Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 1952–1967
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

High order numerical methods to a type of delta function integrals

Xin Wen [*]

Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, China

Received 23 April 2007; accepted 22 June 2007
Available online 5 July 2007

Abstract

We study second to fourth order numerical methods to a type of delta function integrals in one to three dimensions. These delta function integrals arise from recent efficient level set methods for computing the multivalued solutions of nonlinear PDEs. We show that the natural quadrature approach with usual discrete delta functions and support size formulas to the two dimensional delta function integrals suffer from nonconvergence. We then design high order numerical methods to this type of delta function integrals based on interpolation approach. Numerical examples are presented to verify the efficiency and accuracy of our methods.
© 2007 Elsevier Inc. All rights reserved.

AMS subject classification: 65Jxx

Keywords: Delta function integral; Level set method; Discrete delta function

1. Introduction

In this paper we study efficient numerical methods to the following type of delta function integrals

$$\int_{\Omega} \alpha(\mathbf{x}) \prod_{i=1}^{d} \delta(\beta_i(\mathbf{x}))\mathrm{d}\mathbf{x}, \quad \Omega \subset \mathbb{R}^d, \mathbf{x} \in \mathbb{R}^d, \tag{1.1}$$

where $d = 1, 2, 3$ correspond to one to three dimensions, respectively. The values of the weight function $\alpha(\mathbf{x})$ and the level set functions $\beta_i(\mathbf{x}), 1 \leqslant i \leqslant d$ are provided at grid points of a uniform mesh. We assume the functions $\alpha(\mathbf{x}), \beta_i(\mathbf{x}), 1 \leqslant i \leqslant d$ are smooth, and $\beta_i(\mathbf{x}), 1 \leqslant i \leqslant d$ have finite number of common zero points in the integral domain.

[*] Tel.: +8610 62623711; fax: +8610 62542285.
E-mail address: wenxin@amss.ac.cn

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.06.025

Numerical computations of delta function integrals (1.1) appear in the recent area of efficient level set methods for computing the multivalued solutions in the semiclassical limit of the linear Schrödinger equation and the high frequency limit of the linear wave equation [13,14]. As discussed in Section 2, this leads to the delta function integrals of (1.1) type.

Compared with (1.1), a class of delta function integrals taking the following form has been much more widely studied in the literature

$$\int_{\mathbb{R}^n} f(\mathbf{x})\|\nabla u(\mathbf{x})\|\delta(u(\mathbf{x}))\mathrm{d}\mathbf{x}, \quad n = 2, 3, \tag{1.2}$$

where $u(\mathbf{x})$ is a level set function whose zero points consist a manifold of codimension one. The functions $f(\mathbf{x}), u(\mathbf{x})$ are only defined at grid points of a uniform mesh. Numerical approximations to (1.2) widely appear in the applications of level set methods [34,29,38,11,23,26,19,22,24,4]. To approximate (1.2) using the information of $f(\mathbf{x}), u(\mathbf{x})$ at grid points, a thoroughly studied approach consists using discrete delta functions to compose numerical quadrature. Namely one uses simple quadrature to approximate (1.2) with the delta function in the integrand replaced by a discrete delta function which can be represented numerically. Among the various forms of the discrete delta functions, some simple choices of the discrete delta functions include [2,35,31,32,8]

$$\delta^L_w(x) = \begin{cases} \frac{1}{w}\left(1 - \left|\frac{x}{w}\right|\right), & \left|\frac{x}{w}\right| \leqslant 1 \\ 0, & \left|\frac{x}{w}\right| > 1 \end{cases}, \tag{1.3}$$

$$\delta^C_w(x) = \begin{cases} \frac{1}{2w}\left(1 + \cos\left(\frac{\pi x}{w}\right)\right), & \left|\frac{x}{w}\right| \leqslant 1 \\ 0, & \left|\frac{x}{w}\right| > 1 \end{cases}, \tag{1.4}$$

$$\delta^{PC}_w(x) = \begin{cases} \frac{2}{w}\left(1 - \left|\frac{x}{w}\right| - 4\left|\frac{x}{w}\right|^2 + 4\left|\frac{x}{w}\right|^3\right), & \left|\frac{x}{w}\right| \leqslant \frac{1}{2} \\ \frac{2}{w}\left(1 - \frac{11}{3}\left|\frac{x}{w}\right| + 4\left|\frac{x}{w}\right|^2 - \frac{4}{3}\left|\frac{x}{w}\right|^3\right), & \frac{1}{2} < \left|\frac{x}{w}\right| \leqslant 1 \\ 0, & \left|\frac{x}{w}\right| > 1 \end{cases}, \tag{1.5}$$

where $2w$ is the support size of the discrete delta function. This approach is easy to implement and successful in treating delta function integrals of (1.2) type. Up to second order methods based on quadrature approach have been designed and implemented in the literature [30–32,3,8,27,33].

In comparison, the numerical methods to (1.1) have been much less investigated. Due to the success of the quadrature approach in treating (1.2), it is then natural to extend the same idea to design numerical methods to the delta function integrals (1.1). Namely one approximates (1.1) by a simple quadrature with delta functions in the integrand replaced by discrete delta functions. Such a quadrature method has been used in [13,16] to approximate (1.1). The advantage of this approach is easy to implement. High order numerical quadratures in one dimension has been analyzed in [36]. However, we will show in this paper by counter examples that in two dimension such an approach with usually used discrete delta functions including (1.3)–(1.5) and currently available support size formulas suffer from nonconvergence. This implies that to successfully design quadrature methods to (1.1) in high dimensions needs further investigation, and so far no convergent numerical method to (1.1) in high dimensions has been available yet.

In this paper we investigate efficient numerical methods to (1.1) in one to three dimensions different from the quadrature approach. Our idea is to directly interpolate the exact value of the delta function integrals (1.1) from the values of the weight and level set functions at grid points. This idea is even more natural than the quadrature approach. A possible disadvantage of this method one may argue is that one needs to determine the existence of the common zero points of the level set functions in order to apply the interpolation. In comparison, such an issue is not encountered in the quadrature approach, where one just evaluates the values of the discrete delta functions and then sum them up. As we will see in this paper, to check the common zero points of the level set functions in high dimensions indeed is nontrivial. However, easy-to-implement rule to check the common zero points can still be designed, as done in this paper. After the check of a common zero point, a natural way to perform interpolation is to first approximate the common zero point position, then interpolate the exact value of the delta function integral. However this approach needs solving nonlinear algebraic equations system if high order accuracy is expected. In this paper we adapt this natural way by changing the interpolation space. Our approach does not need to explicitly determine the common zero points positions and avoids solving nonlinear algebraic equations system in order to achieve high order accuracy. In this paper we will study second to fourth order methods, respectively.

This paper is organized as follows. In Section 2, we describe the motivation of our study of the delta function integrals (1.1) in the level set methods. In Section 3, we give counter examples to show the nonconvergence of the quadrature approach with usual discrete delta functions and currently available support size formulas to (1.1) in two dimension. In Section 4, we design efficient methods to (1.1) based on interpolation approach in one to three dimensions gradually. In Section 5, we give numerical examples to verify the efficiency and accuracy of our methods. We conclude the paper in Section 6.

2. Motivation in level set methods

As mentioned in the Introduction, our study of the numerical computations of the delta function integrals (1.1) is motivated by the recent efficient level set methods for computing the multivalued solutions in the semiclassical limit of the linear Schrödinger equation and the high frequency limit of the linear wave equation [13,14], see also the related work [16–18]. Computation of multivalued solutions in nonlinear PDEs has been a very active area of research, for example [1,5,9,6,7,15,21,37], to mention a few.

In the high frequency limit of WKB ansatz for Schrödinger or wave equation, one has to solve the Liouville equation

$$f_t + H_{\mathbf{v}} \cdot \nabla_{\mathbf{x}} f - H_{\mathbf{x}} \cdot \nabla_{\mathbf{v}} f = 0, \quad t > 0, \quad \mathbf{x}, \mathbf{v} \in R^d, \tag{2.6}$$

with measure-valued initial data

$$f(\mathbf{x}, \mathbf{v}, 0) = \rho_0(\mathbf{x})\delta(\mathbf{v} - \mathbf{u}_0(\mathbf{x})), \tag{2.7}$$

see for example [25,10,20], where $f(t, \mathbf{x}, \mathbf{v})$ is the particle density function depending on position $\mathbf{x}$, time t and the velocity or slowness vector $\mathbf{v}$, $H(\mathbf{x}, \mathbf{v})$ is the Hamiltonian depending on specific problem. For semiclassical limit of the linear Schrödinger equation H takes the form

$$H(\mathbf{x}, \mathbf{v}) = \frac{1}{2}|\mathbf{v}|^2 + V(\mathbf{x}), \tag{2.8}$$

where $V(\mathbf{x})$ is the potential, $\mathbf{v}$ is the particle velocity, and for geometric optics limit of the linear wave equation

$$H(\mathbf{x}, \mathbf{v}) = c(\mathbf{x})|\mathbf{v}| = c(\mathbf{x})\sqrt{v_1^2 + v_2^2 + \ldots + v_d^2},$$

where $c(\mathbf{x})$ is the local wave speed, $\mathbf{v}$ is the slowness vector.

The solution of (2.6) and (2.7) at later time remains measure-valued (with finite or even infinite number of concentrations-corresponding to multivalued solutions in the physical space). Numerical methods by directly computing such measure-valued solutions could easily suffer from poor resolution due to the numerical approximation of the initial data as well as numerical dissipation. The level set method proposed in [13,14] obtains the equivalent form of f which is $\phi\prod_{i=1}^{d}\delta(\psi_i)$, where ϕ and ψ_i $(i = 1, \ldots, d)$ solve the same Liouville Eq. (2.6) with initial data

$$\phi(\mathbf{x}, \mathbf{v}, 0) = \rho_0(\mathbf{x}), \quad \psi_i(\mathbf{x}, \mathbf{v}, 0) = v_i - u_{i0}(\mathbf{x}), \tag{2.9}$$

respectively. (The common zeroes of ψ_i give the multivalued velocity or slowness vector, see [28,12,14]). Thus the computation of measure-valued f can be decomposed into the computations of bounded valued level set functions satisfying the same Liouville equation, which are much easier to compute with high accuracy. The moments can be recovered through

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t)\mathrm{d}\mathbf{v} = \int \phi(\mathbf{x}, \mathbf{v}, t)\prod_{i=1}^{d}\delta(\psi_i)\mathrm{d}\mathbf{v}, \tag{2.10}$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho(\mathbf{x}, t)}\int f(\mathbf{x}, \mathbf{v}, t)\mathbf{v}\mathrm{d}\mathbf{v} = \int \phi(\mathbf{x}, \mathbf{v}, t)\mathbf{v}\prod_{i=1}^{d}\delta(\psi_i)\mathrm{d}\mathbf{v}/\rho(\mathbf{x}, t). \tag{2.11}$$

Thus to implement the above decomposition technique requires efficient numerical methods for evaluating delta function integrals (2.10) and (2.11) provided the values of ϕ, ψ_i at grid points of a uniform mesh. These integrals belong to the type (1.1). This gives motivation to our study in this paper.

3. Nonconvergence of numerical quadratures to the two dimensional delta function integrals

In this Section we show that in two dimension the numerical quadrature approach to the delta function integrals (1.1), although simple to implement, suffers from nonconvergence when using usual discrete delta functions and currently available support size formulas. Assume the integral domain is covered by a uniform mesh (x_i, y_j) with mesh sizes $\Delta x = \Delta y = h$, and the values of the weight and level set functions are given at the mesh points. We consider the following two dimensional numerical quadratures to (1.1) which have been used in [13,16]

$$\sum_i \sum_j \alpha(x_i, y_j)\delta_{w^1_{ij}}(\beta_1(x_i, y_j))\delta_{w^2_{ij}}(\beta_2(x_i, y_j))h^2, \tag{3.12}$$

where $\delta_{w^k_{ij}}, k = 1, 2$ are discrete delta functions such as (1.3)–(1.5) and w^k_{ij} are their support sizes.

A key issue for using the numerical quadrature (3.12) is how to choose support sizes $w^k_{i,j}, k = 1, 2$ properly to ensure the convergence of the method. In [13,16] the following natural support size formulas are used

$$w^1_{ij} = w^2_{ij} = \|\partial\Psi/\partial(x, y)|_{x=x_i, y=y_j}|h \tag{3.13}$$

and

$$w^1_{ij} = w^2_{ij} = \max(\|\partial\Psi/\partial(x, y)|_{x=x_i, y=y_j}|, 1)h, \tag{3.14}$$

where $\Psi = (\beta_1, \beta_2)$, and the Jacobian is approximated by the central differences in computations. Another possible support size formula following the principle proposed in [8] takes the form

$$w^1_{ij} = \left(\left|\frac{\partial\beta_1}{\partial x}\right| + \left|\frac{\partial\beta_1}{\partial y}\right|\right)h\bigg|_{x=x_i, y=y_j}, \quad w^2_{ij} = \left(\left|\frac{\partial\beta_2}{\partial x}\right| + \left|\frac{\partial\beta_2}{\partial y}\right|\right)h\bigg|_{x=x_i, y=y_j}. \tag{3.15}$$

The derivatives in (3.15) again are approximated by central differences.

However, we will show in the following the numerical quadratures (3.12) using the discrete delta functions $\delta^L_w(x)$, $\delta^C_w(x)$, $\delta^{PC}_w(x)$ in (1.3)–(1.5) with above support size formulas suffer from nonconvergence.

We consider the computation of the following delta function integral

$$\int_{\mathbb{R}^2} \delta\left(\frac{\sqrt{2}x + y}{\sqrt{3}}\right)\delta\left(\frac{\sqrt{2}y - x}{\sqrt{3}}\right)\mathrm{d}x\mathrm{d}y. \tag{3.16}$$

For this problem, the support size formulas (3.13) and (3.14) become $w^1_{ij} = w^2_{ij} = h$, and (3.15) is $w^1_{ij} = w^2_{ij} = \frac{\sqrt{2}+1}{\sqrt{3}}h$. So we test a class of support size formulas $w^1_{ij} = w^2_{ij} = \gamma h, \gamma \in \mathbb{R}^+$. We choose the mesh $x_i = ih, y_j = jh, i, j \in \mathbb{Z}$. Then the numerical quadrature (3.12) to the integral (3.16) is

$$J = \sum_{i\in\mathbb{Z}} \sum_{j\in\mathbb{Z}} \delta_{\gamma h}\left(\frac{\sqrt{2}x_i + y_j}{\sqrt{3}}\right)\delta_{\gamma h}\left(\frac{\sqrt{2}y_j - x_i}{\sqrt{3}}\right)h^2. \tag{3.17}$$

We assume the discrete delta function in (3.17) holds the property $\delta_w(x) = \frac{1}{h}\delta_{\frac{w}{h}}(\frac{x}{h})$, which are true for $\delta^L_w(x)$, $\delta^C_w(x)$, $\delta^{PC}_w(x)$. Then the quadrature (3.17) becomes

$$J(\gamma) = \sum_{i\in\mathbb{Z}} \sum_{j\in\mathbb{Z}} \delta_\gamma\left(\frac{\sqrt{2}i + j}{\sqrt{3}}\right)\delta_\gamma\left(\frac{\sqrt{2}j - i}{\sqrt{3}}\right), \tag{3.18}$$

which is independent of h. Thus the numerical quadrature (3.17) is convergent only if (3.18) gives the exact value of the integral (3.16) which is 1. The support size formulas (3.13) and (3.14) correspond to $\gamma = 1$, and (3.15) corresponds to $\gamma = \frac{\sqrt{2}+1}{\sqrt{3}}$. Thus we will test whether the quadrature (3.18) gives the value 1 under the choices $\gamma = 1, \frac{\sqrt{2}+1}{\sqrt{3}}$.

Fig. 1 depicts the curve $\log_{10}(J(\gamma)-1)$ versus γ in the range $1 \leqslant \gamma \leqslant 100$ computed numerically when choosing the discrete delta function in (3.18) as $\delta_w^L(x)$. The results show that in such choice, $J(\gamma) > 1$ for all the tested γ values including $\gamma = 1, \frac{\sqrt{2}+1}{\sqrt{3}}$.

As a comparison, Fig. 2 depicts the curve

$$\mathrm{sign}(J(\gamma)-1)\log_{10}(|J(\gamma)-1|) \equiv V(\gamma)$$

versus γ in the range $1 \leqslant \gamma \leqslant 10$ computed numerically when choosing the discrete delta function in (3.18) as $\delta_w^C(x)$. Since the quantity $J(\gamma)$ is close to 1, $\log_{10}(|J(\gamma)-1|) < 0$, thus $V(\gamma) > 0$ when $J(\gamma) < 1$ and $V(\gamma) < 0$ when $J(\gamma) > 1$. The discontinuity points of $V(\gamma)$ represents the positions where $J(\gamma) = 1$. The results show that in this situation, there are countable number of γ values sorted in ascending order can ensure the quadrature (3.18) being 1, but $\gamma = 1, \frac{\sqrt{2}+1}{\sqrt{3}}$ still lead to non-convergent solutions. Moreover, although numerical tests show that when the discrete delta function $\delta_w^C(x)$ is used, there generally exist support size choices ensuring the convergence of the numerical quadrature (3.12), it is not clear yet how to express these support size conditions

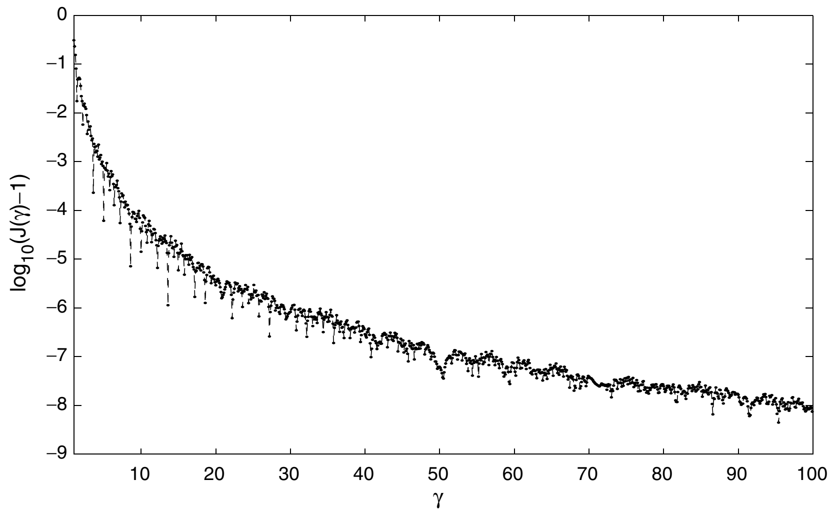

Fig. 1. Curve $\log_{10}(J(\gamma)-1)$ computed numerically in the range $1 \leqslant \gamma \leqslant 100$, using the discrete delta function $\delta_w^L(x)$.

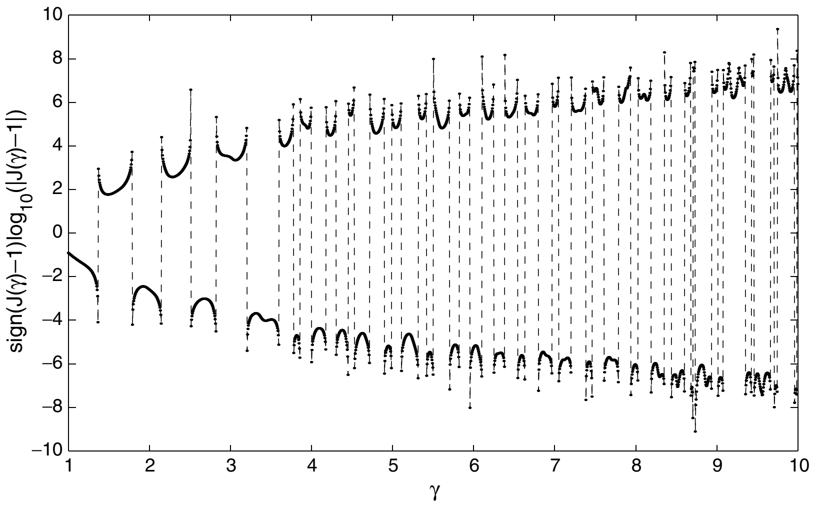

Fig. 2. Curve $\mathrm{sign}(J(\gamma)-1)\log_{10}(|J(\gamma)-1|)$ computed numerically in the range $1 \leqslant \gamma \leqslant 10$, using the discrete delta function $\delta_w^C(x)$.

explicitly by the derivatives of the level set functions as (3.13)–(3.15) do. Without an explicit support size formula, the numerical quadrature (3.12) cannot be used in practical computations since one does not actually know how to choose support sizes ensuring the convergence of the numerical method.

Using the discrete delta function $\delta_w^{PC}(x)$ in the numerical quadrature (3.18) gives similar phenomenon to that using $\delta_w^L(x)$, which is that the numerical quadrature (3.18) does not give value 1 for all the tested γ values including $\gamma = 1, \frac{\sqrt{2}+1}{\sqrt{3}}$.

4. Efficient numerical methods to the delta function integrals

4.1. One dimension

In one dimensional case the delta function integrals (1.1) take the form

$$\int_a^b \alpha(x)\delta(\beta(x))\,dx, \tag{4.19}$$

where $\beta(x)$ may have finitely many zero points on the integral interval.

Instead of using discrete delta functions to compose numerical quadrature, another natural way to evaluate the delta function integral is to use interpolation to directly approximate its exact value.

Assume the zero points of $\beta(x)$ are $x_c^i, i = 1, \ldots, K$, and $\beta^{(1)}(x_c^i) \neq 0, i = 1, \ldots, K$, where $\beta^{(1)}$ is the derivative of β. Then the exact value of (4.19) is

$$\sum_{i=1}^{K} \frac{\alpha(x_c^i)}{|\beta^{(1)}(x_c^i)|}. \tag{4.20}$$

A natural way to evaluate this value is as follows

Algorithm I*

- for each cell, check whether this cell contains a zero point of $\beta(x)$.
- if it is true, then
  - get an approximate zero point position, denoted by c^*.
  - use difference approximation to evaluate $\beta^{(1)}(x)$ at grid points, so that approximate grid points values of $\frac{\alpha(x)}{|\beta^{(1)}(x)|}$ are available.
  - use interpolation to approximate $\frac{\alpha(c^*)}{|\beta^{(1)}(c^*)|}$.
- add up the results of all the cells containing a zero point of $\beta(x)$.

In the first step in Algorithm I*, one needs to check the existence of the zero point of the level set function. This is trivial in one dimensional case. One just checks the signs of the level set function at two endpoints of the cell. However this is not obvious in high dimensions. In the later parts of this paper we will discuss convenient strategy to determine the common zero points of the level set functions in high dimensions.

In the second step in Algorithm I*, one may approximate $\beta(x)$ by a polynomial using interpolation, then solve the root of this polynomial as the approximate zero point position c^*. However, if high order accuracy is required, namely the polynomial is high order, then one has to solve a nonlinear algebraic equation. In high dimensions, in order to achieve high order accuracy, this approach even requires solving a system of nonlinear algebraic equations. This is certainly inconvenient. In this paper we propose a technique to adapt Algorithm I* so that it does not require determining the zero point position and solving the nonlinear algebraic equations when achieving high order accuracy. We will extend our one dimensional algorithm to two and three dimensional cases in the later parts of this paper.

Let x_c be a zero point of $\beta(x)$. Our idea is to interpolate $\frac{\alpha(x_c)}{|\beta^{(1)}(x_c)|}$ in the level set function variable space rather than in the integral variable space. Since $\beta^{(1)}(x_c) \neq 0$, $\beta(x)$ has the inverse function at the neighborhood of x_c, denoted by $x = \gamma(\beta)$. One has $\gamma(0) = x_c$. Denote $F(y) = \frac{\alpha(\gamma(y))}{|\beta_D^{(1)}(\gamma(y))|}$, where $|\beta_D^{(1)}(x)|$ denotes the difference approximation to $|\beta^{(1)}(x)|$. Then $F(0)=\frac{\alpha(x_c)}{|\beta_D^{(1)}(x_c)|}$. Assume x_c is contained in a cell $[x_i,x_{i+1}]$, and denote $\beta_j=\beta(x_j)$ for all j. Then $F(\beta_j)=\frac{\alpha(x_j)}{|\beta_D^{(1)}(x_j)|}$, for $j=i,i+1,i-1,i+2$ and so on. $\{\beta_j\}$ typically consist a nonuniform mesh near 0. Thus the problem becomes to interpolate $F(0)$ provided the values of $F(y)$ at the grid points of a nonuniform mesh around 0. From this viewpoint, one does not need to determine the zero point position x_c in the integral variable space, and avoids solving a nonlinear algebraic equation.

To interpolate $F(0)$ from the values $F(\beta_j)$, we approximate the function $F(y)$ by a polynomial $p_0+p_1y+\dots$ which has the same values as $F(y)$ at grid points β_j for some j, then the coefficients $p_0,p_1,\dots$ can be determined, and p_0 is the interpolation value to $F(0)$. For example, to achieve second order accuracy, one chooses the first order polynomial p_0+p_1y which has the same values as $F(y)$ at β_i,β_{i+1}, then the coefficients satisfy the linear algebraic equations

$$\begin{pmatrix}1 & \beta_i\\ 1 & \beta_{i+1}\end{pmatrix}\begin{pmatrix}p_0\\ p_1\end{pmatrix}=\begin{pmatrix}F(\beta_i)\\ F(\beta_{i+1})\end{pmatrix}. \tag{4.21}$$

After solving this linear algebraic equations system, one obtains the interpolation value p_0. Since $\beta(x)$ is reversible near the zero point, $x_i\neq x_{i+1}$ implies $\beta_i\neq\beta_{i+1}$ for a fine enough mesh. Thus the matrix in (4.21) is of Vandermonde-type and nonsingular. This ensures that our approach is always implementable.

High order methods can be similarly designed. For example, to achieve fourth order accuracy, one solves a fourth order linear algebraic equations system similar to (4.21).

One can also use Newton interpolation to approximate $F(0)$. This approach does not need to solve linear algebraic equations. But it is not extendible to high dimensional cases. The approach we used above via solving linear algebraic equations has the advantage that it can be systematically extended to high dimensional cases. On the other hand, since the level set function only has finitely many zero points, the extra computational complexity of the operations in a cell containing a zero point is relatively insignificant. This consideration is also proper in high dimensional cases, in which the level set functions only have finitely many common zero points.

In the last step in Algorithm I*, directly add up the results of all the cells containing a zero point of $\beta(x)$ can cause problems in the situation that a zero point of $\beta(x)$ is located at a grid point. For example, if $\beta(x_i)=0$, then the cells $[x_{i-1},x_i]$ and $[x_i,x_{i+1}]$ both contain x_i, and the approximation to $\frac{\alpha(x_i)}{|\beta^{(1)}(x_i)|}$ are added up twice. To resolve this issue, we propose to introduce an indication function $s(x)$ defined at grid points. Initially we set $s(x_j)=0$ for all j. If a cell $[x_i,x_{i+1}]$ contains a zero point of $\beta(x)$, and the result in the cell is added, then we set $s(x_i)=s(x_{i+1})=1$. For any cell $[x_j,x_{j+1}]$, we first check whether $s(x_j)=s(x_{j+1})=0$. If not, then the cell is directly regarded as not containing a zero point of $\beta(x)$ and no further operations are needed in this cell. This approach uses the fact that any two zero points of $\beta(x)$ have $O(1)$ distance. So if a cell contains a zero point of $\beta(x)$, then its neighboring cells can not contain another zero point. By this approach we avoid adding the result of the same zero point twice. The advantage of this approach is that it can be conveniently extended to high dimensions.

After the above discussions, the algorithm we adapt from Algorithm I* is described as follows

Algorithm I

```
• set $s(x_j)=0$ for all grid points.
• for each cell$[x_i,x_{i+1}]$, check whether $s(x_i)=s(x_{i+1})=0$.
  – if it is true, check whether this cell contains a zero point of $\beta(x)$.
    * if it is true, then
      · use difference approximation to evaluate $\beta^{(1)}(x)$ at grid points, so that approximate grid points
        values of $\frac{\alpha(x)}{|\beta^{(1)}(x)|}$ are available. These values are used as right hand sides of the linear algebraic
        equations of (4.21) type.
      · solve a linear algebraic equations system of (4.21) type according to the expected order accuracy,
        and get the value of $p_0$.
      · set $s(x_i)=s(x_{i+1})=1$.
• add up the values of $p_0$ of all the cells containing a zero point of $\beta(x)$.
```

In Algorithm I, to approximate $\beta^{(1)}(x)$ at a grid point by the grid point values of $\beta(x)$, we use a fourth order difference formula as follows

$$\beta^{(1)}(x_i) \approx \frac{1}{h}\left(-\frac{1}{12}\beta_{i+2} + \frac{2}{3}\beta_{i+1} - \frac{2}{3}\beta_{i-1} + \frac{1}{12}\beta_{i-2}\right), \tag{4.22}$$

since we will study up to fourth order method in this paper. This formula will also be used in high dimensional cases.

4.2. Two dimension

In two dimensional case the delta function integrals (1.1) take the form

$$\int_a^b \int_c^d \alpha(\mathbf{x})\delta(\beta_1(\mathbf{x}))\delta(\beta_2(\mathbf{x}))\mathrm{d}\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^2. \tag{4.23}$$

Assume the common zero points of $\beta_1(x,y)$ and $\beta_2(x,y)$ are $(x_c^i, y_c^i), i = 1,\dots,K$, and $|\partial\Psi/\partial(x,y)|_{x=x_c^i,y=y_c^i} \neq 0, i = 1,\dots,K$, where $\Psi = (\beta_1,\beta_2)$, $|\partial\Psi/\partial(x,y)|$ is the Jacobian. Then the exact value of (4.23) is

$$\sum_{i=1}^{K} \frac{\alpha(x_c^i,y_c^i)}{||\partial\Psi/\partial(x,y)|_{x=x_c^i,y=y_c^i}|}.$$

Compared with one dimensional case, in two dimension an additional issue needs to be studied is how to check the existence of a common zero point of the level set functions. As stated before, in one dimension to check the existence of a zero point of the level set function is trivial. But this is not obvious in high dimensions.

Let us consider a common zero point (x_c,y_c) of the level set functions $\beta_1(x,y)$ and $\beta_2(x,y)$. Since $|\partial\Psi/\partial(x,y)|_{x=x_c,y=y_c} \neq 0$, the map $\Gamma : (x,y) \to (\beta_1(x,y),\beta_2(x,y))$ is one-to-one at the neighborhood of the common zero point. Assume the common zero point (x_c,y_c) is in or near a cell

$$C_{ij} : [x_i,x_{i+1}] \times [y_j,y_{j+1}] \tag{4.24}$$

in (x,y)-space, consider the set

$$\widetilde{C}_{ij} : \{(p,q)|\exists(x,y) \in C_{ij}, \text{s.t.}(p,q) = \Gamma(x,y)\}. \tag{4.25}$$

(4.25) typically is a set with curved boundary. To check that the cell C_{ij} contains a common zero point of the level set functions is equivalent to check that the set $\widetilde{C}_{ij}$ contains the point $(0,0)$. However this is inconvenient since $\widetilde{C}_{ij}$ has curved boundary. This difficulty inspire us to consider the quadrilateral modified from $\widetilde{C}_{ij}$ which has flat boundary. Denote $\beta_{1,k,l} = \beta_1(x_k,y_l), \beta_{2,k,l} = \beta_2(x_k,y_l)$ for all k, l. Let the level set functions values of the four vertexes of the cell C_{ij} be: $P_1 : (\beta_{1,i,j},\beta_{2,i,j})$, $P_2 : (\beta_{1,i+1,j},\beta_{2,i+1,j})$, $P_3 : (\beta_{1,i+1,j+1},\beta_{2,i+1,j+1})$, $P_4 : (\beta_{1,i,j+1},\beta_{2,i,j+1})$. We consider the quadrilateral

$$\widehat{C}_{ij} : \text{the quadrilateral enclosed by the line segments } \overline{P_1P_2}, \overline{P_2P_3}, \overline{P_3P_4}, \overline{P_4P_1}. \tag{4.26}$$

Since Γ is the linear map plus high order terms near the common zero point (x_c,y_c), $\widehat{C}_{ij}$ is approximately a parallelogram and is convex for fine enough mesh. If the common zero point (x_c,y_c) is located in a cell C_{ij}, then for this cell and its neighboring cells, the corresponding $\widehat{C}_{ij}$ compose an irregular mesh covering the point $(0,0)$ for fine enough mesh. Thus our strategy is for each cell C_{ij} to check whether the point $(0,0)$ is contained by the quadrilateral $\widehat{C}_{ij}$ rather than to check the set $\widetilde{C}_{ij}$. We will use the indication function as adopted in one dimensional case. Namely if a quadrilateral $\widehat{C}_{ij}$ is checked to contain a common zero point, then its neighboring quadrilaterals are directly regarded to not contain a common zero point. This avoids to add the result of one common zero point more than once if the zero point is located at the boundary of $\widehat{C}_{ij}$.

One convenient way to check whether the point $(0,0)$ is covered by a convex quadrilateral $\widehat{C}_{ij}$ is to compare the position of $(0,0)$ towards the oriented line segments $\overrightarrow{P_1P_2}$, $\overrightarrow{P_2P_3}$, $\overrightarrow{P_3P_4}$, $\overrightarrow{P_4P_1}$. If $(0,0)$ is at the same side of the four oriented line segments, then it is enclosed by the four line segments, thus being covered by the quadrilateral. Algebraically, this can be checked by observing the following four determinants:

$$\begin{vmatrix} \beta_{1,i,j} & \beta_{1,i+1,j} \\ \beta_{2,i,j} & \beta_{2,i+1,j} \end{vmatrix}, \begin{vmatrix} \beta_{1,i+1,j} & \beta_{1,i+1,j+1} \\ \beta_{2,i+1,j} & \beta_{2,i+1,j+1} \end{vmatrix}, \begin{vmatrix} \beta_{1,i+1,j+1} & \beta_{1,i,j+1} \\ \beta_{2,i+1,j+1} & \beta_{2,i,j+1} \end{vmatrix}, \begin{vmatrix} \beta_{1,i,j+1} & \beta_{1,i,j} \\ \beta_{2,i,j+1} & \beta_{2,i,j} \end{vmatrix}. \tag{4.27}$$

If these four determinants have the same sign, including some of them being zero, then the point $(0,0)$ is covered by the quadrilateral $\widehat{C}_{ij}$, otherwise not.

If we have checked that the point $(0,0)$ is contained by a quadrilateral $\widehat{C}_{ij}$, implying the zero point (x_c, y_c) being in or near the cell C_{ij}, then we can use interpolation to approximate $\frac{\alpha(x_c,y_c)}{\|\partial\Psi/\partial(x,y)|_{x=x_c,y=y_c}|}$. First we choose a point among the four points (x_i,y_j), (x_{i+1},y_j), (x_{i+1},y_{j+1}), (x_i,y_{j+1}). From the viewpoint of numerical accuracy, we choose the point denoted by (x_k,y_l) which has the minimum value of $\beta^2_{1,k,l}+\beta^2_{2,k,l}$. Then in the same spirit as in the one dimensional case, we interpolate in the level set functions variables (β_1,β_2) space instead of in the integral variables (x,y) space. Denote

$$G_{ij} = \frac{\alpha(x_i,y_j)}{\|\partial\Psi_D/\partial(x,y)|_{x=x_i,y=y_j}|}, \quad \text{for all } i,j, \tag{4.28}$$

where $\partial\Psi_D/\partial(x,y)$ is the difference approximation to the Jacobian matrix $\partial\Psi/\partial(x,y)$, which can be achieved by using the formula (4.22). Then to achieve second order accuracy, we can solve a third order linear algebraic equations system, for example

$$\begin{pmatrix} 1 & \beta_{1,k,l} & \beta_{2,k,l} \\ 1 & \beta_{1,k+1,l} & \beta_{2,k+1,l} \\ 1 & \beta_{1,k,l+1} & \beta_{2,k,l+1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} G_{kl} \\ G_{k+1,l} \\ G_{k,l+1} \end{pmatrix}. \tag{4.29}$$

To achieve third order accuracy, we can solve a sixth order linear algebraic equations system, for example

$$\begin{pmatrix} 1 & \beta_{1,k,l} & \beta_{2,k,l} & \beta^2_{1,k,l} & \beta^2_{2,k,l} & \beta_{1,k,l}\beta_{2,k,l} \\ 1 & \beta_{1,k+1,l} & \beta_{2,k+1,l} & \beta^2_{1,k+1,l} & \beta^2_{2,k+1,l} & \beta_{1,k+1,l}\beta_{2,k+1,l} \\ 1 & \beta_{1,k,l+1} & \beta_{2,k,l+1} & \beta^2_{1,k,l+1} & \beta^2_{2,k,l+1} & \beta_{1,k,l+1}\beta_{2,k,l+1} \\ 1 & \beta_{1,k-1,l} & \beta_{2,k-1,l} & \beta^2_{1,k-1,l} & \beta^2_{2,k-1,l} & \beta_{1,k-1,l}\beta_{2,k-1,l} \\ 1 & \beta_{1,k,l-1} & \beta_{2,k,l-1} & \beta^2_{1,k,l-1} & \beta^2_{2,k,l-1} & \beta_{1,k,l-1}\beta_{2,k,l-1} \\ 1 & \beta_{1,k+1,l+1} & \beta_{2,k+1,l+1} & \beta^2_{1,k+1,l+1} & \beta^2_{2,k+1,l+1} & \beta_{1,k+1,l+1}\beta_{2,k+1,l+1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix} = \begin{pmatrix} G_{kl} \\ G_{k+1,l} \\ G_{k,l+1} \\ G_{k-1,l} \\ G_{k,l-1} \\ G_{k+1,l+1} \end{pmatrix}. \tag{4.30}$$

To achieve fourth order accuracy, we solve a tenth order linear algebraic equations system which can be constructed following the same principle as (4.29) and (4.30).

We can check that the matrixes in (4.29) and (4.30) are nonsingular for fine enough mesh. Denote

$$B_{1,x} = \frac{\partial\beta_1}{\partial x}|_{x=x_c,y=y_c}, \quad B_{1,y} = \frac{\partial\beta_1}{\partial y}|_{x=x_c,y=y_c}, \quad B_{2,x} = \frac{\partial\beta_2}{\partial x}|_{x=x_c,y=y_c}, \quad B_{2,y} = \frac{\partial\beta_2}{\partial y}|_{x=x_c,y=y_c}.$$

Denote the matrix in (4.29) to be M_1, then

$$|M_1| = \begin{vmatrix} 1 & B_{1,x}(x_k-x_c)+B_{1,y}(y_l-y_c) & B_{2,x}(x_k-x_c)+B_{2,y}(y_l-y_c) \\ 1 & B_{1,x}(x_{k+1}-x_c)+B_{1,y}(y_l-y_c) & B_{2,x}(x_{k+1}-x_c)+B_{2,y}(y_l-y_c) \\ 1 & B_{1,x}(x_k-x_c)+B_{1,y}(y_{l+1}-y_c) & B_{2,x}(x_k-x_c)+B_{2,y}(y_{l+1}-y_c) \end{vmatrix} + \text{H.O.},$$

where H.O. represents high order terms. So

$$|M_1| = \begin{vmatrix} 1 & B_{1,x}x_k+B_{1,y}y_l & B_{2,x}x_k+B_{2,y}y_l \\ 1 & B_{1,x}x_{k+1}+B_{1,y}y_l & B_{2,x}x_{k+1}+B_{2,y}y_l \\ 1 & B_{1,x}x_k+B_{1,y}y_{l+1} & B_{2,x}x_k+B_{2,y}y_{l+1} \end{vmatrix} + \text{H.O.} = \begin{vmatrix} 1 & x_k & y_l \\ 1 & x_{k+1} & y_l \\ 1 & x_k & y_{l+1} \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & B_{1,x} & B_{2,x} \\ 0 & B_{1,y} & B_{2,y} \end{vmatrix} + \text{H.O.} \tag{4.31}$$

The first matrix in (4.31) can be regarded as the matrix in (x, y) space corresponding to M_1. Since the second determinant in (4.31) is $|\partial\Psi/\partial(x, y)|_{x=x_c,y=y_c} \neq 0$, (4.31) implies that M_1 is nonsingular for fine enough mesh if its corresponding matrix in (x, y) space is nonsingular. This is clearly valid, thus M_1 is nonsingular for fine enough mesh.

Denote the matrix in (4.30) to be M_2, denote

$$B^1_{ij} = B_{1,x}x_i + B_{1,y}y_j, \quad B^2_{ij} = B_{2,x}x_i + B_{2,y}y_j, \quad \text{for all } i, j.$$

Then

$$|M_2| = \begin{vmatrix} 1 & B^1_{kl} & B^2_{kl} & (B^1_{kl})^2 & (B^2_{kl})^2 & B^1_{kl}B^2_{kl} \\ 1 & B^1_{k+1,l} & B^2_{k+1,l} & (B^1_{k+1,l})^2 & (B^2_{k+1,l})^2 & B^1_{k+1,l}B^2_{k+1,l} \\ 1 & B^1_{k,l+1} & B^2_{k,l+1} & (B^1_{k,l+1})^2 & (B^2_{k,l+1})^2 & B^1_{k,l+1}B^2_{k,l+1} \\ 1 & B^1_{k-1,l} & B^2_{k-1,l} & (B^1_{k-1,l})^2 & (B^2_{k-1,l})^2 & B^1_{k-1,l}B^2_{k-1,l} \\ 1 & B^1_{k,l-1} & B^2_{k,l-1} & (B^1_{k,l-1})^2 & (B^2_{k,l-1})^2 & B^1_{k,l-1}B^2_{k,l-1} \\ 1 & B^1_{k+1,l+1} & B^2_{k+1,l+1} & (B^1_{k+1,l+1})^2 & (B^2_{k+1,l+1})^2 & B^1_{k+1,l+1}B^2_{k+1,l+1} \end{vmatrix} + \text{H.O.} = |\widetilde{M}_2||\widetilde{B}| + \text{H.O.}, \tag{4.32}$$

where

$$|\widetilde{M}_2| = \begin{vmatrix} 1 & x_k & y_l & (x_k)^2 & (y_l)^2 & x_k y_l \\ 1 & x_{k+1} & y_l & (x_{k+1})^2 & (y_l)^2 & x_{k+1}y_l \\ 1 & x_k & y_{l+1} & (x_k)^2 & (y_{l+1})^2 & x_k y_{l+1} \\ 1 & x_{k-1} & y_l & (x_{k-1})^2 & (y_l)^2 & x_{k-1}y_l \\ 1 & x_k & y_{l-1} & (x_k)^2 & (y_{l-1})^2 & x_k y_{l-1} \\ 1 & x_{k+1} & y_{l+1} & (x_{k+1})^2 & (y_{l+1})^2 & x_{k+1}y_{l+1} \end{vmatrix}, \tag{4.33}$$

$$|\widetilde{B}| = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & B_{1,x} & B_{2,x} & 0 & 0 & 0 \\ 0 & B_{1,y} & B_{2,y} & 0 & 0 & 0 \\ 0 & 0 & 0 & (B_{1,x})^2 & (B_{2,x})^2 & B_{1,x}B_{2,x} \\ 0 & 0 & 0 & (B_{1,y})^2 & (B_{2,y})^2 & B_{1,y}B_{2,y} \\ 0 & 0 & 0 & 2B_{1,x}B_{1,y} & 2B_{2,x}B_{2,y} & B_{1,x}B_{2,y} + B_{2,x}B_{1,y} \end{vmatrix}. \tag{4.34}$$

The matrix in (4.33) is the corresponding matrix to M_2 in (x, y) space. Since $|\widetilde{B}|$ in (4.34) equals $(|\partial\Psi/\partial(x, y)|_{x=x_c,y=y_c})^4 \neq 0$, (4.32) again implies that M_2 is nonsingular for fine enough mesh if its corresponding matrix in (x, y) space is nonsingular. One can check that $\widetilde{M}_2$ is nonsingular, thus M_2 is nonsingular for fine enough mesh.

Similarly, one can check that for the tenth order matrix in achieving fourth order accuracy, it also holds that the matrix is nonsingular for fine enough mesh if its corresponding matrix in (x, y) space is nonsingular. To satisfy this, one can choose the ten indices in the matrix for example to be the eight indices among $\{(i, j)|i = k-1, k, k+1, j = l-1, l, l+1\}$ except $(k-1, l-1)$, plus two indices $(k+2, l), (k+2, l+2)$.

Thus we show that our second to fourth order accurate methods are all ensured to be implementable. After solving the linear algebraic system, which is (4.29) for second order method, (4.30) for third order method, and a tenth order system for fourth order method, the resulting p_0 is the approximation to $\frac{\alpha(x_c,y_c)}{\|\partial\Psi/\partial(x,y)|_{x=x_c,y=y_c}|}$.

After the above discussions, our two dimensional algorithm can be described as follows

Algorithm II

```
• set $s_{ij} = 0$ for all indices.
• for each cell $C_{ij} : [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, check whether $s_{ij} + s_{i+1,j} + s_{i+1,j+1} + s_{i,j+1} = 0$.
  – if it is true, check whether the quadrilateral $\widehat{C}_{ij}$ contains the point $(0,0)$.
    * if it is true, then
      · use difference approximation to evaluate the quantities (4.28) at grid points. These values are
        used as right hand sides of the linear algebraic equations to be solved.
      · solve a linear algebraic equations system according to the expected order accuracy, and get the
        value of $p_0$.
      · set $s_{ij} = s_{i+1,j} = s_{i+1,j+1} = s_{i,j+1} = 1$.
• add up the values of $p_0$ of all the quadrilaterals $\widehat{C}_{ij}$ containing the point $(0,0)$.
```

4.3. Three dimension

In three dimensional case the delta function integrals (1.1) take the form

$$\int_a^b \int_c^d \int_e^f \alpha(\mathbf{x})\delta(\beta_1(\mathbf{x}))\delta(\beta_2(\mathbf{x}))\delta(\beta_3(\mathbf{x}))\,\mathrm{d}\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^3. \tag{4.35}$$

Assume the common zero points of $\beta_1(x,y,z)$, $\beta_2(x,y,z)$ and $\beta_3(x,y,z)$ are $(x_c^i, y_c^i, z_c^i), i = 1,\ldots,K$, and $|\partial\Psi/\partial(x,y,z)|_{x=x_c^i,y=y_c^i,z=z_c^i} \neq 0, i = 1,\ldots,K$, where $\Psi = (\beta_1,\beta_2,\beta_3)$, $|\partial\Psi/\partial(x,y,z)|$ is the Jacobian. Then the exact value of (4.35) is

$$\sum_{i=1}^{K} \frac{\alpha(x_c^i, y_c^i, z_c^i)}{\|\partial\Psi/\partial(x,y,z)|_{x=x_c^i,y=y_c^i,z=z_c^i}|}.$$

Denote $\beta_{1,i,j,k} = \beta_1(x_i,y_j,z_k), \beta_{2,i,j,k} = \beta_2(x_i,y_j,z_k), \beta_{3,i,j,k} = \beta_3(x_i,y_j,z_k)$, $P_{i,j,k} = (\beta_{1,i,j,k}, \beta_{2,i,j,k}, \beta_{3,i,j,k})$ for all i,j,k. Let the level set functions values of the eight vertexes of the cell

$$C_{ijk} : [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}] \tag{4.36}$$

be: $P_1 : P_{i,j,k}, P_2 : P_{i+1,j,k}, P_3 : P_{i+1,j+1,k}, P_4 : P_{i,j+1,k}, P_5 : P_{i,j,k+1}, P_6 : P_{i+1,j,k+1}, P_7 : P_{i+1,j+1,k+1}, P_8 : P_{i,j+1,k+1}$. In the same spirit as in the two dimensional case, we consider the dodecahedron

$$\widehat{C}_{ijk} : \text{the dodecahedron enclosed by the faces}$$
$$\overline{P_1P_2P_4}, \overline{P_1P_5P_2}, \overline{P_1P_4P_5}, \overline{P_2P_5P_6}, \overline{P_2P_6P_3}, \overline{P_5P_8P_6},$$
$$\overline{P_2P_3P_4}, \overline{P_3P_8P_4}, \overline{P_4P_8P_5}, \overline{P_3P_6P_7}, \overline{P_6P_8P_7}, \overline{P_3P_7P_8}. \tag{4.37}$$

If a common zero point (x_c, y_c, z_c) is located in a cell C_{ijk}, then for this cell and its neighboring cells, the corresponding $\widehat{C}_{ijk}$ compose an irregular mesh covering the point $(0,0,0)$ for fine enough mesh. To design a convenient way to check a common zero point (x_c, y_c, z_c), observe that the dodecahedron $\widehat{C}_{ijk}$ is covered by the union of the following two hexahedrons

$$\widehat{C}^1_{ijk} : \text{the hexahedron enclosed by the planes}$$
$$\overline{P_1P_2P_4}, \overline{P_1P_5P_2}, \overline{P_1P_4P_5}, \overline{P_3P_6P_7}, \overline{P_6P_8P_7}, \overline{P_3P_7P_8}, \tag{4.38}$$

$$\widehat{C}^2_{ijk} : \text{the hexahedron enclosed by the planes}$$
$$\overline{P_2P_5P_6}, \overline{P_2P_6P_3}, \overline{P_5P_8P_6}, \overline{P_2P_3P_4}, \overline{P_3P_8P_4}, \overline{P_4P_8P_5}. \tag{4.39}$$

For a cell near the zero point (x_c, y_c, z_c), the two hexahedrons $\widehat{C}^1_{ijk}$ and $\widehat{C}^2_{ijk}$ are convex for fine enough mesh. Moreover, $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$ can only be slightly larger than $\widehat{C}_{ijk}$ in the sense that for two dodecahedrons $\widehat{C}_{ijk}$ and $\widehat{C}_{i',j',k'}$ which are not adjacent, then their corresponding $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$ and $\widehat{C}^1_{i',j',k'} \cup \widehat{C}^2_{i',j',k'}$ also do not contact for fine enough mesh. Therefore for a cell C_{ijk} it is more convenient to check the existence of the point $(0,0,0)$ in the set $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$, for which we check the existence of the point $(0,0,0)$ in the two hexahedrons

$\widehat{C}^1_{ijk}$ and $\widehat{C}^2_{ijk}$, respectively. To check whether the point $(0,0,0)$ is covered by a convex hexahedron, we take the similar strategy as for a convex quadrilateral in two dimensional case. We compare the position of $(0,0,0)$ towards the oriented planes of the six faces of the hexahedron. If $(0,0,0)$ is at the same side of the six oriented planes, then it is enclosed by the six faces, thus being covered by the hexahedron. Algebraically, for $\widehat{C}^1_{ijk}$ this can be checked by observing the following six determinants:

$$\begin{vmatrix} \beta_{1,i,j,k} & \beta_{1,i+1,j,k} & \beta_{1,i,j+1,k} \\ \beta_{2,i,j,k} & \beta_{2,i+1,j,k} & \beta_{2,i,j+1,k} \\ \beta_{3,i,j,k} & \beta_{3,i+1,j,k} & \beta_{3,i,j+1,k} \end{vmatrix}, \begin{vmatrix} \beta_{1,i,j,k} & \beta_{1,i,j,k+1} & \beta_{1,i+1,j,k} \\ \beta_{2,i,j,k} & \beta_{2,i,j,k+1} & \beta_{2,i+1,j,k} \\ \beta_{3,i,j,k} & \beta_{3,i,j,k+1} & \beta_{3,i+1,j,k} \end{vmatrix},$$
$$\begin{vmatrix} \beta_{1,i,j,k} & \beta_{1,i,j+1,k} & \beta_{1,i,j,k+1} \\ \beta_{2,i,j,k} & \beta_{2,i,j+1,k} & \beta_{2,i,j,k+1} \\ \beta_{3,i,j,k} & \beta_{3,i,j+1,k} & \beta_{3,i,j,k+1} \end{vmatrix}, \begin{vmatrix} \beta_{1,i+1,j+1,k} & \beta_{1,i+1,j,k+1} & \beta_{1,i+1,j+1,k+1} \\ \beta_{2,i+1,j+1,k} & \beta_{2,i+1,j,k+1} & \beta_{2,i+1,j+1,k+1} \\ \beta_{3,i+1,j+1,k} & \beta_{3,i+1,j,k+1} & \beta_{3,i+1,j+1,k+1} \end{vmatrix}, \tag{4.40}$$
$$\begin{vmatrix} \beta_{1,i+1,j,k+1} & \beta_{1,i,j+1,k+1} & \beta_{1,i+1,j+1,k+1} \\ \beta_{2,i+1,j,k+1} & \beta_{2,i,j+1,k+1} & \beta_{2,i+1,j+1,k+1} \\ \beta_{3,i+1,j,k+1} & \beta_{3,i,j+1,k+1} & \beta_{3,i+1,j+1,k+1} \end{vmatrix}, \begin{vmatrix} \beta_{1,i+1,j+1,k} & \beta_{1,i+1,j+1,k+1} & \beta_{1,i,j+1,k+1} \\ \beta_{2,i+1,j+1,k} & \beta_{2,i+1,j+1,k+1} & \beta_{2,i,j+1,k+1} \\ \beta_{3,i+1,j+1,k} & \beta_{3,i+1,j+1,k+1} & \beta_{3,i,j+1,k+1} \end{vmatrix}.$$

If these six determinants have the same sign, including some of them being zero, then the point $(0,0,0)$ is covered by the hexahedron $\widehat{C}^1_{ijk}$, otherwise not. Similarly, to check the existence of the point $(0,0,0)$ in $\widehat{C}^2_{ijk}$ we compare the signs of the following six determinants:

$$\begin{vmatrix} \beta_{1,i+1,j,k} & \beta_{1,i,j,k+1} & \beta_{1,i+1,j,k+1} \\ \beta_{2,i+1,j,k} & \beta_{2,i,j,k+1} & \beta_{2,i+1,j,k+1} \\ \beta_{3,i+1,j,k} & \beta_{3,i,j,k+1} & \beta_{3,i+1,j,k+1} \end{vmatrix}, \begin{vmatrix} \beta_{1,i+1,j,k} & \beta_{1,i+1,j,k+1} & \beta_{1,i+1,j+1,k} \\ \beta_{2,i+1,j,k} & \beta_{2,i+1,j,k+1} & \beta_{2,i+1,j+1,k} \\ \beta_{3,i+1,j,k} & \beta_{3,i+1,j,k+1} & \beta_{3,i+1,j+1,k} \end{vmatrix},$$
$$\begin{vmatrix} \beta_{1,i,j,k+1} & \beta_{1,i,j+1,k+1} & \beta_{1,i+1,j,k+1} \\ \beta_{2,i,j,k+1} & \beta_{2,i,j+1,k+1} & \beta_{2,i+1,j,k+1} \\ \beta_{3,i,j,k+1} & \beta_{3,i,j+1,k+1} & \beta_{3,i+1,j,k+1} \end{vmatrix}, \begin{vmatrix} \beta_{1,i+1,j,k} & \beta_{1,i+1,j+1,k} & \beta_{1,i,j+1,k} \\ \beta_{2,i+1,j,k} & \beta_{2,i+1,j+1,k} & \beta_{2,i,j+1,k} \\ \beta_{3,i+1,j,k} & \beta_{3,i+1,j+1,k} & \beta_{3,i,j+1,k} \end{vmatrix}, \tag{4.41}$$
$$\begin{vmatrix} \beta_{1,i+1,j+1,k} & \beta_{1,i,j+1,k+1} & \beta_{1,i,j+1,k} \\ \beta_{2,i+1,j+1,k} & \beta_{2,i,j+1,k+1} & \beta_{2,i,j+1,k} \\ \beta_{3,i+1,j+1,k} & \beta_{3,i,j+1,k+1} & \beta_{3,i,j+1,k} \end{vmatrix}, \begin{vmatrix} \beta_{1,i,j+1,k} & \beta_{1,i,j+1,k+1} & \beta_{1,i,j,k+1} \\ \beta_{2,i,j+1,k} & \beta_{2,i,j+1,k+1} & \beta_{2,i,j,k+1} \\ \beta_{3,i,j+1,k} & \beta_{3,i,j+1,k+1} & \beta_{3,i,j,k+1} \end{vmatrix}.$$

By this way we can check the existence of the point $(0,0,0)$ in $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$. We introduce the indication function as adopted in one and two dimensional cases. Namely if a cell C_{ijk} whose corresponding $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$ is checked to contain the point $(0,0,0)$, then we do not check its neighboring cells. Thus the result of one common zero point will not be added more than once.

If the point $(0,0,0)$ is checked to be contained in a $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$, implying the zero point (x_c, y_c, z_c) being in or near the cell C_{ijk}, then we can use interpolation to approximate $\frac{\alpha(x_c,y_c,z_c)}{\|\partial\Psi/\partial(x,y,z)|_{x=x_c,y=y_c,z=z_c}|}$. As in the two dimensional case, we first choose a point denoted by (x_l, y_m, z_n) among the eight points (x_i, y_j, z_k), (x_{i+1}, y_j, z_k), (x_{i+1}, y_{j+1}, z_k), (x_i, y_{j+1}, z_k), (x_i, y_j, z_{k+1}), (x_{i+1}, y_j, z_{k+1}), $(x_{i+1}, y_{j+1}, z_{k+1})$, (x_i, y_{j+1}, z_{k+1}) which has the minimum value of $\beta^2_{1,l,m,n} + \beta^2_{2,l,m,n} + \beta^2_{3,l,m,n}$. Denote

$$H_{ijk} = \frac{\alpha(x_i, y_j, z_k)}{\|\partial\Psi_D/\partial(x,y,z)|_{x=x_i,y=y_j z=z_k}|}, \quad \text{for all } i,j,k, \tag{4.42}$$

where $\partial\Psi_D/\partial(x,y,z)$ is the difference approximation to the Jacobian matrix $\partial\Psi/\partial(x,y,z)$, which can be achieved by using the formula (4.22). Then to achieve second order accuracy, we can solve a fourth order linear algebraic equations system, for example

$$\begin{pmatrix} 1 & \beta_{1,l,m,n} & \beta_{2,l,m,n} & \beta_{3,l,m,n} \\ 1 & \beta_{1,l+1,m,n} & \beta_{2,l+1,m,n} & \beta_{3,l+1,m,n} \\ 1 & \beta_{1,l,m+1,n} & \beta_{2,l,m+1,n} & \beta_{3,l,m+1,n} \\ 1 & \beta_{1,l,m,n+1} & \beta_{2,l,m,n+1} & \beta_{3,l,m,n+1} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} H_{l,m,n} \\ H_{l+1,m,n} \\ H_{l,m+1,n} \\ H_{l,m,n+1} \end{pmatrix}. \tag{4.43}$$

To achieve third and fourth order accuracy, we solve a tenth and a twentieth order linear algebraic equations system, respectively which can be constructed following the same principle as (4.43).

Similar to two dimensional case, we can check that the matrix in (4.43) is nonsingular for fine enough mesh. Denote

$$B_{p,x} = \frac{\partial \beta_p}{\partial x}|_{x=x_c,y=y_c,z=z_c}, \quad B_{p,y} = \frac{\partial \beta_p}{\partial y}|_{x=x_c,y=y_c,z=z_c}, \quad B_{p,z} = \frac{\partial \beta_p}{\partial z}|_{x=x_c,y=y_c,z=z_c},$$
$$B^p_{ijk} = B_{p,x}x_i + B_{p,y}y_j + B_{p,z}z_k \text{ for all } i,j,k$$

for $p = 1,2,3$.

Denote the matrix in (4.43) to be M_3, then

$$|M_3| = \begin{vmatrix} 1 & B^1_{l,m,n} & B^2_{l,m,n} & B^3_{l,m,n} \\ 1 & B^1_{l+1,m,n} & B^2_{l+1,m,n} & B^3_{l+1,m,n} \\ 1 & B^1_{l,m+1,n} & B^2_{l,m+1,n} & B^3_{l,m+1,n} \\ 1 & B^1_{l,m,n+1} & B^2_{l,m,n+1} & B^3_{l,m,n+1} \end{vmatrix} + \text{H.O.} = \begin{vmatrix} 1 & x_l & y_m & z_n \\ 1 & x_{l+1} & y_m & z_n \\ 1 & x_l & y_{m+1} & z_n \\ 1 & x_l & y_m & z_{n+1} \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & B_{1,x} & B_{2,x} & B_{3,x} \\ 0 & B_{1,y} & B_{2,y} & B_{3,y} \\ 0 & B_{1,z} & B_{2,z} & B_{3,z} \end{vmatrix} + \text{H.O.} \tag{4.44}$$

The first matrix in (4.44) can be regarded as the matrix in (x,y,z) space corresponding to M_3. The second determinant in (4.44) is $|\partial\Psi/\partial(x,y,z)|_{x=x_c,y=y_c,z=z_c} \neq 0$. Similar to two dimensional case, we again have that M_3 is nonsingular for fine enough mesh if its corresponding matrix in (x,y,z) space is nonsingular, which is clearly valid. Thus M_3 is nonsingular for fine enough mesh.

Similarly, one can check that for the tenth and the twentieth order matrixes in achieving third and fourth order accuracy, the matrix is nonsingular for fine enough mesh if its corresponding matrix in (x,y,z) space is nonsingular, respectively. To satisfy this, one can choose the ten indices in the tenth order matrix for example to be $(l+i, m+j, n+k)$ for (i,j,k) being

$$\{(0,0,0),(1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1),(1,1,0),(1,0,1),(0,1,1)\}. \tag{4.45}$$

The twenty indices in the twentieth order matrix can be chosen for example to be $(l+i, m+j, n+k)$ for (i,j,k) being

$$\begin{aligned}\{&(0,0,0),(1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1),(1,1,0),(1,0,1),(0,1,1),(-1,-1,0),\\ &(-1,0,-1),(0,-1,-1),(1,-1,0),(1,0,-1),(0,1,-1),(1,1,1),(2,0,0),(0,2,0),(0,0,2)\}.\end{aligned} \tag{4.46}$$

Thus our second to fourth order accurate methods are all ensured to be implementable. After solving the linear algebraic system, the resulting p_0 is the approximation to $\frac{\alpha(x_c,y_c,z_c)}{\|\partial\Psi/\partial(x,y,z)|_{x=x_c,y=y_c,z=z_c}|}$.

After the above discussions, our three dimensional algorithm can be described as follows

Algorithm III

```
• set $s_{i,j,k} = 0$ for all indices.
• for each cell $C_{ijk} : [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]$, check whether $s_{i,j,k} + s_{i+1,j,k} + s_{i+1,j+1,k} + s_{i,j+1,k} + s_{i,j,k+1} + s_{i+1,j,k+1} + s_{i+1,j+1,k+1} + s_{i,j+1,k+1} = 0$.
  – if it is true, check whether the union of hexahedrons $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$ contains the point $(0,0,0)$.
    * if it is true, then
      · use difference approximation to evaluate the quantities (4.42) at grid points. These values are used as right hand sides of the linear algebraic equations to be solved.
      · solve a linear algebraic equations system according to the expected order accuracy, and get the value of $p_0$.
      · set $s_{i,j,k} = s_{i+1,j,k} = s_{i+1,j+1,k} = s_{i,j+1,k} = s_{i,j,k+1} = s_{i+1,j,k+1} = s_{i+1,j+1,k+1} = s_{i,j+1,k+1} = 1$.
• add up the values of $p_0$ of all the $\widehat{C}^1_{ijk} \cup \widehat{C}^2_{ijk}$ containing the point $(0,0,0)$.
```

5. Numerical examples

In this Section we give numerical examples in one to three dimensions respectively to show the efficiency and accuracy of our methods proposed in this paper.

Example 5.1 Consider the numerical computation of the one dimensional delta function integral (4.19). We choose $\alpha(x) = \log(x + e), \beta(x) = e^{\sqrt{2}x} - 1$. We test a number of mesh sizes. For each mesh size, we select 100 uniform meshes via random shifts. We show the largest relative numerical errors of our methods among using these 100 meshes for each mesh size.

Table 5.1 lists the results of our second to fourth order methods. The last column in the Table presents the estimated convergence rates. These results clearly show that our numerical methods achieve the expected numerical accuracy.

Example 5.2 Consider the numerical computation of the two dimensional delta function integral (4.23). We choose $\alpha(x, y) = \cos(x + y), \beta_1(x, y) = e^{5x+y} - 1, \beta_2(x, y) = e^{x+\frac{2}{3}y} - 1$. We test a number of mesh sizes. For each mesh size, we select 100 uniform meshes via random shifts in two axis directions. We show the largest relative numerical errors of our methods among using these 100 meshes for each mesh size.

Table 5.2 lists the results of our second to fourth order methods. The last column in the Table presents the estimated convergence rates. These results show that our numerical methods achieve the expected numerical accuracy.

Example 5.3 Consider the numerical computation of the three dimensional delta function integral (4.35). We choose $\alpha(x, y, z) = \cos(x + y + z)$, $\beta_1(x, y, z) = e^{\frac{1}{\sqrt{3}}x+\frac{1}{\sqrt{3}}y+\frac{1}{\sqrt{3}}z} - 1$, $\beta_2(x, y, z) = e^{\frac{1}{\sqrt{2}}x+\frac{1}{\sqrt{2}}z} - 1$, $\beta_3(x, y, z) = e^{\frac{2}{3}x+\frac{2}{3}y+\frac{1}{3}z} - 1$. We test a number of mesh sizes. For each mesh size, we select 50 uniform meshes via random shifts in three axis directions. We show the largest relative numerical errors of our methods among using these 50 meshes for each mesh size.

Table 5.3 lists the results of our second to fourth order methods. The last column in the Table presents the estimated convergence rates. These results show that our numerical methods achieve the expected numerical accuracy.

Table 5.1
Example 5.1, relative errors of the one dimensional methods

Mesh size	0.1	0.05	0.025	0.0125	0.00625	R_e
Second order method	3.21E−3	8.27E−4	2.08E−4	5.19E−5	1.29E−5	1.99
Third order method	5.06E−4	6.22E−5	7.72E−6	9.60E−7	1.20E−7	3.01
Fourth order method	6.34E−5	3.99E−6	2.50E−7	1.56E−8	9.76E−10	4.00

Table 5.2
Example 5.2, relative errors of the two dimensional methods

Mesh size	0.1	0.05	0.025	0.0125	0.00625	R_e
Second order method	1.00E−1	3.01E−2	6.96E−3	2.16E−3	5.80E−4	1.87
Third order method	1.51E−1	1.75E−2	2.18E−3	2.54E−4	3.47E−5	3.03
Fourth order method	4.04E−1	6.02E−3	4.96E−4	2.52E−5	1.16E−6	4.47

Table 5.3
Example 5.3, relative errors of the three dimensional methods

Mesh size	0.1	0.05	0.025	0.0125	0.00625	R_e
Second order method	8.03E−3	2.36E−3	6.21E−4	1.48E−4	3.43E−5	1.97
Third order method	5.13E−3	6.18E−4	1.00E−4	1.24E−5	1.56E−6	2.90
Fourth order method	4.11E−4	3.24E−5	2.46E−6	1.26E−7	3.95E−9	4.13

6. Conclusion

In this paper we studied high order numerical methods to a type of delta function integrals in one to three dimensions. Such delta function integrals arise from recent efficient level set methods for computing the multivalued solutions in the semiclassical limit of the linear Schrödinger equation and the high frequency limit of the linear wave equation [13,14].

We show that the numerical quadratures to the two dimensional delta function integrals, which are designed based on very natural idea, suffer from nonconvergence with usual discrete delta functions and support size formulas. We then proceed to design efficient numerical methods to the delta function integrals in one to three dimensions based on interpolation approach. In such an approach, one needs to check the existence of the common zero points of the level set functions, which is avoided in the quadrature approach. Such an issue indeed is nontrivial in high dimensions. In this paper we give convenient strategy to check the common zero points of the level set functions. After the check of a common zero point, the usual way to perform interpolation needs to approximate the common zero point position, which requires solving nonlinear algebraic equations system if high order accuracy is expected. We adapt the usual way by changing the interpolation space. Our approach does not need to explicitly determine the common zero points positions and avoids solving nonlinear algebraic equations system in order to achieve high order accuracy. In this paper we have designed second to fourth order methods. Numerical examples are presented which verify that our methods proposed in this paper are efficient and achieve the expected accuracy.

Acknowledgement

Research supported in part by the Knowledge Innovation Project of the Chinese Academy of Sciences Nos. K5501312S1 and K5502212F1, and NSFC Grant No. 10601062.

References

[1] J.-D. Benamou, Big ray tracing: multivalued travel time field computation using viscosity solutions of the Eikonal equation, J. Comput. Phys. 128 (1996) 463–474.
[2] R.P. Beyer, R.J. LeVeque, Analysis of a one dimensional model for the immersed boundary method, SIAM J. Num. Anal. 29 (1992) 332–364.
[3] D. Calhoun, P. Smereka, The numerical approximation of a delta function, preprint, 2004.
[4] L.-T. Cheng, Construction of shapes arising from the Minkowski problem using a level set approach, J. Sci. Comp. 19 (2003) 123–138.
[5] L.-T. Cheng, H.-L. Liu, S. Osher, Computational high frequency wave propagation using the Level Set method, with applications to the semiclassical limit of Schrödinger equations, Comm. Math. Sci. 1 (2003) 593–621.
[6] B. Engquist, O. Runborg, Computational high frequency wave propagation, Acta Numer. 12 (2003) 181–266.
[7] B. Engquist, O. Runborg, A.-K. Tornberg, High frequency wave propagation by the segment projection method, J. Comput. Phys. 178 (2) (2002) 373–390.
[8] B. Engquist, A.K. Tornberg, R. Tsai, Discretization of dirac delta functions in level set methods, J. Comput. Phys. 207 (1) (2005) 28–51.
[9] E. Fatemi, B. Engquist, S. Osher, Numerical solution of the high frequency asymptotic expansion for the scalar wave equation, J. Comput. Phys. 120 (1995) 145–155.
[10] P. Gérard, P.A. Markowich, N.J. Mauser, F. Poupaud, Homogenization limits and Wigner transforms, Comm. Pure Appl. Math. 50 (1997) 321–377.
[11] T.Y. Hou, Z.L. Li, S. Osher, H.K. Zhao, A hybrid method for moving interface problems with application to the Hele–Shaw flow, J. Comput. Phys. 134 (1997) 236–252.
[12] S. Jin, X.T. Li, Multi-phase computations of the semiclassical limit of the Schrödinger equation and related problems: Whitham vs. Wigner, Physica D 182 (2003) 46–85.
[13] S. Jin, H.L. Liu, S. Osher, R. Tsai, Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation, J. Comput. Phys. 205 (2005) 222–241.
[14] S. Jin, H.L. Liu, S. Osher, R. Tsai, Computing multivalued physical observables for high frequency limit of symmetric hyperbolic systems, J. Comput. Phys. 210 (2005) 497–518.
[15] S. Jin, S. Osher, A level set method for the computation of multivalued solutions to quasi-linear hyperbolic PDE's and Hamilton–Jacobi equations, Comm. Math. Sci. 1 (3) (2003) 575–591.
[16] S. Jin, X. Wen, Hamiltonian-preserving schemes for the Liouville equation with discontinuous potentials, Commun. Math. Sci. 3 (2005) 285–315.

[17] S. Jin, X. Wen, Hamiltonian-preserving schemes for the Liouville equation of geometrical optics with discontinuous local wave speeds, J. Comput. Phys. 214 (2006) 672–697.

[18] S. Jin, X. Yang, Computation of the semiclassical limit of the Schrodinger equation with phase shift by a level set method, J. Sci. Comp. (2007), doi:10.1007/s10915-007-9137-9.

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

[20] P.L. Lions, T. Paul, Sur les measures de Wigner, Revista. Mat. Iberoamericana 9 (1993) 553–618.

[21] S. Osher, L.-T. Cheng, M. Kang, H. Shim, Y.-H. Tsai, Geometric optics in a phase-space-based level set and Eulerian framework, J. Comput. Phys. 179 (2) (2002) 622–648.

[22] S. Osher, R.P. Fedkiw, Level set methods and dynamic implicit surfaces, Springer Verlag, 2002.

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

[24] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–511.

[25] L. Ryzhik, G. Papanicolaou, J. Keller, Transport equations for elastic and other waves in random media, Wave Motion 24 (1996) 327–370.

[26] J.A. Sethian, Level set methods and fast marching methods, Evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science, Cambridge University Press, 1999.

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

[28] C. Sparber, P. Markowich, N. Mauser, Multivalued geometrical optics: Wigner vs. WKB, Asymptotic Anal. 33 (2003) 1530187.

[29] M. Sussman, P. Smereka, S. Osher, A level set method for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.

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

[31] A.-K. Tornberg, B. Engquist, Regularization techniques for numerical approximation of PDEs with singularities, J. Sci. Comp. 19 (2003) 527–552.

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

[33] J.D. Towers, Two methods for discretizing a delta function supported on a level set, J. Comput. Phys. 220 (2) (2007) 915–931.

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

[35] J. Waldén, On the approximation of singular source terms in differential equations, Numer. Meth. Part. D.E. 15 (1999) 503–520.

[36] X. Wen, High order numerical quadratures to one dimensional delta function integrals, preprint.

[37] L. Ying, E. Candes, The phase flow method, J. Comput. Phys. 220 (1) (2006) 184–215.

[38] H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.