Journal of Computational Mathematics
Vol.29, No.3, 2011, 305–323.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1010-m3285

HIGH ORDER NUMERICAL METHODS TO TWO DIMENSIONAL HEAVISIDE FUNCTION INTEGRALS*

Xin Wen
LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
Email: wenxin@amss.ac.cn

Abstract

In this paper we design and analyze a class of high order numerical methods to two dimensional Heaviside function integrals. Inspired by our high order numerical methods to two dimensional delta function integrals [19], the methods comprise approximating the mesh cell restrictions of the Heaviside function integral. In each mesh cell the two dimensional Heaviside function integral can be rewritten as a one dimensional ordinary integral with the integrand being a one dimensional Heaviside function integral which is smooth on several subsets of the integral interval. Thus the two dimensional Heaviside function integral is approximated by applying standard one dimensional high order numerical quadratures and high order numerical methods to one dimensional Heaviside function integrals. We establish error estimates for the method which show that the method can achieve any desired accuracy by assigning the corresponding accuracy to the sub-algorithms. Numerical examples are presented showing that the second to fourth-order methods implemented in this paper achieve or exceed the expected accuracy.

Mathematics subject classification: 65D05, 65D30, 65D32.
Key words: Heaviside function integral, High order numerical method, Irregular domain.

1. Introduction

We study in this paper a class of high order numerical methods to the two dimensional Heaviside function integrals

$$\int_{\mathbb{R}^2} f(x,y)H(u(x,y))dxdy, \tag{1.1}$$

where $f(x,y)$ is an integrand function, $u(x,y)$ is a level set function whose zero points define certain curve in the two dimensional space which compose the boundaries of an irregular bounded domain $\Omega = \{(x,y)|u(x,y) > 0\}$. The Heaviside function integral (1.1) is equivalent to

$$\int_{\Omega} f(x,y)dxdy. \tag{1.2}$$

We consider that the functions f, u have sufficient smoothness and their values are only provided at grid points of a regular mesh. The domain Ω is defined implicitly by the level set function u. Studying the computations of Heaviside function integrals in two and three dimensions in the above context is applicable to many problems. One example is computing immiscible multiphase flow [2, 9, 11]. In such applications the unknown quantities such as density and viscosity are generally discontinuous across interfaces separating the immiscible fluids. One convenient strategy is to employ fixed computational mesh and allow the moving interface to cut through mesh cells. In this situation the computations by finite element method requires

* Received December 22, 2009 / Revised version received August 18, 2010 / Accepted September 14, 2010 / Published online February 28, 2011 /

evaluating integrals with discontinuous integrands in the variational formulation, which can be performed by resorting to computations of Heaviside function integrals.

The computation of Heaviside function integrals has much relation with computing delta function integrals. The latter problem corresponds to evaluating integrals restricted on the domain boundary $\partial\Omega$ which is a codimension one manifold. For the study of numerical methods to delta function integrals one can refer to [1, 3–5, 8, 10, 12–14, 16–20]. Most of these methods have also been extended to study the computations of Heaviside function integrals.

In [10] Tornberg studied the computations of two dimensional Heaviside and delta function integrals. The approach is to regularize the discontinuous or singular integrand, and then apply a standard quadrature to the integral with the regularized integrand. This approach allows the error analysis by separately considering the analytical error from regularization and numerical error from quadrature. The error of the approach is determined by the moment and regularity conditions of the regularized delta function and the order of the quadrature method. This approach can be designed to be of arbitrary high order accuracy. However high order method requires utilizing regularized delta functions with very high order moment or regularity conditions which can be complicated and may influence the efficiency of the method.

In [3] Engquist, Tornberg and Tsai studied the regularization of multidimensional Heaviside function based on regularized one dimensional Heaviside functions and a variable support size formula. The method is shown to be second-order accurate which improves on the first-order accuracy of the conventional regularization based on regularized one dimensional Heaviside functions and the constant support size formula. They also presented a regularized Heaviside function based on integrating a product formula of one dimensional discrete delta functions. The product formula method following Peskin [6, 7] has the advantage that it can achieve any desired accuracy by using one dimensional discrete delta functions with corresponding discrete moment conditions (see the proof in [12]). However the high order version of the product formula method has not been implemented in the case of the domain Ω implicitly defined by a level set function.

In [4, 5] Min and Gibou designed a geometric integration method for computing Heaviside and delta function integrals. The approach to Heaviside function integrals is to decompose the domain Ω into simplices on which the numerical quadrature can be applied. This method gives second-order results.

In [15] Towers proposed a type of methods for discretizing multidimensional Heaviside function based on approximating the Heaviside function by finite differencing its first few primitives. This idea has been adopted to study the approximation of delta function integrals in [13, 14, 16]. Two variants of the methods are presented in [15] for computing Heaviside function integrals. The first method gives second-order accuracy. The second method is shown to give third-order accuracy for a specific one dimensional example and behave fourth-order convergent for general multidimensional computations. Error analysis for the second-order version method is given in [15]. We will give a comparison between the numerical results of the high order version method in [15] and our high order methods in this paper, which shows the advantage of our third- and fourth-order methods.

In this paper we design and analyze a class of high order numerical methods to the two dimensional Heaviside function integrals (1.1). We construct these methods by considering the approximation of the restriction of the two dimensional Heaviside function integral in each mesh cell. Such a natural strategy of approximating mesh cell restrictions of integrals has already been adopted in [17] for designing high order methods to delta function integrals of full codimension

type, in [4, 5] for designing geometric integration method for computing Heaviside and delta function integrals and in [19,20] for designing high order methods to two and three dimensional delta function integrals of codimension one type. By applying this strategy we need to check the intersection between mesh cells and zero points of the level set function u representing the domain boundary $\partial\Omega$. In a mesh cell away from $\partial\Omega$ the computation of the Heaviside function integral (1.1) is straightforward. The strategy to approximate the restriction of the Heaviside function integral (1.1) in a mesh cell intersecting with $\partial\Omega$ is inspired by our design of high order methods for computing codimension one delta function integrals in [19, 20]. The method is based on that the two dimensional Heaviside function integral in the mesh cell intersecting with $\partial\Omega$ can be rewritten as a one dimensional ordinary integral with the integrand being a one dimensional Heaviside function integral. We select the transformed one dimensional integral from one of two forms according to the comparison of the two components of gradient of u in the cell which can be checked from the mesh point values of u. With this treatment, the one dimensional Heaviside function integral as the integrand is well-defined and can be computed by standard numerical quadrature. The one dimensional Heaviside function integral being the integrand however can be not sufficiently smooth on the integral interval. This issue is resolved by showing that the integral interval of the one dimensional ordinary integral can be divided into subsets so that the integrand is smooth on each of the subsets. Thus we construct our high order methods by approximating each subset of the integral interval of the one dimensional ordinary integral on which the integrand is smooth and applying standard high order numerical quadratures to the resulting integrals on these subsets and to the evaluation of the integrands which are the one dimensional Heaviside function integrals. The algorithm so designed to approximate the mesh cell restrictions of the two dimensional Heaviside function integral (1.1) comprises the numerical method developed in this paper. We carry out error analysis for the method in this paper and prove that the method can achieve any desired accuracy to the two dimensional Heaviside function integrals (1.1) by imposing corresponding accuracy in the sub-algorithms in the method. The key point of the error estimates is that the approximation of the two dimensional Heaviside function integral (1.1) in any given mesh cell may not be accurate enough, while for such a mesh cell there exists certain neighboring cell so that the approximate two dimensional Heaviside function integral restricted to the union of the mesh cells has sufficient accuracy. We implement second- to fourth-order numerical methods in this paper and the numerical examples show that these methods achieve or exceed the expected accuracy indicated by the error analysis. In this paper we consider the computation of two dimensional Heaviside function integrals (1.1). It is of interest to adopt similar idea to further study high order methods to three dimensional Heaviside function integrals in the future.

This paper is organized as follows. In Section 2 we discuss the main strategy, numerical implementation and algorithm description of the method proposed in this paper. In Section 3 we establish the error estimates for our method which show that the method can achieve any desired accuracy by choosing corresponding accuracy in the sub-algorithms. In Section 4 we present numerical examples in which second- to fourth-order methods are shown to achieve or exceed the expected accuracy. We conclude the paper in Section 5.

2. High Order Numerical Methods

2.1. Main strategy and numerical implementation

Assume $\mathbb{R}^2$ is covered by a uniform mesh $(x_i, y_j), (i, j) \in \mathbb{Z}^2$ with the mesh size h on which

the values of f and u are given. Denote

$$I_{i,j} = \int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} f(x,y)H(u(x,y))dxdy. \tag{2.1}$$

Then the Heaviside function integral (1.1) is given by

$$\int_{\mathbb{R}^2} f(x,y)H(u(x,y))dxdy = \sum_{(i,j)\in\mathbb{Z}^2} I_{i,j}. \tag{2.2}$$

Therefore the Heaviside function integral can be computed by approximating each $I_{i,j}$. Let $\widehat{I}_{i,j}$ be the approximation to $I_{i,j}$. Then our goal is to provide algorithm to yield $\widehat{I}_{i,j}$.

If the vertex values of the level set function are the same sign for a cell $C_{i,j} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, indicating the cell is away from $\partial\Omega$, then the computation of $\widehat{I}_{i,j}$ is straightforward. If the vertex values are all positive, mainly corresponding to the cell being contained in Ω, then naturally $\widehat{I}_{i,j}$ is computed as $\int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} f(x,y)dxdy$. If the vertex values are all negative, mainly corresponding to the cell being outside Ω, then naturally we set $\widehat{I}_{i,j} = 0$. Moreover we also treat a special case that two vertex values of u are zero and the other two vertex values of u are negative. This case implies the cell is totally or nearly outside Ω and naturally we also set $\widehat{I}_{i,j} = 0$. This treatment is related to achieving high order accuracy of our method and the reason will be further explained in this subsection after the presentation of Strategy I and Algorithm I. Therefore we need to consider the computation of $\widehat{I}_{i,j}$ in other nontrivial cases not belonging to the above situations, mainly corresponding to that the cell essentially intersects with $\partial\Omega$. Inspired by the approach in [19], our strategy for the computation in the nontrivial cases is based on the observation that the two dimensional Heaviside function integral (2.1) can be rewritten as the following two forms

$$I_{i,j} = \int_{x_i}^{x_{i+1}} \left(\int_{y_j}^{y_{j+1}} f(x,y)H(u(x,y))dy \right) dx, \tag{2.3a}$$

$$I_{i,j} = \int_{y_j}^{y_{j+1}} \left(\int_{x_i}^{x_{i+1}} f(x,y)H(u(x,y))dx \right) dy. \tag{2.3b}$$

Namely the two dimensional Heaviside function integral $I_{i,j}$ can be regarded as a one dimensional integral with the integrand being a one dimensional Heaviside function integral. Therefore in principle high order approximation to $I_{i,j}$ can be achieved by applying standard one dimensional high order numerical quadratures and high order methods to one dimensional Heaviside function integrals.

One issue is that we need to choose one of the two forms (2.3a) and (2.3b) as the formula for designing numerical methods in each mesh cell. It is convenient to choose the form so that the level set function u is monotone with respect to the integral variable of the one dimensional Heaviside function integral which in such case is well-defined and easy to compute. In fact at $\partial\Omega$ at least one of u_x and u_y is $\mathcal{O}(1)$ away from zero since $\sqrt{(u_x)^2 + (u_y)^2}$ is the normal derivative of $u(x,y)$ at $\partial\Omega$ which should have $\mathcal{O}(1)$ positive lower bound for a well-defined level set function u (see also the discussion in [19]). Therefore we can compare u_x and u_y in the cell $C_{i,j}$ to check which one of them is away from zero. In practical computation, we compare the difference approximations to $|u_x|$ and $|u_y|$ at center point of the cell. Namely we compare the

quantities

$$\widetilde{u}^x_{i,j} = |u_{i+1,j+1} + u_{i+1,j} - u_{i,j+1} - u_{i,j}|, \tag{2.4a}$$

$$\widetilde{u}^y_{i,j} = |u_{i+1,j+1} + u_{i,j+1} - u_{i+1,j} - u_{i,j}|. \tag{2.4b}$$

If $\widetilde{u}^x_{i,j} \geq \widetilde{u}^y_{i,j}$, then u_x is away from zero near the cell $C_{i,j}$, and we choose (2.3b) as the formula for performing computations. Otherwise u_y is away from zero near the cell, and (2.3a) is chosen as the formula for computations.

For the convenience of description, we discuss the case that the form (2.3a) is chosen for computations. Denote

$$F(x) = \int_{y_j}^{y_{j+1}} f(x,y)H(u(x,y))dy, \quad x_i \leq x \leq x_{i+1}. \tag{2.5}$$

Then (2.3a) is written to be

$$I_{i,j} = \int_{x_i}^{x_{i+1}} F(x)dx. \tag{2.6}$$

In order to get high order numerical results by applying numerical quadratures to the one dimensional integral (2.6), the function $F(x)$ needs to have sufficient smoothness. However $F(x)$ generally can have discontinuous first derivative on $[x_i, x_{i+1}]$. For example, consider $f(x,y) \equiv 1$ and the situations shown in Fig. 2.1, which have been similarly considered in [19] for illustrating situations in computing delta function integrals. In the left part of the figure, the derivative of $F(x)$ is zero on $[x_i, x^*)$ while nonzero on $(x^*, x_{i+1}]$, thus is discontinuous at $x = x^*$. The right part of the figure shows a situation that the derivative of $F(x)$ has two discontinuity points at $x = x', x''$ due to $\partial\Omega$ having multiple intersection points between the horizontal side of $C_{i,j}$.

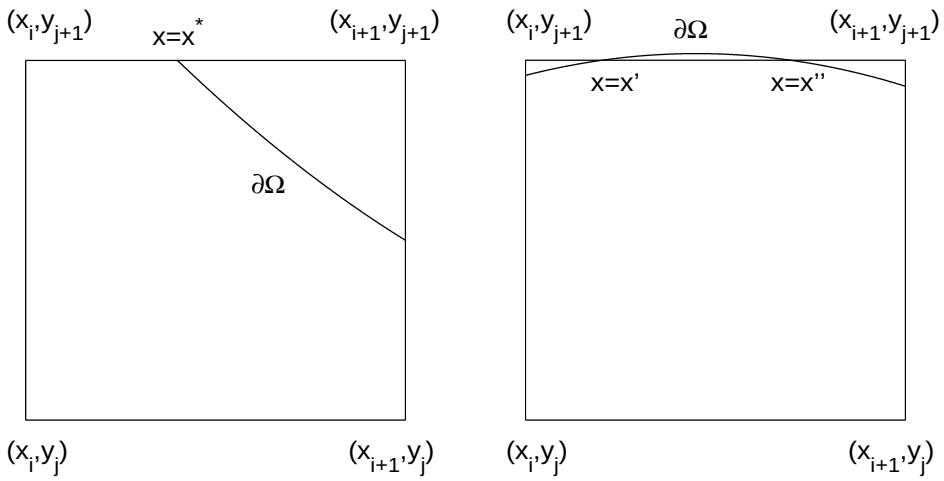

Figure 2.1 Situations in which $F(x)$ has discontinuous derivative on $[x_i, x_{i+1}]$.

Since $F(x)$ can have discontinuous derivative on $[x_i, x_{i+1}]$ with one or more discontinuity points, it is improper to directly apply numerical quadrature to (2.6) in order to obtain high order numerical results. Nevertheless, we can show that the interval $[x_i, x_{i+1}]$ can be split

into subsets so that $F(x)$ is smooth on each subset. Since u_y is away from zero near the cell, introduce the definitions:

$$Y_{i,j}(x) \text{ satisfying that } (x, Y_{i,j}(x)), x \in [x_i, x_{i+1}] \text{ are points on } \partial\Omega, \tag{2.7a}$$

$$\left(y^D_{i,j,x}, y^U_{i,j,x}\right) = \begin{cases} (y_j, Y_{i,j}(x)), & \text{if } u_{i,j+1} < u_{i,j}, \\ (Y_{i,j}(x), y_{j+1}), & \text{if } u_{i,j+1} > u_{i,j}, \end{cases} \tag{2.7b}$$

$$F_m(x) = \int_{y^D_{i,j,x}}^{y^U_{i,j,x}} f(x,y)dy, \quad x_i \le x \le x_{i+1}, \tag{2.7c}$$

$$F_M(x) = \int_{y_j}^{y_{j+1}} f(x,y)dy, \quad x_i \le x \le x_{i+1}, \tag{2.7d}$$

$$\Lambda_{i,j} = \left\{ x \in [x_i, x_{i+1}] \;\middle|\; Y_{i,j}(x) \in [y_j, y_{j+1}] \right\}, \tag{2.7e}$$

$$\Lambda^-_{i,j} = \left\{ x \in [x_i, x_{i+1}] \;\middle|\; Y_{i,j}(x) \notin [y_j, y_{j+1}], u\left(x, \frac{y_j + y_{j+1}}{2}\right) < 0 \right\}, \tag{2.7f}$$

$$\Lambda^+_{i,j} = \left\{ x \in [x_i, x_{i+1}] \;\middle|\; Y_{i,j}(x) \notin [y_j, y_{j+1}], u\left(x, \frac{y_j + y_{j+1}}{2}\right) > 0 \right\}. \tag{2.7g}$$

Notice that $F_m(x), F_M(x)$ are smooth on $[x_i, x_{i+1}]$ and that $[x_i, x_{i+1}] = \Lambda_{i,j} \cup \Lambda^-_{i,j} \cup \Lambda^+_{i,j}$. The three sets $\Lambda_{i,j}, \Lambda^-_{i,j}, \Lambda^+_{i,j}$ are nonoverlapping. We can show that $F(x)$ is smooth on each of these sets since

$$F(x) = \begin{cases} F_m(x), & x \in \Lambda_{i,j}, \\ 0, & x \in \Lambda^-_{i,j}, \\ F_M(x), & x \in \Lambda^+_{i,j}, \end{cases} \tag{2.8}$$

Therefore

$$I_{i,j} = \int_{x_i}^{x_{i+1}} F(x)dx = \int_{\Lambda_{i,j}} F_m(x)dx + \int_{\Lambda^+_{i,j}} F_M(x)dx. \tag{2.9}$$

With formula (2.9) $I_{i,j}$ is split into two parts, each of which is a one dimensional integral with smooth integrands. Thus (2.9) is a suitable formula for designing high order numerical methods.

In order to construct numerical methods based on (2.9), we need to provide approximation to the integral domains $\Lambda_{i,j}, \Lambda^+_{i,j}$. The domain $\Lambda_{i,j}$ is the projection to x-axis of $\partial\Omega$ restricted to the cell $C_{i,j}$. Constructing approximate $\Lambda_{i,j}$ has been studied in [19] for designing high order methods to two dimensional delta function integrals. Here for completeness we present the approach provided in [19] yielding the approximate $\Lambda_{i,j}$. One can refer to [19] for more details on the design of this approach.

We approximate $\Lambda_{i,j}$ by an interval $[x^L_{i,j}, x^R_{i,j}]$ contained in $[x_i, x_{i+1}]$. By using the fact that the boundary points of $\Lambda_{i,j}$ are the x-component of the intersection points between $\partial\Omega$ and the edges of the cell $C_{i,j}$, we have the following strategy to obtain $x^L_{i,j}, x^R_{i,j}$.

Strategy I: give $x_{i,j}^L, x_{i,j}^R$

$x_{i,j}^L = x_{i+1}$

- ❍ if $u_{i,j}u_{i,j+1} \le 0$, then $x_{i,j}^L = x_i$
- ❍ else
  - ❑ if $u_{i,j+1}u_{i+1,j+1} < 0$

    Let $(\widetilde{x}, y_{j+1}), x_i < \widetilde{x} < x_{i+1}$ be one zero point of u, set $x_{i,j}^L = \min(x_{i,j}^L, \widetilde{x})$
  - ❑ if $u_{i,j}u_{i+1,j} < 0$

    Let $(\widetilde{x}, y_j), x_i < \widetilde{x} < x_{i+1}$ be one zero point of u, set $x_{i,j}^L = \min(x_{i,j}^L, \widetilde{x})$
- ❍ end

$x_{i,j}^R = x_i$

- ❍ if $u_{i+1,j}u_{i+1,j+1} \le 0$, then $x_{i,j}^R = x_{i+1}$
- ❍ else
  - ❑ if $u_{i,j+1}u_{i+1,j+1} < 0$

    Let $(\widetilde{x}, y_{j+1}), x_i < \widetilde{x} < x_{i+1}$ be one zero point of u, set $x_{i,j}^R = \max(x_{i,j}^R, \widetilde{x})$
  - ❑ if $u_{i,j}u_{i+1,j} < 0$

    Let $(\widetilde{x}, y_j), x_i < \widetilde{x} < x_{i+1}$ be one zero point of u, set $x_{i,j}^R = \max(x_{i,j}^R, \widetilde{x})$
- ❍ end

The computation of $\widetilde{x}$ in Strategy I when $u_{i,k}u_{i+1,k} < 0, k = j$ or $j+1$ is given by the following algorithm in which we let $\widetilde{x}_c$ denote the computed $\widetilde{x}$.

Algorithm I: give computed $\widetilde{x}$

- Choose an Rth-order interpolation polynomial to $u(x, y_k)$ near the interval $[x_i, x_{i+1}]$.
- Let the initial guess x_{ini} to be the root of the first-order interpolation polynomial and use Newton iteration to compute the zero point of the Rth-order interpolation polynomial.
- If the iterated values of the zero point during the Newton iteration come outside the interval $[x_i, x_{i+1}]$, then quit the Newton iteration and set $\widetilde{x}_c = x_{ini}$.
- Otherwise set $\widetilde{x}_c = x_{New}$, where x_{New} denotes the computed zero point by the Newton iteration with given tolerance E_T. Namely the absolute value of the polynomial at x_{New} is less than E_T.

The reason of setting $\widetilde{x}_c = x_{ini}$ when the Newton iteration comes outside the interval $[x_i, x_{i+1}]$ is that this implies that the derivative of $u(x, y_k)$ is close to zero on $[x_i, x_{i+1}]$, and as will be shown in Section 3 in the error estimates for our methods that in this case it is not necessary to obtain accurate approximation to $\widetilde{x}$. This has also been discussed in [19].

With Strategy I and Algorithm I, we can compute $x_{i,j}^L, x_{i,j}^R$ and obtain their approximate values $\widehat{x}_{i,j}^L, \widehat{x}_{i,j}^R$. We remark that it is possible that $[x_{i,j}^L, x_{i,j}^R]$ provided by Strategy I is not

accurate enough for approximating $\Lambda_{i,j}$ for certain cells. One example is shown in the right part of Fig. 2.1 in which $\Lambda_{i,j} = [x_i, x'] \cup [x'', x_{i+1}]$ while Strategy I gives $[x^L_{i,j}, x^R_{i,j}] = [x_i, x_{i+1}]$. See more discussions in [19]. Therefore it is possible that a single $\widehat{I}_{i,j}$ computed based on formula (2.9) with inaccurate approximation to $\Lambda_{i,j}$ is not accurate enough to approximate $I_{i,j}$. The key idea in designing high order methods in this paper, in similar spirit to that adopted in [19] for treating delta function integrals, is that although the approximation to the two dimensional Heaviside function integral in a single cell may not be accurate enough, for such a cell there exists certain neighboring cell so that the approximation to the Heaviside function integral in the union of the cells has high order accuracy. As an embodiment of this strategy we explain the treatment for the special case mentioned at the beginning of this subsection. For the special case that two cell vertex values of u are zero and the other two vertex values of u are negative, our treatment is to set $\widehat{I}_{i,j} = 0$. To explain the reasonableness of this treatment, we consider the case that $u_{i,j} = u_{i+1,j} = 0$ and $u_{i,j+1} < 0, u_{i+1,j+1} < 0$. Thus $u_y < 0$ near the cell $C_{i,j}$. Consider the neighboring cell $C_{i,j-1}$ in which $u_{i,j} = u_{i+1,j} = 0$ and $u_{i,j-1} > 0, u_{i+1,j-1} > 0$. For this cell Strategy I gives $x^L_{i,j-1} = x_i, x^R_{i,j-1} = x_{i+1}$. As shown in the subsequent part of the paper, in this case the integral on $\Lambda^+_{i,j}$ in formula (2.9) is approximated by zero. From (2.7b) one has for this case $\left(y^D_{i,j-1,x}, y^U_{i,j-1,x}\right) = (y_{j-1}, Y_{i,j-1}(x))$. Therefore $\widehat{I}_{i,j-1}$ can be designed to be high order accuracy to the integral $\int_{x_i}^{x_{i+1}} \int_{y_{j-1}}^{Y_{i,j-1}(x)} f(x,y)dydx$ which is $I_{i,j} + I_{i,j-1}$. Thus the setting of $\widehat{I}_{i,j} = 0$ enables that $\widehat{I}_{i,j} + \widehat{I}_{i,j-1}$ is the high order accurate approximation to $I_{i,j} + I_{i,j-1}$ and our strategy of achieving high order accuracy in the union of neighboring cells takes effect in this situation.

With $\widehat{x}^L_{i,j}, \widehat{x}^R_{i,j}$ provided by Strategy I and Algorithm I we can also construct the approximation to the domain $\Lambda^+_{i,j}$. Let $\Lambda^+_{i,j}$ be approximated by an interval $[\widetilde{x}^L_{i,j}, \widetilde{x}^R_{i,j}]$. From the definition of $\Lambda^+_{i,j}$ in (2.7g) we can set

$$\widetilde{x}^L_{i,j} = x_i, \widetilde{x}^R_{i,j} = \widehat{x}^L_{i,j}, \qquad \text{if } \widehat{x}^L_{i,j} > x_i \text{ and } u_{i,j} > 0, \tag{2.10a}$$

$$\widetilde{x}^L_{i,j} = \widehat{x}^R_{i,j} \widetilde{x}^R_{i,j} = x_{i+1}, \qquad \text{if } \widehat{x}^R_{i,j} < x_{i+1} \text{ and } u_{i+1,j} > 0, \tag{2.10b}$$

$$\widetilde{x}^L_{i,j} = \widetilde{x}^R_{i,j}, \qquad \text{otherwise.} \tag{2.10c}$$

Note that the two cases $\widehat{x}^L_{i,j} > x_i, u_{i,j} > 0$ and $\widehat{x}^R_{i,j} < x_{i+1}, u_{i+1,j} > 0$ do not happen simultaneously. The third case in (2.10c) implies the integral on $\Lambda^+_{i,j}$ in formula (2.9) is approximated by zero.

With the construction of approximate $\Lambda_{i,j}, \Lambda^+_{i,j}$, the numerical quadrature to the formula (2.9) which yields the value of $\widehat{I}_{i,j}$ is written to be

$$\begin{aligned} &\left(\widehat{x}^R_{i,j} - \widehat{x}^L_{i,j}\right) \sum_{k=1}^{K} w_k F_m\left(\widehat{x}^L_{i,j} + n_k\left(\widehat{x}^R_{i,j} - \widehat{x}^L_{i,j}\right)\right) \\ &\qquad + \left(\widetilde{x}^R_{i,j} - \widetilde{x}^L_{i,j}\right) \sum_{k=1}^{K} w_k F_M\left(\widetilde{x}^L_{i,j} + n_k\left(\widetilde{x}^R_{i,j} - \widetilde{x}^L_{i,j}\right)\right), \end{aligned} \tag{2.11}$$

where w_k and $n_k \in [0,1]$ are weights and nodal points of the quadrature rule respectively.

Therefore for computing $\widehat{I}_{i,j}$ we need to compute the values of $F_m(x), F_M(x)$ at several quadrature points belonging to $[x_i, x_{i+1}]$. The computation of these function values are per-

formed using the standard numerical quadrature similarly to (2.11) as follows

$$F_m(x) \approx \left(y^U_{i,j,x} - y^D_{i,j,x}\right) \sum_{k=1}^{K} w_k f\left(x, y^D_{i,j,x} + n_k\left(y^U_{i,j,x} - y^D_{i,j,x}\right)\right), \tag{2.12a}$$

$$F_M(x) \approx (y_{j+1} - y_j) \sum_{k=1}^{K} w_k f\left(x, y_j + n_k\left(y_{j+1} - y_j\right)\right), \tag{2.12b}$$

where $y^D_{i,j,x}, y^U_{i,j,x}$ are defined in (2.7b), in which the computation of $Y_{i,j}(x)$ is achieved using Newton iteration. We notice that since $[\widehat{x}^L_{i,j}, \widehat{x}^R_{i,j}]$ can be insufficiently accurate approximation to $\Lambda_{i,j}$, it is also possible that a quadrature point $x \in [\widehat{x}^L_{i,j}, \widehat{x}^R_{i,j}]$ does not belong to $\Lambda_{i,j}$ and thus $Y_{i,j}(x) \notin [y_j, y_{j+1}]$. This leads to that $y^U_{i,j,x} < y^D_{i,j,x}$ in (2.12a). In this case the computation of $F_m(x)$ is still performed using formula (2.12a). As mentioned before, although in this case the approximate Heaviside function integral (2.11) can be not accurate enough to approximate $I_{i,j}$, our method is designed in such a consistent way that certain neighboring cell also gives inaccurate approximate Heaviside function integral, and the approximation to the Heaviside function integral in the union of these cells can be high order accurate. In Section 3 we will give error analysis for our methods showing that our methods can achieve any desired accuracy by choosing the corresponding accuracy in the sub-algorithms.

In summary our procedure to yield $\widehat{I}_{i,j}$ is as follows. If the cell vertex values of u are all positive then $\widehat{I}_{i,j}$ is set to be the standard numerical quadrature to $\int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} f(x,y)dxdy$. If the vertex values of u are all negative then set $\widehat{I}_{i,j} = 0$. If two vertex values of u are zero and the other two vertex values are negative then we also set $\widehat{I}_{i,j} = 0$. For other nontrivial cases, we check which one of the two forms (2.3a), (2.3b) is chosen for computation according to the comparison of two quantities (2.4a), (2.4b). In the case that the form (2.3a) is chosen, we firstly compute $\widehat{x}^L_{i,j}, \widehat{x}^R_{i,j}, \widetilde{x}^L_{i,j}, \widetilde{x}^R_{i,j}$ which are the boundary points of the intervals for approximating the integral domains $\Lambda_{i,j}, \Lambda^+_{i,j}$ in (2.9). Then we set $\widehat{I}_{i,j}$ to be the numerical quadrature to the formula (2.9) as given by (2.11), with the evaluations of $F_m(x), F_M(x)$ at nodal points using the numerical quadratures (2.12a), (2.12b). We also use the Newton iteration to solve $Y_{i,j}(x)$ for evaluating $y^D_{i,j,x}, y^U_{i,j,x}$ in (2.12a). Similar strategy can be adopted in the case that the form (2.3b) is chosen for computations.

We mention the bandwidth issue of our method. Considering numerical efficiency, the level set function u only needs to be defined at grid points in a narrow band around the domain boundary $\partial\Omega$. The integrand function f is defined at grid points in the domain Ω and in the narrow band on the outer side of the domain. Similarly to the analysis given in [20], this narrow band can be determined from the cells whose distance to a mesh cell intersecting with $\partial\Omega$ is no more than $D_b \equiv [\frac{R-1}{2}]^+$ mesh cells, where R is the order of interpolation polynomial. For example our methods A, B, C which have second- to fourth-order accuracy tested in Section 4 correspond to $R = 1, 2, 3$ which gives $D_b = 0, 1, 1$ respectively.

2.2. Algorithm description

With the discussions of strategy and numerical implementation of our method in the above subsection, we now present the algorithm of our method. The algorithm is given by

Algorithm I*

- For each cell $C_{i,j}$ compute $\widehat{I}_{i,j}$ which is the approximation to $I_{i,j}$.
- Sum up $\widehat{I}_{i,j}$ for all mesh cells to give the numerical approximation to the Heaviside function integral (1.1).

In the above algorithm the key algorithm to compute $\widehat{I}_{i,j}$ for each cell is given as follows

Algorithm II*: give $\widehat{I}_{i,j}$

- If the cell vertex values of u are all positive then set $\widehat{I}_{i,j}$ to be an Sth-order numerical quadrature to $\int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} f(x,y)dxdy$, in which the values of f not at grid points are approximated by an Rth-order polynomial interpolation.
- If the vertex values of u are all negative then set $\widehat{I}_{i,j} = 0$.
- If two vertex values of u are zero and the other two vertex values of u are negative then set $\widehat{I}_{i,j} = 0$.
- In other cases first compare the quantities $\widetilde{u}^x_{i,j}, \widetilde{u}^y_{i,j}$ defined in (2.4a), (2.4b).

  If $\widetilde{u}^x_{i,j} < \widetilde{u}^y_{i,j}$,

  Set $\widehat{I}_{i,j}$ to be the quadrature formula (2.11), with $\widehat{x}^L_{i,j}, \widehat{x}^R_{i,j}$ given by Strategy I and Algorithm I in Subsection 2.1, $\widetilde{x}^L_{i,j}, \widetilde{x}^R_{i,j}$ given by (2.10c), and values of $F_m(x), F_M(x)$ at nodal points computed by numerical quadratures (2.12a), (2.12b), in which the values of f not at grid points are approximated by an Rth-order polynomial interpolation. The computation of $Y_{i,j}(x)$ for evaluating $y^D_{i,j,x}, y^U_{i,j,x}$ in (2.12a) is achieved using an Rth-order polynomial interpolation and Newton iteration with tolerance E_T.

  If $\widetilde{u}^x_{i,j} \geq \widetilde{u}^y_{i,j}$,

  $\widehat{I}_{i,j}$ can be computed in similar principle based on the form (2.3b).

Algorithm II* include the following parameters: R, E_T, S, where R is the order of interpolation polynomial, E_T is the tolerance in Newton iteration and S denotes the order of the quadrature rule used in the algorithm respectively. In the next section we will investigate error estimates for our method given by Algorithm I* and Algorithm II*. We show that our method can achieve any desired convergence order by selecting corresponding parameters.

3. Error Estimates

In this section we give error estimates for our method given by Algorithm I* and Algorithm II* which show that our method can achieve any desired convergence order.

The main Theorem we will prove in this section is as follows

Theorem 3.1. *Let $m_0 = \min(R+1, S)$, where R, S are parameters in Algorithm II*. Assume parameter E_T in Algorithm II* satisfies $E_T = \mathcal{O}(h^{R+1})$, the integrand and level set functions f, u in (1.1) have bounded $\max(R+1, S)$th, $R+1$th derivative respectively. Then the method given by Algorithm I* and Algorithm II* is m_0th-order accurate, namely*

$$\left| \sum_{(i,j)\in\mathbb{Z}^2} \widehat{I}_{i,j} - \int_{\mathbb{R}^2} f(x,y)H(u(x,y))dxdy \right| = \mathcal{O}(h^{m_0}). \tag{3.1}$$

In the following we present some lemmas. We firstly give some definitions which have also been adopted in analyzing the numerical methods to two dimensional delta function integrals [19].

Consider a cell $C_{i,j}$ having intersection with $\partial\Omega$. Recall the definition of $Y_{i,j}(x), x \in [x_i, x_{i+1}]$ in (2.7a), which is defined if the quantities $\widetilde{u}^x_{i,j}, \widetilde{u}^y_{i,j}$ given in (2.4a), (2.4b) satisfy $\widetilde{u}^x_{i,j} < \widetilde{u}^y_{i,j}$. Similarly if $\widetilde{u}^x_{i,j} \geq \widetilde{u}^y_{i,j}$, we can define the smooth function $X_{i,j}(y), y \in [y_j, y_{j+1}]$ as follows

$$X_{i,j}(y) \text{ satisfying that } (X_{i,j}(y), y), y \in [y_j, y_{j+1}] \text{ are points on } \partial\Omega. \tag{3.2}$$

Introduce the quantity

$$d_{i,j} = \begin{cases} 0, & \text{if } \widetilde{u}^x_{i,j} < \widetilde{u}^y_{i,j}, \\ 1, & \text{if } \widetilde{u}^x_{i,j} \geq \widetilde{u}^y_{i,j}. \end{cases} \tag{3.3}$$

Define the sets

$$D_{i,j} = \begin{cases} \{(i,l)|\ \exists x \in [x_i, x_{i+1}], s.t. (x, Y_{i,j}(x)) \in C_{i,l}\}, & \text{if } d_{i,j} = 0, \\ \{(k,j)|\ \exists y \in [y_j, y_{j+1}], s.t. (X_{i,j}(y), y) \in C_{k,j}\}, & \text{if } d_{i,j} = 1, \end{cases} \tag{3.4}$$

$$E_{i,j} = \begin{cases} D_{i,j}, & \text{if } d_{k,l} = d_{i,j}, \forall (k,l) \in D_{i,j}, \\ \{(i,j)\}, & \text{if } \exists (k,l) \in D_{i,j} \text{ s.t. } d_{k,l} \neq d_{i,j}. \end{cases} \tag{3.5}$$

We see that if the elements of $D_{i,j}$ yield the same relation between the two components of the gradient of u, then $E_{i,j}$ is the same as $D_{i,j}$. Otherwise $E_{i,j}$ contains only the element (i,j). Denote

$$M_I = \inf_{(x,y)\in\Gamma} \sqrt{(u_x(x,y))^2 + (u_y(x,y))^2}, \tag{3.6}$$

which is a positive quantity for a well-defined level set function u.

The following Lemmas 3.1 and 3.2 can be directly checked.

Lemma 3.1. *Assume parameters E_T, R in Algorithm II* satisfy $E_T = \mathcal{O}(h^{R+1})$, the integrand and level set functions f, u have bounded $\max(R+1, S)$th, $R+1$th derivative respectively, the cell $C_{i,j}$ belongs to nontrivial cases for computing $\widehat{I}_{i,j}$ and $d_{i,j} = 0$. Then $\widehat{I}_{i,j}$ given by Algorithm II* satisfies*

$$\widehat{I}_{i,j} - \int_{\widehat{x}^L_{i,j}}^{\widehat{x}^R_{i,j}} F_m(x)dx - \int_{\widetilde{x}^L_{i,j}}^{\widetilde{x}^R_{i,j}} F_M(x)dx = \mathcal{O}(h^{m_0+1}), \tag{3.7}$$

where $m_0 = \min(R+1, S)$ is defined in Theorem 3.1.

Lemma 3.2. *For a cell $C_{i,j}$ satisfying $d_{i,j} = 0$, it holds*

$$\sum_{(i,l)\in D_{i,j}} I_{i,l} = \begin{cases} \int_{x_i}^{x_{i+1}} \int_{y_{j_1}}^{Y_{i,j}(x)} f(x,y)\,dydx, & \text{if } u_{i,j+1} < u_{i,j}, \\ \int_{x_i}^{x_{i+1}} \int_{Y_{i,j}(x)}^{y_{j_2+1}} f(x,y)\,dydx, & \text{if } u_{i,j+1} > u_{i,j}, \end{cases} \tag{3.8}$$

where $j_1 = \min\limits_{(i,l)\in D_{i,j}} l, j_2 = \max\limits_{(i,l)\in D_{i,j}} l.$

With the above lemmas we prove the following Lemma 3.3. The proof is classified according to $E_{i,j} = D_{i,j}$ and $E_{i,j} \neq D_{i,j}$. For the latter case one has $E_{i,j} = \{(i,j)\}$. The former case includes that $E_{i,j}$ can contain the index (i,j) as well as its neighboring ones. Therefore Lemma 3.3 ensures that the summation $\sum_{(k,l)\in E_{i,j}} \widehat{I}_{k,l}$ is the high order accurate approximation to $\sum_{(k,l)\in E_{i,j}} I_{k,l}$, while each $\widehat{I}_{k,l}$ is not necessarily accurate enough to approximate $I_{k,l}$. This is the key point in the proof of error estimates for our method, as mentioned in Introduction and Subsection 2.1.

Lemma 3.3. *Assume parameters E_T, R in Algorithm II* satisfy $E_T = \mathcal{O}(h^{R+1})$, the integrand and level set functions f, u have bounded $\max(R+1, S)$th, $R+1$th derivative respectively. Then $\widehat{I}_{k,l}$ for (k,l) adjacent to (i,j) yielded by Algorithm II* have the following error estimates*

$$\sum_{(k,l)\in E_{i,j}} \left(\widehat{I}_{k,l} - I_{k,l}\right) = \mathcal{O}(h^{m_0+1}), \tag{3.9}$$

where $m_0 = \min(R+1, S)$ is defined in Theorem 3.1.

Proof. We prove (3.9) for the case that $d_{i,j} = 0$. The case $d_{i,j} = 1$ can be similarly analyzed.

We firstly prove (3.9) in the case of $E_{i,j} = D_{i,j}$. According to (3.5) this implies $d_{k,l} = d_{i,j}, \forall (k,l) \in D_{i,j}$. Let $N_{i,j}$ denote the number of elements in the set $D_{i,j}$. According to Lemma 3.1 in [19], we have that $N_{i,j} \leq 3$ for fine enough mesh. The following discussions are classified according to value of $N_{i,j}$.

If $N_{i,j} = 1$, namely $D_{i,j} = \{(i,j)\}$, then $y_j < Y_{i,j}(x) < y_{j+1}$, for $x \in [x_i, x_{i+1}]$ and thus $\Lambda_{i,j} = [x_i, x_{i+1}], \Lambda^+_{i,j} = \emptyset$. In this case Strategy I gives $\widehat{x}^L_{i,j} = x^L_{i,j} = x_i, \widehat{x}^R_{i,j} = x^R_{i,j} = x_{i+1}$, and from (2.10c) one has $\widetilde{x}^L_{i,j} = \widetilde{x}^R_{i,j}$. According to Lemma 3.1 and formula (2.9) one has

$$\begin{aligned} \widehat{I}_{i,j} &= \int_{\widehat{x}^L_{i,j}}^{\widehat{x}^R_{i,j}} F_m(x)dx + \mathcal{O}(h^{m_0+1}) \\ &= \int_{\Lambda_{i,j}} F_m(x)dx + \mathcal{O}(h^{m_0+1}) = I_{i,j} + \mathcal{O}(h^{m_0+1}). \end{aligned} \tag{3.10}$$

If $N_{i,j} = 2$, then the two cells corresponding to the elements in $D_{i,j}$ are adjacent since $\partial\Omega$ is smooth. Without loss of generality we assume the two elements in $D_{i,j}$ are (i,j) and $(i,j+1)$. Denote $\widehat{Y}_i = Y_{i,j}(x_i), \widehat{Y}_{i+1} = Y_{i,j}(x_{i+1})$. There are four cases for $\widehat{Y}_i, \widehat{Y}_{i+1}$:

i) $\widehat{Y}_i = \widehat{Y}_{i+1} = y_{j+1}$;

ii) one of $\widehat{Y}_i, \widehat{Y}_{i+1}$ is y_{j+1}, the other is not;

iii) $\left(\widehat{Y}_i - y_{j+1}\right)\left(\widehat{Y}_{i+1} - y_{j+1}\right) > 0$;

iv) $\left(\widehat{Y}_i - y_{j+1}\right)\left(\widehat{Y}_{i+1} - y_{j+1}\right) < 0$.

For case i), without loss of generality we discuss $u_y < 0$ near the cell $C_{i,j}$. Namely $u_{k,j+2} < 0, u_{k,j} > 0, k = i, i+1$. The cell $C_{i,j+1}$ belongs to the special case that two vertex values of u are zero and the other two vertex values of u are negative. Thus Algorithm II* gives $\widehat{I}_{i,j+1} = 0$. For cell $C_{i,j}$ Strategy I and (2.10c) give $\widehat{x}^L_{i,j} = x^L_{i,j} = x_i, \widehat{x}^R_{i,j} = x^R_{i,j} = x_{i+1}, \widetilde{x}^L_{i,j} = \widetilde{x}^R_{i,j}$. Using

Lemmas 3.1 and 3.2 and expressions (2.7b), (2.7c) we then have

$$\begin{aligned}\widehat{I}_{i,j} =& \int_{\widehat{x}^L_{i,j}}^{\widehat{x}^R_{i,j}} F_m(x)dx + \mathcal{O}(h^{m_0+1}) \\ =& \int_{x_i}^{x_{i+1}} \int_{y_j}^{Y_{i,j}(x)} f(x,y)\, dydx + \mathcal{O}(h^{m_0+1}) \\ =& I_{i,j} + I_{i,j+1} + \mathcal{O}(h^{m_0+1}) \end{aligned} \tag{3.11}$$

which leads to (3.9).

The analysis of case ii) is similar to case iii). For case iii), without loss of generality we discuss $\widehat{Y}_i, \widehat{Y}_{i+1} < y_{j+1}$. For the case $u_y < 0$ near $C_{i,j}$, the four vertex values of u for the cell $C_{i,j+1}$ are all negative. Thus Algorithm II* gives $\widehat{I}_{i,j+1} = 0$. Similarly to the deduction of (3.11) for this case one has

$$\widehat{I}_{i,j} = I_{i,j} + I_{i,j+1} + \mathcal{O}(h^{m_0+1}),$$

which leads to (3.9). For the case $u_y > 0$ near $C_{i,j}$, the four vertex values of u for the cell $C_{i,j+1}$ are all positive. Algorithm II* gives

$$\widehat{I}_{i,j+1} = \int_{x_i}^{x_{i+1}} \int_{y_{j+1}}^{y_{j+2}} f(x,y)dydx + \mathcal{O}(h^{m_0+2}).$$

For cell $C_{i,j}$ Strategy I and (2.10c) give $\widehat{x}^L_{i,j} = x^L_{i,j} = x_i, \widehat{x}^R_{i,j} = x^R_{i,j} = x_{i+1}, \widetilde{x}^L_{i,j} = \widetilde{x}^R_{i,j}$. Then using Lemmas 3.1 and 3.2 and expressions (2.7b), (2.7c) we have

$$\begin{aligned}\widehat{I}_{i,j} + \widehat{I}_{i,j+1} =& \int_{x_i}^{x_{i+1}} \int_{Y_{i,j}(x)}^{y_{j+1}} f(x,y)\, dydx + \int_{x_i}^{x_{i+1}} \int_{y_{j+1}}^{y_{j+2}} f(x,y)dydx + \mathcal{O}(h^{m_0+1}) \\ =& I_{i,j} + I_{i,j+1} + \mathcal{O}(h^{m_0+1}). \end{aligned} \tag{3.12}$$

For case iv), without loss of generality we discuss $\widehat{Y}_i > y_{j+1}, \widehat{Y}_{i+1} < y_{j+1}$ and $u_y < 0$ near $C_{i,j}$. For this case Strategy I, Algorithm I and (2.10c) give $x_i = \widehat{x}^L_{i,j+1} < \widehat{x}^R_{i,j+1} = \widehat{x}^L_{i,j} < \widehat{x}^R_{i,j} = x_{i+1}$ and $\widetilde{x}^L_{i,j+1} = \widetilde{x}^R_{i,j+1}, \widetilde{x}^L_{i,j} = x_i, \widetilde{x}^R_{i,j} = \widehat{x}^L_{i,j}$. Then using Lemmas 3.1 and 3.2 and expressions (2.7b), (2.7c), (2.7d), we have

$$\begin{aligned}\widehat{I}_{i,j} + \widehat{I}_{i,j+1} =& \int_{\widehat{x}^L_{i,j}}^{\widehat{x}^R_{i,j}} \int_{y_j}^{Y_{i,j}(x)} f(x,y)\, dydx + \int_{\widetilde{x}^L_{i,j}}^{\widetilde{x}^R_{i,j}} \int_{y_j}^{y_{j+1}} f(x,y)\, dydx \\ &+ \int_{\widehat{x}^L_{i,j+1}}^{\widehat{x}^R_{i,j+1}} \int_{y_{j+1}}^{Y_{i,j}(x)} f(x,y)\, dydx + \mathcal{O}(h^{m_0+1}) \\ =& \int_{x_i}^{\widehat{x}^L_{i,j}} \int_{y_j}^{Y_{i,j}(x)} f(x,y)\, dydx + \int_{\widehat{x}^L_{i,j}}^{x_{i+1}} \int_{y_j}^{Y_{i,j}(x)} f(x,y)\, dydx + \mathcal{O}(h^{m_0+1}) \\ =& I_{i,j} + I_{i,j+1} + \mathcal{O}(h^{m_0+1}). \end{aligned} \tag{3.13}$$

If $N_{i,j} = 3$, without loss of generality we assume the three adjacent elements in $D_{i,j}$ are $(i,j), (i,j+1), (i,j+2)$. Since the curve $(x, Y_{i,j}(x)), x \in [x_i, x_{i+1}]$ occupies three cells, one has that $Y_{i,j}(x)$ is monotone on $[x_i, x_{i+1}]$. Without loss of generality we discuss $\widehat{Y}_i \geq y_{j+2}, \widehat{Y}_{i+1} \leq y_{j+1}$ and $u_y < 0$ near $C_{i,j}$. If at least one of $\widehat{Y}_i, \widehat{Y}_{i+1}$ equals y_{j+1}, then we can deduce (3.9) similarly to the cases $N_{i,j} = 1$ or 2. For the case $\widehat{Y}_i > y_{j+2}, \widehat{Y}_{i+1} < y_{j+1}$ and reasonably fine mesh, as discussed in the proof of Lemma 3.7 in [19], Strategy I and Algorithm I give

$$x_i = \widehat{x}^L_{i,j+2} < \widehat{x}^R_{i,j+2} = \widehat{x}^L_{i,j+1} < \widehat{x}^R_{i,j+1} = \widehat{x}^L_{i,j} < \widehat{x}^R_{i,j} = x_{i+1}.$$

Formula (2.10c) gives

$$\widetilde{x}^L_{i,j+2}=\widetilde{x}^R_{i,j+2},\widetilde{x}^L_{i,j+1}=x_i,\widetilde{x}^R_{i,j+1}=\widehat{x}^L_{i,j+1},\widetilde{x}^L_{i,j}=x_i,\widetilde{x}^R_{i,j}=\widehat{x}^L_{i,j}.$$

Then using Lemmas 3.1 and 3.2 and expressions (2.7b), (2.7c), (2.7d) we have

$$\begin{aligned}\widehat{I}_{i,j}+\widehat{I}_{i,j+1}+\widehat{I}_{i,j+2}=&\int_{\widehat{x}^L_{i,j}}^{\widehat{x}^R_{i,j}}\int_{y_j}^{Y_{i,j}(x)}f(x,y)\,dydx+\int_{\widetilde{x}^L_{i,j}}^{\widetilde{x}^R_{i,j}}\int_{y_j}^{y_{j+1}}f(x,y)\,dydx\\&+\int_{\widehat{x}^L_{i,j+1}}^{\widehat{x}^R_{i,j+1}}\int_{y_{j+1}}^{Y_{i,j}(x)}f(x,y)\,dydx+\int_{\widetilde{x}^L_{i,j+1}}^{\widetilde{x}^R_{i,j+1}}\int_{y_{j+1}}^{y_{j+2}}f(x,y)\,dydx\\&+\int_{\widehat{x}^L_{i,j+2}}^{\widehat{x}^R_{i,j+2}}\int_{y_{j+2}}^{Y_{i,j}(x)}f(x,y)\,dydx+\mathcal{O}(h^{m_0+1})\\=&\int_{x_i}^{\widehat{x}^L_{i,j+1}}\int_{y_j}^{Y_{i,j}(x)}f(x,y)\,dydx+\int_{\widehat{x}^L_{i,j+1}}^{\widehat{x}^R_{i,j+1}}\int_{y_j}^{Y_{i,j}(x)}f(x,y)\,dydx\\&+\int_{\widehat{x}^L_{i,j}}^{x_{i+1}}\int_{y_j}^{Y_{i,j}(x)}f(x,y)\,dydx+\mathcal{O}(h^{m_0+1})\\=&I_{i,j}+I_{i,j+1}+I_{i,j+2}+\mathcal{O}(h^{m_0+1}).\end{aligned}\tag{3.14}$$

From the above discussions, we have proved that in the case of $E_{i,j}=D_{i,j}$, for all possible cases $N_{i,j}=1,2$ or 3 the estimate (3.9) can always be proved.

We then prove (3.9) when $E_{i,j}\neq D_{i,j}$. From (3.5) for this case one has $E_{i,j}=\{(i,j)\}$ and $\exists(i,l)\in D_{i,j}$ s.t. $d_{i,l}=1\neq d_{i,j}=0$. From the deduction in the proof of Lemma 3.7 in [19], one has that for this case and reasonably fine mesh it holds

$$\big|u_x\big(x,Y_{i,j}(x)\big)\big|\geq\frac{M_I}{\sqrt{5}},\quad\forall\big(x,Y_{i,j}(x)\big)\in\bigcup_{(i,k)\in D_{i,j}}C_{i,k},\tag{3.15}$$

where M_I is defined in (3.6). Therefore $Y_{i,j}(x)$ is monotone on $x\in[x_i,x_{i+1}]$ which implies $\Lambda_{i,j}=[x^L_{i,j},x^R_{i,j}]$. There are three cases for $\Lambda^+_{i,j}$:

i*) $x^L_{i,j}>x_i$ and $\Lambda^+_{i,j}=[x_i,x^L_{i,j}]$;

ii*) $x^R_{i,j}<x_{i+1}$ and $\Lambda^+_{i,j}=[x^R_{i,j},x_{i+1}]$;

iii*) $\Lambda^+_{i,j}=\emptyset$.

We discuss the case i*). The other two cases can be similarly analyzed. For case i*), formula (2.9) gives

$$I_{i,j}=\int_{x^L_{i,j}}^{x^R_{i,j}}F_m(x)dx+\int_{x_i}^{x^L_{i,j}}F_M(x)dx.\tag{3.16}$$

Utilizing (3.15), according to Lemma 3.5 in [19] one has

$$\widehat{x}^L_{i,j}=x^L_{i,j}+\mathcal{O}(h^{R+1}),\quad\widehat{x}^R_{i,j}=x^R_{i,j}+\mathcal{O}(h^{R+1}).\tag{3.17}$$

Case i*) implies $u_{i,j}>0$. Thus (2.10c) gives $\widetilde{x}^L_{i,j}=x_i,\widetilde{x}^R_{i,j}=\widehat{x}^L_{i,j}$. Combining (3.16), (3.17) and using Lemma 3.1 gives

$$I_{i,j}=\int_{\widehat{x}^L_{i,j}}^{\widehat{x}^R_{i,j}}F_m(x)dx+\int_{x_i}^{\widehat{x}^L_{i,j}}F_M(x)dx+\mathcal{O}(h^{R+2})=\widehat{I}_{i,j}+\mathcal{O}(h^{m_0+1}).\tag{3.18}$$

Thus we prove (3.9) in the case of $E_{i,j} \neq D_{i,j}$. □

We now give the proof of the main Theorem on the error estimates for our method.

3.1. Proof of Theorem 3.1

Proof. Denote

$$S_U = \Big\{(k,l) \;\Big|\; \text{the cell } C_{k,l} \text{ contains points on } \partial\Omega\Big\}. \tag{3.19}$$

According to definition of $I_{i,j}$ and Algorithm II*, for any $(i,j) \notin S_U$ one has

$$I_{i,j} = \widehat{I}_{i,j} = 0, \quad \text{if } u_{i,j} < 0, \tag{3.20a}$$

$$I_{i,j} = \int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} f(x,y)dxdy, \;\; \widehat{I}_{i,j} = I_{i,j} + \mathcal{O}(h^{m_0+2}), \quad \text{if } u_{i,j} > 0. \tag{3.20b}$$

Therefore

$$\sum_{(i,j)\notin S_U} \Big(\widehat{I}_{k,l} - I_{k,l}\Big) = \mathcal{O}(h^{m_0}). \tag{3.21}$$

By Lemma 3.2 in [19], the set $E_{i,j}$ satisfies the following properties:

i) $(i,j) \in E_{i,j}$;

ii) $E_{k,l} = E_{i,j}, \forall (k,l) \in E_{i,j}$;

iii) $E_{k,l} \cap E_{i,j} = \emptyset$, if $(k,l) \notin E_{i,j}$.

Then according to the proof of Theorem 3.1 in [19], the following equalities can be deduced:

$$\sum_{(i,j)\in S_U} I_{i,j} = \sum_{(i,j)\in S_U} \frac{1}{N_{i,j}} \left(\sum_{(k,l)\in E_{i,j}} I_{k,l} \right), \tag{3.22}$$

$$\sum_{(i,j)\in S_U} \widehat{I}_{i,j} = \sum_{(i,j)\in S_U} \frac{1}{N_{i,j}} \left(\sum_{(k,l)\in E_{i,j}} \widehat{I}_{k,l} \right). \tag{3.23}$$

Now utilizing (3.21)-(3.23) and (2.2), applying Lemma 3.3 and noticing that the number of elements in S_U is $\mathcal{O}\left(\frac{1}{h}\right)$ one obtains

$$\begin{aligned} &\left| \sum_{(i,j)\in\mathbb{Z}^2} \widehat{I}_{i,j} - \int_{\mathbb{R}^2} f(x,y)H(u(x,y))dxdy \right| \\ =&\left| \sum_{(i,j)\in S_U} \frac{1}{N_{i,j}} \left(\sum_{(k,l)\in E_{i,j}} \Big(\widehat{I}_{k,l} - I_{k,l}\Big) \right) \right| + \mathcal{O}(h^{m_0}) = \mathcal{O}(h^{m_0}), \end{aligned} \tag{3.24}$$

which completes the proof of Theorem 3.1. □

4. Numerical Examples

In this section we present numerical examples to show the accuracy of our methods. In the following examples we use Simpson rule in the numerical quadratures (2.11)-(2.12b), namely we choose $S = 4$ in Algorithm II*. For other parameters in Algorithm II* we test the following three set of choices:

Method A: $R = 1, E_T = h^2$,
Method B: $R = 2, E_T = h^3$,
Method C: $R = 3, E_T = h^4$.

According to Theorem 3.1, if the integrand and level set functions have sufficient smoothness then methods A, B, C have second-, third- and fourth-order accuracy respectively. We will show that these methods achieve or behave better than the expected accuracy in our numerical examples. We also give a comparison on numerical accuracy between the high order version method in [15] and our high order methods in the last example, which shows the advantage of our third- and fourth-order methods.

Example 4.1. This is an example tested in [5]. Let

$$u(x,y) = -\left(2\sqrt{x^2+y^2} - 2 - \frac{x^5+5xy^4-10x^3y^2}{(x^2+y^2)^{2.5}}\right), \qquad f(x,y) = 1.$$

The set $\Omega = \{(x,y)|u(x,y) \geq 0\}$ is an irregular domain represented in the polar coordinates as $r \leq 1 + \frac{1}{2}\cos(5\theta)$, which is shown in Fig. 4.1. The exact value of (1.1) corresponding to the area of Ω is $\frac{9}{8}\pi$. For a given mesh size we randomly shift the uniform mesh in the x and y directions and rotated for 50 times. Table 4.1 presents the averaged relative errors of the three methods over the 50 trials. The last column in the table presents the estimated convergence rates from the numerical errors. In this example methods A, C achieve the expected second- and fourth-order accuracy, while method B also behaves fourth-order accurate, better than the third-order accuracy ensured by Theorem 3.1.

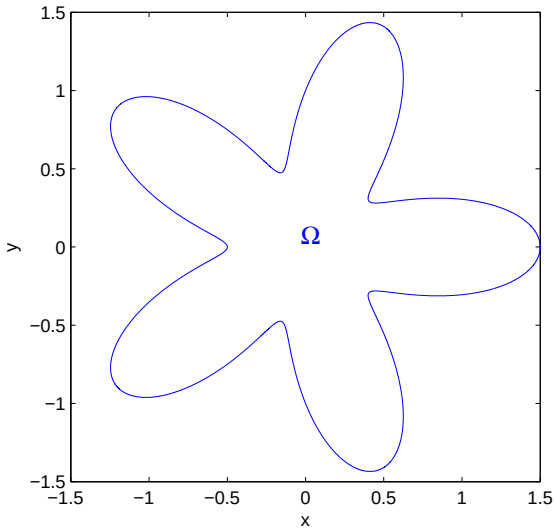

Fig. 4.1. The domain Ω for Example 4.1.

Table 4.1: Example 4.1, averaged relative errors of methods A, B, C

Mesh size	0.1	0.05	0.025	0.0125	0.00625	0.003125	R_e
method A	3.41E-3	1.01E-3	2.59E-4	6.61E-5	1.65E-5	4.14E-6	1.95
method B	1.19E-3	9.63E-5	5.68E-6	3.73E-7	1.91E-8	1.12E-9	4.03
method C	1.31E-3	9.58E-5	6.22E-6	3.91E-7	2.47E-8	1.59E-9	3.94

Example 4.2. This is an example tested in [5]. Let $u(x,y) = 1 - x^2 - y^2, f(x,y) = e^{-x^2-y^2}$. The exact value of (1.1) is $\pi\left(1 - \frac{1}{e}\right)$. In [5] the authors considered defining $f(x,y)$ only inside the domain $\{(x,y)|u(x,y) \geq 0\}$ and extending the values of f outside the domain by extrapolation to simulate the situations in the application of evaluating singular source terms. In this paper we do not examine this issue while only focus on illustrating the high order accuracy of our methods. Table 4.2 presents the averaged relative errors of the three methods over 50 trials in which the uniform computational mesh is randomly shifted in the x and y directions. Similarly to the previous example methods A, C achieve the expected second and fourth order accuracy, while method B also behaves fourth-order accurate. Method B however is not generally better than third-order accuracy as shown in the next example.

Table 4.2: Example 4.2, averaged relative errors of methods A, B, C

Mesh size	0.1	0.05	0.025	0.0125	0.00625	0.003125	R_e
method A	3.88E-3	9.68E-4	2.42E-4	6.06E-5	1.52E-5	3.79E-6	2.00
method B	5.93E-5	2.76E-6	1.34E-7	7.29E-9	3.92E-10	2.23E-11	4.26
method C	1.64E-5	1.01E-6	6.25E-8	3.89E-9	2.42E-10	1.52E-11	4.01

Example 4.3. Let

$$u(x,y) = -\left(\frac{x^2}{(1.5)^2} + \frac{y^2}{(0.75)^2} - 1\right)\left(\frac{(x-0.5)^2}{(0.5)^2} + \frac{y^2}{(0.4)^2} - 1\right), \quad f(x,y) = e^{\frac{x^2}{(1.5)^2} + \frac{y^2}{(0.75)^2}}.$$

The domain $\Omega = \{(x,y)|u(x,y) \geq 0\}$ is enclosed by two elliptic curves as shown in Fig. 4.2. The exact value of (1.1) is approximately 5.29117243471186. Table 4.3 presents the averaged

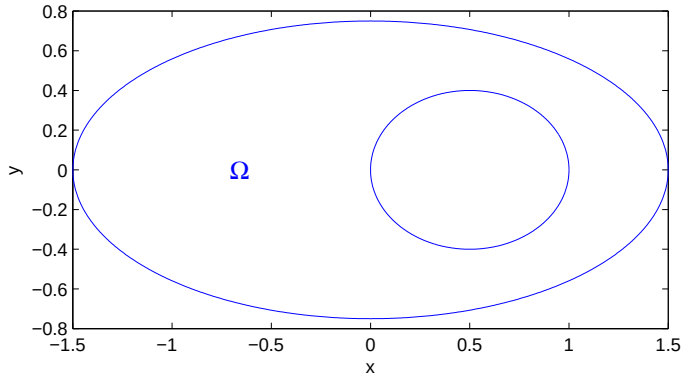

Fig. 4.2. The domain Ω for Example 4.3.

relative errors of the three methods over 50 trials in which the uniform computational mesh is randomly shifted in the x and y directions and rotated. The methods A, B, C are observed to be second- to fourth-order accurate respectively as ensured by Theorem 3.1. We also compare our results with that using the high order version method proposed in [15] which is called method D in this test. As shown in [15] method D generally behaves fourth-order convergent for computing multidimensional Heaviside function integrals. The results in Table 4.3 verify that method D behaves fourth-order convergent for this example. The comparison shows that our third- and fourth-order methods B, C are more accurate than method D up to the mesh sizes tested in this example.

Table 4.3: Example 4.3, averaged relative errors of methods A, B, C, D

Mesh size	0.1	0.05	0.025	0.0125	0.00625	0.003125	0.0015625	R_e
method A	1.55E-2	3.87E-3	9.76E-4	2.44E-4	6.09E-5	1.52E-5	3.80E-6	2.00
method B	6.84E-4	6.78E-5	8.93E-6	8.95E-7	1.11E-7	1.41E-8	1.66E-9	3.10
method C	1.41E-4	1.01E-5	7.65E-7	5.07E-8	3.56E-9	2.22E-10	1.38E-11	3.88
method D	4.53E-2	9.03E-3	2.20E-4	1.40E-5	9.19E-7	5.61E-8	3.47E-9	4.05

5. Conclusion

In this paper we studied a class of high order numerical methods to two dimensional Heaviside function integrals. Inspired by our approach for computing two dimensional delta function integrals [19], the methods were constructed by considering the approximation of the Heaviside function integral restricted to mesh cells. In each mesh cell the two dimensional Heaviside function integral can be rewritten as a one dimensional ordinary integral with the integrand being a one dimensional Heaviside function integral. The form of the one dimensional integral takes one of two choices according to the comparison of the two components of gradient of the level set function which can be checked from the mesh point values of the level set function. Under such choice, the one dimensional Heaviside function integral being the integrand of the one dimensional ordinary integral is well-defined and smooth on several subsets of the integral interval. Consequently the mesh cell restriction of the two dimensional Heaviside function integral were approximated by constructing approximate subsets of the integral interval of the one dimensional ordinary integral and applying standard one dimensional high order numerical quadratures and high order numerical methods to one dimensional Heaviside function integrals. The algorithm designed under such principle to approximate the mesh cell restrictions of the two dimensional Heaviside function integral (1.1) comprises the numerical method proposed in this paper. We established error estimates for the proposed method which show that the method can achieve any desired accuracy by choosing the corresponding accuracy in the sub-algorithms of the method. We presented numerical examples in which second- to fourth-order methods were implemented and shown to achieve or exceed the expected accuracy indicated by the error analysis. We considered the computation of two dimensional Heaviside function integrals in this paper. It is interesting to consider applying similar idea to the design of high order methods to three dimensional Heaviside function integrals in the future.

Acknowledgments. Research supported in part by the Knowledge Innovation Project of the Chinese Academy of Sciences grant K3502012S8, NSFC grant 10601062 and the National Basic

Research Program grant 2010CB731505.

References

[1] J.T. Beale, A proof that a discrete delta function is second-order accurate, *J. Comput. Phys.*, **227** (2008), 2195-2197.

[2] Y. Di, R. Li, T. Tang and P.W. Zhang, Level set calculations for incompressible two-phase flows on a dynamically adaptive grid, *J. Sci. Comput.*, **31** (2006), 75-98.

[3] B. Engquist, A.K. Tornberg and R. Tsai, Discretization of dirac delta functions in level set methods, *J. Comput. Phys.*, **207**:1 (2005), 28-51.

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

[5] C. Min and F. Gibou, Robust second-order accurate discretizations of the multi-dimensional Heaviside and Dirac delta functions, *J. Comput. Phys.*, **227** (2008), 9686-9695.

[6] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.*, **25** (1977), 220-252.

[7] C.S. Peskin, The immersed boundary method, *Acta Numerica*, **11** (2002), 479-511.

[8] P. Smereka, The numerical approximation of a delta function with application to level set methods, *J. Comput. Phys.*, **211** (2006), 77-90.

[9] A.K. Tornberg, Interface Tracking Methods with Applications to Multiphase Flows, Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.

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

[11] A.K. Tornberg and B. Engquist, A finite element based level set method for multiphase flow applications, *Comput. Vis. Sci.*, **3** (2000), 93-101.

[12] A.K. Tornberg and 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**:2 (2007), 915-931.

[14] J.D. Towers, A convergence rate theorem for finite difference approximations to delta functions, *J. Comput. Phys.*, **227**:13 (2008), 6591-6597.

[15] J.D. Towers, Finite difference methods for approximating Heaviside functions, *J. Comput. Phys.*, **228**:9 (2009), 3478-3489.

[16] J.D. Towers, Discretizing delta functions via finite differences and gradient normalization, *J. Comput. Phys.*, **228**:10 (2009), 3816-3836.

[17] X. Wen, High order numerical methods to a type of delta function integrals, *J. Comput. Phys.*, **226** (2007), 1952-1967.

[18] X. Wen, High order numerical quadratures to one dimensional delta function integrals, *SIAM J. Sci. Comput.*, **30**:4 (2008), 1825-1846.

[19] X. Wen, High order numerical methods to two dimensional delta function integrals in level set methods, *J. Comput. Phys.*, **228**:11 (2009), 4273-4290.

[20] X. Wen, High order numerical methods to three dimensional delta function integrals in level set methods, *SIAM J. Sci. Comput.*, **32**:3 (2010), 1288-1309.