SIAM J. Sci. Comput.
Vol. 33, No. 5, pp. 2402–2420

A THIRD ORDER ACCURATE FAST MARCHING METHOD FOR THE EIKONAL EQUATION IN TWO DIMENSIONS[*]

SHAHNAWAZ AHMED[†], STANLEY BAK[‡], JOYCE MCLAUGHLIN[§], AND DANIEL RENZI[¶]

Abstract. In this paper, we develop a third order accurate fast marching method for the solution of the eikonal equation in two dimensions. There have been two obstacles to extending the fast marching method to higher orders of accuracy. The first obstacle is that using one-sided difference schemes is unstable for orders of accuracy higher than two. The second obstacle is that the points in the difference stencil are not available when the gradient is closely aligned with the grid. We overcome these obstacles by using a two-dimensional (2D) finite difference approximation to improve stability, and by locally rotating the grid 45 degrees (i.e., using derivatives along the diagonals) to ensure all the points needed in the difference stencil are available. We show that in smooth regions the full difference stencil is used for a suitably small enough grid size and that the difference scheme satisfies the von Neumann stability condition for the linearized eikonal equation. Our method reverts to first order accuracy near caustics without developing oscillations by using a simple switching scheme. The efficiency and high order of the method are demonstrated on a number of 2D test problems.

Key words. Hamilton–Jacobi, Eikonal equation, fast marching

AMS subject classification. 65N06

DOI. 10.1137/10080258X

1. Introduction.

In this paper we focus on third order accurate numerical solutions of a particular static Hamilton–Jacobi equation, the eikonal equation,

$$|\nabla T| = s(x). \tag{1}$$

This equation is most often used to model the first arrival times, T, of a wave or moving interface. The slowness, or inverse, of the speed of the wave or moving interface is $s(x)$. Even for very simple problems with smooth boundary data and constant slowness, s, the solution, T, can be multivalued. Uniqueness is restored by restricting attention to the "viscosity" solution [6]. This is similar to the use of entropy conditions to define a unique physically relevant solution for hyperbolic conservations laws. Numerical techniques are well developed for hyperbolic conservation laws (see [5] for an extensive review) and have been adapted to solve time-dependent Hamilton–Jacobi equations [23].

We first emphasize that there is a complication involved in generating a numerical solution to a static Hamilton–Jacobi equation that does not exist for time-dependent Hamilton–Jacobi equations or hyperbolic conservation laws. For hyperbolic conservation laws and Hamilton–Jacobi equations the characteristics are very important. In a loose sense stable schemes need to use points that are found backwards along the

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section July 19, 2010; accepted for publication (in revised form) June 6, 2011; published electronically September 29, 2011.
http://www.siam.org/journals/sisc/33-5/80258.html

[†] Alfaisal University, P.O. Box 50927, Riyadh, Saudi Arabia (sahmed@coe.alfaisal.edu).

[‡] Department of Computer Science, University of Illinois, Chicago, IL (sbak2@illinois.edu).

[§] Mathematical Sciences Department, Rensselaer Polytechnic Institute, Troy, NY 12180 (mclauj@rpi.edu).

[¶] Department of Mathematical Sciences, Weil Cornell Medical College in Qatar, Sun Prairie, WI (danielprenzi@yahoo.com).

characteristic. For time-dependent problems, we know a priori one component of the characteristic. This provides a "marching" direction. The problem is solved forward in time, and the solution at each point is calculated from points at previous time steps. Static equations are fully nonlinear, and there is no such a priori "marching" direction. In fact, there can be different marching directions at different locations in the domain. Fortunately, for the eikonal equation the direction of the characteristic is in the same direction as the gradient. The solution can be obtained in one pass if the points in the domain are ordered from smallest arrival time to largest arrival time and an upwind scheme is used to discretize equation (1). Of course this ordering is not known in advance. For this reason, fast eikonal solvers generally consist of two parts: (1) a local solver that computes the arrival time at a grid point using only smaller arrival times at nearby grid points; and (2) an ordering strategy. We describe these two parts in the following two paragraphs.

There are three categories of local solvers used to solve equation (1). The first strategy is to use upwind finite differences [18, 19]. The primary advantage of the finite difference strategy is that it is very fast and easy to program. There are difficulties with extending this technique to higher order accuracy, as we explain in sections 2 and 3. This is the strategy we will use in this paper. The second strategy is to use dynamic programming techniques. This is a semi-Lagrangian approach where the problem is recast as a optimal control problem; see [2]. In this formulation the value function of the optimal cost functional that satisfies the Hamilton–Jacobi–Bellman equation is the arrival time. This method is easy to extend to higher orders of accuracy, but is very expensive because the method requires iterations and high-dimensional interpolation. We also note that this method can be used to solve more general Hamilton–Jacobi equations. The last strategy is to use finite element techniques [11]. An advantage of this method over finite difference techniques is that the local finite element solver is very compact, that is, the solution at each element requires only the solution at neighboring elements. For some ordering strategies, in particular the fast sweeping method described in the paragraph below, a compact local solver is needed to ensure fast convergence. Extending these methods to orders of accuracy greater than two is an active area of current research.

Applying one of the local solvers described above to equation (1) leads to a nonlinear system of equations that has to be solved. Generally nonlinear systems require a large number of iterations to solve. However, for this problem the nonlinear system needs only one local solve per grid point if the grid points are ordered from smallest arrival time to largest arrival time. Eikonal solvers use an ordering strategy to take advantage of this fact. The two most common ordering strategies are the fast marching method [10, 18, 20, 21, 22, 25] and the fast sweeping method [4, 7, 8, 13, 14, 24, 30]. The fast marching method is an adaption of Dijkstra's algorithm for finding the shortest path on a network. This method essentially computes the optimal ordering "on the fly." The running time is $O(N \log N)$ because sorting is required. The fast sweeping method uses alternate direction Gauss–Seidel iterations. In n dimensions the characteristic directions are split into 2^n groups, and the method attempts to solve all the characteristics in each group concurrently. The iterations required are bounded by the maximum number of times a characteristic changes direction into another group, so the running time is $O(N)$, but now the constant depends heavily on the slowness function, $s(x)$. There have also been a number of strategies for improving these methods, by either developing an $O(N)$ fast marching method [9, 27] or reducing the size and variability of the constant in the fast sweeping method [3]. We stress that optimized low order accurate eikonal solvers are fast. One should expect to solve equation (1) using the local solver about 2–3 times per grid point in two dimensions for all but the most degenerate slowness profiles. The goal of this paper is to solve equation (1) to third order of accuracy with the same efficiency using the fast marching method.

There are several high order accurate methods for solving the eikonal equation. The first method is to introduce a time-dependent term to the eikonal equation and solve the problem to steady state through iteration. High order accurate methods for time-dependent Hamilton–Jacobi Equation are well developed [23]. Additionally, alternate direction Gauss–Seidel iterations have been shown to speed up the convergence to steady state [28, 29]. However, the number of computations required per grid point depends on the size of the problem and is typically two orders of magnitude greater than the required computations per grid point of the low order accurate eikonal solvers described in the above paragraph. The level set method [12, 17, 19] can also be used to solve the eikonal equation [16]. With this method the arrival times are embedded as the zero level set of a higher-dimensional function, at the expense of adding another dimension to the problem. Narrow banded level set methods [1, 15] were created to address this problem, but to maintain the high order of accuracy the "re-initialization" process still requires a higher order accurate computation of the signed distance function, which is the solution to an eikonal equation. In addition, there is a CFL condition, which for some problems causes additional computational effort. We note that the methods described in this paragraph can also solve more general Hamilton–Jacobi equations.

In this paper we extend the two dimensional (2D) fast marching method to third order of accuracy. The fast marching method was extended to second order of accuracy some time ago [19] by incorporating second order accurate one-sided finite differences in the local solver combined with a simple switching scheme to drop the method to first order accuracy in "rough" regions where the solution is not smooth. This switching mechanism parallels the role of a slope limiter in hyperbolic conservation laws. The resulting local solver is causal (depends only on grid points with smaller arrival times), and a Dijkstra-like iteration strategy yields the same fast convergence. There are two obstacles to extending the fast marching method to orders of accuracy greater than two. The first is that using one-sided differences is unstable when the order is higher than two. We overcome this problem by using a 2D one-sided difference approximation. The second obstacle is that the points required by the difference stencils are not available when the gradient is nearly aligned with a grid line. In this case, we will rotate the grid 45 degrees (i.e., use derivatives along the diagonals) so that the gradient is no longer nearly aligned with a grid line.

The rest of the paper is composed as follows. In sections 2 and 3 we describe the fast marching method and its second order accurate extension. Then in section 4 we extend the fast marching method to third order of accuracy and give results for several 2D problems in section 5. Finally, we give some concluding remarks in section 6.

2. Fast marching method.

In this section we describe the 2D fast marching method to solve the eikonal equation,

$$|\nabla T| = s(x), \tag{2}$$

on a uniform grid with constant step size h. The fast marching method uses an upwind discretization of the eikonal equation,

$$\max(-D^{+x}T_{i,j}, D^{-x}T_{i,j}, 0)^2 + \max(-D^{+y}T_{i,j}, D^{-y}T_{i,j}, 0)^2 = s_{i,j}^2, \tag{3}$$

where the first order accurate difference operators are defined as

$$\begin{aligned} D^{+x}T_{i,j} &= (T_{i+1,j} - T_{i,j})/h, \\ D^{-x}T_{i,j} &= (T_{i,j} - T_{i-1,j})/h, \\ D^{+y}T_{i,j} &= (T_{i,j+1} - T_{i,j})/h, \\ D^{-y}T_{i,j} &= (T_{i,j} - T_{i,j-1})/h, \end{aligned}$$

and an ordering strategy based on Dijkstra's algorithm for finding the shortest path on a network. Each grid point, $x_{i,j}$, is given one of three labels: (1) *known*: the arrival time at $x_{i,j}$ is known and will not be changed; (2) *trial*: $x_{i,j}$ is a neighbor of an accepted grid point; (3) *far*: $x_{i,j}$ is not a neighbor of an accepted grid point. The fast marching algorithm is as follows:

1. Find the trial grid point with smallest arrival time and change it to known.
2. Compute the arrival time at all neighbors that are not known using (3) and place these neighbors in trial if they are not already.
3. If the trial set is not empty, return to (1)

The running time of this algorithm is $O(N \log N)$. We refer the interested reader to [18] for more details on the fast marching method.

3. Second order accurate fast marching method.

In this section we analyze some interesting properties of the second order accurate fast marching method. The fast marching method was extended to second order [19] by using a second order one-sided finite difference stencil,

$$(T_x)_{i,j} = \max\left(\frac{3}{2}T_{i+2,j} - 2T_{i+1,j} + \frac{1}{2}T_{i,j}, \frac{3}{2}T_{i,j} - 2T_{i-1,j} + \frac{1}{2}T_{i-2,j}, 0\right)/h,$$

$$(T_y)_{i,j} = \max\left(\frac{3}{2}T_{i,j+2} - 2T_{i,j+1} - \frac{1}{2}T_{i,j}, \frac{3}{2}T_{i,j} - 2T_{i,j-1} - \frac{1}{2}T_{i,j-2}, 0\right)/h,$$

at most points in the domain. For the fast marching method to work, the difference stencil can only contain points that have smaller arrival times. There are two cases where this stencil cannot be used. The first is at or near caustics, that is, places where two characteristics converge and the solution is only Lipschitz continuous. Note that only first order accuracy can be obtained in these regions. The second case is when the gradient is nearly aligned with the grid. In this case one component of the gradient is nearly zero and there are not two upwind points available (i.e., two points with smaller arrival time) to calculate the derivative of the nearly zero component of the gradient. For these reasons switches are used to revert to the first order method. With the switches the discretization scheme [19] is

$$\begin{aligned} &\max\left(-\left(D^{+x}T_{i,j} - \frac{h}{2}sw^{+x}_{i,j}(D^{+x})^2T_{i,j}\right), D^{-x}T_{i,j} + \frac{h}{2}sw^{-x}_{i,j}(D^{-x})^2T_{i,j}, 0\right)^2 \\ &+ \max\left(-\left(D^{+y}T_{i,j} - \frac{h}{2}sw^{+y}_{i,j}(D^{+y})^2T_{i,j}\right), D^{-y}T_{i,j} + \frac{h}{2}sw^{-y}_{i,j}(D^{-y})^2T_{i,j}, 0\right)^2 \\ &= s^2_{i,j}, \end{aligned}$$

where the switches are defined as follows:

$$sw_{i,j}^{+x} = \begin{cases} 1 & \text{if } T_{i+1,j} \text{ and } T_{i+2,j} \text{ are known and } T_{i+2,j} < T_{i+1,j}, \\ 0 & \text{otherwise,} \end{cases}$$

$$sw_{i,j}^{-x} = \begin{cases} 1 & \text{if } T_{i-1,j} \text{ and } T_{i-2,j} \text{ are known and } T_{i-2,j} < T_{i-1,j}, \\ 0 & \text{otherwise,} \end{cases}$$

$$sw_{i,j}^{+y} = \begin{cases} 1 & \text{if } T_{i,j+1} \text{ and } T_{i,j+2} \text{ are known and } T_{i,j+2} < T_{i,j+1}, \\ 0 & \text{otherwise,} \end{cases}$$

$$sw_{i,j}^{-y} = \begin{cases} 1 & \text{if } T_{i,j-1} \text{ and } T_{i,j-2} \text{ are known and } T_{i,j-1} < T_{i,j-2}, \\ 0 & \text{otherwise.} \end{cases}$$

The reader may find the fact that the scheme will use lower order differences even on smooth problems troubling and question whether the scheme is really second order. These switches parallel the use of slope limiters for hyperbolic conservation laws, where it is well known that only first order can be obtained near extremal points. In fact, the second order fast marching method does not automatically revert to first order if the points aren't available for one of the derivatives. For example, if there are no neighboring points available for one of the derivatives, that derivative is assigned a value of zero leading to one of the following equations:

$$|T_x| = s(x), \tag{4}$$

$$|T_y| = s(x). \tag{5}$$

The remaining derivative is still approximated with a second order difference scheme. Recall that this happens when the characteristic is roughly aligned with the grid, so one component of the gradient is small. It turns out that we are very fortunate when T is smooth. In this case the neglected derivative is small enough so that second order accuracy is maintained. We make these ideas precise in the following theorem.

THEOREM 1. *The second order fast marching method is globally second order, that is, second order in the max norm, if the solution, T, is smooth (has at least three continuous derivatives) on a closed rectangular 2D domain, Ω, and the slowness is bounded below, $s(x) > m$.*

Proof. Let $\vec{x}_{i,j} = (x_i, y_j)$ be a regularly spaced grid on Ω with constant step size h and let $T_{i,j} = T(\vec{x}_{i,j})$. Without loss of generality, assume $T_x(\vec{x}_{i,j}) > T_y(\vec{x}_{i,j}) \geq 0$. Using the lower bound on the slowness, $T_x(\vec{x}_{i,j}) \geq \sqrt{(2)}m/2$. Furthermore, since T_x is uniformly continuous, an $e > 0$ exists such that $T_x(x, y_j) > m/2$ for all $|x - x_i| < e$, and therefore $T_{i,j} > T_{i-1,j} > T_{i-2,j}$ when $h < e/2$. This implies the full second order stencil, $T_x(\vec{x}_{i,j}) = \frac{3}{2}T_{i,j} - 2T_{i-1,j} + \frac{1}{2}T_{i-2,j}$, is available. Examining the switch, $sw_{i,j}^{-y}$, and the second order fast marching discretization scheme, there are three cases.

Case 1: $T_{i,j-1}, T_{i,j+1} > T_{i,j}$. In this case, the second order fast marching method solves $|T_x| = s(x)$. Using Taylor series, we have $T_{i,j} < T_{i,j} + hT_y(x_{i,j}) + h^2T_{yy}(x_{i,j})/2 + O(h^3)$ and $T_{i,j} < T_{i,j} - hT_y(x_{i,j}) + h^2T_{yy}(x_{i,j})/2 + O(h^3)$. Combining these equations yields $|T_y(\vec{x}_{i,j}|$ is $O(h)$. Now, solving the eikonal equation for T_x and expanding the resulting square root using Taylor series once again yields $|T_x| = \sqrt{s^2 - T_y^2} = s + O(T_y^2)$. Since T_y is $O(h)$, $|T_x| = s(x)$ is a second order approximation to the eikonal equation.

Case 2: $T_{i,j-1}, T_{i,j} > T_{i,j+1}$ and $T_{i,j+2} > T_{i,j+1}$. In this case, the second order fast marching method uses a first order forward approximation for T_y. Using Taylor series, we have $T_{i,j+1} < T_{i,j+1} - hT_y(x_{i,j+1}) + h^2T_{yy}(x_{i,j+1})/2 + O(h^3)$ and $T_{i,j+1} <$

$T_{i,j+1} + hT_y(x_{i,j+1}) + h^2T_{yy}(x_{i,j+1})/2 + O(h^3)$. This implies $(T_y)_{i,j+1}$ is $O(h)$. Using Taylor series again, we have $(T_y)_{i,j}$=$(T_y)_{i,j+1} + O(h)$ and therefore $(T_y)_{i,j}$ is $O(h)$ and $|T_x| = \sqrt{s^2 - (\frac{T_{i,j+1}-T_{i,j}}{h})^2} = s + O(h^2)$ is a second order approximation to the eikonal equation.

Case 3: $T_{i,j+1}, T_{i,j} > T_{i,j-1}$ and $T_{i,j-2} > T_{i,j-1}$. In this case, the second order fast marching method uses a first order backward difference approximation for T_y. Using Taylor series, we have $T_{i,j-1} < T_{i,j-1} - hT_y(x_{i,j-1}) + h^2T_{yy}(x_{i,j-1})/2 + O(h^3)$ and $T_{i,j-1} < T_{i,j-1} + hT_y(x_{i,j-1}) + h^2T_{yy}(x_{i,j-1})/2 + O(h^3)$. This implies $(T_y)_{i,j-1}$ is $O(h)$. Using Taylor series again, we have $(T_y)_{i,j}$=$(T_y)_{i,j-1} + O(h)$, and therefore $(T_y)_{i,j}$ is $O(h)$ and $|T_x| = \sqrt{s^2 - (\frac{T_{i,j}-T_{i,j-1}}{h})^2} = s + O(h^2)$ is a second order approximation to the eikonal equation.

Case 4: $T_{i,j-1}, T_{i,j} > T_{i,j+1} > T_{i,j+2}$ or $T_{i,j+1}, T_{i,j} > T_{i,j-1} > T_{i,j-2}$ In this case, the second order fast marching method uses a second order approximation for T_y and the method is clearly second order. □

Remark. Only for sufficiently small h are we guaranteed to be using a second order difference approximation.

Remark. There are some degenerate cases where the solution is smooth and the gradient and therefore the slowness is zero; see section 5.2. In a small region near points where the slowness is zero only first order approximations for all the derivatives is possible. Even so, the method is still second order near zero slowness regions (for smooth T) because the error term for the first order method is proportional to the slowness.

Remark. This theorem requires the arrival times, $T(x, y)$, to be smooth. In a complex velocity field with smooth initial conditions, the arrival times, $T(x, y)$, can be guaranteed to be smooth only in the close vicinity of the source and eventually develops singularities with an overwhelming probability [26].

4. Extending the fast marching method to third order of accuracy.

Given the success of the second order accurate fast marching method, a simple approach to extending the method to third order of accuracy would be to use a third order one-sided difference approximation,

$$\begin{aligned}
(T_x)_{i,j} &= \max\left(\frac{11T_{i,j}}{6} - \frac{3T_{i+s_x,j} + 3T_{i+2s_x,j}}{2} - \frac{T_{i+3s_x,j}}{3}, 0\right)/h, \\
(T_y)_{i,j} &= \max\left(\frac{11T_{i,j}}{6} - \frac{3T_{i,j+s_y} + 3T_{i+2s_y,j}}{2} - \frac{T_{i,j+3s_y}}{3}, 0\right)/h, \\
s_x &= -sign(T_{i+1,j} - T_{i-1,j}), \\
s_y &= -sign(T_{i,j+1} - T_{i,j-1}),
\end{aligned}$$

to calculate the derivatives when the points are available. There are two problems with this approach. The first issue to be addressed is what to do if the points are not available to use third order one-sided difference schemes. This is not a large problem for a third order method because in smooth regions the points are not available for third order one-sided difference approximations only when the gradient is closely aligned to the grid, which is a lower-dimensional space. So, we can still expect to get third order accuracy in the L_1 norm for smooth problems if we simply use the second order fast marching scheme when there aren't sufficient points available for the third order stencil. This will limit us to second order accuracy in the L_∞ norm and prevent

Table 1
Instability of third order one-sided derivatives.

h	L_1 error	L_∞ error
.02	1.0819e-004	4.2367e-004
.01	3.6776e-005	9.2821e-005
.005	4.7501e-006	2.0567e-005
.0025	4.0655e-005	0.0089

the development of a fourth order scheme (in a later paper), so we will tackle this problem later in this section.

A much more serious problem is that using high order one-sided difference schemes is unstable. We demonstrate the instability on this simple test problem:

$$\begin{aligned}|\nabla T(\vec{x})| &= 1,\ \vec{x} \in [0,1]\times[0,1],\\ T(0,0) &= 0.\end{aligned}$$

For this problem we initialize the problem by using exact values at all points where $T < .3$ and using exact values 3 grid points deep along each edge. We note that with this setup the full third order stencils are available at every point where we calculate arrival times, and backward differences are always used. In Table 1 we tabulate the errors as the step size is decreased. For very large step sizes the method almost seems to be working properly, but as the step size is decreased instabilities develop and the error starts to grow. In Appendix A we show that using third order one-sided derivatives on the linearized eikonal equation does not satisfy the von Neumann condition for any step size.

To improve the stability of the third order scheme we consider a 2D derivative stencil with the form

$$(T_x)_{i,j} = \sum_{k=0}^{n}\sum_{l=0}^{m} A_{k,l} T_{i+s_x k, j+s_y l}/h, (T_y)_{i,j} = \sum_{k=0}^{n}\sum_{l=0}^{m} A_{l,k} T_{i+s_x l, j+s_y k}/h, \tag{6}$$

where $(s_x, s_y) = -(sign(T_x), sign(T_y))$ is the upwind direction. The goal is to choose $A_{k,l}$ so that the approximation is third order accurate and satisfies the von Neumann condition when applied to the linearized eikonal equation. One choice that satisfies these conditions under some mild restrictions (see Appendix B) is

$$\begin{aligned} &A_{0,0} = \frac{3}{2},\ A_{1,0} = -2,\ A_{2,0} = \frac{1}{2},\\ &A_{0,1} = \frac{1}{3},\ A_{1,1} = -1,\ A_{2,1} = 1,\ A_{3,1} = -\frac{1}{3},\\ &A_{0,2} = \frac{1}{9},\ A_{1,2} = -\frac{4}{9},\ A_{2,2} = \frac{2}{3},\ A_{3,2} = -\frac{4}{9},\ A_{4,2} = \frac{1}{9}.\end{aligned} \tag{7}$$

Notice that the derivative approximation for T_x depends on the signs of the components of the gradient, T_x and T_y, so there are now four possible derivative stencils for T_x. Notice also that the second and third rows of the approximation are a "correction" to the second order one-sided difference scheme, and this scheme for the gradient, ∇T, uses the same grid points as the second order scheme for the gradient, ∇T, at the locations (i, j), $(i + s_x, j + s_y)$, and $(i + 2 * s_x, j + 2 * s_y)$. Finally, we remark that while the 12 function evaluations in the derivative approximation might seem like a lot, it is not an unusually high number for this type of problem. For comparison,

Table 2
Third order method without rotating.

h	L_1 error	Order	L_∞ error	Order
.02	3.8608e-005		2.7236e-004	
.01	4.8944e-006	2.9797	6.7361e-005	2.0155
.005	6.2527e-007	2.9686	1.6699e-005	2.0122
.0025	7.9103e-008	2.9827	4.1542e-006	2.0071

a third order WENO approximation of a derivative [23, 28, 29] requires 11 function evaluations.

Now, to solve the eikonal equation using the 2D derivative stencil, (6), at (i,j) we first calculate $(s_x, s_y){=}{-}(sign(T_{i+1,j} - T_{i-1,j}), sign(T_{i,j+1} - T_{i,j-1}))$. Next we check to see if all the points are available to use the third order approximation, (6). These points are available if the second order stencil is available and the upwind directions are the same at locations (i,j), $(i{+}s_x, j{+}s_y)$, and $(i{+}2s_x, j{+}2s_y)$. If these conditions are satisfied, we solve $\max(T_x, 0)^2 + \max(T_y, 0)^2 = s^2h^2$, and if not, we use the second order fast marching method.

Once again, the astute reader may question whether this is really a third order method since at some locations only second or even first order differences will be used to solve the eikonal equation. There are two cases where the third order derivative approximations cannot be used. The first is at or near caustics, where only first order accuracy can be obtained. The second case is when the gradient is nearly aligned with the grid. In this case, because one component of the gradient is nearly zero there are not enough upwind points available to use a high order approximation for this component of the gradient. From the proof of Theorem 1, equations (4) and (5), which result from neglecting the small component of the gradient in the eikonal equation, are only second order accurate approximations of the eikonal equation. So, unfortunately, mimicking the second order method and neglecting the component of the gradient that is small will not lead to a third order accurate solution even if the arrival times, T, are smooth and third order derivative approximations are used for the remaining derivative. This is not a large problem for a third order method. The reason for this is that equations (4) and (5) are used only when the gradient is closely aligned to the grid, which is generally a lower-dimensional space. So, we can still expect to get third order accuracy in the L_1 norm, but not the L_∞ norm for smooth problems. We demonstrate this on the following simple test problem:

$$\begin{aligned} &|\nabla T(\vec{x}| = 1, \quad \vec{x} \in [-1,1] \times [-1,1], \\ &T(0,0) = 0. \end{aligned}$$

For this problem we initialize the problem by using exact values at all points where $T < .3$. The method will drop to second order 2 grid points to either side of the lines $x = 0$ and $y = 0$. We tabulate the L_1 and L_∞ errors for different step sizes in Table 2.

Remark. The above example is only intended to show that relying on equations (4), (5) when T_x or T_y is nearly zero cannot result in a third order scheme in the L_∞ norm. The stability region for the 2D scheme, (6), is not large enough for the method to be "stable," but the amplification factor is small enough so that the above method is "safe" for problems with fewer than 1,000 grid points per dimension; see Appendix B for details. Below we will describe a rotation scheme. We introduce this rotation to fix the causality problem that occurs when a component of the gradient is small. It will also fix the stability problem.

To develop a fully third order scheme, and pave the way for schemes of higher than third order, we now return to the problem of not having a full stencil when the gradient is nearly aligned with the grid, implying that one component of the gradient is nearly zero. Note that the gradient can be defined in two dimensions using any two perpendicular directions. The grid lines are usually chosen for convenience. Our strategy will be to locally rotate the gradient operator 45 degrees and take derivatives along the diagonals. The idea is that with the rotated gradient neither component of the gradient will be small and a full stencil will be available. With this in mind we define our rotated gradient as

$$\nabla^{45^\circ} T = (T_d, T_{d^\perp}), \tag{8}$$

where $T_d = \nabla T \cdot (\sqrt{2}/2, \sqrt{2}/2)$ and $T_{d^\perp} = \nabla T \cdot (\sqrt{2}/2, -\sqrt{2}/2)$. Now a first order upwind discretization of the rotated eikonal equation is simply

$$\begin{aligned}
(T_d)^2_{i,j} + (T_{d^\perp})^2_{i,j} &= s(x,y)^2,\\
(T_d)_{i,j} &= \max(-D^{+d}_{45^\circ} T_{i,j}, D^{-d}_{45^\circ} T_{i,j}, 0),\\
(T_{d^\perp})_{i,j} &= \max(-D^{+d^\perp}_{45^\circ} T_{i,j}, D^{-d^\perp}_{45^\circ} T_{i,j}, 0),\\
D^{+d}_{45^\circ} T_{i,j} &= \frac{T_{i+1,j+1} - T_{i,j}}{\sqrt{2}h},\\
D^{-d}_{45^\circ} T_{i,j} &= \frac{T_{i,j} - T_{i-1,j-1}}{\sqrt{2}h},\\
D^{+d^\perp}_{45^\circ} T_{i,j} &= \frac{T_{i+1,j-1} - T_{i,j}}{\sqrt{2}h},\\
D^{-d^\perp}_{45^\circ} T_{i,j} &= \frac{T_{i,j} - T_{i-1,j+1}}{\sqrt{2}h}.
\end{aligned}$$

Likewise, the second order upwind discretization of the rotated eikonal equation is

$$\begin{aligned}
&\max\left(-\left(D^{+d}_{45^\circ} T_{i,j} - \frac{\sqrt{2}h}{2} sw^{+d}_{i,j} (D^{+d}_{45^\circ})^2 T_{i,j}\right), D^{-d}_{45^\circ} T_{i,j} + \frac{\sqrt{2}h}{2} sw^{-d}_{i,j} (D^{-d}_{45^\circ})^2 T_{i,j}, 0\right)^2\\
&\quad + \max\left(-\left(D^{+d^\perp}_{45^\circ} T_{i,j} - \frac{\sqrt{2}h}{2} sw^{+d^\perp}_{i,j} (D^{+d^\perp}_{45^\circ})^2 T_{i,j}\right),\right.\\
&\qquad\qquad \left. D^{-d^\perp}_{45^\circ} T_{i,j} + \frac{\sqrt{2}h}{2} sw^{-d^\perp}_{i,j} (D^{-d^\perp}_{45^\circ})^2 T_{i,j}, 0\right)^2 = s^2_{i,j},
\end{aligned}$$

$$sw^{+d}_{i,j} = \begin{cases} 1 & \text{if } T_{i+1,j+1} \text{ and } T_{i+2,j+2} \text{ are known and } T_{i+2,j+2} < T_{i+1,j+1},\\ 0 & \text{otherwise,} \end{cases}$$

$$sw^{-d}_{i,j} = \begin{cases} 1 & \text{if } T_{i-1,j-1} \text{ and } T_{i-2,j-2} \text{ are known and } T_{i-2,j-2} < T_{i-1,j-1},\\ 0 & \text{otherwise,} \end{cases}$$

$$sw^{+d^\perp}_{i,j} = \begin{cases} 1 & \text{if } T_{i+1,j-1} \text{ and } T_{i+2,j-2} \text{ are known and } T_{i+2,j-2} < T_{i+1,j-1},\\ 0 & \text{otherwise,} \end{cases}$$

$$sw^{-d^\perp}_{i,j} = \begin{cases} 1 & \text{if } T_{i-1,j+1} \text{ and } T_{i-2,j+2} \text{ are known and } T_{i-1,j+1} < T_{i-2,j+2},\\ 0 & \text{otherwise.} \end{cases}$$

Finally, the third order approximation to the rotated eikonal equation is

$$(T_d)_{i,j} = \sum_{k=0}^{n}\sum_{l=0}^{m} A_{k,l} T_{i+d_x k+d_x^\perp k, j+d_y k+d_y^\perp l}/\sqrt{2}h, \tag{9}$$

$$(T_{d\perp})_{i,j} = \sum_{k=0}^{n}\sum_{l=0}^{m} A_{l,k} T_{i+d_x k+d_x^\perp k, j+d_y k+d_y^\perp l}/\sqrt{2}h,$$

where the upwind directions are now

$$(d_x, d_y) = -sign(T_{i+1,j+1} - T_{i-1,j-1})(1,1),$$
$$(d_x^\perp, d_y^\perp) = -sign(T_{i+1,j-1} - T_{i-1,j+1})(1,-1),$$

and the $A_{k,l}$ are the same as in (7).

Now we can obtain a fully third order scheme for the eikonal equation. First, we try to solve $\max(T_x,0)^2 + \max(T_y,0)^2 = f^2h^2$ using (6). If the points are not available, or if the computed gradient doesn't satisfy the stability condition (Appendix B), we then try to solve $\max(T_d,0)^2 + \max(T_{d^\perp},0)^2 = f^2h^2$ using (9). Then if the points are not available or the stability condition is violated, we use the second order fast marching method. In smooth regions the eight possible difference schemes are given in Appendix B (Table 14).

Once again, we return to the question of the order of the method. This method, like all methods, has to revert to first order near caustics, but the hope is that higher order is achieved in regions where T is smooth. To help answer this question we prove the following theorem.

THEOREM 2. *The third order fast marching method is globally third order, that is, third order in the max norm, if the solution, T, is smooth (has at least 4 continuous derivatives) on a rectangular 2D domain, Ω, and the slowness is bounded below, $s(x) > m$.*

Proof. Let $\vec{x}_{i,j} = (x_i, y_j)$ be a regularly spaced grid on Ω with constant step size h and let $T_{i,j} = T(\vec{x}_{i,j})$. Without loss of generality, assume $T_d(\vec{x}_{i,j}) > T_y(\vec{x}_{i,j}), T_x(\vec{x}_{i,j}) \geq 0 \geq T_{d\perp}(\vec{x}_{i,j})$. This implies that the gradient vector, $(T_x(\vec{x}_{i,j}), T_y(\vec{x}_{i,j}))$, lies in the first quadrant between the vectors $(\cos(22.5°), \sin(22.5°))$ and $(\cos(67.5°), \sin(67.5°))$. Let θ_1 and θ_2 be the angles between the gradient and the vectors $(1,0)$ and $(0,1)$, respectively. Using the lower bound on the slowness, we have $T_x(\vec{x}_{i,j}) = |\nabla T|\cos(\theta_1) \geq |\nabla T|\cos(67.5°) > m\cos(67.5°)$ and, likewise, $T_y(\vec{x}_{i,j}) = |\nabla T|\cos(\theta_2) \geq |\nabla T|\cos(67.5°) > m\cos(67.5°)$. Furthermore, since $T_x(T_y)$ is uniformly continuous, an $e > 0$ exists such that $T_x(x,y)(T_y(x,y)) > m\cos(67.5°)/2$ for all $|(x-x_i)^2 + (y-y_j)^2| < e$, and therefore $T_{i,j} > T_{i-1,j} > T_{i-2,j}$ $(T_{i,j} > T_{i,j-1} > T_{i,j-2})$ when $h < e/2$. Since we also have $T_x, T_y > m\cos(67.5°)/2$ at every grid point in the stencil, a similar argument establishes that $T_{i,j-1} > T_{i-1,j-1} > T_{i-2,j-1} > T_{i-3,j-1}$, that $T_{i,j-2} > T_{i-1,j-2} > T_{i-2,j-2} > T_{i-3,j-2} > T_{i-4,j-2}$, that $T_{i-1,j} > T_{i-1,j-1} > T_{i-1,j-2} > T_{i-1,j-3}$, and that $T_{i-2,j} > T_{i-2,j-1} > T_{i-2,j-2} > T_{i-2,j-3} > T_{i-2,j-4}$, when $h < e/5$. This implies the full third order stencil is available. □

Remark. This theorem proves that in smooth regions the full high order stencil is used for small enough h, but it does not prove that the method will revert to a first order scheme near caustics. This is typically very difficult to do. However, by construction the third order method always reverts to first order whenever the second order fast marching method does, and that method is well established to convert to first order in the presence of caustics [19]. We also note that the rotation scheme is needed for both causality and stability (see Appendix B).

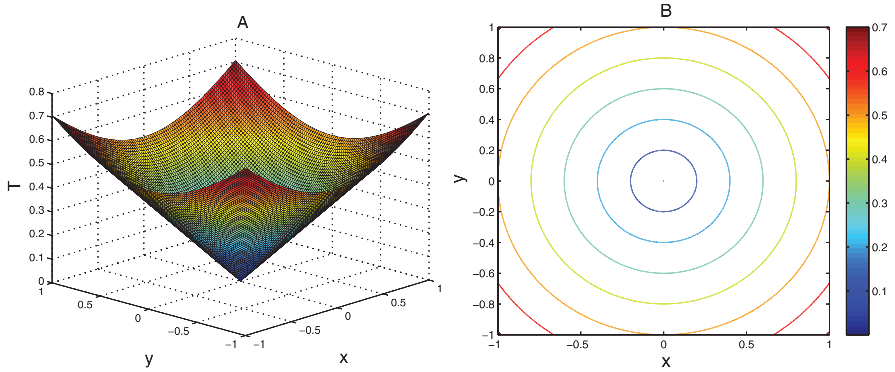

Fig. 1. *Contour plot (right) and 3D plot (left) of the calculated arrival times for a point source problem with $h = .0025$.*

Table 3
Error table for the point source problem using the third order fast marching method.

h	L_1 error	Order	L_∞ error	Order
.02	3.38e-005		1.088e-004	
.01	4.649e-006	2.9431	1.4147e-005	2.862
.005	5.9225e-007	3.0201	1.7439e-006	2.9726
.0025	7.4442e-008	3.0214	2.1477e-007	2.992

5. Examples.

In this section we test the third order accurate fast marching method on three test problems. We will not address the issues of initialization and boundary conditions in this paper. As is typically done for convergence tests, we will use the exact solution in a small region near the source and when required use the exact solution in a thin band at the edges of the computational domain.

5.1. Point source problem.

We first test our method on a classic point source problem,

$$|\nabla T(\vec{x}| = 1, \quad \vec{x} \in [-1,1] \times [-1,1],$$
$$T(0,0) = 0.$$

For this problem we initialize the problem by using exact values at all points where $T < .2$. The calculated arrival times with step size, $h = .0025$, are displayed in Figure 1. We tabulate the L_1 and L_∞ errors for different step sizes in Table 3. The L_1 errors are roughly the same as those for the method used in Table 2, but now the L_∞ errors clearly show third order convergence. We also tabulate the number of grid points where the solution is initialized, calculated using the first order scheme, calculated using the second order scheme, calculated using the third order scheme, and calculated using the third order rotated scheme in Table 4. For the largest step size, there are eight grid points near the point source where the points are not available to use a third order scheme. After the first grid refinement a third order scheme is used at every noninitialized grid point. Finally, we note that one should not expect the number of grid points using the third order scheme and the number of grid points using the third order rotated scheme to be the same. This is because some grid points can use both schemes, and the third order scheme without rotation is attempted first, and because the domain is a square and not a circle.

TABLE 4

Tabulation of the number of grid points where the solution is initialized (second row), calculated using the first order scheme (third row), calculated using the second order scheme (fourth row), calculated using the third order scheme (fifth row), and calculated using the third order rotated scheme (sixth row) when using the third order method on the point source problem.

h	Initialized	First order	Second order	Third order	Third order rotated
.02	325	0	8	6297	3370
.01	1289	0	0	25233	13478
.005	5105	0	0	101021	53874
.0025	20253	0	0	404313	215434

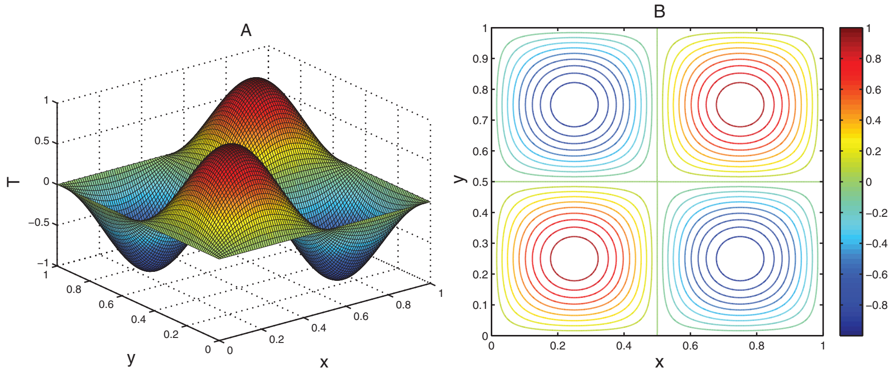

FIG. 2. *Contour plot (right) and 3D plot (left) of the calculated arrival times for the shape from shading problem with $h = .00125$.*

5.2. Shape from shading problem. Next, we test our method on a shape from shading problem,

$$|\nabla T(\vec{x}| = 2\pi\sqrt{\cos^2(2\pi x)\sin^2(2\pi y) + \cos^2(2\pi y)\sin^2(2\pi x)},\ \vec{x} \in [0,1]\times[0,1],$$

$$T(x,0) = T(0,y) = T\left(\frac{1}{2},\frac{1}{2}\right) = 0,\ T\left(\frac{1}{4},\frac{1}{4}\right) = 1,\ T\left(\frac{3}{4},\frac{3}{4}\right) = -1.$$

The exact solution to this problem is $T(x,y) = \sin(2\pi x)\sin(2\pi y)$. While the solution is smooth, this is a very difficult problem to solve to high order, and in particular the slowness does not have a lower bound as $s(1/2,1/2) = 0$. For this example we initialize the problem by using exact values at all points where $T < -.9$ and $T > .9$ and in a thin region around the edges. The calculated arrival times with step size, $h = .00125$, are displayed in Figure 2. We tabulate the L_1 and L_∞ errors for different step sizes in Table 5. The error table clearly shows third order convergence for this problem in both the L_1 and L_∞ norms. We also tabulate in Table 6 the number of grid points where the solution is initialized in the interior, initialized on the edge, calculated using the first order scheme, calculated using the second order scheme, calculated using the third order scheme, and calculated using the third order rotated scheme.

We note that the requirements for Theorem 2 are not satisfied by this problem because the slowness, s, does not have a positive lower bound. During the computations, a full third order stencil is *not* used at every point even after refining the grid. For example, at all of the grid points neighboring $(1/2, 1/2)$ only the first order method, (3), is used. This would suggest that only first order in the L_∞ norm is possible. However, the error term using equation (3) is proportional to the slowness

Table 5
Third order method tested on the shape from shading example.

h	L_1 error	Order	L_∞ error	Order
.01	2.7233e-005		.00147025	
.005	2.6245e-006	3.3753	.000182731	3.0083
.0025	2.6294e-007	3.3192	2.2751e-005	3.0055
.00125	2.8414e-008	3.2101	2.8444e-006	2.9999

Table 6
Tabulation of the number of grid points where the solution is initialized in the interior (second row), initialized on the edge (third row), calculated using the first order scheme (fourth row), calculated using the second order scheme (fifth row), calculated using the third order scheme (sixth row), and calculated using the third order rotated scheme (seventh row) when using the third order method on the shape from shading problem. Note that we have categorized a grid point as first order if either of the derivatives were calculated using a first order finite difference, or replaced with zero. This does not necessarily mean that the approximation is first order; see Theorem 1 *for details.*

h	Initialized (interior)	Initialized (edge)	First order	Second order	Third order	Third order rotated
.01	717	2632	29	211	2898	3313
.005	2734	5432	19	104	13730	18384
.0025	10574	11032	21	103	57474	81597
.00125	42150	22232	19	102	232266	344832

s. Since the slowness at grid points neighboring the point $(1/2, 1/2)$ is clearly $O(h)$, the local truncation error will be $O(h^3)$, and we should expect at least second order convergence in this region. Finally, we note that for this example the local truncation is only $O(h^3)$ at a *fixed* number of grid points as the mesh is refined, so third order convergence should be expected for this problem. This is similar to the use of a quadrature rule which is one order less at the end points of a numerical integration scheme.

Finally, we comment on the initialization for this problem. At the points $(1/4, 1/4)$, $(3/4, 3/4), (1/4, 3/4), (3/4, 1/4)$ the solution is not smooth. To achieve 3rd order accuracy a region of fixed size surrounding these points needs to be initialized. At the edges, the solution is smooth, but when the characteristic is pointing into the domain, the solution needs to be initialized at every grid point that has a computation domain of dependency that lies outside the computation domain. This depends on the size of the discretization stencil. Here we demonstrate that using the same initialization as for the second order fast marching method still results in a third order method. We recompute the error tables in Table 7 using the third order method while only initializing the outer two layers of grid points on the edge. A thin band (of fixed width in grid points) uses the second order fast marching method. The error table still clearly shows third order accuracy in both the L_1 and L_∞ norms with a reduced error constant. We again tabulate the number of grid points where the solution is initialized, calculated using the first order scheme, calculated using the second order scheme, calculated using the third order scheme, and calculated using the third order rotated scheme in Table 8.

5.3. Nonsmooth problem. Finally, we test our method on a nonsmooth problem,

$$|\nabla T(\vec{x}| = 1, \ \vec{x} \in [-1, 1] \times [-1, 1],$$
$$T(-1/2, -1/2) = T(1/2, 1/2) = 0.$$

Table 7

Third order method tested on the shape from shading example with second order fast marching initialization on the domain edges.

h	L_1 error	Order	L_∞ error	Order
.01	.00011032		2.7236e-004	
.005	1.1837e-005	3.2201	6.7361e-005	2.9501
.0025	1.4888e-006	2.9911	1.6699e-005	2.9623
.00125	1.8607e-007	3.0002	4.1542e-006	3.0747

Table 8

Tabulation of the number of grid points where the solution is initialized in the interior (second row), initialized on the edge (third row), calculated using the first order scheme (fourth row), calculated using the second order scheme (fifth row), calculated using the third order scheme (sixth row), and calculated using the third order rotated scheme (seventh row) when using the third order method on the shape from shading problem with the second order fast marching initialization at the domain edges. Note that we have categorized a grid point as first order if either of the derivatives were calculated using a first order finite difference, or replaced with zero. This does not necessarily mean that the approximation is first order; see Theorem 1 for details.

h	Initialized	Initialized	First order	Second order	Third order	Third order rotated
	(interior)	(edge)				
.01	717	792	104	1388	3300	3899
.005	2734	1592	105	2826	14210	18934
.0025	10574	3192	105	6013	57986	82931
.00125	42150	6392	105	12404	232794	347756

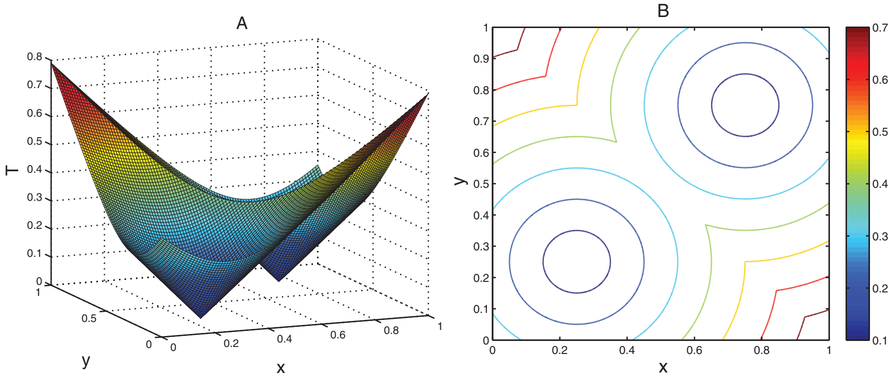

Fig. 3. *Contour plot (right) and 3D plot (left) of the calculated arrival times for a nonsmooth problem with $h = .00125$.*

The exact solution to this problem is $T = \min(\sqrt{(x-1/2)^2+(y-1/2)^2}, \sqrt{(x+1/2)^2+(y+1/2)^2}$. The solution is only Lipschitz continuous along the line $y = -x$, and the method will revert to first order in this region. For this problem we initialize the problem by using exact values at all points where $T < .2$. The calculated arrival times with step size, $h = .00125$, are displayed in Figure 3. We tabulate the L_1 and L_∞ errors for different step sizes in Table 9. The L_∞ convergence is now only first order. Because the first order region is codimension 1, the L_1 convergence is limited to second order, as shown in Table 9. We also tabulate in Table 10 the number of grid points where the solution is initialized, calculated using the first order scheme, calculated using the second order scheme, calculated using the third order scheme, and calculated using the third order rotated scheme. Finally, for comparison

TABLE 9

Third order method tested on a nonsmooth problem.

h	L_1 error	Order	L_∞ error	Order
.02	1.9480e-005		.00211706	
.01	4.0006e-006	2.2837	.00104132	1.0236
.005	8.7932e-007	2.1858	.00051375	1.0193
.0025	2.0405e-007	2.1075	.00025480	1.0117

TABLE 10

Tabulation of the number of grid points where the solution is initialized (second row), calculated using the first order scheme (third row), calculated using the second order scheme (fourth row), calculated using the third order scheme (fifth row), and calculated using the third order rotated scheme (sixth row) when using the third order method on a nonsmooth problem.

h	Initialized	First order	Second order	Third order	Third order rotated
.01	2610	26	540	2591	3258
.005	10244	36	1146	12529	14070
.0025	40604	36	2310	54697	58378
.00125	161602	36	4642	228179	237566

TABLE 11

First order method tested on a nonsmooth problem.

h	L_1 error	Order	L_∞ error	Order
.02	.00387101		.00946064	
.01	.00194126	.9957	.00475993	1.0190
.005	.00097584	.9923	.00239118	.9932
.0025	.000490236	.9932	.00120061	.9940

TABLE 12

Second order method tested on a nonsmooth problem.

h	L_1 error	Order	L_∞ error	Order
.02	3.5121e-04		.00647762	
.01	8.05714e-05	2.124	.00317493	1.0287
.005	1.96268e-05	2.0374	.00157880	1.0079
.0025	4.71312e-06	2.0581	.00078705	1.0043

we have also tabulated errors for the first and second order fast marching methods in Tables 11 and 12.

6. Conclusion.

In this paper we have developed a 2D third order accurate fast marching method. This method uses a stable one-sided 2D derivative approximation and a rotation strategy to ensure a full stencil is available for smooth problems. We have demonstrated the third order convergence on smooth problems, and lower order convergence without oscillations on nonsmooth problems. Like all fast marching methods, it is fast, requiring only about 2 local solves per grid point.

Finally, we give a few concluding remarks about the extension of this method to higher orders of accuracy and higher dimensions. For 2D problems, we hypothesize that higher order can be accomplished by developing higher order 2D difference schemes in the usual way, increasing the number of function evaluations and choosing coefficients that satisfy the order and stability conditions. This will be the subject of a future paper. Extending the method to higher-dimensions is more complicated and would require a higher-dimensional rotation strategy and higher-dimensional derivative approximations.

7. Appendix A.

In this section we demonstrate that the use of third order accurate one-sided differences yields an unstable algorithm when applied to the linearized eikonal equation. So, letting V be a small perturbation of a background arrival time, T, letting s_1 be a small perturbation from the background slowness, and denoting $s_0 = |\nabla T|$, we have

$$|\nabla(T+V)| = s_0 + s_1. \tag{10}$$

Using $s_0 = |\nabla T|$ and keeping only the first variation terms, this equation becomes

$$V_x T_x + V_y T_y = s_0 s_1. \tag{11}$$

In the first sample problem in section 4 the arrival times satisfy $T_x > T_y > 0$ everywhere in the computational domain, and we will assume the same here. In this situation the unstable third order fast marching method that uses one-sided differences uses one-sided backward differences for all computations, so we will check the von Neumann stability using third order backward differences for V_x and V_y in the homogeneous equation

$$V_x T_x + V_y T_y = 0. \tag{12}$$

Letting $V_x = \frac{11V_{i,j}}{6h} - \frac{3V_{i-1,j}}{h} + \frac{3V_{i-2,j}}{2h} - \frac{V_{i-3,j}}{3h}$ and $V_y = \frac{11V_{i,j}}{6h} - \frac{3V_{i,j-1}}{h} + \frac{3V_{i,j-2}}{2h} - \frac{V_{i,j-3}}{3h}$ and assuming $V_{i,j} = F(i)e^{-ikj} = F(i)z$ leads to the recurrence relation

$$\left(\frac{11}{6} + \frac{T_y}{T_x}\left(\frac{11}{6} - \frac{3}{z} + \frac{3}{2z^2} - \frac{1}{3z^3}\right)\right)F(i) - 3F(i-1) + \frac{3}{2}F(i-2) - \frac{1}{3}F(i-3) = 0. \tag{13}$$

The characteristic equation for this recurrence relation is found by setting $F(i) = w^i$ into the recurrence relation:

$$\left(\frac{11}{6} + \frac{T_y}{T_x}\left(\frac{11}{6} - \frac{3}{z} + \frac{3}{2z^2} - \frac{1}{3z^3}\right)\right)w^3 - 3w^2 + \frac{3}{2}w - \frac{1}{3} = 0. \tag{14}$$

The von Neumann condition is satisfied for a particular value of $\frac{T_y}{T_x}$ if all solutions, w, to the characteristic equation satisfy $|w| < 1$ for all $|z| \le 1$. We have investigated equation (14) numerically and found that there is no choice of $\frac{T_y}{T_x}$ that satisfies this condition with $T_x, T_y > 0$.

8. Appendix B.

In this section we demonstrate that the 2D one-sided differences are stable. First, we note that in a smooth region the first order accurate fast marching method, (3), uses one of eight difference schemes depending on the signs of the computed components of the gradient. These eight schemes are outlined in Table 13. Likewise, the third order fast marching method, (6) or (9), also uses one of eight difference schemes depending on the computed gradient. We define θ as the angle between the vector (T_x, T_y) and the x axis and outline the eight difference schemes in Table 14.

The goal now is to show that for each row in Table 14 the difference scheme given in the second column is stable when applied to the linearized eikonal equation, (12), when the gradient satisfies the corresponding condition given in the first column. We will first show that the first row in Table 14 satisfies linear stability. With this in mind, we substitute $V_x = \frac{3V_{i,j}}{2h} - \frac{2V_{i-1,j}}{h} + \frac{V_{i-2,j}}{2h} + \frac{V_{i,j-1}}{3h} - \frac{V_{i-1,j-1}}{h} + \frac{V_{i-2,j-1}}{h} - \frac{V_{i-3,j-1}}{3h} + \frac{V_{i,j-2}}{9h} - \frac{4V_{i-1,j-2}}{9h} + \frac{2V_{i-2,j-2}}{3h} - \frac{4V_{i-3,j-2}}{9h} + \frac{V_{i-4,j-2}}{9h}$, $V_y = \frac{3V_{i,j}}{2h} - \frac{2V_{i,j-1}}{h} + \frac{V_{i,j-2}}{2h} + \frac{V_{i-1,j}}{3h} -$

Table 13

The eight possible difference schemes using the first order fast marching method are given in this table. We say that $T_x > 0 (T_x < 0)$ if both forward and backward difference approximations for T_x are positive (negative). If the forward and backward approximations for T_x are either both zero or have opposite signs, then we say $T_x \approx 0$. We use the same convention for T_y.

T_x	T_y	Difference scheme for T_x	Difference scheme for T_y
>0	>0	backward difference	backward difference
>0	<0	backward difference	backward difference
<0	<0	forward difference	forward difference
<0	>0	forward difference	backward difference
>0	≈ 0	backward difference	none (set $T_y = 0$)
<0	≈ 0	forward difference	none (set $T_y = 0$)
≈ 0	<0	none (set $T_x = 0$)	forward difference
≈ 0	>0	none (set $T_x = 0$)	backward difference

Table 14

The eight possible difference schemes using the third order fast marching method are given in this table. We define θ as the angle between the vector (T_x, T_y) and the x axis.

θ	Difference scheme
$22.5 < \theta < 67.5$	(6) with $(s_x, s_y) = (-1, -1)$
$67.5 < \theta < 112.5$	(9) with $(d_x, d_y) = (-1, -1)$
$112.5 < \theta < 157.5$	(6) with $(s_x, s_y) = (1, -1)$
$157.5 < \theta < 202.5$	(9) with $(d_x, d_y) = (1, -1)$
$202.5 < \theta < 247.5$	(6) with $(s_x, s_y) = (1, 1)$
$247.5 < \theta < 292.5$	(9) with $(d_x, d_y) = (1, 1)$
$292.5 < \theta < 337.5$	(6) with $(s_x, s_y) = (-1, 1)$
$-22.5 < \theta < 22.5$	(9) with $(d_x, d_y) = (-1, 1)$

$\frac{V_{i-1,j-1}}{h} + \frac{V_{i-1,j-2}}{h} - \frac{V_{i-1,j-3}}{3h} + \frac{V_{i-2,j}}{9h} - \frac{4V_{i-2,j-1}}{9h} + \frac{2V_{i-2,j-2}}{3h} - \frac{4V_{i-2,j-3}}{9h} + \frac{V_{i-2,j-4}}{9h}$, and $V_{i,j} = F(i)e^{-ikj} = F(i)z$ into (12), which leads to the recurrence relation

$$
\begin{aligned}
(15) \qquad & \left(\frac{3}{2} + \frac{1}{3z} + \frac{1}{9z^2} + \frac{T_y}{T_x}\left(\frac{3}{2} - \frac{2}{z} + \frac{1}{2z^2}\right)\right) F(i) \\
& + \left(-2 - \frac{1}{z} - \frac{4}{9z^2} + \frac{T_y}{T_x}\left(\frac{1}{3} - \frac{1}{z} + \frac{1}{z^2} - \frac{1}{3z^3}\right)\right) F(i-1) \\
& + \left(\frac{1}{2} + \frac{1}{z} + \frac{2}{3z^2} + \frac{T_y}{T_x}\left(\frac{1}{9} - \frac{4}{9z} + \frac{2}{3z^2} - \frac{4}{9z^3} + \frac{1}{9z^4}\right)\right) F(i-2) \\
& - \left(\frac{1}{3z} + \frac{4}{9z^2}\right) F(i-3) + \frac{1}{9z^2} F(i-4) = 0.
\end{aligned}
$$

The characteristic equation for this recurrence relation is found by setting $F(i) = w^i$ into the recurrence relation:

$$
\begin{aligned}
(16) \qquad & \left(\frac{3}{2} + \frac{1}{3z} + \frac{1}{9z^2} + \frac{T_y}{T_x}\left(\frac{3}{2} - \frac{2}{z} + \frac{1}{2z^2}\right)\right) w^4 \\
& + \left(-2 - \frac{1}{z} - \frac{4}{9z^2} + \frac{T_y}{T_x}\left(\frac{1}{3} - \frac{1}{z} + \frac{1}{z^2} - \frac{1}{3z^3}\right)\right) w^3 \\
& + \left(\frac{1}{2} + \frac{1}{z} + \frac{2}{3z^2} + \frac{T_y}{T_x}\left(\frac{1}{9} - \frac{4}{9z} + \frac{2}{3z^2} - \frac{4}{9z^3} + \frac{1}{9z^4}\right)\right) w^2 \\
& - \left(\frac{1}{3z} + \frac{4}{9z^2}\right) w + \frac{1}{9z^2} = 0.
\end{aligned}
$$

The von Neumann condition is satisfied for a particular value of T_y/T_x if all solutions, w, to the characteristic equation satisfy $|w| < 1$ for all $|z| <= 1$. We have investigated equation (16) numerically and found that this condition is satisfied when $.35 < T_y/T_x < \frac{1}{.35}$, or $\mathrm{atan}(.35) \approx 19.3^\circ < \theta < 70.7^\circ \approx \mathrm{atan}(1/.35)$. Finally, we note that (1) the von Neumann stability condition obtained for the second row by linearizing the rotated eikonal equation $|\nabla^{45^\circ} T = s(x)|$ is equation (16) with T_y/T_x replaced by $T_{d^\perp}/T_d$; and (2) the approximations for T_x and T_y, as well as the approximations for T_d and $T_{d^\perp}$, are symmetric in equations (3) and (9). Therefore if any entry in Table 14 satisfies linear stability, then all entries will. The combined stability region from all 8 gradient operators proves linear stability for all θ.

Finally we make a few comments about the results in Table 2. Note that without the rotation (as the errors in Table 2 were generated) the method is not stable. However, the greatest amplification factor, $w_{\max}$, is still small, $w_{\max} \leq 1.007$. We expect errors to grow linearly, which is why the local truncation error is always one order lower than the global error. A rough back-of-the-envelope calculation, say by solving the equation $w_{\max}^n = n$, suggests that not using the rotation is "safe" if the number of points per dimension is less than about 1,000.

REFERENCES

[1] D. Adalsteinsson and J. A. Sethian, *A fast level set method for propagating interfaces*, J. Comput. Phys., 118 (1995), pp. 269–277.

[2] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Springer, New York, 1997.

[3] S. Bak, J. McLaughlin, and D. Renzi, *Some improvements for the fast sweeping method*, SIAM J. Sci. Comput., 32 (2010), pp. 2853–2874.

[4] M. Boué and P. Dupuis, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic costs in the control*, SIAM J. Numer. Anal., 36 (1999), pp. 667–695.

[5] B. Cockburn, C. Johnson, C.-W. Shu, and E. Tadmor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, A. Quarteroni, ed., Lecture Notes in Math. 1697, Springer, Berlin, 1998.

[6] M. G. Crandall and P. L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc., 277 (1983), pp. 1–42.

[7] C. Y. Kao, S. J. Osher, and J. Qian, *Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations*, J. Comput. Phys., 196 (2004), pp. 367–391.

[8] C.-Y. Kao, S. Osher, and Y.-H. Tsai, *Fast sweeping methods for static Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 42 (2005), pp. 2612–2632.

[9] S. Kim, *An $\mathcal{O}(N)$ level set method for eikonal equations*, SIAM J. Sci. Comput., 22 (2001), pp. 2178–2193.

[10] R. Kimmel and J. A. Sethian, *Fast marching methods on triangulated domains*, Proc. Natl. Acad. Sci., 95 (1998), pp. 8341–8435.

[11] F. Li, C. W. Shu, Y. T. Zhang, and H. Zhao, *A second order discontinuous Galerkin fast sweeping method for Eikonal equations*, J. Comput. Phys., 227 (2008), pp. 8191–8208. SIAM J. Sci. Comput., 31 (2007), pp. 237–271.

[12] S. J. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, 2002.

[13] J. Qian, Y.-T. Zhang, and H.-K. Zhao, *Fast sweeping methods for eikonal equations on triangular meshes*, SIAM J. Numer. Anal., 45 (2007), pp. 83–107.

[14] J. Qian, Y. T. Zhang, and H. Zhao, *A fast sweeping method for static convex Hamilton–Jacobi equations*, J. Sci. Comput., 31 (2007), pp. 237–271.

[15] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys., 155 (1999), pp. 410–438.

[16] S. Osher, *A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations*, SIAM J. Math. Anal., 24 (1993), pp. 1145–1152.

[17] S. Osher and J. A. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation*, J. Comput. Phys., 79 (1988), pp. 12–49.

[18] J. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci., 93 (1995), pp. 1591–1595.

[19] J. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed., Cambridge University Press, Cambridge, UK, 1999.

[20] J. A. Sethian, *Fast marching methods*, SIAM Rev., 41 (1999), pp. 199–235.

[21] J. A. Sethian and A. Vladimirsky, *Fast methods for the eikonal and related Hamilton-Jacobi equations on unstructured meshes*, Proc. Natl. Acad. Sci., 97 (2000), pp. 5699–5703.

[22] J. A. Sethian and A. Vladimirsky, *Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), pp. 325–363.

[23] C. W. Shu, *High order numerical methods for time dependent Hamilton-Jacobi equations*, in Mathematics and Computation in Imaging Science and Information Processing, Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap. 11, World Scientific, Hackensack, NJ, 2007, pp. 47–91.

[24] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao, *Fast sweeping algorithms for a class of Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 41 (2003), pp. 673–694.

[25] J. N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Automat. Control, 40 (1995), pp. 1528–1538.

[26] B. S. White, *The stochastic caustic*, SIAM J. Appl. Math., 44 (1984), pp. 127–149.

[27] L. Yatziv, A. Bartesaghi, and G. Sapiro, *O(N) implementation of the fast marching algorithm*, J. Comput. Phys., 212 (2005), pp. 393–399.

[28] Y. Zhang, H. Zhao, and S. Chen, *Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations*, Methods Appl. Anal., 13 (2006), pp. 299–320.

[29] Y. T. Zhang, H. Zhao, and J. Qian, *High order fast sweeping methods for static Hamilton–Jacobi equations*, J. Sci. Comput., 29 (2006), pp. 25–56.

[30] H. Zhao, *A fast sweeping method for Eikonal equations*, Math. Comp., 74 (2004), pp. 603–627.