The Numerical Approximation of a Delta Function with Application to Level Set Methods

Peter Smereka
Department of Mathematics, University of Michigan
Ann Arbor, MI 48109

Abstract

It is shown that a discrete delta function can be constructed using a technique developed by Anita Mayo (*SIAM J. Sci. Comput.* **21** 285-299 (1984)) for the numerical solution of elliptic equations with discontinuous source terms. This delta function is concentrated on the zero level set of a continuous function. In two space dimensions this corresponds to a line and a surface in three space dimensions. Delta functions that are first and second order accurate are formulated in both two and three dimensions in terms of a level set function. The numerical implementation of these delta functions achieves the expected order of accuracy.

1 Introduction

Level set and front tracking methods often rely on discrete delta functions for many aspects of their implementation from singular force distributions to the computation of surface area [6, 9, 10, 11, 13]. In many situations the delta function is expressed as a one dimensional function of a level set function or the signed distance function to the interface. However, Tornberg & Engquist[12] show that delta functions expressed in this fashion can fail to correctly compute arc-length and result in poor convergence when solving elliptic equations.

Recently, Engquist et al.[3] and Calhoun & Smereka [2] have developed discrete delta functions which ameliorate the problem outlined in [12]. In both papers, the authors devised simple (but different) expressions for discrete delta functions that are first order accurate. In addition, Engquist et al.[3] also deduce a second order accurate version based on a product of delta functions which is more complex and is only implemented in two dimensions.

In this paper, we provide a detailed derivation of the results contained in [2] and extend the method to second order accuracy in both two and three dimensions. The proposed delta function is relatively straightforward to evaluate and its support is contained within a region whose width is at most that of a single mesh cell.

The approach used in this paper is based on the work of Mayo[7, 8] who developed a technique for solving elliptic problems with discontinuities. Mayo's work has been central in the development of a number of numerical methods for interface problems, see, for example [5, 6].

2 Delta functions

Let us consider a subset of R^2 denoted, Ω, with a closed curve Γ contained inside this region. Let $\delta(d)$ be the usual one dimensional Dirac delta function. Then, if $d(x,y)$ is the signed distance to Γ from a point (x,y) then the arc-length of Γ is given by:

$$L = \int_\Omega \delta(d(x,y))dxdy.$$

We let (x_i, y_j) denote the location of the grid points and let h represent the mesh size. Our goal is to find a discrete version of δ denoted $\widetilde{\delta}_{i,j}$ so that this property is maintained to some order in h. In other words

$$L = \sum_{i,j} h^2 \widetilde{\delta}_{i,j} + O(h^p). \tag{1}$$

where $p > 0$. We shall provide expressions for $\widetilde{\delta}_{i,j}$ that are both first and second order accurate. We also would like to use this discrete delta function in the evaluation line and surface integrals. That is to say, if

$$I = \int_\Gamma f(x,y)ds = \int_\Omega f(x,y)\delta(d(x,y))dxdy \tag{2}$$

then we expect our discrete delta function to have the property

$$I = \sum_{i,j} h^2 \widetilde{\delta}_{i,j} f_{i,j} + O(h^p). \tag{3}$$

Our computations seem to indicate that (3) will be second order provided that one uses a second order accurate delta function, see the results presented in Tables 3 and 5.

3 Discrete Green's and delta functions

The Green's function for Laplace's equation in one space dimension on the unit interval satisfies

$$\frac{d^2g}{dx^2} = \delta(x-\alpha) \quad \text{with} \quad g(0) = g(1) = 0 \tag{4}$$

where $\delta(x)$ is the one-dimensional delta function and $0 < \alpha < 1$. It is well known that this can be replaced by the equivalent problem

$$g''(x) = 0 \quad \text{with} \quad g(0) = g(1) = 0 \tag{5}$$

subject to the jump conditions at $x = \alpha$:

$$[g'(\alpha)]_x = 1 \quad \text{and} \quad [g(\alpha)]_x = 0, \tag{6}$$

where $[q(\alpha)]_x$ denotes the jump in q in the x-direction. More precisely, $[q(\alpha)]_x = q(\alpha^+) - q(\alpha^-)$ with $q(\alpha^\pm) = \lim_{\varepsilon\to 0^+} q(\alpha \pm \varepsilon)$. The notation, $[]_x$, may seem a little strange but it will prove useful in our extension to two dimensions.

Mayo [7, 8] introduced a numerical method to solve problems like (5-6) by finding the corresponding discrete version of (4); thereby yielding a discrete delta function. For the convenience of the reader we will review Mayo's work and consider the discretization of (5-6) in which the jumps are located at $x = \alpha$. This is done as follows. We let x_i denote the location of the grid points and let h represent the mesh size. We define two types of grid points. The first type are *irregular points* (after Mayo [7]). These are grid points that are within one mesh spacing of the interface; in other words if the interface is between x_i and x_{i+1} then points x_i and x_{i+1} are irregular points. All the other grid points are denoted as *regular points.* For regular points one has the center differenced approximation for the second derivative

$$g''(x_i) = \frac{g(x_{i+1}) - 2g(x_i) + g(x_{i-1})}{h^2} + O(h^2). \tag{7}$$

Next, a finite difference approximation for $g''(x_i)$ will be formulated when x_i is an irregular point. First consider the situation when $x_{i-1} < \alpha < x_i$. From a Taylor series expansion one has

$$g(x_{i-1}) = g(\alpha^-) - h_1 g'(\alpha^-) + \frac{h_1^2}{2} g''(\alpha^-) + O(h^3) \tag{8}$$

and

$$g(x_i) = g(\alpha^+) + h_2 g'(\alpha^+) + \frac{h_2^2}{2} g''(\alpha^+) + O(h^3) \tag{9}$$

where $h_1 = \alpha - x_{i-1}$ and $h_2 = x_i - \alpha$. It also follows from a Taylor series expansion that

$$g'(\alpha^+) = g'(x_i) - h_2 g''(x_i) + O(h^2) \quad \text{and} \quad g''(\alpha^+) = g''(x_i) + O(h). \tag{10}$$

One can use (10) in (9) and the fact that $h_1 + h_2 = h$ to obtain

$$g(x_i) = g(\alpha^+) - h_1 g'(\alpha^+) + \frac{h_1^2}{2} g''(\alpha^+) + h g'(x_i) - \frac{h^2}{2} g''(x_i) + O(h^3). \tag{11}$$

Combining (8) and (11) we find

$$\begin{aligned} g(x_{i-1}) - g(x_i) &= -h g'(x_i) + \frac{h^2}{2} g''(x_i) \\ &- [g(\alpha)]_x + h_1 [g'(\alpha)]_x - \frac{h_1^2}{2} [g''(\alpha)]_x + O(h^3). \end{aligned} \tag{12}$$

Since there are no jumps in g for $x_i > \alpha$ then

$$g(x_{i+1}) - g(x_i) = h g'(x_i) + \frac{h^2}{2} g''(x_i) + O(h^3). \tag{13}$$

Adding (12) and (13) and manipulating we find

$$\begin{aligned} g''(x_i) &= \frac{g(x_{i+1}) - 2g(x_i) + g(x_{i-1})}{h^2} \\ &+ \frac{1}{h^2} \left([g(\alpha)]_x - h_1 [g'(\alpha)]_x + \frac{h_1^2}{2} [g''(\alpha)]_x \right) + O(h). \end{aligned} \tag{14}$$

For the problem at hand $[g(\alpha)]_x = 0$, $[g'(\alpha)]_x = 1$, $[g''(\alpha)]_x$=0 and we find (14) becomes

$$g''(x_i) = \frac{g(x_{i+1}) - 2g(x_i) + g(x_{i-1})}{h^2} - \frac{h_1}{h^2} + O(h). \tag{15}$$

A similar analysis can be completed when $x_i < \alpha < x_{i+1}$; combining this with (15) we have, for irregular points, the following

$$g''(x_i) = \frac{g_{i+1} - 2g_i + g_{i-1}}{h^2} - \tilde{\delta}_i + O(h) \tag{16}$$

where

$$\tilde{\delta}_i = \tilde{\delta}_i^+ + \tilde{\delta}_i^- \tag{17}$$

and

$$\tilde{\delta}_i^+ = \begin{cases} (x_{i+1} - \alpha)/h^2 & \text{if} \quad x_i \le \alpha < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{\delta}_i^- = \begin{cases} (\alpha - x_{i-1})/h^2 & \text{if} \quad x_{i-1} < \alpha < x_i \\ 0 & \text{otherwise.} \end{cases}$$

We combine (7) and (16) to obtain, for all grid points, the following

$$g''(x_i) = \frac{g_{i+1} - 2g_i + g_{i-1}}{h^2} - \tilde{\delta}_i + O_I(h) + O(h^2) \tag{18}$$

where $O_I(\cdot)$ denotes errors that occur only at irregular points.

For the problem of interest $g''(x_i) = 0$ and we use (18) to obtain the following finite difference approximation

$$\frac{g_{i+1} - 2g_i + g_{i-1}}{h^2} = \tilde{\delta}_i \tag{19}$$

Upon comparing (19) to (4) and we infer that (17) is a discrete delta function.

One can verify that if α does not lie exactly on a grid point, the function $\tilde{\delta}_i$ is nonzero only at the two grid points x_i, x_{i+1} for which $x_i < \alpha < x_{i+1}$. If α lies exactly on a grid point, then $\tilde{\delta}_i$ is non-zero only at $x_i = \alpha$. Furthermore, for any α, $0 < \alpha < 1$, we have

$$\sum_i \tilde{\delta}_i h = 1. \tag{20}$$

We recall that a delta function has the following property

$$\int_0^1 \delta(x - y) dy = 1$$

for $0 < x < 1$; (20) is a discrete version of this property. We point out that $\tilde{\delta}$ is the same as δ_h (defined by (55)) in one dimension but as we shall see they differ in two dimensions.

4 Extension to two dimensions

In this section we extend the previous results to two dimensions. Let $\boldsymbol{n}$ be the outward drawn unit normal vector to Γ and $[q]$ be the jump across Γ; more precisely

$$[q] = \lim_{\varepsilon \to 0^+} q(\boldsymbol{\alpha} + \varepsilon \boldsymbol{n}) - q(\boldsymbol{\alpha} - \varepsilon \boldsymbol{n}) \tag{21}$$

where $\boldsymbol{\alpha} = (\alpha_x, \alpha_y)$ is a point on the interface. It also useful to define $[]_x$, the jump in the x-direction and $[]_y$ the jump in the y-direction as

$$[q]_x = q(\alpha_x^+, \alpha_y) - q(\alpha_x^-, \alpha_y) \quad \text{and} \quad [q]_y = q(\alpha_x, \alpha_y^+) - q(\alpha_x, \alpha_y^-). \tag{22}$$

It is easy to verify that (21) and (22) are related as follows

$$[q]_x = [q]\,\mathrm{sgn}[n_x] \quad \text{and} \quad [q]_y = [q]\,\mathrm{sgn}[n_y]. \tag{23}$$

As was done in one dimension, we devise the discrete delta function by considering the elliptic problem

$$\Delta g(x, y) = \delta(d(x, y)) \tag{24}$$

where $\Delta = \partial_x^2 + \partial_y^2$. Let us now, using (21), rewrite (24) as

$$\Delta g(x, y) = 0 \tag{25}$$

subject to the jump conditions at $\boldsymbol{x} = \Gamma$

$$[\partial_n g] = 1 \quad \text{and} \quad [g] = 0 \tag{26}$$

where ∂_n is the directional derivative in the normal direction. We shall deduce the delta function by applying the results of the previous section.

First we observe that for regular points we have the standard center differenced approximation

$$\Delta g(x_i, y_i) = \Delta_h g_{i,j} + O(h^2) \tag{27}$$

where (x_i, y_j) are the grid locations and Δ_h is the discrete five-point Laplacian, namely

$$\Delta_h g_{i,j} = \frac{g_{i+1,j} - 2g_{i,j} + g_{i-1,j}}{h^2} + \frac{g_{i,j+1} - 2g_{i,j} + g_{i,j-1}}{h^2}. \tag{28}$$

To obtain an expression similar to (27) for irregular points we apply (14) in both the x and y directions while recognizing that $[g] = 0$ implies $[g]_x = [g]_y = 0$. In the x-direction the corresponding Taylor series expansion is about the point (α_x, y_j) and in the y-direction it is about the point (x_i, α_y). Both of these points are on the interface. This yields an expression for the discrete Laplacian of g when g has discontinuities in its first and second derivative along an interface. We have for irregular points

$$\Delta g(x_i, y_i) = \Delta_h g_{i,j} - \widetilde{\delta}_{i,j} + O(h) \tag{29}$$

where

$$\widetilde{\delta}_{i,j} = \widetilde{\delta}_{i,j}^{(+x)} + \widetilde{\delta}_{i,j}^{(-x)} + \widetilde{\delta}_{i,j}^{(+y)} + \widetilde{\delta}_{i,j}^{(-y)} \tag{30}$$

with

$$h^2 \widetilde{\delta}_{i,j}^{(+x)} = \begin{cases} h_x^+ [\partial_x g]_x + \frac{1}{2}(h_x^+)^2 [\partial_{xx}^2 g]_x & \text{if} \quad x_i \le \alpha_x < x_{i+1} \\ 0 & \text{otherwise,} \end{cases}$$

$$h^2 \widetilde{\delta}_{i,j}^{(-x)} = \begin{cases} h_x^- [\partial_x g]_x - \frac{1}{2}(h_x^-)^2 [\partial_{xx}^2 g]_x & \text{if} \quad x_{i-1} < \alpha_x < x_i \\ 0 & \text{otherwise,} \end{cases}$$

$$h^2 \widetilde{\delta}_{i,j}^{(+y)} = \begin{cases} h_y^+ [\partial_y g]_y + \frac{1}{2}(h_y^+)^2 [\partial_{yy}^2 g]_y & \text{if} \quad y_j \le \alpha_y < y_{j+1} \\ 0 & \text{otherwise,} \end{cases}$$

$$h^2 \widetilde{\delta}_{i,j}^{(-y)} = \begin{cases} h_y^+ [\partial_y g]_y - \frac{1}{2}(h_y^-)^2 [\partial_{yy}^2 g]_y & \text{if} \quad y_{j-1} < \alpha_y < y_j \\ 0 & \text{otherwise.} \end{cases}$$

We also have $h_x^+ = x_{i+1} - \alpha_x$ and $h_x^- = \alpha_x - x_{i-1}$. h_y^- and h_y^+ are defined in a similar fashion. We point out that $[\cdot]_x$ terms are evaluated at (α_x, y_j). In a similar way $[\cdot]_y$ terms are evaluated at (x_i, α_y).

Since $\Delta g(x_i, y_j) = 0$ it follows from (27) and (29) that

$$\Delta_h g_{i,j} = \widetilde{\delta}_{i,j} + O_I(h) + O(h^2) \tag{31}$$

for all grid points. The reader is reminded $O_I(\cdot)$ indicates a term that is only nonzero at irregular points. This is the discrete form of (24) and the right hand side represents the discrete delta function plus error terms. It is easy to extend (30) to three space dimensions.

5 Order of Accuracy

A discrete delta function said to have an order of accuracy p if

$$I = \sum_{i,j} h^2 \tilde{\delta}_{i,j} f_{i,j} + O(h^p) \quad \text{as} \quad h \to 0 \tag{32}$$

where I is given by (2). For example, if we consider the one dimension version of (32) and use the delta function given by (17) then one can prove that $p = 2$; see Ref.[1].

It would be natural to rigorously establish the order of accuracy of the approximate delta function given by (30). This appears to be a difficult task and will not be undertaken here. Instead, we shall provide a heuristic argument for the order of accuracy by considering the calculation of the arc-length. Of course, this merely provides an upper bound on the order of accuracy since (32) is a much more stringent test. Nevertheless, the argument below will give us some idea of what to expect.

We begin the discussion by observing that (30) can be written in the form

$$\tilde{\delta}_{i,j} = \frac{a}{h} + b \tag{33}$$

where a and b are somewhat complicated functions determined from (30). The important feature is that a and b are $O(1)$ in h. It practice a and b will be determined to some level of accuracy in h and we write

$$a = a_m + O(h^m) \quad \text{and} \quad b = b_m + O(h^m).$$

The computation of a_m and b_m will be explained in §§6 and 7. We substitute these expressions into (33) to obtain

$$\tilde{\delta}_{i,j} = \tilde{\delta}^{(m)}_{i,j} + O(h^m) \tag{34}$$

where

$$\tilde{\delta}_{i,j} = \frac{a_{m+1}}{h} + b_m. \tag{35}$$

Eq. (34) is used to rewrite (31) as

$$\Delta_h g_{i,j} = \tilde{\delta}^{(m)}_{i,j} + O_I(h^m) + O_I(h) + O(h^2). \tag{36}$$

We recall that

$$\int_\Omega \Delta g dx dy = \int_\Omega \delta(d(x,y)) dx dy = L.$$

The discrete version of the above expression is

$$L_h = \sum_{i,j} h^2 \Delta_h g_{i,j} = \sum_{i,j} h^2 \tilde{\delta}_{i,j}^{(m)} + E_h,$$

where

$$E_h = \sum_{i,j} h^2 \left(O_I(h^m) + O_I(h) + O(h^2) \right). \tag{37}$$

We conjecture that

$$\lim_{h \to 0} \sum_{i,j} h^2 \tilde{\delta}_{i,j}^{(m)} = L$$

and that the rate of convergence is determined by the error term, E_h. Let us examine E_h in more detail. We first observe that there $O(h^{-2})$ grid points. Since the interface is a one dimensional object then there are $O(h^{-1})$ irregular points. The $O(h^2)$ terms in (37) are present at all grid points whereas any O_I term is only nonzero at irregular points. Combining these observations with (37) one obtains

$$E_h = O(h^2) + O(h^{m+1}). \tag{38}$$

Therefore, it follows that the arc-length can be computed to second order accuracy if we evaluate a to second order accuracy and b to first order accuracy. In addition, it follows if we evaluate a to first order accuracy and ignore the b term then the arc-length computation will be first order accurate. We stress that these are heuristic arguments and only establish a plausible order of accuracy for the computation of the arc-length. The numerical results presented in §8 confirm these conjectures. In addition, the numerical results also provide evidence that the order of accuracy is maintained for integrals of the form given by (2).

6 First Order Implementation

It follows from (30), that if we neglect the jumps in the second derivatives and evaluate $[\partial_x g]_x$ and $[\partial_y g]_y$ to first order then $m = 0$. The argument in

§5 implies this should result in a first order method. The necessary jumps can be computed as follows. Let $\boldsymbol{n} = (n_x, n_y)$ be the unit normal vector and $\boldsymbol{s} = (s_x, s_y)$ be a unit vector tangent to Γ. We have the following directional derivatives

$$\partial_n = n_x \partial_x + n_y \partial_y \tag{39}$$

and

$$\partial_s = s_x \partial_x + s_y \partial_y. \tag{40}$$

For the problem at hand, $[\partial_n g] = 1$ and $[g] = 0$. The latter implies $[\partial_t g] = 0$. Combining $[\partial_n g] = 1$ and $[\partial_t g] = 0$ with (39) and (40) yields

$$[g_x] = n_x \quad \text{and} \quad [g_y] = n_y. \tag{41}$$

Eq (41) and (23) produce

$$[g_x]_x = |n_x| \quad \text{and} \quad [g_y]_y = |n_y|. \tag{42}$$

We will represent the interface as the zero level set of $\phi(x, y)$ and assume that we have a given discretization $\phi_{i,j} \equiv \phi(x_i, y_j)$ on a grid of mesh size h. We make the following definitions

$$D_x^+ \phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{h}, \quad D_x^- \phi_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{h}$$

$$\text{and} \quad D_x^0 \phi_{i,j} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h};$$

$D_y^+ \phi_{i,j}$, $D_y^- \phi_{i,j}$, and $D_y^0 \phi_{i,j}$ are analogously defined. It is convenient to make the definition

$$|\nabla_0^\varepsilon \phi_{i,j}| = \sqrt{(D_x^0 \phi_{i,j})^2 + (D_y^0 \phi_{i,j})^2 + \varepsilon}$$

where ε is a small number that prevents division by zero. In the computations presented in §8 we take $\varepsilon = 10^{-10}$. We will now use the level set implementation and the above definitions to deduce a first order approximation to (30). By using a Taylor series it is easy for one to establish

$$h_x^\pm = \left| \frac{\phi_{i\pm1,j}}{D_x^\pm \phi_{i,j}} \right| + O(h^2) \quad \text{and} \quad h_y^\pm = \left| \frac{\phi_{i,j\pm1}}{D_y^\pm \phi_{i,j}} \right| + O(h^2). \tag{43}$$

In addition it is straightforward to verify

$$n_x = \frac{\partial_x \phi}{|\nabla \phi|} = \frac{D_x^0 \phi_{i,j}}{|\nabla_0^\varepsilon \phi_{i,j}|} + O(h) \quad \text{and} \quad n_y = \frac{\partial_y \phi}{|\nabla \phi|} = \frac{D_y^0 \phi_{i,j}}{|\nabla_0^\varepsilon \phi_{i,j}|} + O(h). \quad (44)$$

These are of second order accuracy at grid points; however, since they are used at the interface their accuracy is first order.

If we use the approximations given by (43) and (44) in (30) and ignore the jumps in the second derivatives then we arrive at the following first order expression for the discrete delta function

$$\widetilde{\delta}(\phi_{i,j}) = \widetilde{\delta}_{i,j}^{(+x)} + \widetilde{\delta}_{i,j}^{(-x)} + \widetilde{\delta}_{i,j}^{(+y)} + \widetilde{\delta}_{i,j}^{(-y)}. \quad (45)$$

where

$$\widetilde{\delta}_{i,j}^{(+x)} = \begin{cases} \dfrac{|\phi_{i+1,j} D_x^0 \phi_{i,j}|}{h^2 |D_x^+ \phi_{i,j}| |\nabla_0^\varepsilon \phi_{i,j}|} & \text{if} \quad \phi_{i,j}\phi_{i+1,j} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\widetilde{\delta}_{i,j}^{(-x)} = \begin{cases} \dfrac{|\phi_{i-1,j} D_x^0 \phi_{i,j}|}{h^2 |D_x^- \phi_{i,j}| |\nabla_0^\varepsilon \phi_{i,j}|} & \text{if} \quad \phi_{i,j}\phi_{i-1,j} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$\widetilde{\delta}_{i,j}^{(+y)} = \begin{cases} \dfrac{|\phi_{i,j+1} D_y^0 \phi_{i,j}|}{h^2 |D_y^+ \phi_{i,j}| |\nabla_0^\varepsilon \phi_{i,j}|} & \text{if} \quad \phi_{i,j}\phi_{i,j+1} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\widetilde{\delta}_{i,j}^{(-y)} = \begin{cases} \dfrac{|\phi_{i,j-1} D_y^0 \phi_{i,j}|}{h^2 |D_y^- \phi_{i,j}| |\nabla_0^\varepsilon \phi_{i,j}|} & \text{if} \quad \phi_{i,j}\phi_{i,j-1} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

A first order expression in three space dimensions is easy to infer from the above result.

Remark. The above implementation is slightly different from the one presented in Ref [2]. It appears to be marginally more accurate.

7 Second Order Implementation

In this section, we extend the formulation to second order accuracy. As we shall see the extension to three dimensions from two dimensions is not as straightforward as it is in the first order case. In both cases, the main issue is the computation of the jumps in the second derivatives of g.

7.1 Two dimensional case

To deduce a second order accurate discrete delta function from (30) we must deduce $h_x^{\pm}$ and $h_y^{\pm}$ to third order and compute the jumps in the second derivatives to first order. We begin with the latter. It follows from (25) that

$$[\partial^2_{xx} g + \partial^2_{yy} g] = 0$$

and from the jump conditions that

$$[\partial^2_{sn} g] = [\partial^2_{ss} g] = 0.$$

From the last three equations and (41) we can deduce the following system

$$\begin{pmatrix} s_x n_x & s_x n_y + s_y n_x & s_y n_y \\ s_x^2 & 2 s_y s_x & s_y^2 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} [\partial^2_{xx} g] \\ [\partial^2_{xy} g] \\ [\partial^2_{yy} g] \end{pmatrix} + \begin{pmatrix} \boldsymbol{n} \cdot \partial_s \boldsymbol{n} = 0 \\ \boldsymbol{n} \cdot \partial_s \boldsymbol{s} \\ 0 \end{pmatrix} = 0.$$

The above system is solved to obtain

$$[\partial^2_{xx} g] = -D \quad \text{and} \quad [\partial^2_{yy} g] = D$$

where

$$D = (s_x n_y + s_y n_x) \boldsymbol{n} \cdot \partial_s \boldsymbol{s}. \tag{46}$$

Finally we note that

$$[\partial^2_{xx} g]_x = -D \operatorname{sgn}(n_x) \quad \text{and} \quad [\partial^2_{yy} g]_y = D \operatorname{sgn}(n_y). \tag{47}$$

Expressions (42) and (47) will be used in (30). In our implementation we use the following tangential derivative $\boldsymbol{s} = (-n_y, n_x)$. When evaluating the derivatives in (47) we use center differences.

Now we shall discuss how to find $h_x^\pm$ and $h_y^\pm$ to third order. It is sufficient to describe this in one dimension since the computation is done direction-by-direction. We present the derivation of h_x^+; the other expressions can be deduced in a similar fashion. Since we are concerned with h_x^+, the case when the interface is between x_i and x_{i+1} must be considered. We will expand ϕ in a Taylor series expansion about the center of this cell,

$$\phi(u) = \phi_c + \phi_c' u + \frac{1}{2}\phi_c'' u^2 + O(h^3) \tag{48}$$

where $u = x - (x_i + \frac{h}{2})$ and the subscript denotes the value at the cell center. The interface is located by the relation $\phi(u_I) = 0$ which we solve to determine u_I; we obtain

$$u_I = -\frac{\phi_c}{\phi_c'} - \frac{\phi_c''\phi_c^2}{2\phi_c'^3} + O(h^3).$$

Since u_I is measured with respect to the cell center then it follows that

$$h_x^+ = \frac{h}{2} - u_I$$

In the numerical implementation of the above formula we use the following approximations

$$\phi_c = \frac{1}{16}(-\phi_{i-1} + 9\phi_i + 9\phi_{i+1} - \phi_{i+2}) + O(h^4),$$

$$\phi_c' = \frac{\phi_{i+1} - \phi_i}{h} + O(h^2),$$

and

$$\phi_c'' = \frac{\phi_{i-1} - (\phi_i + \phi_{i+1}) + \phi_{i+2}}{2h^2} + O(h^2).$$

When the interface is not properly resolved by the grid then the above formula can fail. Therefore we modify it as follows

$$h_x^+ = \begin{cases} \dfrac{h}{2} - u_I & \text{if} \quad |u_I| < h/2 \\ \left|\dfrac{\phi_{i+1}}{D_x^+\phi_i}\right| & \text{otherwise.} \end{cases} \tag{49}$$

We observe that the provisional case in the above formula is just the first order expression used in the previous section.

To obtain a second order delta function one starts with (30). For the jumps in the first derivative one uses (42) and for the jumps in the second derivative one uses (47). All derivatives are computed using center differenced approximations. This yields approximations that are first order at the interface for the second derivative jump terms. Third order accurate expressions for $h_x^{\pm}$ and $h_y^{\pm}$ (e.g. (49)) are used in (30). In addition, it would appear that we need a second order approximations for n_x and n_y to evaluate $[\partial_x g]_x$ and $[\partial_y g]_y$, however (44) is sufficient as one can show the $O(h)$ terms cancel when computing (30). Finally, the conditionals contained in (30) are implemented as in (45). This yields a second order accurate discrete delta function.

Remark. Performing the Taylor series expansion in (48) about the center of the cell will ensure $h_x^+ + h_x^- = h$ which improves the accuracy.

7.2 Three dimensional case

The above result can be extended to three dimensions in a relatively straightforward fashion. The main difference is that we need to introduce another tangent vector which we will denote as $\boldsymbol{t}$ and assume that $\{\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}\}$ forms an orthonormal triad at each point on the surface. We shall use the following directional derivatives

$$\partial_n = n_x \partial_x + n_y \partial_y + n_z \partial_z, \tag{50}$$

$$\partial_s = s_x \partial_x + s_y \partial_y + s_z \partial_z, \tag{51}$$

and

$$\partial_t = t_x \partial_x + t_y \partial_y + t_z \partial_z. \tag{52}$$

It follows from the jump conditions (26) that

$$[\partial_n g] = 1 \quad \text{and} \quad [\partial_t g] = [\partial_s g] = 0$$

from which one can deduce

$$[\nabla g] = \boldsymbol{n}. \tag{53}$$

To obtain second order accuracy we need to find the jumps in the second derivatives. These are determined by noting that

$$[\partial_{sn}^2 g] = [\partial_{tn}^2 g] = [\partial_{st}^2 g] = [\partial_{ss}^2 g] = [\partial_{tt}^2 g] = 0.$$

The above relations along with $[g_{xx}+g_{yy}+g_{zz}]=0$ and (53) allow to us deduce a set of equations for the jumps in the second derivatives, namely

$$\begin{pmatrix} s_x n_x & s_x n_y + s_y n_x & s_x n_z + s_z n_x & s_y n_y & s_y n_z + s_z n_y & s_z n_z \\ t_x n_x & t_x n_y + t_y n_x & t_x n_z + t_z n_x & t_y n_y & t_y n_z + t_z n_y & t_z n_z \\ s_x t_x & s_x t_y + s_y t_x & s_x t_z + s_z t_x & s_y t_y & s_y t_z + s_z t_y & s_z t_z \\ s_x^2 & 2 s_y s_x & 2 s_x s_z & s_y^2 & 2 s_y s_z & s_z^2 \\ t_x^2 & 2 t_y t_x & 2 t_x t_z & t_y^2 & 2 t_y t_z & t_z^2 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} [\partial^2_{xx} g] \\ \left[\partial^2_{xy} g\right] \\ [\partial^2_{xz} g] \\ \left[\partial^2_{yy} g\right] \\ \left[\partial^2_{yz} g\right] \\ [\partial^2_{zz} g] \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \boldsymbol{n}\cdot\partial_s \boldsymbol{t} \\ \boldsymbol{n}\cdot\partial_s \boldsymbol{s} \\ \boldsymbol{n}\cdot\partial_t \boldsymbol{t} \\ 0 \end{pmatrix} = 0.$$

There are many choices for the tangential vectors, we use the following:

$$\begin{aligned} \boldsymbol{s} &= \begin{cases} \dfrac{(0, n_z, -n_y)^T}{\sqrt{n_y^2+n_z^2}} & \text{if} \quad n_x^2+n_y^2 < \frac{1}{4} \\ \dfrac{(n_y, -n_x, 0)^T}{\sqrt{n_x^2+n_y^2}} & \text{otherwise,} \end{cases} \\ \boldsymbol{t} &= \boldsymbol{n}\times\boldsymbol{s}. \end{aligned}$$

The vectors, $\{\boldsymbol{n}, \boldsymbol{s}, \boldsymbol{t}\}$, form an orthonormal triad at each point on the surface. With these vectors, the above system can be solved to obtain $[\partial^2_{xx} g]$, $[\partial^2_{yy} g]$, and $[\partial^2_{zz} g]$. One can obtain an analytical solution but it is rather unwieldy so we used Gaussian elimination (numerically) instead. Once the jumps are known the discrete delta function can be computed using a natural extension of (30) to three dimensions.

8 Results

The first example we consider is motivated by the work of Tornberg & Engquist[12]. In that work they show that one dimensional delta functions of distance functions can fail. In more detail, consider the discrete approximation to the arc-length given by:

$$L_{h,w} = \sum_{i,j} \delta_w(d_{i,j}) h^2 \tag{54}$$

where

$$\delta_w(x) = \begin{cases} (w - |x|)/w^2 & |x| < w \\ 0 & |x| \geq w \end{cases} \tag{55}$$

and $d_{i,j}$ is the signed distance from a grid point to the interface. Tornberg & Engquist prove that for this type of delta function, there can be $O(1)$ errors in the approximation of $L_{h,w}$ to L. In particular, they prove that if the interface is exactly at 45° to the grid then one finds

$$\lim_{h \to 0} \frac{L_{h,h}}{L} \approx 1.12 \quad \text{and} \quad \lim_{h \to 0} \frac{L_{h,2h}}{L} \approx 1.018.$$

Consider a square whose sides are at 45° to the grid and are of length $\sqrt{2}$, then the signed distance function is:

$$d(x,y) = \begin{cases} \theta(x,y) & \text{if} \quad |x-y| > 1 \quad \text{and} \quad |x+y| > 1 \\ \psi(x,y) & \text{otherwise} \end{cases}$$

where

$$\theta(x,y) = \min\left(\sqrt{(x \pm 1)^2 + y^2}, \sqrt{x^2 + (y \pm 1)^2}\right)$$

and

$$\psi(x,y) = (|x| + |y| - 1)/\sqrt{2}.$$

If we use (54) then one can show (see [2]) that the formula does not converge and the relative error is approximately .12, for all values of h: in agreement with [12]. On the other hand, we observe that both the first and second order implementation of our delta function converge to the correct answer. Due the presence of the corners we cannot obtain second order accuracy with the second order method.

The next example we will discuss is the computation of the arc-length of an ellipse. For this we take

$$\phi = \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1.$$

In our computations we take $a = 1.5$ and $b = .75$. The arc-length can be computed directly and the exact answer is approximately 7.266336165. The results of our computation are shown in Table 2. In these computations we have presented the average over 50 trials in which the ellipse has been shifted

Results for the Square				
	first order		second order	
mesh size	relative error	order	relative error	order
.2	5.85×10^{-2}		9.33×10^{-3}	
.1	2.93×10^{-2}	1.00	4.67×10^{-3}	1.00
.05	1.46×10^{-2}	1.00	2.33×10^{-3}	1.00
.025	7.32×10^{-3}	1.00	1.17×10^{-3}	1.00
.0125	3.66×10^{-3}	1.00	5.83×10^{-4}	1.00
.00625	1.83×10^{-3}	1.00	2.92×10^{-4}	1.00

Table 1: Computation of the arc-length for the square using both the first order and second order discrete delta function

Results for the Ellipse				
	first order		second order	
mesh size	relative error	order	relative error	order
.2	9.38×10^{-3}		9.33×10^{-3}	
.1	2.23×10^{-3}	2.07	4.67×10^{-4}	2.29
.05	8.12×10^{-4}	1.46	2.33×10^{-4}	2.05
.025	2.71×10^{-4}	1.58	1.17×10^{-5}	2.18
.0125	7.58×10^{-5}	1.83	5.83×10^{-6}	1.90
.00625	3.04×10^{-5}	1.32	2.92×10^{-6}	2.08

Table 2: Computation of the arc-length for an ellipse using both the first order and second order discrete delta functions

Results for the Line Integral				
	first order		second order	
mesh size	relative error	order	relative error	order
.2	9.83×10^{-3}		1.23×10^{-2}	
.1	3.94×10^{-3}	1.31	3.13×10^{-3}	2.05
.05	1.78×10^{-3}	1.14	7.78×10^{-3}	2.01
.025	5.57×10^{-4}	1.68	1.96×10^{-4}	1.99
.0125	2.18×10^{-4}	1.35	4.92×10^{-5}	1.99
.00625	7.49×10^{-5}	1.60	1.22×10^{-5}	2.01
.003125	2.62×10^{-5}	1.46	3.05×10^{-6}	2.00

Table 3: Computation of the line integral (56) using both the first order and second order discrete delta functions

in the x and y directions and rotated by random amounts. The first order results are slightly better than first order whereas the computations using the second order algorithm are clearly second order.

Now, we consider the computation of the line integral:

$$\int_{x^2+y^2=1} f(x,y)ds \tag{56}$$

where $f(x,y) = 3x^2 - y^2$. The exact answer is 2π. This was computed by first evaluating the discrete delta function with $\phi = x^2 + y^2 - 1$ and then using (3). The results are shown in Table 3. These results are averaged over 50 trials, where, in each case the grid was shifted in the x and y directions by random amounts. The results are consistent with the expected order of accuracy.

The third example we consider is the computation of the surface area of an ellipsoid. For this we take

$$\phi = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1$$

In our computations we take $a = 1.5$, $b = .75$, and $c = .5$. Here, the surface area can be computed accurately using numerical quadrature and its value is approximately is 9.901821. Table 4 presents the results of our computations in which the ellipsoid has been rotated (using the 3 Euler

Results for the Ellipsoid				
	first order		second order	
mesh size	relative error	order	relative error	order
.2	2.75×10^{-2}		7.00×10^{-3}	
.1	6.81×10^{-3}	2.01	9.69×10^{-4}	2.85
.05	1.71×10^{-4}	1.99	1.79×10^{-4}	2.43
.025	4.32×10^{-4}	1.99	4.08×10^{-5}	2.13
.0125	1.20×10^{-4}	1.86	9.30×10^{-6}	2.13

Table 4: Computation of the surface area of an ellipsoid using both the first order and second order discrete delta functions

Results for the Surface Integral				
	first order		second order	
mesh size	relative error	order	relative error	order
.2	1.68×10^{-2}		1.24×10^{-2}	
.1	2.92×10^{-3}	2.52	5.10×10^{-4}	4.61
.05	7.60×10^{-4}	1.95	1.39×10^{-4}	1.87
.025	1.53×10^{-4}	2.31	3.39×10^{-5}	2.04
.0125	5.12×10^{-5}	1.57	8.42×10^{-6}	2.01

Table 5: Computation of the surface integral using both the first and second order discrete delta functions

angles) and translated in each coordinate direction by random amounts. We have used 20 trials. The results clearly demonstrate that the method is consistent with the expected order of accuracy.

The last example we consider is the computation of the following surface integral

$$\int_{x^2+y^2+z^2=1} (4 - 3x^2 + 2y^2 - z^2) dA = \frac{40\pi}{3}.$$

The discrete delta function was evaluation using $\phi = x^2 + y^2 + z^2 - 1$. Our computations were the average of 20 trials each of which was shifted by a random amount in each coordinate direction. The results are shown in Table 5 and are consistent with the expected order of accuracy.

Remark. In all of our calculations we take $\varepsilon = 10^{-10}$.

9 Summary

In this work, we have developed a discrete delta function that is concentrated on lines in two dimensions and surfaces in three dimensions. A level set representation of the interface is used, consequently this delta function concentrated near its zero level set. In fact, the delta function in concentrated within one grid cell on either side of the interface. We have provided both a first and second order accurate formulation of this delta function in two and three dimensions. We have used this discrete delta function to compute various line and surface integrals. Our computed examples show that the method realizes the expected order of accuracy.

The formulation of this discrete delta function is based on a method developed by Mayo[7, 8] for the solution of elliptic equations with discontinuities. In this approach the Laplacian is discretized in a manner that properly accounts for the jump conditions that need to be satisfied near the interface. In this way, the jump conditions now appear as source terms in a nonhomogeneous elliptic equation. The resulting source terms are, in fact, the discrete delta function.

Acknowledgments

This work was supported in part by the National Science Foundation through Grant No. DMS-0207402.

References

[1] R.P. Beyer and R.J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, *SIAM J. Numer. Anal.* **29**, 332-364 (1992).

[2] D. Calhoun and P. Smereka, The numerical approximation of a delta function, preprint (2004).(available at `http://www.math.lsa.umich.edu/~psmereka/notes.html`)

[3] B. Engquist, A.K. Tornberg, and R. Tsai, Discretization of Dirac delta functions in level set methods, UCLA CAM Report (2004).

[4] T.Y. Hou, Z.L. Li, S. Osher, and H.K. Zhao, A hybrid method for moving interface problems with application to the Hele-Shaw flow, *J. of Comput. Phys*, **134**, 236-252 (1997).

[5] R.J. LeVeque and Z.L. Li, Immersed interface methods for Stokes flow with elastic boundaries, *SIAM J. Sci. Comput.* **18** 709-735 (1997).

[6] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comput. Phys.* **160** 151-178 (2000).

[7] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, *SIAM J. Sci. Comput.* **21** 285-299 (1984).

[8] A. Mayo, The rapid evaluation of volume integrals of potential theory on general regions, *J. Comput. Phys.* **100** 236-245 (1992).

[9] S. Osher and R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York (2000).

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

[11] M. Sussman, P. Smereka, and S. Osher A level set method for computing solutions to incompressible two phase flow, *J. Comput. Phys.* **119**, 146-159 (1994).

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

[13] S.O. Unverdi and G. Trygvasson, A front-tracking method for the computation of multiphase flow, *J. Comput. Phys.* **100** 25-37 (1992)