A Second-Order Accurate Method for Solving the Signed Distance Function Equation

Peter Schwartz
Phillip Colella

September 16, 2009

Abstract

We present a numerical method for computing the signed distance to a piecewise-smooth surface defined as the zero set of a function. It is based on a marching method by Kim [Kim01] and a hybrid discretization of first- and second-order discretizations of the signed distance function equation. If the solution is smooth at a point and at all of the points in the domain of dependence of that point, the solution is second-order accurate; otherwise, the method is first-order accurate, and computes the correct entropy solution in the presence of kinks in the initial surface.

1 Introduction

Let Γ be a continuous, piecewise smooth $\mathbf{D}-1$ – dimensional manifold in $\mathbb{R}^{\mathbf{D}}$ defined implicitly as the zero set of a function, i.e. there is a continuous piecewise smooth ϕ defined on some ϵ – neighborhood of Γ such that

$$\Gamma = \{\boldsymbol{x} : \phi(\boldsymbol{x}) = 0\}. \tag{1}$$

We also assume that $\nabla\phi$ is bounded and piecewise smooth on Γ, and that that there is a constant $c > 0$ such that $|\nabla\phi(\boldsymbol{x}_0)| \geq c$ at all points, $\boldsymbol{x}_0 \in \Gamma$ where $\nabla\phi$ is defined. At such points, $\hat{\boldsymbol{n}}$, the unit normal to Γ, is given by

$$\hat{\boldsymbol{n}} = \frac{\nabla\phi}{|\nabla\phi|}$$

Given such a surface Γ, we can define the signed distance function ψ

$$\psi(\boldsymbol{x}) = s \min_{\boldsymbol{x}' \in \Gamma} |\boldsymbol{x} - \boldsymbol{x}'| = \mathrm{sdist}(\boldsymbol{x}, \Gamma) \tag{2}$$

where s is defined to be the positive on one side of Γ and negative on the other. If $\boldsymbol{x}_0 \in \Gamma$, is a point at which the minimum in the right-hand side of (2) is achieved, and Γ is smooth at that point, then, $s = sign((\boldsymbol{x} - \boldsymbol{x}_0) \cdot \nabla\phi(\boldsymbol{x}_0))$. If Γ is not smooth at that point, then s is the single value taken on by $sign((\boldsymbol{x} - \boldsymbol{x}') \cdot \nabla\phi(\boldsymbol{x}'))$ at all points sufficiently close to $\boldsymbol{x}_0$ such that $\nabla\phi(\boldsymbol{x}')$ is defined. In any case, $s = \pm 1$ on $\mathbb{R}^{\mathbf{D}} - \Gamma$ and changes only at Γ.

If $\psi(\boldsymbol{x})$ is smooth, then ψ satisfies the signed distance function equation.

$$|\nabla\psi(\boldsymbol{x})| = 1 \tag{3}$$

In that case, solutions to the signed distance function equation satisfy the characteristic equations.

$$\begin{aligned}
\frac{d\boldsymbol{x}}{d\sigma} &= \boldsymbol{w} \ , \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
\frac{d\boldsymbol{w}}{d\sigma} &= 0 \ , \quad \boldsymbol{w}(0) = (\nabla\psi)(\boldsymbol{x}_0) \\
\frac{d\psi}{d\sigma} &= 1 \ , \quad \psi(0) = \psi(\boldsymbol{x}(0))
\end{aligned}$$

where σ denotes arc length. These equations can be solved analytically to obtain

$$\boldsymbol{x}(\sigma) = \boldsymbol{x}(0) + \sigma(\nabla\psi)(\boldsymbol{x}(0)) \ , \ \boldsymbol{w}(\sigma) = (\nabla\psi)(\boldsymbol{x}(0)) \ , \ \psi(\sigma) = \psi(\boldsymbol{x}(0)) + \sigma \tag{4}$$

i.e. the curves are straight lines in $(\boldsymbol{x}, \psi)$ space, while $\boldsymbol{w} = \nabla\psi$ is constant along each trajectory.

The characteristic form of the equations suggest that signed-distance functions can be constructed incrementally. Given that ψ is known on $\Omega_r = \{\boldsymbol{x} : |\psi(\boldsymbol{x})| \leq r\}$, then one can extend ψ to $\Omega_{r+\delta}$ using (4). It is easy to show that this reasoning extends to non-smooth signed distance functions, i.e. ones defined by (2). Fast marching methods [Set96, HPCD96] are numerical methods for computing the signed distance function based on this observation. Fast marching methods have two components:

1. A discretization of the signed distance function equation that permits the calculation of the signed distance at a given grid point by using a stencil of nearby values that have already been computed.

2. A marching algorithm, which is a method for determining the order in which grid values are to be computed.

For example, the method in [HPCD96, Set96] uses a first-order accurate discretization of the signed distance function equation, and a marching algorithm based on computing, at each step, the value of ψ that has the minimum magnitude among all of the uncomputed values adjacent to valid values.

A number of problems in numerical simulation related to implicit function representation of surfaces require the computation of the signed distance from a given

surface; for a review see [Set99]. The motivating application for this paper is the use of narrow-band level-set methods for representing the propagation of fronts in large-scale fluid dynamics simulations combined with second-order accurate volume-of-fluid methods [Col01] for discretizing the PDE on either side of the front. This imposes two requirements that have not been simultaneously met by previous methods. The first is the use of a marching method that is a good match for adaptive and parallel implementation based on patch-based domain decomposition. We impose this requirement for compatibility with the software frameworks typically used for high-performance implementations of block-structured adaptive grid methods. In such an approach, the construction of a solution is based on steps that update independently the points on a collection of rectangles whose disjoint union covers the domain, interleaved with steps that communicate ghost cell data. The marching method in [HPCD96, Set96, Dia69] does not fit into this category: it is specified as a serial algorithm, in that the values on a grid are computed one at a time, with the next value / location determined by the previously computed values. Not only is this a poor match for the block-structured software frameworks, but it also imposes a serial bottleneck in a parallel computation. The second requirement is that we obtain a solution that is second-order accurate at all points whose domain of dependence includes no singularities, since the volume-of-fluid discretizations requires that level of accuracy [SBCL06, CGKM06, SAC$^+$]. In all cases, the solution should converge to a signed-distance function, even in regions whose domain of dependence include discontinuities in the derivatives. While second-order accurate algorithms have been proposed [Set99, Cho01, MG07, RS00, SSO94] based on the original marching method, not a great deal of attention has been paid to distinguishing between converging and diverging characteristics for an initial surface that contains kinks in the context of second-order accurate methods.

In the present work, we present a method that meets our requirements. We use a variation on the Global Marching Method in [Kim01]. Given the values at grid points in Ω_r, we compute simultaneously and independently all of the grid values in $\Omega_{r+\delta}$, where δ is comparable to the mesh spacing. Since the method computes the solution at a large number of points independently as local functions of the previously-computed values in Ω_r, the method maps naturally onto a block-structured domain-decomposition implementation. Second, our discretization of the signed distance function equation is analogous to the construction of the fluxes for a second-order Godunov method for a scalar conservation law. It is a hybridization of a high-order and low-order method, where the choice of hybridization coefficient is based on a local curvature calculation. The high-order method is a straightforward difference approximation to the characteristic form of the equations (4). The low-order method is similar to the method in [HPCD96, Set96] but uses a least-squares approach for computing $\nabla\psi$ based on different approximations depending on whether the characteristics are locally converging or diverging. The choice of δ is based on a condition analogous to a CFL condition under which all the points in the high-order stencil should be available for computing the value of ψ at a grid point. The use of a least-

squares algorithm for approximating the gradient in the low-order method involving all of the valid nearest neighbors maximizes the likelihood that there will be sufficient valid points for computing the low-order estimate for ψ when it is needed.

The resulting method is second-order accurate in regions where the solution is smooth, and characteristics trace back to portions of the original surface Γ that are smooth. If there are kinks in the original surface or that form away from the original surface due to convergence of characteristics, the method is first-order accurate in the range of influence of the kinks. The method appears to provide solutions that satisfy the entropy condition, correctly distinguishing between the two directions of propagation from kinks in the original surface. The solution on the side of the surface corresponding to converging characteristics propagates as a kink, while the solution on the side corresponding to diverging characteristics takes the form of a centered expansion fan.

2 Kim's Global Marching Method

We discretize the problem to a grid consisting of equally spaced points in $\mathbb{Z}^{\mathbf{D}}$. We denote the grid-spacing by h. Given $\phi_{\boldsymbol{i}} = \phi(\boldsymbol{i}h)$ where $\boldsymbol{i} \in \mathbb{Z}^{\mathbf{D}}$ and $\boldsymbol{i}h$ in a $\epsilon-$neighborhood of Γ, we wish to compute

$$\psi_{\boldsymbol{i}} \approx \psi(\boldsymbol{i}h) \ , \ |\psi_{\boldsymbol{i}}| \leq R. \tag{5}$$

Our marching algorithm for computing such solutions is given in Figure 1. Here the function $\mathcal{E}(\psi, \Omega^{valid}, \boldsymbol{i})$ computes a value for ψ at $\boldsymbol{i}$ using only the set of values $\{\psi_{\boldsymbol{i}}\}$ that have been computed on Ω^{valid}. $\mathcal{E}$ can be undefined, for example, if there are insufficient points in a neighborhood of $\boldsymbol{i}$ to perform the computation. The quantity σ is a CFL number for the marching method, and depends on the details of $\mathcal{E}$. In determining which points over which to iterate in the **for** loop, we have assumed that $\sigma < 1$. The computation in the **for** loop can be performed in parallel using a domain-decomposition strategy over the points in $\bigcup_{\boldsymbol{s}:|s_d|\leq 1} (\Omega^{valid} + \boldsymbol{s}) - \Omega^{valid}$. In principle, the method described here could iterate an arbitrarily large number of times before updating r. For the discretization method described in the next section, we have observed that `numUpdate` = `0` on the third iteration, so we could replace the **while** loop by one performing a fixed number of iterations before updating r.

3 Discretizing the Signed Distance Function Equation

In this section, we define the discretization of the signed distance function equation used to define $\mathcal{E}$. It is computed as a linear combination of of a low-order(first-order)

```
Ω^new = ∅
r = ε + σh
while r ≤ R do
  for i ∈ ⋃_{s:|s_d|≤1} (Ω^valid + s) − Ω^valid do
    if E(ψ, v, Ω^valid, i) is defined then
      (ψ̃_i, ṽ_i) = E(ψ, v, Ω^valid, i)
      if |ψ̃_i| ≤ r then
        Ω^new += {i}
        numUpdate += 1
      end if
    end if
  end for
  Ω^valid += Ω^new
  ψ = ψ̃, v = ṽ on Ω^new
  Ω^new = ∅
  if numUpdate = 0 then
    r += σh
  end if
  numUpdate = 0
end while
```

Figure 1: The Global Marching Method. In each iteration of the **while** loop, we compute the solution to on points adjacent to $\Omega_{r-\sigma h} \subseteq \Omega^{valid} \subseteq \Omega_r$ independently of the other values being computed in that iteration. After there are no longer any points to compute, we increment $r \to r + \sigma h$.

method and a high-order (second-order) method, with the hybridization coefficient depending on the local curvature. This approach is analogous to that taken in constructing fluxes for hyperbolic conservation laws. The low-order method is based on a least-squares discretization of the gradient that distinguishes between locally converging and diverging characteristics. The signed distance function equation (3) is used to determine the free parameter in the gradient corresponding to the unknown value of Ψ. This step is similar to that used in ([Set96]) and ([HPCD96]). The high-order method is based on solving the characteristic form of the equations.

3.1 Least-Squares Discretization

Given a collection of points $\boldsymbol{p} \in \boldsymbol{P} \subset \mathbb{Z}^{\mathbf{D}}$, we have the following relationship between the values of the distance function, ψ, and the gradient:

$$\frac{1}{h}(\psi(\boldsymbol{i}h) - \psi((\boldsymbol{i}+\boldsymbol{p})h)) = -\boldsymbol{p}\cdot\nabla\psi + O(h) \text{ for } \boldsymbol{p} \in \boldsymbol{P} \tag{6}$$

If $\boldsymbol{P}$ has $\mathbf{D}$ linearly independent elements, then we can use (6) as the starting point for deriving a first-order accurate method for computing solutions to (3). Given

$$\psi_{\boldsymbol{i}+\boldsymbol{p}} \approx \psi((\boldsymbol{i}+\boldsymbol{p})h) \text{ for } \boldsymbol{p} \in \boldsymbol{P} \tag{7}$$

we define $\tilde{\psi} \approx \psi(\boldsymbol{i}h)$, $\boldsymbol{v} \approx (\nabla\psi)(\boldsymbol{i}h)$ as satisfying a least squares solution to the coupled equations:

$$A\boldsymbol{v} = -\frac{1}{h}(\tilde{\psi}\Upsilon - \Psi) \tag{8}$$

where the unknown $\tilde{\psi}$ is viewed as a free parameter, to be determined later, and

$$\Psi = (\psi_{\boldsymbol{i}+\boldsymbol{p}_1}, \psi_{\boldsymbol{i}+\boldsymbol{p}_2}, \ldots \psi_{\boldsymbol{i}+\boldsymbol{p}_r})^T, \tag{9}$$

$$A = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots \boldsymbol{p}_r)^T, \text{and} \tag{10}$$

$$\Upsilon = (1, 1, \ldots 1)^T. \tag{11}$$

Since A is of rank $\mathbf{D}$, the least squares solution to (8) is given by

$$\begin{aligned} \boldsymbol{v} =& -(A^TA)^{-1}A^T\frac{1}{h}(\tilde{\psi}\Upsilon - \Psi) \\ =& \frac{(\tilde{\psi} - \bar{\psi})}{\ell}\hat{\boldsymbol{n}} - (\boldsymbol{\omega}_2 - (\boldsymbol{\omega}_2\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}) \end{aligned} \tag{12}$$

where

$$\boldsymbol{\omega}_1 = -\frac{1}{h}(A^TA)^{-1}A^T\Upsilon \ , \ \boldsymbol{\omega}_2 = -\frac{1}{h}(A^TA)^{-1}A^T\Psi \ , \ \ell = \frac{1}{||\boldsymbol{\omega}_1||} \ , \ \hat{\boldsymbol{n}} = \boldsymbol{\omega}_1\ell \tag{13}$$

$$\bar{\psi} = (\boldsymbol{\omega}_2\cdot\boldsymbol{\omega}_1)\ell^2 = \psi(\boldsymbol{i}h - \ell\hat{\boldsymbol{n}}) + O(h^2)$$

We assume here that $\boldsymbol{\omega}_1$ is not the zero vector. If $\boldsymbol{\omega}_1 = \boldsymbol{\omega}_1(\boldsymbol{P}) = \mathbf{0}$, then the least-squares problem does not produce a value for $\tilde{\psi}$, although the expression (12) for the gradient is still well-defined.

Following [HPCD96, Set96], the condition $||\boldsymbol{v}||^2 = 1$ leads to a quadratic equation for $\tilde{\psi}$:

$$(\tilde{\psi} - \bar{\psi})^2 + \ell^2 ||(\boldsymbol{\omega}_2 - (\boldsymbol{\omega}_2 \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}})||^2 = \ell^2 \tag{14}$$

If (14) has two real roots, we choose the root for which $|\tilde{\psi}| > |\bar{\psi}|$. If (14) has no real roots, we set $\tilde{\psi} = \bar{\psi}$.

We denote by $\mathcal{L}(\psi, \boldsymbol{i}, h, \boldsymbol{P})$ the value of $\tilde{\psi}$ obtained from the least-squares algorithm above. We can then define

$$\mathcal{E}^L(\psi, \Omega^{valid}, \boldsymbol{i}, h) = (\psi^L, \boldsymbol{v}^L) \tag{15}$$

$$\psi^L = s_{\boldsymbol{i}} \min_{\boldsymbol{B}} |\mathcal{L}(\psi, \boldsymbol{i}, h, \boldsymbol{B})| \text{ if } \kappa_{\boldsymbol{i}} < 0 \text{ or } \boldsymbol{\omega}_1 = \mathbf{0} \tag{16}$$

$$= \mathcal{L}(\psi, \boldsymbol{i}, h, \boldsymbol{P}) \qquad \text{if } \kappa_{\boldsymbol{i}} \geq 0 \text{ and } \boldsymbol{\omega}_1 \neq \mathbf{0} \tag{17}$$

$$\boldsymbol{P} = \boldsymbol{U} \cap (\Omega^{valid} - \boldsymbol{i}) \ , \ (A, \Psi, \boldsymbol{\omega}_1) = (A(\boldsymbol{P}), \Psi(\boldsymbol{P}), \boldsymbol{\omega}_1(\boldsymbol{P})) \tag{18}$$

$$\boldsymbol{v}^L = -(AA^T)^{-1} A^T \frac{1}{h} (\psi^L \Upsilon - \Psi) \tag{19}$$

The minimum in (16) is over the collection of all sets $\boldsymbol{B}$ of pairs of adjacent points (2D) / 2x2 blocks of points (3D) contained in $\boldsymbol{U}$ such that $\boldsymbol{B} + \boldsymbol{i} \subset \Omega^{valid}$. The quantity κ is a local estimate of the curvature.

$$\kappa_{\boldsymbol{i}} = \min_{\boldsymbol{t}} s_{\boldsymbol{i}} (\Delta^h \psi)_{\boldsymbol{i}+\boldsymbol{t}} \tag{20}$$

where Δ^h is the $2\mathbf{D}+1$-point centered-difference discretization of Laplacian, and the minimum is taken over all points $\boldsymbol{t} \in [-2 \ldots 2]^{\mathbf{D}}$ such that the stencil for Δ^h evaluated at $\boldsymbol{i} + \boldsymbol{t}$ is contained in Ω^{valid}. The minimum assumption for $\mathcal{E}^L$ to be defined is that at least one of the $\boldsymbol{B}$ in (16) is defined, and at least one of the $\Delta^h \psi$ in (20) is defined. Otherwise, $\mathcal{E}^L$ is undefined.

We use the two different least-squares algorithm depending on the sign of the curvature in order to obtain the correct distance function in the neighborhood of a kink. If the curvature is negative, the characteristics are converging, and the distance function is the minimum over as many candidates as possible based on using the least-squares algorithm on $2^{\mathbf{D}-1}$ points, analogous to choosing the minimum over multiple distinct characteristics that might be reaching the same point. If the curvature is positive, the characteristics are diverging, and the use of the single stencil involving all of the valid points in $\boldsymbol{U} + \boldsymbol{i}$ leads to interpolated intermediate values for $\bar{\psi}$ and $\hat{\boldsymbol{n}}$, analogous to sampling inside a centered rarefaction fan in computing a flux for Godunov's method at a sonic point.

3.2 A Second-Order Accurate Method

We define a function that computes a second order approximation to the distance function and the gradient of the distance function. In the following, let $\pi = \psi, \boldsymbol{v}$ denote the field that we wish to compute at $\boldsymbol{i} \notin \Omega^{valid}$, assuming that π is known on Ω^{valid}. We also assume that we know $\hat{\boldsymbol{v}} \approx (\nabla\psi)(\boldsymbol{i}h)$. The calculation of

$$\bar{\pi} = Q(\pi, \boldsymbol{i}, \hat{\boldsymbol{v}}, h) \approx \pi(\boldsymbol{i}h) \tag{21}$$

is given as follows.

1. Compute $\bar{\boldsymbol{x}}$, the first point along the ray $\{\boldsymbol{i}h - s_{\boldsymbol{i}}\hat{\boldsymbol{v}}\delta : \delta > 0\}$ that intersects a coordinate plane of gridpoints.

$$\bar{\boldsymbol{x}} = \boldsymbol{i}h - s_{\boldsymbol{i}}h\frac{\hat{\boldsymbol{v}}}{\hat{v}_{max}} \tag{22}$$

   where $\hat{v}_{max}$ is the component of $\hat{\boldsymbol{v}}$ whose magnitude is largest, with d_{max} the corresponding coordinate direction.

2. Compute a quadratic interpolant in the coordinate plane containing $\bar{\boldsymbol{x}}$.

$$\boldsymbol{j} = \left\lfloor \frac{\bar{\boldsymbol{x}}}{h} - \frac{1}{2}(\boldsymbol{u} - \boldsymbol{e}^{d_{max}}) \right\rfloor, \ \bar{\boldsymbol{y}} = \bar{\boldsymbol{x}} - \boldsymbol{j}h \tag{23}$$

$$\bar{\pi} = \pi_{\boldsymbol{j}} + \sum_{d \neq d_{max}} \left(\frac{\partial \pi}{\partial x_d}\bar{y}_d + \frac{1}{2}\frac{\partial^2 \pi}{\partial x_d^2}\bar{y}_d^2 \right) + \frac{\partial^2 \pi}{\partial x_{d_1}\partial x_{d_2}}\bar{y}_{d_1}\bar{y}_{d_2} \tag{24}$$

   where all of the derivatives are evaluated at $\boldsymbol{j}h$. The last term in (24) is defined only for $\mathbf{D} = 3$ and $d_1 \neq d_2, d_1, d_2 \neq d_{max}$. We denote by $\boldsymbol{e}^d$ the unit vector in the dth coordinate direction, and $\boldsymbol{u} = (1 \ldots 1)$, both elements of $\mathbb{Z}^{\mathbf{D}}$

The derivatives appearing in the sum in (24) are computed using second-order accurate centered differences at $\boldsymbol{j}h$, assuming $\boldsymbol{j}, \boldsymbol{j} \pm \boldsymbol{e}^d \in \Omega^{valid}$. The mixed derivative is approximated by the average of centered differences:

$$\frac{\partial^2 \pi}{\partial x_{d_1}\partial x_{d_2}} \approx \frac{1}{N}\sum (D^2_{d_1,d_2}\pi)_{\boldsymbol{j}+\frac{1}{2}\boldsymbol{s}} \tag{25}$$

where

$$(D^2_{d_1,d_2}\pi)_{\boldsymbol{k}+\frac{1}{2}\boldsymbol{e}^{d_1}+\frac{1}{2}\boldsymbol{e}^{d_1}} = \frac{1}{h^2}(\pi_{\boldsymbol{k}} + \pi_{\boldsymbol{k}+\boldsymbol{e}^{d_1}+\boldsymbol{e}^{d_1}} - \pi_{\boldsymbol{k}+\boldsymbol{e}^{d_1}} - \pi_{\boldsymbol{k}+\boldsymbol{e}^{d_2}}) \tag{26}$$

is defined if $\boldsymbol{k}, \boldsymbol{k}+\boldsymbol{e}^{d_1}, \boldsymbol{k}+\boldsymbol{e}^{d_2}, \boldsymbol{k}+\boldsymbol{e}^{d_1}+\boldsymbol{e}^{d_2}$ are all in Ω^{valid}. The sum in (25) is taken over all $\boldsymbol{s}$ of the form $\alpha_1\boldsymbol{e}^{d_1} + \alpha_2\boldsymbol{e}^{d_2}$, $\alpha_1 = \pm 1, \alpha_2 = \pm 1$ for which $(D^2_{d_1,d_2})$ is defined, and N is the number of terms in the sum.

Given the function Q defined above, we can define a second-order accurate discretization of the characteristic form of the equations (4) at $\boldsymbol{i}h$. We iterate twice to obtain a sufficiently accurate computation of $\boldsymbol{v}$, computing $\hat{\boldsymbol{v}}$ at the point $\boldsymbol{i}$ using the least-squares algorithm defined in the previous section, and then

$$\hat{\boldsymbol{v}} := Q(\boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}, \hat{\boldsymbol{v}}, h) \ , \ \boldsymbol{v}^H = Q(\boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}, \hat{\boldsymbol{v}}, h) \tag{27}$$

We then use $\boldsymbol{v}^H$ to compute ψ^H.

$$\psi^H = Q(\psi, \Omega^{valid}, \boldsymbol{i}, \hat{\boldsymbol{v}}^H, h) + s_{\boldsymbol{i}} h \left| \frac{\boldsymbol{v}^H}{v^H_{max}} \right| \tag{28}$$

We denote by $\mathcal{E}^H$ the resulting second-order accurate method for computing $\psi, \boldsymbol{v}$.

$$\mathcal{E}^H(\phi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}) \equiv (\psi^H, \boldsymbol{v}^H). \tag{29}$$

If the low-order method is defined, and the points required for the various evaluations of Q are defined, then (29) is defined. Otherwise, it is undefined.

3.3 Hybridization

We hybridize the low- and high-order methods based on the magnitude of the curvature. Assuming that both $\mathcal{E}^L$ and $\mathcal{E}^H$ are defined, we compute

$$(\psi^L, \boldsymbol{v}^L) = \mathcal{E}^L(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}) \tag{30}$$

$$(\psi^H, \boldsymbol{v}^H) = \mathcal{E}^H(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}) \tag{31}$$

$$\mathcal{E}(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}) = \big((1-\eta_{\boldsymbol{i}})\psi^H + \eta_{\boldsymbol{i}}\psi^L, (1-\eta_{\boldsymbol{i}}^2)\boldsymbol{v}^H + \eta_{\boldsymbol{i}}^2 \boldsymbol{v}^L)\big) \tag{32}$$

where the hybridization parameter η is given by

$$\eta_{\boldsymbol{i}} = \begin{cases} 1 & \text{if } h|\Delta^h \psi|_{max} > C \\ \frac{h}{C}|\Delta^h \psi|_{max} & \text{otherwise,} \end{cases} \tag{33}$$

$$|\Delta^h \psi|_{max} = \max_{\boldsymbol{t}} |(\Delta^h \psi)_{\boldsymbol{i}+\boldsymbol{t}}| \tag{34}$$

where the range over which the max is taken is the same as in (20). If the high-order value $\mathcal{E}^H(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i})$ is not defined, but the low-order value is, then $\mathcal{E}(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}) = \mathcal{E}^L(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i})$. If the low order value is not defined, then $\mathcal{E}(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i})$ is not defined. The constant C is an empirically determined parameter, independent of h. In our numerical experiments, $C = 1$.

If $\sigma < \frac{1}{\sqrt{5}}$, and we replace the values of ψ, $\boldsymbol{v}$ on grid points in Ω_r with those of a smooth distance function ψ^e, it is possible to show that, for sufficiently small h, both $\mathcal{E}^H$ and $\mathcal{E}^L$ are defined for all grid points in $\Omega_{r+\sigma h}$ and that

$$\psi^H_{\boldsymbol{i}} = \psi^e(\boldsymbol{i}h) + O(h^3) \ , \ \boldsymbol{v}^H = \nabla\psi^e(\boldsymbol{i}h) + O(h^3) \tag{35}$$

$$\psi^L_{\boldsymbol{i}} = \psi^e(\boldsymbol{i}h) + O(h^2) \ , \ \boldsymbol{v}^L = \nabla\psi^e(\boldsymbol{i}h) + O(h) \tag{36}$$

from which it follows that

$$\mathcal{E}(\psi, \boldsymbol{v}, \Omega^{valid}, \boldsymbol{i}) = (\psi^e(\boldsymbol{i}h), \nabla\psi^e(\boldsymbol{i}h)) + O(h^3). \tag{37}$$

Thus we expect that the global error in our solution will be $O(h^2)$. This also explains why we use η^2, rather than η, to hybridize the gradient calculation. Otherwise, we would introduce an $O(h^2)$ contribution to the error in the gradient at every step, leading to a first-order accurate method for the gradient, and hence for ψ. In the neighborhood of kinks in the level sets of ψ, the value of the curvature is $O(h^{-1})$, and we will use the low-order method, leading to a first-order accurate method in the range of influence of the kinks.

3.4 Initialization

We now describe the method we use to provide test problems with an initial narrow band three or four cells wide. We are given an initial representation of the surface by a discretized implicit function, from which we construct the distance function and the gradient of the distance function an $O(h)$ distance. If the surface is smooth, then our initialization procedure is an $O(h^2)$ approximation to the distance function. If the characteristics cross near the surface or the surface is not smooth, then the initialization reduces to a first-order accurate method within the range of influence of the kink.

We require some more notation. Denote by $(G^0\phi)$ the centered difference approximation to the gradient of ϕ. Given a grid location $\boldsymbol{i}$, let $d = \frac{\phi_{\boldsymbol{i}}}{||(G^0\phi)_{\boldsymbol{i}}||}$. Let $\mathcal{P} \subset \Omega$ denote $\boldsymbol{i}$ and its neighbors. Let

$$m_{\boldsymbol{i}} = \min_{\boldsymbol{p}\in\mathcal{P}} ||(G^0\phi)_{\boldsymbol{p}}|| \tag{38}$$

$$M_{\boldsymbol{i}} = \max_{\boldsymbol{p}\in\mathcal{P}} ||(G^0\phi)_{\boldsymbol{p}}|| \tag{39}$$

We choose a non-dimensional parameter, ϵ, independent of h and atempt to detect a discontinuity in the gradient by checking whether M exceeds m by an amount greater than ϵ. If so, we make a robust but lower order estimate of the gradient:

$$\text{if } 1 - \frac{m_{\boldsymbol{i}}}{M_{\boldsymbol{i}}} \geq \epsilon \text{ then} \tag{40}$$

$$\boldsymbol{v} = (G^0\phi)_{\boldsymbol{p}} : ||(G^0\phi)_{\boldsymbol{p}}|| = M_{\boldsymbol{i}} \tag{41}$$

In our numerical experiments, $\epsilon = \frac{1}{2\sqrt{2}}$. Alternatively, if

$$1 - \frac{m_{\boldsymbol{i}}}{M_{\boldsymbol{i}}} < \epsilon, \tag{42}$$

then we define a point,

$$\boldsymbol{x}_0 = \boldsymbol{i}h - d\frac{(G^0\phi)_{\boldsymbol{i}}}{||(G^0\phi)_{\boldsymbol{i}}||} \tag{43}$$

Table 1: Solution error for 2**D** Curve in Polar Coordinates: $h = \frac{1}{100}, \frac{1}{200}$

L^1 Norm	rate	L^2 Norm	rate	L^∞ Norm	rate
4.4566e-02		1.0240e-02		2.1393e-02	
1.0592e-02	2.07	2.4083e-03	2.08	5.9743e-03	1.84

At $\boldsymbol{x}_0$ we bi-quadratically interpolate an estimate of the gradient $\boldsymbol{v}$. Finally, we use root-finding in the direction $\boldsymbol{v}$ to make an estimate of the distance.

4 Numerical Results

For our fast marching problems, we always compute the max norm of the error. Where useful, we also compute the the L_1 and the L_2- norm of the solution error.

For a discrete variable, ζ, the max norm is given by

$$||\zeta||_\infty = \max_i |\zeta_i|. \tag{44}$$

The L_p-norm is given by

$$||\zeta||_p = \left(\sum_i |(\zeta_i)^p h^{\mathbf{D}} \right)^{\frac{1}{p}} \tag{45}$$

For all of the test problems that follow we have used a marching parameter of $\sigma = \frac{1}{2\sqrt{5}}$.

Our first test problem uses the implicit function $r = 2\cos 4\theta + 7$. The domain has a lower left corner with coordinates $(-10, -10, -10)$ and an upper right corner with coordinates $(10, 10, 10)$. The initial bandwidth is approximately six grid cells wide at all resolutions. The final bandwidth is approximately 1.2. Calculations were performed on grids with $h = \frac{1}{100}, \frac{1}{200}$, and $\frac{1}{400}$. Richardson error extrapolation was used to calculate the results presented in Table 1. The solution is shown in Figure 2 and the error is shown in Figure 3.

Our second test problem has as its zero-level set a surface of revolution. The domain has a lower left corner with coordinates $(-10, -10, -10)$ and an upper right corner with coordinates $(10, 10, 10)$. The surface is centered at $(0, 0, 0)$ and obtained by rotating the function $r = 2\cos 2\theta + 7$ around the y-axis. The initial bandwidth is approximately six grid cells wide at all resolutions. The final bandwidth is 1.5. Calculations were performed on grids with $h = \frac{1}{100}, \frac{1}{200}$, and $\frac{1}{400}$. Richardson error estimation was used to calculate the results presented in Table 2. Slices of the error are presented in Figure 4.

Our next example uses as an implicit function whose zero set is the surface of a cube. In this case, to test the robustness of the algorithm we initialized the an-

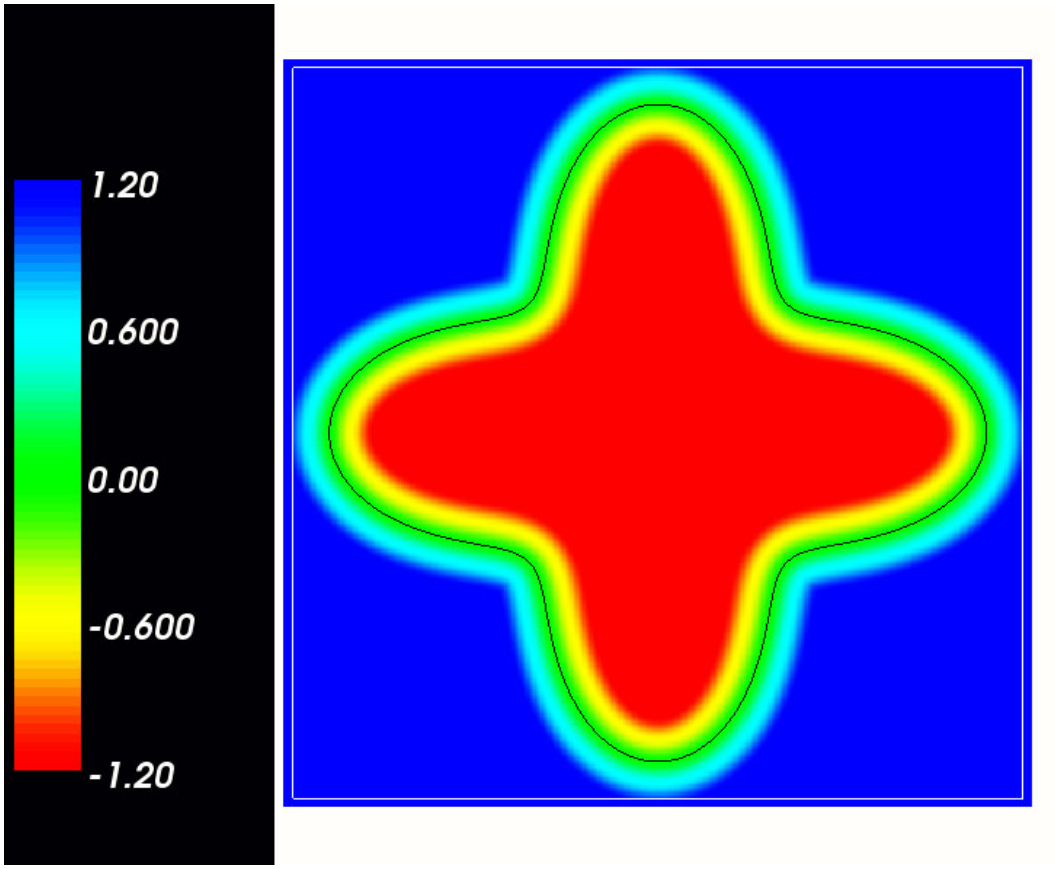

Figure 2: Curve in Polar Coordinates

Table 2: Solution Error for Surface of Revolution: $h = \frac{1}{100}, \frac{1}{200}$

L^1 Norm	rate	L^2 Norm	rate	L^∞ Norm	rate
7.0725 e-01		1.5893 e-02		7.2842e-04	
1.2275 e-01	2.52	3.0446 e-03	2.38	1.813e-04	2.00

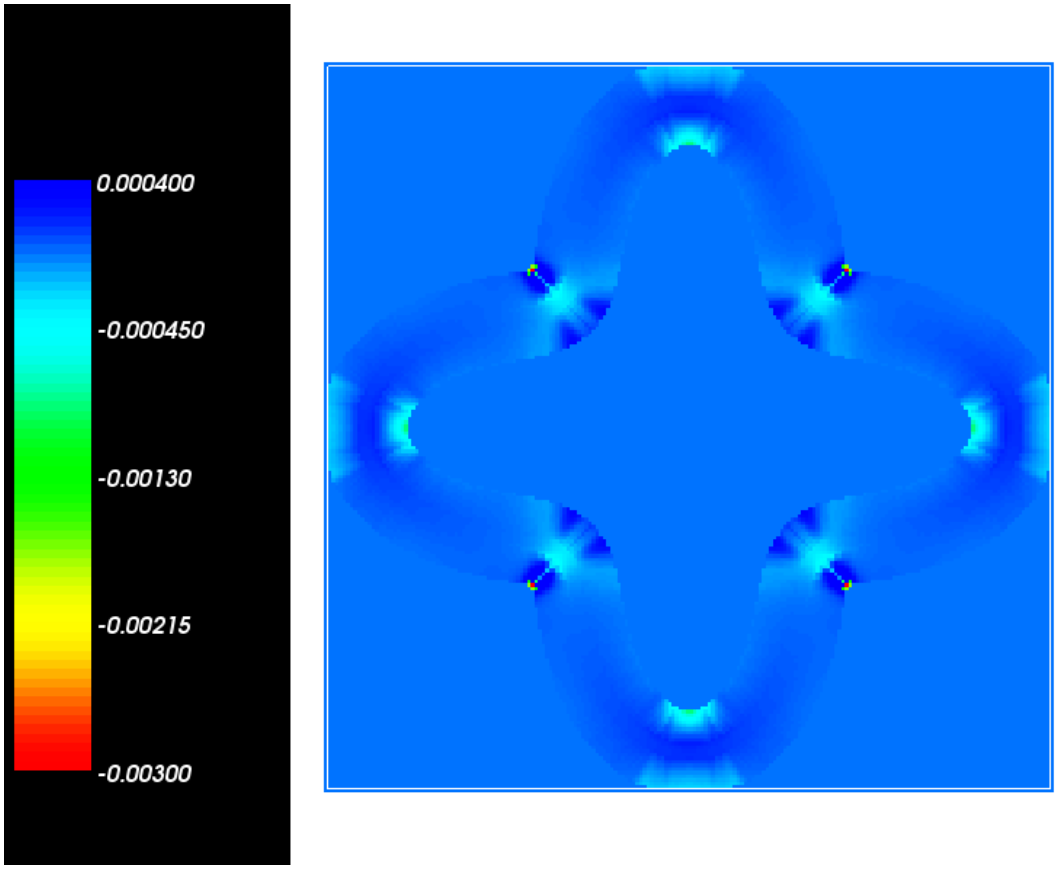

Figure 3: Error for a Curve given in Polar Coordinates

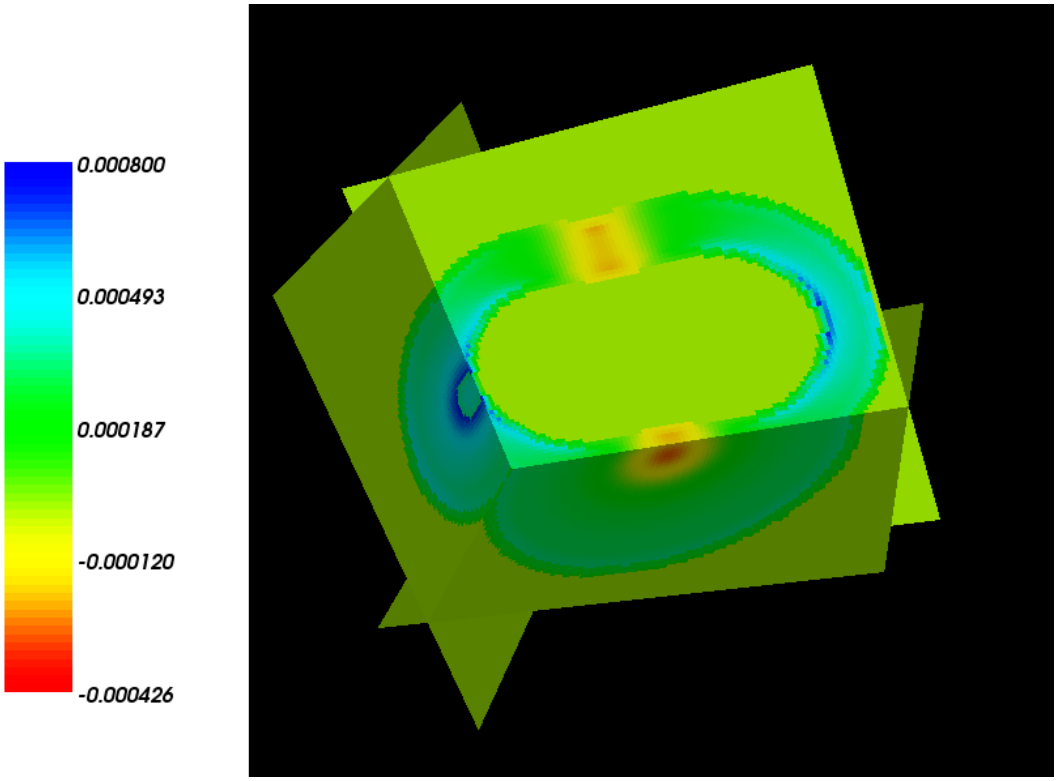

Figure 4: Slices of the error for a Surface of Revolution

Table 3: Solution error for Distance to a Cube : $h = \frac{1}{50}, \frac{1}{100}, \frac{1}{200}$

L^∞ Norm	rate
.00493	
.00260	.92
.0013	1.0

Table 4: Solution error for distance to a union of parallelopipeds: $h = \frac{1}{50}, \frac{1}{100}, \frac{1}{200}$

L^∞ Norm	rate
.00120	.
.000580	1.05

nular region to the wrong weak solution of the signed distance function equation. In particular, where the characteristics diverge we do not round the corners in the initial narrow band. Nonetheless our algorithm extends this initial data to a distance function.

In this example, the initial band has a diameter of about four grid cells at the coarse resolution. The final bandwidth is about two and one half times the diameter of the initial band. Since the only error occurs in places where the gradient is discontinuous, we present the max norm of the error in Table 3

Our final example uses an implicit function generated by taking the union of parallelopipeds. The zero-set is in the shape of a cube whose corners are removed. Two dimensional slices are in the shape of a cross. This example tests cases where characteristics meet at a corner as well cases where the characteristics diverge at a corner.

In this problem the domain has a lower left corner with coordinates $(0, 0)$ and an upper right corner with coordinates $(1, 1)$. The initial band is approximately six cells in diameter at all resolutions. The final bandwidth is 0.15 Since the errors only occur in places where the gradient is discontinuous, we present the max norm of the error in Table(4). The error is in Figure (5). Three isosurfaces, including the zero level set, are presented in Figures (6), (7) and (8).

5 Conclusion

We have described a numerical method for solving the signed distance function equation that is second-order accurate at points whose domain of dependence includes no singularities, which is useful for second-order accurate volume-of-fluid discretizations.

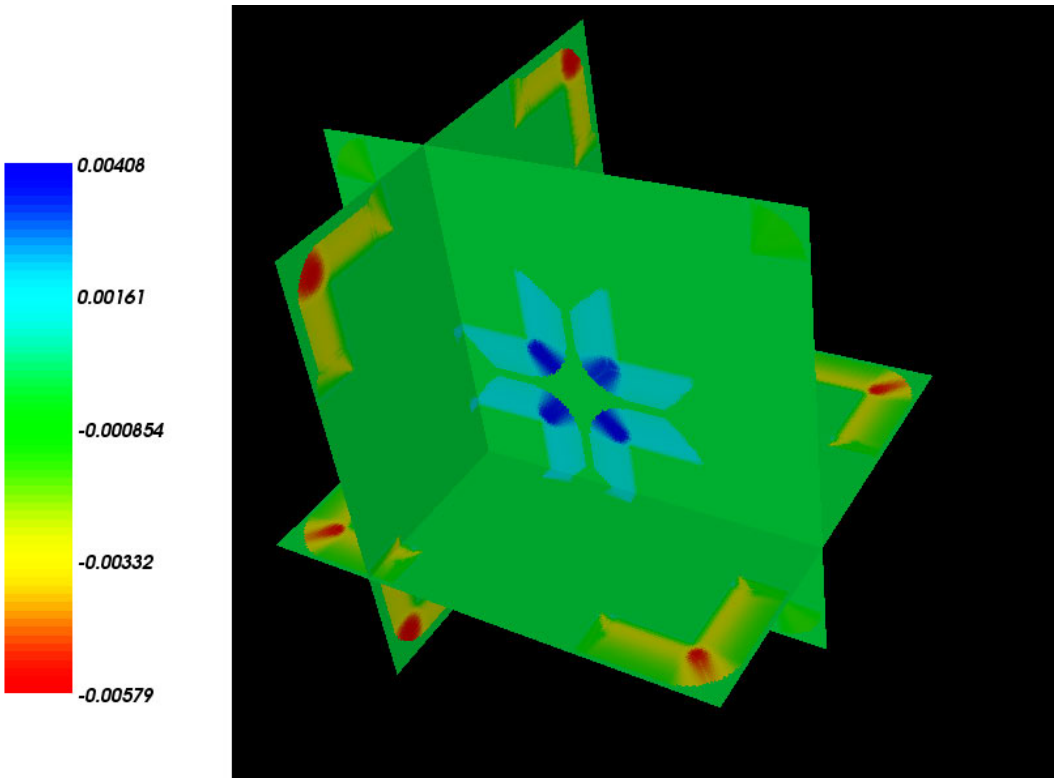

Figure 5: Slices of the error for a union of parallelopipeds

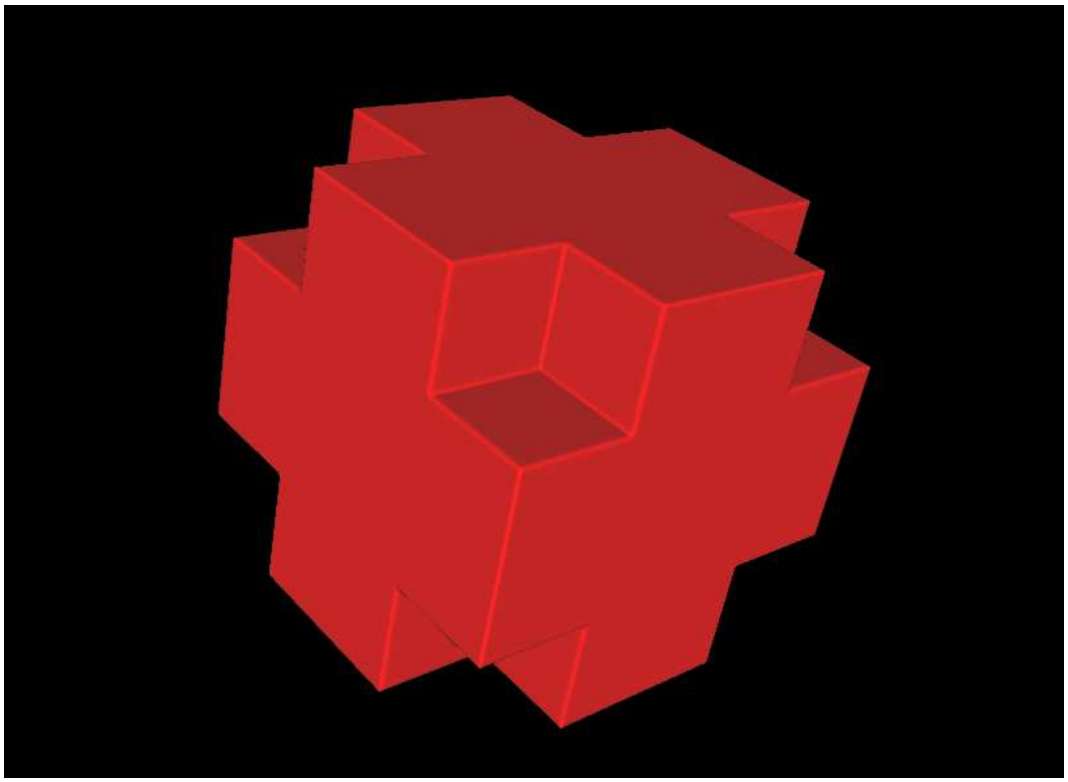

Figure 6: The zero isosurface of the union of parallelopipeds

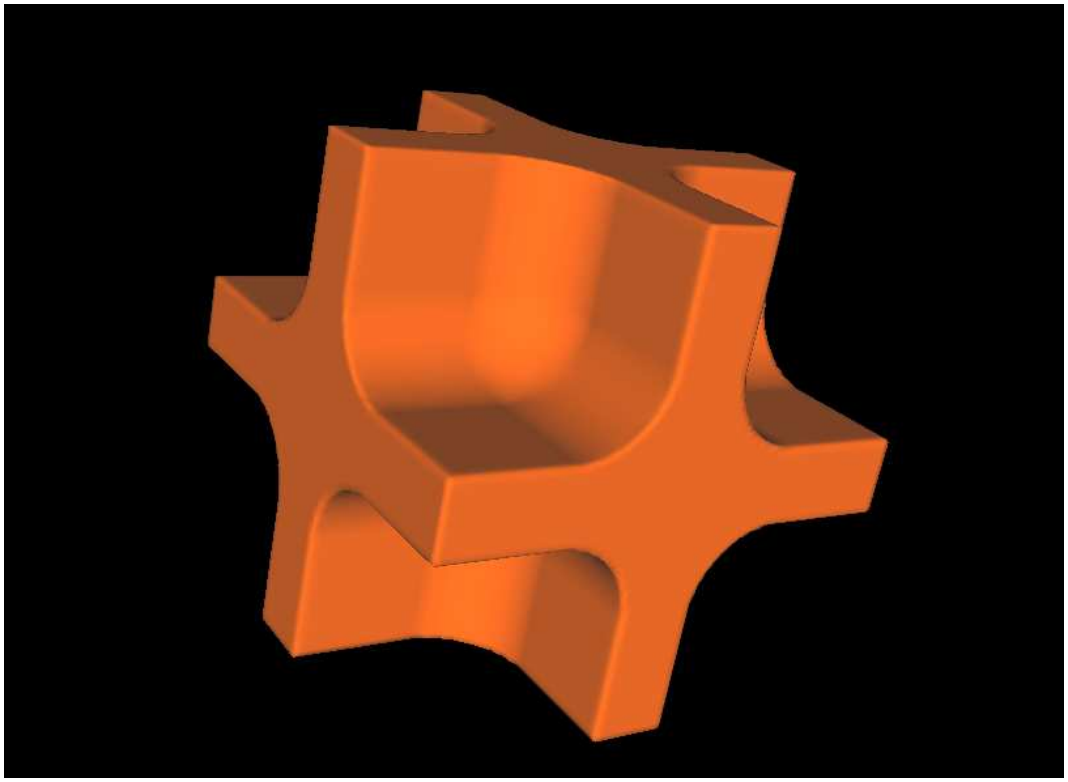

Figure 7: An interior isosurface (at a distance = -.12 from the zero set) of the union of parallelopipeds

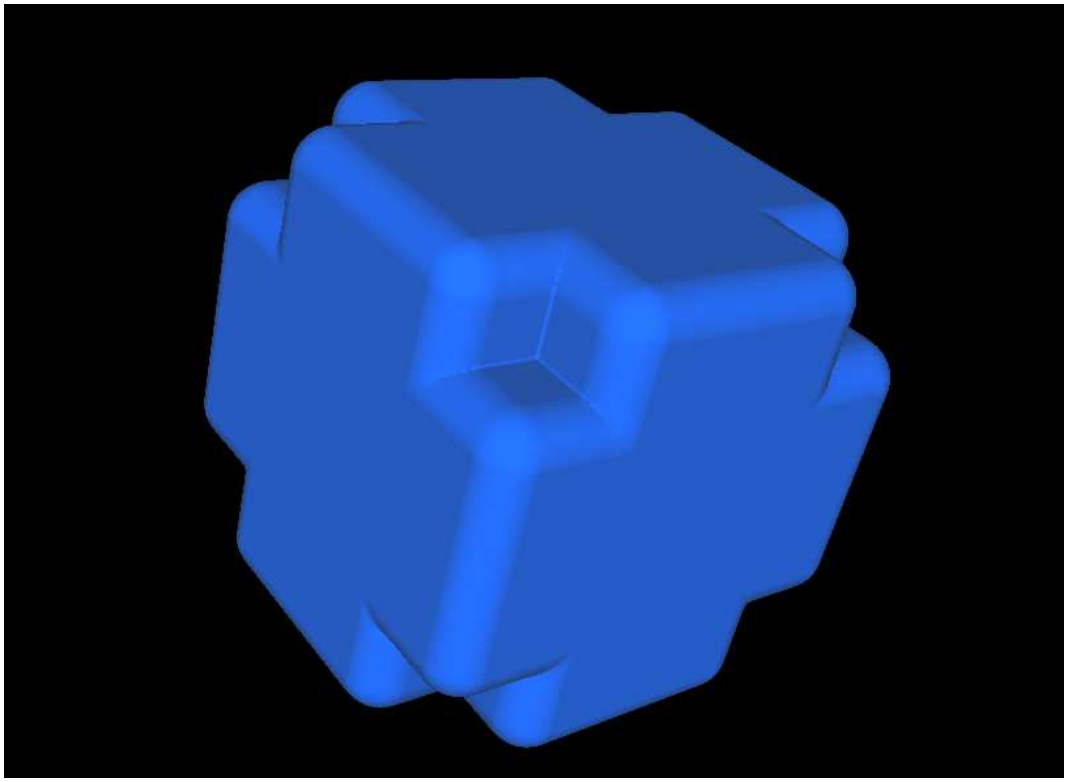

Figure 8: An exterior isosurface (at a distance = .12 from the zero set) of the union of parallelopipeds

A salient feature of our algorithm is the hybridization of a high-order and low-order method, where the choice of hybridization coefficient is based on a local curvature calculation. The resulting calculation appears to provide solutions that satisfy the entropy condition, correctly distinguishing between the two directions of propagation from kinks in the original surface. In addition, we use a marching method that is a good match for adaptive and parallel implementation based on patch-based domain decomposition, which is the software framework typically used for high-performance implementations of block-structured adaptive grid methods.

Our future work will focus on tracking moving fronts in hyperbolic problems. In these problems, the motion of the interface naturally decomposes into advection by a vector velocity combined with motion of the interface normal to itself at a known scalar speed. The importance of the signed distance function equation may be observed in the special case where the vector velocity is zero and the scalar speed is spatially constant. In this context, a method of solving the Hamilton-Jacobi reduces to a method for solving signed distance function equation, up to a relabeling of contours, which leads to the conclusion that numerical methods for Hamilton-Jacobi can be no more accurate than the associated solution to the signed distance function equation. Considering the general front-tracking problem, one may begin by extending velocites and scalar speeds in the normal direction off the interface by solving the transport equation, as was done in [AS99]. Established algorithms for advection may be employed for the velocity component of the motion, while an algorithm for solving the signed distance function equation may be employed for motion given by scalar speeds.

References

[AS99] D. Adalsteinsson and J.A. Sethian. A fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148:2–22, 1999.

[CGKM06] P. Colella, D. T. Graves, B. Keen, and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *J. Comput. Phys.*, 211:347–66, 2006.

[Cho01] D.L. Chopp. Some improvements of the fast marching method. *SIAM Journal on Scientific Computing*, 23(1):230–44, 2001.

[Col01] P. Colella. Volume-of-fluid methods for partial differential equations. *In: E.F. Toro, Editor, Godunov Methods: Theory and Applications, Kluwer Academic/Plenum Publishers, New York*, 3, 2001.

[Dia69] R. Dial. Algorithm 360: Shortest path forest with topological ordering. *Communications of the ACM*, 12:632–633, 1969.

[HPCD96] J. Helmsen, E.G. Puckett, P. Colella, and M. Dorr. Two new methods for simulating photolithography development. *International Symposium on Microlithography*, 2726:503–555, 1996.

[Kim01] S. Kim. An O(N) level set method for eikonal equations. *SIAM J. Sci. Computing*, 22:2178–2193, 2001.

[MG07] Chohong Min and Frederic Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *J. Comput. Phys.*, 225:300–321, 2007.

[RS00] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.

[SAC+] P. Schwartz, D. Adalsteinsson, P. Colella, A. Arkin, , and M. Onsum. Numerical computation of diffusion on a surface. *Proc. Natl. Aca. Sci.*, 102.

[SBCL06] P. Schwartz, M. Barad, P. Colella, and T. Ligocki. A Cartesian grid embedded boundary method for the heat equation and poisson's equation in three dimensions. *J. Comput. Phys.*, 211(2):531–550, 2006.

[Set96] J.A. Sethian. A fast marching method for monotonically advancing fronts. *Proc. Natl. Aca. Sci.*, 93:1591, 1996.

[Set99] J.A. Sethian. Fast marching methods. *SIAM Review*, 41:199–235, 1999.

[SSO94] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114:146–159, 1994.