Journal of Computational Physics 228 (2009) 3478–3489

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Finite difference methods for approximating Heaviside functions

John D. Towers

MiraCosta College, 3333 Manchester Avenue, Cardiff-by-the-Sea, CA 92007-1516, USA

ARTICLE INFO

Article history:
Received 18 July 2008
Received in revised form 19 January 2009
Accepted 27 January 2009
Available online 6 February 2009

Keywords:
Heaviside function
Level set method
Quadrature
Irregular region
Singular source term
Finite difference
Regular grid
Convergence rate

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 [J.D. Towers, Two methods for discretizing a delta function supported on a level set, J. Comput. Phys. 220 (2007) 915–931; J.D. Towers, Discretizing delta functions via finite differences and gradient normalization, Preprint at http://www.miracosta.edu/home/jtowers/; J.D. Towers, A convergence rate theorem for finite difference approximations to delta functions, J. Comput. Phys. 227 (2008) 6591–6597]. We consider our approximate Heaviside functions 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 at least third order accuracy if the integrand f and $\partial\Omega$ are sufficiently smooth. Numerical experiments also indicate that our approximations are effective when used to discretize certain singular source terms in partial differential equations.

We mostly focus on smooth f and u. By this we mean that f is smooth in a neighborhood of Ω, u is smooth in a neighborhood of $\partial\Omega$, and the level set $u(x) = 0$ is a manifold of codimension one. However, our algorithms still give reasonable results if either f or u has jumps in its derivatives. Numerical experiments indicate approximately second order accuracy for both algorithms if the regularity of the data is reduced in this way, assuming that the level set $u(x) = 0$ is a manifold.

Numerical experiments indicate that dependence on the placement of Ω with respect to the grid is quite small for our algorithms. Specifically, a grid shift results in an $O(h^p)$ change in the computed solution, where p is the observed rate of convergence.

1. Introduction

We are 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 [7,8,10]. In this context it is common practice to write the integral $\mathcal{I}$ as

E-mail address: john.towers@cox.net
URLs: http://www.miracosta.edu/home/jtowers/

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.01.026

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

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, e.g. [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 $\Omega_1, \Omega_2 \subset \mathbb{R}^n$, 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 $\{\vec{x}_{\mathbf{k}} = (x^1_{k_1}, \ldots, x^n_{k_n}) | \mathbf{k} := (k_1, \ldots, k_n) \in \mathbb{Z}^n\}$ 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

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

Another commonly used approximate Heaviside function is

$$H^{L,\epsilon}(z) = \begin{cases} H(z), & |z| \geqslant \epsilon, \\ \frac{1}{2\epsilon}(z + \epsilon), & |z| \leqslant \epsilon, \end{cases} \quad \epsilon = O(h). \tag{7}$$

Engquist et al. [4] 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 [11]. 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 [12] 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 [5] have also proposed a method for approximating the integral $\mathcal{I}$. Their method involves decomposition of the region into simplices. A quadrature rule is then applied on the simplices, and the resulting approximation to $\mathcal{I}$ is second order accurate. In [6] these authors used their technique to produce an explicit approximation to the Heaviside function. An advantage of their method is that it is not sensitive to the placement of the region Ω with respect to the mesh. These authors attribute this to the fact that their algorithms do not use derivative data.

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 [13–15].

We assume that u is defined and positive on Ω, and that Ω is bounded (see Fig. 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 $\vec{n} = -\nabla u/|\nabla u|$.

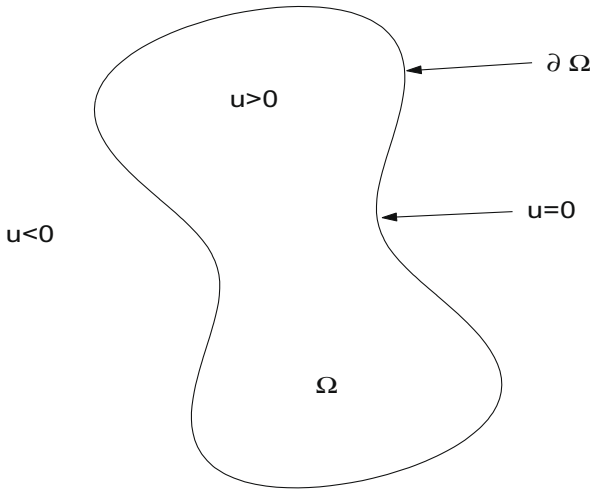

Fig. 1. The region of integration Ω defined by the level set $u = 0$.

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 $H(z)$:

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

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

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

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

$$\vec{x}_{\mathbf{k}} \in \mathcal{N}_v \Longleftrightarrow u(\vec{x}_{\mathbf{k}}) u(\vec{x}_{\mathbf{k}} \pm vh\mathbf{e}_m) \leqslant 0 \quad \text{for some } m \in \{1, \ldots, n\}. \tag{13}$$

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

FDMH$_1$:

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

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

$\mathbf{FDMH}_2$:

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

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 [1,9].

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.

The advantages of the algorithms presented here are simplicity and accuracy. A potential drawback, depending on the application, is that our algorithms are slightly dependent on the placement of Ω with respect to the grid. This grid placement sensitivity is consistent with the observation of Min and Gibou [5,6] that algorithms which use derivative data will be somewhat more sensitive to perturbations of the data than those that do not. Our FDMH_1 algorithm, which uses only first derivative data, is generally less sensitive to grid perturbations than FDMH_2, which uses second derivative data. We emphasize that our algorithms are nevertheless quite accurate, and that the observed grid placement sensitivity is small. Our numerical experiments indicate that the change in the computed solution due to a grid shift is $O(h^p)$, where p is the observed rate of convergence.

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 approximately $O(h^2)$ convergence for both methods, assuming that the level set $u = 0$ is a manifold. A limited amount of testing in the more degenerate situation where $u = 0$ fails to be a manifold (for example $u = 0$ consists of two touching circles in $\mathbb{R}^2$) indicates second order convergence for FDMH_1 and first order convergence for FDMH_2. Also in Section 3, we test our FDMH approximate Heaviside functions as singular source terms in the heat equation, and the Poisson equation. We observe convergence rates of approximately second order, and find that our new algorithms are more accurate than a commonly used approximate Heaviside function.

2. A convergence proof for 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 $|r| \geqslant \alpha/2$. Let $\rho(\vec{x}) = \mu(u(\vec{x}))$, and define

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

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 $\mathrm{supp}(H(u)\check{f}) \subset \Omega$, and for h sufficiently small, we will have

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

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

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

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*

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

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

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

Proof. We integrate by parts:

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

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* $f \in C^3(\Omega\cup B_\alpha), u \in C^4(B_\alpha)$, *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):

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

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 multidimensional 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

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

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

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

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

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

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 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°. 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 appears in parentheses in the heading of the table.

In Table 2, we focus on the sensitivity of the algorithms to grid shifts. This table shows the ratio of the maximum error to the average error (the type of error shown in Table 1). This maximum/average error ratio can be viewed as a measure of the sensitivity to grid shifts. The idea of considering a ratio like this comes from Min and Gibou [5] who use instead the ratio of the maximum to the minimum error. We see from Table 2 that FDMH$_2$ is somewhat more sensitive to grid shifts than FDMH$_1$. Nevertheless, in all cases the maximum error is a small multiple of the average error. This means for example that the average errors displayed in Table 1 are not too different from the worst case. Based on the observed convergence rates in Table 1, these small ratios also imply that, although a grid shift causes a change in the computed answer, the magnitude of that change is $O(h^2)$ for FDMH$_1$ and $O(h^4)$ for FDMH$_2$.

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

h	FDMH$_1$				FDMH$_2$			
	Dist. function (64)		Non-dist. function (8)		Dist. function (1024)		Non-dist. function (32)	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
.05	3.55e−4		3.38e−3		1.04e−5		1.63e−5	
.05/2	8.96e−5	1.99	8.46e−4	2.00	4.03e−7	4.69	8.69e−7	4.22
.05/4	2.11e−5	2.09	2.12e−4	2.00	2.03e−8	4.31	5.50e−8	3.98
.05/8	5.50e−6	1.94	5.31e−5	2.00	1.27e−9	4.00	3.49e−9	3.98
.05/16	1.36e−6	2.20	1.32e−5	2.00	7.77e−11	4.03	2.05e−10	4.09

Table 2
Sensitivity to grid shifts for Example 1. $\Omega \in \mathbb{R}^2$ is an ellipse.

h	$FDMH_1$		$FDMH_2$	
	Dist. function (64) Ratio	Non-dist. function (8) Ratio	Dist. function (1024) Ratio	Non-dist. function (32) Ratio
.05	1.47	1.02	1.74	1.35
.05/2	1.25	1.01	2.36	1.37
.05/4	1.27	1.00	3.54	1.96
.05/8	1.09	1.00	2.85	1.62
.05/16	1.09	1.00	3.28	2.23

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

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

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 3 indicates that $FDMH_1$ converges at a rate of $O(h^2)$ for this problem, and $FDMH_2$ converges at a rate of $O(h^4)$.

Table 4 shows the results of a grid placement sensitivity study. We used 16 random grid shifts for each reported mesh size. The maximum/average error ratios are close to one for this example, indicating that the average errors reported in Table 3 are close to the worst case. As in the case of Example 1, this table indicates that a grid shift results in an $O(h^2)$ change in the result for $FDMH_1$, and the change is $O(h^4)$ for $FDMH_2$.

Example 3. This example is borrowed from [4]. 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 [4]. In addition to testing $FDMH_1$ and $FDMH_2$, we tested the regularized Heaviside function described in [4], 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

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

Here $|\cdot|_1$ denotes the L^1 norm. Table 5 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 [4]. The FDMH algorithms also give accurate results for this problem. It is not clear from the data in Table 5 what their rates of convergence are.

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

h	$FDMH_1$ (4)		$FDMH_2$ (4)	
	Error	Rate	Error	Rate
.05	4.48e−3		1.57e−4	
.05/2	1.12e−3	2.00	9.47e−6	4.05
.05/4	2.79e−4	2.00	5.55e−7	4.09
.05/8	6.97e−5	2.00	3.38e−8	4.04

Table 4
Sensitivity to grid shifts for Example 2. $\Omega \in \mathbb{R}^3$ is an ellipsoid.

h	$FDMH_1$ (16) Ratio	$FDMH_2$ (16) Ratio
.05	1.02	1.04
.05/2	1.10	1.03
.05/4	1.00	1.02

Table 5
Errors for Example 3. (capsule region) using a signed distance function.

h	$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$(64)	
	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 6 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 5. 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 $O(h^4)$.

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

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

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

Table 7 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 8 shows the sensitivity to grid shifts for this example. As in Examples 1 and 2, these ratios indicate that the worst case errors are at most a small multiple of the average errors shown in Table 7.

Example 5. 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 9. We include this example because we were unable to construct any multidimensional examples with smooth data that showed less than fourth order accuracy.

Table 6
Errors for Example 3. (capsule region) using a smoother level set function.

h	$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$ (64)	
	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

Table 7
Errors for Example 4. $\Omega \subset \mathbb{R}^2$ is a half disk.

h	$H^{L,\epsilon}$, $\epsilon = h$ (64)		$H^{L,\tilde{\epsilon}}$ (64)		FDMH$_1$ (64)		FDMH$_2$ (64)	
	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 8
Sensitivity to grid shifts for Example 4. $\Omega \subset \mathbb{R}^2$ is a half disk.

h	$FDMH_1$ (64) Ratio	$FDMH_2$ (64) Ratio
.03/2	1.45	2.01
.03/4	1.38	2.48
.03/8	1.45	2.26
.03/16	1.38	2.59

Table 9
Errors for Example 5. One-dimensional example.

h	$FDMH_1$ (512)		$FDMH_2$ (512)	
	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

Example 6. 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 10, 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 $O(h^{1.5})$.

Table 11 shows the sensitivity to grid shifts for this example. The results are similar to the results of the sensitivity tests in Examples 1, 2 and 4. $FDMH_2$ is more sensitive than $FDMH_1$, but in each case, the largest error is only a small multiple of the average error.

Example 7. In this example, we relax the assumption that the level set is a manifold. The integrand is $f(x, y) = 1$, and the level set function is

$$u_1(x, y) = \max(1 - (x - 1)^2 - y^2, 1 - (x + 1)^2 - y^2). \tag{39}$$

The level set $u_1(x, y) = 0$ consists of two circles that touch at a single point. This level set function has a jump in its derivative. In addition, its zero level set $u_1 = 0$ fails to be a manifold, due to the way that the curves are joined together. For comparison, we also consider the level set function

$$u_2(x, y) = \max(1 - (x - 1/2)^2 - y^2, 1 - (x + 1/2)^2 - y^2). \tag{40}$$

Table 10
Errors for Example 6. $\Omega \subset \mathbb{R}^2$ is a disk. f is not smooth.

h	$H^{L,\epsilon}, \epsilon = h$ (128)		$H^{L,\tilde{\epsilon}}$ (64)		$FDMH_1$ (64)		$FDMH_2$ (1024)	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
.05	3.28e−3		9.71e−4		1.05e−3		1.23e−4	
.05/2	1.18e−3	1.47	2.37e−4	2.03	2.60e−4	2.01	3.06e−5	2.01
.05/4	3.83e−4	1.62	6.48e−5	1.87	6.47e−5	2.01	7.70e−6	1.99
.05/8	1.49e−4	1.36	1.67e−5	1.96	1.60e−5	2.02	1.87e−6	2.04

Table 11
Sensitivity to grid shifts for Example 6. $\Omega \subset \mathbb{R}^2$ is a disk. f is not smooth.

h	$FDMH_1$ (64) Ratio	$FDMH_2$ (1024) Ratio
.05	1.96	2.50
.05/2	1.15	2.53
.05/4	1.17	2.53
.05/8	1.17	2.59

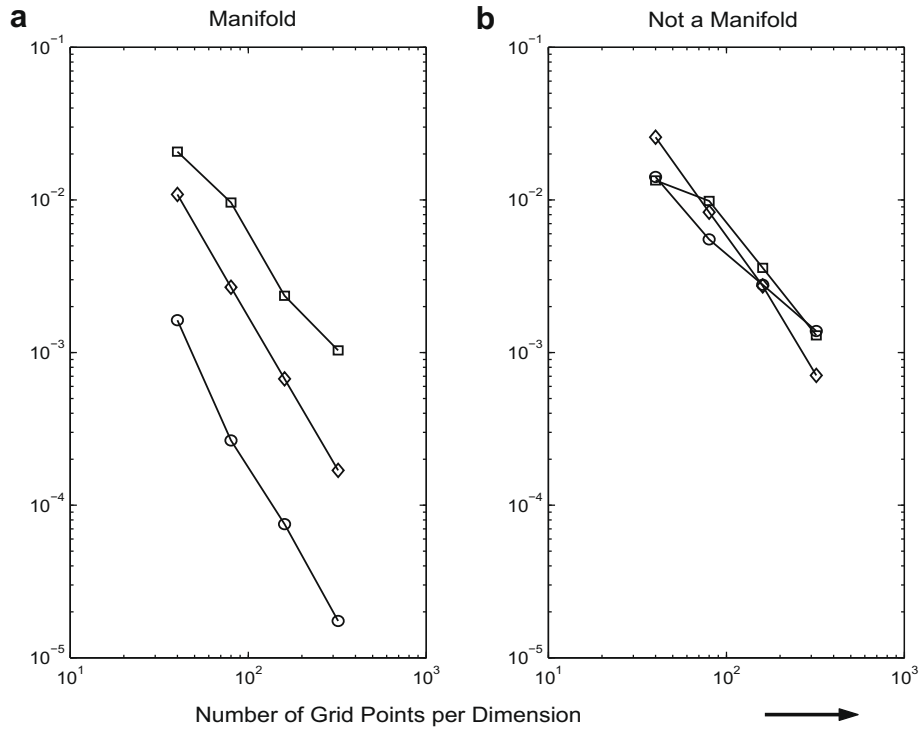

Fig. 2. Example 7. Level set formed by joining two circles. Errors as a function of grid refinement for $FDMH_1$ (diamonds), $FDMH_2$ (circles), $H^{C,\epsilon}$ (squares).

The level set function u_2 also has a jump in its derivative, but because of the way that the two circles are joined together, its zero level set consists of a manifold. We tested $FDMH_1$ and $FDMH_2$, as well as $H^{C,\epsilon}$, with $\epsilon = 1.5h$, recording the maximum errors of a large number of random grid shifts. Fig. 2 shows that there is a large difference in accuracy between the two cases (manifold and non-manifold) for the FDMH algorithms, especially for $FDMH_2$. For $H^{C,\epsilon}$, the convergence rate observed for the

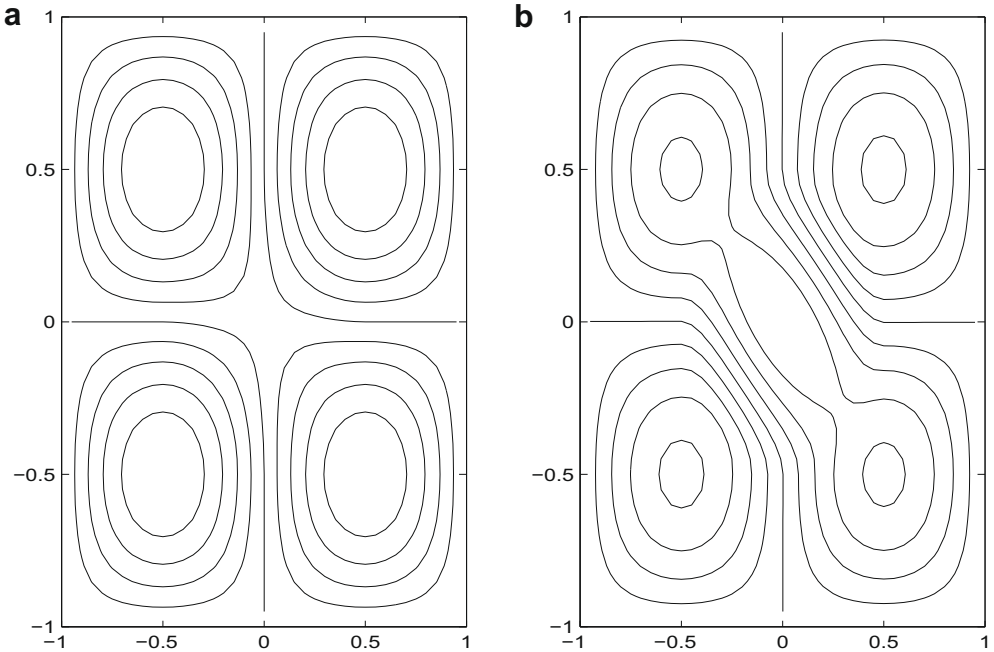

Fig. 3. Example 8. Heat equation. Plot (a) shows the initial data, and plot (b) shows the solution at $t = .125$, computed using $FDMH_1$.

Table 12
Errors for Example 8. Initial-boundary value problem for the heat equation.

h	Time steps	$H^{C,\epsilon}, \epsilon = 1.5h$		$FDMH_1$		$FDMH_2$	
		Max. error	Rate	Max. error	Rate	Max. error	Rate
.025	20	9.479e−4		4.108e−4		1.217e−4	
.025/2	40	2.462e−4	1.95	1.093e−4	1.91	2.970e−5	2.03
.025/4	80	6.266e−5	1.97	2.787e−5	1.98	7.962e−6	1.90
.025/8	160	1.593e−5	1.97	7.136e−6	1.97	2.049e−6	1.96

non-manifold case is not very different from that of the manifold case (about 1.25 in the former compared to 1.5 in the latter). For $FDMH_1$, the convergence rate is 2 for both cases. However, for $FDMH_2$ the convergence rate drops from 2 in the manifold case to 1, or even slightly less, in the non-manifold case.

Example 8. In this example we test our approximate Heaviside functions as a source term in an initial-boundary value problem for the heat equation on the square $[-1,1]\times[-1,1]$. The problem is

$$\begin{cases} v_t = v_{xx} + v_{yy} + 10(2r^2 - 1/4)H(1/4 - r^2), \quad r = \sqrt{x^2+y^2}, \\ v(-1,y,t) = v(1,y,t) = v(x,-1,t) = v(x,1,t) = 0, \\ v(x,y,0) = \sin(\pi x)\sin(\pi y) - \frac{10}{8}(\max(0,1/4 - r^2))^2. \end{cases} \tag{41}$$

The solution of this problem is

$$v(x,y,t) = e^{-2\pi^2 t}\sin(\pi x)\sin(\pi y) - \frac{10}{8}(\max(0,1/4 - r^2))^2. \tag{42}$$

The source term is discontinuous along the circle $r = 1/2$, and the solution has jumps in the second order derivatives along $r = 1/2$. We used the ADI method to approximate the solution, and recorded the maximum errors at $t = .125$ for various mesh sizes. Fig. 3 shows a contour plot of the initial data, and the solution at $t = .125$. In all cases, we used a step size of $\Delta t = h/4$, where $h = \Delta x = \Delta y$. Table 12 shows the maximum (L^∞) errors at the terminal time. All three of the discrete Heaviside functions in this test appear to result in approximately second order convergence rates. Clearly the FDMH algorithms are more accurate than the commonly used $H^{C,\epsilon}$ (with $\epsilon = 1.5h$) approximation. The $FDMH_2$ algorithm gives the best results in this experiment.

Example 9. In this example we solve a Poisson problem on the square $[0,2]\times[0,2]$ with homogeneous Dirichlet boundary conditions:

$$\begin{cases} v_{xx} + v_{yy} = 10(2r^2 - 1/4)H(1/4 - r^2), \quad r = \sqrt{(x-1)^2 + (y-1)^2}, \\ v(-1,y) = v(1,y) = v(x,-1) = v(x,1) = 0. \end{cases} \tag{43}$$

The solution of this problem, which is

$$v(x,y) = \frac{10}{8}(\max(0,1/4 - r^2))^2, \tag{44}$$

is C^1 with jumps in the second order derivatives. We discretized $v_{xx} + v_{yy}$ using the standard five-point difference approximation for the Laplacian, and employed a standard Poisson solver. As in Example 8, we tested $H^{C,\epsilon}$ with $\epsilon = 1.5h$, and both FDMH algorithms. Both of the FDMH approximations give more accurate results than $H^{C,\epsilon}$, as can be seen in Table 13. Moreover, $FDMH_2$ is more accurate than $FDMH_1$. Fig. 4 compares the solutions using all three methods on a coarse mesh. Recalling that the actual solution vanishes for $r > 1/2$, it is evident from Fig. 4 that the solution computed using the FDMH discretizations capture this feature of the solution better than the one using $H^{C,\epsilon}$.

Table 13
Errors for Example 9. Poisson problem with C^1 solution.

h	$H^{C,\epsilon}, \epsilon = 1.5h$		$FDMH_1$		$FDMH_2$	
	Max. error	Rate	Max. error	Rate	Max. error	Rate
.05	3.374e−3		2.117e−3		5.069e−4	
.05/2	8.804e−4	1.93	5.976e−4	1.85	9.740e−5	2.38
.05/4	2.234e−4	1.98	1.598e−4	1.90	2.410e−5	2.01
.05/8	5.686e−5	1.97	4.036e−5	1.99	6.056e−6	1.99

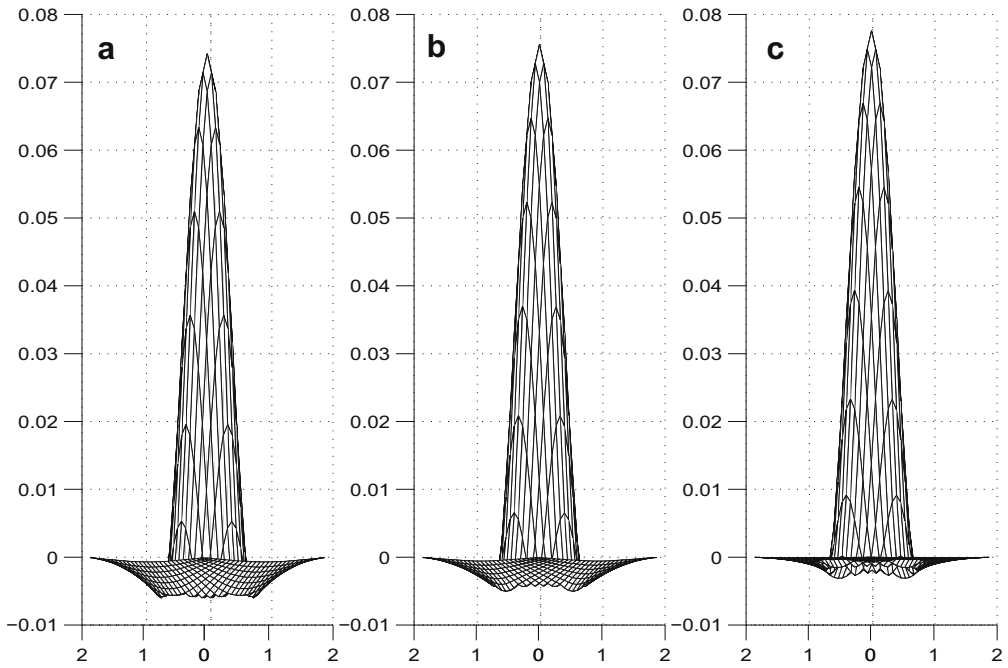

Fig. 4. Example 9. Poisson equation. Side view of solutions on a coarse mesh. Plot (a) shows the solution using $H^{C,\epsilon}$ with $\epsilon = 1.5h$, plot (b) shows the solution using FDMH_1, and plot (c) shows FDMH_2. The maximum of the actual solution is $\approx .078$.

Acknowledgments

I thank the anonymous referees. Their comments helped to improve the paper.

References

[1] D. Adalsteinsson, J. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (1995) 269.
[2] R. Burden, J. Faires, Numerical Analysis, fourth ed., PWS-Kent Publishing Company, Boston, 1989.
[3] G. Dahlquist, A. Björk, Numerical Methods, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.
[4] B. Engquist, A.K. Tornberg, R. Tsai, Discretization of Dirac delta functions in level set methods, J. Comput. Phys. 207 (2005) 28–51.
[5] C. Min, F. Gibou, Geometric integration over irregular domains with application to level-set methods, J. Comput. Phys. 226 (2007) 1432–1443.
[6] C. Min, F. Gibou, Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions, J. Comput. Phys. 227 (2008) 9686–9695.
[7] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2003.
[8] S. Osher, J. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[9] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[10] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, 1999.
[11] A.K. Tornberg, Multi-dimensional quadrature of singular and discontinuous functions, BIT 42 (2002) 644–669.
[12] A.K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2004) 462–488.
[13] J.D. Towers, Two methods for discretizing a delta function supported on a level set, J. Comput. Phys. 220 (2007) 915–931.
[14] J.D. Towers, Discretizing delta functions via finite differences and gradient normalization, J. Comput. Phys. in press, doi:10.1016/j.jcp.2009.02.012.
[15] J.D. Towers, A convergence rate theorem for finite difference approximations to delta functions, J. Comput. Phys. 227 (2008) 6591–6597.