Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 219 (2006) 228–246
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method

Y.C. Zhou [a], G.W. Wei [a,b,*]

[a] Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA
[b] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

Received 18 August 2005; received in revised form 9 February 2006; accepted 20 March 2006
Available online 2 June 2006

This paper is dedicated to Professor Charles S. Peskin on the occasion of his 60th birthday.

Abstract

This work overcomes the difficulty of dealing with large curvatures in a high order matched interface and boundary (MIB) method proposed for solving elliptic interface problems. The MIB method smoothly extends the solution across the interface so that standard high order central finite difference schemes can be used without the loss of accuracy. One feature of the MIB is that it disassociates the discretization of the elliptic equation from the enforcement of interface jump conditions. The other is to make iterative use of only the lowest order jump conditions to determine the fictitious values on extended domains. It is of arbitrarily high order in convergence, in principle. However, its applicability was hindered by the lack of sufficiently many grid points to determine all the fictitious values required for high order schemes at the location where the curvature of the interface is relatively large. We remove this obstacle by introducing a new concept, the disassociation between the discretization and the domain extension. We show that the improved MIB method is robust for handling general irregular interfaces by extensive numerical experiments on the Poisson equation and the Helmholtz equation. To better understand the MIB method and other potential high order interface schemes, we propose an alternative interpolation formulation of the MIB method and show that the new formulation is essentially equivalent to the improved one.

Keywords: Elliptic interface problems; Immersed interface method; Immersed boundary method; Ghost fluid method; High order method

1. Introduction

This work is a continuation of authors' effort to develop high order numerical methods for solving elliptic interface problems

$$\nabla \cdot (\beta \nabla u) - \kappa u(\mathbf{x}) = q(\mathbf{x}), \quad \mathbf{x} \in \Omega = \Omega^+ \cup \Omega^- \tag{1}$$

* Corresponding author. Tel.: +1 517 3534689; fax: +1 517 4321562.
E-mail address: wei@math.msu.edu (G.W. Wei).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.03.027

with Dirichlet boundary condition at $\partial\Omega$ on regular Cartesian grids. Here, the computational domain Ω is assumed to be regular, such as rectangle in two dimensions (2D) or parallelepiped in three dimensions (3D). The elliptic interface problems are distinguished by an interface Γ in domain Ω, where locates the discontinuity of the coefficient function $\beta(\mathbf{x})$ of the equation. The source term $q(\mathbf{x})$ may be singular at the interface. For the existence and the regularity of the solution to this equation, the reader is referred to Ref. [6]. The main conclusion of the analysis is that the elliptic interface problem is otherwise unsolvable unless it is supplemented from the underlying physics with two jump conditions across the interface Γ

$$[u] = u^+(\mathbf{X}(s)) - u^-(\mathbf{X}(s)) = \phi(s), \quad [\beta u_n] = \beta^+ u_n^+(\mathbf{X}(s)) - \beta^- u_n^-(\mathbf{X}(s)) = \psi(s), \tag{2}$$

where $\mathbf{X}(s)$ is a point on the interface Γ which is parametrized with arc-length s, and n is the unit outer normal vector. The superscript $-$ or $+$ denotes the limiting value of a function from one side or the other of the interface. A simple Cartesian grid is preferred in our study since the complicated procedure of generating unstructured grid could be bypassed and well developed fast algebraic solvers could be utilized. Attaining highly accurate numerical solutions to these problems with standard numerical methods is subject to the constraint of the low global regularity of the solution. In fact, traditional numerical methods are usually constructed with the assumption that the solution has enough regularity, which is not true due to the interface jumps. Nevertheless, by delicate use of interface jump conditions, it is possible to formulate special interface schemes that are accurate and efficient.

Peskin pioneered an immersed boundary method (IBM) [28–30] in modeling blood flows in heart, where the singular source at the time-varying boundary is regularized by using discrete delta functions [28,30]. The singularity of the problem is then removed and standard discretization methods become applicable. Although the original IBM is only of first order in convergence, it has been extensively used in engineering computations due to its simplicity, efficiency and robustness [9,13,19]. High order generalization of the IBM has been achieved by Peskin and his coworkers [15,23]. By choosing some sophisticate discrete delta functions with a narrow support, Tornberg and Engquist [32] proposed a globally fourth-order scheme for problems with singular sources at the interface. Since Peskin's pioneer work, the importance of the elliptic interface problem has been well established in a variety of disciplines, such as fluid dynamics, molecular biology, electromagnetics and material science.

A major progress in the field was due to LeVeque and Li who proposed the immersed interface method (IIM) [24]. The IIM is one of the most popular schemes which are designed to preserve interface jumps in solving elliptic equations. By defining polynomials up to the second order at each side of the interface, the IIM is of second order in convergence, although its local truncation error at irregular points is of $O(h)$. The original IIM has been improved in many aspects, such as the preservation of discrete maximum principle [26], a multigrid method [1] and a fast algorithm if the problem comes with piecewise constant coefficient [25]. The coupling of the IIM with the level set approach to deal with moving interfaces has also been established [7,31]. The IIM has been successfully applied to a number of important problems [18,20,22,33]. It will be interesting to show that a higher order IIM can be constructed following the IIM procedure.

Another important technique that does not suffer from the numerical smearing of discontinuity at the interface is the ghost fluid method (GFM) proposed by Osher and co-workers [10]. In the GFM, interface jump conditions are captured, i.e., the interface jump conditions are applied on the nearest grid points of an interface, instead of at the exact interface position. This treatment gives rise to a symmetric matrix for the associated linear system, and thus many advanced linear solvers could be taken advantage of. The GFM is simple and easy to use for complex interfaces.

An important application of interface methods is the treatment of elliptic problems defined in an irregular domain [8,11,21]. By embedding the irregular domain into a slightly larger while regular domain, one ends up with a *pseudo* interface problem, in the sense that at the interface, i.e., the real irregular boundary, there is only one condition coming from the original definition of the problem, whereas for genetic elliptic interface problems there are two interface jump conditions. Because the solution outside the interface is not of interest and is often trivial, as given by the Dirichlet condition, such a pseudo interface problem is relatively easy to solve. In such a case, one can simply adopt high order extrapolation techniques to find the special discretization near the boundary [14]. It has been shown by Gibou and Fedkiw [14] that using linear, quadratic or cubic extrapolation, one could attain, respectively second, third and fourth-order boundary schemes for certain irregular boundary shapes.

Apart from the IBM, IIM and GFM, many other interesting approaches have been proposed in the literature, such as the finite element method [2,5,17], discontinuous Galerkin approach [16], integral equation approach [27] and others [3,4].

The matched interface and boundary (MIB) technique, originally proposed in [34], is the first method which provides a systematic approach to generate finite difference schemes of arbitrarily high orders for interface problems in computational electromagnetics. For elliptic interface problems, the MIB can be regarded as a higher-order generalization of the IIM and the GFM. In the MIB technique, the solution on each side of the interface is smoothly extended beyond the interface by means of fictitious domains. As a result, standard high order central finite difference discretization can therefore be applied on the fictitious domains without the loss of accuracy. The fictitious values on fictitious domains are determined simultaneously from enforcing the interface jump conditions at the exact position of the interface. The MIB method iteratively uses the lowest order interface jump conditions so that sufficiently many fictitious values can be determined to support higher order schemes. For straight interfaces, MIB schemes of up to 16th order have been constructed [34,35]. For curved interfaces, up to sixth-order schemes have been demonstrated [35]. The nature of high accuracy, the robustness against large jumps in coefficients, and the flexibility in treating complex interfaces have been examined by extensive numerical experiments. Key ingredients that are crucial to the success of the MIB method are two concepts. One is the disassociation of the discretization of the differential equation from the implementation of interface jump conditions. The other is the iterative use of lowest order jump conditions. However, authors' attempt to formulate high order schemes for general curved interfaces is hindered by the difficulty that there are no sufficient grid points at certain curved interface locations to determine sufficiently many fictitious values required for high order schemes [35]. Consequently, although the original MIB method is of arbitrarily high order in principle, it cannot be implemented beyond the second order for interfaces with large curvatures.

The objective of the present work is to overcome the aforementioned difficulty of the MIB method by introducing a new concept, i.e., the disassociation of the discretization of the differential equation from the extension of the solution across the interface. Specifically, in the original MIB method [35], fictitious values are determined along the direction of each discretization. Consequently, situations arise that there are no sufficient grid points along the discretization direction to support high order schemes near the interface where the curvature is large. In the present procedure, we first identify irregular grid points (i.e., fictitious domains) near the interface according to the need of a high order discretization scheme. We then smoothly extend the solution across the interface onto the fictitious domain *disregarding* the origin of each irregular grid point, by the iterative use of lowest order jump conditions. Finally, we discretize the differential equation across the interface with the fictitious values. With this new procedure, we show that the improved MIB is indeed arbitrarily high order for general curved interface and boundary. We also propose an alternative interpolation formulation of the MIB method without the explicit use of fictitious domains. We show that the new formulation is essentially equivalent to the improved fictitious domain formulation. It is believed that this alternative formulation not only offers a better understanding of the MIB method, but also provides insights into other similar high order interface methods that might be constructed.

The rest of this paper is organized as follows. In Section 2, we review the essential ideas of the fictitious domain formulation of the MIB method. The difficulty in the domain extension is discussed and the new concept, i.e., the disassociation of the domain extension from the discretization, is introduced. Section 3 is devoted to the new interpolation formulation of the MIB method. A comparison between two formulations is given. The numerical experiments of the proposed MIB method are carried out in Section 4. We demonstrate the first observation of fourth-order convergence for a variety of general curved elliptic interface problems. This paper ends with a conclusion.

2. Fictitious domain formulation

In this section, we present a brief description of the MIB method via the fictitious domain formulation. The existing difficulty in constructing high order interface schemes is analyzed and its remedy is proposed.

2.1. Basic formalism

It is illustrative to present the MIB method with a simple 1D elliptic interface equation

$$(\beta u_x)_x - \kappa u = q(x), \quad x \in [0,1], \tag{3}$$

with two interface jump conditions accompanying the discontinuity of coefficient β at $x = a \in (0,1)$

$$[u] = u^+ - u^- = \phi, \quad [\beta u_x] = \beta^+ u_x^+ - \beta^- u_x^- = \psi, \tag{4}$$

which is to be solved on a uniform mesh of size $\Delta x = h$. To simplify the presentation we assume β is piecewise constant while the MIB scheme for general piecewise continuous β is almost identical. Let x_i and x_{i+1} be two grid points closest to the interface at $x = a$, as in Fig. 1, the application of conventional finite difference schemes over the interface is problematic. For example, $(u_{i+1} - u_i)/h$ is of O$(1/h)$, i.e., divergent, due to the finite jump $[u]$ at $x = a$. The MIB method, like many other interface methods, manages to improve the convergence by incorporating interface conditions into the numerical discretization. In particular, with the MIB, the solutions on two subdomains separated by the interface will be smoothly and simultaneously extended using two interface conditions. The numerical difference can then be carried out on smoothly extended subdomains. The length of this extension depends on the width of the discretization stencil of a given central finite difference scheme, because only the discretization at nodes near the interface will involve grid points in the other subdomain. If a fourth-order central difference scheme is chosen as the basic difference scheme, we expect the second-order derivative u_{xx} at points x_{i-1}, x_i, x_{i+1} and x_{i+2} to be discretized as

$$u_{xx} = -\frac{f_{i+1}}{12h^2} + \frac{4u_i}{3h^2} - \frac{u_{i-1}}{4h^2} + \frac{4u_{i-2}}{3h^2} - \frac{u_{i-3}}{12h^2} \quad \text{at } x_{i-1}, \tag{5}$$

$$u_{xx} = -\frac{f_{i+2}}{12h^2} + \frac{4f_{i+1}}{3h^2} - \frac{u_i}{4h^2} + \frac{4u_{i-1}}{3h^2} - \frac{u_{i-2}}{12h^2} \quad \text{at } x_i, \tag{6}$$

$$u_{xx} = -\frac{u_{i+3}}{12h^2} + \frac{4u_{i+2}}{3h^2} - \frac{u_{i+1}}{4h^2} + \frac{4f_{i-1}}{3h^2} - \frac{f_{i-2}}{12h^2} \quad \text{at } x_{i+1}, \tag{7}$$

$$u_{xx} = -\frac{u_{i+4}}{12h^2} + \frac{4u_{i+3}}{3h^2} - \frac{u_{i+2}}{4h^2} + \frac{4u_{i+1}}{3h^2} - \frac{f_i}{12h^2} \quad \text{at } x_{i+2}, \tag{8}$$

where f_{i+1} and f_{i+2} are the extension of the solution in the left subdomain to x_{i+1} and x_{i+2}, respectively. Similarly, the solution in the right subdomain need to be continued to points x_i and x_{i-1}, which are represented as f_i and f_{i-1}. Here f_{i-1} through f_{i+2} are referred as fictitious values, to distinguish them from solution values u_i through u_{i+2} defined on the same set of grid points. These fictitious values will be related to solution values by appropriate implementation of the interface conditions, which can be regarded as the governing equations for fictitious values in this sense.

In the MIB, these four fictitious values are solved via iterative use of the interface conditions as only two can be solved from two interface conditions. Since β is a piecewise constant as assumed above, we could first solve f_i and f_{i+1} from

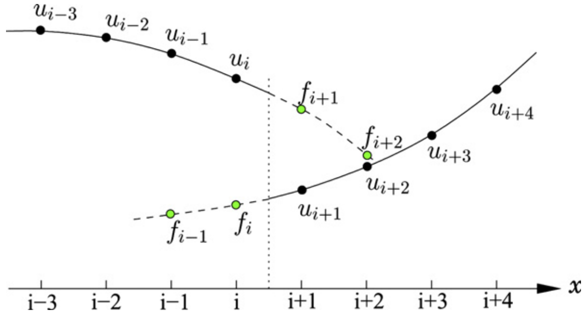

Fig. 1. Illustration of fictitious values for 1D problem.

$$\left(w^+_{0,i}f_i + \sum_{k=i+1}^{i+4} w^+_{0,k}u_k\right) - \left(\sum_{k=i-3}^{i} w^-_{0,k}u_k + w^-_{0,i+1}f_{i+1}\right) = \phi, \tag{9}$$

$$\beta^+\left(w^+_{1,i}f_i + \sum_{k=i+1}^{i+4} w^+_{1,k}u_k\right) - \beta^-\left(\sum_{k=i-3}^{i} w^-_{1,k}u_k + w^-_{1,i+1}f_{i+1}\right) = \psi, \tag{10}$$

and then solve f_{i-1} and f_{i+2} from

$$\left(w^+_{0,i-1}f_{i-1} + w^+_{0,i}f_i + \sum_{k=i+1}^{i+4} w^+_{0,k}u_k\right) - \left(\sum_{k=i-3}^{i} w^-_{0,k}u_k + w^-_{0,i+1}f_{i+1} + w^-_{0,i+2}f_{i+2}\right) = \phi, \tag{11}$$

$$\beta^+\left(w^+_{1,i-1}f_{i-1} + w^+_{1,i}f_i + \sum_{k=i+1}^{i+4} w^+_{1,k}u_k\right) - \beta^-\left(\sum_{k=i-3}^{i} w^-_{1,k}u_k + w^-_{1,i+1}f_{i+1} + w^-_{1,i+2}f_{i+2}\right) = \psi, \tag{12}$$

where Eqs. (9) and (10) are the approximation of jump conditions (4), $w^+_{0,:}$ and $w^-_{0,:}$ are the interpolation coefficients on the respective subdomain, and $w^+_{1,:}$ and $w^-_{1,:}$ are the first-order finite difference coefficients. Once being solved as the linear combinations of $u_{i-1}, \ldots, u_{i+2}, \phi$ and ψ, fictitious values f_i and f_{i+1} become known values in Eqs. (11) and (12) from which other two fictitious values f_{i-1} and f_{i+2} are also solved. See Refs. [34,35] for the general expressions of fictitious values and difference schemes, as well as the generalization of above derivation to a 2D domain. The standard interpolation weights and finite difference coefficients in these schemes can be calculated using Fornberg's method [12].

In addition to the convergence rate of the scheme, we are also interested in the relation between the accuracy of the MIB scheme and the magnitude of the jump in diffusion coefficient β. Considering a second-order MIB scheme for which only one fictitious value is defined in either side of the interface, i.e., f_i and f_{i+1}, the interface conditions (4) can be approximated by

$$(w^+_{0,i}f_i + w^+_{0,i+1}u_{i+1} + w^+_{0,i+2}u_{i+2}) - (w^-_{0,i-1}u_{i-1} + w^-_{0,i}u_i + w^-_{0,i+1}f_{i+1}) = \phi, \tag{13}$$

$$\beta^+(w^+_{1,i}f_i + w^+_{1,i+1}u_{i+1} + w^+_{1,i+2}u_{i+2}) - \beta^-(w^-_{1,i-1}u_{i-1} + w^-_{1,i}u_i + w^-_{1,i+1}f_{i+1}) = \psi. \tag{14}$$

On the other hand, interface conditions (4) can also be approximated by using the *exactly* extended values u^+_i and u^-_{i+1}, respectively defined at x_i and x_{i+1}

$$(w^+_{0,i}u^+_i + w^+_{0,i+1}u_{i+1} + w^+_{0,i+2}u_{i+2} + C^+_0h^3) - (w^-_{0,i-1}u_{i-1} + w^-_{0,i}u_i + w^-_{0,i+1}u^-_{i+1} + C^-_0h^3) = \phi, \tag{15}$$

$$\beta^+(w^+_{1,i}u^+_i + w^+_{1,i+1}u_{i+1} + w^+_{1,i+2}u_{i+2} + C^+_1h^2) - \beta^-(w^-_{1,i-1}u_{i-1} + w^-_{1,i}u_i + w^-_{1,i+1}u^-_{i+1} + C^-_1h^2) = \psi, \tag{16}$$

where C^-_0, C^+_0, C^-_1 and C^+_1 are the coefficients of the leading truncation errors. Note that fictitious values f_i and f_{i+1} are the approximation to the exact values u^+_i and u^-_{i+1} on fictitious domains, respectively. Denote the difference between these fictitious values and the extended exact values as ϵ_i and ϵ_{i+1}, i.e.,

$$\epsilon_i = u^+_i - f_i, \tag{17}$$

$$\epsilon_{i+1} = u^-_{i+1} - f_{i+1}. \tag{18}$$

By subtracting Eq. (13) from Eq. (15), and Eq. (14) from Eq. (16), we notice that ϵ_i and ϵ_{i+1} satisfy the following two equations

$$w^+_{0,i}\epsilon_i - w^-_{0,i+1}\epsilon_{i+1} = (C^-_0 - C^+_0)h^3, \tag{19}$$

$$\beta^+w^+_{1,i}\epsilon_i - \beta^-w^-_{1,i+1}\epsilon_{i+1} = (\beta^-C^-_1 - \beta^+C^+_1)h^2. \tag{20}$$

It can be solved that

$$\epsilon_i = \frac{-\beta^-(C^-_0 - C^+_0)w^-_{1,i+1}h^3 - w^-_{0,i+1}(C^-_1\beta^- - C^+_1\beta^+)h^2}{-\beta^-w^+_{0,i}w^-_{1,i+1} + \beta^+w^+_{1,i}w^-_{0,i+1}}. \tag{21}$$

Since $w^-_{0,i+1} \sim O(1), w^+_{0,i} \sim O(1), w^-_{1,i+1} \sim O(1/h)$ and $w^+_{1,i} \sim O(1/h)$, it follows that $\epsilon_i \sim O(h^3)$. Moreover, since

$$\epsilon_i = \frac{-(C_0^- - C_0^+)w_{1,i+1}^- h^3 - w_{0,i+1}^-(C_1^- - C_1^+ \frac{\beta^+}{\beta^-})h^2}{-w_{0,i}^+ w_{1,i+1}^- + \frac{\beta^+}{\beta^-} w_{1,i}^+ w_{0,i+1}^-}, \tag{22}$$

as β^+/β^- increasing, ϵ_i approaches asymptotically to $C(h^3)$, where constant C depends only on C_1^+ and $w_{1,i}^+$. In other words, for large jumps of the coefficient, the approximation error depends only on the solution of the problem. This proves that the present method is robust against the large contrast of β at the interface.

2.2. Subtleness in domain extensions

In 2D, our previous MIB method in [35] did not specify in sufficient details how to determine fictitious values for a given topology of irregular points. In fact, we have learned that this is a subtle issue. A computational practice used in [35] is to solve a fictitious value along the direction of discretization. This is based on the following understanding: A fictitious value is required because the need for a smooth continuation in a particular discretization in either the x- or the y-direction. As a fictitious value at a given point varies when it is obtained via smooth extension from different directions, a fictitious value is regarded as a smooth continuation *only* in the direction associated with the original discretization. According to this association between the domain extension and the discretization, $f_{i+1,j}$ should only be solved along the x-direction, see Fig. 2. Similarly, $f_{i,j-1}$ should be determined twice, along the x- and y-directions, respectively. Unfortunately, $f_{i,j-1}$ cannot be solved along the y-direction up to fourth-order accuracy due to the large curvature in Fig. 2. It is this difficulty that has severely constrained the applicability of the MIB scheme in Ref. [35], where only one generic elliptic interface problem with relatively low curvatures was solved up to fourth-order accuracy.

A close look at the fictitious values in Figs. 2 and 3 reveals that a fictitious value must fall into one of the following five categories:

1. That can be solved only along one of the x- and y-directions and will be used for the discretization along the same direction.
2. That can be solved only along one of the x- and y-directions, but will be used for the discretization along the other direction.
3. That can be solved only along one of the x- and y-directions, and will be used for the discretization along both directions.
4. That can be solved along both the x- and y-directions, and will be used for the discretization along either one, or both directions.
5. That cannot be solved along any direction, but will be used for the discretization along one or both directions.

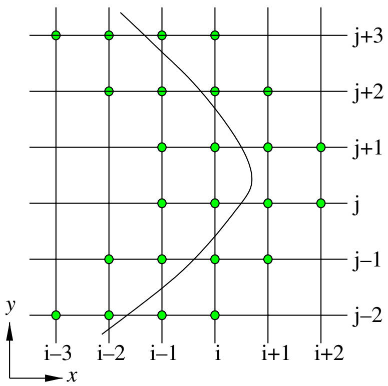

Fig. 2. Fictitious value $f_{i,j+2}$ shall be used for the discretization of u_y, u_{yy} at grid point $(i,j+1)$. However, $f_{i,j+2}$ cannot be solved along the y-direction but it can be solved along x-direction, i.e., solved together with fictitious values $f_{i-2,j+2}$, $f_{i-1,j+2}$ and $f_{i+1,j+2}$.

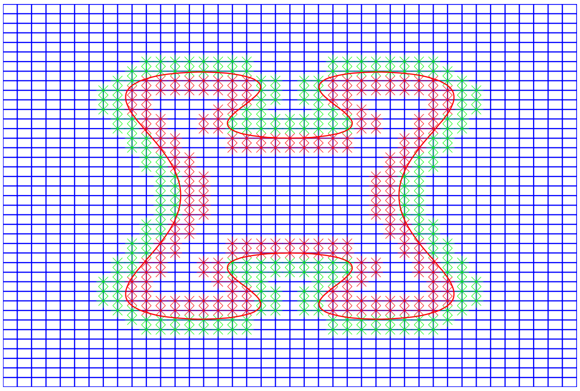

Fig. 3. The distribution of fictitious points (stars) for a 40×40 mesh with irregular interface. Each fictitious domain roughly has two layers of fictitious points, supporting a fourth-order central difference scheme. The green stars represent the smooth continuation of the interior subdomain and the red stars denote the smooth continuation of the outer subdomain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

It is seen that a fictitious value in either the second, the third or the fifth category would be deemed as unsolvable. In Fig. 2, for example, to find four fictitious values $f_{i,j},\ldots,f_{i,j+3}$ along the y-direction, grid points (x_i, y_{j-2}) through (x_i, y_{j+1}) are supposed to be on same side of the interface, whereas the other four grid points, (x_i, y_{j+2}) through (x_i, y_{j+5}), must be on the other side. Similarly, to find four fictitious values $f_{i,j-2},\ldots,f_{i,j+1}$, by considering the extension in the y-direction, we need points (x_i, y_{j-4}) through (x_i, y_{j-1}) on one side and (x_i, y_j) through (x_i, y_{j+3}) on the other side. It is obvious that the distribution of grid points with respect to the interface in Fig. 2 does not satisfy these requirements. In particular, it is unable to find fictitious values $f_{i,j-2}$ through $f_{i,j+3}$ by the extension in the y-direction along ith mesh line, since there are only two grid points in one subdomain on that mesh line. According to the above discussion, however, at least four grid points in either subdomain are needed in order to define a fourth-order extension. These fictitious values can be classified into the third type, and were unsolvable by using the practice of Ref. [35].

In this work, we introduce a new concept, the disassociation of the domain extension from the discretization. This concept is motivated by the following error analysis. If the approximation errors of fictitious values at a given point (x_i, y_j) obtained from the x- and y-directions are both of $O(h^n)$ for a positive n, their difference must be of $O(h^n)$. Therefore, a fictitious value at an irregular point (x_i, y_j), regardless the direction in its calculation, can be used for the discretization in any direction involving the grid point without the loss of accuracy. For example, fictitious values $f_{i,j}$ through $f_{i,j+3}$ are essentially the extension of the inner or the outer subdomain. Although they cannot be calculated through the extension in the y-direction, they indeed can be obtained by the extension in the x-direction. Particularly, one can show that $f_{i,j+2}$ can be calculated considering the extension along $(j+2)$th horizontal mesh line. The known $f_{i,j+2}$ can be used to facilitate the discretization, not only for u_x and u_{xx} at (x_{i-2}, y_{j+2}) and (x_{i-1}, y_{j+2}), respectively, but also for u_y and u_{yy} at (x_i, y_j) and (x_i, y_{j+1}), respectively. This new understanding makes it possible to solve the fictitious values of the second and third types, and significantly broadens the applicability of the MIB method to general interface geometry.

For the fictitious values of the fourth type, i.e., those that can be extended in both the x- and y-directions, such as $f_{i-1,j-2}, f_{i-1,j+3}$ in Fig. 2, we may attain their values in the most convenient manner and use them for necessary discretization.

Finally, fictitious values of the fifth type may be made available in most cases by refining the mesh. For example, if there are only four grid points inside a circular interface, fictitious values cannot be solved for all higher order schemes, i.e., orders higher than two, on such a grid. However, when the grid size is doubled in both directions, the fourth order fictitious values can be obtained. For this reason, the present MIB method is a robust high order approach for all curved interfaces.

It was claimed in Ref. [35] that the proposed MIB scheme is of arbitrarily high order in principle. Indeed, the MIB procedure is systematic and it encompasses a variety of high order schemes. However, the high order convergence was demonstrated only for an interface problem on a special geometry due to the association of the domain extension from the discretization. It is believed that with the present understanding and extension technique, we are able to fully realize the high-order potential of the MIB method for arbitrarily curved geometry.

3. Interpolation formulation

The MIB method described in the preceding section makes use of fictitious domains and values for the standard high order finite difference discretization of the governing equation near the interface. The fictitious values give a smooth continuation of the solution across the interface. Interface jump conditions are iteratively used to determine fictitious values. In this section, the possibility of the smooth continuation in terms of polynomial expansions is investigated. This interpolation formulation does not require the use of fictitious domains and values. To illustrate the idea, we start our discussion with a 1D problem. The general principle is then applied to 2D problems with curved interfaces. Particular attentions are paid to the relation between these formulations.

3.1. One dimensional formalism

Consider the 1D elliptic problem given in Eq. (3). The interface is located at $x_i \leqslant a \leqslant x_{i+1}$. In a second-order central finite difference scheme, x_i is the irregular point to the left of the interface, and x_{i+1} the irregular point to the right. On each side of the interface, we define a second-order polynomial

$$u^-(x) = a_0^- + \frac{a_1^-}{h}(x - x_i) + \frac{a_2^-}{h^2}(x - x_i)^2, \tag{23}$$

$$u^+(x) = a_0^+ + \frac{a_1^+}{h}(x - x_{i+1}) + \frac{a_2^+}{h^2}(x - x_{i+1})^2, \tag{24}$$

where the polynomial coefficients are scaled by h to reduce the condition number of the coefficient matrix. Obviously, $a_0^- = u_i$ and $a_0^+ = u_{i+1}$ and the remaining four coefficients can be determined from the expansion of the polynomials at two grid points, and from two interface jump conditions

$$-a_1^- + a_2^- = u_{i-1} - u_i, \tag{25}$$

$$a_1^+ + a_2^+ = u_{i+2} - u_{i+1}, \tag{26}$$

$$(-a_1^+ x_r + a_2^+ x_r^2) - (a_1^- x_l + a_2^- x_l^2) = \phi - u_{i+1} + u_i, \tag{27}$$

$$\beta^+(a_1^+ - 2a_2^+ x_r) - \beta^-(a_1^- + 2a_2^- x_l) = h\psi, \tag{28}$$

where $x_l = \frac{a - x_i}{h}, x_r = \frac{x_{i+1} - a}{h}$. The first two Eqs. (25) and (26), are the realizations of $u^-(x)$ at x_{i-1}, and $u^+(x)$ at x_{i+2}, respectively. The last two Eqs. (27) and (28), are the approximations of the interface jump conditions with $u^-(x)$ and $u^+(x)$. Eqs. (25)–(28) essentially provide an algebraic system for the coefficients of $u^-(x)$ and $u^+(x)$, whose solutions are the representations of those four polynomial coefficients in terms of u_{i-1}, u_i, u_{i+1}, u_{i+2}, ϕ and ψ. These polynomial coefficients are solved by inverting the coefficient matrix of equation

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -x_l & -x_l^2 & -x_r & x_r^2 \\ -\beta^- & -2\beta^- x_l & \beta^+ & -2\beta^+ x_r \end{pmatrix} \begin{pmatrix} a_1^- \\ a_2^- \\ a_1^+ \\ a_2^+ \end{pmatrix} = \begin{pmatrix} u_{i-1} - u_i \\ u_{i+2} - u_{i+1} \\ \phi - u_{i+1} + u_i \\ h\psi \end{pmatrix}. \tag{29}$$

It can be seen from Eqs. (23) and (24) that $u_x^-(x_i) = \frac{a_1^-}{h}, u_{xx}^-(x_i) = \frac{2a_2^-}{h^2}, u_x^+(x_{i+1}) = \frac{a_1^+}{h}$, and $u_{xx}^+(x_{i+2}) = \frac{2a_2^+}{h^2}$. The elliptic equation can therefore be approximated at irregular points by

$$\beta^- \frac{2a_2^-}{h^2} - \kappa(x_i)u_i = q(x_i) \quad \text{at } x_i, \tag{30}$$

$$\beta^+ \frac{2a_2^+}{h^2} - \kappa(x_{i+1})u_{i+i} = q(x_{i+1}) \quad \text{at } x_{i+1}, \tag{31}$$

if β is piecewise constant. Otherwise the approximation equations would be

$$\beta_x(x_i)\frac{a_1^-}{h}+\beta(x_i)\frac{2a_2^-}{h^2}-\kappa(x_i)u_i=q(x_i)\quad \text{at } x_i, \tag{32}$$

$$\beta_x(x_{i+1})\frac{a_1^+}{h}+\beta(x_{i+1})\frac{2a_2^+}{h^2}-\kappa(x_{i+1})u_{i+i}=q(x_{i+1})\quad \text{at } x_{i+1}. \tag{33}$$

The constants in the representations of a_1^-, a_2^-, a_1^+ and a_2^+ in the above equations, i.e., the terms involving given jumps ϕ and ψ, should be moved to the right hand side. This finishes the establishment of a second-order interpolation scheme at both irregular points.

The above procedure can be easily generalized to construct higher order schemes. For example, to attain a fourth order scheme at irregular points near the interface, one can start the formulation of the scheme by defining a fourth-order polynomial on each side of the interface

$$u^-(x)=a_0^-+\frac{a_1^-}{h}(x-x_i)+\frac{a_2^-}{h^2}(x-x_i)^2+\frac{a_3^-}{h^3}(x-x_i)^3+\frac{a_4^-}{h^4}(x-x_i)^4, \tag{34}$$

$$u^+(x)=a_0^++\frac{a_1^+}{h}(x-x_{i+1})+\frac{a_2^+}{h^2}(x-x_{i+1})^2+\frac{a_3^+}{h^3}(x-x_{i+1})^3+\frac{a_4^+}{h^4}(x-x_{i+1})^4. \tag{35}$$

The first two coefficients a_0^- and a_0^+ again have to be u_i and u_{i+1}, respectively. The remaining eight coefficients are to be determined by using two interface jump conditions, and by expanding these two polynomials at six grid points, i.e., points x_{i-1}, x_{i-2} and x_{i-3} for Eq. (34), and points x_{i+2}, x_{i+3} and x_{i+4} for Eq. (35). We could therefore end up with eight linear algebraic equations

$$\begin{pmatrix} -3 & 9 & -27 & 81 & 0 & 0 & 0 & 0 \\ -2 & 4 & -8 & 16 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 4 & 8 & 16 \\ 0 & 0 & 0 & 0 & 3 & 9 & 27 & 81 \\ -x_l & -x_l^2 & -x_l^3 & -x_l^4 & -x_r & x_r^2 & -x_r^3 & x_r^4 \\ -\beta^- & -2\beta^-x_l & -3\beta^-x_l^2 & -4\beta^-x_l^3 & \beta^+ & -2\beta^+x_r & 3\beta^+x_r^2 & -4\beta^+x_r^3 \end{pmatrix}\begin{pmatrix} a_1^- \\ a_2^- \\ a_3^- \\ a_4^- \\ a_1^+ \\ a_2^+ \\ a_3^+ \\ a_4^+ \end{pmatrix}=\begin{pmatrix} u_{i-3}-u_i \\ u_{i-2}-u_i \\ u_{i-1}-u_i \\ u_{i+2}-u_{i+1} \\ u_{i+3}-u_{i+1} \\ u_{i+4}-u_{i+1} \\ \phi-u_{i+1}+u_i \\ h\psi \end{pmatrix}, \tag{36}$$

where the first three equations are generated from $u^-(x)$, while the next three equations are generated from $u^+(x)$. The last two equations are the approximation of two interface jump conditions with polynomials $u^-(x)$ and $u^+(x)$ at $x = a$. By inverting the coefficient matrix, the representations of all the eight polynomial coefficients are solved, and expressions for $u^-(x)$ and $u^+(x)$ are fully determined.

Unlike the second-order case where each polynomial is used only once in formulating the difference scheme at either x_i or x_{i+1}, for the fourth-order case, each polynomial are applied twice since there are two irregular points on each side of the interface, i.e., x_{i-1} and x_i on the left, and x_{i+1} and x_{i+2} on the right. These points are identified since the regular central difference scheme at these points involves the grid point(s) on the different side. In particular, $u^-(x)$ will be used to formulate the difference scheme at points x_{i-1} and x_i, while $u^+(x)$ is used at x_{i+1} and x_{i+2}. Assuming a piecewise continuous β, the difference schemes at two closest irregular points are

$$\beta_x(x_i)\frac{a_1^-}{h}+\beta(x_i)\frac{2a_2^-}{h^2}-\kappa(x_i)u_i=q(x_i)\quad \text{at } x_i, \tag{37}$$

$$\beta_x(x_{i+1})\frac{a_1^+}{h}+\beta(x_{i+1})\frac{2a_2^+}{h^2}-\kappa(x_{i+1})u_{i+i}=q(x_{i+1})\quad \text{at } x_{i+1}. \tag{38}$$

At other two irregular points, the difference schemes are more complicated since the derivatives of the polynomials involve more terms

$$\frac{\beta_x(x_{i-1})}{h}(a_1^- - 2a_2^- + 3a_3^- - 4a_4^-) + \frac{\beta(x_{i-1})}{h^2}(2a_2^- - 6a_3^- + 12a_4^-) - \kappa(x_{i-1})u_{i-1} = q(x_{i-1}) \quad \text{at } x_{i-1}, \tag{39}$$

$$\frac{\beta_x(x_{i+2})}{h}(a_1^+ + 2a_2^+ + 3a_3^+ + 4a_4^+) + \frac{\beta(x_{i+2})}{h^2}(2a_2^+ + 6a_3^+ + 12a_4^+) - \kappa(x_{i+2})u_{i+2} = q(x_{i+2}) \quad \text{at } x_{i+2}. \tag{40}$$

It can be concluded from above discussion that at irregular points near the interface, a difference scheme of arbitrary order can be formulated by following these steps:

1. Definite a polynomial on each side of the interface

$$u^-(x) = u_i + \sum_{k=1}^{n} \frac{a_k^-}{h^k}(x - x_i), \tag{41}$$

$$u^+(x) = u_{i+1} + \sum_{k=1}^{n} \frac{a_k^+}{h^k}(x - x_{i+1}), \tag{42}$$

   where n, the order of the polynomial, depends on the global accuracy required. These polynomials are essentially the approximations of the solution in the vicinity of the interface.
2. Expand each polynomial at the nearest $n-1$ grid points to the interface on the same side. The total number of expansions is therefore $2n-2$.
3. Approximate two interface jump conditions with these polynomials. Differentiation is needed in approximating $[\beta u_x]$, the jump in flux.
4. Compose a $2n \times 2n$ linear algebraic equations system by combining $2n-2$ expansion polynomials and two discretized interface jump conditions. Solve this linear system by inversion, and the solution is the representation of polynomials coefficients in terms of the approximation solution at involved grid points and the given jumps.
5. Approximate the original elliptic equation at all the irregular points by differentiating the polynomials. Replace the polynomial coefficients with their representations. This gives rise to a finite difference scheme at the corresponding grid point.

3.2. Two dimensional formalism

The new formulation of the MIB for 2D problems with curved interfaces can be accomplished similarly. The essential idea is to construct 1D interpolation polynomials, instead of 2D ones. Referring to Fig. 4, we have two irregular points, x_i and x_{i+1}, along the x-axis for a second order scheme. Two 1D polynomials, in exactly the same form as Eqs. (23) and (24), are defined

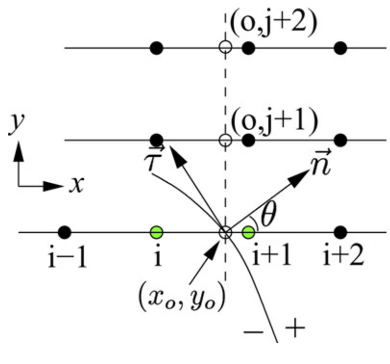

Fig. 4. Irregular point (i,j) and the interface. The interface crosses the x-mesh line at (x_o, y_o). The vertical dash line is the auxiliary line on which three auxiliary points (in empty circle) are defined: $(o, j+2)$, $(o, j+1)$ and (o, j) right at (x_o, y_o). The jumps $[u]$, $[\beta u_n]$ and $[u_\tau]$ are evaluated at (x_o, y_o).

$$u^-(x,y_j) = a_0^- + \frac{a_1^-}{h}(x-x_i) + \frac{a_2^-}{h^2}(x-x_i)^2, \tag{43}$$

$$u^+(x,y_j) = a_0^+ + \frac{a_1^+}{h}(x-x_{i+1}) + \frac{a_2^+}{h^2}(x-x_{i+1})^2. \tag{44}$$

These polynomials can be regarded as the 1D approximation of the solution in the vicinity of the interface along $y = y_j$ mesh line. Here, one has $a_0^- = u_{i,j}$ and $a_0^+ = u_{i+1,j}$ by definition. To solve for the rest four coefficients, these two polynomials are to be evaluated at (x_{i-1}, y_j) and (x_{i+2}, y_j), which gives two equations. The other two equations are obtained from the approximations of interface jump conditions (45) and (46) [35]

$$[u] = u^+ - u^-, \tag{45}$$

$$[\beta u_n] - \beta^- \tan\theta [u_\tau] = C_x^+ u_x^+ + C_x^- u_x^- + C_y^+ u_y^+, \tag{46}$$

where θ is the angle between the normal vector and the x–axis, $C_x^+ = \beta^+ \cos\theta + \beta^- \tan\theta \sin\theta$, $C_x^- = -\beta^-(\cos\theta + \tan\theta\sin\theta)$, and $C_y^+ = (\beta^+ - \beta^-)\sin\theta$. The approximation to Eq. (45) is trivial though, special care has to be taken in treating Eq. (46), which couples two directions. Here we approximate Eq. (46) as

$$\frac{C_x^+}{h}(a_1^+ - 2a_2^+ x_r) + \frac{C_x^-}{h}(a_1^- + 2a_2^- x_l) + C_y^+(p_{1,j}^+ u_{o,j}^+ + p_{1,j+1}^+ u_{o,j+1} + p_{1,j+2}^+ u_{o,j+2}) = [\beta u_n] - \beta^- \tan\theta [u_\tau], \tag{47}$$

where p is the FD weight in the y-direction, and $x_l = \frac{x_o - x_i}{h}$, $x_r = \frac{x_{i+1} - x_o}{h}$. Note that here auxiliary values $u_{o,j}^+$, $u_{o,j+1}$ and $u_{o,j+2}$ are also introduced to calculate u_y^+ as did in fictitious domain formulation [35], where

$$u_{o,j}^+ = u_{x_o,y_o}^- + [u] = (u_{i,j} + a_1^- x_l + a_2^- x_l^2) + [u]. \tag{48}$$

The final algebraic system for four polynomial coefficients is

$$\begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -x_l & -x_l^2 & -x_r & x_r^2 \\ \frac{1}{h}C_x^- + C_y^+ p_{1,j}^+ x_l & \frac{2}{h}C_x^- x_l + C_y^+ p_{1,j}^+ x_l^2 & \frac{1}{h}C_x^+ & -\frac{2}{h}C_x^+ x_r \end{pmatrix} \begin{pmatrix} a_1^- \\ a_2^- \\ a_1^+ \\ a_2^+ \end{pmatrix}$$
$$= \begin{pmatrix} u_{i-1,j} - u_{i,j} \\ u_{i+2,j} - u_{i+1,j} \\ \phi - u_{i+1,j} + u_{i,j} \\ [\beta u_n] - \beta^- \tan\theta[u_\tau] - C_y^+\{p_{1,j}^+(u_{i,j} + [u]) + p_{1,j+1}^+ u_{o,j+1} + p_{1,j+2}^+ u_{o,j+2}\} \end{pmatrix}. \tag{49}$$

Desirable difference schemes for u_x or u_{xx} at two irregular points can be obtained from the direct differentiation of these two polynomials after all of their coefficients are determined. Following a similar procedure, one can determine polynomials $u^-(x_i, y)$ and $u^+(x_i, y)$, which are then used to approximate u_y^-, u_{yy}^-, u_y^+ and u_{yy}^+ at corresponding irregular points.

High order 2D interpolation schemes can be established in a manner similar to what described for 1D high order schemes. However, for general interfaces with large curvatures, the construction of a high order interpolation MIB scheme is subject to the same difficulty as that in the fictitious domain formulation, i.e., there is no sufficient grid points to support high order polynomials along all the x- and y-directions on one side of an interface, see $x = x_i$ mesh line in Fig. 2. It normally takes four grid points to determine a fourth-order polynomial $u^-(x_i, y)$, which is required for calculating u_y and u_{yy} at four irregular points, $(i,j-2)$, $(i,j-1)$, $(i,j+2)$ and $(i,j+3)$, in a fourth-order scheme. One solution to this problem is to make use of two interface intersection points, which would provide four jump conditions. However, a simple remedy is to make use of other two polynomials found in the x-direction, namely, $u^-(x, y_{j+2})$ and $u^-(x, y_{j-1})$ to calculated the extended function values at $(i,j+2)$ and $(i,j-1)$, respectively. These two extended function values, together with the original function values $u_{i,j}$ and $u_{i,j+1}$, as well as two interface jump conditions, can determine $u^-(x_i, y)$. As usual, this determination is sought in coupling with the determination of $u^+(x_i, y)$, whose solution has no additional problem.

3.3. Comparison of two formulations

In this subsection, we discuss the similarity and difference of the fictitious domain formulation and the interpolation formulation. In terms of similarities, both formulations share the same set of irregular points for a given interface geometry and given order of the scheme. Both approaches utilize only the standard (high order) central finite difference discretization and the lowest order interface jump conditions. The essential equivalence between the two formulation would be obvious if the fictitious value formulation is also casted in polynomial expressions. Referring to Fig. 1, bridged by fictitious value f_{i+1}, the solution in the left vicinity of the interface is actually approximated by a Lagrange interpolation polynomial

$$u^-(x) = \sum_{k=0}^{1} L_k u_{i-1+k} + L_2 f_{i+1} \tag{50}$$

to an accuracy of $O(h^3)$ or

$$u^-(x) = \sum_{k=0}^{3} L_k u_{i-3+k} + L_4 f_{i+1} \tag{51}$$

to an accuracy of $O(h^5)$. These two polynomials, although are not explicitly constructed, are actually approximated in seeking the fictitious values. Moreover, a very important common feature is that, in either 1D or higher dimensions, both approaches make use of *only* 1D polynomials. This treatment significantly simplifies the scheme.

Comparing Eq. (50) with Eq. (23), or Eq. (50) with Eq. (34) it can be found that with the fictitious domain approach one only need to determine two parameters, i.e., the fictitious values f_i and f_{i+1}, to resolve the approximate solutions in both the left and the right vicinity of the interface. With the new formulation, however, one has to solve for all the six polynomial coefficients to determine the approximate solution.

Since approximate polynomials are explicitly determined in the new formulation, they are then directly differentiated to provide the approximation to the first and second derivatives in the elliptic equation. In the fictitious domain approach, instead of differentiating approximate polynomials implicitly determined with the first pair of fictitious values f_i and f_{i+1}, a formal central difference scheme involving f_i or f_{i+1} is adopted at each irregular point to approximate the partial derivative. Moreover, to support a fourth or higher order scheme, more fictitious values are needed, and they have to be solved progressively after determining f_i and f_{i+1}. With the new formulation, however, all the coefficients in the high order polynomials are solved simultaneously.

Two formulations also differ from each other in dealing with large curvatures. The fictitious domain approach avoids determining a fictitious value along a discretization direction whenever there is no sufficient number of grid values along the direction inside an interface. It makes use of grid values in the other direction to determine the fictitious value. However, the interpolation formulation cannot avoid determining polynomials in all directions at an irregular point because it has to calculate both the x- and y-derivatives at the irregular point. It therefore makes use of other polynomials to provide function values outside the interface to complete the determination of each required polynomial. These two approaches might involve different sets of grid values in dealing with a given situation.

Table 1
Comparison of two formulations of the MIB method

n_x	Fictitious domain formulation				Interpolation formulation			
	2nd		4th		2nd		4th	
	L_∞	Order	L_∞	Order	L_∞	Order	L_∞	Order
20	3.45E − 2		6.79E − 4		3.45E − 2		6.79E − 4	
40	5.76E − 3	2.58	3.48E − 5	4.29	5.76E − 3	2.58	3.48E − 5	4.29
80	2.12E − 3	1.44	2.33E − 6	3.90	2.12E − 3	1.44	2.33E − 6	3.90
160	3.51E − 4	2.59	1.49E − 7	3.97	3.51E − 4	2.59	1.49E − 7	3.97
320	1.32E − 4	1.41	9.43E − 9	3.98	1.32E − 4	1.41	9.43E − 9	3.98

To complete the comparison of two formulations, a simple 1D Poisson problem is solved with both approaches. The computational domain is $[-1,1]$ and the interface is located at $x = 1/3$. To the left of the interface $\beta = x^2 + 1$, $u(x) = \sin(4x)$ and $\beta = e^x$, $u(x) = \cos(4x)$ in the right subdomain. The jumps at the interface can be calculated accordingly. The convergence tests showing in Table 1 verifies the equivalence of these two formulations. Here, small difference can be detected if more significant digits were kept. For example, for $n_x = 20$ the L_∞ error for the second-order fictitious domain formulation is 3.45332×10^{-2}, while for the interpolation formulation is 3.45481×10^{-2}.

4. Numerical experiments

We conducted a number of numerical experiments to examine the performance of the proposed high order MIB method, with special care on the order of convergence. Interface problems with general curved geometry are selected to test robustness of the present method in dealing with large curvatures. Tests are also conducted on problems with high frequency oscillations and large coefficient jumps. The numerical errors of computations are measured in the discrete L_2 norm

$$L_2 = \sqrt{\frac{1}{n_x \cdot n_y} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} (u_{i,j} - \tilde{u}_{i,j})^2}$$

or the L_∞ norm

$$L_\infty = \max_{i=1,\cdots,n_x} \max_{j=1,\cdots,n_y} |u_{i,j} - \tilde{u}_{i,j}|,$$

given an analytical solution $u_{i,j}$ and a computed solution $\tilde{u}_{i,j}$. Here, n_x and n_y are the numbers of grid points in the x- and y-direction, respectively. A main-diagonal preconditioned biconjugate gradient stabilized solver is adopted to solve the linear system.

Example 1. The Poisson equation is defined in a square $[-1,1]\times[-1,1]$ with a circular interface $r^2 \equiv x^2 + y^2 = \frac{1}{4}$. The analytical solution to the equation, the coefficient β, and the inhomogeneous term of the equation are given as follows

$$u(x,y) = \begin{cases} x^2 + y^2 - 1, & r \leqslant 0.5, \\ \frac{1}{4}\left(1 - \frac{1}{8b} - \frac{1}{b}\right) + \left(\frac{r^4}{2} + r^2\right) \Big/ b, & \text{otherwise}. \end{cases}$$

$$\beta(x,y) = \begin{cases} 2, & r \leqslant 0.5, \\ b, & \text{otherwise}. \end{cases}$$

$$q(x,y) = \begin{cases} 8.0, & r \leqslant 0.5, \\ 8(x^2 + y^2) + 4.0, & \text{otherwise}. \end{cases}$$

By choosing $b = 10$, it can be checked that on the interface $[u] = 1$ and $[\beta u_n] = -0.75$. In particular, it can be found that the analytical solution is a fourth-order polynomial. Therefore, a fourth-order method should render a numerical solution of the machine error. The plot in Fig. 5 exactly illustrates this prediction, where the maximum error is around 10^{-14}.

Example 2. It is generally believed that a high order method usually comes with high resolution, which is absolutely needed in solving Maxwell's equations or the Helmholtz equation for high frequency wave propagation and scattering. To examine the resolution of the proposed method, a Helmholtz-like equation

$$\nabla \cdot (\beta \nabla u(x,y)) + k^2(x,y) u(x,y) = q(x,y)$$

is considered, where $k(x,y) = \kappa\sigma(x,y)$ is the dielectric function describing macroscopically the properties of the medium in which the wave propagates. Both β and k are discontinuous at the interface. The analytical solution to the equation is designed to be highly oscillatory

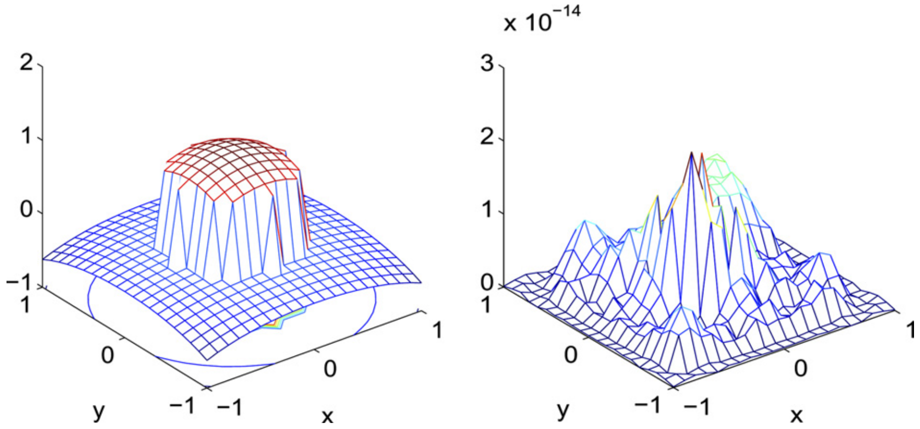

Fig. 5. The computed solution (left) and the error (right) for the Example 1.

$$u(x,y) = \begin{cases} x^2 + y^2, & r \leqslant 0.5, \\ \sin(\kappa x)\cos(\kappa y), & \text{otherwise.} \end{cases}$$

where κ can be tuned to produce solutions of desired frequency. The computational domain and the interface are the same as those in Example 1, while

$$\beta = \begin{cases} 10, & r \leqslant 0.5, \\ 1, & \text{otherwise} \end{cases}$$

and

$$\sigma(x,y) = \begin{cases} 1, & r \leqslant 0.5, \\ \sqrt{10}, & \text{otherwise.} \end{cases}$$

Two different frequencies, $\kappa = 2$ and $\kappa = 12$, are considered as in Table 2 where the solution of the present fourth-order scheme is compared with that of the second-order MIB scheme. It is observed that the order of convergence agrees with the theoretical analysis for both schemes. Also observed is that the low frequency solution can be well approximated by both schemes. However, for the high frequency case, considerable difference can be found in two schemes. In particular, for the low frequency case, a sparse mesh (20×20) is sufficient for both schemes to produce a result of moderate accuracy, i.e., around 10^{-3}. For the high frequency problem, since the solution admits large gradients and is anisotropic, one has to use a very dense mesh (320×320) to resolve the fast variation in the solution. It is noted that by using the fourth-order MIB scheme, an 80×80 mesh can provide a sufficient resolution (see Fig. 6). It is therefore anticipated that with the proposed high-order interface method, significant saving on computing time can be achieved for problems that involve both material interface and high frequency oscillations.

Table 2
Numerical convergence test and accuracy tests for Example 2

$n_x = n_y$	$\kappa = 2$				$\kappa = 12$			
	4th MIB		2nd MIB		4th MIB		2nd MIB	
	L_∞	Order	L_∞	Order	L_∞	Order	L_∞	Order
20	3.91E − 5		3.51E − 3		2.13E − 1		7.80E − 1	
40	1.04E − 5	1.91	1.41E − 3	1.32	2.86E − 2	1.90	1.50E − 1	2.34
80	6.34E − 7	4.04	3.70E − 4	1.93	1.92E − 3	4.55	3.31E − 2	2.18
160	4.37E − 8	3.86	1.01E − 4	1.87	2.04E − 4	4.32	1.06E − 2	1.64
320	2.78E − 9	3.97	2.36E − 5	2.10	8.43E − 6	4.60	1.95E − 3	2.44

Example 3. We consider the Poisson equation with a computational domain $[-1,1]\times[-1,1]$ and an ellipse interface

$$\left(\frac{x}{18/27}\right)^2+\left(\frac{y}{10/27}\right)^2=1.$$

The analytical solution and the coefficient β are given as follows

$$u(x,y)=\begin{cases} e^x\cos(y), & \text{inside } \Gamma,\\ 5\exp\left(-x^2-\frac{y^2}{2}\right), & \text{otherwise.}\end{cases}$$

$$\beta(x,y)=\begin{cases} b, & \text{inside } \Gamma,\\ 1, & \text{otherwise.}\end{cases}$$

Two cases are considered, one with $b=10$ and the other with $b=1000$. The latter shows a strong discontinuity in the coefficient β and demands more iterations in solving the linear system as it is ill-conditioned due to the large jump in β. The lower accuracy for the case with $b=1000$ can be attributed to the larger jump in the coefficient (see Table 3).

To validate the asymptotic behavior of the approximation error with the increasing of the jump in β, a series of β^+ are chosen for numerical tests. The results are collected in Table 4. It can be seen that for moderate magnitude the numerical error is increasing while for large jumps the error is almost a constant (see Fig. 7).

Example 4. This example was introduced in Ref. [25], and is adopted here to examine the flexibility of proposed scheme in dealing with complex interface. The analytical solution to the equation, the coefficient β and a jigsaw puzzle-like interface Γ are given below

$$u(x,y)=\begin{cases} e^x(y^2+x^2\sin(y)), & \text{inside } \Gamma,\\ -(x^2+y^2), & \text{otherwise.}\end{cases}$$

$$\beta(x,y)=\begin{cases} 1, & \text{inside } \Gamma,\\ 10, & \text{otherwise.}\end{cases}$$

$$\Gamma:\begin{cases} x(\theta)=0.6\cos(\theta)-0.3\cos(3\theta),\\ y(\theta)=1.5+0.7\sin(\theta)-0.07\sin(3\theta)+0.2\sin(7\theta).\end{cases}$$

A discretization of the interface is plotted in Fig. 3. The computed solution and the error for a 100×100 mesh are plotted in Fig. 8 and the numerical error in terms of L_2 norm are collected in Table 5, together with the data of the second-order counterpart. We note that the predicted convergence rate for both methods are verified, whereas the fourth-order method gives a much more accurate result. The maximum error occurs at the irregular points near the interface where the local truncation error is one-order lower than that at regular points.

Example 5. This is another standard test case for testing numerical methods designed for solving elliptic interface problems. The interface is parametrized with the polar angle θ, as

Table 3
Numerical convergence test and accuracy tests for Example 3

$n_x=n_y$	$b=10$				$b=1000$			
	4th MIB		2nd MIB		4th MIB		2nd MIB	
	L_∞	Order	L_∞	Order	L_∞	Order	L_∞	Order
40	3.92E − 5		5.21E − 3		6.19E − 3		2.76E − 2	
80	2.92E − 6	3.75	1.49E − 3	1.8	2.65E − 4	4.55	7.52E − 3	1.9
160	1.70E − 7	4.10	3.75E − 4	2.0	1.33E − 5	4.32	2.17E − 3	2.0
320	8.57E − 9	4.31	7.80E − 5	2.3	6.73E − 7	4.30	4.84E − 4	2.2

Table 4
Robustness tests of a high order MIB scheme

β^+	10	20	100	500	10^3	10^4	10^5	10^8
L_∞	3.92E − 5	7.10E − 5	1.89E − 4	3.04E − 4	3.28E − 4	3.53E − 4	3.56E − 4	3.56E − 4

$\beta^- = 1$. $n_x = n_y = 40$.

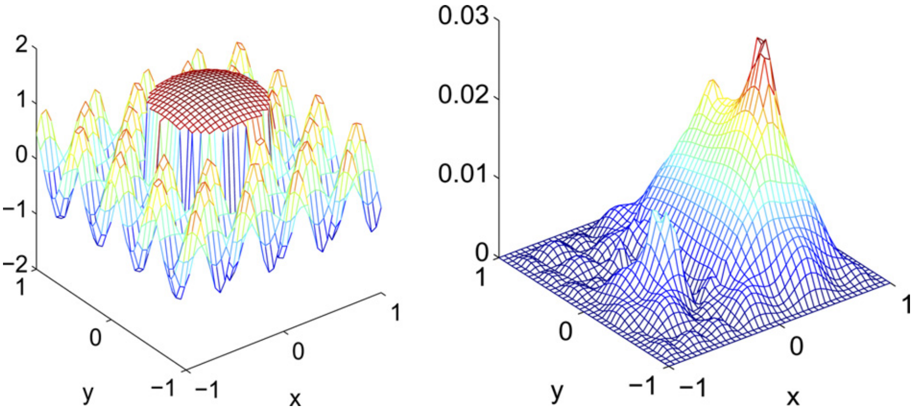

Fig. 6. The computed solution (left) and the error (right) for the Example 2.

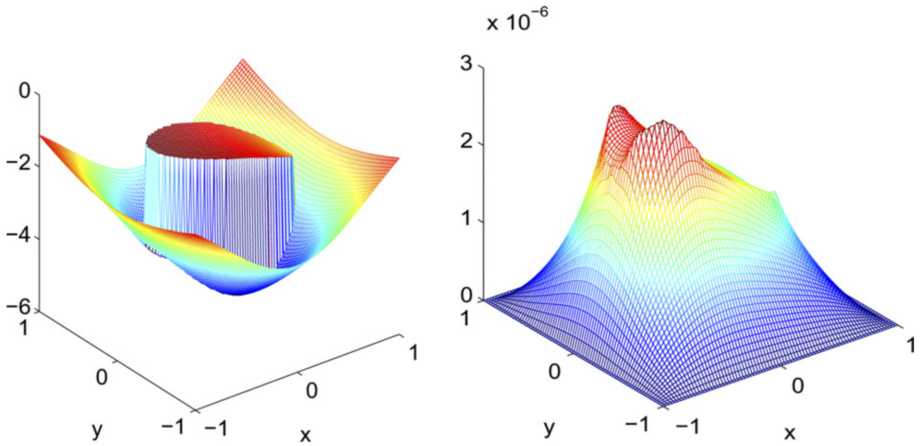

Fig. 7. The computed solution (left) and the error (right) for Example 3. $n_x = n_y = 40$.

$$r = \frac{1}{2} + \frac{\sin(\theta)}{7}.$$

The exact solution to the problem can be arbitrarily designed. Here, we choose

$$u(x,y) = \begin{cases} \exp(x^2+y^2), & \text{inside } \Gamma, \\ 0.1(x^2+y^2)^2 - 0.01\ln(2\sqrt{x^2+y^2}), & \text{otherwise.} \end{cases}$$

$$\beta(x,y) = \begin{cases} 1, & \text{inside } \Gamma, \\ 10, & \text{otherwise.} \end{cases}$$

This solution also prescribes the Dirichlet boundary condition of the problem. The non-homogeneous term of the Poisson equation can also be derived from the exact solution. Fig. 9 plots the computed solution and the

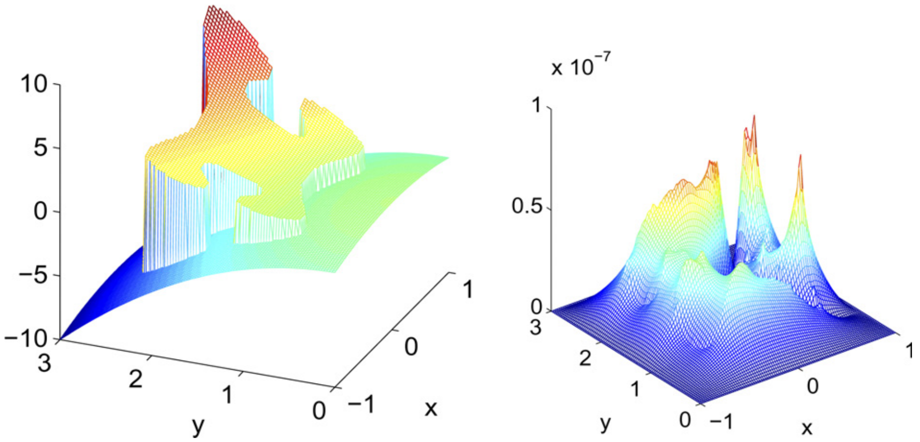

Fig. 8. The computed solution (left) and the error (right) for Example 4. $n_x = n_y = 100$.

Table 5
Numerical convergence test for Example 4

$n_x = n_y$	4th MIB		2nd MIB	
	L_2	Order	L_2	Order
100	2.10E − 8		3.78E − 5	
200	1.54E − 9	3.77	1.06E − 5	1.83
400	5.11E − 11	3.95	2.49E − 6	2.09

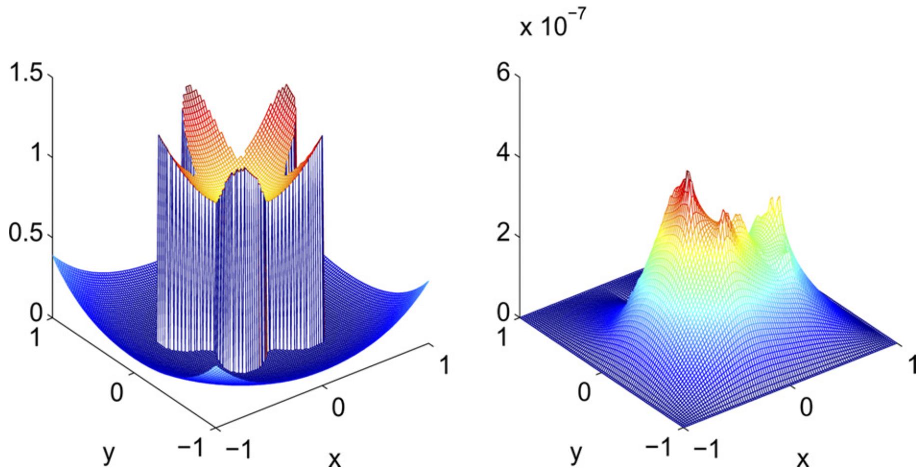

Fig. 9. The computed solution (left) and the error (right) for Example 5. $n_x = n_y = 100$.

Table 6
Numerical convergence test for Example 5

$n_x = n_y$	4th MIB		2nd MIB	
	L_2	Order	L_2	Order
100	1.29E − 7		5.03E − 5	
200	1.17E − 8	3.46	1.35E − 5	1.90
400	7.09E − 10	4.04	3.41E − 6	1.99

error with a mesh of 100×100. Table 6 shows the results of the numerical accuracy tests on three successively refined meshes, in comparing with the second-order MIB method.

5. Conclusion

As a part of our effort to construct high order numerical methods for solving elliptic equations with discontinuous coefficients and singular sources, we introduce a new concept to disassociate the domain extension from the discretization in the matched interface and boundary (MIB) method [34,35]. The MIB is a systematic approach to smoothly extend the solution across the interface by enforcing the lowest order interface jump conditions. The standard high order central finite different discretization is then employed in the whole domain without the loss of accuracy. The MIB method is of arbitrarily high order in principle as the jump conditions are iteratively used. However, a domain extension practice, i.e., the association of the discretization and the domain extension, used in the previous MIB method [35] prevents it from attaining high order convergence for interfaces involving large curvatures. The present work overcomes this difficulty by allowing the extended domain to be utilized for discretization in arbitrary directions regardless how the domain was extended. We show that the improved MIB method is truly high order for general interface geometry by extensive numerical experiments. Some first known results of fourth-order convergence are observed for a set of benchmark test examples involving irregular geometry. The robustness of the proposed method against large jumps in the coefficient amplitude is analyzed and validated. To improve the understanding of the MIB and other potential high order methods for elliptic interface problems, we propose an alternative interpolation formulation of the MIB method. We show that the new formulation is essentially equivalent to the improved MIB method. The generalization of the MIB to three dimensions and the construction of fast algebraic solvers are under our consideration.

Acknowledgements

This work was supported in part by NSF Grant IIS-0430987 and IRGP Grant 71-4834. Zhou acknowledges the Michigan State University for a Quantitative Biology Interdisciplinary Research Award and for a Paul and Wilma Dressel Scholarship.

References

[1] L. Adams, Z.L. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2002) 463–479.
[2] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (1970) 207–213.
[3] P.A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (2004) 364–386.
[4] G. Biros, L.X. Ying, D. Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (2004) 317–348.
[5] J. Bramble, J. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math. 6 (1996) 109–138.
[6] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202.
[7] S.Z. Deng, K. Ito, Z.L. Li, Three-dimensional elliptic solvers for interface problems and applications, J. Comput. Phys. 184 (2003) 215–243.
[8] M.A. Dumett, J.P. Keener, An immersed interface method for solving anisotropic elliptic boundary value problems in three dimensions, SIAM J. Sci. Comput. 25 (2003) 348–367.
[9] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 30–60.
[10] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[11] A.L. Fogelson, J.P. Keener, Immersed interface methods for Neumann and related problems in two and three dimensions, SIAM J. Sci. Comput. 22 (2000) 1630–1654.
[12] B. Fornberg, Calculation of weights in finite difference formulas, SIAM Rev. 40 (1998) 685–691.
[13] M. Francois, W. Shyy, Computations of drop dynamics with the immersed boundary method, part 2: Drop impact and heat transfer, Numer. Heat Trans. Part B-Fund. 44 (2003) 119–143.

[14] F. Gibou, R.P. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.
[15] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.
[16] G. Guyomarch, C.-O. Lee, A discontinuous Galerkin method for elliptic interface problems with application to electroporation, AIST DAM Research Report (2004) 04–14.
[17] S. Hou, X.-D. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, J. Comput. Phys. 202 (2005) 411–445.
[18] T.Y. Hou, Z.L. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the Hele–Shaw flow, J. Comput. Phys. 134 (1997) 236–252.
[19] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (2003) 331–347.
[20] S. Jin, X.L. Wang, Robust numerical simulation of porosity evolution in chemical vapor infiltration II. Two-dimensional anisotropic fronts, J. Comput. Phys. 179 (2002) 557–577.
[21] H. Johansen, P. Colella, A Cartesian grid embedding boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60–85.
[22] J.D. Kandilarov, Immersed interface method for a reaction-diffusion equation with a moving own concentrated source, Lect. Notes Comput. Sci. 2542 (2003) 506–513.
[23] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.
[24] R.J. LeVeque, Z.L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[25] Z.L. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1998) 230–254.
[26] Z.L. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM J. Sci. Comput. 23 (2001) 339–361.
[27] A. Mayo, A. Greengard, Fast parallel solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Sci. Stat. Comput. 13 (1992) 101–118.
[28] C.S. Peskin, Numerical analysis of blood flow in heart, J. Comput. Phys. 25 (1977) 220–252.
[29] C.S. Peskin, Lectures on mathematical aspects of physiology, Lect. Appl. Math. 19 (1981) 69–107.
[30] C.S. Peskin, D.M. Mcqueen, A3-dimensional computational method for blood-flow in the heart. 1. Immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (1989) 372–405.
[31] J.A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, J. Comput. Phys. 163 (2000) 489–528.
[32] A.K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2004) 462–488.
[33] J.H. Walther, G. Morgenthal, An immersed interface method for the vortex-in-cell algorithm, J. Turbulence 3 (039) (2002).
[34] S. Zhao, G.W. Wei, High order FDTD methods via derivative matching for Maxwell's equations with material interfaces, J. Comput. Phys. 200 (2004) 60–103.
[35] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (2006) 1–30.