, Kumar, Olver, Tannenbaum and Yezzi 1995), work on affine invariant scale space (Sapiro and Tannenbaum 1993*a*), and some related work on the scalar wave equation (Fatemi, Engquist and Osher 1995). We also refer the reader to the collection of papers from the International Conference on Mean Curvature Flow (Buttazzo and Visitin 1994).

Acknowledgements All calculations were performed at the University of California at Berkeley and the Lawrence Berkeley Laboratory. The detailed applications of the level set schemes discussed in this work are joint with D. Adalsteinsson, D. Chopp, R. Malladi, B. Milne, C. Rhee, J. Strain, L. Talbot and J. Zhu.

REFERENCES

D. Adalsteinsson and J. A. Sethian (1995*a*), 'A fast level set method for propagating interfaces', *J. Comp. Phys.* **118**, 269–277.

D. Adalsteinsson and J. A. Sethian (1995*b*), 'A unified level set approach to etching, deposition and lithography I: Algorithms and two-dimensional simulations', *J. Comp. Phys.* **120**, 128–144.

D. Adalsteinsson and J. A. Sethian (1995*c*), 'A unified level set approach to etching, deposition and lithography II: Three-dimensional simulations', *J. Comp. Phys.* **122**, 348–366.

D. Adalsteinsson and J. A. Sethian (1996), 'A unified level set approach to etching, deposition and lithography III: Complex simulations and multiple effects', *J. Comp. Phys.* To be submitted.

L. Alvarez, P.-L. Lions and M. Morel (1992), 'Image selective smoothing and edge detection by nonlinear diffusion. II', *SIAM J. Num. Anal.* **29**, 845–866.

L. Ambrosio and H. M. Soner (1994), Level set approach to mean curvature flow in arbitrary codimension. Preprint.

S. Angenent (1992), Shrinking doughnuts, in *Proceedings of Nonlinear Diffusion Equations and Their Equilibrium States, 3* (N. G. Lloyd et al., eds), Birkhäuser, Boston.

M. Bardi and M. Falcone (1990), 'An approximation scheme for the minimum time function', *SIAM J. Control Optim.* **28**, 950–965.

G. Barles (1985), Remarks on a flame propagation model, report 464, INRIA.

G. Barles (1993), 'Discontinuous viscosity solutions of first order Hamilton–Jacobi equations: A guided visit', *Non-linear Analysis: Theory, Methods, and Applications* **20**, 1123–1134.

G. Barles and P. E. Souganidis (1991), 'Convergence of approximation schemes for fully non-linear second order equations', *Asymptotic Anal.* **4**, 271–283.

J. B. Bell, P. Colella and H. M. Glaz (1989), 'A second-order projection method for the incompressible Navier–Stokes equations', *J. Comp. Phys.* **85**, 257–283.

M. Berger and P. Colella (1989), 'Local adaptive mesh refinement for shock hydrodynamics', *J. Comp. Phys.* **82**, 62–84.

J. U. Brackbill, D. B. Kothe and C. Zemach (1992), 'A continuum method for modeling surface tension', *J. Comp. Phys.* **100**, 335–353.

K. A. Brakke (1978), *The Motion of a Surface by Its Mean Curvature*, Princeton University Press, Princeton University.

K. A. Brakke (1990), Surface evolver program, Technical Report GCC 17, University of Minnesota. Geometry Supercomputer Project.

L. Bronsard and B. Wetton (1995), 'A numerical method for tracking curve networks moving with curvature motion', *J. Comp. Phys.* **120**, 66–87.

G. Buttazzo and A. Visitin (1994), Motion by mean curvature and related topics, in *Proceedings of the International Conference at Trento, 1992*, Walter de Gruyter, New York.

J. E. Cahn and J. E. Hilliard (1958), 'Free energy of a non-uniform system. I. Interfacial Free Energy', *J. Chem. Phys.* **28**, 258–267.

T. S. Cale and G. B. Raupp (1990*a*), 'Free molecular transport and deposition in cylindrical features', *J. Vac. Sci. Tech., B* **8**, 649–655.

T. S. Cale and G. B. Raupp (1990*b*), 'Free molecular transport and deposition in long rectangular trenches', *J. Appl. Phys.* **68**, 3645–8652.

T. S. Cale and G. B. Raupp (1990*c*), 'A unified line-of-sight model of deposition in rectangular trenches', *J. Vac. Sci. Tech., B* **8**, 1242–1248.

V. Caselles, F. Catte, T. Coll and F. Dibos (n.d.), A geometric model for active contours in image processing, Technical Report 9210, CEREMADE, Université de Paris-Dauphiné, France. Internal report.

J. E. Castillo (1991), *Mathematical Aspects of Grid Generation*, Frontiers in Applied Mathematics 8, SIAM Publications.

Y. C. Chang, T. Y. Hou, B. Merriman and S. J. Osher (1994), 'A level set formulation of Eulerian interface capturing methods for incompressible fluid flows', *J. Comp. Phys.* Submitted.

Y. Chen, Y. Giga and S. Goto (1991), 'Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations', *J. Diff. Geom.* **33**, 749–786.

D. L. Chopp (1993), 'Computing minimal surfaces via level set curvature flow', *J. Comp. Phys.* **106**, 77–91.

D. L. Chopp (1994), 'Numerical computation of self-similar solutions for mean curvature flow', *J. Exper. Math.* **3**, 1–15.

D. L. Chopp and J. A. Sethian (1993), 'Flow under curvature: Singularity formation, minimal surfaces, and geodesics', *J. Exper. Math.* **2**, 235–255.

A. J. Chorin (1968), 'Numerical solution of the Navier–Stokes equations', *Math. Comp.* **22**, 745.

A. J. Chorin (1973), 'Numerical study of slightly viscous flow', *J. Fluid Mech.* **57**, 785–796.

A. J. Chorin (1980), 'Flame advection and propagation algorithms', *J. Comp. Phys.* **35**, 1–11.

L. D. Cohen (1991), 'On active contour models and balloons', *Computer Vision, Graphics, and Image Processing* **53**, 211–218.

P. Colella and E. G. Puckett (1994), *Modern Numerical Methods for Fluid Flow*, Lecture Notes, Department of Mechanical Engineering, University of California, Berkeley.

M. G. Crandall and P.-L. Lions (1983), 'Viscosity solutions of Hamilton–Jacobi equations', *Tran. AMS* **277**, 1–43.

M. G. Crandall, L. C. Evans and P.-L. Lions (1984), 'Some properties of viscosity solutions of Hamilton–Jacobi equations', *Tran. AMS* **282**, 487–502.

M. G. Crandall, H. Ishii and P.-L. Lions (1992), 'User's guide to viscosity solutions of second order partial differential equations', *Bull. AMS* **27**, 1–67.

K. Ecker and G. Huisman (1991), 'Interior estimates for hypersurfaces moving by mean curvature', *Inventiones Mathematica* **105**, 547–569.

B. Engquist and S. J. Osher (1980), 'Stable and entropy-satisfying approximations for transonic flow calculations', *Math. Comp.* **34**, 45.

L. C. Evans and J. Spruck (1991), 'Motion of level sets by mean curvature I', *J. Diff. Geom.* **33**, 635–681.

L. C. Evans and J. Spruck (1992*a*), 'Motion of level sets by mean curvature II', *Trans. AMS* **330**, 321–332.

L. C. Evans and J. Spruck (1992*b*), 'Motion of level sets by mean curvature III', *J. Geom. Anal.* **2**, 121–150.

L. C. Evans and J. Spruck (1995), 'Motion of level sets by mean curvature IV', *J. Geom. Anal.* **5**, 77–114.

L. C. Evans, H. M. Soner and P. E. Souganidis (1992), 'Phase transitions and generalized motion by mean curvature', *Comm. Pure Appl. Math.* **45**, 1097–1123.

M. Falcone (1994), The minimum time problem and its applications to front propagation, in *Motion by Mean Curvature and Related Topics. Proceedings of the International Conference at Trento, 1992*, Walter de Gruyter, New York.

M. Falcone, T. Giorgi and P. Loretti (1994), 'Level sets of viscosity solutions: Some applications to fronts and rendez-vous problems', *SIAM J. Appl. Math.* **54**, 1335–1354.

E. Fatemi, B. Engquist and S. J. Osher (1995), 'Numerical solution of the high frequency asymptotic wave equation for the scalar wave equation', *J. Comp. Phys.* **120**, 145–155.

M. Gage (1984), 'Curve shortening makes convex curves circular', *Inventiones Mathematica* **76**, 357.

M. Gage and R. Hamilton (1986), 'The heat-equation shrinking convex plane-curves', *J. Diff. Geom.* **23**, 69–96.

Y. Giga and S. Goto (1992), 'Motion of hypersurfaces and geometric equations', *J. Math. Soc. Japan* **44**, 99–111.

Y. Giga, S. Goto and H. Ishii (1992), 'Global existence of weak solutions for interface equations coupled with diffusion equations', *SIAM J. Math. Anal.* **23**, 821–835.

M. A. Grayson (1987), 'The heat equation shrinks embedded plane curves to round points', *J. Diff. Geom.* **26**, 285–314.

M. A. Grayson (1989), 'A short note on the evolution of a surface by its mean-curvature', *Duke Math. J.* **58**, 555–558.

L. Greengard and J. Strain (1990), 'A fast algorithm for evaluating heat potentials', *Comm. Pure Appl. Math.* **43**, 949–963.

A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy (1987), 'Uniformly high order accurate essentially non-oscillatory schemes. III', *J. Comp. Phys.* **71**, 231–303.

J. J. Helmsen (1994), A Comparison of Three-Dimensional Photolithography Development Methods, PhD thesis, University of California, Berkeley.

C. W. Hirt and B. D. Nicholls (1981), 'Volume of fluid (COF) method for dynamics of free boundaries', *J. Comp. Phys.* **39**, 201–225.

G. Huisken (1984), 'Flow by mean curvature of convex surfaces into spheres', *J. Diff. Geom.* **20**, 237–266.

T. Ilmanen (1992), 'Generalized flow of sets by mean curvature on a manifold', *Indiana University Mathematics Journal* **41**, 671–705.

T. Ilmanen (1994), *Elliptic Regularization and Partial Regularity for Motion by Mean Curvature*, Memoirs of the American Mathematical Society, 108.

M. Kass, A. Witkin and D. Terzopoulos (1988), 'Snakes: Active contour models', *Int. J. Comp. Vision* pp. 321–331.

I. V. Katardjiev, G. Carter and M. J. Nobes (1988), 'Precision modeling of the mask-substrate evolution during ion etching', *J. Vac. Sci. Tech. A* **6**, 2443–2450.

S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum and A. Yezzi (1995), *Gradient Flows and Geometric Active Contours, 1994*, ICCV.

R. Kimmel (1995), Curve Evolution on Surfaces, PhD thesis, Dept. of Electrical Engineering, Technion, Israel.

R. Kimmel and A. Bruckstein (1992), Shape from shading via level sets, Technical Report 9209, Technion, Israel. Center for Intelligent Systems.

R. Kimmel and A. Bruckstein (1993), 'Shape offsets via level sets', *Computer-Aided Design* **25**, 154–161.

P. Knupp and S. Steinberg (1993), The fundamentals of grid generation. Preprint.

P. D. Lax (1970), *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Reg. Conf. Series, Lectures in Applied Math, 11, SIAM, Philadelphia.

F. A. Leon, S. Tazawa, K. Saito, A. Yoshi and D. L. Scharfetter (1993), Numerical algorithms for precise calculation of surface movement in 3-D topography simulation, in *1993 International Workshop on VLSI Process and Device Modeling.*

R. J. LeVeque (1992), *Numerical Methods for Conservation Laws*, Birkhäuser, Basel.

P.-L. Lions (1982), *Generalized Solution of Hamilton–Jacobi Equations*, Pitman, London.

A. Majda and J. A. Sethian (1984), 'Derivation and numerical solution of the equations of low mach number combustion', *Combustion Science and Technology* **42**, 185–205.

R. Malladi, D. Adalsteinsson and J. A. Sethian (1995*a*), 'A fast level set algorithm for 3D shape recovery', *IEEE Transactions on PAMI.* Submitted.

R. Malladi and J. A. Sethian (1994), A unified approach for shape segmentation, representation, and recognition, Technical Report 36069, Lawrence Berkeley Laboratory, University of California, Berkeley.

R. Malladi and J. A. Sethian (1995), 'Image processing via level set curvature flow', *Proc. Natl. Acad. of Sci., USA* **92**, 7046–7050.

R. Malladi and J. A. Sethian (1996*a*), 'Image processing: Flows under min/max curvature and mean curvature', *Graphical Models and Image Processing.* In press.

R. Malladi and J. A. Sethian (1996*b*), 'A unified approach to noise removal, image enhancement, and shape recovery', *IEEE Image Processing.* In press.

R. Malladi, J. A. Sethian and B. C. Vemuri (1994), Evolutionary fronts for topology-independent shape modeling and recovery, in *Proceedings of Third European Conference on Computer Vision*, LNCS Vol. 800, Stockholm, pp. 3–13.

R. Malladi, J. A. Sethian and B. C. Vemuri (1995*b*), 'Shape modeling with front propagation: A level set approach', *IEEE Trans. on Pattern Analysis and Machine Intelligence* **17**, 158–175.

J. P. McVittie, J. C. Rey, A. J. Bariya et al. (1991), SPEEDIE: A profile simulator for etching and deposition, in *Proceedings of the SPIE – The International Society for Optical Engineering*, Vol. 1392, pp. 126–38.

B. Merriman, J. Bence and S. J. Osher (1994), 'Motion of multiple junctions: A level set approach', *J. Comp. Phys.* **112**, 334–363.

B. Milne and J. A. Sethian (1995), 'Adaptive mesh refinement for level set methods for propagating interfaces', *J. Comp. Phys.* Submitted.

W. Mulder, S. J. Osher and J. A. Sethian (1992), 'Computing interface motion in compressible gas dynamics', *J. Comp. Phys.* **100**, 209–228.

W. W. Mullins and R. F. Sekerka (1963), 'Morphological stability of a particle growing by diffusion or heat flow', *J. Appl. Phys.* **34**, 323–329.

W. Noh and P. Woodward (1976), A simple line interface calculation, in *Proceedings, Fifth International Conference on Fluid Dynamics* (A. I. van de Vooran and P. J. Zandberger, eds), Springer, Berlin.

S. Osher and L. I. Rudin (1990), 'Feature-oriented image enhancement using shock filters', *SIAM J. Num. Anal.* **27**, 919–940.

S. Osher and L. I. Rudin (1992), Rapid convergence of approximate solutions of shape-from-shading. To appear.

S. Osher and J. A. Sethian (1988), 'Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulation', *J. Comp. Phys.* **79**, 12–49.

S. Osher and C. Shu (1991), 'High-order nonoscillatory schemes for Hamilton–Jacobi equations', *J. Comp. Phys.* **28**, 907–922.

P. Perona and J. Malik (1990), 'Scale-space and edge detection using anisotropic diffusion', *IEEE Trans. Pattern Analysis and Machine Intelligence* **12**, 629–639.

M. Z. Pindera and L. Talbot (1986), Flame-induced vorticity: The effects of stretch, in *Twenty-First Symposium (Int'l) on Combustion*, The Combustion Institute, Pittsburgh, pp. 1357–1366.

W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling (1988), *Numerical Recipes*, Cambridge University Press.

E. G. Puckett (1991), A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction, in *Proceedings of the 4th International Symposium on Computational Computational Fluid Dynamics, Davis, California.*

J. C. Rey, L.-Y. Cheng, J. P. McVittie and K. C. Saraswat (1991), 'Monte Carlo low pressure deposition profile simulations', *J. Vac. Sci. Tech. A* **9**, 1083–7.

C. Rhee, L. Talbot and J. A. Sethian (1995), 'Dynamical study of a premixed V flame', *J. Fluid Mech.* **300**, 87–115.

E. Rouy and A. Tourin (1992), 'A viscosity solutions approach to shape-from-shading', *SIAM J. Num. Anal.* **29**, 867–884.

L. Rudin, S. Osher and E. Fatemi (1992), Nonlinear total variation-based noise removal algorithms, in *Modelisations Mathématiques pour le Traitement d'Images*, INRIA, pp. 149–179.

G. Sapiro and A. Tannenbaum (1993*a*), 'Affine invariant scale-space', *Int. J. Comp. Vision* **11**, 25–44.

G. Sapiro and A. Tannenbaum (1993*b*), Image smoothing based on affine invariant flow, in *Proc. of the Conference on Information Sciences and Systems*, Johns Hopkins University, Baltimore.

G. Sapiro and A. Tannenbaum (1994), Area and length preserving geometric invariant scale-spaces, in *Proc. of Third European Conference on Computer Vision*, Vol. 801 of *LNCS*, Stockholm, pp. 449–458.

E. W. Scheckler (1991), PhD thesis, EECS, University of California, Berkeley.

E. W. Scheckler, K. K. H. Toh, D. M. Hoffstetter and A. R. Neureuther (1991), 3D lithography, etching and deposition simulation, in *Symposium on VLSI Technology*, Oiso, Japan, pp. 97–98.

R. Sedgewick (1988), *Algorithms*, Addison-Wesley, New York.

J. A. Sethian (1982), An Analysis of Flame Propagation, PhD thesis, Department of Mathematics, University of California, Berkeley.

J. A. Sethian (1984), 'Turbulent combustion in open and closed vessels', *J. Comp. Phys.* **54**, 425–456.

J. A. Sethian (1985), 'Curvature and the evolution of fronts', *Comm. Math. Phys.* **101**, 487–499.

J. A. Sethian (1987), Numerical methods for propagating fronts, in *Variational Methods for Free Surface Interfaces* (P. Concus and R. Finn, eds), Springer, New York.

J. A. Sethian (1989), Parallel level set methods for propagating interfaces on the connection machine. Unpublished manuscript.

J. A. Sethian (1990), 'Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservation laws', *J. Diff. Geom.* **31**, 131–161.

J. A. Sethian (1991), A brief overview of vortex methods, in *Vortex Methods and Vortex Motion* (K. Gustafson and J. A. Sethian, eds), SIAM Publications, Philadelphia.

J. A. Sethian (1994), 'Curvature flow and entropy conditions applied to grid generation', *J. Comp. Phys.* **115**, 440–454.

J. A. Sethian (1995*a*), 'Algorithms for tracking interfaces in CFD and material science', *Annual Review of Computational Fluid Mechanics.*

J. A. Sethian (1995*b*), Level set techniques for tracking interfaces; fast algorithms, multiple regions, grid generation and shape/character recognition, in *Curvature flow and related topics*, Gakuto Int. Series, Volume 5, Tokyo.

J. A. Sethian (1995*c*), 'A marching level set method for monotonically advancing fronts', *Proc. Nat. Acad. Sci.* To appear.

J. A. Sethian (1996), *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Science*, Cambridge University Press. To appear.

J. A. Sethian, D. Adalsteinsson and R. Malladi (1996), 'Efficient fast marching level set methods'. In progress.

J. A. Sethian and J. D. Strain (1992), 'Crystal growth and dendritic solidification', *J. Comp. Phys.* **98**, 231–253.

P. E. Souganidis (1985), 'Approximation schemes for viscosity solutions of Hamilton–Jacobi equations', *J. Diff. Eqns.* **59**, 1–43.

J. Strain (1988), 'Linear stability of planar solidification fronts', *Physica D* **30**, 297–320.

J. Strain (1989), 'A boundary integral approach to unstable solidification', *J. Comp. Phys.* **85**, 342–389.

J. Strain (1990), 'Velocity effects in unstable solidification', *SIAM J. Appl. Math.* **50**, 1–15.

M. Sussman, P. Smereka and S. J. Osher (1994), 'A level set method for computing solutions to incompressible two-phase flow', *J. Comp. Phys.* **114**, 146–159.

J. E. Taylor, J. W. Cahn and C. A. Handwerker (1992), 'Geometric models of crystal growth', *Acta Metallurgica et Materialia* **40**, 1443–74.

D. Terzopoulos, A. Witkin and M. Kass (1988), 'Constraints on deformable models: Recovering 3d shape and nonrigid motion', *Artificial Intelligence* **36**, 91–123.

K. K. H. Toh (1990), PhD thesis, EECS, University of California, Berkeley.

K. K. H. Toh and A. R. Neureuther (1991), 'Three-dimensional simulation of optical lithography', *Proceedings SPIE, Optical/Laser Microlithography IV* **1463**, 356–367.

M. S. Young, D. Lee, R. Lee. and A. R. Neureuther (1993), 'Extension of the Hopkins theory of partially coherent imaging to include thin-film interference effects', *Proceedings SPIE, Optical/Laser Microlithography VI* **1927**, 452–463.

J. Zhu and P. D. Ronney (1995), 'Simulation of front propagation at large non-dimensional flow disturbance intensities', *Comb. Sci. Tech.* To appear.

J. Zhu and J. A. Sethian (1992), 'Projection methods coupled to level set interface techniques', *J. Comp. Phys.* **102**, 128–138.