

Geometric Integration Over Irregular Domains with Application to Level Set Methods

Chohong Min *

Frédéric Gibou †

May 26, 2007

Abstract

In this paper, we present a geometric approach for calculating integrals over irregular domains described by a level-set function. This procedure produces results that are second-order accurate and robust to the perturbation of the interface location on the grid. **Comparing with the well-known delta formulation approaches of [16], we discussed the pros and cons of our geometric formulation and the delta formulations.** Moreover, since we use a cell-wise approach, it is easily extended to quadtree and octree grids. We demonstrate the second-order accuracy and the robustness of the method in two and three spatial dimensions.

1 Introduction

Integration over interfaces and irregular domains defined by a level set function ϕ is traditionally computed using approximated one-dimensional dirac delta functions or approximated Heaviside functions [11, 10]: Consider a domain $\Omega \subset \mathbb{R}^n$ and a lower dimensional interface Γ separating two disjoint subdomains Ω^- and Ω^+ . A level set function ϕ can be used to represent Γ as the set of points x such that $\phi(x) = 0$, Ω^- by $\phi(x) \leq 0$ and Ω^+ by $\phi(x) > 0$. The integral of a function f on Γ and the integral of f over the subdomain Ω^- can be calculated as:

$$\int_{\Gamma} f d\Gamma = \int_{\mathbb{R}^n} f(x) \cdot \delta(\phi(x)) \cdot \|\nabla\phi\| dx, \quad (1)$$

$$\int_{\Omega^-} f d\Omega = \int_{\mathbb{R}^n} f(x) \cdot [1 - H(\phi(x))] dx, \quad (2)$$

where δ and H are approximations of the delta and Heaviside functions, respectively.

However, rather recently, Enquist *et al.* pointed out that in some cases, the above approach using the delta formulation may lead to non convergent approximations [2]. The authors then presented a discretization of the Dirac delta function that removes the problem of convergence. Later, Smereka proposed efficient first- and second-order accurate discretizations [16]. The first-order accurate method is straightforward to implement and produces in some cases results that are more than first-order accurate. The second-order accurate method is less straightforward to

*Mathematics Department, KyungHee University, Seoul, Korea

†Mechanical Engineering Department & Computer Science Department, University of California, Santa Barbara, CA 93106.

implement. However, the first order accurate method may lead to grid dependencies and the rate of convergence may be sensitive to small perturbations of the interface location, which we empirically show in the example section 4.

In this paper, we present an alternative approach for calculating integrals such as (1) and (2). This geometric approach produces results that are second-order accurate and independent of the interface location on the grid. We discretize the interface into a disjoint union of simplices [1, 7] and then use a numerical integration quadrature rule on simplices [4]. This direct discretization of the interface using tools from computational geometry produces theoretically sound and numerically efficient algorithms.

2 Numerical Integration

In this section, we present a numerical integration over the interface $\Gamma = \{x \in \mathbb{R}^n | \phi(x) = 0\}$ and the irregular domain $\Omega^- = \{x \in \mathbb{R}^n | \phi(x) \leq 0\}$ of a continuous function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$. Although this algorithm can be easily extended to higher dimensions, we focus on the two and three dimensional cases.

2.1 Triangulation of a Grid Cell

The approach we choose is to decompose grid cells crossed by the interface into a union of simplices, for which integration is straightforward. A simplex, or n -simplex is the convex hull of a set of $(n+1)$ affinely independent points, i.e. a triangle in two spatial dimensions and a tetrahedron in three spatial dimensions. The affinely independent points are called the vertices of the simplex, and the dimension of the simplex S is n , and denoted by $dim(S)$.

Simplices provides a straightforward way to compute lengths, areas and volumes. Consequently, there have been thorough studies to decompose geometric shapes into simplices for many applications, for example meshing in finite elements [3]. In two spatial dimensions a grid cell can be decomposed into two triangles, as shown in figure 1. In three spatial dimensions a grid cell can be decomposed into five tetrahedra (called the middle cut triangulation [13]) or into six tetrahedra (called the Kuhn triangulation [6]). The Kuhn triangulation can be more easily extended to higher dimensions [7] and is more convenient to match triangulations between adjacent cells. On the other hand, the angles of the tetrahedra created by the middle cut triangulation are less acute than those of the Kuhn triangulation and the smaller number of tetrahedra leads to faster computations. In this work, we choose the middle cut triangulation since we limit ourselves to two and three spatial dimensional cases and because integrations can be computed cell-wise, with no interactions between adjacent cells.

Cartesian cells can be mapped onto the unit cells, so with no loss of generality, we present the middle cut triangulation on the unit cells in two and three spatial dimensions as illustrated in figure 1:

$$\begin{aligned} [0, 1]^2 &= conv(P_{00}, P_{10}, P_{11}) \\ &\cup conv(P_{00}, P_{01}, P_{11}) \end{aligned}$$

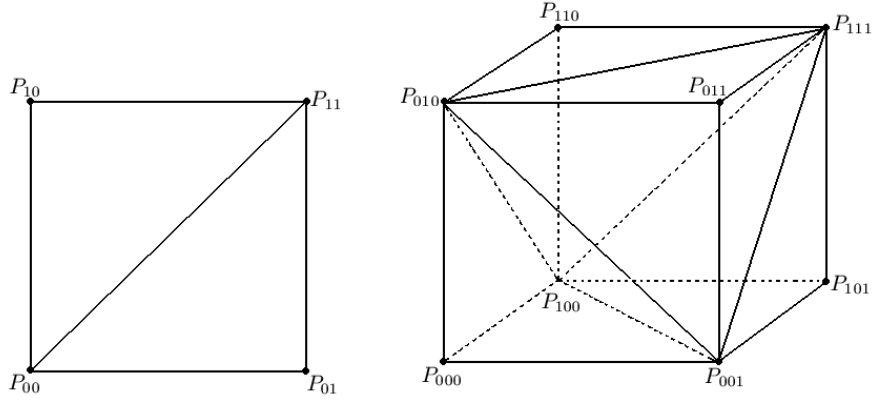


Figure 1: Triangulation of a two (left) and three (right) dimensional cell.

and

$$\begin{aligned}
[0, 1]^3 &= \text{conv}(P_{000}, P_{100}, P_{010}, P_{001}) \\
&\cup \text{conv}(P_{110}, P_{100}, P_{010}, P_{111}) \\
&\cup \text{conv}(P_{101}, P_{100}, P_{111}, P_{001}) \\
&\cup \text{conv}(P_{011}, P_{111}, P_{010}, P_{001}) \\
&\cup \text{conv}(P_{111}, P_{100}, P_{010}, P_{001}),
\end{aligned}$$

where $\text{conv}(P_1, \dots, P_n)$ is the convex hull span by the points P_1, \dots, P_n and defines a simplex. We denote by $T(C)$ the triangulation of a cell C , i.e. C is a disjoint union of simplices in $T(C)$:

$$C = \cup_{S \in T(C)} S$$

Note that the triangulations do not create new vertices, thus a discrete function sampled on a uniform grid is well defined on the vertices of each simplex. Then the integrations are calculated simplex-wise as:

$$\begin{aligned}
\int_{\Gamma} f d\Gamma &= \sum_{C: \text{grid cell}} \int_{C \cap \Gamma} f d\Gamma = \sum_{C: \text{grid cell}} \sum_{S \in T(C)} \int_{S \cap \Gamma} f d\Gamma, \\
\int_{\Omega^-} f d\Omega &= \sum_{C: \text{grid cell}} \int_{C \cap \Omega^-} f d\Omega = \sum_{C: \text{grid cell}} \sum_{S \in T(C)} \int_{S \cap \Omega^-} f d\Omega,
\end{aligned}$$

The integrations are then reduced to $\int_{S \cap \Gamma} f d\Gamma$ and $\int_{S \cap \Omega^-} f d\Omega$ for each simplex S , which necessitate a discretization of $S \cap \Gamma$ and $S \cap \Omega^-$.

2.2 Discretization of $S \cap \Gamma$ and $S \cap \Omega^-$

In general, $S \cap \Gamma$ and $S \cap \Omega^-$ are continuous manifolds possibly with some singularities (sharp corners or sharp edges). For simpler calculations, we approximate the sets with the linear interpolation

of ϕ , using the ϕ values on the vertices of the simplex S , as in [7]. Then, $S \cap \Gamma$ and $S \cap \Omega^-$ are defined by polytopes, i.e. convex hulls of finite points. This allows for efficient representation, since only a finite number of points need to be stored. Specifically, we first approximate the location of the set Γ on each edge of S : Let $\{P_0, \dots, P_n\}$ be the vertices of S , if Γ crosses the edge $P_i P_j$, i.e. $\phi(P_i)\phi(P_j) < 0$, we define the intersection point P_{ij} between Γ and the edge as:

$$P_{ij} = P_j \frac{\phi(P_i)}{\phi(P_i) - \phi(P_j)} - P_i \frac{\phi(P_j)}{\phi(P_i) - \phi(P_j)}.$$

Then $S \cap \Gamma$ and $S \cap \Omega^-$ are polytopes whose vertices are given by:

$$S \cap \Gamma \approx \text{conv}(\{P_{ij} | \phi(P_i)\phi(P_j) < 0\}),$$

and

$$S \cap \Omega^- \approx \text{conv}(\{P_{ij} | \phi(P_i)\phi(P_j) < 0\} \cup \{P_i | \phi(P_i) < 0\}),$$

as illustrated in figures 2, 3 and 4.

Integration over such discretizations of $S \cap \Gamma$ or $S \cap \Omega^-$ can still be complicated since the area or volume computations of polytopes are in general not straightforward. For this reason, if the sets $S \cap \{\phi = 0\}$ or $S \cap \{\phi \leq 0\}$ are not simplices, we further divide them into simplices as described next: Using the values of ϕ at the vertices of S , we linearly interpolate ϕ inside S , thus Γ and Ω^- are geometrically hyperplane and halfspace, respectively. First note that two geometric configurations given by the intersection between a simplex S and a hyperplane Γ are equivalent if they have the same number of vertices below and above the hyperplane Γ , as noted in [17]. Likewise, two geometric configurations given by the intersection between a simplex S and a halfspace Ω^- are equivalent if they have the same number of vertices below and above the hyperplane Γ . Obviously, since the sum of the number of vertices below the hyperplane and the number of vertices above the hyperplane equals the total number of vertices of S , it is enough to know the number of vertices below the hyperplane to classify all the possible geometric configurations. We define this number by

$$\eta(\phi, S) := n(\{P_i | \phi(P_i) < 0\}).$$

In two spatial dimensions, the possible values of $\eta(\phi, S)$ are 0,1,2, and 3. When $\eta(\phi, S) = 0$ or 3, ϕ is positive or negative, therefore $S \cap \Gamma = \emptyset$. The case of $\eta(\phi, S) = 2$ can be treated in the same fashion as the case $\eta(\phi, S) = 1$ by simply negating ϕ (since $S \cap \{\phi = 0\} = S \cap \{-\phi = 0\}$). When $\eta(\phi, S) = 1$, $S \cap \Gamma$ is a line segment as illustrated in figure 2 (left) and table 1 summarizes the different cases. We characterize the decomposition of $S \cap \Omega^-$ in the same way: When $\eta(\phi, S) = 0$, ϕ is positive, therefore $S \cap \Omega^- = \emptyset$. When $\eta(\phi, S) = 3$, ϕ is negative and $S \cap \Omega^- = S$. Figure 3 illustrates the decompositions when $\eta(\phi, S) = 1$ (left) and $\eta(\phi, S) = 2$ (right). Table 3 summarizes the different cases.

In three spatial dimensions, the possible values of $\eta(\phi, S)$ are 0, 1, 2, 3, and 4. When $\eta(\phi, S) = 0$ (resp. $\eta(\phi, S) = 4$), ϕ is positive (resp. negative) and $S \cap \Gamma = \emptyset$. The case $\eta(\phi, S) = 3$ can be treated in the same fashion as the case $\eta(\phi, S) = 1$ by negating ϕ . When $\eta(\phi, S) = 1$ and $\eta(\phi, S) = 2$, the decomposition of $S \cap \Gamma$ is illustrated in figure 2 (center and right) and table 2 summarizes the different cases. Likewise, we characterize the decomposition of $S \cap \Omega^-$: When $\eta(\phi, S) = 0$, ϕ is positive, therefore $S \cap \Omega^- = \emptyset$. When $\eta(\phi, S) = 4$, ϕ is negative, therefore $S \cap \Omega^- = S$. Figure 4 illustrates the decompositions when $\eta(\phi, S) = 1$ (left), $\eta(\phi, S) = 2$ (center) and $\eta(\phi, S) = 3$ (right), and table 4 summarizes the different cases.

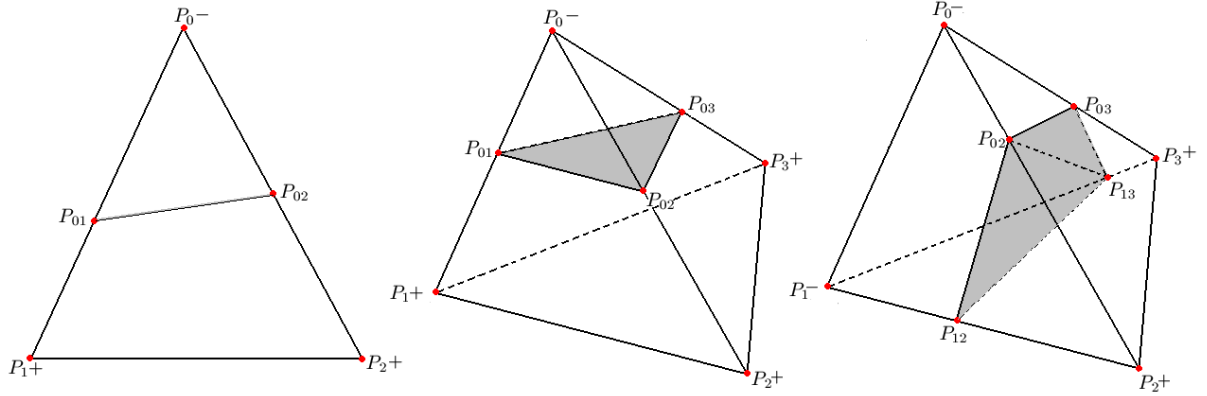


Figure 2: Representation of the set $S \cap \Gamma$ when $\dim(S) = 2$ (left) and $\dim(S) = 3$ (center and right), which is approximated by $\text{conv}(\{P_{ij} | \phi(P_i)\phi(P_j) < 0\})$, where the P_{ij} s are found by linear interpolations of ϕ . The left figure represents the generic case in two spatial dimensions and the right two figures represent the generic cases in three spatial dimensions.

$\phi(P_0)$	$\phi(P_1)$	$\phi(P_2)$	Q_0	Q_1
-	+	+	P_{01}	P_{02}

Table 1: Generic case for representing $S \cap \Gamma$ in two spatial dimensions. Q_0 and Q_1 are the vertices of a line segment as depicted in figure 2 (left).

$\phi(P_0)$	$\phi(P_1)$	$\phi(P_2)$	$\phi(P_3)$	Q_0	Q_1	Q_2
-	+	+	+	P_{01}	P_{02}	P_{03}
-	-	+	+	P_{02}	P_{03}	P_{13}
				P_{02}	P_{12}	P_{13}

Table 2: The two generic cases for representing $S \cap \Gamma$ in three spatial dimensions. Q_0 , Q_1 , and Q_2 are the vertices of a triangle as depicted in figure 2 (center and right).

$\phi(P_0)$	$\phi(P_1)$	$\phi(P_2)$	Q_0	Q_1	Q_2
-	+	+	P_0	P_{01}	P_{02}
-	-	+	P_0	P_1	P_{02}
			P_1	P_{12}	P_{02}

Table 3: The two generic cases for representing $S \cap \Omega^-$ in two spatial dimensions. Q_0 , Q_1 , and Q_2 are the vertices of a triangle as depicted in figure 3.

$\phi(P_0)$	$\phi(P_1)$	$\phi(P_2)$	$\phi(P_3)$	Q_0	Q_1	Q_2	Q_3
-	+	+	+	P_0	P_{01}	P_{02}	P_{03}
-	-	+	+	P_0	P_1	P_{02}	P_{13}
-	-	-	+	P_0	P_{03}	P_2	P_{13}
-	-	-	+	P_0	P_3	P_2	P_{13}

Table 4: The three generic cases for representing $S \cap \Omega^-$ in three spatial dimensions. $Q_0, Q_1, Q_2,$ and Q_3 are the vertices of a tetrahedron, as depicted in figure 4.

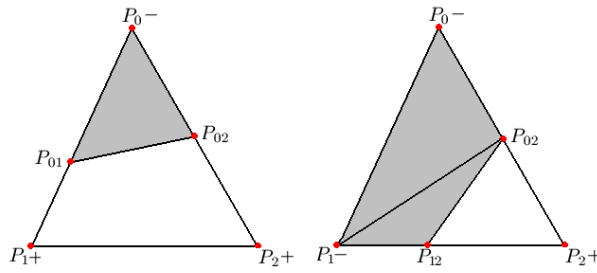


Figure 3: The two generic representations of the set $S \cap \Omega^-$ in two spatial dimensions: One triangle (left) or the union of two triangles (right).

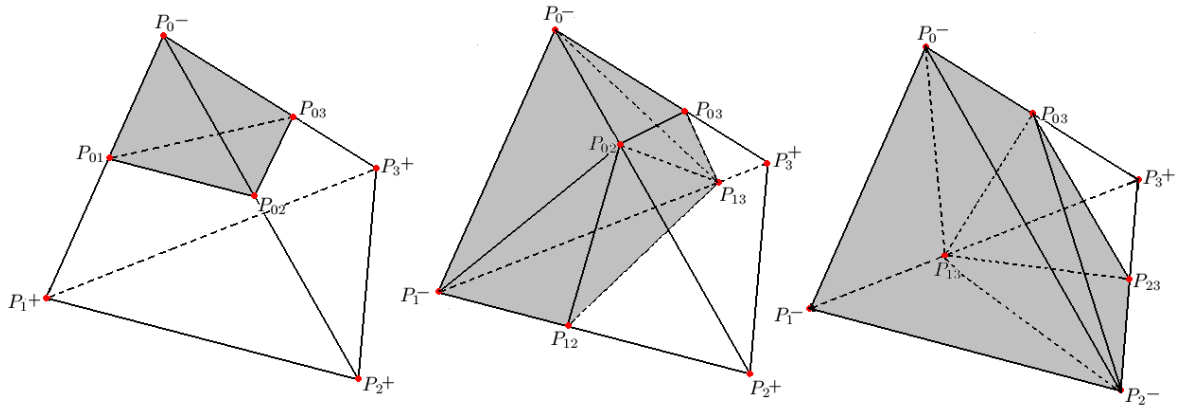


Figure 4: The three generic representations of the set $S \cap \Omega^-$ in three spatial dimensions: One tetrahedron (left) or the union of three tetrahedra (center and right).

We now denote the decompositions of $S \cap \Gamma$ and $S \cap \{\phi \leq 0\}$ by $Sec(\phi, S)$ and $Fru(\phi, S)$, respectively:

$$\begin{aligned} S \cap \Gamma &= \bigcup_{S' \in Sec(\phi, S)} S', \\ S \cap \Omega^- &= \bigcup_{S' \in Fru(\phi, S)} S'. \end{aligned}$$

Using these decompositions, the integrals over Γ and $\{\phi \leq 0\}$ are defined as the sum of integrals over simplices:

$$\begin{aligned} \int_{\Gamma} f d\Gamma &= \sum_{C: \text{grid cell}} \sum_{S \in T(C)} \int_{S \cap \Gamma} f d\Gamma = \sum_{C: \text{grid cell}} \sum_{S \in T(C)} \sum_{S' \in Sec(\phi, S)} \int_{S'} f d\Gamma \\ \int_{\Omega^-} f d\Omega &= \sum_{C: \text{grid cell}} \sum_{S \in T(C)} \int_{S \cap \Omega^-} f d\Omega = \sum_{C: \text{grid cell}} \sum_{S \in T(C)} \sum_{S' \in Fru(\phi, S)} \int_{S'} f d\Omega \end{aligned}$$

2.3 Numerical Integration on a Simplex

The integral of a function f over a simplex S with vertices P_0, \dots, P_n can be approximated by the following second-order midpoint rule: [4].

$$\int_S f dx = vol(S) \cdot \frac{f(P_0) + \dots + f(P_n)}{n+1}.$$

We note that there exist higher order accurate quadrature rules to approximate integrals [4, 5], but in our work the use of linear interpolation for ϕ limits the accuracy to second-order and therefore approximating the integral by the second-order midpoint rule is sufficient. The volume of the simplex is given by

$$vol(S) = \frac{1}{n!} \left| \det \begin{pmatrix} (P_1 - P_0) e_1 & \dots & (P_n - P_0) e_1 \\ \vdots & & \vdots \\ (P_1 - P_0) e_n & \dots & (P_n - P_0) e_n \end{pmatrix} \right|,$$

where e_i represents the i^{th} canonical unit basis vector.

remarks:

1. In the case where $\phi = 0$ on an edge, the summation over the adjacent grid cell will duplicate the integral on the edge. Therefore, we exclude this case by simply perturbing ϕ . In the example section we use $\epsilon = 10^{-20}$, and let

$$\phi_{ijk} = \begin{cases} \epsilon & \text{if } |\phi_{ijk}| < \epsilon \text{ and } \phi_{ijk} > 0 \\ -\epsilon & \text{if } |\phi_{ijk}| < \epsilon \text{ and } \phi_{ijk} \leq 0 \end{cases}.$$

2. In the case where all the vertices of a cell C belongs to Ω^- in the calculation of $\int_{C \cap \Omega^-} f(x) d\Omega$, the triangulation procedure is not necessary. By the standard second order mid-point rule, the integration is approximated as the product of the volume of the cell and the average of f at the vertices of the cell.

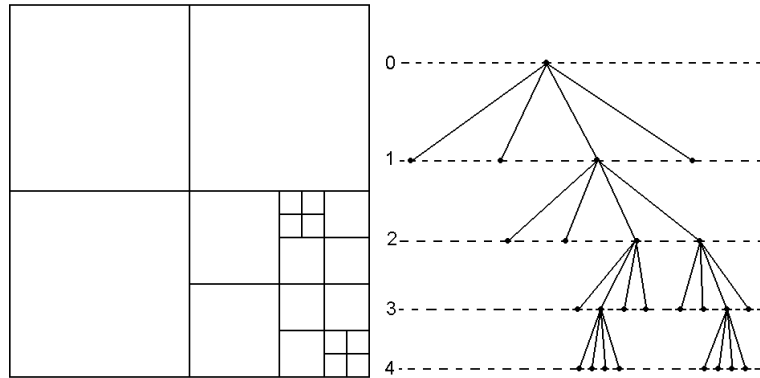


Figure 5: Discretization of a two dimensional domain (left) and its quadtree representation (right). The entire domain corresponds to the root of the tree (level 0). Then each cell can be recursively subdivided further into four children. In this example, the tree is ungraded since the difference of level between cells exceeds one.

3 Extension to Quadtree and Octree Data Structures

This approach can be trivially extended to unstructured Cartesian meshes since the algorithm is based on integrating over a grid cell, with no dependence from one cell to others. For the sake of efficiency, it is desirable to utilize data structures that reduce the total amount of cells to be used, since only those adjacent to the set Γ contribute to the integration of $\int_{\Gamma} f d\Gamma$. In this section, we describe a simple implementation using Quadtree and Octree data structures.

3.1 Quadtree/Octree Data Structures

Quadtree (resp. octree) data structures can be used to represent the spatial discretization of a physical domain in two (resp. three) spatial dimensions as depicted in figure 5: Initially the root of the tree is associated with the entire domain, then we recursively split each cell into four children until the desired level of detail is achieved. This is done similarly in three spatial dimensions, except that cells are split into eight children. We refer the reader to the books of Samet [15, 14] for more details on quadtree/octree data structures.

3.2 Constructing Quadtrees/Octrees from Implicit Representation of the Domains

Since the accuracy of the method depends on the size of the cells adjacent to the interface, we impose that the finest cells lie on the interface. This can be achieved by using the signed distance function to the interface along with the Whitney decomposition, as first proposed by Strain in [18]. Simply stated, one *"splits any cell whose edge length exceeds its distance to the interface"*. For a general function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ with Lipschitz constant $Lip(\phi)$, the Whitney decomposition was

extended in Min [9]: Starting from a root cell, split any cell C if

$$\max_{v \in \text{vertices}(C)} |\phi(v)| \leq \frac{1}{2} \cdot \text{Lip}(\phi) \cdot \text{diag-size}(C),$$

where $\text{diag-size}(C)$ refers to the length of the diagonal of the current cell C and v refers to a vertex (node) of the current cell. We note that a signed distance function can be obtained from a given level set function ϕ by using the so-called reinitialization equation, introduced for uniform grids by Sussman *et al.* [19] (see also Russo and Smereka [12]) and extended to unstructure grids in Min and Gibou [8].

4 Examples

In this section, we demonstrate that our approach produces robust, second-order accurate results in two and three spatial dimensions. All the examples were computed on a PC with 2.2GHz CPU and 2GB memory.

4.1 Robustness to Grid Perturbations in 2D

Consider an ellipse represented as the zero level set of $\phi(x, y) = \frac{x^2}{1.5^2} + \frac{y^2}{.75^2} - 1$. Its exact arc-length is given as $\simeq 7.266336165$ [16]. Table 5 shows that the proposed scheme is second-order accurate and produces results that are stable in the case where the interface is randomly shifted on the grid. The robustness is demonstrated by the fact that the ratios of the maximum error over the minimum error in table 5 is close to one. We compare our results with those obtained with the first-order delta formulation of [16]. In this case, the ratios of the maximum error over the minimum error can be large for some configurations (see table 5), indicating a dependence on the interface location on the grid. Table 6 shows the convergence rates for the three methods (geometric integral, and first and second order delta function approaches) on the uniform grid with one trial. **HERE WE MENTION THE JUMPING ACCURACY OF DELTA FUNCTION APPROACHES.**

Table 7 illustrates the second-order accuracy of our method for the computation of the area of an ellipse.

4.2 Robustness to Grid Perturbations in 3D

Consider an ellipsoid represented as the zero level set of $\phi(x, y, z) = \frac{x^2}{1.5^2} + \frac{y^2}{.75^2} + \frac{z^2}{.5^2} - 1$. Its exact surface area is given as $\simeq 9.901821$ [16]. Table 8 demonstrates that the proposed scheme is second-order accurate and compares our results to those obtained in [16]. In both cases the interface location is shifted randomly. Our method is robust as it is the case in two spatial dimensions. **The second order delta formulation shows results with huge discrepancy with respect to small perturbations of interface. We note that in the three dimensional case, the results obtained with the first order delta formulation of [16] seem to be much more robust than in the two dimensional case and produces results comparable to ours. In addition we note that our geometric approach depends on the continuity of the level function, while the first and second order delta formulations depends on the first and second order differentiability of the level function, respectively. In the situation when the level function is not guaranteed to be smooth, our approach or the first order delta formulation is recommended much more than the second order delta formulation.**

Geometric Numerical Integration								
Δx	Avg	Order	Stdev	Min	Order	Max	Order	$\frac{\max}{\min}$
.2	5.04E-3		2.15E-4	4.63E-3		5.49E-3		1.19
.1	1.26E-3	2.00	3.23E-5	1.17E-3	1.99	1.30E-3	2.08	1.11
.05	3.14E-4	2.00	6.61E-6	3.03E-4	1.95	3.26E-4	2.00	1.08
.025	7.84E-5	2.00	1.25E-6	7.50E-5	2.02	7.99E-5	2.03	1.07
.0125	1.96E-5	2.00	2.15E-7	1.90E-5	1.98	1.99E-5	2.01	1.04
.00625	4.90E-6	2.00	3.18E-8	4.83E-6	1.98	4.94E-6	2.01	1.02

First Order Delta Function Approach [16]								
Δx	Avg	Order	Stdev	Min	Order	Max	Order	$\frac{\max}{\min}$
.2	8.96E-3		7.39E-3	1.28E-4		2.67E-2		208
.1	2.70E-3	1.73	2.96E-3	9.13E-5	1.31	1.07E-2	0.49	118
.05	9.55E-4	1.50	1.12E-3	4.19E-7	1.28	4.43E-3	7.77	10600
.025	3.21E-4	1.57	3.58E-4	7.32E-6	1.54	1.52E-3	-4.12	208
.0125	1.13E-4	1.51	1.22E-4	9.15E-6	1.53	5.28E-4	-0.32	57.7
.00625	3.94E-5	1.52	4.17E-5	2.01E-6	1.52	1.84E-4	2.19	91.7

Second Order Delta Function Approach [16]								
Δx	Avg	Order	Stdev	Min	Order	Max	Order	$\frac{\max}{\min}$
.2	3.23E-3		2.74E-3	6.07E-4		1.30E-2		21.5
.1	5.74E-4	2.49	5.25E-4	2.93E-6	7.69	3.02E-3	2.10	1030
.05	1.13E-4	2.34	4.08E-5	2.55E-5	3.12	2.04E-4	3.88	8.01
.025	3.08E-5	1.87	8.31E-6	1.56E-5	0.70	4.72E-5	2.11	3.00
.0125	7.61E-6	2.01	1.51E-6	1.37E-6	3.50	1.21E-5	1.96	8.82
.00625	1.89E-6	2.01	1.82E-7	1.61E-6	0.23	2.21E-6	2.45	1.37

Table 5: Statistics of 50 trials for computing the arc-length of an ellipse for example 4.1. Top: Geometric approach. Middle and Bottom: The first and second order accurate delta formulation approaches of [16].

Δx	Geometric Integral	Order	First Order Delta	Order	Second Order Delta	Order
0.2	5.49E-3		3.94E-3		8.22E-4	
0.1	1.24E-3	2.13	9.03E-3	-1.19	1.15E-2	-3.80
0.05	3.03E-4	2.03	4.42E-3	1.02	4.99E-3	1.20
0.025	7.49E-5	2.01	1.52E-3	1.54	1.41E-3	1.82
0.0125	1.90E-5	1.97	5.28E-4	1.52	3.66E-4	1.94

Table 6: Convergence rate for computing the arc-length of an ellipse for example 4.1. Top: Geometric approach. Middle and Bottom: The first and second order accurate delta formulation approaches of [16].

Δx	Geometric Integral	Order
.2	1.59E-2	
.1	3.76E-3	2.08
.05	9.46E-4	1.99
.025	2.25E-4	2.07
.0125	5.78E-5	1.97
.00625	1.46E-5	1.99

Table 7: Relative errors on computing the area of an ellipse for example 4.1.

Table 9 demonstrate second-order accuracy of our approach for computing the volume of the ellipsoid.

4.3 Surface Integral on a Torus

Consider a torus T described by the zero level set of $\phi(x, y, z) = (\sqrt{x^2 + y^2} - 2)^2 + z^2 - 1$. We compute the surface integral $\int_T x^2 dS = 22\pi^2$ on the torus. Figure 6 (left) illustrate the geometry of the surface and table 10 demonstrates the second-order accuracy of our method.

4.4 Surface Integral on a Surface of Genus Two

Consider a surface represented by the zero level set of $\phi(x, y, z) = ((1.2 - x^2)x^2 - y^2)^2 + z^2 - .1$. According to the Gauss-Bonnet theorem, the surface integral of the Gaussian curvature κ_G is -4π , since the surface has genus two, i.e. two handles as shown in figure 6 (right). The Gaussian curvature can be calculated from the level function as:

$$\kappa_G = \frac{\phi_x^2(\phi_{yy}\phi_{zz} - \phi_{yz}^2) + 2\phi_y\phi_z(\phi_{xy}\phi_{zx} - \phi_{yz}\phi_{xx}) + \phi_y^2(\phi_{xx}\phi_{zz} - \phi_{xz}^2) + 2\phi_x\phi_z(\phi_{xy}\phi_{yz} - \phi_{xz}\phi_{yy}) + \phi_z^2(\phi_{xx}\phi_{yy} - \phi_{xy}^2) + 2\phi_x\phi_y(\phi_{yz}\phi_{zx} - \phi_{xy}\phi_{zz})}{(\phi_x^2 + \phi_y^2 + \phi_z^2)^2}.$$

In this example, all the derivatives are approximated with the standard central finite differences. Table 11 demonstrates the second-order accuracy of our method.

4.5 Surface Integral using Octrees

Consider a surface, called orthocircles, represented as the zero contour of the level-set function:

$$\phi(x, y, z) = ((x^2 + y^2 - 1)^2 + z^2) ((y^2 + z^2 - 1)^2 + x^2) ((z^2 + x^2 - 1)^2 + y^2) - .075^2 (1 + 3(x^2 + y^2 + z^2)).$$

This surface is topologically equivalent to a surface with seven holes. We compute the surface integral of the Gaussian curvature as in the previous example but we use the octree data structure

Geometric Numerical Integration								
Δx	Avg	Order	Stdev	Min	Order	Max	Order	$\frac{\max}{\min}$
.2	3.17E-2		2.90E-4	3.12E-2		3.22E-2		1.03
.1	7.91E-3	1.98	1.02E-5	7.89E-3	1.98	7.94E-3	2.02	1.00
.05	1.98E-3	2.00	6.81E-7	1.98E-3	2.00	1.98E-3	2.00	1.00
.025	4.94E-4	2.00	1.13E-7	4.94E-4	2.00	4.95E-4	2.00	1.00

First Order Delta Function Approach [16]								
Δx	Avg	Order	Stdev	Min	Order	Max	Order	$\frac{\max}{\min}$
.2	3.03E-2		7.12E-3	1.75E-2		4.73E-2		1.49
.1	7.77E-3	1.96	2.26E-3	3.95E-3	2.14	1.32E-3	1.84	1.51
.05	2.12E-3	1.87	7.36E-4	6.39E-4	2.62	4.48E-3	1.56	2.16
.025	5.20E-4	2.03	1.36E-4	3.41E-4	0.91	8.51E-4	2.39	1.54

Second Order Delta Function Approach [16]								
Δx	Avg	Order	Stdev	Min	Order	Max	Order	$\frac{\max}{\min}$
.2	6.86E-2		7.60E-2	1.01E-2		4.46E-1		44.4
.1	1.33E-2	2.37	8.38E-3	9.47E-4	3.41	2.91E-2	3.93	3.08
.05	2.60E-3	2.35	2.11E-3	2.49E-4	1.92	1.06E-2	1.45	42.4
.025	8.00E-4	1.70	1.01E-3	4.40E-6	5.82	6.88E-3	0.62	1570

Table 8: Statistics of 50 trials for computing the surface area of an ellipsoid for example 4.2. Top: Geometric approach. Middle and Bottom: The first and second order accurate delta formulation approaches of [16].

Δx	Geometric Integral	Order
.1	1.36E-2	
.05	3.40E-3	2.00
.025	8.50E-4	2.00
.0125	2.12E-4	2.00
.00625	5.31E-5	2.00

Table 9: Relative errors on computing the volume of an ellipsoid for example 4.2.

Δx	Geometric Integral	Order
.2	7.08E-3	
.1	1.78E-3	1.99
.05	4.49E-4	1.99
.025	1.12E-4	1.99
.0125	2.85E-5	2.00

Table 10: Relative errors on computing the surface integral for example 4.3.

Δx	Geometric Integral	Order
.1	1.96E-1	
.05	6.38E-2	1.62
.025	1.65E-2	1.95
.0125	4.11E-3	2.01
.00625	1.03E-3	1.99

Table 11: Relative errors on computing the surface integral for example 4.4.

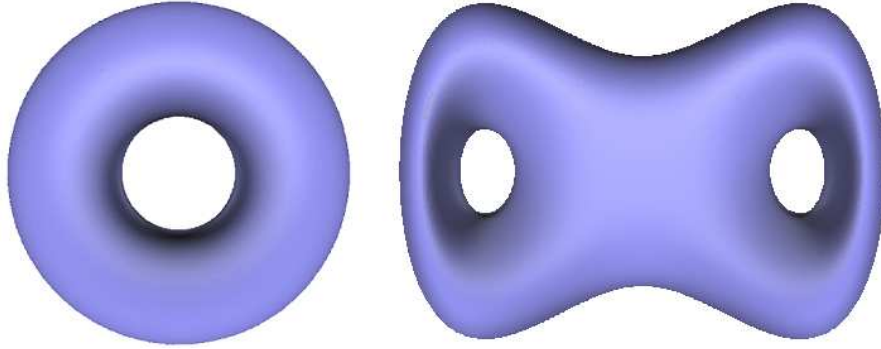


Figure 6: Torus used in example 4.3 (left) and genus two surface (right) used in example 4.4.

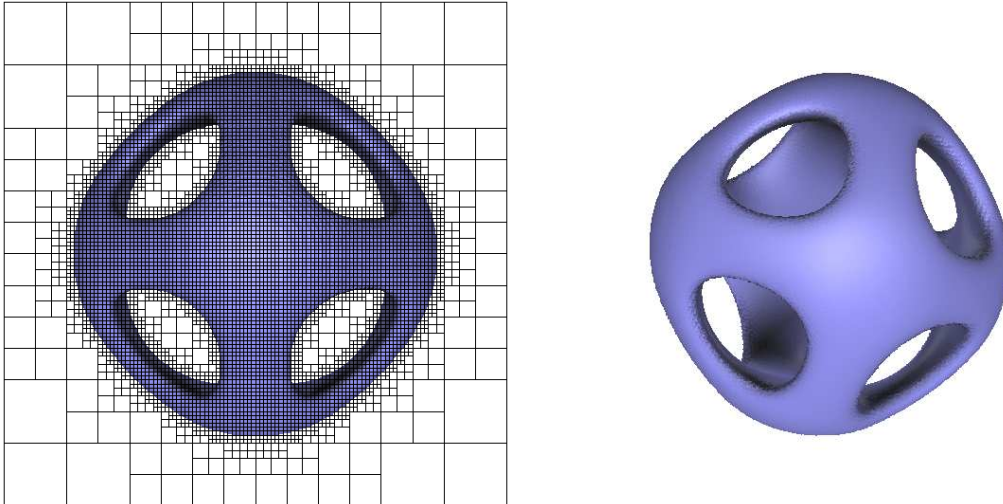


Figure 7: Three dimensional object (orthocircles) with genus 7 used in example 4.5. Left: Octree representation of the interface demonstrating that the smallest cells are placed around the interface. Right: A different view of the same surface.

Finest Resolution	Δx	number of nodes	order	relative error	order	genus	order
64^3	.05	45113		2.22-1		8.337614	
128^3	.025	206241	2.19	4.95-2	2.17	7.297007	2.19
256^3	.0125	834557	2.02	1.23-2	2.00	7.074171	2.01
512^3	.00625	3372753	2.01	3.08-3	2.00	7.018503	2.01
1024^3	.003125	13546163	2.00	7.69-4	2.00	7.004615	2.01

Table 12: Genus calculation for example 4.5.

described in section 3. Figure 7 depicts the geometrical shape and the octree representation. From the surface integral of the Gaussian curvature, we derive the genus number g of the surface by the Gauss-Bonne theorem : $\int_{\Gamma} \kappa_G dx = 4\pi(1 - g)$. Table 12 demonstrate the second-order accuracy of our method. Note that the number of nodes grows quadratically, since the octree data structure put most resource near the interface. The number of nodes for a 1024^3 uniform grid is about 80 times more than for the octree data structure used in this example, which produces the same effective resolution.

5 Conclusion

In this paper, we have presented a geometric approach for calculating integrals over an irregular domain described by a level-set function. This procedure produces results that are second-order accurate and robust to the perturbation of the interface location on the grid. We have demonstrated the robustness and the second-order accuracy of the method in two and three spatial dimensions.

We have compared our geometric formulation with the well-known delta function formulations of [16]. Their respective pros and cons were shown. Our geometric formulation shows second order accuracy in all the examples. The delta formulations show first and second order accuracies only in averages, and may show potential unpredictable results in each trial. The delta formulation can be directly used in formulating singular forces, for example surface tension between two materials and electro-magnetic force acting on a thin film. The singular forcing term is directly discretized by the delta function. In our geometric formulation, singular forces would be treated as its average on each cell. At each grid node, we consider a grid cell containing the grid node, and then integrate the singular force on the grid cell. The integral value give the discretization of the singular force. Simply speaking, delta formulation can take finite difference approach(direct approach), and geometric formulation takes finite volume approach in treating singular forces(indirect approach). Based on the above pros and cons of the two formulations, users should decide proper choice based on their applications.

6 Acknowledgments

The authors acknowledge stimulating discussions with Peter Smereka, and Guofang Wei for discussion on the genus in the orthocircles' example. The research of F. Gibou was supported in part by a Sloan Research Fellowship in Mathematics. This research of the first author, Chohong Min was supported by the Kyung Hee University Research Fund(KHU-20070608) in 2007.

References

- [1] B.P. Carnerio, C. Silva, and A.E. Kaufman. Tetra-cubes: An algorithm to generate 3d isosurfaces bases upon tetrahedra. *Anais do IX SIBGRAPI*, pages 205–210, 1996.
- [2] B. Engquist, A.K. Tornberg, and R. Tsai. Discretization of dirac delta functions in level set methods. *J. Comput. Phys.*, 207:28–51, 2005.
- [3] J. E. Goodman and J. O’Rourke. The handbook of discrete and computational geometry. *CRC Press LL*, 1997.
- [4] A. Grundmann and M. Moeller. Invariant integration formulas for the n-simplex by combinatorial methods. *SIAM J. Numer. Anal.*, 2:282–290, 1978.
- [5] C.P. Hammer and A.H. Stroud. Numerical integration over simplexes. *Mathematical Tables and Other Aids to Computation*, 10:137–139, 1956.
- [6] H.W. Kuhn. Some combinatorial lemmas in topology. *IBM J. Res. Develop.*, 4:508–524, 1960.
- [7] C. Min. Simplicial isosurfacing in arbitrary dimension and codimension. *J. Comput. Phys.*, 190:295–310, 2003.
- [8] C. Min and F. Gibou. A second order accurate level set method on non-graded adaptive Cartesian grids. *under review, available at UCLA CAM report (06-22)* <http://www.math.ucla.edu/applied/cam/index.html>, 2006.
- [9] C.-H. Min. Local level set method in high dimension and codimension. *J. Comput. Phys.*, 200:368–382, 2004.
- [10] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
- [11] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155:410–438, 1999.
- [12] G. Russo and P. Smereka. A remark on computing distance functions. *J. Comput. Phys.*, 163:51–67, 2000.
- [13] J.F. Sallee. The middle-cut triangulations of the n-cube. *SIAM J. Alg. Disc. Methods*, 5:407–419, 1984.
- [14] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, New York, 1989.
- [15] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS*. Addison-Wesley, New York, 1990.
- [16] P. Smereka. The numerical approximation of a delta function with application to level set methods. *J. Comput. Phys.*, 211:77–90, 2006.
- [17] D.M.Y. Sommerville. *An Introduction to the Geometry of N Dimensions*. Dover Publications, 1958.

- [18] J. Strain. Tree methods for moving interfaces. *J. Comput. Phys.*, 151:616–648, 1999.
- [19] M. Sussman and E. Fatemi. An efficient interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM J. of Scientific Comput.*, 20:1165–1191, 1999.