# A Fast Marching Level Set Method for Monotonically Advancing Fronts *

**J.A. Sethian**
**Dept. of Mathematics**
**University of California, Berkeley**

Oct. 20, 1995
Submitted for Publication:
Proceedings of the National Academy of Sciences

**Abstract**

We present a fast marching level set method for monotonically advancing fronts, which leads to an extremely fast scheme for solving the Eikonal equation. Level set methods are numerical techniques for computing the position of propagating fronts. They rely on an initial value partial differential equation for a propagating level set function, and use 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. In this paper, we describe a particular case of such methods for interfaces whose speed depends only on local position. The technique works by coupling work on entropy conditions for interface motion, the theory of viscosity solutions for Hamilton-Jacobi equations and fast adaptive narrow band level set methods. The technique is applicable to a variety of problems, including shape-from-shading problems, lithographic development calculations in microchip manufacturing, and arrival time problems in control theory.

## 1 Introduction

In this paper, we describe and test a numerical algorithm for tracking the evolution of interfaces. The technique applies in the case of a front propagating normal to itself with a speed $F$ that

*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.

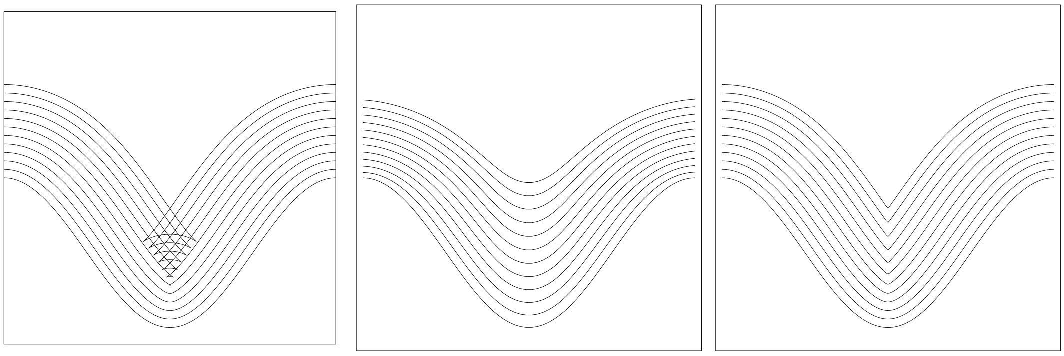
depends only on position and is always either positive or negative. The applications of such a technique include some global illumination problems and problems from control theory, as well as surface advancement in lithographic development and isotropic etching and deposition in the manufacturing of microelectronic structures. This scheme was first described in [25]; in this paper, we present the details of this scheme and show results and timings.

## 2 Background

Consider a boundary, either a curve in two dimensions or a surface in three dimensions, separating one region from another, and imagine that this curve/surface moves in its normal direction with a known speed function $F$. The 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, and shall ignore tangential motion.

As shown in [27, 28, 31], a propagating interface can develop corners and discontinuities as it evolves, which require the introduction of a weak solution in order to proceed. The correct weak solution comes from enforcing an entropy condition for the propagating interface, similar to the one in gas dynamics. Furthermore, this entropy-satisfying weak solution is the one obtained by considering the limit of smooth solutions for the problem in which curvature plays a regularizing role.

As an example, consider the initial cosine curve propagating with speed $F = 1$ shown in Figure 1. As the front moves, a corner forms in the propagating front which corresponds to a shock in the slope, and a weak solution must be developed beyond this point. If the motion of each individual point is continued, the result is the swallowtail solution shown in Fig. 1a, which is multiple-valued and does not correspond to a clear interface separating two regions. Instead, an appropriate weak solution is obtained by considering the associated smooth flow obtained by adding curvature $\kappa$ to the speed law, that is, letting $F = 1 - \epsilon\kappa$, see Fig. 1b. The limit of these smooth solutions as $\epsilon$ goes to zero produces the weak solution shown in Fig. 1c; this is the same solution obtained by enforcing an entropy condition, similar to the one for a scalar hyperbolic conservation law, which selects the envelope obtained by Huygens principle as the



$Swallowtail(F = 1.0)$ $F = 1. - 0.25\kappa$ $Entropy\ Solution(F = 1.0)$

Figure 1: Cosine Curve Propagating with Unit Speed

correct solution, see [27]. This weak solution corresponds to a decrease in total variation of the propagating front and is irreversible [28]. For details, see [28].

As a numerical technique, this suggests using the technology from hyperbolic conservation laws to solve the equations of motion, as described in [32]. This leads to the *level set* formulation introduced in [20], which we now describe.

## 2.1 Level Set Methods

Given an initial position for an interface $\Gamma$, where $\Gamma$ is a closed curve in $R^2$, and a speed function $F$ which gives the speed of $\Gamma$ in its normal direction, the level set method takes the perspective of viewing $\Gamma$ as the zero level set of a function $\phi(x, t = 0)$ from $R^2$ to $R$. That is, let $\phi(x, t = 0) = \pm d$, where $d$ is the distance from $x$ to $\Gamma$, and the plus (minus) sign is chosen if the point $x$ is outside (inside) the initial hypersurface $\Gamma$. Then, by the chain rule, an evolution equation for the interface may be produced [20, 31], namely

$$\phi_t + F|\nabla\phi| = 0, \tag{1}$$

$$\phi(x, t = 0) = given. \tag{2}$$

This is an initial value partial differential equation in one higher dimension than the original problem. In Figure 2 (taken from [30]), we show the outward propagation of an initial curve and the accompanying motion of the level set function $\phi$.

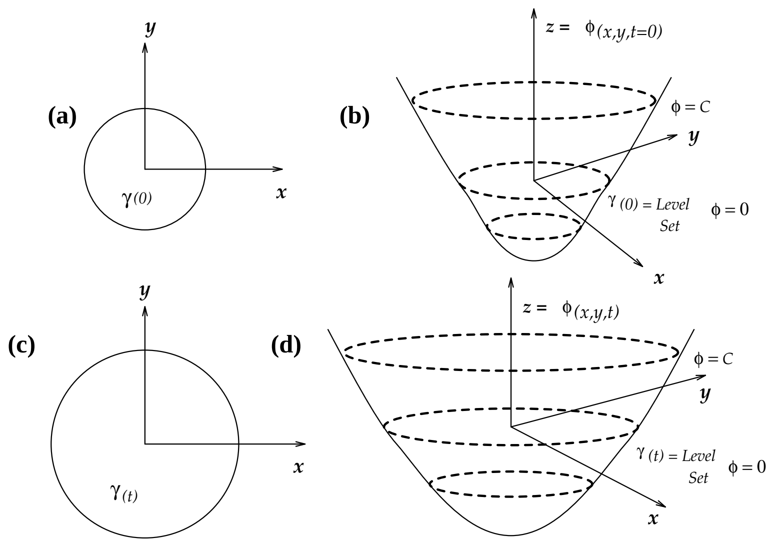



Figure 2: Propagating Circle

There are several advantages to this level set perspective:

1. Although $\phi(x, t)$ remains a function, the level surface $\phi = 0$ corresponding to the propagating hypersurface may change topology, as well as form sharp corners as $\phi$ evolves (see [20]).

2. Second, a discrete grid can be used together with finite differences to devise a numerical scheme to approximate the solution. Care must taken to adequately account for the spatial derivatives in the gradient.

3. Third, intrinsic geometric properties of the front are easily determined from the level set function $\phi$. The normal vector is given by $\vec{n} = \frac{\nabla \phi}{|\nabla \phi|}$ and the curvature of each level set is $\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$.

4. Finally, the formulation is unchanged for propagating interfaces in three dimensions.

Since its introduction in [20], the above level set approach has been used in a wide collection of problems involving moving interfaces. Some of these applications include the generation of minimal surfaces [8], singularities and geodesics in moving curves and surfaces in [9], flame propagation [22, 36], fluid interfaces [6, 18], shape reconstruction [17, 16], as well as etching, deposition and lithography calculations in [2, 3]. Extensions of the basic technique include fast methods in [1], level set techniques for multiple fluid interfaces and triple point junctions in [26], and grid generation in [30]. The fundamental Eulerian perspective presented by this approach has since been adopted in many theoretical analyses of mean curvature flow, in particular, see [12, 7].

## 2.2 Numerical Approximation

As mentioned above, a careful approximation to the gradient in the level set equation (Eqn. 1) is required in order to produce the correct weak solution. One of the simplest such schemes is given in [20], namely

$$\phi_{ij}^{n+1} = \phi_{ij}^{n} - \Delta t \,(\max(D_{ij}^{-x}\phi, 0)^2 + \min(D_{ij}^{+x}\phi, 0)^2 \max(D_{ij}^{-y}\phi, 0)^2 + \min(D_{ij}^{+y}\phi, 0)^2)^{1/2}, \quad (3)$$

where here we have taken the speed $F = 1$ and employed difference operator notation that, for example, $D_{ij}^{+x}\phi = (\phi_{i+1,j} - \phi_{i,j})/(\Delta x)$. The crucial point in this (any such appropriate) numerical scheme is the correct direction of the upwinding and treatment of sonic points.

## 2.3 Narrow Band Methods

The above technique relys on computing the evolution of *all* the level sets, not simply the zero level set corresponding to the front itself. As such, it is a computationally expensive technique, since an extra dimension has been added to the problem.

As an alternative, an efficient modification is to perform work only in a neighborhood of the zero level set; this is known as the *narrow band approach.* In this case, the operation count in three dimensions for $N^3$ grid points drops to $O(kN^2)$, where $k$ is the number of cells in the width of the narrow band, providing a significant cost reduction. This "narrow band method" method was introduced in [8], used in recovering shapes from images in [17], and analyzed extensively in [1].

The basic idea is to tag grid points as either "alive", "land mines", or "far away", depending on whether they are inside the band, near its boundary, or outside the band, respectively, (see Figure 3). Thus, work is performed only onthe alive points, and the band is reconstructed once land mine points are reached. An extreme one-cell version of this leads to the fast marching level set method presented below.

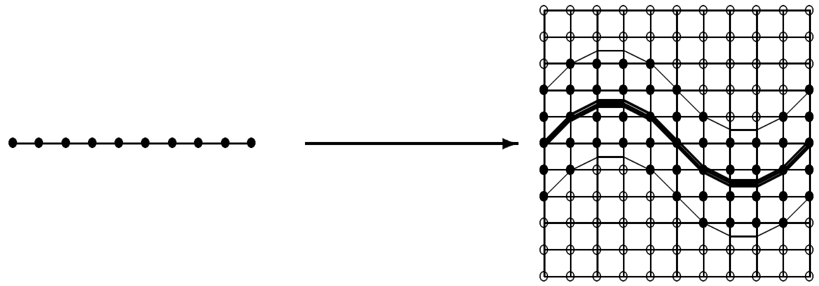


Figure 3: Pointer Array Tags Alive, Narrow Band, and Far Away Points

# 3 A Fast Marching Level Set Method

We now discuss in detail the fast marching level set method introduced in [25]. Consider the special case of a front moving with speed $F = F(x, y, z)$, $F > 0$ (the case where $F$ is everywhere negative is also allowed). We then have a monotonically advancing front whose level set equation is of the form

$$\phi_t + F(x, y, z)|\nabla\phi| = 0 \tag{4}$$

$$\phi(x, t = 0) = \Gamma. \tag{5}$$

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

$$|\nabla T|F = 1. \tag{6}$$

Eqn. 6 simply says that the gradient of arrival time surface is inversely proportional to the speed of the front.

This is a form of the well-known Eikonal equation, and the recasting of the problem into a stationary one is common in a variety of applications. The notion of viscosity solutions is intimately connected to this equation; a central idea, discussed in detail in [15], is that the use of monotone, consistent schemes will lead to schemes which select the correct viscous limit of the partial differential equation, as was done in the level set scheme in [20]. We refer the interested reader to large literature on this subject, including relevant theory in [4, 10, 15, 35], and numerical algorithms in [5, 13, 14, 23, 19]. Roughly speaking, two possible ways to view these solution techniques are either iteration towards the solution or direct construction of the stationary solution surface $T(x,y)$. We now discuss a technique that relys on the level set methodology and narrow band work described above.

### 3.1 Approximation Scheme

For this 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. Using our approximation to the gradient, we are then looking for a solution in the unit box to the equation

$$\left[\max(D_{ij}^{-x}T,0)^2 + \min(D_{ij}^{+x}T,0)^2 + \max(D_{ij}^{-y}T,0)^2 + \min(D_{ij}^{+y}T,0)^2)\right] = \frac{1}{F^2} \tag{7}$$

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

Since Eqn. 7 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 in [23]; there, a different approximation to the gradient was chosen which is less diffusive motivated by the scheme given in [19], namely

$$\left[\max(\max(D_{ij}^{-x}T,0), -\min(D_{ij}^{+x}T,0))^2 + \max(\max(D_{ij}^{-y}T,0), -\min(D_{ij}^{+y}T,0))^2\right] = 1/F_{ij}^2 \tag{8}$$

For details of this approach, see [23]. We shall use this approximation to the gradient in our fast marching level set method.

### 3.2 A Fast Marching Level Set Method

The key to constructing a fast marching algorithm is the observation that the upwind difference structure of Eqn. 8 means that information propagates "one way", that is, from smaller values of $T$ to larger values. Hence, our algorithm rests on "solving" Eqn. 8 by building the solution

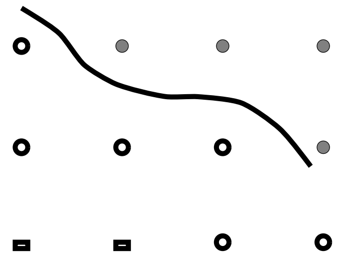


Figure 4: Narrow Band Approach to Marching Level Set Method

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 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 Figure 4, taken from [25]. We imagine that we want to propagate a front upwards 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 $j = 1$ correspond to the $y$ axis, and we assume that $F_{ij} > 0$.

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) Tag as neighbors any points $(i_{\min}-1, j_{\min})$, $(i_{\min}+1, j_{\min})$, $(i_{\min}, j_{\min}-1)$, $(i_{\min}, j_{\min}+1)$ that are either in $NarrowBand$ or $FarAway$. If the neighbor is in $FarAway$,

remove it from that list and add it to the set $NarrowBand$.

(d) Recompute the values of $T$ at all neighbors according to Eqn. 8, 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 neighboring points (that have not been previously accepted) cannot yield a value smaller than any of that 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 neighbors. Another way to look at this is that each minimum trial value begins an application of Huygen's principle, and the expanding wave front touches and updates all others.

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 the calculation is performed on an $N$ by $N$ grid.

# 4 Proof that the algorithm constructs a viable solution

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

$$\left[\max(\max(D_{ij}^{-x}T,0),-\min(D_{ij}^{+x}T,0))^2+\max(\max(D_{ij}^{-y}T,0),-\min(D_{ij}^{+y}T,0))^2\right]=f_{ij}^2 \quad (9)$$

| | | |
|---|---|---|
| | B | |
| A | ? | C |
| | D | |

Figure 5: Matrix of neighboring values

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 neighbors obtain new values less than the accepted value. If this is true, then we will always be marching ahead in time, and the 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 Figure 5. Our argument will follow the computation of the new value of $T$ in the center grid point to replace the value of ?, based on the neighboring 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 center grid point (called $T_{recomputed-from-A}$), 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 neighbors $B$, $C$, or $D$, are "alive" (2) one of these neighbors is "alive" (3) two of the neighors are "alive" and (4) all three of these neighbors are "alive". [1]

### $A$, $B$, $C$ and $D$ are "trial", $A$ is the smallest

In this case, all of the neighbors around the center 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 center grid point. We now show that the recomputed value $A \leq T_{recomputed-from-A} \leq A + f$.

1. Suppose $A + f \leq \min(B, D)$. Then $T_{recomputed-from-A} = (A + f)$ is a solution to the problem, since only the difference operator to the left grid point is non-zero. footnoteWe are absorbing the grid size $\Delta x$ into the inverse speed function $f$.

[1] 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.

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

$$(T_{recomputed-from-A} - A)^2 + (T_{recomputed-from-A} - B)^2 = f^2 \tag{10}$$

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 $A \leq T_{recomputed-from-A} \leq A+f$, and therefore $T_{recomputed-from-A}$ cannot be less than the just converted value $A$.

This case will act a 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 $T_{recomputed_A}$, 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 $B \leq T_{recomputed-from-B} \leq B + f$; furthermore, since the value at the center was *not* chosen as the smallest trial value, we must have that $A \leq B + f$. By the above case, we then have that $B \leq A \leq T_{recomputed-from-A} \leq B + f$, and hence the recomputed value cannot be less than the just converted value of $A$.

**$C$ 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.

## 4.1 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 with back pointers, see [21, 24]. 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 neighbors of that point, we use the pointers from the grid array to the heapsort structure. The values of the neighbors 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 $O(logN)$ 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 [34].

## 4.2 Arbitrary Initial Fronts

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 $\infty$. Then, construct the signed distance function in a one-grid cell wide band around the initial hypersurface $\Gamma$. 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 [30]. 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.

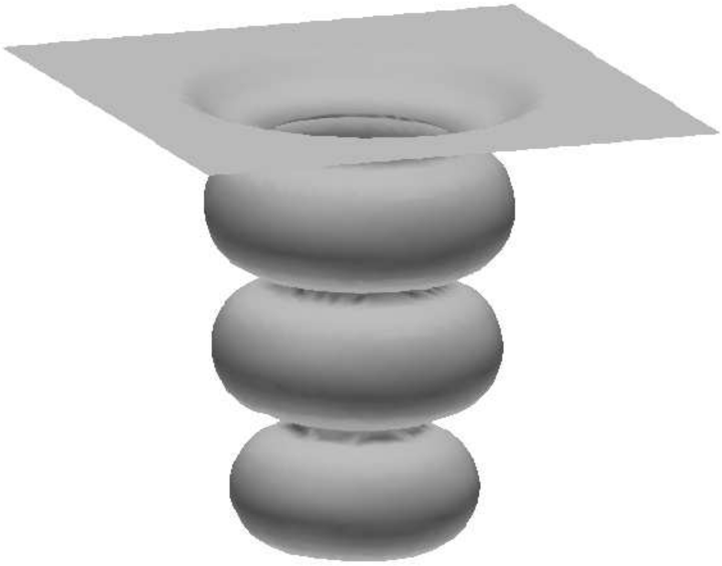


Figure 6: Lithographic Development on 50x50x50 Grid

# 5 Results

## 5.1 Simple Initial Front

As a first example, we use the above algorithm to compute a lithographic development profile for an evolving front. We start with a flat profile at height $z = 1$ in the unit cube centered at $(.5, .5, .5)$ and follow the evolution of the interface downwards with speed given by the model Gaussian rate function

$$F(x, y, z) = e^{-64(r^2)}(\cos^2(12z) + .01) \tag{11}$$

where $r = \sqrt{(x - .5)^2 + (y - .5)^2)}$. This rate function $F$ corresponds to effect of standing waves which change the resist properties of the material, and causes sharp undulations and turns in the evolving profile. In Figure 6, we show the profile etched out by such an initial state; the calculation is carried out until $T = 10$.

In Figure 7, 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.

| Grid Size | Time to Load Rate File | Time to Propagate Front | Total Time |
|---|---|---|---|
| 50x50x50 | 0.1 secs | 0.1 secs | 0.2 secs |
| 100x100x100 | 1.2 secs | 8.2 secs | 9.4 secs |
| 150x150x150 | 3.9 secs | 37.8 secs | 41.7 secs |
| 200x200x200 | 9.0 secs | 80.0 secs | 89 secs |

Figure 7: Timings for Development to T=10: Sparc 10

## 5.2 Deposition Problem

Next, we consider the case of simple isotropic deposition above a trench, with corresponding speed function $F = 1$. In Figure 8, we show a two-dimensional trench being filled in with a deposition layer; we note the sharp corner that develops when the entropy condition is invoked.

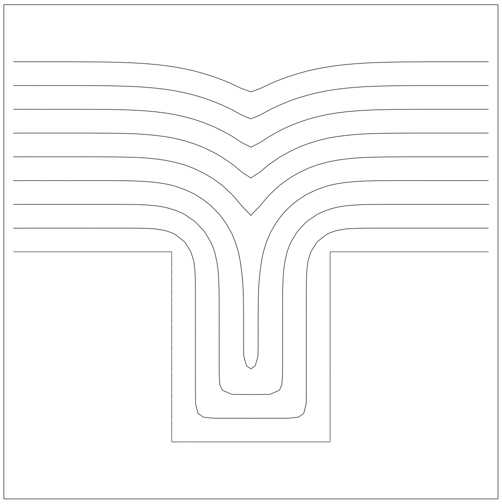


Figure 8: Isotropic Deposition Above Trench

There are a large number of applications of this fast marching level set method, including problems in control theory, etching/deposition/lithography, and global illumination. At the same time, the above technique can be extended, with modification, to general convex speeds laws, and perhaps, non-convex speed functions. We shall report on these issues elsewhere [33]. Furth

**Acknowledgements**: All calculations were performed at the University of California at Berkeley and the Lawrence Berkeley Laboratory. The author would like to thank D. Adalsteinsson and R. Malladi for valuable discussions and help with the relevant sorting algorithms.

## References


[1] Adalsteinsson, D., and Sethian, J.A., *A Fast Level Set Method for Propagating Interfaces*, J. Comp. Phys., Vol. 118(2), pp. 269–277, May 1995.

[2] Adalsteinsson, D., and Sethian, J.A., *A Unified Level Set Approach to Etching, Deposition and Lithography I: Algorithms and Two-dimensional Simulations*, J. Comp. Phys. 120, 1, pp. 128-144, 1995.

[3] Adalsteinsson, D., and Sethian, J.A., *A Unified Level Set Approach to Etching, Deposition and lithography II: Three-dimensional Simulations*, to appear, J. Comp. Phys., 1995.

[4] Barles, G., and Souganidis, P.E., *Convergence of Approximation Schemes for Fully Nonlinear Second Order Equations*, Asymptotic Anal., 4, pp.271-283, 1991.

[5] Bardi, M., and Falcone, M., *An Approximation Scheme for the Minimum Time Function*, SIAM J. Control Optim, 28, pp. 950-965, (1990),

[6] Chang, Y.C., Hou, T.Y., Merriman, B., and Osher, S.J., *A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows*, submitted for publication, Jour. Comp. Phys., 1994.

[7] Chen, Y., Giga, Y., and Goto, S., *Uniqueness and Existence of Viscosity Solutions of Generalized Mean Curvature Flow Equations*, J. Diff. Geom, Vol. 33, 749, 1991.

[8] Chopp, D.L., *Computing Minimal Surfaces via Level Set Curvature Flow*, Jour. of Comp. Phys., Vol. 106, pp. 77–91, 1993.

[9] Chopp, D.L., and Sethian, J.A,. *Flow Under Curvature: Singularity Formation, Minimal Surfaces, and Geodesics*, Jour. Exper. Math., Vol. 2(4), pp. 235–255, 1993.

[10] Crandall, M.G., Evans, L.C., and Lions, P-L., *Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations*, Tran. AMS, 282, pp. 487-502, 1984.

[11] Engquist, B., and Osher, S.J., *Stable and Entropy-Satisfying Approximations for Transonic Flow Calculations*, Math. Comp., Vol. 34, 45, 1980.

[12] Evans, L.C., and Spruck, J., *Motion of Level sets by Mean Curvature I*, J. Diff. Geom, Vol. 33, 635, 1991.

[13] Falcone, M., Giorgi, T., and Loretti, P., *Level Sets of Viscosity Solutions: Some Applications to Fronts and Rendez-Vous Problems*, SIAM J. Appl. Math., 54, 5, pp. 1335-1354, 1994.

[14] Kimmel, R., and Bruckstein, A., *Shape from Shading via Level Sets*, Center for Intelligent Systems Report No.9209, Technion- Israel Institute of Technology, June 1992,

[15] Lions, P.L. *Generalized Solution of Hamilton-Jacobi Equations*, Pittman, London,1982.

[16] Malladi, R., and Sethian, J.A., *Image Processing via Level Set Curvature Flow*, Proc. Natl. Acad. of Sci., USA, Vol. 92(15), pp. 7046–7050, July 1995.

[17] Malladi, R., Sethian, J.A., and Vemuri, B.C., *Evolutionary Fronts for Topology-independent shape Modeling and Recovery*, in Proceedings of Third European Conference on Computer Vision, LNCS Vol. 800, pp. 3–13, Stockholm, Sweden, May 1994.

[18] Mulder, W., Osher, S.J., Sethian, J.A., *Computing Interface Motion in Compressible Gas Dynamics*, Jour. Comp. Phys., Vol. 100, pp. 209-228, 1992.

[19] Osher, S., and Rudin, L.I., *Rapid Convergence of Approximate Solutions of Shape-From-Shading*, to appear, 1992.

[20] Osher, S., and Sethian, J.A., *Fronts Propagating with Curvature Dependent speed: Algorithms Based on Hamilton-Jacobi Formulation*, Journal of Computational Physics, Vol. 79, pp. 12-49, 1988.

[21] Press, W. H., *Numerical Recipes*, Cambridge University Press, New York, 1988

[22] Rhee, C., Talbot, L., and Sethian, J.A., *Dynamical Study of a Premixed V flame*, Jour. Fluid Mech., 300, pp.87-115, 1995.

[23] Rouy, E. and Tourin, A., *A Viscosity Solutions Approach to Shape-From-Shading*, SIAM J. Num. Anal, 29, 3. pp. 867-884, 1992.

[24] Sedgewick, R., Algorithms, Addison-Wesley, 1988.

[25] Sethian, J.A., *A Review of the Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces*, to appear, in press, Acta Numerica, 1995.

[26] Sethian, J.A., *Algorithms for Tracking Interfaces in CFD and Material Science*, Annual Review of Computational Fluid Mechanics, 1995

[27] Sethian, J.A., *An Analysis of Flame Propagation*, Ph.D. Dissertation, Mathematics, University of California, Berkeley, 1982.

[28] Sethian, J.A., *Curvature and the Evolution of Fronts*, Comm. in Mathematical Physics, Vol. 101, pp. 487–499, 1985.

[29] Sethian, J.A., *A Brief Overview of Vortex Methods*, in Vortex Methods and Vortex Motion, Eds. K. Gustafson and J.A. Sethian, SIAM Publications, Philadelphia, 1991.

[30] Sethian, J.A., *Curvature Flow and Entropy Conditions Applied to Grid Generation*, J. Comp. Phys. Vol. 115, pp.440-454, 1994.

[31] Sethian, J.A., *Numerical Algorithms for Propagating Interfaces: Hamilton-Jacobi Equations and Conservation Laws*, Journal of Differential Geometry, Vol. 31, pp. 131–161, 1990.

[32] Sethian, J.A., *Numerical Methods for Propagating Fronts*, in Variational Methods for Free Surface Interfaces, edited by P. Concus and R. Finn, (Springer-Verlag, New Work, 1987).

[33] Sethian, J.A., *Extensions to a Fast Marching Level Set Method for Advancing Fronts*, in progress, 1995.

[34] Sethian, J.A., Adalsteinsson, D., and Malladi, R., *An Efficient Level Set Technique for Propagating Interfaces*, in progress, 1995.

[35] Souganidis, P.E., *Approximation Schemes for Viscosity Solutions of Hamilton-Jacobi Equations*, J. Diff. Eqns., 59, pp.1-43, 1985.

[36] Zhu, J., and Sethian, J.A., *Projection Methods Coupled to Level Set Interface Techniques*, J. Comp. Phys., vol. 102, pp.128-138, 1992.