Finite difference methods for approximating Heaviside functions

John D. Towers [a]

[a] *MiraCosta College, 3333 Manchester Avenue, Cardiff-by-the-Sea, CA 92007-1516, USA.*

Abstract

We present a finite difference method for discretizing a Heaviside function $H(u(\vec{x}))$, where u is a level set function $u : \mathbb{R}^n \mapsto \mathbb{R}$ that is positive on a bounded region $\Omega \subset \mathbb{R}^n$. There are two variants of our algorithm, both of which are adapted from finite difference methods that we proposed for discretizing delta functions in [18–20]. We consider our approximate Heaviside functions primarily as they are used to approximate integrals over Ω. We prove that our first approximate Heaviside function leads to second order accurate quadrature algorithms. Numerical experiments verify this second order accuracy. For our second algorithm, numerical experiments indicate third order accuracy if the integrand and the boundary of the region of integration are sufficiently smooth.

As a secondary application, we also describe the results of two numerical experiments that demonstrate the use of our approximate Heaviside functions for solving time dependent partial differential equations with discontinuous solutions. We use the Fourier method, which can result in spurious oscillations near discontinuities. By using our approximate Heaviside functions, we reduce these oscillations considerably, improving the global accuracy of the solution.

Key words: heaviside function, level set method, quadrature, irregular region, finite difference, regular grid, convergence rate, fourier method, discontinuous solution

Email address: john.towers@cox.net (John D. Towers).
URL: http://www.miracosta.edu/home/jtowers/ (John D. Towers).

1 Introduction

We are primarily concerned with the problem of approximating the integral

$$\mathcal{I} := \int_{\Omega} f(\vec{x})\, d\vec{x} \tag{1}$$

where $\vec{x} = (x^1, \ldots, x^n) \in \mathbb{R}^n$, and $\Omega = \{\vec{x} \in \mathbb{R}^n : u(\vec{x}) > 0)\}$. We assume that the zero level set $\partial\Omega = \{\vec{x} \in \mathbb{R}^n : u(\vec{x}) = 0\}$ is a compact manifold of codimension one defined by the zero level set of a function $u(\vec{x})$. The data f and u are only defined at the discrete set of mesh points of a regular grid. This problem arises frequently in applications of the level set method [12], [13], [15]. In this context it is common practice to write the integral $\mathcal{I}$ as

$$\mathcal{I} = \int_{\mathbb{R}^n} H(u(\vec{x}))f(\vec{x})\, d\vec{x}, \tag{2}$$

where $H(\cdot)$ denotes the Heaviside function

$$H(z) = \begin{cases} 0, & z < 0 \\ 1, & z > 0. \end{cases} \tag{3}$$

When viewed in this manner, the problem of approximating the integral $\mathcal{I}$ boils down to producing a discrete approximation of the Heaviside function $H(u(\vec{x}))$.

The problem of quadrature over an irregular region $\Omega \subset \mathbb{R}^n$, while not as much studied as quadrature on an interval of $\mathbb{R}^1$, is classical and can be found in textbooks, eg., [2], [3]. The problem considered here has some constraints not generally imposed in classical treatments, primarily that the boundary is defined by a level set, and secondarily that the data are only given on a regular grid.

Even for quadrature problems not related to the level set method, the new methods that we describe below may be useful for obtaining moderately accurate results with minimal problem-dependent setup. This is based on the fact that if u_1 and u_2 are level set functions for a pair of regions Ω_1, $\Omega_2 \subset \mathbb{R}^n$, then $\min(u_1, u_2)$ is a level set function for the set $\Omega_1 \cap \Omega_1$ and $\max(u_1, u_2)$ is a level set function for the set $\Omega_1 \cup \Omega_1$. This makes it easy to construct level set functions for fairly complicated sets Ω.

Let $\left\{\vec{x}_{\mathbf{k}} = \left(x^1_{k_1}, \ldots x^n_{k_n}\right) | \mathbf{k} := (k_1, \ldots, k_n) \in \mathbb{Z}^n\right\}$ denote the set of mesh points of the regular grid. We assume that the mesh spacing h is the same in all directions, $x^i_{k_i} = k_i h$, $k_i \in \mathbb{Z}$. Clearly the most straightforward approximation of $\mathcal{I}$ is

$$\mathcal{I} \approx h^n \sum_{\mathbf{k} \in \mathbb{Z}^n} H(u(\vec{x}_{\mathbf{k}}))f(\vec{x}_{\mathbf{k}}). \tag{4}$$

This method yields convergent approximations, but is only first order accurate.

One can also use a regularized Heaviside function H^ϵ, approximating $\mathcal{I}$ via

$$\mathcal{I} \approx h^n \sum_{\mathbf{k}\in\mathbb{Z}^n} H^\epsilon(u(\vec{x}_\mathbf{k}))f(\vec{x}_\mathbf{k}). \tag{5}$$

A regularized version of the Heaviside function that is often used in level set applications is $H^\epsilon = H^{C,\epsilon}$, where

$$H^{C,\epsilon}(z) = \begin{cases} H(z), & |z| \geq \epsilon \\ \frac{1}{2} + \frac{z}{2\epsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi z}{\epsilon}\right), & |z| \leq \epsilon \end{cases}, \quad \epsilon = O(h). \tag{6}$$

Another commonly used approximate Heaviside function is

$$H^{L,\epsilon}(z) = \begin{cases} H(z), & |z| \geq \epsilon \\ \frac{1}{2\epsilon}\left(z+\epsilon\right), & |z| \leq \epsilon \end{cases}, \qquad \epsilon = O(h). \tag{7}$$

Engquist, Tornberg and Tsai [6] gave a numerical example where the method (5) using $H^{L,\epsilon}$ defined in (7) gives only first order convergence. They proposed replacing the constant $\epsilon = h/2$ with a version $\epsilon = \epsilon(h, \nabla u)$ that takes the gradient of the level set function u into account, and reported second order convergence.

Approximating the integral (2) can be viewed as the problem of quadrature of a discontinuous function. This problem has been studied by Tornberg [16]. Tornberg proposed regularizing the Heaviside function, and then applying a standard quadrature technique to the resulting smooth integrand. This approach allows one to analyze separately the error contributions from regularization and quadrature.

Tornberg and Engquist [17] proposed a method of regularizing the characteristic function of a region Ω that is based on integrating a product of one-dimensional smeared out delta functions, the region of integration being Ω. They proved that it is possible to construct algorithms with any desired order of accuracy (as measured by approximating the integral $\mathcal{I}$), depending on the order of the one-dimensional delta functions that are used. These algorithms also have the desirable property that away from the boundary $\partial\Omega$, the discretized version of the characteristic function is the same as the exact one.

Min and Gibou [11] have also proposed a method for approximating the integral $\mathcal{I}$. Rather than discretizing Heaviside functions, their method involves decomposition of the region into simplices. A numerical quadrature rule is then applied on the simplices, and the resulting approximation to $\mathcal{I}$ is second order accurate. An advantage of their method is that it is not sensitive to the placement of the region Ω with respect to the mesh.

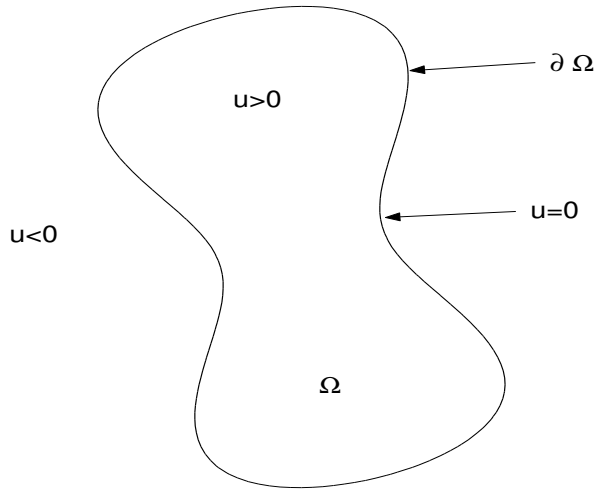

Fig. 1. The region of integration Ω defined by the level set $u = 0$.

The approach in the present paper is based on the idea of approximating the Heaviside function by finite differencing its first few primitives. We used this technique to approximate delta functions in [18–20].

We assume that u is defined and positive on Ω, and that Ω is bounded. See Figure 1. In addition, we assume that for some $\alpha > 0$, u is defined and smooth on a band of the form $B_\alpha = \{\vec{x} : |u(\vec{x})| < \alpha\}$ surrounding $\partial\Omega$. We further assume that f is also defined and smooth on $\Omega \bigcup B_\alpha$, and that for some $\sigma > 0$, $|\nabla u| > \sigma$ for $\vec{x} \in B_\alpha$. Note that since $u > 0$ in Ω, the unit outward (from Ω) normal vector $\vec{n}$ satisfies $\vec{n} = -\nabla u / |\nabla u|$.

Let $\{\vec{e}_1, \ldots, \vec{e}_n\}$ be the standard basis for $\mathbb{R}^n$. If $v_{\mathbf{k}} = v(\vec{x}_{\mathbf{k}})$ is a function defined at each grid point $\vec{x}_{\mathbf{k}}$, we define the second order accurate discrete gradient operator $\nabla^{2,h}$ via

$$\nabla^{2,h} v_{\mathbf{k}} = \sum_{m=1}^{n} \left(\frac{v(\vec{x}_{\mathbf{k}} + h\vec{e}_m) - v(\vec{x}_{\mathbf{k}} - h\vec{e}_m)}{2h} \right) \vec{e}_m, \tag{8}$$

and the fourth order discrete gradient:

$$\nabla^{4,h} v_{\mathbf{k}} = \frac{4}{3} \nabla^{2,h} v_{\mathbf{k}} - \frac{1}{3} \sum_{m=1}^{n} \left(\frac{v(\vec{x}_{\mathbf{k}} + 2h\vec{e}_m) - v(\vec{x}_{\mathbf{k}} - 2h\vec{e}_m)}{4h} \right) \vec{e}_m. \tag{9}$$

We will need the first two primitives of $H(z)$:

$$I(z) = \int_0^z H(\zeta)\, d\zeta, \qquad J(z) = \int_0^z I(\zeta)\, d\zeta. \tag{10}$$

To derive our approximations to $H(\cdot)$, we start from the relationships

$$\begin{aligned} \nabla J(u(\vec{x})) &= I(u(\vec{x}))\nabla u(\vec{x}), \\ \nabla I(u(\vec{x})) &= H(u(\vec{x}))\nabla u(\vec{x}), \end{aligned} \tag{11}$$

then take the inner product with ∇u, and next solve for H and I. This yields the relationships

$$\begin{aligned} I(u) &= \nabla J(u) \cdot \nabla u / \left|\nabla u\right|^2, \\ H(u) &= \nabla I(u) \cdot \nabla u / \left|\nabla u\right|^2. \end{aligned} \tag{12}$$

The final step in deriving our approximations is to discretize (12). Before doing so, note that these expressions are undefined wherever ∇u vanishes. This is not really a problem, since our approximate Heaviside functions will only differ from the exact Heaviside function in a narrow band surrounding $\partial\Omega$, where $|\nabla u| > 0$.

Let $\mathcal{N}_\nu$ denote the set of grid points $\vec{x}_{\mathbf{k}}$ which are separated from the interface $\partial\Omega$ by ν mesh widths or less:

$$\vec{x}_{\mathbf{k}} \in \mathcal{N}_\nu \iff u(\vec{x}_{\mathbf{k}})u(\vec{x}_{\mathbf{k}} \pm \nu h \mathbf{e}_m) \leq 0, \quad \text{for some } m \in \{1, \ldots, n\}. \tag{13}$$

By discretizing just the second relationship in (12), we get the one-step algorithm FDMH$_1$.

FDMH$_1$:

$$H_{\mathbf{k}}^{1,h} = \begin{cases} H(u_{\mathbf{k}}), & \vec{x}_{\mathbf{k}} \notin \mathcal{N}_1 \\ \nabla^{2,h} I(u_{\mathbf{k}}) \cdot \nabla^{2,h} u_{\mathbf{k}} / \left|\nabla^{2,h} u_{\mathbf{k}}\right|^2, & \vec{x}_{\mathbf{k}} \in \mathcal{N}_1 \end{cases} \tag{14}$$

FDMH$_1$ requires that u be smooth for all points $\vec{x}_{\mathbf{k}} \in \mathcal{N}_2$, basically a band two grid points wide on each side of the interface. Note that the band $\mathcal{N}_2$ where we require that u be smooth is slightly wider than the band $\mathcal{N}_1$ where the formula for $H_{\mathbf{k}}^{1,h}$ is nontrivial. This is due to the stencil of the discrete gradient $\nabla^{2,h}$. For points $\vec{x}_{\mathbf{k}}$ outside of $\mathcal{N}_2$, we only need to know the sign of $u_{\mathbf{k}}$.

By discretizing both relationships in (12), we get the two-step algorithm FDMH$_2$.

FDMH$_2$:

$$\begin{aligned} I_{\mathbf{k}}^{2,h} &= \begin{cases} I(u_{\mathbf{k}}), & \vec{x}_{\mathbf{k}} \notin \mathcal{N}_4 \\ \nabla^{4,h} J(u_{\mathbf{k}}) \cdot \nabla^{4,h} u_{\mathbf{k}} / \left|\nabla^{4,h} u_{\mathbf{k}}\right|^2, & \vec{x}_{\mathbf{k}} \in \mathcal{N}_4 \end{cases} \\ H_{\mathbf{k}}^{2,h} &= \begin{cases} H(u_{\mathbf{k}}), & \vec{x}_{\mathbf{k}} \notin \mathcal{N}_4 \\ \nabla^{4,h} I_{\mathbf{k}}^{2,h} \cdot \nabla^{4,h} u_{\mathbf{k}} / \left|\nabla^{4,h} u_{\mathbf{k}}\right|^2, & \vec{x}_{\mathbf{k}} \in \mathcal{N}_4 \end{cases} \end{aligned} \tag{15}$$

Note that for FDMH$_2$ we are using the fourth order discrete gradient. FDMH$_2$

requires that u be smooth for all points $\vec{x}_{\mathbf{k}} \in \mathcal{N}_6$, a band that is basically six grid points wide on each side of the interface.

Once we have computed the approximate Heaviside function $H_{\mathbf{k}}^{q,h}$ ($q = 1$ or 2), we approximate the integral (16) via

$$\mathcal{I}^{q,h} = h^n \sum_{\mathbf{k} \in \mathbb{Z}^n} H_{\mathbf{k}}^{q,h} f(\vec{x}_{\mathbf{k}}). \tag{16}$$

For our convergence theory, we require that the level set function u be smooth in the narrow band B_α, of width $O(1)$, near $\partial\Omega$. Outside of this narrow band, there are no regularity requirements; we only need u to be positive inside Ω and negative outside Ω. This is important in level set applications, where u is often a signed distance function, and is likely to have kinks at some finite distance from $\partial\Omega$. In practice, we only need to compute the nontrivial version of u in an $O(h)$ band around $\partial\Omega$; outside of that band only the sign of u is used. This is also important in level set applications; the computational cost can be reduced greatly by computing u only in an $O(h)$ band [1], [14].

We require for our convergence theory that the integrand f be defined and smooth not only inside Ω, but also in a narrow band, $B_\alpha \setminus \Omega$, of width $O(1)$ outside of Ω. In many level set applications, $f \equiv 1$. In these cases extending f is trivial. In applications where f is not constant, and is only known inside of Ω, we require that it is possible to extend it smoothly onto a narrow band of width $O(1)$ outside of Ω. For our algorithms, one would only need to compute this extension of f in an $O(h)$ band.

In Section 2, we prove that the algorithm FDMH_1 converges at a rate of $O(h^2)$ for sufficiently smooth data. In Section 3, we verify this rate of convergence via numerical examples. We also provide numerical examples indicating that if the data is smooth enough, FDMH_2 converges at a rate of $O(h^3)$. Our numerical experiments indicate that both methods degrade gracefully with reduced regularity, and continue to give usable results. More specifically, if u or f has jumps in the first derivatives, numerical experiments indicate $O(h^2)$ convergence for both methods.

Most of our examples in Section 3 are quadrature problems, but the last two briefly demonstrate that our approximate Heaviside functions can be effective when solving time dependendent partial differential equations with discontinuous solutions. We use the Fourier method for these examples. The Fourier method, like other pseudo-spectral methods, is extremely accurate if the solution is smooth. However, in the presence of discontinuities, spurious oscillations (the Gibbs phenomenon) can greatly reduce the accuracy globally (not just at the discontinuities). The two very simple examples that we consider indicate that our finite difference approach for computing Heaviside functions

(and delta functions) is a promising method for approximating singularities in this context. Specifically, the computed solutions to the PDEs display fairly limited spurious oscillation. Moreover, very high order accuracy can be recovered (away from the discontinuities) by a smoothing process described in [10]. This smoothing process increases the amplitude of unwanted oscillation slightly, but only in a very small region near the discontinuities.

2 A convergence proof for $\mathbf{FDMH}_1$

In this section, we show that under suitable regularity assumptions, the approximation $\mathcal{I}^{1,h}$ converges to $\mathcal{I}$ at a rate of $O(h^2)$. Let μ be a C^∞ function $\mu : \mathbb{R} \mapsto [0,1]$ such that $\mu(r) = 1$ for $|r| < \alpha/4$ and $\mu(r) = 0$ for $|r| \geq \alpha/2$. Let $\rho(\vec{x}) = \mu(u(\vec{x}))$, and define

$$\hat{f}(\vec{x}) = \rho(\vec{x})f(\vec{x}), \qquad \check{f} = (1 - \rho(\vec{x}))f(\vec{x}). \tag{17}$$

Note that both $\hat{f}$ and $H(u)\check{f}$ are as smooth as f, and that both $\hat{f}$ and $H(u)\check{f}$ have compact support. Moreover $\operatorname{supp}\left(H(u)\check{f}\right) \subset \Omega$, and for h sufficiently small, we will have

$$H^{1,h}_{\mathbf{k}}\check{f}(\vec{x}_{\mathbf{k}}) = H(u_{\mathbf{k}})\check{f}(\vec{x}_{\mathbf{k}}). \tag{18}$$

One more observation that will prove useful is that for $\vec{x}_{\mathbf{k}} \in B_{\alpha/2}$, and for h small enough,

$$H^{1,h}_{\mathbf{k}} = \nabla^{2,h} I(u_{\mathbf{k}}) \cdot \nabla^{2,h} u_{\mathbf{k}} / \left|\nabla^{2,h} u_{\mathbf{k}}\right|^2. \tag{19}$$

This results from (14) and the fact that the formula on the right side reduces to $H(u_{\mathbf{k}})$ for $\mathbf{k} \notin \mathcal{N}_1$ in the region where u is smooth.

For our analysis, we will use the following decomposition of $\mathcal{I}$:

$$\mathcal{I} = \underbrace{\int_{\mathbb{R}^n} H(u(\vec{x}))\hat{f}(\vec{x})\,d\vec{x}}_{=:P} + \underbrace{\int_{\mathbb{R}^n} H(u(\vec{x}))\check{f}(\vec{x})\,d\vec{x}}_{=:Q}. \tag{20}$$

The first of these integrals, P, contains a discontinuous integrand. This is where our method of discretizing $H(u(\vec{x}))$ comes into play. On the other hand, the second integral, Q, involves a smooth integrand with compact support, and can be approximated accurately without employing any special processing for the $H(u(\vec{x}))$ term.

Lemma 2.1 *Let*

$$\mathcal{F}(\vec{x}) = -\nabla \cdot \left(\hat{f}(\vec{x})\nabla u / \left|\nabla u\right|^2\right). \tag{21}$$

If $f \in C^1(B_\alpha)$, $u \in C^2(B_\alpha)$, then

$$\int_{\mathbb{R}^n} I(u)\mathcal{F}(\vec{x})\,d\vec{x} = P. \tag{22}$$

PROOF. We integrate by parts:

$$\begin{aligned}\int_{\mathbb{R}^n} I(u)\mathcal{F}(\vec{x})\,d\vec{x} &= -\int_{\mathbb{R}^n} I(u)\nabla\cdot\left(\hat{f}(\vec{x})\nabla u/\left|\nabla u\right|^2\right)\,d\vec{x}\\ &= -\int_{\mathbb{R}^n} \nabla\cdot\left(I(u)\hat{f}(\vec{x})\nabla u/\left|\nabla u\right|^2\right)\,d\vec{x}\\ &\quad+\int_{\mathbb{R}^n} \nabla I(u)\cdot\left(\hat{f}(\vec{x})\nabla u/\left|\nabla u\right|^2\right)\,d\vec{x}\\ &= \int_{\mathbb{R}^n} H(u)\nabla u\cdot\left(\hat{f}(\vec{x})\nabla u/\left|\nabla u\right|^2\right)\,d\vec{x}\\ &= \int_{\mathbb{R}^n} H(u)\hat{f}(\vec{x})\,d\vec{x}.\end{aligned} \tag{23}$$

Here we have used the fact that $\hat{f}$ has compact support to conclude that the integral of the form $\int \nabla\cdot(\cdots)\,d\vec{x}$ vanishes.

Before stating our convergence theorem, note that if $\vec{w}_{\mathbf{k}}$ vanishes for $\max\{|k_1|,\ldots,|k_n|\}$ sufficiently large, then the following summation by parts formula holds:

$$\sum_{\mathbf{k}\in\mathbb{Z}^n} \nabla^{2,h} v_{\mathbf{k}}\cdot\vec{w}_{\mathbf{k}} = -\sum_{\mathbf{k}\in\mathbb{Z}^n} v_{\mathbf{k}}\nabla^{2,h}\cdot\vec{w}_{\mathbf{k}}. \tag{24}$$

Theorem 2.1 *If $f\in C^3(B_\alpha)$, $u\in C^4(B_\alpha)$, then $\mathcal{I}^{1,h}\to\int_\Omega f(\vec{x})\,d\vec{x}$ as $h\to 0$, and*

$$\mathcal{I}^{1,h} = \int_\Omega f(\vec{x})\,d\vec{x} + O(h^2). \tag{25}$$

PROOF. We first break $\mathcal{I}^{1,h}$ into two parts, resulting in a discrete version of the decomposition (20):

$$\begin{aligned}\mathcal{I}^{1,h} &= h^n\sum_{\mathbf{k}\in\mathbb{Z}^n} H^{1,h}_{\mathbf{k}}\, f(\vec{x}_{\mathbf{k}})\\ &= h^n\sum_{\mathbf{k}\in\mathbb{Z}^n} H^{1,h}_{\mathbf{k}}\,\hat{f}(\vec{x_{\mathbf{k}}}) + h^n\sum_{\mathbf{k}\in\mathbb{Z}^n} H^{1,h}_{\mathbf{k}}\check{f}(\vec{x}_{\mathbf{k}})\\ &= \underbrace{h^n\sum_{\mathbf{k}\in\mathbb{Z}^n} H^{1,h}_{\mathbf{k}}\,\hat{f}(\vec{x_{\mathbf{k}}})}_{=:P^h} + \underbrace{h^n\sum_{\mathbf{k}\in\mathbb{Z}^n} H(u_{\mathbf{k}})\check{f}(\vec{x}_{\mathbf{k}})}_{=:Q^h}.\end{aligned} \tag{26}$$

Here we have used (18) to replace $H^{1,h}_{\mathbf{k}}$ by $H(u_{\mathbf{k}})$ in the second sum. Recalling that $H(u)\check{f}$ is smooth and compactly supported, we can view Q^h as a multi-dimensional midpoint rule approximation to the integral Q defined in (20). It follows that $Q^h = Q + O(h^2)$.

The remainder of the proof consists of showing that $P^h = P + O(h^2)$. Let $R_{\mathbf{k}}$ denote the grid cube centered at $\vec{x}_{\mathbf{k}}$ whose edges all have length h. Let K denote the set of indices $\mathbf{k}$ where $I(u)\,\mathcal{F}(\vec{x})$ is not identically zero on $R_{\mathbf{k}}$. In

view of Lemma 2.1, it is clear that

$$P = \sum_{\mathbf{k}\in K} \int_{R_{\mathbf{k}}} I(u)\mathcal{F}(\vec{x})\, d\vec{x}. \tag{27}$$

On the other hand, recalling the definition (14) of $H^{1,h}_{\mathbf{k}}$, and then employing (19), the sum P^h is given by

$$P^h = h^n \sum_{\mathbf{k}\in\mathbb{Z}^n} \left(\nabla^{2,h} I(u_{\mathbf{k}}) \cdot \nabla^{2,h} u_{\mathbf{k}} / \left| \nabla^{2,h} u_{\mathbf{k}} \right|^2 \right) \hat{f}(\vec{x_{\mathbf{k}}}). \tag{28}$$

Let $\mathcal{F}^h_{\mathbf{k}}$ be the following discrete analog of the quantity $\mathcal{F}$ defined in (21):

$$\mathcal{F}^h_{\mathbf{k}} = -\nabla^{2,h} \cdot \left(\hat{f}(\vec{x}_{\mathbf{k}}) \nabla^{2,h} u_{\mathbf{k}} / \left| \nabla^{2,h} u_{\mathbf{k}} \right|^2 \right). \tag{29}$$

Summing (28) by parts using (24), and then recalling (29) yields

$$P^h = h^n \sum_{\mathbf{k}\in\mathbb{Z}^n} I(u_{\mathbf{k}})\, \mathcal{F}^h_{\mathbf{k}}. \tag{30}$$

Due to the smoothness of $\hat{f}$ and u, $\mathcal{F}^h_{\mathbf{k}} = \mathcal{F}(\vec{x}_{\mathbf{k}}) + O(h^2)$. Also, since $\hat{f}$ has compact support, the number of indices $\mathbf{k}$ where $\mathcal{F}^h_{\mathbf{k}}$ is nonzero is $O(h^{-n})$. These observations allow us to replace (30) by

$$\begin{aligned} P^h &= h^n \sum_{\mathbf{k}\in\mathbb{Z}^n} I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}}) + O(h^2) \\ &= h^n \sum_{\mathbf{k}\in K} I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}}) + O(h^2). \end{aligned} \tag{31}$$

To get this last equality, we have used the fact that $\sum_{\mathbf{k}\in\mathbb{Z}^n} I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}}) = \sum_{\mathbf{k}\in K} I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}})$. By comparing (31) and (27), it is evident that the proof will be complete as soon as we show that

$$h^n \sum_{\mathbf{k}\in K} I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}}) = \sum_{\mathbf{k}\in K} \int_{R_{\mathbf{k}}} I(u)\mathcal{F}(\vec{x})\, d\vec{x} + O(h^2). \tag{32}$$

Let K_1 denote the set of indices $\mathbf{k} \in K$ where $R_{\mathbf{k}}$ does not intersect $\partial\Omega$, and let $K_2 = K \setminus K_1$. For $\mathbf{k} \in K_1$, $I(u)\mathcal{F}(\vec{x}) \in C^2(R_{\mathbf{k}})$. For these indices, the multidimensional version of the midpoint rule yields

$$h^n I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}}) = \int_{R_{\mathbf{k}}} I(u)\mathcal{F}(\vec{x})\, d\vec{x} + O(h^{n+2}), \quad \mathbf{k} \in K_1. \tag{33}$$

For $\mathbf{k} \in K_2$, we have $I(u)\mathcal{F}(\vec{x}) \in \mathrm{Lip}(R_{\mathbf{k}})$. For these indices,

$$h^n I(u_{\mathbf{k}})\, \mathcal{F}(\vec{x}_{\mathbf{k}}) = \int_{R_{\mathbf{k}}} I(u)\mathcal{F}(\vec{x})\, d\vec{x} + O(h^{n+1}), \quad \mathbf{k} \in K_2. \tag{34}$$

Since $\mathcal{F}(\vec{x})$ has compact support, the number of indices $\mathbf{k} \in K_1$ is $O(h^{-n})$. Due to the fact that $\partial\Omega$ is a compact $n-1$ dimensional manifold, the number

Table 1
Relative errors for Example 1. $\Omega \in \mathbb{R}^2$ is an ellipse.

	FDMH$_1$				FDMH$_2$			
	Dist. Function (64)		Non-dist. Function (8)		Dist. Function (128)		Non-dist. Function (32)	
h	Error	Rate	Error	Rate	Error	Rate	Error	Rate
.05	3.55e−4		3.38e−3		3.55e−5		1.63e−5	
.05/2	8.96e−5	1.99	8.46e−4	2.00	8.96e−7	4.69	8.69e−7	4.22
.05/4	2.11e−5	2.09	2.12e−4	2.00	2.11e−8	4.01	5.50e−8	3.98
.05/8	5.50e−6	1.94	5.31e−5	2.00	5.50e−9	4.04	3.49e−9	3.98
.05/16	1.36e−6	2.20	1.32e−5	2.00	1.36e−11	4.47	2.05e−10	4.09

of indices $\mathbf{k} \in K_2$ is $O(h^{1-n})$. Combining these observations with (33) and (34), we have proven (32), and the proof of the theorem is complete.

Remark 2.1 *The regularity conditions imposed on f and u in Theorem 2.1 are probably stronger than required. For example, our numerical examples seem to indicate $O(h^2)$ convergence for FDMH$_1$ even if f or u has jumps in the first derivatives that occur along a finite number of $n-1$ dimensional manifolds.*

Remark 2.2 *The proof of a similar theorem for FDMH$_2$ is not quite a straightforward modification of the proof of Theorem 2.1. We leave this for a future paper.*

3 Numerical Examples

Example 1. In this example, $\Omega \subset \mathbb{R}^2$ is the interior of the ellipse $x^2+(2y)^2 = 1$, and the integrand is $f(x,y) = e^x$. We rotated the grid by 45^o. The value of the integral is $\mathcal{I} \approx 1.775499689218604$. We tested both FDMH algorithms using both a signed distance function $u(x,y) = 1 - \sqrt{x^2+(2y)^2}$, and also $u(x,y) = 1-(x^2+(2y)^2)$, which is not a signed distance function. Table 1 shows that FDMH$_1$ seems to be converging at a rate of about $O(h^2)$, and FDMH$_2$ seems to be converging at a rate of $O(h^4)$. The errors shown in Table 1 are the absolute values of the relative errors, averaged over a number of small random grid shifts. The number of grid shifts for any given calculation is appears in parentheses in the heading of the table.

Example 2. In this example $\Omega \subset \mathbb{R}^3$ is the ellipsoid

$$\rho := \sqrt{(x/a)^2 + (y/b)^2 + (z/c)^2} < 1, \quad a = 1/2, b = 1/2.5, c = 1/3. \qquad (35)$$

Table 2
Errors for Example 2. $\Omega \in \mathbb{R}^3$ is an ellipsoid.

	FDMH$_1$ (4)		FDMH$_2$(4)	
h	Error	Rate	Error	Rate
.05	4.48e−3		1.49e−3	
.05/2	1.12e−3	2.00	9.47e−6	7.30
.05/4	2.79e−4	2.00	5.55e−7	4.09
.05/8	6.97e−5	2.00	3.38e−8	4.04

The integrand is $f(x, y, z) = \exp(-\rho^3/3)$. The exact value of the integral is $\mathcal{I} = 4\pi abc(1 - e^{-1/3})$. We used a signed distance function for the level set function $u(x, y, z)$. We rotated all coordinates by the matrix A defined by

$$A = \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} \\ 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} \end{bmatrix} \tag{36}$$

before applying the grid.

Table 2 indicates that FDMH$_1$ converges at a rate of $O(h^2)$ for this problem, and FDMH$_2$ converges at a rate of $O(h^4)$.

Example 3. This example is borrowed from [6]. The problem is to compute an integral of the form (16) where $f(x, y) = 1$, and Ω is the capsule shaped region described in [6]. In addition to testing FDMH$_1$ and FDMH$_2$, we tested the regularized Heaviside function described in [6], which we denote $H^{L,\tilde{\epsilon}}$. $H^{L,\tilde{\epsilon}}$ is given by (7), except with the constant ϵ replaced by the variable $\tilde{\epsilon}$ defined by

$$\tilde{\epsilon} = \frac{\left|\nabla^{2,h} u_{j,k}\right|_1}{|\nabla^{2,h} u_{j,k}|} \cdot \frac{h}{2}. \tag{37}$$

Here $|\cdot|_1$ denotes the L^1 norm. Table 3 shows the absolute values of the relative errors using a signed distance function for the level set function $u(x, y)$. The algorithm based on $H^{L,\tilde{\epsilon}}$ seems to be converging like $O(h^2)$, in agreement with the results of [6]. The FDMH algorithms also give accurate results for this problem. It is not clear from the data in Table 3 what their rates of convergence are.

Table 4 shows the results using a level set function that is not a signed distance function, but is smoother than the signed distance function used to construct Table 3. The algorithm based on $H^{L,\tilde{\epsilon}}$ and FDMH$_1$ seem to be converging like $O(h^2)$, and FDMH$_2$ appears to be converging at a rate of $O(h^4)$.

Table 3
Errors for Example 3. (capsule region) using a signed distance function

	$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$(64)	
h	Error	Rate	Error	Rate	Error	Rate
.02	3.92e−4		1.36e−5		1.26e−5	
.02/2	9.43e−5	2.06	5.42e−6	1.33	5.10e−7	4.63
.02/4	2.37e−5	1.99	6.20e−7	3.13	4.83e−8	3.40
.02/8	5.56e−6	2.09	1.18e−7	2.39	2.04e−9	4.57

Table 4
Errors for Example 3. (capsule region) using a smoother level set function

	$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$(64)	
h	Error	Rate	Error	Rate	Error	Rate
.02	2.00e−2		4.22e−3		8.59e−5	
.02/2	4.56e−3	2.13	1.06e−3	1.99	5.37e−6	4.00
.02/4	1.13e−3	2.01	2.66e−4	1.99	3.31e−7	4.02
.02/8	2.78e−4	2.02	6.63e−5	2.00	2.04e−8	4.02

Both level set functions used in this example have jumps in their second derivatives; the jump for the smoother version is about 1/8 the size of the jump for the signed distance function. This lack of smoothness may explain why the numerical results do not show a clear rate of convergence in some cases for the FDMH algorithms.

Example 4. For this example, the region $\Omega \subset \mathbb{R}^2$ is the intersection of the disk $\sqrt{x^2+y^2} < R := .35\sqrt{2}$ with the half plane $x - y > 0$. Ω is a half disk, with the linear part of the boundary misaligned with the coordinate axes by 45 degrees. The integrand is $f(x,y) = x^2$, and the exact value of the integral is $\mathcal{I} = (\pi/8)R^4$. Note that $\partial\Omega$ has corners. For a level set function we used

$$u(x,y) = \min\left(R - \sqrt{x^2+y^2},\, x - y\right), \tag{38}$$

which is continuous, but only piecewise smooth, with jumps in the first derivatives.

Table 5 indicates that the algorithms based on $H^{L,\epsilon}$ (with $\epsilon = h$) and $H^{L,\tilde{\epsilon}}$ (defined by (37)) are converging at a rate of about $O(h)$, while the FDMH algorithms give approximately second order convergence. The cause of the reduced rates of convergence for $H^{L,\tilde{\epsilon}}$ and FDMH$_2$ is no doubt the lower regularity of the level set function u.

Table 5
Errors for Example 4. $\Omega \subset \mathbb{R}^2$ is a half disk.

	$H^{L,\epsilon}$, $\epsilon = h$ (64)		$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$ (64)	
h	Error	Rate	Error	Rate	Error	Rate	Error	Rate
.03	1.23e−2		4.20e−3		3.86e−3		1.75e−3	
.03/2	5.91e−3	1.06	1.08e−3	1.96	9.54e−4	2.02	3.21e−4	2.45
.03/4	2.25e−3	1.39	4.46e−4	1.28	2.42e−4	1.98	7.35e−5	2.09
.03/8	1.28e−3	0.81	1.99e−4	1.16	5.87e−5	2.04	1.73e−5	2.10
.03/16	5.17e−4	1.31	9.64e−5	1.05	1.47e−5	1.98	4.69e−6	1.89
.03/32	2.94e−4	0.84	5.03e−5	0.94	3.65e−6	2.01	1.03e−6	2.19

Table 6
Errors for Example 5. Intersection of two cylinders in $\mathbb{R}^3$. The notation FDMH$_1^*$, FDMH$_2^*$ means that the coordinates were rotated by the matrix A.

	FDMH$_1$ (16)		FDMH$_2$ (16)		FDMH$_1^*$ (16)		FDMH$_2^*$ (16)	
h	Error	Rate	Error	Rate	Error	Rate	Error	Rate
.05	2.75e−4		1.40e−2		6.66e−4		5.92e−2	
.05/2	7.18e−5	1.94	4.07e−4	5.10	1.71e−4	1.97	1.04e−2	2.51
.05/4	1.76e−5	2.03	1.03e−4	1.98	4.69e−5	1.88	2.07e−3	2.33

Example 5. In this example $\Omega \subset \mathbb{R}^3$ is the intersection of the two cylindrical regions $\Omega_1 = \{(x,y,z)|u_1(x,y,z) := 1 - \sqrt{1-x^2-y^2} > 0\}$ and $\Omega_2 = \{(x,y,z)|u_2(x,y,z) := 1 - \sqrt{1-x^2-z^2} > 0\}$. As a level set function for the desired region $\Omega = \Omega_1 \cap \Omega_2$, we used $u = \min(u_1, u_2)$. The integrand is $f = 1$, and the exact value of the integral is $\mathcal{I} = 16/3$. We ran both of the FDMH algorithms, first with the coordinates not rotated, and then rotated using the matrix A defined in (36). Table 6 indicates that in all the cases, the rate of convergence is close to $O(h^2)$. Clearly FDMH$_1$ is more accurate than FDMH$_2$ for this problem, and both methods are more accurate when the coordinates are not rotated. In this example, $\partial\Omega$ is not smooth. The level set function u has jumps in its first derivatives. We attribute the reduced rate of convergence for FDMH$_2$ to this lack of smoothness.

Example 6. In this example $\Omega = [-1,1] \subset \mathbb{R}^1$, and $f(x) = e^x$. In the previous examples, FDMH$_2$ gave fourth order accuracy for smooth data. In each case where the rate of convergence was less than $O(h^4)$, the data had reduced regularity. These results might give the impression that FDMH$_2$ always gives fourth order accuracy for sufficiently smooth data. However, this one-dimensional example shows that FDMH$_2$ is generally only third order accurate, as can be seen in Table 7. We include this example because we were unable to construct any multidimensional examples with smooth data that showed less than fourth

Table 7
Errors for Example 6. One-dimensional example.

	FDMH$_1$ (512)		FDMH$_2$ (512)	
h	Error	Rate	Error	Rate
.04	2.72e−4		3.57e−7	
.04/2	6.70e−5	2.02	4.20e−8	3.09
.04/4	1.66e−5	2.01	5.18e−9	3.02
.04/8	4.13e−6	2.00	6.57e−10	2.98
.04/16	1.03e−6	2.00	7.89e−11	3.06

Table 8
Errors for Example 7. $\Omega \subset \mathbb{R}^2$ is a disk. f is not smooth.

	$H^{L,\epsilon}$, $\epsilon = h$ (128)		$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$ (64)	
h	Error	Rate	Error	Rate	Error	Rate	Error	Rate
.05	3.28e−3		9.71e−4		1.05e−3		1.60e−2	
.05/2	1.18e−3	1.47	2.37e−4	2.03	2.60e−4	2.01	3.13e−5	9.00
.05/4	3.83e−4	1.62	6.48e−5	1.87	6.47e−5	2.01	8.32e−6	1.91
.05/8	1.49e−4	1.36	1.67e−5	1.96	1.60e−5	2.02	1.60e−6	2.38

order accuracy.

Example 7. In this example $\Omega \subset \mathbb{R}^2$ is the open disk $x^2 + y^2 < 1$, which we represent with a signed distance function. The integrand is $f(x) = |x|$. We rotated the grid by 45°. The exact value of the integral is $\mathcal{I} = 4/3$. The point of this experiment is that f has a jump in its first derivatives. According to Table 8, the algorithm based on $H^{L,\tilde{\epsilon}}$ defined by (37), and both FDMH algorithms are converging at (or close to) a rate of $O(h^2)$. The algorithm based on $H^{L,\epsilon}$ (with $\epsilon = h$) is converging at a lower rate, approximately $O(h^{1.5})$.

Example 8. In this example, we use the one-dimensional version of our Heaviside function FDMH$_1$ to compute a discrete delta function for use in a Fourier method [9] approximation of the scalar conservation law

$$w_t + cw_x = \delta(x - a), \quad w(x, 0) = 0, \tag{39}$$

which we solve for $t > 0$ on the spatial domain $x \in \mathbb{R}$. Here $\delta(\cdot)$ denotes the Dirac delta function. The solution of this PDE is

$$w(x,t) = \begin{cases} 0, & x \notin [a, a + ct] \\ 1/c, & x \in [a, a + ct] \end{cases}. \tag{40}$$

We choose $a \in [-\pi, \pi)$, and only consider times $t > 0$ that are small enough

that $a + ct \in [-\pi, \pi)$. We use the interval $S = [-\pi, \pi)$ as our computational domain, letting $h = \pi/N$, and defining gridpoints $x_j = jh$, for $-N \le j \le N-1$. Letting $\hat{w}^n$ denote the discrete Fourier transform (DFT) of the approximate solution at time level $t^n = n\Delta t$, we advance $\hat{w}^n$ in time according to

$$\hat{w}^{n+1}(k) = e^{-ick\Delta t}\hat{w}^n(k) + \Delta t\hat{\delta}^h(k), \quad \Delta t = h/2, \quad \hat{w}^0(k) = 0. \tag{41}$$

Here $\hat{\delta}^h$ is the DFT of our approximate delta function. Recall that the one-dimensional DFT, and inverse DFT, of a grid function $v_j := v(x_j)$ are

$$\hat{v}(k) = h\sum_{j=-N}^{N-1} e^{-ikx_j}v_j, \quad v_j = \frac{1}{2\pi}\sum_{k=-N}^{N-1} e^{ikx_j}\hat{v}(k), \quad -N \le j,k \le N-1. \tag{42}$$

Note that the marching formula (41) is only first order accurate in time. This level of accuracy is sufficient since the solution is piecewise constant. What is important for our purposes is that (41) is not dissipative.

We tested two different approximate delta functions. The first of these is the linear hat delta function (the derivative of $H^{L,\epsilon}$ defined by (7)):

$$\delta_j^h = \delta^{L,\epsilon}(u_j) = \begin{cases} (1 - |u_j/\epsilon|)/\epsilon, & |u_j/\epsilon| < 1 \\ 0, & |u_j/\epsilon| > 1. \end{cases}, \quad \epsilon = h. \tag{43}$$

Here we used $u(x) = x - a$ as our level set function, and $u_j := u(x_j)$.

The other approximate delta function that we tested is based on the one-dimensional version of our FDMH_1 Heaviside function (14):

$$\delta_j^h = \delta_j^{F,h} = \left(H_{j+1}^{1,h} - H_{j-1}^{1,h}\right)/(2h) = \left(I(u_{j+2}) - 2I(u_j) + I(u_{j-2})\right)/(4h^2), \tag{44}$$

again using the level set function $u(x) = x - a$. This approximate delta function is the one-dimensional version of the second order approximate delta function that we proposed in [18] (referred to there as Method 2), and studied further in [19] and [20] (referred to there as FDM_2).

In addition to the basic method just described, we also used the smoothed Fourier method of Majda, McDonough, and Osher [10]. This smoothing greatly increases accuracy away from the discontinuities. For this very simple problem, smoothing means that we get rid of the highest wave numbers in the DFT $\hat{\delta}^h$ by multiplying it by a smooth cutoff function. According to [10], the cutoff function σ^h is defined by

$$\sigma^h(k) = \begin{cases} e^{-\alpha(\omega+\omega_0)^{2p}}, & \omega < -\omega_0 \\ 1, & -\omega_0 \le \omega \le \omega_0 \\ e^{-\alpha(\omega-\omega_0)^{2p}}, & \omega > \omega_0 \end{cases} \tag{45}$$

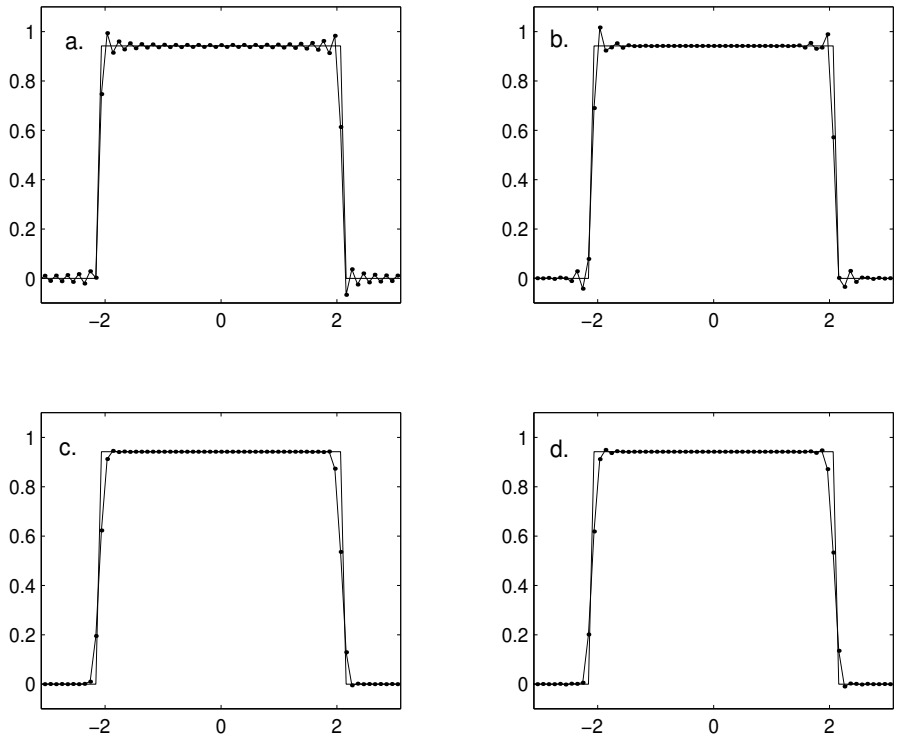

Fig. 2. Approximate solutions (dotted line) of $w_t + cw_x = \delta(x - a)$ via Fourier method. $N = 32$, with 80 time steps. Using $\delta^{L,\epsilon}$, without smoothing method of [10] (Plot a.), and with smoothing (Plot b.). Using $\delta^{F,h}$, without smoothing method (Plot c.), and with smoothing (Plot d.). Line without dots is exact solution.

where $\omega = hk/\pi$. There is some latitude in the choice of the parameters p, α, and ω_0. We chose $p = 3$, $\omega_0 = .3$, which are among the values explicitly recommended in [10]. We used $\alpha = .75/h$ because we found that it gave good results for this problem.

We used $c = 1.062$, $a = -2.07$, and ran our tests both with and without the smoothing of [10]. Figure 2 shows that the use of $\delta^{L,\epsilon}$ results in a noticeable amount of spurious oscillation (Gibbs phenomenon). Away from the discontinuities, the oscillation is damped out considerably with the application of smoothing. The calculation based on our FDMH-based $\delta^{F,h}$ results in very little spurious oscillation. The smoothing operation increases the oscillation slightly in a very small interval near the discontinuities.

Figure 3 shows that all four methods result in $O(1)$ errors near the jumps in the solution. Away from those jumps, it is clear that the approximations based on $\delta^{F,h}$ are much more accurate than their counterparts based on $\delta^{L,\epsilon}$. Also, in each case, the smoothing operation significantly reduces the error in the region away from the discontinuities in the solution.

References [7] and [8] partially motivate our interest in (39). Those works

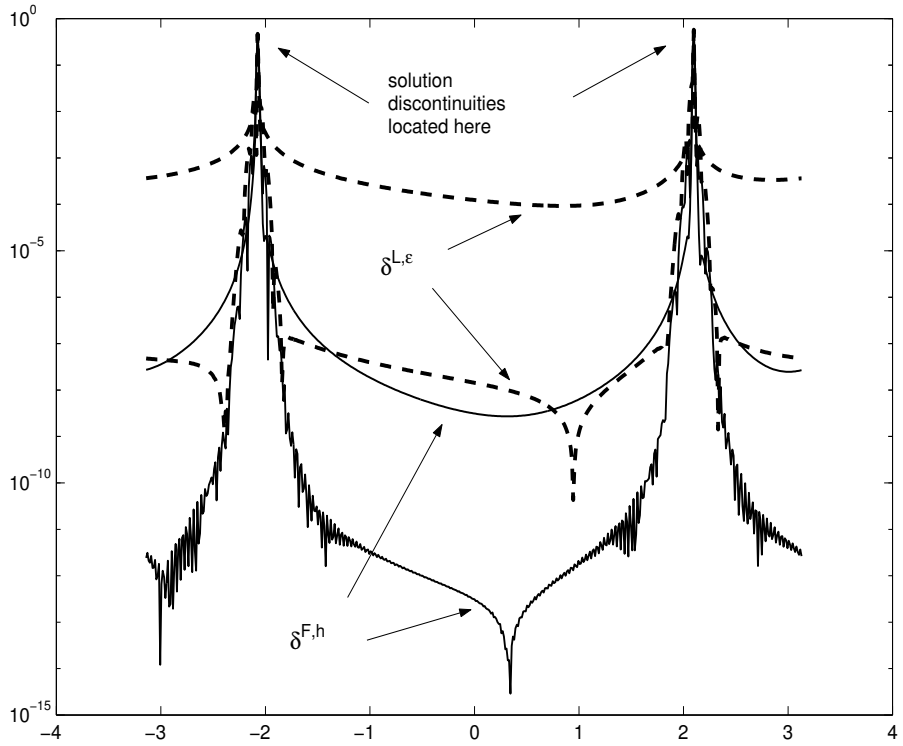

Fig. 3. Approximation errors for Fourier method applied to $w_t + cw_x = \delta(x-a)$. Using $\delta^{L,\epsilon}$ (broken line, upper curve is without smoothing and lower curve is with smoothing), and using $\delta^{F,h}$ (solid line, upper curve is without smoothing and lower curve is with smoothing) based on differencing $H_j^{1,h}$. $N = 256$, with 640 time steps.

discuss the spectral collocation method as it applies to solving (39) and related singular problems. Reference [8] also discusses the use of a difference scheme, and in fact uses an approximate delta function that is somewhat similar to the one proposed here, i.e, based on differencing a Heaviside function. Specifically, they used a first order delta function which coincides with the one referred to as Method 1 in [18], and FDM_1 in [19] and [20].

As further motivation, the PDE (39) can also be seen as a simple linearized model for a nonlinear conservation law with source term:

$$w_t + f(w)_x = \delta(x-a). \tag{46}$$

Conservation laws like (46) arise when modeling sedimentation in clarifier-thickener units. In this situation it is natural to first express the equation in conservation form,

$$w_t + f(w)_x = H(x-a)_x \text{ or } w_t + (f(w) - H(x-a))_x = 0, \tag{47}$$

before discretizing by eg., a difference scheme [5] or front tracking algorithm [4].

Example 9. In this example, we employ our FDMH algorithms as a component in the Fourier method of solving the Cauchy problem for the constant coefficient PDE

$$w_t + aw_x + bw_y = 0, \quad a = 1.47, \quad b = 0.93. \tag{48}$$

The initial data is $\chi(x, y)$, the characteristic function of the disk $\Omega = \{(x, y)|(x+.853)^2 + (y + .781)^2 < 1.25^2\}$. Our computational domain is the square $S = [-\pi, \pi) \times [-\pi, \pi)$. We let $h = \pi/N$, and define $x_j = jh$, $y_k = kh$, $-N \le j, k \le N - 1$.

We compare three different methods of discretizing the initial data $\chi(x, y)$. In the first method, we use (4), i.e., $\chi(x_j, y_j) = H(u(x_j, y_k))$, and then compute the two-dimensional DFT $\hat{\chi}^0$:

$$\hat{\chi}^0(l, m) = h^2 \sum_{j=-N}^{N-1} \sum_{k=-N}^{N-1} e^{-i(lx_j+my_k)} H(u(x_j, y_k)), \quad -N \le l, m \le N-1. \tag{49}$$

For the other two discretization methods, we use our FDMH approximations, $\chi(x_j, y_j) \approx H^{q,h}_{j,k}$ for $q = 1$ or 2, and then compute the DFT $\hat{\chi}^q$:

$$\hat{\chi}^q(l, m) = h^2 \sum_{j=-N}^{N-1} \sum_{k=-N}^{N-1} e^{-i(lx_j+my_k)} H^{q,h}_{j,k}, \quad q = 1 \text{ or } 2. \tag{50}$$

For the level set function required for (49) and (50), we used the signed distance function $u(x, y) = 1.25 - \sqrt{(x + .853)^2 + (y + .781)^2}$. We compute both (49) and (50) with the Fast Fourier Transform (FFT).

We then advance $\hat{w}$ in time (we used $t = 1.2$ for our experiment) via

$$\hat{w}^q(l, m, t) = e^{-i(al+bm)t}\, \hat{\chi}^q(l, m), \tag{51}$$

and finally recover the approximate solution via inverse DFT (implemented via FFT):

$$w^q(x_j, y_k, t) = \left(\frac{1}{2\pi}\right)^2 \sum_{l=-N}^{N-1} \sum_{m=N}^{N-1} e^{i(lx_j+my_k)} \hat{w}^q(l, m, t), \quad q = 0, 1 \text{ or } 2. \tag{52}$$

Figure 4 shows the approximations at $t = 1.2$ using $N = 32$. There is some spurious oscillation located near where the solution is discontinuous. In the cases where the FDMH algorithms are used to compute the initial DFTs $\hat{\chi}^q$, the oscillation is only visible in a very narrow region. For the simplest method of computing the initial DFT, i.e. (49), the unwanted oscillation is visible in a much larger region.

To aid visualization, Figure 5 shows the results of a similar one-dimensional experiment. The discontinuities in the initial data are not aligned with the

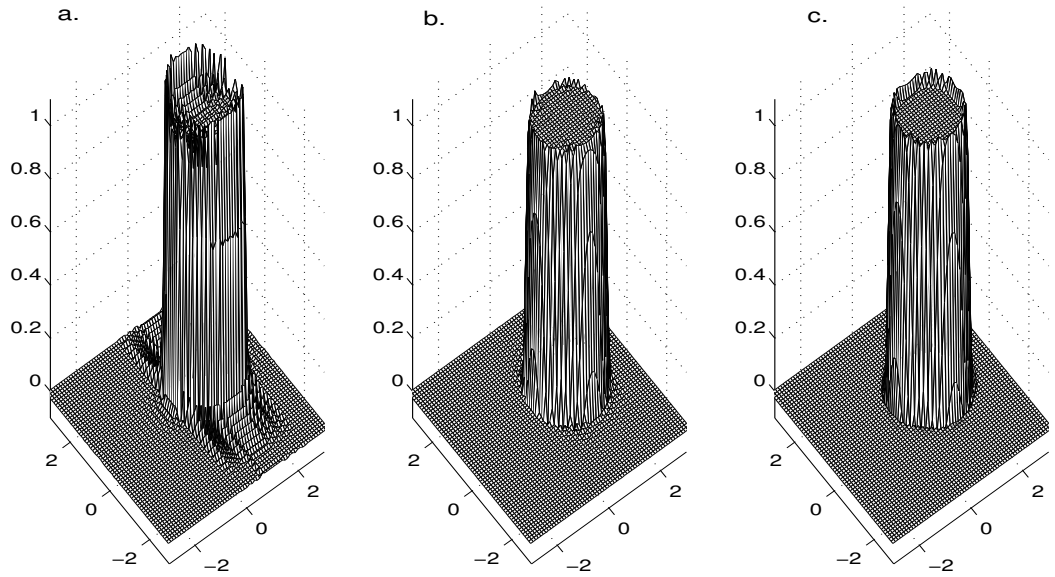

Fig. 4. Fourier method applied to 2D problem (without smoothing method of [10]). Initial data discretized via simplest method, (49) (Plot a.), via FDMH_1 (Plot b.), via FDMH_2 (Plot c.).

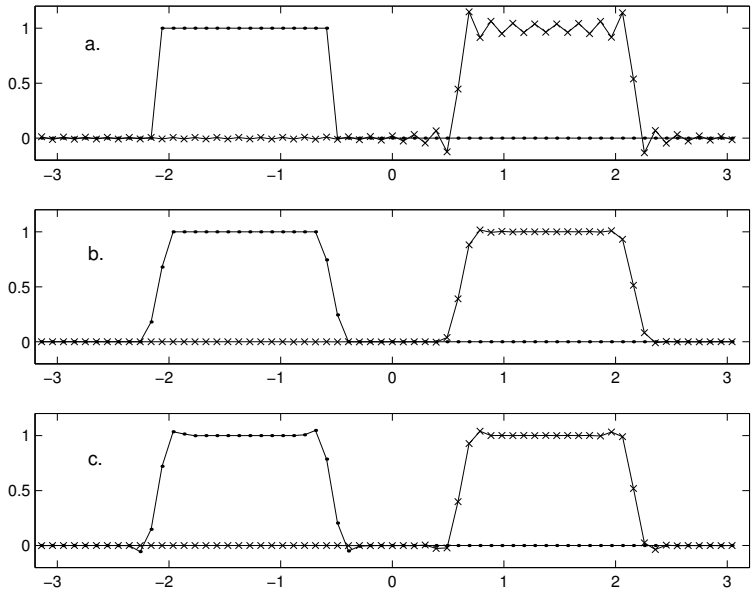

Fig. 5. Similar one-dimensional problem (without smoothing method of [10]). Simplest method, (49) (Plot a.), FDMH_1 (Plot b.), FDMH_2 (Plot c.) Initial data on the left (dots), Fourier method solution after some time on the right (x's).

mesh. From these plots, it is clear that the initial data using FDMH_1 or FDMH_2 is advected with reasonable accuracy by the Fourier method. On the other hand, with the simplest method (49) of initializing the data, the Fourier method introduces a large amount of spurious oscillation. The situation for our two-dimensional problem is similar.

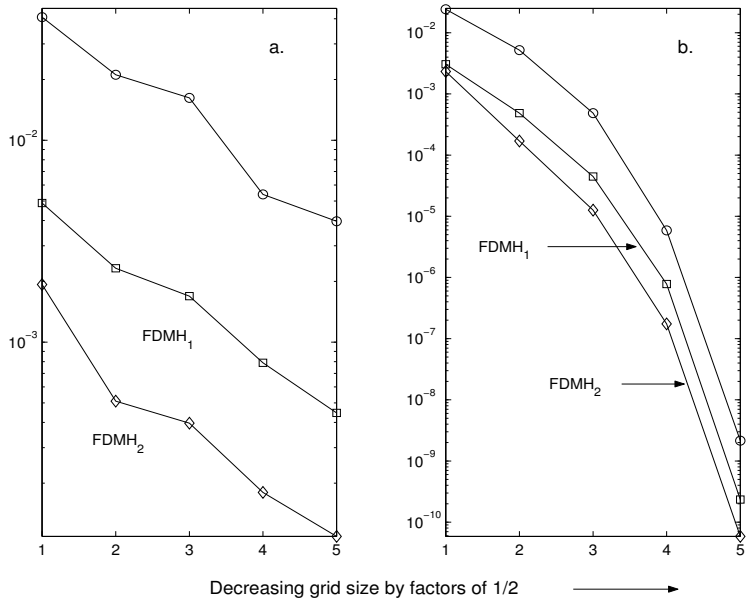

Fig. 6. Error away from interface for Fourier method. Without smoothing method of [10] (Plot a.), and with smoothing (Plot b.). Comparison of solutions using initial Fourier transform computed using simplest method (circles), FDMH_1 (squares), FDMH_2 (diamonds).

In Figure 6, we show the maximum absolute error in the solution as a function of mesh refinement, with the coarsest mesh corresponding to $N = 32$. For these error calculations, we only take into account points that are at least a distance .4 from the location of the discontinuity. The choice of .4 is only for convenience - any positive value would give the same qualitative results. Plot a. of Figure 6 shows that without further processing, all three methods give only first order accuracy, although FDMH_1 and FDMH_2 are much more accurate than the simplest method (49). This low convergence rate is very different from what would be observed with smooth initial data. In fact, with C^∞ data, all three of these methods of computing the initial Fourier transform would result in infinite order convergence.

We also used the smoothed version of [10] on this problem. In this case, that means that we multiply $\hat{\chi}^q$ by the two-dimensional cutoff function $\rho^h(l,m) = \sigma^h(l)\sigma^h(m)$, where σ^h is the one-dimensional cutoff function defined by (45).

It is apparent from Figure 6 that with the smoothing method, all three methods give much better accuracy than their unsmoothed counterparts at the same level of grid refinement, and also that the rate of convergence is much higher than first order. In fact, the rate of convergence appears to be increasing with decreasing mesh size, consistent with infinite order of convergence.

Our results for this example show that by using FDMH_1 or FDMH_2 to discretize the initial data, we can improve the accuracy of the Fourier method,

and this is true for both the original and smoothed version.

References

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

[2] R. Burden, J. Faires, Numerical Analysis, Fourth Edition, PWS-Kent Publishing Company, Boston, 1989.

[3] G. Dahlquist, A. Björk, Numerical Methods, Prentice-Hall, Inglewood Cliffs, New Jersey, 1974.

[4] R. Bürger, K.H. Karlsen, C. Klingenberg, N.H. Risebro, A front tracking approach to a model of continuous sedimentation in ideal clarifier-thickener units, Nonlin. Anal. Real World Appl. **4** (2003), 457–481.

[5] R. Bürger, K.H. Karlsen, N.H. Risebro, J.D. Towers, Well-posedness in BV_t and convergence of a difference scheme for continuous sedimentation in ideal clarifier-thickener units, Numer. Math. **97** (2004), 25–65.

[6] B. Engquist, A.K. Tornberg, R. Tsai, Discretization of Dirac Delta Functions in Level Set Methods, J. Comput. Phys. **207** (2005), 28-51.

[7] J.H. Jung, On the spectral collocation approximation of the discontinuous solution of singularly perturbed differential equations in one dimension. Preprint.

[8] J.H. Jung, G. Khanna, I. Nagle, A spectral collocation approximation of one-dimensional head-on collisions of black holes. Preprint.

[9] H.O. Kreiss, J. Oliger, Methods for the approximate solution of time dependent problems, GARP Publications Series no. 10, 1973.

[10] A. Majda, J. McDonough, S. Osher, The Fourier method for nonsmooth initial data, Math. Comp., **32** (144) (1978), 1041–1081.

[11] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, J. Comput. Phys. **226** (2007) 1432–1443.

[12] S. Osher, R. Fedkiw, Level set methods and dynamic implicit surfaces, Springer-Verlag, New York, 2003.

[13] S. Osher, J. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. **79** (1988) 12-49.

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

[15] J.A. Sethian, Level set methods and fast marching methods, Cambridge University Press, Cambridge, 1999.

[16] A.K. Tornberg. Multi-dimensional quadrature of singular and discontinuous functions, BIT **42** (2002) 644–669.

[17] A.K. Tornberg, B. Engquist. Numerical approximations of singular source terms in differential equations, J. Comput. Phys. **200** (2004) 462–488.

[18] J.D. Towers. Two methods for discretizing a delta function supported on a level set, J. Comput. Phys. **220** (2007) 915-931.

[19] J.D. Towers. Discretizing delta functions via finite differences and gradient normalization, Preprint at http://www.miracosta.edu/home/jtowers/.

[20] J.D. Towers. A convergence rate theorem for finite difference approximations to delta functions, J. Comput. Phys. **227** (2008) 6591-6597.