INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 16, Number 1, Pages 1–17

A NUMERICAL METHOD FOR SOLVING THE VARIABLE COEFFICIENT WAVE EQUATION WITH INTERFACE JUMP CONDITIONS

LIQUN WANG, SONGMING HOU, LIWEI SHI, AND PING ZHANG

Abstract. Wave equations with interface jump conditions have wide applications in engineering and science, for example in acoustics, elastodynamics, seismology, and electromagnetics. In this paper, an efficient non-traditional finite element method with non-body-fitted grids is proposed to solve variable coefficient wave equations with interface jump conditions. Numerical experiments show that this method is approximately second order accurate both in the L^∞ norm and L^2 norm for piecewise smooth solutions.

Key words. Non-traditional finite element method, wave equation, jump condition, variable coefficient.

1. Introduction

Problems involving wave equations with interfaces have a wide variety of applications in science and engineering, for example in acoustics, seismology and electromagnetics. Designing highly effective and computational efficient methods for these problems is nontrivial.

Before studying the wave interface problems, one needs to study the method for solving the elliptic interface problem since that is one of the major challenges in the problem. Therefore we first summarize the past work on elliptic interface problems below.

For nearly four decades, extensive research has been performed in the area of numerical solutions of elliptic equations with discontinuous coefficients and singular sources on Cartesian grids. The choice of uniform Cartesian grids saves the cost of mesh generation. It started with the pioneering work of Peskin [1] on the first order accurate immersed boundary method developed to simulate the pattern of blood flow in the heart.

Also, a great amount of work has been done to use finite difference methods on elliptic interface problems. The main idea is to use difference schemes and stencils near the interface to incorporate the jump conditions and interface in the Taylor expansions. Using finite difference schemes requires the use of high order derivatives of jump conditions and interface conditions. LeVeque and Li proposed the immersed interface method for solving elliptic equations with discontinuous coefficients and singular sources [2]. This method incorporates the interface conditions in both solution and flux, $[u] \neq 0$ and $[\beta u_n] \neq 0$, into the finite difference stencil resulting in second order accuracy. The method produces a linear system that is sparse, but may not be symmetric or positive definite if there is a jump in the coefficient. Detailed information about the IIM can be found in [3].

In [4], the matched interface and boundary method (MIB) was proposed to solve elliptic equations with smooth interfaces. In [5], the MIB method was generalized

Received by the editors November 15, 2017 and, in revised form, February 4, 2018.
2000 *Mathematics Subject Classification.* 65N30.

for problems involving sharp-edged interfaces. In [6], the MIB method was generalized for problems involving triple junction points. This method has achieved 2nd order accuracy in the L^∞ norm even for sharp-edged interfaces.

In [7] and [8], the immersed finite element method (IFEM) was developed using non-body fitted Cartesian meshes for homogeneous jump conditions. The method was extended to treat non-homogeneous jump conditions in [9]. The partially penalized IFEM was developed in [10].

Also, there has been a large body of work from the finite volume perspective for developing high order methods for elliptic equations in complex domains, such as [11, 12] for two dimensional problems and [13] for three dimensional problems. Furthermore, Discontinuous Galerkin method [14] can be used to solve elliptic interface problems. Both the mesh and polynomial degree can be adaptively refined in a remeshing scheme [15]. Recently, the gradient recovery method [30–32] was introduced for accurate gradient computation.

Some theoretical discussions about interface problems can be found in [16] and [17].

This paper is based on the Petrov-Galerkin type non-traditional finite element method for solving elliptic interface problems that was first introduced in [18] and improved in [19] and [20]. [19] extended the original method to include the case of sharp edged interfaces with matrix coefficients. This extension improved the accuracy for sharp edged interfaces from 0.8th order to nearly second order. In [18] and [19], if the interface hits a grid point exactly, it is perturbed away. [20] treats this case carefully without perturbation. The second improvement in [20] is that not only Dirichlet but also Neumann boundary conditions are considered. The third improvement is that the coefficient matrix data can only be given at the grid points, not as an analytic function. In [21], the method was extended to three dimensions. The extension for the elasticity equations can be found in [22]. The method for multi-domain problems can be found in [23]. Some other extensions can be found in [24] and [25]. The method has two advantages: first, this method uses non-body-fitted grid so that the cost of mesh generation can be saved. Second, compared with other methods the method is easier to implement for complicated problems with non-homogeneous jump conditions and matrix coefficients.

In [26], the second order accurate immersed interface method is used to solve the wave equation with interface jump conditions. The wave equation is rewritten as a first order system. In [27], the correction function method is proposed to solve the wave equation with interface jump conditions. The result is excellent with 4th order accuracy. However, the scope of the work is for constant coefficient case only. On both sides of the interface, it is the Laplacian operator. The scope of our work in this paper is the variable matrix coefficient case, which has wider applications.

2. Formulation and Numerical Method

2.1. Problem Definition and Weak Formulation. In this paper, we solve the wave equation with discontinuous variable matrix coefficients along the interface. Consider a rectangular domain $\Omega = (x_{min}, x_{max}) \times (y_{min}, y_{max})$. Γ is an interface prescribed by the zero level-set $\{(x, y) \in \Omega \mid \phi(x, y) = 0\}$ of a level-set function $\phi(x, y)$. The unit normal vector of Γ is $\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}$ pointing from $\Omega^- = \{(x, y) \in \Omega \mid \phi(x, y) < 0\}$ to $\Omega^+ = \{(x, y) \in \Omega \mid \phi(x, y) > 0\}$, see Figure 1. Now the governing equation reads

$$\frac{\partial^2 u(x,y,t)}{\partial t^2} - \bigtriangledown \cdot (\beta(x,y,t) \bigtriangledown u(x,y,t)) \quad = \quad f(x,y,t), \text{ in } (\Omega \setminus \Gamma) \times [0,T], \tag{1}$$

in which $\bigtriangledown$ is the gradient operator. The coefficient $\beta(x, y, t)$ for each fixed t is assumed to be a 2×2 matrix that is uniformly elliptic on each disjoint subdomain, Ω^- and Ω^+, and its components are continuously differentiable on each disjoint subdomain, but they may be discontinuous across the interface Γ.

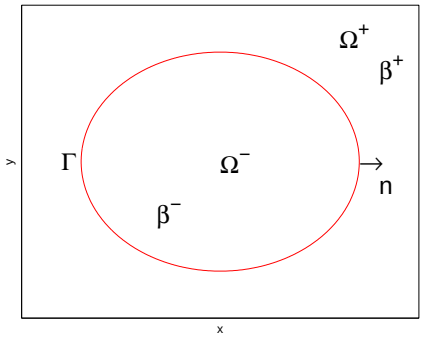


FIGURE 1. A rectangular domain $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$.

Given functions a and b along the interface Γ, we prescribe the jump conditions on $\Gamma \times [0, T]$

$$
\begin{aligned}
a(x, y, t) &= [u]_\Gamma (x, y, t) \\
&\equiv u^+(x, y, t) - u^-(x, y, t), \\
b(x, y, t) &= [(\beta \bigtriangledown u) \cdot \boldsymbol{n}]_\Gamma (x, y, t) \\
&\equiv \boldsymbol{n} \cdot (\beta^+(x, y, t) \bigtriangledown u^+(x, y, t)) - \boldsymbol{n} \cdot (\beta^-(x, y, t) \bigtriangledown u^-(x, y, t)),
\end{aligned} \tag{2}
$$

The superscripts " $\pm$ " refer to limits taken from within the subdomains $\Omega^\pm$. The initial conditions can be given as

$$
\begin{aligned}
u(x, y, 0) &= u^0(x, y), \text{ in } \Omega, \\
\frac{\partial u}{\partial t}(x, y, 0) &= u_t^0(x, y), \text{ in } \Omega.
\end{aligned} \tag{3}
$$

where u_0 and u_t^0 are given functions on the domain Ω. Finally, we prescribe the boundary condition

$$
u(x, y, t) = g(x, y, t), \text{ on } \partial\Omega \times [0, T] \tag{4}
$$

for a given function g on the boundary $\partial\Omega$.

In this paper, we use non-traditional finite element method [18–20] to solve the wave equation with interface jump conditions described by equations (1)-(4). By multiplying both sides of equation (1) with the test function $\psi \in H_0^1(\Omega)$ and integrating over the domain Ω, from Green's theorem we deduce the weak formulation of equation (1) with $u = g$ on boundary points as

$$
\int_\Omega f\psi = \int_{\Omega^+ \bigcup \Omega^-} \frac{\partial^2 u}{\partial t^2}\psi + \int_{\Omega^+ \bigcup \Omega^-} \beta \nabla u \cdot \nabla \psi + \int_\Gamma b\psi. \tag{5}
$$

2.2. Time Discretization. Below we employ the weak formulation for the wave equation. First divide the time interval $[0, T]$ into N equally spaced subintervals $[t_{n-1}, t_n], n = 1, 2, \cdots, N$, with $t_n = n\Delta t$, where $\Delta t = T/N$ is the time step. For simplicity, we use u^n to denote $u(\boldsymbol{x}, t_n)$.

In order to derive a second order accuracy scheme, an efficient method called θ-Scheme introduced in [28] is used. According to the θ-Scheme, equation (1) can be rewritten as

$$u_{tt}^n = \Delta u^{n,\theta} + f^{n,\theta}, \quad n = 1, 2, \cdots, N. \tag{6}$$

where

$$\begin{aligned} u_{tt}^n &= \frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2}, \\ \Delta u^{n,\theta} &= \theta \bigtriangledown \cdot(\beta^{n+1} \bigtriangledown u^{n+1}) + (1-2\theta) \bigtriangledown \cdot(\beta^n \bigtriangledown u^n) + \theta \bigtriangledown \cdot(\beta^{n-1} \bigtriangledown u^{n-1}), \\ f^{n,\theta} &= \theta f^{n+1} + (1-2\theta) f^n + \theta f^{n-1}. \end{aligned}$$

Combining equations (5) and (6) together, the weak formulation of semidiscrete problem reads

$$\begin{aligned} & \int_{\Omega^+ \bigcup \Omega^-} \left(u^{n+1}\psi + \theta\Delta t^2 \beta^{n+1} \nabla u^{n+1} \cdot \nabla\psi\right) \\ = & \int_{\Omega^+ \bigcup \Omega^-} \left(2u^n\psi - (1-2\theta)\Delta t^2 \beta^n \nabla u^n \cdot \nabla\psi\right) - \\ & \int_{\Omega^+ \bigcup \Omega^-} \left(u^{n-1}\psi + \theta\Delta t^2 \beta^{n-1} \nabla u^{n-1} \cdot \nabla\psi\right) + \\ & \Delta t^2 \int_\Omega \left(\theta f^{n+1} + (1-2\theta) f^n + \theta f^{n-1}\right)\psi - \\ & \Delta t^2 \int_\Gamma \left(\theta b^{n+1} + (1-2\theta) b^n + \theta b^{n-1}\right)\psi, \\ & \forall \psi \in H_0^1(\Omega), \end{aligned}$$

According to [29], the θ-Scheme is an unconditional stable scheme when $\theta \geq \frac{1}{4}$. From equation (6), the θ-Scheme we used here is a three-level scheme. It requires appropriate initial conditions u^0 and u^1. Since u^0 and u_t^0 are given in (3), in a normal research domain without interface, the central difference scheme $\frac{u^1 - u^{-1}}{2\Delta t} = u_t^0$ as in [28] can be used to get a second order accuracy scheme. An alternative approach to get u^1 is the Taylor expansion method proposed in [32], which can achieve second order accuracy. However, in this paper we use non-traditional finite element method to solve problems with interfaces. The construction of local system is needed at every time step, which means the jump conditions (2) are needed at every time step. We cannot construct the local system for the fictitious value u^{-1}. In this paper, the forward difference scheme $\frac{u^1 - u^0}{\Delta t} = u_t^0$ is employed to get u^1. For stability concern, we could use a much smaller time step (compared with the time step Δt defined above) to get u^1, say $\frac{\Delta t}{10}$. We use it for all our numerical experiments. The results are not sensitive to this choice.

2.3. Domain Discretization. The spacial domain discretization for wave eqation with interface is similar to the discretization for elliptic interface problems in [18–20]. For completeness of the work, we explain in this section how the discretization is done. More details can be found in [18–20]. We also present in this section the new contribution of combining the θ-Scheme with the spacial discretization. The equations we derived will be used in the next section for numerical experiments.

We restrict ourselves to a rectangular domain $\Omega = (x_{min}, x_{max}) \times (y_{min}, y_{max})$ in the plane. Given positive integers I and J, set $\Delta x = (x_{max} - x_{min})/I$ and $\Delta y = (y_{max} - y_{min})/J$. Define $(x_i, y_j) = (x_{min} + i\Delta x, y_{min} + j\Delta y)$ for $i = 0, ..., I$ and $j = 0, ..., J$ as a uniform Cartesian grid. Each (x_i, y_j) is called a grid point. For the case $i = 0, I$ or $j = 0, J$, a grid point is called a boundary point, otherwise it is called an interior point. If $\phi(x_i, y_j) < 0$, we count the grid point (x_i, y_j) as in Ω^-; if $\phi(x_i, y_j) > 0$, we count the grid point (x_i, y_j) as in Ω^+; otherwise we count it as on the interface Γ. The grid size is defined as $h = max(\Delta x, \Delta y) > 0$.

Two sets of grid functions are needed and they are denoted by

$$H^{1,h} = \{\omega^h = (\omega_{i,j}) : 0 \le i \le I, 0 \le j \le J\},$$

and

$$H_0^{1,h} = \{\omega^h = (\omega_{i,j}) \in H_{\pm}^{1,h} : \omega_{i,j} = 0 \texttt{ if } i = 0, I \texttt{ or } j = 0, J\}.$$

Next we cut every rectangular region $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$ into two right triangles: one is bounded by $x = x_i, y = y_j$ and $y = \frac{y_{j+1} - y_j}{x_i - x_{i+1}}(x - x_{i+1}) + y_j$, the other is bounded by $x = x_{i+1}, y = y_{j+1}$ and $y = \frac{y_{j+1} - y_j}{x_i - x_{i+1}}(x - x_{i+1}) + y_j$. Collecting all those triangular regions, we obtain a uniform triangulation $T^h : \bigcup_{K \in T^h} K$, see Figure 2.

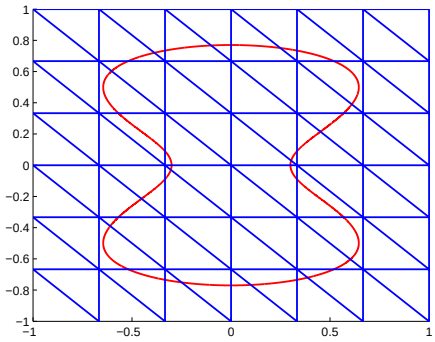


FIGURE 2. A uniform triangulation.

A cell K is called a regular cell if all of its vertices belong to the same subdomain, see Figure 3; otherwise it is called an interface cell, see Figure 4. For an interface cell, we write $K = K^+ \bigcup K^-$. K^+ and $K^-$are approximations of the regions $K \bigcap \Omega^+$ and $K \bigcap \Omega^-$, respectively. K^+ and K^- are separated by a straight line segment, denoted by Γ_K^h. The two end points of the line segment Γ_K^h are located on interface Γ and their locations are calculated from the linear interpolations of the discrete level-set functions $\phi^h = \phi(x_i, y_j)$. In this way, we obtain the triangulation $\mathcal{T}_h$ of the domain Ω.

For any $\psi^h \in H_0^{1,h}$, define $T^h(\psi^h)$ as a standard continuous piecewise linear function, which is a linear function in each triangular cell and $T^h(\psi^h)$ matches ψ^h on grid points. For any $u^{h,n} \in H^{1,h}$ with $u^{h,n} = g^{h,n}$ on boundary points at t_n, $U^h(u^{h,n})$ is a piecewise linear function and matches $u^{h,n}$ on grid points at t_n. It is a linear function in each regular cell, just like the first extension operator $U^h(u^{h,n}) = T^h(u^{h,n})$ in a regular cell at t_n. In each interface cell, it consists of two

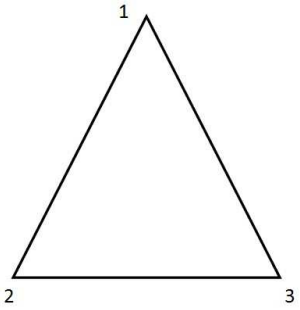
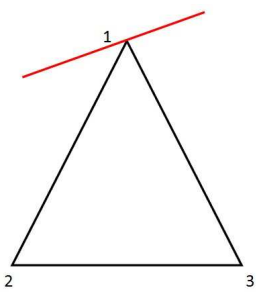
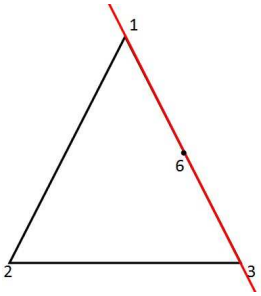


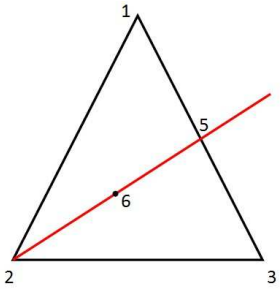
FIGURE 3. Regular cell.



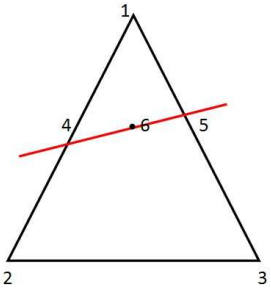
(a) Case 1. The interface comes across one vertex of the triangle.



(b) Case 2. The interface covers one side of the triangle.



(c) Case 3. The interface comes across one vertex and one side of the triangle.



(d) Case 4. The interface cuts two sides of the triangle.

FIGURE 4. Interface cells.

pieces of linear functions, one is on K^+ and the other is on K^-. See [19] for more details.

We shall construct an approximate solution to the interface problem taking into account the jump conditions. Note that the Neumann jump condition $[(\beta\nabla u)\cdot \boldsymbol{n}] = b$ along the interface Γ is already absorbed into the weak formulation, hence, we only need to take care of the Dirichlet jump condition $[u] = a$ along the interface Γ. We shall seek an approximate solution which is piecewise linear on both subdomains Ω^- and Ω^+, but discontinuous along the interface Γ. Clearly, in cases 1-3, when a vertex (x, y) of the interface cell K is on the interface, we need to get two solutions

$u^{h,n}(x,y) = \begin{cases} u^{+,n}(x,y) \\ u^{-,n}(x,y) \end{cases}$ defined at one point. To this end, we introduce a globally piecewise linear approximation $u^{h,n}(x,y)$ defined below:

if $\phi(x,y) > 0$,

$$u^{h,n}(x,y) = u^{+,n}(x,y),$$

and if $\phi(x,y) < 0$,

$$u^{h,n}(x,y) = \begin{cases} u^{+,n}(x,y), \text{ if } \phi(x,y) = 0, \\ u^{-,n}(x,y), \text{ if } \phi(x,y) < 0. \end{cases}$$

For a regular cell(see Figure 3) or an interface cell as in **Case 1** (see Figure 4.(a)), we have the following equation

$$\begin{aligned} & \int_K \left(U^h(u^{h,n+1})T^h(\psi^h) + \theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1})\cdot\nabla T^h(\psi^h)\right) \\ = & \int_K \left(2U^h(u^{h,n})T^h(\psi^h) - (1-2\theta)\Delta t^2\beta^n\nabla U^h(u^{h,n})\cdot\nabla T^h(\psi^h)\right) - \\ & \int_K \left(U^h(u^{h,n-1})T^h(\psi^h) + \theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1})\cdot\nabla T^h(\psi^h)\right) + \\ & \Delta t^2\int_K \left(\theta f^{n+1} + (1-2\theta)f^n + \theta f^{n-1}\right)T^h(\psi^h). \end{aligned} \tag{7}$$

In **Case 2**, The interface is along an edge of the cell. We need to take the Neumann jump condition into consideration.

If $\phi(x_2,y_2) > 0$, we have

$$\begin{aligned} & \int_{K^+} \left(U^h(u^{h,n+1})T^h(\psi^h) + \theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1})\cdot\nabla T^h(\psi^h)\right) \\ = & \int_{K^+} \left(2U^h(u^{h,n})T^h(\psi^h) - (1-2\theta)\Delta t^2\beta^n\nabla U^h(u^{h,n})\cdot\nabla T^h(\psi^h)\right) - \\ & \int_{K^+} \left(U^h(u^{h,n-1})T^h(\psi^h) + \theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1})\cdot\nabla T^h(\psi^h)\right) + \\ & \Delta t^2\int_{K^+} \left(\theta f^{n+1} + (1-2\theta)f^n + \theta f^{n-1}\right)T^h(\psi^h), \end{aligned} \tag{8}$$

and if $\phi(x_2,y_2) < 0$, we have

$$\begin{aligned} & \int_{K^-} \left(U^h(u^{h,n+1})T^h(\psi^h) + \theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1})\cdot\nabla T^h(\psi^h)\right) \\ = & \int_{K^-} \left(2U^h(u^{h,n})T^h(\psi^h) - (1-2\theta)\Delta t^2\beta^n\nabla U^h(u^{h,n})\cdot\nabla T^h(\psi^h)\right) - \\ & \int_{K^-} \left(U^h(u^{h,n-1})T^h(\psi^h) + \theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1})\cdot\nabla T^h(\psi^h)\right) + \\ & \Delta t^2\int_{K^-} \left(\theta f^{n+1} + (1-2\theta)f^n + \theta f^{n-1}\right)T^h(\psi^h) - \\ & \Delta t^2\int_{\Gamma_K^h} \left(\theta b^{n+1} + (1-2\theta)b^n + \theta b^{n-1}\right)T^h(\psi^h). \end{aligned} \tag{9}$$

In the next two cases, the interface cells are separated into two pieces K^+ and K^-. At this time, both the Dirichlet jump condition and the Neumann jump condition need to be treated. Before getting the integral equation, we need to take care of the jump conditions first.

In **Case 3**, the local system can be constructed as follows:

$$\begin{cases} [u]_5 = & a_5, \\ [(\beta \bigtriangledown u) \cdot \boldsymbol{n}]_6 = & b_6 \end{cases}$$

where point 6 is the middle point of points 2 and 5. Solve this system to get the values of $u_5^{\pm}$, which are denoted by the linear combinations of u_1, u_2 and u_3.

In **Case 4**, notice that points $2, 3, 4, 5$ are coplanar, the value of point 5 can be denoted as a linear combination of the values of points $2, 3, 4$: $u_5 = c_1u_2 + c_2u_3 + c_3u_4$. Then a local system defined on the interface cell K can be constructed as

$$\begin{cases} [u]_4 = & a_4, \\ [u]_5 = & a_5, \\ c_1u_2 + c_2u_3 + c_3u_4 = & u_5 \\ [(\beta \bigtriangledown u) \cdot \boldsymbol{n}]_6 = & b_6 \end{cases}$$

where point 6 is the middle point of points 4 and 5. The least squares method is used to solve this system and get the values of $u_4^{\pm}, u_5^{\pm}$, they are denoted by the linear combinations of u_1, u_2 and u_3.

In cases 3 and 4, after solving the local system, we have the integral equation defined on this interface cell as

$$\begin{aligned} & \int_{K^+} \left(U^h(u^{h,n+1})T^h(\psi^h) + \theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1}) \cdot \nabla T^h(\psi^h)\right) + \\ & \int_{K^-} \left(U^h(u^{h,n+1})T^h(\psi^h) + \theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1}) \cdot \nabla T^h(\psi^h)\right) \\ = & \int_{K^+} \left(2U^h(u^{h,n})T^h(\psi^h) - (1-2\theta)\Delta t^2\beta^n\nabla U^h(u^{h,n}) \cdot \nabla T^h(\psi^h)\right) - \\ & \int_{K^-} \left(2U^h(u^{h,n})T^h(\psi^h) - (1-2\theta)\Delta t^2\beta^n\nabla U^h(u^{h,n}) \cdot \nabla T^h(\psi^h)\right) - \\ & \int_{K^+} \left(U^h(u^{h,n-1})T^h(\psi^h) + \theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1}) \cdot \nabla T^h(\psi^h)\right) + \\ & \int_{K^-} \left(U^h(u^{h,n-1})T^h(\psi^h) + \theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1}) \cdot \nabla T^h(\psi^h)\right) + \\ & \Delta t^2 \int_{K^+} \left(\theta f^{n+1} + (1-2\theta)f^n + \theta f^{n-1}\right) T^h(\psi^h) - \\ & \Delta t^2 \int_{K^-} \left(\theta f^{n+1} + (1-2\theta)f^n + \theta f^{n-1}\right) T^h(\psi^h) - \\ & \Delta t^2 \int_{\Gamma_K^h} \left(\theta b^{n+1} + (1-2\theta)b^n + \theta b^{n-1}\right) T^h(\psi^h). \end{aligned} \tag{10}$$

With the above local system, all the nonzero elements of the large sparse linear system could be generated and solved. The main idea is to write the solution on the interface cut points in terms of linear combinations of the unknowns on the uniform Cartesian grid.

We propose the following method:

Method 2.1. Find a discrete function $u^{h,n} \in H_{\pm}^{1,h}$ with $u^{h,n} = g^{h,n}$ on boundary points such that for all $\psi^h \in H_0^{1,h}$, we have

$$
\begin{aligned}
&\sum_{K\in\mathcal{T}_h}\left(\int_{K^+}\left(U^h(u^{h,n+1})T^h(\psi^h)+\theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1})\cdot\nabla T^h(\psi^h)\right)+\right.\\
&\left.\int_{K^-}\left(U^h(u^{h,n+1})T^h(\psi^h)+\theta\Delta t^2\beta^{n+1}\nabla U^h(u^{h,n+1})\cdot\nabla T^h(\psi^h)\right)\right)\\
=\;&\sum_{K\in\mathcal{T}_h}\left(\int_{K^+}\left(2U^h(u^{h,n})T^h(\psi^h)-(1-2\theta)\Delta t^2\beta^{n}\nabla U^h(u^{h,n})\cdot\nabla T^h(\psi^h)\right)-\right.\\
&\int_{K^-}\left(2U^h(u^{h,n})T^h(\psi^h)-(1-2\theta)\Delta t^2\beta^{n}\nabla U^h(u^{h,n})\cdot\nabla T^h(\psi^h)\right)-\\
&\int_{K^+}\left(U^h(u^{h,n-1})T^h(\psi^h)+\theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1})\cdot\nabla T^h(\psi^h)\right)+\\
&\int_{K^-}\left(U^h(u^{h,n-1})T^h(\psi^h)+\theta\Delta t^2\beta^{n-1}\nabla U^h(u^{h,n-1})\cdot\nabla T^h(\psi^h)\right)+\\
&\Delta t^2\int_{K^+}\left(\theta f^{n+1}+(1-2\theta)f^n+\theta f^{n-1}\right)T^h(\psi^h)-\\
&\Delta t^2\int_{K^-}\left(\theta f^{n+1}+(1-2\theta)f^n+\theta f^{n-1}\right)T^h(\psi^h)-\\
&\left.\Delta t^2\int_{\Gamma_K^h}\left(\theta b^{n+1}+(1-2\theta)b^n+\theta b^{n-1}\right)T^h(\psi^h)\right). \qquad (11)
\end{aligned}
$$

Compared with the large sparse matrix for elliptic interface problem using the same non-traditional finite element formulation, there is an extra diagonal term. The smaller the time step size compared with spacial step size, the stronger this extra diagonal term affects the matrix, the more diagonal dominance. In the end, a sufficiently small time step size would give a strictly diagonally dominant (therefore invertible) well-conditioned matrix. This is the reason why in some complicated numerical examples, we used smaller time step size to obtain more accurate result.

3. Numerical Experiments

In this section, we present some numerical examples with known exact solutions to demonstrate the accuracy of our method. In all numerical experiments below, the level-set function $\phi(x,y)$, the coefficients $\beta^{\pm}(x,y,t)$ and the solutions

$$
u=\begin{cases} u^+(x,y,t), \ \texttt{in}\ \Omega^+\times[0,T],\\ u^-(x,y,t), \ \texttt{in}\ \Omega^-\times[0,T],\end{cases}
$$

are given. Hence

$$
\begin{aligned}
u_0(x,y)&=u(x,y,0), \ \texttt{in}\ \Omega\setminus\Gamma,\\
u_{t0}(x,y)&=\frac{\partial u}{\partial t}(x,y,0), \ \texttt{in}\ \Omega\setminus\Gamma,\\
g(x,y,t)&=u(x,y,t), \ \texttt{on}\ \partial\Omega\times[0,T],\\
f(x,y,t)&=-\nabla\cdot(\beta(x,y,t)\nabla u(x,y,t)), \ \texttt{in}\ (\Omega\setminus\Gamma)\times[0,T],\\
a(x,y,t)&=u^+(x,y,t)-u^-(x,y,t), \ \texttt{on}\ \Gamma\times[0,T],\\
b(x,y,t)&=(\beta^+(x,y,t)\nabla u^+(x,y,t))\cdot\boldsymbol{n}-(\beta^-(x,y,t)\nabla u^-(x,y,t))\cdot\boldsymbol{n}, \texttt{on}\ \Gamma\times[0,T].
\end{aligned}
$$

All the examples are defined on the domain $[0,1]\times[0,1]\times[0,1]$ and the value of θ is set to be $\frac{1}{2}$. All errors in solutions are measured in the L^∞ norm and L^2 norm in the whole domain Ω .

Example 1. The level-set function $\phi(x,y)$, the coefficients $\beta^\pm$ and the solutions $u^\pm$ are given as follows:

$$\begin{aligned}
\phi_1(x,y) &= (x-0.3)^2+(y-0.3)^2-0.2^2,\\
\phi_2(x,y) &= (x-0.7)^2+(y-0.3)^2-0.2^2,\\
\phi_3(x,y) &= (x-0.5)^2+(y-\sqrt{0.12}-0.3)^2-0.21^2,\\
\phi(x,y) &= \min(\min(\phi_1,\phi_2),\phi_3),\\
\beta^+(x,y,t) &= \begin{pmatrix} 2+x^2 & y\\ y & 3+y^2t\end{pmatrix},\\
\beta^-(x,y,t) &= \begin{pmatrix} 4+x^2t & y\\ y & 3+y^2\end{pmatrix},\\
u^+(x,y,t) &= \sin(2\pi xt)\cos(3\pi yt),\\
u^-(x,y,t) &= \exp(xyt)\sin(2\pi x)\cos(2\pi y).
\end{aligned}$$

Figure 5 is the interface of this example with a grid of 24×24. The interface is a combination of three circles and it is a sharp-edged interface. The difficulty of solving this example comes from the complicated geometry of the interface.

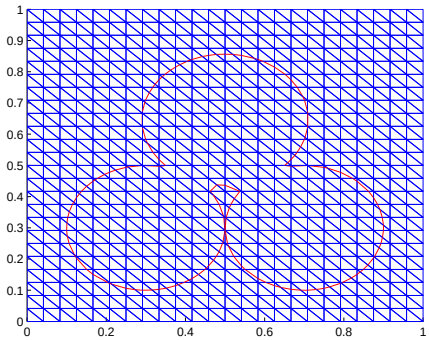
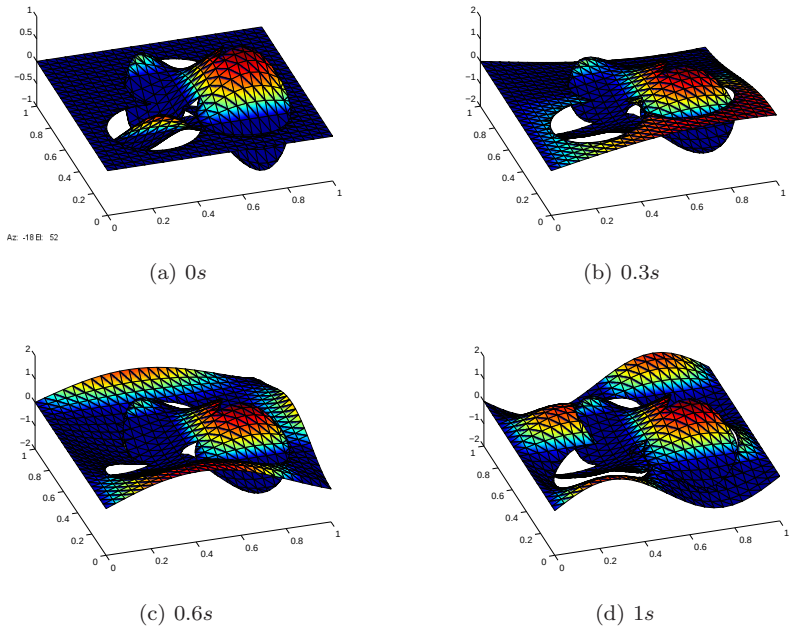


FIGURE 5. Interface of Example 1.

Figure 6 shows the numerical results of the solution at $t=0s, 0.3s, 0.6s$ and $1s$. Table 1 is the numerical errors with different grids at $t=1s$. Figure 7 shows that our method is about 2nd order accurate both in the L^∞ and L^2 norm.

TABLE 1. Example 1, numerical error with different grids at t=1s.

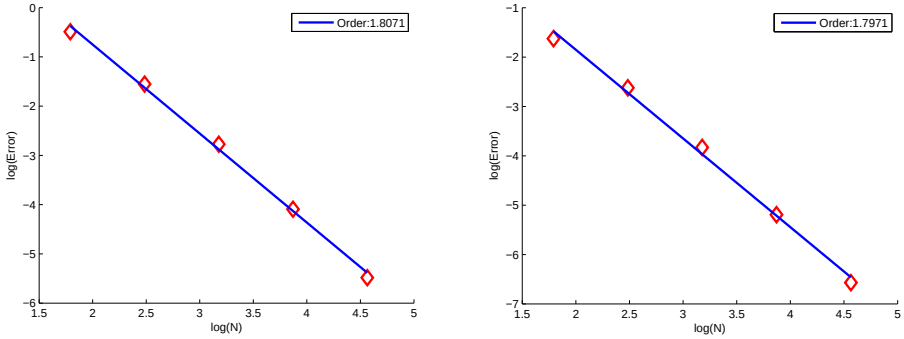
$n_x\times n_y\times n_t$	$\|u-u^h\|_\infty$	Order	$\|u-u^h\|_2$	Order
$6\times 6\times 10$	0.6124		0.1967	
$12\times 12\times 20$	0.2116	1.53	0.0726	1.44
$24\times 24\times 40$	0.0625	1.76	0.0217	1.74
$48\times 48\times 80$	0.0167	1.90	0.0056	1.97
$96\times 96\times 160$	0.0042	2.01	0.0014	1.98



(a) $0s$ (b) $0.3s$

(c) $0.6s$ (d) $1s$

FIGURE 6. Numerical solution of Example 1 at different time.



(a) Numerical errors in the L^∞ norm (b) Numerical errors in the L^2 norm

FIGURE 7. Numerical errors of Example 1.

Example 2. The level-set function $\phi(x,y)$, the coefficients $\beta^\pm$ and the solutions $u^\pm$ are given as follows:

$$\begin{aligned}
\phi(x,y) &= (x-0.5)^2+(y-0.5)^2-0.25^2,\\
\beta^+(x,y,t) &= \begin{pmatrix} 2+x & x \\ x & 3+yt \end{pmatrix},\\
\beta^-(x,y,t) &= \begin{pmatrix} 5+xt & y \\ y & 2-t+y \end{pmatrix},\\
u^+(x,y,t) &= -2\sin(2\pi x)\sin(2\pi y)\cos(2\pi t),\\
u^-(x,y,t) &= \sin(2\pi x)\sin(2\pi y)\cos(2\pi t).
\end{aligned}$$

Figure 8 shows the interface of this example with a grid of 20×20. The interface of this example is a circle. Although the interface in this example is simple, the solution function is much more complicated than other examples, which makes the simulation much harder.

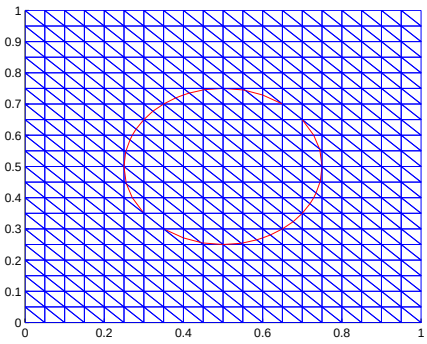
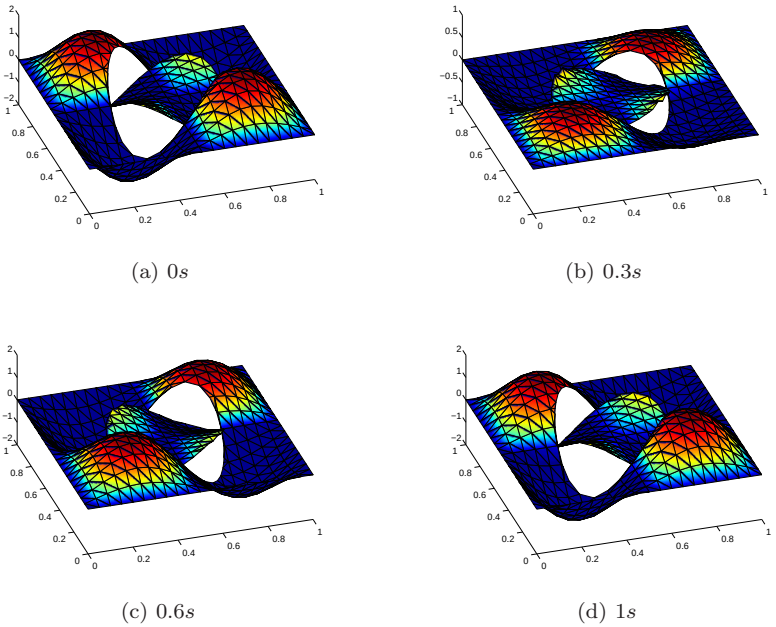


FIGURE 8. Interface of Example 2.

Figure 9 shows the numerical results of the solution at $t = 0s, 0.3s, 0.6s$ and $1s$. Table 2 shows the numerical errors with different grids at $t = 1s$. Figure 10 shows that our method is about 2nd order accurate both in the L^∞ and L^2 norm.



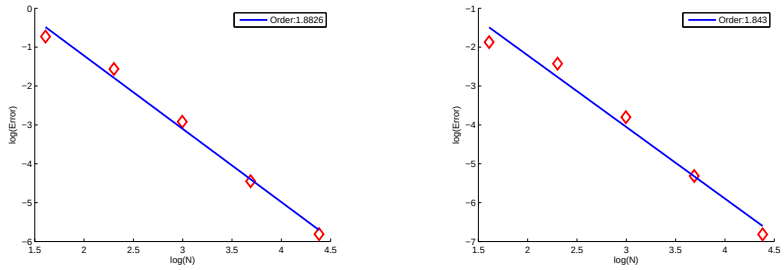
(a) $0s$ (b) $0.3s$

(c) $0.6s$ (d) $1s$

FIGURE 9. Numerical solution of Example 2 at different time.

TABLE 2. Example 2, numerical error with different grids at t=1s.

$n_x \times n_y \times n_t$	$\|u-u^h\|_\infty$	Order	$\|u-u^h\|_2$	Order
$5\times5\times40$	0.4819		0.1538	
$10\times10\times80$	0.2096	1.20	0.0883	0.80
$20\times20\times160$	0.0539	1.96	0.0223	1.99
$40\times40\times320$	0.0117	2.21	0.0049	2.18
$80\times80\times640$	0.0029	1.96	0.0011	2.16



(a) Numerical errors in the L^∞ Norm (b) Numerical errors in the L^2 Norm

FIGURE 10. Numerical errors of Example 2.

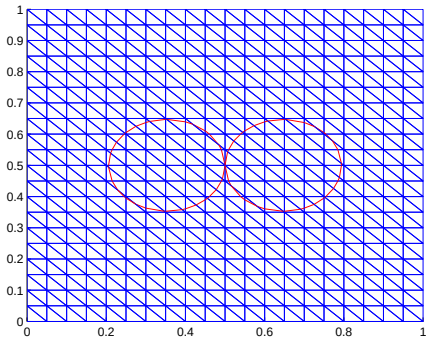


FIGURE 11. Interface of Example 3.

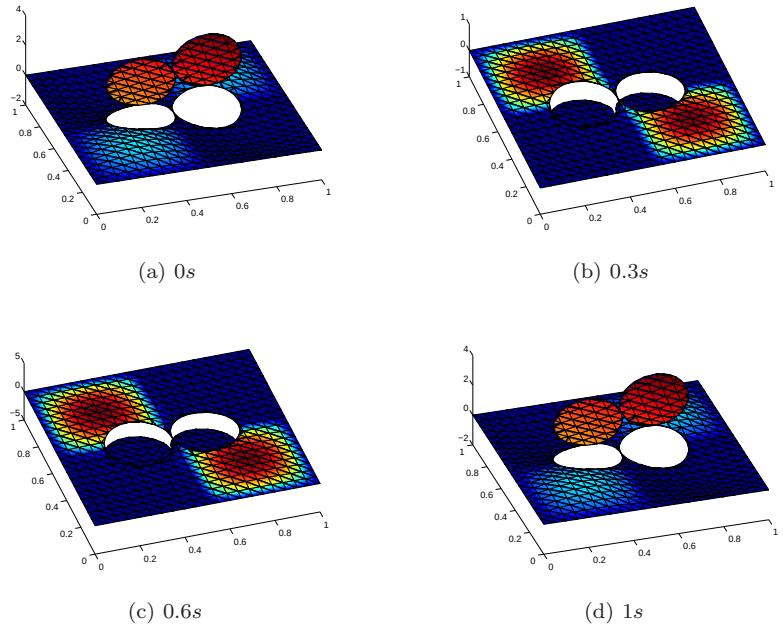
Example 3. The level-set function $\phi(x,y)$, the coefficients $\beta^\pm$ and the solutions $u^\pm$ are given as follows:

$$\begin{aligned}
\phi_1(x,y) &= (x-0.647)^2+(y-0.5)^2-0.147^2,\\
\phi_2(x,y) &= (x-0.353)^2+(y-0.5)^2-0.147^2,\\
\phi(x,y) &= \min(\phi_1,\phi_2),\\
\beta^+(x,y,t) &= 10\begin{pmatrix} 8+\cos(x) & x \\ x & 6+\sin(yt)\end{pmatrix},\\
\beta^-(x,y,t) &= \begin{pmatrix} 8+\cos(x) & x \\ x & 6+\sin(yt)\end{pmatrix},\\
u^+(x,y,t) &= 0.5\sin(2\pi x)\sin(2\pi y)\cos(2\pi t),\\
u^-(x,y,t) &= \exp(x+y)\cos(2\pi t).
\end{aligned}$$

Although the coefficients $\beta^{\pm}$ in examples 1 and 2 are variable 2×2 matrices with respect to x, y and t, the coefficients in example 3 are much more complicated, because it contains some terms with trigonometric function, and the difference of magnitude of the coefficients on interfaces is 10 times.

Figure 11 shows the interface of this example with a grid of 20×20. The interface of this example is a combination of two circles. Again, the interface in this example is a sharp-edged interface.

Figure 12 shows the numerical results of the solution at $t = 0s, 0.3s, 0.6s$ and $1s$. Table 3 shows the numerical errors with different grids at $t = 1s$. Figure 13 shows that our method is about 2nd order accurate both in the L^{∞} and L^2 norm.



(a) $0s$ (b) $0.3s$

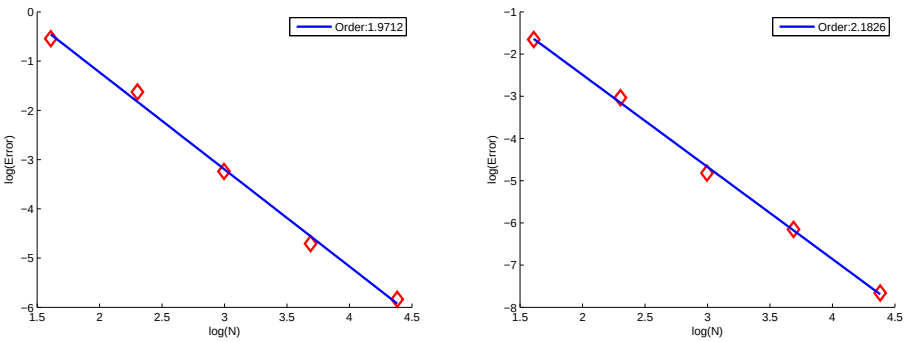
(c) $0.6s$ (d) $1s$

FIGURE 12. Numerical solution of Example 3 at different time.

TABLE 3. Example 3, numerical error with different grids at t=1s.

$n_x \times n_y \times n_t$	$\|u - u^h\|_{\infty}$	Order	$\|u - u^h\|_2$	Order
$5 \times 5 \times 20$	0.5801		0.1909	
$10 \times 10 \times 40$	0.1966	1.56	0.0481	1.99
$20 \times 20 \times 80$	0.0391	2.33	0.0081	2.58
$40 \times 40 \times 160$	0.0090	2.11	0.0021	1.92
$80 \times 80 \times 320$	0.0029	1.63	4.7093e-4	2.17

From the above examples, we can see that the solutions are smooth almost everywhere but discontinuous along the interfaces. The numerical experiments show that when the grid size gets smaller, the numerical result gets much better.



(a) Numerical errors in the L^∞ Norm (b) Numerical errors in the L^2 Norm

FIGURE 13. Numerical errors of Example 3.

4. Conclusion

We propose a non-traditional finite element method for solving the variable matrix coefficient wave equation with non-homogeneous interface jump conditions. Numerical examples justify that our method is close to 2nd order accurate both in the L^∞ norm and L^2 norm. Compared with the most recent previous work on wave equation with interface jump conditions [27], they have the advantage of achieving 4th order accuracy, however, the problem setup is much simpler than our setup. They have constant scalar coefficients while we have matrix variable coefficients. It is the simplicity of the non-traditional finite element formulation that allows us to implement the code for this type of complicated interface problem.

Acknowledgements

L. Shi's research is supported by Science Foundation of China University of Political Science and Law (No.16ZFQ11001) and National Natural Science Foundation of China (No.11701569). L. Wang's research is supported by Science Foundation of China University of Petroleum–Beijing (No.2462015BJB05). S. Hou's research is supported by Dr. Walter Koss Endowed Professorship. This professorship is made available through the State of Louisiana Board of Regents Support Funds.

References

[1] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys., 25 (1977) 220–252.
[2] R.J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal., 31 (1994) 1019-1044.
[3] Z. Li and K. Ito, The immersed interface method: Numerical solutions of pdes involving interfaces and irregular domains, SIAM, Philadelphia, 2006.
[4] Y.C. Zhou, S. Zhao, M. Feig and 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.
[5] S. Yu, Y. Zhou and G.W. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys., 224 (2007) 729-756.

[6] K. Xia, M. Zhan and G.W. Wei, The matched interface and boundary (MIB) method for multi-domain elliptic interface problems, J. Comput. Phys., 230 (2011) 8231-8258.
[7] Z. Li, The immersed interface method using a finite element formulation, Appl. Numer. Math., 27 (1998) 253-267.
[8] Z. Li, T. Lin, Y. Lin and R. Rogers, An immersed finite element space and its approximation capability, Numer. Methods Partial Differential Equations, 20 (2004) 338-367.
[9] X. He, T. Lin and Y. Lin, Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions, Int. J. Numer. Anal. Model., 8 (2011) 284-301.
[10] T. Lin, Y. Lin and X. Zhang, Partially penalized immersed finite element methods for elliptic interface problems, SIAM J. Numer. Anal., 53 (2015) 1121-1144.
[11] P. Colella and H. Johansen, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys., 60 (1998) 85-147.
[12] M. Oevermann and R. Klein, A Cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces, J. Comput. Phys., 219 (2006) 749-769.
[13] M. Oevermann, C. Scharfenberg and R. Klein, A sharp interface finite volume method for elliptic equations on Cartesian grids, J. Comput. Phys., 228 (2009) 5184-5206.
[14] G. Guyomarch and C. Lee, A discontinuous Galerkin method for elliptic interface problems with application to electroporation, Commun. Numer. Meth. Engng., 25 (2009) 991–1008.
[15] R. Massjung, An hp-error estimate for an unfitted discontinuous Galerkin method applied to elliptic interface problems, RWTH 300, IGPM Report, 2009.
[16] Z. Chen and J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math., 79 (1996) 175-202.
[17] J. Huang and J. Zou, Some new A priori estimates for second-Order elliptic and parabolic interface problems, J. Diff. Eqns., 184 (2002) 570-586.
[18] S. Hou and X. Liu, A numerical method for solving variable coefficient elliptic equations with interfaces, J. Comput. Phys., 202 (2005) 411-445.
[19] S. Hou, W. Wang and L. Wang, Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces, J. Comput. Phys., 229 (2010) 7162-7179.
[20] L. Wang, S. Hou and L. Shi, An Improved Non-traditional Finite Element Formulation for Solving the Elliptic Interface Problems, J. Comput. Math., 32 (2014) 39-57.
[21] S. Hou, P. Song, L. Wang and H. Zhao, A weak formulation for solving elliptic interface problems without body fitted grid, J. Comput. Phys., 249 (2013) 80-95.
[22] S. Hou, Z. Li, L. Wang and W. Wang, A numerical method for solving elasticity equations with interfaces, Commun. Comput. Phys., 12 (2012) 595-612.
[23] S. Hou, L. Wang and W. Wang, A numerical method for solving the elliptic interface problem with multi-domains and triple junction points, J. Comput. Math., 30 (2012) 504-516.
[24] L. Wang, S. Hou, L. Shi and J. Solow, A numerical method for solving elasticity equations with interface involving multi-domain and triple junction points, Appl. Math. Comput., 251 (2015) 615-625.

[25] L. Wang, S. Hou and L. Shi, A weak formulation for solving elliptic interface problems with imperfect contact, Appl. Math. Mech., 9 (2017) 1189-1205.
[26] C. Zhang, Immersed interface methods for hyperbolic systems of partial differential equations with discontinuous coefficients, PhD Dissertation, University of Washington, 1996.
[27] D. Abraham, A. Marques and J.-C. Nave, A correction function method for the wave equation with interface jump conditions, J. Comput. Phys., 353 (2018) 281-299.
[28] S. Karaa, Finite element θ-schemes for the acoustic wave equation, Adv. Appl. Math. Mech., 3 (2011) 181-203.
[29] R. Li and G. Feng, The numerical solution of differential equation, Beijing, Higher Education Press, 1997.
[30] H. Guo and X. Yang, Gradient recovery for elliptic interface problem: I. body-fitted mesh, arXiv:1607.05898 [math.NA], 2016.
[31] H. Guo and X. Yang, Gradient recovery for elliptic interface problem: II. immersed finite element methods, J. Comput. Phys., 338 (2017) 606-619.
[32] H. Guo and X. Yang, Polynomial preserving recovery for high frequency wave propagation, J. Sci. Comput., 71 (2017) 594-614.

Dept of Mathematics, College of Science, China University of Petroleum (Beijing), 102249, P.R.China
E-mail: wliqunhmily@gmail.com

Department of Mathematics and Statistics and Center of Applied Physics, Louisiana Tech University
E-mail: shou@latech.edu

Dept of Science and Technology Teaching, China University of Political Science and Law, Beijing, 102249, P.R.China
E-mail: sliweihmily@gmail.com

Dept of Mathematics, College of Science, China University of Petroleum (Beijing), 102249, P.R.China
E-mail: zhangpingby@126.com