Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 197 (2004) 364–386
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions

Petter Andreas Berthelsen *

The Fluids Engineering Group, Department of Energy and Process Engineering, Norwegian University of Science and Technology, Kolbjørn Hejes vei 2, N-7491 Trondheim, Norway

Received 16 September 2003; received in revised form 3 December 2003; accepted 3 December 2003
Available online 22 January 2004

Abstract

A second order accurate finite difference method is presented for solving two-dimensional variable coefficient elliptic equations on Cartesian grids, in which the coefficients, the source term, the solution and its derivatives may be non-smooth or discontinuous across an interface. A correction term is introduced to the standard central difference stencil so that the numerical discretization is well-defined across the interface. We also propose a new method to approximate the correction term as part of the iterative procedure. The method is easy to implement since the correction term only needs to be added to the right-hand-side of the system. Therefore, the coefficient matrix remains symmetric and diagonally dominant, allowing for most standard solvers to be used. Numerical examples show good agreements with exact solutions, and the order of accuracy is comparable with other immersed interface methods.

PACS: 02.70.Bf; 02.60.Cb; 02.60.Lj

AMS: 65N06; 76T05; 35J05

Keywords: Immersed interface method; Discontinuous coefficients; Irregular domain; Elliptic interface problem; Finite difference method; Cartesian grid; Level set method

1. Introduction

Elliptic problems with discontinuous coefficients and singular sources are often encountered in fluid dynamics and material science. These interface problems usually lead to non-smooth or discontinuous

* Tel.: +4773595348; fax: +4773593491.
E-mail address: petterab@mtf.ntnu.no (P.A. Berthelsen).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2003.12.003

solutions across an interface. Traditional Cartesian finite difference methods work poorly for these problems since the numerical discretization is not well-defined across the interface.

In Peskin's [15] immersed boundary method (IB), originally developed to model blood flow in the heart, singular forces are smeared out by a discrete delta function. The idea has been extended and applied to solve a number of interface-related problems. For example, in [3,4,16,17] the surface tension effect was introduced as a new, smooth forcing term in the momentum equation leading to continuity in pressure. Material properties may also be smeared out (e.g. using a level set function [4,16]) removing discontinuities across the interface, making the solution continuous, smooth and suitable for standard finite difference schemes. The IB is widely used because of its robustness, and it can easily be implement into existing CFD codes, even in multiple spatial dimensions. However, the numerical smearing makes the method not very accurate and unable to properly produce discontinuities.

With the weakness of IB in mind, several new techniques classified as sharp interface methods have been developed. The immersed interface method (IIM), as presented in [8,9], handles two- and three-dimensional interface problems based on the analysis of [2]. The IIM is second order accurate and includes the interfacial boundary conditions into the finite difference discretization in such a way that it preserves the jumps in the solution and its derivatives. In the original IIM this was done by adding additional nodes to the numerical stencil, leading to a non-symmetric coefficient matrix. This non-symmetric matrix reduces the numbers of efficient numerical solvers to be used and convergence is not always guaranteed. In fact, the method has only been shown to be stable for one-dimensional problems and for two-dimensional problems with piecewise constant coefficients [6].

To avoid this convergence problem, a new version of the IIM was proposed in [12]. A maximum principle preserving method is enforced to obtain a diagonally dominant linear system. This way, some iterative methods are guaranteed to converge. The maximum principle approach was successfully implemented with a specially designed multigrid method in [1].

Another sharp interface technique is the ghost fluid method (GFM) introduced in [5] to treat two-phase contact discontinuities in the Euler equations. The basic principle behind GFM is to extend values across the interface into an artificial fluid (ghost fluid) inducing the proper conditions at the interface.

The GFM concept was extended in [13] to solve elliptic equations with variable coefficients. One of the main objectives with their approach was to simplify the IIM and still obtain a sharp solution at the interface. In contrast to IIM, the jump conditions are incorporated into the numerical discretization such that the symmetry of the finite difference stencil is kept. This allows for most standard solvers to be used. The method decomposes the flux jumps in each axis direction treating the problem dimension by dimension. This extended GFM is only first order accurate. The method has been applied to multiphase incompressible flow in [7].

Decomposing the jump conditions into each axis direction was also done in [11,18] using the IIM where the coefficients are piecewise constant. The approach in [18] produces a symmetric problem. Instead of focusing on finding new coefficients for the finite difference scheme they focus on the jumps in the solution and its derivatives. If the jumps $[u]$, $[u_x]$, $[u_{xx}]$, $[u_y]$ and $[u_{yy}]$ are all known, then the standard finite difference discretization can be used with some correction terms to adjust for the discontinuities. They also consider variable coefficients by rewriting the partial differential equation into a more convenient form for their method. The method shows to be second order accurate.

The intention of this paper is to extend the ideas of [11,13,18] to derive a second order sharp interface method capable of solving elliptic interface problems with variable coefficients in two dimensions. The primary objective is to keep the standard finite difference stencil, making only corrections to the right-hand side of the problem. This way we will keep the linear system symmetric and diagonally dominant. We give a more formal derivation of the finite difference scheme than found in [13], and the order of accuracy is improved by including more jump conditions. The main difference from [11,18] is that we also consider the case with variable coefficients when deriving the correction term. We also propose a simple technique for

approximating the solution-dependent jump conditions as part of the iterative method. A level set function [14] is used to represent the interface because of its simplicity and strength in describing fairly complex shapes.

The rest of the paper is organized as follows: In Section 2 we present the mathematical equations to be solved, introduce the level set function and decompose the jump conditions. Then, in Section 3 we describe the numerical discretization for our sharp interface method and suggest an approach to estimate the interfacial jumps. Numerical examples are presented in Section 4 before we conclude with a summary in Section 5.

2. Mathematical formulation

2.1. The elliptic equation

Consider a domain Ω divided into two (or more) separate subdomains Ω^+ and Ω^- by a lower dimensional interface Γ. The two dimensional variable coefficient elliptic equation is given as

$$(\beta u_x)_x + (\beta u_y)_y = f(x,y), \qquad (x,y) \in \Omega, \tag{1}$$

with Dirichlet boundary conditions

$$u(x,y) = g(x,y), \qquad (x,y) \in \delta\Omega,$$

where $\delta\Omega$ is the exterior boundary and (x,y) are the spatial coordinates. The coefficient $\beta(x,y)$ and source term $f(x,y)$ are continuous and smooth on each subdomain, but may have jumps across the interface, i.e.

$$\beta(x,y) = \begin{cases} \beta^+(x,y), & (x,y) \in \Omega^+, \\ \beta^-(x,y), & (x,y) \in \Omega^-, \end{cases}$$

and

$$f(x,y) = \begin{cases} f^+(x,y), & (x,y) \in \Omega^+, \\ f^-(x,y), & (x,y) \in \Omega^-. \end{cases}$$

Discontinuities in the coefficient $\beta(x,y)$ and the source term $f(x,y)$ may make the solution and its derivatives discontinuous and non-smooth at the interface. These jumps in solution and its derivatives can be specified as jump conditions along the interface, i.e.

$$[u] = w(x,y), \qquad (x,y) \in \Gamma, \tag{2}$$

$$[\beta u_n] = v(x,y), \qquad (x,y) \in \Gamma, \tag{3}$$

where $u_n = \partial u/\partial n = \nabla u \cdot \vec{n}$ is the normal derivative of u, $\vec{n}$ is the local unit normal vector to the interface pointing towards the Ω^+-region, and the jumps are defined as the limiting values

$$[u] = \lim_{(x,y)\to\Gamma^+} u(x,y) - \lim_{(x,y)\to\Gamma^-} u(x,y) = u^+ - u^-,$$

$$[\beta u_n] = \lim_{(x,y)\to\Gamma^+} \beta(x,y)u_n(x,y) - \lim_{(x,y)\to\Gamma^-} \beta(x,y)u_n(x,y) = \beta^+ u_n^+ - \beta^- u_n^-.$$

Here $(x,y) \to \Gamma^+$ means approaching the interface from the Ω^+ side and $(x,y) \to \Gamma^-$ from the Ω^- side.

2.2. Level set representation

The main idea of the level set method is to introduce a smooth auxiliary function $\phi(x,y)$ defined as

$$\phi(x,y) = \pm d,$$

where d is the shortest distance to the interface. The sign of ϕ indicates whether (x,y) is in the Ω^+-region (positive) or in the Ω^--region (negative). It will be evident from the definition above that the interface Γ is given by the zero level set of the function ϕ,

$$\Gamma = \{(x,y) \in \mathbb{R}^2 \mid \phi(x,y) = 0\}.$$

The normal vector can easily be deduced from ϕ and at any point is given as $\vec{n} = \nabla\phi/|\nabla\phi|$.

The interface appears as a closed curve in two dimensions. We assume the interface to be infinitely thin so that our problem domain can be defined as (see Fig. 1)

$$\Omega = \begin{cases} \Omega^-, & \phi < 0, \\ \Omega^+, & \phi \geqslant 0. \end{cases} \tag{4}$$

2.3. Decomposing the jump conditions

Since the jump conditions usually are defined in normal or tangential direction of the interface we define a local coordinate system aligned with the interface at (x^*, y^*),

$$\xi = (x - x^*)\cos\theta + (y - y^*)\sin\theta,$$
$$\eta = -(x - x^*)\sin\theta + (y - y^*)\cos\theta,$$

where θ is the angle between the x- and ξ-axis. The ξ-axis is normal to the interface while the η-axis is tangential.

Transforming these jump conditions into Cartesian coordinates yields

$$[\beta u_x] = [\beta u_\xi]\cos\theta - [\beta u_\eta]\sin\theta,$$
$$[\beta u_y] = [\beta u_\xi]\sin\theta + [\beta u_\eta]\cos\theta,$$

and differentiating using the chain rule gives

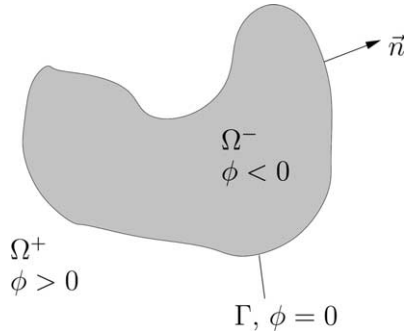

Fig. 1. An irregular interface Γ dividing the domain Ω into two subdomains Ω^+ and Ω^- with normal vector $\vec{n}$ pointing towards the Ω^+-region.

$$[(\beta u_x)_x] = [(\beta u_\xi)_\xi]\cos^2\theta - ([(\beta u_\xi)_\eta] + [(\beta u_\eta)_\xi])\cos\theta\sin\theta + [(\beta u_\eta)_\eta]\sin^2\theta,$$
$$[(\beta u_y)_y] = [(\beta u_\xi)_\xi]\sin^2\theta + ([(\beta u_\xi)_\eta] + [(\beta u_\eta)_\xi])\cos\theta\sin\theta + [(\beta u_\eta)_\eta]\cos^2\theta.$$

Similarly, differentiation of the jump $[u]$ gives

$$[u_x] = [u_\xi]\cos\theta - [u_\eta]\sin\theta,$$
$$[u_y] = [u_\xi]\sin\theta + [u_\eta]\cos\theta,$$

and

$$[u_{xx}] = [u_{\xi\xi}]\cos^2\theta - 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\sin^2\theta,$$
$$[u_{yy}] = [u_{\xi\xi}]\sin^2\theta + 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\cos^2\theta.$$

For now we will assume that the jumps $[u]$, $[u_\xi]$, $[u_{\xi\xi}]$, $[u_\eta]$, $[u_{\eta\eta}]$, $[\beta u_\xi]$, $[(\beta u_\xi)_\xi]$, $[\beta u_\eta]$, $[(\beta u_\eta)_\eta]$, $[(\beta u_\xi)_\eta]$ and $[(\beta u_\eta)_\xi]$ are all known. In this way, we can apply a dimension by dimension approach of the numerical method.

3. Numerical method

3.1. Discretization

For simplicity, we use a uniform, rectangular grid, $[a,b]\times[c,d]$, describing the computational domain Ω. The grid nodes are equally spaced, $h = (b-a)/M = (d-c)/N$, where M and N are the number of grid points in x- and y-direction, respectively. The grid coordinates are defined as

$$x_i = a + \mathrm{i}h,\ y_j = c + \mathrm{j}h \quad \text{for } 0 \leqslant i \leqslant M,\ 0 \leqslant j \leqslant N,$$

and the numerical approximation of u at (x_i, y_j) is written as $U_{i,j}$.

We define a grid node (x_i, y_j) as regular if all neighbouring nodes are on the same side of the interface. On the contrary, a grid node (x_i, y_j) is irregular if at least one adjacent node is on the other side of the interface, i.e. the interface cuts one of the grid lines between the nodes.

The elliptic equation (1) is approximated with the following discretization

$$\mathscr{L}_h^\beta U_{i,j} = f_{i,j} - C_{i,j}, \tag{5}$$

where $\mathscr{L}_h^\beta U_{i,j}$ is the standard five point variable coefficient central difference scheme

$$\mathscr{L}_h^\beta U_{i,j} = \frac{\beta_{i+1/2,j}(U_{i+1,j} - U_{i,j}) - \beta_{i-1/2,j}(U_{i,j} - U_{i-1,j})}{h^2} + \frac{\beta_{i,j+1/2}(U_{i,j+1} - U_{i,j}) - \beta_{i,j-1/2}(U_{i,j} - U_{i,j-1})}{h^2}$$

and $\beta_{i+1/2,j}$ denotes $\beta(x_{i+1/2,j}, y_j)$, and so on. At regular nodes, $\mathscr{L}_h^\beta U_{i,j}$ yields a second order accurate approximation of the second derivatives, i.e.

$$(\beta u_x)_x + (\beta u_y)_y = \mathscr{L}_h^\beta U_{i,j} + \mathscr{O}(h^2).$$

The correction term $C_{i,j}$ is introduced to make the numerical discretization (5) well-defined at irregular nodes and should vanish at regular nodes. In the remaining part of this section we will discuss how to find this correction term.

3.2. Determining the correction term

As shown in Section 2.3, the jump conditions can be decomposed into jumps in the x- and y-directions allowing for a dimension by dimension approach. This way, the correction term $C_{i,j}$ is made up by a component in x-direction and one in y-direction, i.e.

$$C_{i,j} = C^x_{i,j} + C^y_{i,j} \tag{6}$$

The procedure for obtaining the componentwise correction term is equivalent whether we consider the x-direction or y-direction. Therefore, we shall only discuss how to find the correction in x-direction. For simpler notation, we will neglect the subscript j in the notation below, reducing the derivations to a one-dimensional problem.

We wish to find a correction term so that the standard finite difference approximation of $(\beta u_x)_x$ is valid even at the interface. We consider an irregular grid node x_i where the interface is located at $x^* = x_i + ah$, $0 \leqslant a \leqslant 1$ and $a = \phi_i/(\phi_i - \phi_{i+1})$. The correction can be derived in two steps. First, we need to correct the numerical discretization of u_x, then, if necessary, the approximation of the second derivative $(\beta u_x)_x$ also needs to be corrected.

The first derivative is estimated at the centre between x_i and x_{i+1}. The correction of this approximation depends on what side of $x_{i+1/2}$ the interface is located. In other words, using the definitions of Eq. (4), if $\phi_i < 0$ and $0 < a \leqslant 1/2$ then the interface is to the left of $x_{i+1/2}$, i.e. $\{x_{i+1/2}, x_{i+1}\} \in \Omega^+$. Else if $1/2 < a \leqslant 1$ then $\{x_i, x_{i+1/2}\} \in \Omega^-$. Otherwise, if $\phi_i \geqslant 0$, the subregions switches and instead we have if $0 \leqslant a < 1/2$ then $\{x_{i+1/2}, x_{i+1}\} \in \Omega^-$ or if $1/2 \leqslant a < 1$ then $\{x_i, x_{i+1/2}\} \in \Omega^+$.

Following the same approach as [11] using Taylor expansion for $u(x_{i+1})$ at $x^* = x_i + ah$ for the case $\phi_i < 0$ and $1/2 < a \leqslant 1$ yields

$$\begin{aligned}
u(x_{i+1}) &= u(x^* + (1-a)h) = u^+ + u^+_x(1-a)h + \tfrac{1}{2}u^+_{xx}(1-a)^2h^2 + \mathcal{O}(h^3) \\
&= u^- + u^-_x(1-a)h + \tfrac{1}{2}u^-_{xx}(1-a)^2h^2 + C_1(x,a) + \mathcal{O}(h^3) \\
&= u(x_{i+1/2}) + u_x(x_{i+1/2})(a-\tfrac{1}{2})h + \tfrac{1}{2}u_{xx}(x_{i+1/2})(a-\tfrac{1}{2})^2h^2 + u_x(x_{i+1/2})(1-a)h \\
&\quad + \tfrac{1}{2}u_{xx}(x_{i+1/2})(2a-1)(1-a)h^2 + \tfrac{1}{2}u_{xx}(x_{i+1/2})(1-a)^2h^2 + C_1(x,a) + \mathcal{O}(h^3) \\
&= u(x_{i+1/2}) + u_x(x_{i+1/2})\frac{h}{2} + \tfrac{1}{2}u_{xx}(x_{i+1/2})\left(\frac{h}{2}\right)^2 + C_1(x,a) + \mathcal{O}(h^3),
\end{aligned} \tag{7}$$

and Taylor expansion for $u(x_i)$ at $x_{i+1/2}$ becomes

$$u(x_i) = u(x_{i+1/2}) - u_x(x_{i+1/2})\frac{h}{2} + \tfrac{1}{2}u_{xx}(x_{i+1/2})\left(\frac{h}{2}\right)^2 + \mathcal{O}(h^3), \tag{8}$$

where the correction term is given as

$$C_1(x,a) = [u] + [u_x](1-a)h + \tfrac{1}{2}[u_{xx}](1-a)^2h^2.$$

Subtracting Eq. (8) from (7) and rearranging gives

$$u_x(x_{i+1/2}) = \frac{u(x_{i+1}) - u(x_i)}{h} - \frac{C_1(x,a)}{h} + \mathcal{O}(h^2).$$

Or, if $0 < a \leqslant 1/2$, we Taylor expand $u(x_i)$ at $x^* = x_i + ah$:

$$
\begin{aligned}
u(x_i) &= u(x^* - ah) = u^- - u_x^- ah + \tfrac{1}{2}u_{xx}^- a^2h^2 + \mathcal{O}(h^3) \\
&= u^+ - u_x^+ ah + \tfrac{1}{2}u_{xx}^+ a^2h^2 - C_1(x,a) + \mathcal{O}(h^3) = u(x_{i+1/2}) - u_x(x_{i+1/2})(\tfrac{1}{2} - a)h \\
&\quad + \tfrac{1}{2}u_{xx}(x_{i+1/2})(\tfrac{1}{2} - a)^2h^2 - u_x(x_{i+1/2})ah + \tfrac{1}{2}u_{xx}(x_{i+1/2})(1-2a)ah^2 \\
&\quad + \tfrac{1}{2}u_{xx}(x_{i+1/2})a^2h^2 - C_1(x,a) + \mathcal{O}(h^3) \\
&= u(x_{i+1/2}) - u_x(x_{i+1/2})\frac{h}{2} + \tfrac{1}{2}u_{xx}(x_{i+1/2})\left(\frac{h}{2}\right)^2 - C_1(x,a) + \mathcal{O}(h^3)
\end{aligned} \tag{9}
$$

and the Taylor expansion for $u(x_{i+1})$ at $x_{i+1/2}$ will be

$$
u(x_{i+1}) = u(x_{i+1/2}) + u_x(x_{i+1/2})\frac{h}{2} + \tfrac{1}{2}u_{xx}(x_{i+1/2})\left(\frac{h}{2}\right)^2 + \mathcal{O}(h^3), \tag{10}
$$

where

$$
C_1(x,a) = [u] - [u_x]ah + \tfrac{1}{2}[u_{xx}]a^2h^2.
$$

Again, subtracting Eq. (9) from (10) and rearranging gives

$$
u_x(x_{i+1/2}) = \frac{u(x_{i+1}) - u(x_i)}{h} - \frac{C_1(x,a)}{h} + \mathcal{O}(h^2).
$$

In the second step, if the flux βu_x is non-smooth or discontinuous at the interface and $0 < a \leqslant 1/2$, then we need a second correction term, C_2, for the second derivative. Expanding gives

$$
\begin{aligned}
\beta u_x(x_{i+1/2}) &= \beta u_x(x^* + (\tfrac{1}{2} - a)h) = (\beta u_x)^+ + (\beta u_x)_x^+(\tfrac{1}{2} - a)h + \mathcal{O}(h^2) \\
&= (\beta u_x)^- + (\beta u_x)_x^-(\tfrac{1}{2} - a)h + C_2(x,a) + \mathcal{O}(h^2) \\
&= \beta u_x(x_i) + (\beta u_x)_x(x_i)ah + (\beta u_x)_x(x_i)(\tfrac{1}{2} - a)h^2 + C_2(x,a) + \mathcal{O}(h^2) \\
&= \beta u_x(x_i) + (\beta u_x)_x(x_i)\frac{h}{2} + C_2(x,a) + \mathcal{O}(h^2)
\end{aligned} \tag{11}
$$

and similarly at $x_{i-1/2}$

$$
\beta u_x(x_{i-1/2}) = \beta u_x(x_i) - (\beta u_x)_x(x_i)\frac{h}{2} + \mathcal{O}(h^2), \tag{12}
$$

where

$$
C_2(x,a) = [\beta u_x] + \tfrac{1}{2}[(\beta u_x)_x](1-2a)h.
$$

Subtracting Eq. (12) from (11) and rearranging gives

$$
(\beta u_x)_x(x_i) = \frac{\beta u_x(x_{i+1/2}) - \beta u_x(x_{i-1/2})}{h} - \frac{C_2(x,a)}{h} + \mathcal{O}(h).
$$

To summarize, if we extend this approach to consider the case when the interface is located anywhere between x_{i-1} and x_{i+1}, and we replace u with the numerical approximation U, then we can write

$$
(\beta u_x)_x(x_i) = \frac{\beta_{i+1/2}(U_{i+1} - U_i) - \beta_{i-1/2}(U_i - U_{i-1})}{h^2} + C_i + \mathcal{O}(h), \tag{13}
$$

with the correction term

$$C_i = S_\phi \bar{\beta} \frac{C_1(x,a)}{h^2} + S_\phi \frac{C_2(x,a)}{h}, \tag{14}$$

where

$$S_\phi = \begin{cases} -1, & \phi_i < 0, \\ 1, & \phi_i \geqslant 0, \end{cases}$$

and

$$C_1(x,a) = \begin{cases} [u] - \lambda[u_x]ah + \frac{1}{2}[u_{xx}]a^2h^2, & \text{if } (\phi_i \geqslant 0 \text{ and } 0 \leqslant a < 1/2) \\ & \text{or } (\phi_i < 0 \text{ and } 0 < a \leqslant 1/2), \\ [u] + \lambda[u_x](1-a)h + \frac{1}{2}[u_{xx}](1-a)^2h^2, & \text{if } (\phi_i \geqslant 0 \text{ and } 0 \leqslant a < 1/2) \\ & \text{or } (\phi_i < 0 \text{ and } 0 < a \leqslant 1,/2), \end{cases}$$

$$C_2(x,a) = \begin{cases} \lambda[\beta u_x] + \frac{1}{2}[(\beta u_x)_x](1-2a)h, & \text{if } (\phi_i \geqslant 0 \text{ and } 0 \leqslant a < 1/2) \\ & \text{or } (\phi_i < 0 \text{ and } 0 < a \leqslant 1/2), \\ 0, & \text{otherwise.} \end{cases}$$

The parameters λ, a and $\bar{\beta}$ are defined as follows:

- If the interface is between x_i and x_{i+1}
  (i.e. $\phi_i \cdot \phi_{i+1} < 0$, or $\phi_i = 0$ and $\phi_{i+1} < 0$, or $\phi_i < 0$ and $\phi_{i+1} = 0$)

  $\lambda = 1, a = \phi_i/(\phi_i - \phi_{i+1})$ and $\bar{\beta} = \beta_{i+1/2}$.

- If the interface is between x_{i-1} and x_i
  (i.e. $\phi_i \cdot \phi_{i-1} < 0$, or $\phi_i = 0$ and $\phi_{i-1} < 0$, or $\phi_i < 0$ and $\phi_{i-1} = 0$)

  $\lambda = -1, a = \phi_i/(\phi_i - \phi_{i-1})$ and $\bar{\beta} = \beta_{i-1/2}$.

For both cases, we can define the jump of a function f as

$$[f] = \lim_{\phi^+ \to 0} f(x) - \lim_{\phi^- \to 0} f(x) = f^+ - f^-.$$

Remark 1. If the solution is smooth and continuous (regular nodes), the jumps become zero and the correction term vanish.

Remark 2. The local truncation error in Eq. (13) is of first order only, but we can still expect the global accuracy to approach second order since the interface is of one dimension lower than the problem.

Returning to the two-dimensional case, the x-component in Eq. (6) can be replaced by Eq. (14) and a similar expression can be found for $C^y_{i,j}$ making Eq. (5) well-defined at all nodes.

3.3. Approximating the jump conditions

So far we have assumed the interface jump conditions to be known. Unfortunately, that is rarely the case as they are usually solution-dependent and must be obtained as part of the solution. This is probably the most difficult task when using sharp interface methods, as it is crucial for both solving the discrete equations efficiently and maintaining a certain accuracy.

In higher dimensional problems it is common to use some iterative methods to solve the discretized equations. The idea is therefore to obtain the correction term iteratively as well. We can approximate the solution at the interface using Eqs. (2) and (3) by interpolating the result from the previous iteration step (or initial guess). Then using one-sided difference stencils on both sides of the interface we can approximate the jump conditions. The correction term is updated and given explicitly at every iteration step, as it converges to the correct solution.

Using low-order interpolation schemes to approximate the jumps may increase the local truncation error. To keep the local first order accuracy at irregular nodes we see from Eq. (14) that it is necessary to use at least a third order accurate interpolation technique when guessing the interfacial values. To provide a sufficient approximation of the jumps in first and second derivatives, second and first order methods are needed, respectively. We found the Lagrange polynomial of degree two,

$$P(x) = \sum_{i=0}^{2} \left(\prod_{\substack{j=0 \\ j\neq i}} \frac{(x - x_j)}{(x_i - x_j)} f_i \right), \tag{15}$$

to be sufficient to meet our required accuracy.

The first step in finding the correction term is approximating the solution at the interface. Consider an irregular grid node (x_i, y_j) where the interface crosses the horizontal grid line at (x^*, y^*), i.e. $y^* = y_j$. Using a set of interpolated values on both sides of the interface, we can reconstruct the boundary conditions (2) and (3) to solve for the interfacial values at (x^*, y^*). The procedure is as follows (see Fig. 2):

(1) Construct a line which is normal to the interface at (x^*, y^*) using the level set function, e.g find the normal vector $\vec{n} = \nabla\phi/|\nabla\phi|$ in the neighbour gridpoints of (x^*, y^*), and use linear interpolation to estimate $\vec{n}$ at (x^*, y^*).
(2) Chose two points along the normal line on both sides of the interface, e.g. P_1^+, P_2^+, P_1^- and P_2^-, with the distances a_1, a_2, b_1 and b_2 from the interface.
(3) Use the nodes surrounding P_1^+ and the Lagrange polynomial to find a new value U_1^+ in P_1^+. Repeat for U_2^+, U_1^- and U_2^- in P_2^+, P_1^- and P_2^-. The points, P_1^+, P_2^+, P_1^- and P_2^-, must be chosen carefully so that all surrounding nodes lie on the same side of the interface, e.g. let $a_1 = \sqrt{2}h$ and $a_2 = 2\sqrt{2}h$, likewise on the other side of the interface.
(4) Differentiate Eq. (15) to estimate the normal fluxes at the interface as

$$\beta^+ u_n^+ = \beta^+ \left(-\frac{a_1 + a_2}{a_1 a_2} U_0^+ + \frac{a_2}{a_1(a_2 - a_1)} U_1^+ - \frac{a_1}{a_2(a_2 - a_1)} U_2^+ \right),$$
$$\beta^- u_n^- = \beta^- \left(\frac{b_1 + b_2}{b_1 b_2} U_0^- - \frac{b_2}{b_1(b_2 - b_1)} U_1^- + \frac{b_1}{b_2(b_2 - b_1)} U_2^- \right).$$

Combine this with Eqs. (2) and (3) to obtain the interfacial values as

$$U_0^- = \frac{\beta^- \frac{a_1 a_2}{b_2 - b_1} \left(b_2^2 U_1^- - b_1^2 U_2^- \right) + \beta^+ \frac{b_1 b_2}{a_2 - a_1} \left(a_2^2 U_1^+ - a_1^2 U_2^+ \right)}{\beta^- a_1 a_2 (b_1 + b_2) + \beta^+ b_1 b_2 (a_1 + a_2)}$$
$$- \frac{a_1 a_2 b_1 b_2 v(x^*, y^*) + \beta^+ b_1 b_2 (a_1 + a_2) w(x^*, y^*)}{\beta^- a_1 a_2 (b_1 + b_2) + \beta^+ b_1 b_2 (a_1 + a_2)} \tag{16}$$

and

$$U_0^+ = U_0^- + w(x^*, y^*). \tag{17}$$

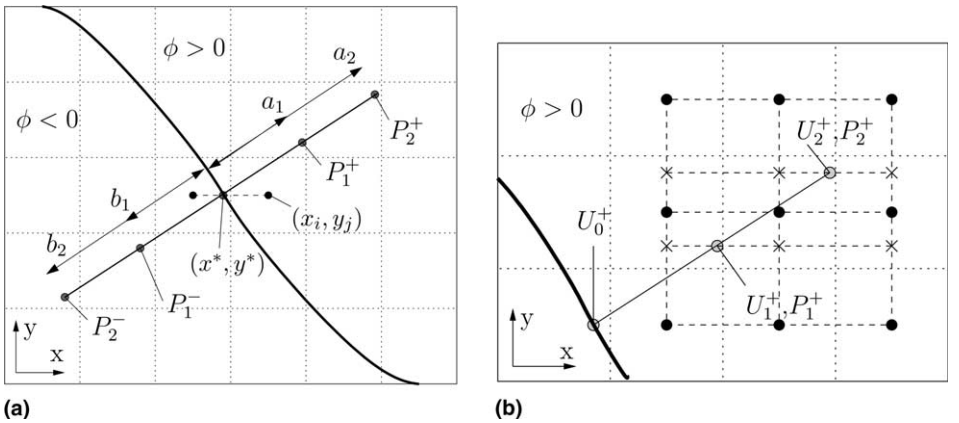

Fig. 2. (a) The interpolation stencil for finding the values at the interface when $y^* = y_j$. (b) The surrounding nodes for the Lagrange polynomial interpolation. First, interpolate in y-direction, then in x-direction to find U_1^+ and U_2^+.

Next, we use the interfacial values to approximate the jumps in Eq. (6) along the x-direction where $a = \phi_{i,j}/(\phi_{i,j} - \phi_{i-1,j})$ and $b = 1 - a$,

$$u_x^+ = \frac{1}{h}\left(-\frac{2a+1}{a(a+1)}U_0^+ + \frac{a+1}{a}U_{i,j} - \frac{a}{a+1}U_{i+1,j}\right),$$

$$u_x^- = \frac{1}{h}\left(\frac{2b+1}{b(b+1)}U_0^- - \frac{b+1}{b}U_{i-1,j} + \frac{b}{b+1}U_{i-2,j}\right),$$

$$u_{xx}^+ = \frac{1}{h^2}\left(\frac{2}{a(a+1)}U_0^+ - \frac{2}{a}U_{i,j} + \frac{2}{a+1}U_{i+1,j}\right),$$

$$u_{xx}^- = \frac{1}{h^2}\left(\frac{2}{b(b+1)}U_0^- - \frac{2}{b}U_{i-1,j} + \frac{2}{b+1}U_{i-2,j}\right),$$

$$\beta^+ u_x^+ = \frac{\beta^+}{h}\left(-\frac{2a+1}{a(a+1)}U_0^+ + \frac{a+1}{a}U_{i,j} - \frac{a}{a+1}U_{i+1,j}\right),$$

$$\beta^- u_x^- = \frac{\beta^-}{h}\left(\frac{2b+1}{b(b+1)}U_0^- - \frac{b+1}{b}U_{i-1,j} + \frac{b}{b+1}U_{i-2,j}\right),$$

$$\begin{aligned}(\beta^+ u_x^+)_x = \frac{1}{a^2(a+1)^2h^2}\Big\{ &\Big(\beta^+(2a+1)^2 - \beta_{i,j}(a+1)^2 - \beta_{i+1,j}a^2\Big)U_0^+ \\ &- \Big(\beta^+(a+1)^2(2a+1) - \beta_{i,j}(1-a^2)(a+1)^2 - \beta_{i+1,j}a^2(a+1)^2\Big)U_{i,j} \\ &+ \Big(\beta^+a^2(2a+1) + \beta_{i,j}a^2(a+1)^2 - \beta_{i+1,j}a^3(2+a)\Big)U_{i+1,j}\Big\},\end{aligned}$$

and finally

$$(\beta^- u_x^-)_x = \frac{1}{b^2(b+1)^2h^2}\Big\{\Big(\beta^-(2b+1)^2 - \beta_{i-1,j}(b+1)^2 - \beta_{i-2,j}b^2\Big)U_0^- - \Big(\beta^-(b+1)^2(2b+1) - \beta_{i-1,j}(1-b^2)(b+1)^2 - \beta_{i-2,j}b^2(b+1)^2\Big)U_{i-1,j} + \Big(\beta^- b^2(2b+1) + \beta_{i-1,j}b^2(b+1)^2 - \beta_{i-2,j}b^3(2+b)\Big)U_{i-2,j}\Big\}.$$

Similar expressions can be found for the jumps in y-direction. This procedure is repeated for every irregular node at every iteration step until convergence is reached in solving Eq. (5).

3.4. Solving the discrete equations

The system of linear discretized equations (5) is a symmetric and diagonally dominant matrix problem which can be solved with most standard linear solvers. Implementation into already existing codes is straightforward, as all correction applies to the right-hand side of the linear system only. We have successfully implemented the method using a Successive Overrelaxation Red–Black Gauss Seidel method.

In each iteration we need to modify the right-hand-side to adjust the discretization at irregular nodes. To ensure numerical stability and convergence, we found it necessary to underrelax the approximation of the correction term $C_{i,j}$, i.e.

$$C_{i,j}^l = C_{i,j}^{l-1} - \alpha(C_{i,j}^{l-1} - C_{i,j}^{\text{new}}),$$

where $C_{i,j}^{\text{new}}$ is the correction term approximated from the lth iteration step and α is the underrelaxation parameter.

The initial guess, normally set equal to zero, for the solution and for the first approximated correction term, $C_{i,j}^0$, deviates of course largely from the correct values, and consequently the term $(C_{i,j}^0 - C_{i,j}^{\text{new}})$ will be large. Such large adjustments in the right-hand-side of the discretized system will lead to numerical instabilities. To suppress these instabilities the underrelaxation parameter α was introduced. This effect was enhanced for finer grids, thereby requiring a smaller α. As the numerical solution was iterated towards the correct solution, the underrelaxation parameter could slowly be increased and the system would still be stable. We did not seek for any optimal value, but for the examples below it was sufficient to set α somewhere between 0.05 and 0.01 in most cases.

4. Numerical examples

We have performed a number of numerical experiments to test our method. From these experiments we are particularly interested in the accuracy of the computed solution and how well it performs when we estimate the jumps according to Section 3.3 compared to using the exact jumps. We also compare some of our results with other authors' work.

All examples below are computed on a square domain Ω, $[-1,1]\times[-1,1]$, with equally spaced nodes, so that $\Delta x = \Delta y = h$ and $N = M = n$. The accuracy of the scheme is found from a grid refinement analysis. The order of the scheme is given as

$$\text{order} = \left|\frac{\log(\|E_n\|_\infty/\|E_{2n}\|_\infty)}{\log(2)}\right|,$$

where we use the maximum norm

$$\|E_n\|_\infty = \max_{i,j} \left|u(x_i, y_j) - U_{i,j}\right|$$

to estimate the error using an $n \times n$ grid.

We are also interested in finding the error when we mimic the correction term. Let $\bar{C}_{i,j}$ represent the approximate correction, then the relative error in the correction term is defined as

$$\|\mathrm{CE}_n\|_\infty = \frac{\max_{i,j} \left|C_{i,j} - \bar{C}_{i,j}\right|}{\max_{i,j} \left|C_{i,j}\right|}.$$

4.1. Example 1

In the first example we consider a case studied in [8,18]. We solve Laplace's equation, $\nabla^2 u = 0$ and the interface is defined by the circle $x^2 + y^2 = 1/4$ with the jump conditions $[u] = 0$ and $[u_n] = 2$. This can be considered as a problem where there is a singular source term along the interface, and the exact solution is given as

$$u(x,y) = \begin{cases} 1, & x^2 + y^2 < 1/4, \\ 1 + \log(2\sqrt{x^2 + y^2}), & x^2 + y^2 \geqslant 1/4. \end{cases}$$

The exterior boundary conditions are given from the exact solution.

Table 1 (top) shows the result of the grid refinement analysis. The first column shows the mesh size. The next two columns give the maximum error and order of convergence when we use the exact correction term calculated from the true solution. The last four columns give the result when we approximate the correction term according to Section 3.3. We notice the absolute error increases when we approximate the correction term, but both cases show that the convergence approaches second order as the grid is refined.

To explain the discrepancy in error between the two approaches, we replaced Eqs. (16) and (17) with the exact values given by the solution at the interface. These fixed values were then used to approximate the one-sided differences at the interface according to Section 3.3. The results are summarized in Table 2. We

Table 1
Grid refinement analysis in example 1

n	Exact correction term		Approximate correction term			
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|\mathrm{CE}_n\|_\infty$	Order
20	1.048×10^{-3}		2.049×10^{-2}		2.519×10^{-2}	
40	2.403×10^{-4}	2.12	6.366×10^{-3}	1.69	8.204×10^{-3}	1.61
80	6.436×10^{-5}	1.90	1.760×10^{-3}	1.85	2.890×10^{-3}	1.51
160	1.565×10^{-5}	2.04	4.747×10^{-4}	1.89	7.550×10^{-4}	1.94
320	3.185×10^{-6}	2.30	1.206×10^{-4}	1.98	1.983×10^{-4}	1.93
n	IIM in [8]		EJIIM in [18]			
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order		
20	2.391×10^{-3}		1.4×10^{-3}			
40	8.346×10^{-4}	1.52	1.8×10^{-4}	2.94		
80	2.445×10^{-4}	1.77	6.6×10^{-5}	1.43		
160	6.686×10^{-5}	1.87	1.9×10^{-5}	1.77		
320	1.567×10^{-5}	2.09	3.4×10^{-6}	2.51		

Top: results obtained in this study. Bottom: results obtained in [8] and [18].

Table 2
Grid refinement analysis with fixed interface in example 1

n	Approximate correction term (fixed interface)			
	$\|E_n\|_\infty$	Order	$\|\mathrm{CE}_n\|_\infty$	Order
20	7.883×10^{-4}		5.635×10^{-3}	
40	2.011×10^{-4}	1.97	1.383×10^{-3}	2.03
80	5.032×10^{-5}	2.00	4.430×10^{-4}	1.64
160	1.267×10^{-5}	1.99	1.181×10^{-4}	1.91
320	3.187×10^{-6}	1.99	2.651×10^{-5}	2.16

notice the maximum error decreases significantly and becomes nearly identical to the error obtained when using the exact correction terms. This is illustrated in Fig. 3 where we plot the error for all three cases together with the numerical solution on a 40×40 mesh.

The order of convergence compares well with the results reported in [8,18] for all three cases (see Table 1, bottom), but an equivalent or better accuracy is only achieved for the case with exact correction term or with a fixed interface. When we approximate the correction term, the maximum error is worse than obtained in [8,18]. We realize a possible weakness in using Eqs. (16) and (17), though we still find the performance satisfactorily. Alternatively, using even higher order differences complicates the implementation by involving more grid nodes. More nodes may require higher grid resolution to allow for one-sided differences at the interface.

If we look at the accuracy of the approximate correction term, we notice that the relative error, $\|\mathrm{CE}_n\|_\infty$, decreases almost with a rate of second order. This is better than what we expected and clearly satisfies our

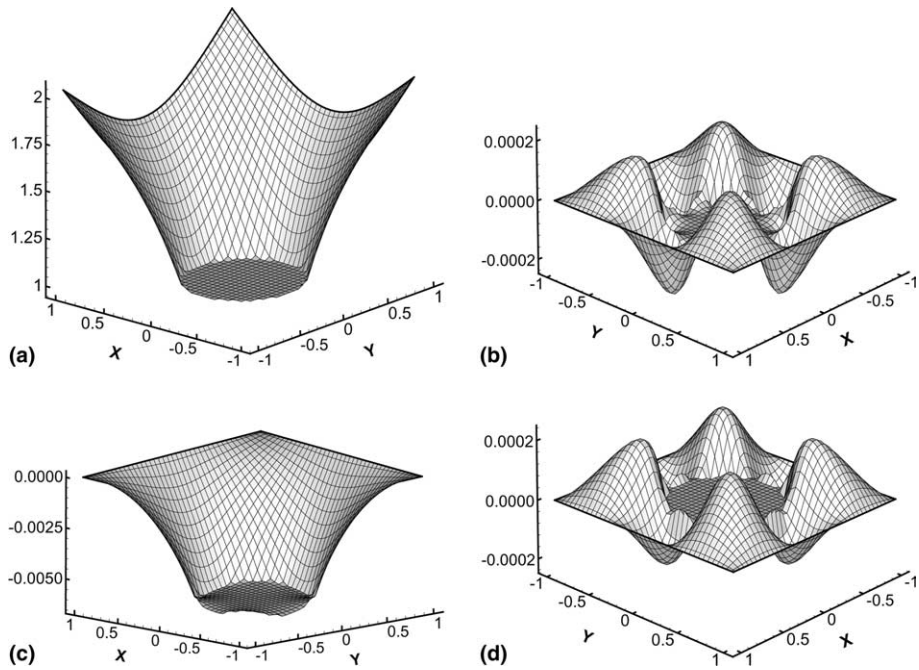

Fig. 3. Example 1. (a) The numerical solution on a 40×40 mesh. (b) The error when using the exact correction term. (c) The error when using approximate correction term. (d) The error when using approximate correction term with fixed interface.

requirements of at least local first order accuracy. Using exact values at the interface improves the error for the correction term.

4.2. Example 2

Now we consider an example with discontinuous coefficients. The problem is also given by [8,12,18] and is defined by the variable coefficient elliptic equation

$$(\beta u_x)_x + (\beta u_y)_y = f(x,y)$$

with

$$\beta(x,y) = \begin{cases} r^2 + 1, & r < 1/2, \\ b, & r \geqslant 1/2, \end{cases}$$

and

$$f(x,y) = 8r^2 + 4,$$

for jumps $[u] = 0$ and $[\beta u_n] = C/r$, where $r = \sqrt{x^2 + y^2}$ and C and b are arbitrary constants. The boundary values are found from the analytical solution,

$$u(x,y) = \begin{cases} r^2, & r < 1/2, \\ \left(1 - \frac{1}{8b} - \frac{1}{b}\right)/4 + \left(\frac{r^4}{2} + r^2\right) + C\log(2r), & r \geqslant 1/2. \end{cases}$$

The results are summarized in Table 3 for $b = 10$ and $C = 0.1$. The maximum error does not show a significant difference whether we approximate the correction term or not. The decrease in error is second order as the grid resolution increases. We obtain better accuracy than [8], but their approach converges slightly faster as the grid is refined. In [18], the convergence is slower than obtained here, however, their maximum norm is very close to our results when we approximate the correction term. Again, the relative error, $\|\mathrm{CE}_n\|_\infty$, in approximating the correction term converges quadratically as the grid is refined.

Table 3
Grid refinement analysis in example 2 with $b = 10$ and $C = 0.1$

n	Exact correction term		Approximate correction term			
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|\mathrm{CE}_n\|_\infty$	Order
20	9.643×10^{-4}		1.394×10^{-3}		1.479×10^{-2}	
40	2.490×10^{-4}	1.95	3.228×10^{-4}	2.11	3.479×10^{-3}	2.09
80	6.315×10^{-5}	1.98	7.857×10^{-5}	2.04	9.863×10^{-4}	1.82
160	1.589×10^{-5}	1.99	1.925×10^{-5}	2.03	2.143×10^{-4}	2.20
320	3.922×10^{-6}	2.02	4.774×10^{-6}	2.01	5.782×10^{-5}	1.89
n	IIM in [8]		EJIIM in [18]			
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order		
20	3.520×10^{-3}		7.6×10^{-4}			
40	7.561×10^{-4}	2.22	2.4×10^{-4}	1.7		
80	1.651×10^{-4}	2.20	7.9×10^{-5}	1.6		
160	3.600×10^{-5}	2.20	2.2×10^{-5}	1.8		
320	8.441×10^{-6}	2.09	5.3×10^{-6}	2.1		

Top: results obtained in this study. Bottom: results obtained in [8] and [18].

Table 4
Grid refinement analysis with fixed interface in example 2 with $b = 10$ and $C = 0.1$

n	Approximate correction term (fixed interface)			
	$\|E_n\|_\infty$	Order	$\|\mathrm{CE}_n\|_\infty$	Order
20	5.378×10^{-4}		4.525×10^{-3}	
40	1.378×10^{-4}	1.96	1.059×10^{-3}	2.10
80	3.470×10^{-5}	1.99	2.732×10^{-4}	1.95
160	8.704×10^{-6}	2.00	5.672×10^{-5}	2.27
320	2.177×10^{-6}	2.00	1.522×10^{-5}	1.90

As in the previous example we want to investigate the effect of replacing Eqs. (16) and (17) with the exact values given by the true solution. Table 4 summarizes the results. As earlier, we notice improvements in accuracy. However, for this problem the improvements are not that significant and we believe that Eqs. (16) and (17) give a fairly good estimate of the values at the interface. Fig. 4 shows the numerical solution and the error for the three cases.

We also consider the same problem with large jumps in the coefficients at the interface. The results are given in Table 5 for $b = 1000$ and $b = 0.001$ when we approximate the correction term. The method still converges with second order and the accuracy is slightly better than obtained in [12]. It should be noted that for small b, the solution in the outer region becomes very large so that the absolute error in Table 5 is actually small compared to the exact solution. We discuss some numerical problems associated with highly discontinuous coefficients in more detail in the next example.

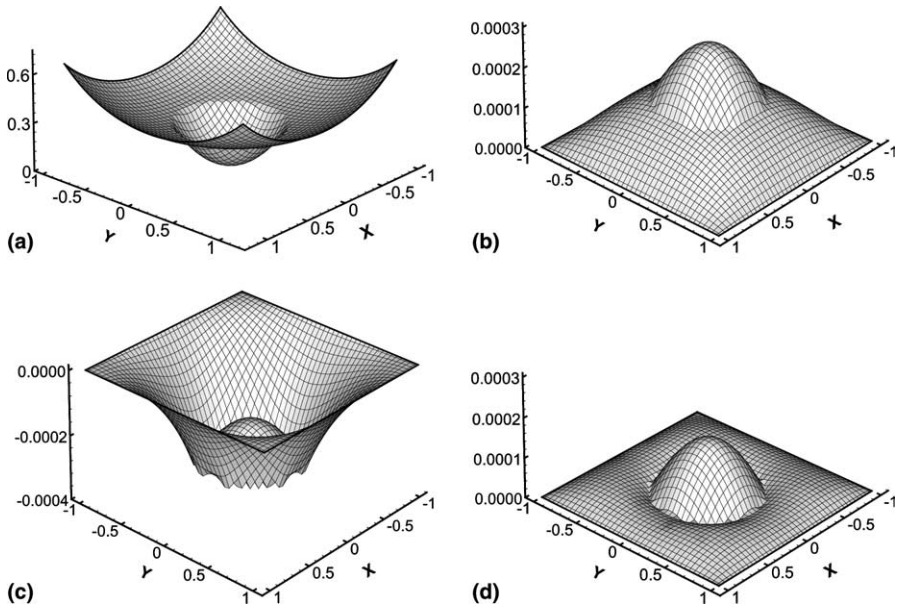

Fig. 4. Example 2. (a) The numerical solution on a 40×40 mesh. (b) The error when using the exact correction term. (c) The error when using approximate correction term. (d) The error when using approximate correction term with fixed interface.

Table 5
Grid refinement analysis in example 2 with large jumps in the coefficients

n	$b = 1000$, $C = 0.1$				$b = 0.001$, $C = 0.1$			
	$\|E_n\|_\infty$	Order	$\|\mathrm{CE}_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|\mathrm{CE}_n\|_\infty$	Order
32	2.083×10^{-4}		7.563×10^{-4}		4.971×10^{0}		1.285×10^{-2}	
64	5.296×10^{-5}	1.98	1.702×10^{-4}	2.15	1.176×10^{0}	2.08	3.097×10^{-3}	2.05
128	1.330×10^{-5}	1.99	4.731×10^{-5}	1.85	2.900×10^{-1}	2.02	7.680×10^{-4}	2.01
256	3.330×10^{-6}	2.00	1.234×10^{-5}	1.93	7.086×10^{-2}	2.03	1.903×10^{-4}	2.01
n	$b = 1000$, $C = 0.1$		$b = 0.001$, $C = 0.1$					
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order				
32	5.136×10^{-4}		9.246×10^{0}					
64	8.235×10^{-5}	2.76	2.006×10^{0}	2.32				
128	1.869×10^{-5}	2.19	5.808×10^{-1}	1.83				
256	4.026×10^{-6}	2.24	1.374×10^{-1}	2.10				

Top: results obtained in this study with approximated correction term. Bottom: results obtained in [12].

4.3. Example 3

In this example we consider a composite material problem with piecewise constant coefficients. We are particularly interested in the case with large differences in material properties. Let

$$u(x,y) = \begin{cases} \dfrac{2x}{\rho + 1 + s^2(\rho - 1)}, & r < s, \\ \dfrac{x(\rho+1) - s^2(\rho-1)x/r^2}{\rho + 1 + s^2(\rho - 1)}, & r \geqslant s, \end{cases}$$

where $\rho = \beta^-/\beta^+$, $r = \sqrt{x^2 + y^2}$ and s is the radius of the circular interface. This is the solution to $\nabla^2 u = 0$ with $[u] = 0$ and $[\beta u] = 0$ at the interface and exterior boundary as given by the analytical solution.

In fact, this problem is identical to Example 7.3 in [18] where they reported poor performance for the fast iterative IIM [10] (FIIM) and their own explicit-jump IIM (EJIIM) in some cases. In EJIIM the correction term is found from approximating the jumps with one-sided interpolation on one side of the interface only. Choosing the correct side of the interface may be crucial for the results. They explained this with support from their observations that the interior solution is pictured by a circular plane and finite differences on this side represent the normal derivatives exactly, thus the jumps are exact. On the exterior side, the finite difference representation is not that exact, introducing an error in the jump approximation and reducing the accuracy of the scheme.

Tables 6 and 7 summarize the results for $\rho = 5000$, while Tables 8 and 9 summarize the grid refinement analysis for $\rho = 1/5000$. We have chosen $s = 1/2$ for both cases. For $\rho = 5000$ ($\beta^- \gg \beta^+$) we clearly have $u_n^+ \gg u_n^-$ and we may assume that Eq. (16) simplifies into solving $u_n^- = 0$ if u_n^+ is finite and not too large (this is actually not true, but Eq. (16) is mostly influenced by the solution in the interior region as long as u_n^+ is not too large). On the interior side, the finite differences approximate the normal derivatives exactly. Therefore, the interface values are correctly approximated. This explains the good performance of our approach for $\rho = 5000$. There are no remarkable differences in accuracy whether we use exact jumps or approximate them with or without the exact solution at the interface. This is also illustrated in Fig. 5 where the numerical solution is plotted together with the error for all three cases. The differences in error are small and all cases show better accuracy than found in [18].

Table 6
Grid refinement analysis in example 3 with $\rho = 5000$

n	Exact correction term		Approximate correction term			
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order
25	9.811×10^{-4}		8.185×10^{-4}		9.441×10^{-3}	
50	2.730×10^{-4}	1.85	3.278×10^{-4}	1.32	4.796×10^{-3}	0.98
100	4.841×10^{-5}	2.50	5.277×10^{-5}	2.64	3.046×10^{-3}	0.65
200	1.260×10^{-5}	1.94	1.371×10^{-5}	1.94	1.802×10^{-3}	0.75
400	3.491×10^{-6}	1.85	3.653×10^{-6}	1.91	9.370×10^{-4}	0.94
n	FIIM		Interior EJIIM		Exterior EJIIM	
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order
25	1.2×10^{-2}		1.4×10^{-3}		9.1×10^{-2}	
50	9.2×10^{-2}		3.5×10^{-4}	2.0	2.5×10^{-2}	1.9
100	5.9×10^{-2}	0.6	9.0×10^{-5}	2.0	6.8×10^{-3}	1.9
200	7.7×10^{-3}	2.9	2.2×10^{-5}	2.0	2.0×10^{-3}	1.8

Top: results obtained in this study. Bottom: results obtained in [18].

Table 7
Grid refinement analysis with fixed interface in example 3 with $\rho = 5000$

n	Approximate correction term (fixed interface)			
	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order
25	8.136×10^{-4}		9.410×10^{-3}	
50	2.249×10^{-4}	1.85	4.611×10^{-3}	1.03
100	5.137×10^{-5}	2.13	3.044×10^{-3}	0.60
200	1.372×10^{-5}	1.90	1.803×10^{-3}	0.76
400	3.423×10^{-6}	2.00	9.368×10^{-4}	0.94

Table 8
Grid refinement analysis in example 3 with $\rho = 1/5000$

n	Exact correction term		Approximate correction term			
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order
25	1.635×10^{-3}		2.267×10^{-2}		6.126×10^{-2}	
50	4.549×10^{-4}	1.85	7.038×10^{-3}	1.68	1.843×10^{-2}	1.73
100	8.066×10^{-5}	2.50	1.934×10^{-3}	1.86	7.108×10^{-3}	1.37
200	2.102×10^{-5}	1.94	5.209×10^{-4}	1.89	3.103×10^{-3}	1.20
400	5.823×10^{-6}	1.85	1.346×10^{-4}	1.95	1.280×10^{-3}	1.28
n	FIIM		Interior EJIIM		Exterior EJIIM	
	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|E_n\|_\infty$	Order
25	5.2×10^{-3}		1.9×10^{-3}		1.3×10^{1}	
50	1.6×10^{-3}	1.7	5.5×10^{-4}	1.8	5.6×10^{0}	1.3
100	2.3×10^{-4}	2.8	1.3×10^{-4}	2.1	6.4×10^{-1}	3.1
200	5.0×10^{-5}	2.2	3.2×10^{-5}	2.0	8.1×10^{-2}	3.0

Top: results obtained in this study. Bottom: results obtained in [18].

For $\rho = 1/5000$ ($\beta^- \ll \beta^+$) we may assume that Eq. (16) simplifies into solving $u_n^+ = 0$ if u_n^- is finite and not too large. In this case, the finite difference approximation of the normal derivative is less accurate on the exterior side, resulting in an inaccurate approximation of the values at the interface. This explains

Table 9
Grid refinement analysis with fixed interface in example 3 with $\rho = 1/5000$

n	Approximate correction term (fixed interface)			
	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order
25	1.356×10^{-3}		9.410×10^{-3}	
50	3.748×10^{-4}	1.85	4.611×10^{-3}	1.03
100	8.560×10^{-5}	2.13	3.044×10^{-3}	0.60
200	2.287×10^{-5}	1.90	1.803×10^{-3}	0.76
400	5.703×10^{-6}	2.00	9.368×10^{-4}	0.94

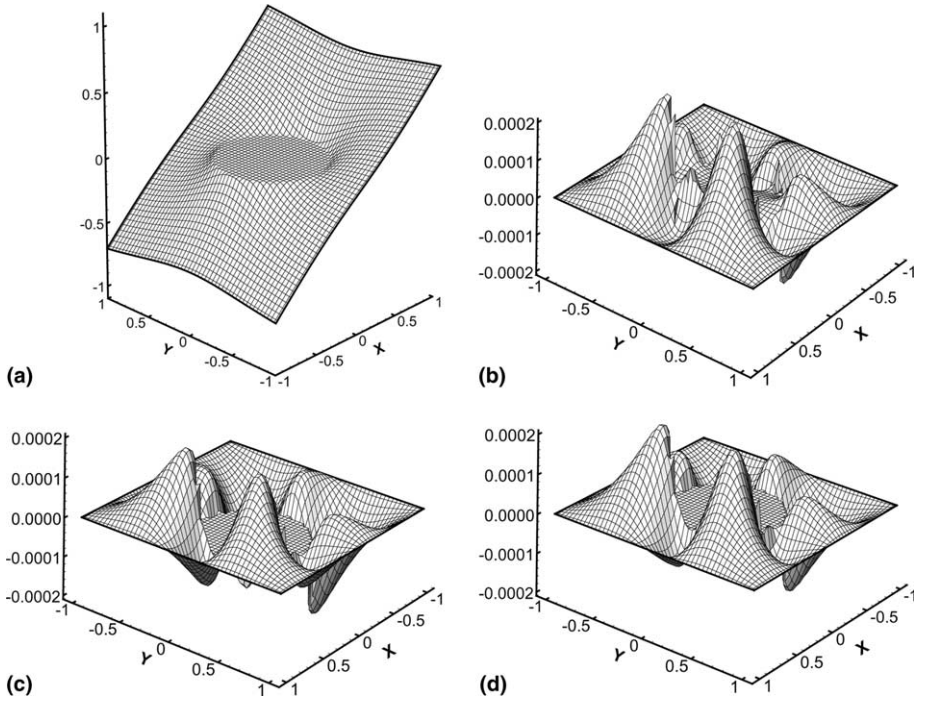

Fig. 5. Example 5, $\rho = 5000$. (a) The numerical solution on a 50×50 mesh. (b) The error when using the exact correction term. (c) The error when using approximate correction term. (d) The error when using approximate correction term with fixed interface.

why the accuracy is lower when we use Eqs. (16) and (17) to approximate the correction terms. If we instead replace Eqs. (16) and (17) with the exact values when approximating the jumps, the accuracy becomes nearly identical with the case where we use exact jumps to find the correction terms. This is also shown in Fig. 6.

Despite the lower accuracy obtained for $\rho = 1/5000$ when we used approximate correction terms, the performance of our method seems acceptable. The result obtained on the coarsest grid for $\rho = 1/5000$ is still better than the best approximation with the exterior EJIIM in [18], but it is worse than the results obtained with the FIIM or interior EJIIM. The rate of convergence is second order for all cases. The relative error, $\|CE_n\|_\infty$, in approximating the correction term converges only with a first order rate here. This is actually what we first expected when we derived the scheme for finding the jumps at the interface.

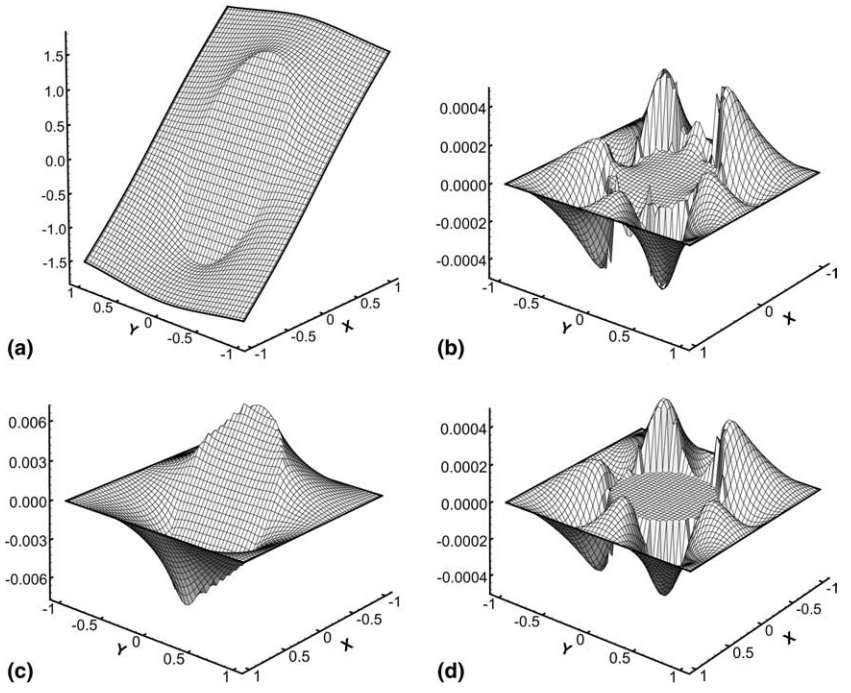

Fig. 6. Example 5, $\rho = 1/5000$. (a) The numerical solution on a 50×50 mesh. (b) The error when using the exact correction term. (c) The error when using approximate correction term. (d) The error when using approximate correction term with fixed interface.

4.4. Example 4

In the two remaining examples we will only focus on the performance of approximating the correction term. We want to compare the performance of the method with constant and variable coefficients by solving two different problems leading to the same exact solution,

$$u(x,y) = \begin{cases} e^x \cos y, & x^2 + y^2 < 1/4, \\ 0, & x^2 + y^2 \geqslant 1/4, \end{cases}$$

with a discontinuity at the interface.

Case I is defined by $\nabla^2 u = 0$ with jumps $[u] = -e^x \cos y$ and $[u_n] = 2e^x(y \sin y - x \cos y)$ (this example is also found in [8,13]). In Case II, we solve the variable coefficient elliptic equation $(\beta u_x)_x + (\beta u_y)_y = f(x,y)$, where

$$\beta(x,y) = \begin{cases} x^2 + y^2 + 1, & x^2 + y^2 < 1/4, \\ 1, & x^2 + y^2 \geqslant 1/4, \end{cases}$$

$$f(x,y) = \begin{cases} 2e^x(y \sin y - x \cos y), & x^2 + y^2 < 1/4, \\ 0, & x^2 + y^2 \geqslant 1/4, \end{cases}$$

and the jumps are $[u] = -e^x \cos y$ and $[\beta u_n] = 2e^x(x^2 + y^2 + 1)(y \sin y - x \cos y)$. The exterior boundary condition is $u = 0$ for both cases.

The results from Case I and II are summarized in Table 10 where we have used approximate correction terms. We can see how the discontinuity is captured sharply in Fig. 7. The accuracy of both numerical

Table 10
Grid refinement analysis in example 4

n	Case I				Case II			
	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order
20	6.429×10^{-4}		1.964×10^{-4}		7.771×10^{-4}		2.309×10^{-3}	
40	1.895×10^{-4}	1.76	3.026×10^{-5}	2.70	2.302×10^{-4}	1.76	4.947×10^{-4}	2.22
80	5.085×10^{-5}	1.90	6.417×10^{-6}	2.24	6.193×10^{-5}	1.89	1.452×10^{-4}	1.77
160	1.321×10^{-5}	1.94	8.480×10^{-7}	2.92	1.601×10^{-5}	1.95	3.357×10^{-5}	2.11
320	3.342×10^{-6}	1.98	1.047×10^{-7}	3.02	4.051×10^{-6}	1.98	8.689×10^{-6}	1.94

n	Results obtained in [8]	
	$\|E_n\|_\infty$	Order
20	4.379×10^{-4}	
40	1.079×10^{-4}	2.02
80	2.778×10^{-5}	1.95
160	7.499×10^{-6}	1.89
320	1.740×10^{-6}	2.11

Top: results obtained in this study. Bottom: results obtained in [8] for Case I.

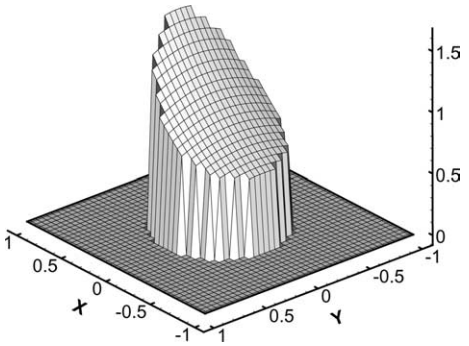

Fig. 7. Example 4. The numerical solution on a 40×40 mesh.

solutions agrees well with each other and both converge quadratically. In Case I, the relative error, $\|CE_n\|_\infty$, in the correction term becomes negligibly small already at coarse grids. The rate of convergence of $\|CE_n\|_\infty$ is close to three. The relative error in the correction term in Case II is also remarkably low compared to previous examples. We have also included the results obtained in [8] for Case I in Table 10. Their results are slightly better than what we achieved, but the differences are not significant.

4.5. Example 5

For this example we solve one problem with different shapes of the interface. Consider the variable coefficient Elliptic equation $(\beta u_x)_x + (\beta u_y)_y = f(x,y)$ with the coefficient

$$\beta(x,y) = \begin{cases} x^2 + y^2 + 1, & \phi(x,y) < 0, \\ x + 2, & \phi(x,y) \geqslant 0, \end{cases}$$

and source term

Table 11
Grid refinement analysis in example 5

n	Case I				Case II			
	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order	$\|E_n\|_\infty$	Order	$\|CE_n\|_\infty$	Order
20	4.141×10^{-4}		1.591×10^{-3}					
40	1.205×10^{-4}	1.78	3.578×10^{-4}	2.15	1.732×10^{-4}		3.935×10^{-4}	
80	3.254×10^{-5}	1.89	1.038×10^{-4}	1.79	4.916×10^{-5}	1.82	1.029×10^{-4}	1.94
160	8.365×10^{-6}	1.96	2.406×10^{-5}	2.11	1.109×10^{-5}	2.15	2.469×10^{-5}	2.06
320	2.130×10^{-6}	1.97	6.141×10^{-6}	1.97	2.933×10^{-6}	1.91	5.468×10^{-6}	2.17

$$f(x,y) = \begin{cases} 2e^x(y\sin y - x\cos y), & \phi(x,y) < 0, \\ -2x - 3, & \phi(x,y) \geqslant 0. \end{cases}$$

The jumps are given as $[u] = x - y^2 - e^x \cos y$ and $[\beta u_n] = \{x + 2 - (x^2 + y^2 + 1)e^x \cos y\}n_x + \{-2y(x+2) + (x^2 + y^2 + 1)e^x \sin y\}n_y$, where the normal vector $\vec{n} = (n_x, n_y)$ is given as $\nabla\phi/|\nabla\phi|$.

For Case I, we define the level set function $\phi(x,y)$ as follows:

$$\phi(x,y) = \text{sign}(x^2 + y^2 - 1/4)\sqrt{x^2 + y^2 - 1/4}.$$

For Case II, the level set function is obtained by solving the minimum value problem

$$d = \min \sqrt{(x - X(\theta))^2 + (y - Y(\theta))^2},$$

where the interface is given by the parametrized curve $(X(\theta), Y(\theta))$,

$$X(\theta) = \frac{3}{8}\cos(\theta) - \frac{1}{4}\cos(3\theta),$$
$$Y(\theta) = \frac{2}{3}\sin(\theta) - \frac{1}{12}\sin(3\theta) + \frac{1}{15}\sin(7\theta),$$

for $\theta \in [0, 2\pi)$. Then $\phi(x,y) = \pm d$, where the negative sign corresponds to the inner region enclosed by the curve $(X(\theta), Y(\theta))$, while the plus sign is for the outer region. The exterior boundary conditions are given from the exact solution

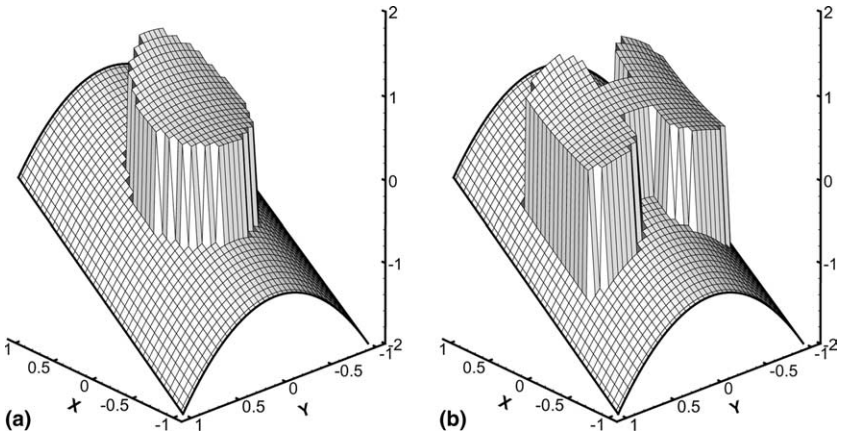

Fig. 8. Example 5. (a) Numerical solution on a 40×40 mesh, Case I. (b) Numerical solution on a 40×40 mesh, Case II.

$$u(x,y) = \begin{cases} e^x \cos y, & \phi(x,y) < 0, \\ x - y^2, & \phi(x,y) \geqslant 0. \end{cases}$$

The grid refinement analysis is summarized in Table 11, and Fig. 8 shows the numerical results for both interface shapes. We had to omit the coarsest grid for the irregular shaped interface to allow for one-sided differences. For both cases, the accuracy is high and the rate of convergence is second order. The relative error, $\|CE_n\|_\infty$, in correction terms are almost identical, regardless of interface shape. The accuracy does not seem to be affected by the irregular interface as long as the grid resolution is high enough to allow for one-sided differences at the interface.

5. Summary

We have derived a finite difference method for two-dimensional elliptic equations with discontinuous, variable coefficients and source terms. Numerical experiments show good agreement with analytical solutions, and the rate of convergence is found to be of second order. The method decomposes the interface problem and introduces componentwise correction terms at irregular grid nodes to make the difference scheme well-defined across interfaces. These correction terms are derived so that the difference stencil remains symmetric and diagonally dominant, allowing for most standard solvers to be used. The main advantage of the present approach is to preserve the symmetry of the discretized elliptic problem for more general coefficients than in previous methods, i.e. piecewise smooth coeffecients, with higher order accuracy.

We have also proposed a new method for estimating the solution-dependent correction terms. The main idea is to obtain the correct correction term iteratively in parallel with the solution. Using sufficient underrelaxation on the approximated correction term gives a converging solution. Optimizing the value of the underrelaxation parameter is crucial for the number of iterations required, and a separate study of this issue could be of interest.

The approximate correction approach does not seem to influence the accuracy noticeably. However, for some cases the values estimated at the interface become too inaccurate, increasing the maximum error of the scheme. Despite this weakness, test cases show second order convergence and the accuracy is found to be acceptable. The only restriction is a sufficient grid resolution to allow for one-sided differencing at the interface.

The present method is very simple to implement in existing codes. It does not require constructions of complex coefficient matrices, as required by many other immersed interface methods. We believe an extension to three dimensions should be straight forward following the same approach as presented here.

Acknowledgements

This work was supported by the Norwegian Research Council through the Petronics programme for doctoral studies. The author would especially like to thank his supervisor Professor Tor Ytrehus for his advice, encouragement and comments on the work presented here. The author is also grateful for the discussions with Associate Professor Reidar Kristoffersen and Tore Flåtten, their suggestions and comments were very helpful while preparing the manuscript.

References

[1] L. Adams, Z. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2) (2002) 463–479.

[2] R.P. Beyer, R.J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, SIAM J. Numer. Anal. 29 (2) (1992) 332–364.

[3] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[4] Y.C. Chang, T.Y. Hou, B. Merriman, S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, J. Comput. Phys. 124 (1996) 449–464.
[5] 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.
[6] H. Huang, Z. Li, Convergence analysis of the immersed interface method, IMA J. Numer. Anal. 19 (1999) 583–608.
[7] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (3) (2000) 323–360.
[8] R.J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044.
[9] Z. Li, A note on immersed interface method for three-dimensional elliptic equations, Comput. Math. Applic 31 (3) (1996) 9–17.
[10] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1) (1998) 230–254.
[11] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, J. Comput. Phys 171 (2001) 822–842.
[12] Z. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM J. Sci. Comput. 23 (1) (2001) 330–361.
[13] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[14] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[15] C.S. Peskin, Numerical analysis of blood flow in heart, J. Comput. Phys. 25 (1977) 220–252.
[16] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[17] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[18] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (3) (2000) 827–862.