

## A Simple FEM for Solving Two-Dimensional Diffusion Equation with Nonlinear Interface Jump Conditions

Liqun Wang<sup>1</sup>, Songming Hou<sup>2</sup> and Liwei Shi<sup>3,\*</sup>

**Abstract:** In this paper, we propose a numerical method for solving parabolic interface problems with nonhomogeneous flux jump condition and nonlinear jump condition. The main idea is to use traditional finite element method on semi-Cartesian mesh coupled with Newton's method to handle nonlinearity. It is easy to implement even though variable coefficients are used in the jump condition instead of constant in previous work for elliptic interface problem. Numerical experiments show that our method is about second order accurate in the  $L^\infty$  norm.

**Keywords:** Finite element method, parabolic equation, interface problems, nonlinear jump conditions.

### 1 Introduction

Parabolic problems with nonhomogeneous flux jump condition and nonlinear jump condition are often encountered in scientific computing and engineering. For example, in Radu et al. [Radu, Meir and Bakker (2004); Hetzer and Meir (2007); Bakker and Meir (2003)], a numerical model is introduced to describe the concentration  $u$  of an ion  $I$  in aqueous solution  $\Omega_{aq}$  and adjoining polymeric membrane  $\Omega_{org}$ . Since the diffusion coefficients of the aqueous solution and of the membrane are different, this is an interface problem with interface  $\Gamma$  at the membrane. The equations for the problem are as follows:

$$u_t - (ku_x)_x = f, \text{ in } \Omega \times (0, T) \quad (1)$$

where  $\Omega = \Omega_{aq} \cup \Omega_{org}$ . The diffusion coefficient is given by

$$k = \begin{cases} k_{aq}, & \text{in } \Omega_{aq} \\ k_{org}, & \text{in } \Omega_{org}. \end{cases}$$

---

<sup>1</sup>China University of Petroleum-Beijing, Department of Mathematics, College of Science, Beijing, 102249, China.

<sup>2</sup>Louisiana Tech University, Department of Mathematics and Statistics, LA, Ruston, 71272, USA.

<sup>3</sup>China University of Political Science and Law, Department of Science and Technology, Beijing, 102249, China.

\*Corresponding Author: Liwei Shi. Email: sliweihmily@gmail.com.

The ion flux continuity across the interface gives a flux jump condition

$$k_{aq}u_{aq,x} - k_{org}u_{org,x} = 0. \quad (2)$$

The analyte ion  $I$  exchanges with ion  $J$  of the same valency at the interface, therefore

$$u_{R,org}u_{aq} - K_{I,J}^{pot}u_{J,aq}u_{org} = u_{aq}u_{org},$$

which can be rewritten as the nonlinear jump condition

$$\sigma_{aq}u_{aq} - \sigma_{org}u_{org} = \sigma u_{aq}u_{org}. \quad (3)$$

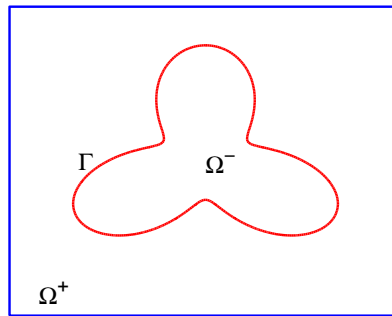
Given proper boundary condition and initial condition, the above mentioned problem is a one-dimensional nonlinear interface problem with homogeneous flux jump condition. The numerical solution of interface problems has attracted much attention in recent years. The numerical method using non-body-fitted grids for elliptic interface problem was first proposed by Peskin [Peskin (1977)]. After that, various numerical methods are developed, including the immersed interface method (IIM) [LeVeque and Li (1994); Li and Ito (2006); Li (1998)], the immersed finite element method(IFEM) [Li, Lin and Wu (2003); He, Lin and Lin (2011); Gong, Li and Li (2008)] and its partially penalized version [Lin, Lin and Zhang (2015)], the immersed finite volume element method(IFVE) [Ewing, Li, Lin et al. (1999)], the matched interface and boundary method (MIB) [Xia, Zhan and Wei (2011); Yu, Zhou and Wei (2007); Zhou, Zhao, Feig et al. (2006)], the non-traditional finite element method(NTFEM) [Hou, Wang and Wang (2010); Hou, Song, Wang et al. (2013); Hou, Li, Wang et al. (2012)], the weak Galerkin method [Mu, Wang, Wei et al. (2013)], the kernel-free boundary integral (KFBI) method [Ying and Henriquez (2007)], and so on. Gradient recovery technique [Guo and Yang (2018, 2017)] is applied to improve the accuracy of the gradient.

In this paper, we propose a numerical method for a much extended scope of problems, which are two-dimensional parabolic interface problems with nonhomogeneous flux jump condition and nonlinear jump condition. Motivated by the numerical model described by Eqs. (1)-(3), and inspired by the traditional finite element method(TFEM) for solving elliptic interface problem we proposed in Wang et al. [Wang and Shi (2014)], we propose a new numerical method for solving the two-dimensional parabolic interface problem with nonhomogeneous flux jump condition and nonlinear jump condition. We use traditional finite element method coupled with Newton's method to deal with these two special kinds of jump conditions and use Crank-Nicolson scheme to deal with the time derivative. The reason we did not use the NTFEM is because the nonlinear term of this problem is special. It comes from the jump condition. Normally it is effective to use Newton's method to linearize the nonlinear system. However, when the nonlinearity comes from the jump condition along the interface, we need to solve the nonlinear system at the local cell for a time step if we use the NTFEM, and the result of this system will be a linear combination of the unknown solution located at the vertices of the element cell. Newton's method need to have an initial entry at every iteration step. However, the NTFEM can only provide an unknown linear combination, which will cause a serious problem at the linearization part. That is why we choose to use the TFEM. Compared with our recent work on elliptic

interface problem with nonlinear jump condition in [Wang, Hou and Shi (2017)], the new contributions include: first, for the parabolic interface problem we need to use Crank-Nicolson’s method to handle the time derivative term. This needs to be done carefully with presence of interface. Second, in our previous work, the coefficients  $\sigma$ ,  $\sigma^+$  and  $\sigma^-$  are all constants for simplicity. In this paper, all these are assumed to be variables depending on both time and space, which makes the nonlinear scheme more difficult to derive. Numerical examples show that our method is very effective and stable, it is efficient for matrix coefficient and sharp-edged interface, and can achieve about second order accuracy in the  $L^\infty$  norm.

**2 Mathematical formulation**

In this paper, we consider a parabolic interface problem defined on an open bounded domain  $\Omega \subset R^2$  cross with a time interval  $[0, T]$ . Let  $\Gamma$  be an interface that divides  $\Omega$  into disjoint open subdomains,  $\Omega^-$  and  $\Omega^+$ , hence  $\Omega = \Omega^- \cup \Omega^+ \cup \Gamma$ . Assume that the boundary  $\partial\Omega$  and the boundary of each subdomain  $\partial\Omega^\pm$  are Lipschitz continuous as submanifolds. Since  $\partial\Omega^\pm$  are Lipschitz continuous, so is  $\Gamma$ . Therefore a unit normal vector of  $\Gamma$  can be defined a.e. on  $\Gamma$ , see Fig. 1.



**Figure 1:** Setup of the problem

We seek solutions of the variable coefficient parabolic problem away from the interface  $\Gamma$  given by

$$u_t(\vec{x}, t) - \nabla \cdot (\beta(\vec{x}, t) \nabla u(\vec{x}, t)) = f(\vec{x}, t), \text{ in } \Omega \times (0, T], \tag{4}$$

in which  $\vec{x} = (x, y)$  denotes the spatial variables and  $\nabla$  is the gradient operator. The coefficient  $\beta(\vec{x}, t)$  is assumed to be a  $2 \times 2$  matrix that is uniformly elliptic on each disjoint subdomain,  $\Omega^-$  and  $\Omega^+$ , and its components are continuously differentiable on each disjoint subdomain, but they may be discontinuous across the interface  $\Gamma$ . The right-hand side  $f(\vec{x}, t)$  is assumed to lie in  $L^2(\Omega)$ . Note that  $f$  is piecewise defined on two sides of the interface.

The flux jump condition along the interface  $\Gamma$  is given by

$$b(\vec{x}, t) = \vec{n} \cdot (\beta^+(\vec{x}, t) \nabla u^+(\vec{x}, t)) - \vec{n} \cdot (\beta^-(\vec{x}, t) \nabla u^-(\vec{x}, t)), \tag{5}$$

and the nonlinear jump condition along the interface  $\Gamma$  is prescribed by

$$\sigma^+(\vec{x}, t)u^+(\vec{x}, t) - \sigma^-(\vec{x}, t)u^-(\vec{x}, t) = \sigma(\vec{x}, t)u^+(\vec{x}, t)u^-(\vec{x}, t). \quad (6)$$

The "  $\pm$  " superscripts refer to limits taken from within the subdomains  $\Omega^\pm$ .

Finally, we prescribe initial and boundary conditions

$$\begin{aligned} u(\vec{x}, 0) &= u_0(\vec{x}), \text{ in } \Omega, \\ u(\vec{x}, t) &= g(\vec{x}, t), \text{ on } \partial\Omega \times (0, T], \end{aligned} \quad (7)$$

for a given function  $g$  on the boundary  $\partial\Omega$  and  $u_0$  on the domain  $\Omega$ .

In order to derive the weak formulations of the mathematical model defined by Eqs. (4)-(7), we need to define some notations as in Chen et al. [Chen and Zou (1998); Huang and Zou (2002)]. For a Banach space  $B$ , define

$$H^m(0, T; B) = \{u(t) \in B \text{ for a.e. } t \in (0, T) \text{ and } \|u\|_{H^m(0, T; B)} < \infty\},$$

where  $\|u\|_{H^m(0, T; B)}$  is the norm of  $H^m(0, T; B)$  and is given by

$$\|u\|_{H^m(0, T; B)} = \left\{ \sum_{k=0}^m \int_0^T \|u^{(k)}(t)\|_B^2 dt \right\}^{1/2},$$

and

$$L^2(0, T; B) = H^0(0, T; B).$$

Now we are ready to derive the weak formulation of Eq. (4). Using the traditional finite element method, with the standard procedure of multiplying by a test function  $\psi$  and integrating by parts, we deduce the weak solution:

$$\int_{\Omega} u_t \psi + \int_{\Omega} \beta \nabla u \cdot \nabla \psi = \int_{\Omega} f \psi - \int_{\Gamma} b \psi, \quad (8)$$

where  $\psi$  is in  $H_0^1(\Omega)$ .

Below the weak formulation is employed to solve the parabolic interface problem. First divide the time interval  $[0, T]$  into  $n_t$  equally spaced subintervals  $[t_{n-1}, t_n]$ ,  $n = 1, 2, \dots, n_t$ , with  $t_n = n\Delta t$ ,  $\Delta t = T/n_t$ . According to the Crank-Nicolson method in Wang et al. [Wang and Shi (2015)], the following semidiscrete problem of Eq. (4) is derived:

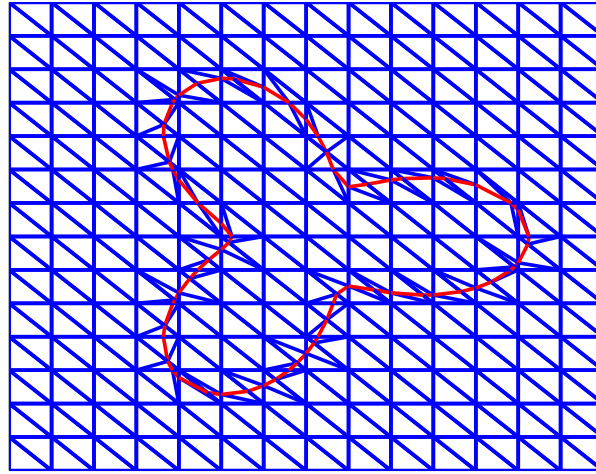
$$\begin{aligned} \frac{u(\vec{x}, t_{n+1}) - u(\vec{x}, t_n)}{\Delta t} - \nabla \cdot \left( \beta(\vec{x}, t_{n+1/2}) \nabla \frac{u(\vec{x}, t_{n+1}) + u(\vec{x}, t_n)}{2} \right) &= f(\vec{x}, t_{n+1/2}), \\ n &= 1, 2, \dots, n_t. \end{aligned} \quad (9)$$

Further, the weak formulation of the semidiscrete problem reads

$$\begin{aligned}
 & \int_{\Omega} u^{n+1} \psi + \frac{\Delta t}{2} \int_{\Omega} \left( \beta^{n+1/2} \nabla u^{n+1} \cdot \nabla \psi \right) \\
 = & \int_{\Omega} u^n \psi - \frac{\Delta t}{2} \int_{\Omega} \left( \beta^{n+1/2} \nabla u^n \cdot \nabla \psi \right) + \Delta t \int_{\Omega} f^{n+1/2} \psi - \Delta t \int_{\Gamma} b^{n+1/2} \psi, \\
 & \forall \psi \in H_0^1(\Omega),
 \end{aligned} \tag{10}$$

### 3 Numerical method

In this section, a numerical method is implemented to solve the nonlinear system described in Eqs. (4)-(7). The grid used in this paper is the semi-Cartesian grid introduced in Wang et al. [Wang and Shi (2014); Wang, Hou and Shi (2017); Xie, Li and Qiao (2011); Wang, Li and Lubkin (2014)], see Fig. 2. Let  $\phi(x, y)$  be a level set function. For simplicity, we use  $\phi_i$  to represent  $\phi(x_i, y_i)$  in the following discussion, where  $x_i, y_i$  is a grid point.



**Figure 2:** A schematic illustration of semi-Cartesian triangulation

For any cell  $T_{o,p,q}$ ,  $o, p, q \in \{1, 2, \dots, N\}$ , where  $N$  is the number of grid points, we define two sets below:

$$\mathcal{T}_h^+ = \{T_{o,p,q} \in \Omega^+ : \exists i \in \{o, p, q\}, \phi_i > 0\},$$

$$\mathcal{T}_h^- = \{T_{o,p,q} \in \Omega^- : \exists i \in \{o, p, q\}, \phi_i < 0\}.$$

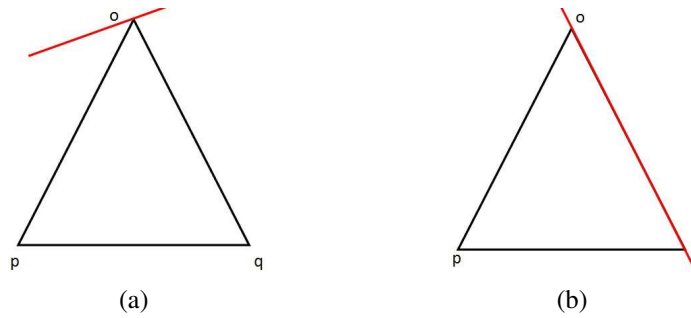
Then the body-fitting triangulation  $\mathcal{T}_h$  of the domain  $\Omega$  is obtained as  $\mathcal{T}_h = \mathcal{T}_h^+ \cup \mathcal{T}_h^-$ . The set  $\mathcal{T}_\Gamma$  of interface cells is defined by

$$\mathcal{T}_\Gamma = \{T_{o,p,q} \in \mathcal{T}_h^- : \exists i \in \{o, p, q\}, \phi_i = 0\},$$

and  $\Gamma_h$  is used to define the interface segment on the interface cell  $T$ . The set  $\mathcal{T}_R$  of regular cells is defined by

$$\mathcal{T}_R = \mathcal{T}_h \setminus \mathcal{T}_\Gamma.$$

Note that for body-fitted grid the interface does not cut through the triangle, it only touches one vertex or one side of the triangle. Therefore there are only two kinds of interface cells, see Fig. 3. It is this desirable property that ensures our method straightforward to implement.



**Figure 3:** Interface cells

Now we introduce the canonical piecewise linear finite element space  $X_h$  on the triangulation  $\mathcal{T}_h$ :

$$X_h = \{v \in C(\Omega) : v|_\tau \in P_1(\tau) \forall \tau \in \mathcal{T}_h\},$$

where  $P_1(\tau)$  denotes the space of linear functions on  $\tau$ .

We shall construct an approximate solution to the interface problem by using the space  $X_h$ , but taking into account the jump conditions. First note that the flux jump condition  $[(\beta \nabla u) \cdot n] = b$  along the interface  $\Gamma$  is already absorbed into the weak formulation. Hence, the most important and challenging part is the nonlinear jump condition prescribed in Eq. (6). The approximate solution is piecewise linear on both subdomains  $\Omega^-$  and  $\Omega^+$ , but discontinuous along the interface  $\Gamma$ .

For grid points that are involved in two different solutions, like point  $o$  in Fig. 3(a) or points  $o, q$  in Fig. 3(b), we have the following definitions:

for any cell  $T \in \mathcal{T}_h$ , if  $T \in \mathcal{T}_h^+$ , then

$$u_i = u_i^+,$$

and if  $T \in \mathcal{T}_h^-$ , then

$$u_i = \begin{cases} u_i^-, & \text{if } \phi_i \neq 0, \\ \frac{\sigma^+ u_i^+}{\sigma^- + \sigma u_i^+}, & \text{if } \phi_i = 0, \end{cases} \quad (11)$$

Next we have notations defined as below [He, Lin and Lin (2011); Wang, Hou and Shi (2017)]:

$$\begin{aligned}\mathcal{N}_h &= \{(x_i, y_i) \in R^2 \mid (x_i, y_i) \text{ is a node of } \mathcal{T}_h\}, \\ \mathcal{N}_h^{it} &= \{(x_i, y_i) \in R^2 \mid (x_i, y_i) \text{ is a node of } \mathcal{T}_h^- \text{ and } \phi_i = 0\}, \\ \mathcal{N}_h^{oi} &= \mathcal{N}_h^{it} \cap \Omega, \\ \mathcal{N}_h^{in} &= \mathcal{N}_h \cap \Omega, \\ \mathcal{N}_h^{bo} &= \mathcal{N}_h \cap \partial\Omega.\end{aligned}$$

Also,  $i \in \mathcal{N}$  is used to denote  $(x_i, y_i) \in \mathcal{N}$ .

With these notations, a globally piecewise linear approximation  $u_h$  can be defined as

$$u_h = \sum_{i \in \mathcal{N}_h^{in}} u_i \psi_i + \sum_{i \in \mathcal{N}_h^{bo}} g_i \psi_i$$

Before deriving the discrete formulation for the weak Eq. (10), for simplicity consideration, let

$$\begin{aligned}I_{u,in}^n &= \sum_{i \in \mathcal{N}_h^{in}} \left( \sum_{T \in \mathcal{T}_h} \int_T \psi_j \psi_i \right) u_i^n, \\ I_{u,bo}^n &= \sum_{i \in \mathcal{N}_h^{bo}} \left( \sum_{T \in \mathcal{T}_h} \int_T \psi_j \psi_i \right) g_i^n, \\ I_{\nabla u,in}^n &= \sum_{i \in \mathcal{N}_h^{in}} \left( \sum_{T \in \mathcal{T}_h} \int_T \beta^{n+1/2} \nabla \psi_j \cdot \nabla \psi_i \right) u_i^n, \\ I_{\nabla u,bo}^n &= \sum_{i \in \mathcal{N}_h^{bo}} \left( \sum_{T \in \mathcal{T}_h} \int_T \beta^{n+1/2} \nabla \psi_j \cdot \nabla \psi_i \right) g_i^n, \\ I_b &= \sum_{T \in \mathcal{T}_h} \int_{\Gamma_h} b^{n+1/2} \psi_j, \\ I_f &= \sum_{T \in \mathcal{T}_h} \int_T f^{n+1/2} \psi_j.\end{aligned}$$

Then we have the following method:

**Method 3.1.** Find a solution  $u_h$  such that for all  $\psi_j \in X_h$ , we have

$$\begin{aligned} & I_{u,in}^{n+1} + I_{u,bo}^{n+1} + \frac{\Delta t}{2} \left( I_{\nabla u,in}^{n+1} + I_{\nabla u,bo}^{n+1} \right) \\ = & I_{u,in}^n + I_{u,bo}^n - \frac{\Delta t}{2} \left( I_{\nabla u,in}^n + I_{\nabla u,bo}^n \right) - \Delta t I_b + \Delta t I_f \end{aligned} \quad (12)$$

Although we can get this discrete Eq. (12), it is a nonlinear equation. In this paper, we use the Newton's method to linearize and solve the problem. Unlike other nonlinear problems having a nonlinear term with specific definition in the governing equation defined on the whole domain, the nonlinear part of this problem comes from a local place: it located at the interface. Therefore we need to separate the interface points from the interior points. Then the nonlinear term needs to be coupled in the governing equation such that we can use Newton's method to linearize the system. Since  $\sigma$  is nonzero in this paper, the solution jump is nonhomogeneous, that means there will be two solutions  $u^\pm$  and two governing equations defined on the point located at the interface. When we are calculating the Jacobian matrix of the nonlinear system, we need to figure out it is with respect to which equations and which unknowns. This is a tricky issue for the linearization procedure.

Based on the above discussion, and take the nonlinear jump condition into consideration, define

$$\begin{aligned} I_{u,in \setminus oi}^n &= \sum_{i \in \mathcal{N}_h^{in} \setminus \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \psi_j \psi_i \right) u_i^n, \\ I_{u,oi}^n &= \sum_{i \in \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \psi_j \psi_i \right) \frac{\sigma^{+,n+1/2} u_i^n}{\sigma^{-,n+1/2} + \sigma^{n+1/2} u_i^n}, \\ I_{\nabla u,in \setminus oi}^n &= \sum_{i \in \mathcal{N}_h^{in} \setminus \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \beta^{n+1/2} \nabla \psi_j \cdot \nabla \psi_i \right) u_i^n \\ I_{\nabla u,oi}^n &= \sum_{i \in \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \beta^{n+1/2} \nabla \psi_j \cdot \nabla \psi_i \right) \frac{\sigma^{+,n+1/2} u_i^n}{\sigma^{-,n+1/2} + \sigma^{n+1/2} u_i^n}, \end{aligned}$$



then we have the following residual equation:

$$\begin{aligned}
 & R(u) \\
 = & I_{u,in}^{n+1} + I_{u,bo}^{n+1} + \frac{\Delta t}{2} \left( I_{\nabla u,in}^{n+1} + I_{\nabla u,bo}^{n+1} \right) \\
 & - I_{u,in}^n - I_{u,bo}^n + \frac{\Delta t}{2} \left( I_{\nabla u,in}^n + I_{\nabla u,bo}^n \right) + \Delta t I_b - \Delta t I_f \\
 = & I_{u,in \setminus oi}^{n+1} + I_{u,oi}^{n+1} + I_{u,bo}^{n+1} + \frac{\Delta t}{2} \left( I_{\nabla u,in \setminus oi}^{n+1} + I_{\nabla u,oi}^{n+1} + I_{\nabla u,bo}^{n+1} \right) \\
 & - I_{u,in \setminus oi}^n - I_{u,oi}^n - I_{u,bo}^n + \frac{\Delta t}{2} \left( I_{\nabla u,in \setminus oi}^n + I_{\nabla u,oi}^n + I_{\nabla u,bo}^n \right) \\
 & + \Delta t I_b - \Delta t I_f.
 \end{aligned} \tag{13}$$

Hence the iteration method can be written as

$$\frac{\partial R(u^l)}{\partial u^l} \delta u^l = -R(u^l), \tag{14}$$

where  $l$  refers to the iteration step of the Newton's method,  $u^l$  is the unknown vector, such that  $u^{l+1} = u^l + \delta u^l$ , and  $\frac{\partial R(u^l)}{\partial u^l}$  is the Jacobian matrix.

Let

$$\begin{aligned}
 I_{u,in \setminus oi}^\delta &= \sum_{i \in \mathcal{N}_h^{in} \setminus \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \psi_j \psi_i \right) \delta u_i^l, \\
 I_{u,oi}^\delta &= \sum_{i \in \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \psi_j \psi_i \frac{\sigma^{-,n+1/2} \sigma^{+,n+1/2}}{(\sigma^{-,n+1/2} + \sigma^{+,n+1/2} u^l)^2} \right) \delta u_i^l, \\
 I_{\nabla u,in \setminus oi}^\delta &= \sum_{i \in \mathcal{N}_h^{in} \setminus \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \beta^{n+1/2} \nabla \psi_j \cdot \nabla \psi_i \right) \delta u_i^l, \\
 I_{\nabla u,oi}^\delta &= \sum_{i \in \mathcal{N}_h^{oi}} \left( \sum_{T \in \mathcal{T}_h} \int_T \beta^{n+1/2} \nabla \psi_j \cdot \nabla \psi_i \frac{\sigma^{-,n+1/2} \sigma^{+,n+1/2}}{(\sigma^{-,n+1/2} + \sigma^{+,n+1/2} u^l)^2} \right) \delta u_i^l,
 \end{aligned}$$

with the notations defined above, and from Eqs. (13) and (14), we have the following method:

**Method 3.2.** Find a solution  $u_h$  such that for all  $\psi_j \in X_h$ , we have

$$\begin{aligned} & I_{u, in \setminus oi}^\delta + I_{u, oi}^\delta + \frac{\Delta t}{2} \left( I_{\nabla u, in \setminus oi}^\delta + I_{\nabla u, oi}^\delta \right) \\ = & -I_{u, in \setminus oi}^l - I_{u, oi}^l - I_{u, bo}^{n+1} - \frac{\Delta t}{2} \left( I_{\nabla u, in \setminus oi}^l + I_{\nabla u, oi}^l + I_{\nabla u, bo}^{n+1} \right) \\ & + I_{u, in \setminus oi}^n + I_{u, oi}^n + I_{u, bo}^n - \frac{\Delta t}{2} \left( I_{\nabla u, in \setminus oi}^n + I_{\nabla u, oi}^n + I_{\nabla u, bo}^n \right) \\ & - \Delta t I_b + \Delta t I_f. \end{aligned}$$

**Remark 3.1.** In our implementation, the integrals are computed with Gaussian quadrature rules. For each triangular cell, the midpoint of each edge is denoted by  $p_{ij}$ . In numerical computation, the average of three  $f(p_{ij})$  in each cell is utilized.

#### 4 Numerical experiments

In this section, we use three specific examples to testify the efficiency of our method (Method 3.2) for solving the two-dimensional diffusion equations with nonlinear interface jump conditions.

In all numerical experiments below, the level-set function  $\phi$ , the coefficients  $\beta^\pm$ ,  $\sigma^\pm$ ,  $\sigma$ , and the solution  $u^+$  in  $\Omega^+$  are given. Hence

$$\begin{aligned} u^- &= \frac{\sigma^+ u^+}{\sigma^- + \sigma u^+}, \text{ in } \Omega^-, \\ f &= -\nabla \cdot (\beta \nabla u), \text{ in } \Omega, \\ b &= (\beta^+ \nabla u^+) \cdot \vec{n} - (\beta^- \nabla u^-) \cdot \vec{n}, \text{ on } \Gamma. \end{aligned}$$

Since the solutions are given,  $g$  and  $u_0$  are obtained as a proper Dirichlet boundary condition and an initial condition. The right hand side functions of the PDEs are evaluated using the left hand side and the true solution.

All the examples are defined on the domain  $[-1, 1] \times [-1, 1] \times (0, 1]$ .

The termination condition for the Newton iteration method in this paper is  $\frac{\min(\Delta x, \Delta y)}{10^9}$ .

All errors in solutions are measured in the  $L^\infty$  norm in the whole domain  $\Omega$ .

**Example 1.** The level-set function  $\phi$ , the coefficients  $\beta^\pm$ ,  $\sigma^\pm$ ,  $\sigma$  and the solution  $u^+$  are

given as:

$$\begin{aligned}\phi(x, y) &= x^2 + \frac{y^2}{5} - 0.4^2, \\ \beta^+(x, y, t) &= \begin{pmatrix} \sin(xyt) + 3 & 0 \\ 0 & xyt + 6 \end{pmatrix}, \\ \beta^-(x, y, t) &= \begin{pmatrix} x^2 + y^2 + t^2 + 1 & 0 \\ 0 & \cos(xy) + 3t + 6 \end{pmatrix}, \\ \sigma^+(x, y, t) &= x^2 + 2y + 3, \\ \sigma^-(x, y, t) &= \sin^2(y) + x + 2, \\ \sigma(x, y, t) &= \cos^2(x + y) + 5, \\ u^+(x, y, t) &= (x + 5yt)^2.\end{aligned}$$

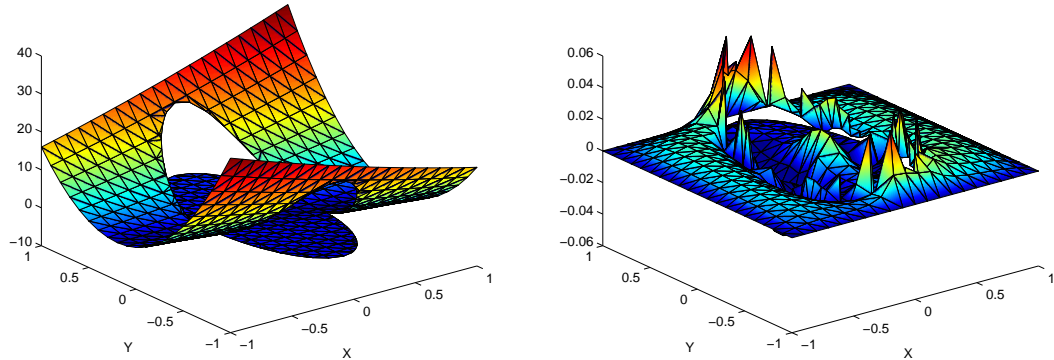
In this example, the coefficients  $\beta^\pm$  are diagonal matrices,  $\sigma^\pm$  and  $\sigma$  are variable coefficients depending on the space. Fig. 4(a) and Fig. 4(b) show respectively the numerical solution and the error at  $T = 1$  with our method using a  $20 \times 20 \times 10$  grid. Fig. 4(c) shows the geometry of the level set function and a semi-Cartesian grid with 20 grid points in both  $x$  and  $y$  directions. Tab. 1 gives the errors on different grids. Numerically, the method is second order accurate and the Newton's iteration steps remain less than 7, which demonstrates the stability of our method.

**Table 1:** Numerical results for Example 1

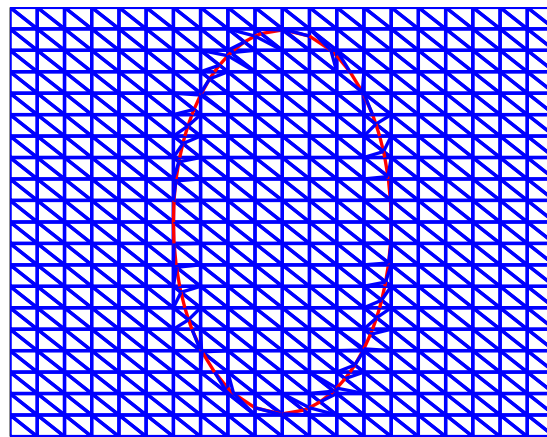
$n_x \times n_y \times n_t$	error in $u$	order	iteration steps
$16 \times 16 \times 5$	2.34e-1		6
$32 \times 32 \times 10$	5.38e-2	2.12	5
$64 \times 64 \times 20$	1.22e-2	2.14	5
$128 \times 128 \times 40$	2.99e-3	2.03	5

In the next two examples, the coefficients  $\beta^\pm$  are symmetric matrices,  $\sigma^\pm$  and  $\sigma$  are variable coefficients depending on space and time.

**Example 2.** The level-set function  $\phi$ , the coefficients  $\beta^\pm$ ,  $\sigma^\pm$ ,  $\sigma$  and the solution  $u^+$  are



(a) Numerical result on a  $20 \times 20 \times 10$  semi-Cartesian grid at  $T = 1$  (b) Error on a  $20 \times 20 \times 10$  semi-Cartesian grid at  $T = 1$



(c) Semi-Cartesian grid:  $20 \times 20$

**Figure 4:** Numerical results for Example 1

given as:

$$\begin{aligned} \phi(x, y) &= (x + 1)^2 + (y + 1)^2 - 1.5, \\ \beta^+(x, y, t) &= \begin{pmatrix} 4 + \sin^2 x + yt & xt \\ xt & 3 + x^2 + y^2 + t^2 \end{pmatrix}, \\ \beta^-(x, y, t) &= \begin{pmatrix} \cos^2(x + y + t) + 3 & yt \\ yt & 2x^2y^2t + 2 \end{pmatrix}, \\ \sigma^+(x, y, t) &= x^2y^2t + 3, \\ \sigma^-(x, y, t) &= \sin^2(xyt) + 1, \\ \sigma(x, y, t) &= x + y + t + 3, \\ u^+(x, y, t) &= \sin(x + yt) + xyt + 7. \end{aligned}$$

Fig. 5(a) and Fig. 5(b) show respectively the numerical solution and the error at  $T = 1$  with our method using a  $20 \times 20 \times 10$  grid. Fig. 5(c) shows the geometry of the level set function and a semi-Cartesian grid with 20 grid points in both  $x$  and  $y$  directions. Tab. 2 gives the errors on different grids. Numerically, the method is second order accurate and the Newton's iteration steps is 3.

**Table 2:** Numerical results for Example 2

$n_x \times n_y \times n_t$	error in $u$	order	iteration steps
$16 \times 16 \times 5$	3.15e-3		3
$32 \times 32 \times 10$	8.21e-4	1.94	3
$64 \times 64 \times 20$	2.34e-4	1.81	3
$128 \times 128 \times 40$	5.84e-5	2.00	3

**Example 3.** The level-set function  $\phi$ , the coefficients  $\beta^\pm$ ,  $\sigma^\pm$ ,  $\sigma$  and the solution  $u^+$  are given as:

$$\phi(x, y) = \begin{cases} y - 2x, & \text{if } x + y > 0 \\ y - (2x + x^2), & \text{else} \end{cases},$$

$$\beta^+(x, y, t) = \begin{pmatrix} \sin(xyt) + 3 & xt \\ xt & xyt + 6 \end{pmatrix},$$

$$\beta^-(x, y, t) = \begin{pmatrix} x^2 + y^2 + t^2 + 1 & yt \\ yt & \cos(xy) + 3t + 6 \end{pmatrix},$$

$$\sigma^+(x, y, t) = \sin^2(xyt) + 1,$$

$$\sigma^-(x, y, t) = \sin(xt) \cos(yt) + 2,$$

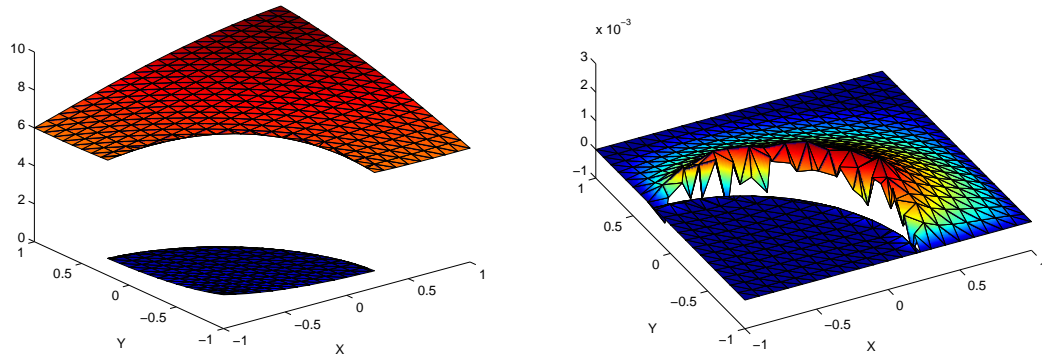
$$\sigma(x, y, t) = \cos^2(xyt) + 1,$$

$$u^+(x, y, t) = e^{x+yt}.$$

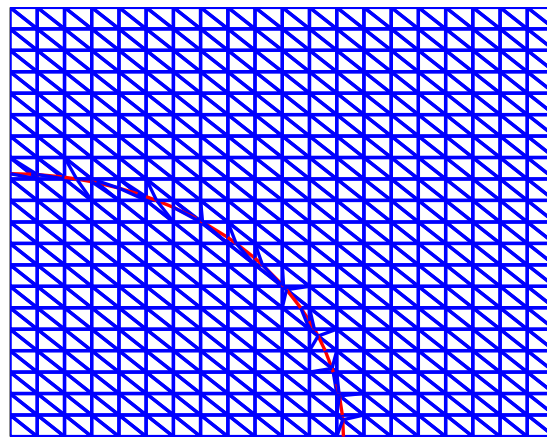
Fig. 6(a) and Fig. 6(b) show respectively the numerical solution and the error at  $T = 1$  with our method using a  $20 \times 20 \times 10$  grid. The geometry of the level set function and a semi-Cartesian grid with 20 grid points in both  $x$  and  $y$  directions are shown in Fig. 6(c). Tab. 3 gives the errors on different grids. Numerically, the method is second order accurate and the Newton's iteration steps remain less than 6.

**Table 3:** Numerical results for Example 3

$n_x \times n_y \times n_t$	error in $u$	order	iteration steps
$16 \times 16 \times 5$	1.92e-2		5
$32 \times 32 \times 10$	4.25e-3	2.18	4
$64 \times 64 \times 20$	1.10e-3	1.96	4
$128 \times 128 \times 40$	2.76e-4	1.99	4



(a) Numerical result on a  $20 \times 20 \times 10$  semi-Cartesian grid at  $T = 1$  (b) Error on a  $20 \times 20 \times 10$  semi-Cartesian grid at  $T = 1$

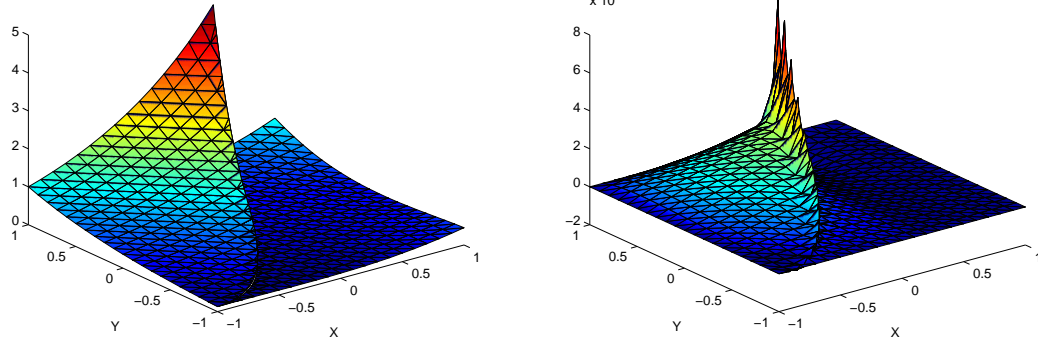


(c) Semi-Cartesian grid:  $20 \times 20$

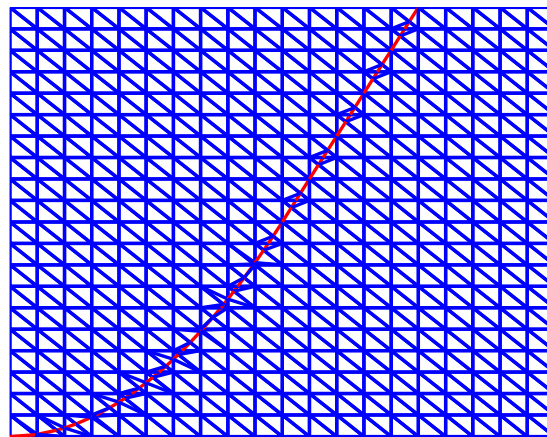
**Figure 5:** Numerical results for Example 2

## 5 Conclusion

We developed a simple, efficient numerical method for matrix-valued coefficient parabolic equations coupled with nonhomogeneous flux jump condition and nonlinear jump condition. We used TFEM coupled with Newton's method to deal with these two special and complicated jump conditions and used Crank-Nicolson's scheme for time derivative. The coefficients in the jump conditions are variable instead of constant in our previous work. We found that our previous work on NTFEM is not suitable treating such problem



(a) Numerical result on a  $20 \times 20 \times 10$  semi-Cartesian grid at  $T = 1$       (b) Error on a  $20 \times 20 \times 10$  semi-Cartesian grid at  $T = 1$



(c) Semi-Cartesian grid:  $20 \times 20$

**Figure 6:** Numerical results for Example 3

due to the linearization part, and this is not a trivial extension of our previous work on TFEM for elliptic interface problem due to the time derivative and the coefficients not being constant in the jump conditions. This is not a trivial extension of the previous work on the one dimensional nonlinear interface problem either, because the geometry in two dimensions is much more complicated than one dimension. As far as we know, a solver for this problem has not been reported in the literature. Extensive numerical experiments indicate that the method is about second-order accurate in the  $L^\infty$ -norm, and it is stable even for problems with very complicated interfaces.

**Acknowledgement:** L. Shi's research is supported by National Natural Science Foundation of China (No. 11701569). S. Hou's research is supported by Dr. Walter Koss Professorship made available through Louisiana Board of Regents. L. Wang's research is supported by Science Foundations of China University of Petroleum-Beijing (No. 2462015BJB05).

## References

- Bakker, E.; Meir, A. J.** (2003): How do pulsed amperometric ion sensors work? a simple pde model. *SIAM Review*, vol. 45, no. 2, pp. 327-344.
- Chen, Z.; Zou, J.** (1998): Finite element methods and their convergence for elliptic and parabolic interface problems. *Numerische Mathematik*, vol. 79, pp. 175-202.
- Ewing, R. E.; Li, Z.; Lin, T.; Lin, Y.** (1999): The immersed finite volume element methods for the elliptic interface problems. *Mathematics and Computers in Simulation*, vol. 50, no. 14, pp. 63-76.
- Gong, Y.; Li, B.; Li, Z.** (2008): Immersed-interface finite-element methods for elliptic interface problems with non-homogeneous jump conditions. *SIAM Journal on Numerical Analysis*, vol. 46, pp. 472-495.
- Guo, H.; Yang, X.** (2017): Gradient recovery for elliptic interface problem: II. immersed finite element methods. *Journal of Computational Physics*, vol. 338, pp. 606-619.
- Guo, H.; Yang, X.** (2018): Gradient recovery for elliptic interface problem: I. body-fitted mesh. *Communications in Computational Physics*, vol. 23, no. 5, pp. 1488-1511.
- He, X.; Lin, T.; Lin, Y.** (2011): Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. *International Journal of Numerical Analysis and Modeling*, vol. 8, no. 2, pp. 284-301.
- Hetzer, G.; Meir, A. J.** (2007): On an interface problem with a nonlinear jump condition, numerical approximation of solutions. *International Journal of Numerical Analysis and Modeling*, vol. 4, no. 3-4, pp. 519-530.
- Hou, S.; Li, Z.; Wang, L.; Wang, W.** (2012): A numerical method for solving elasticity equations with interfaces. *Communications in Computational Physics*, vol. 12, no. 2, pp. 595-612.
- Hou, S.; Song, P.; Wang, L.; Zhao, H.** (2013): A weak formulation for solving elliptic interface problems without body fitted grid. *Journal of Computational Physics*, vol. 249, pp. 80-95.
- Hou, S.; Wang, W.; Wang, L.** (2010): Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces. *Journal of Computational Physics*, vol. 229, pp. 7162-7179.
- Huang, J.; Zou, J.** (2002): Some new a priori estimates for second-order elliptic and parabolic interface problems. *Journal of Differential Equations*, vol. 184, pp. 570-586.
- LeVeque, R. J.; Li, Z.** (1994): The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, vol. 31, pp. 1019-1044.



- Li, Z.** (1998): The immersed interface method using a finite element formulation. *Applied Numerical Mathematics*, vol. 27, pp. 253-267.
- Li, Z.; Ito, K.** (2006): *The Immersed Interface Method: Numerical Solutions of Pdes Involving Interfaces and Irregular Domains*. Philadelphia: SIAM, USA.
- Li, Z.; Lin, T.; Wu, X.** (2003): New Cartesian grid methods for interface problem using finite element formulation. *Numerische Mathematik*, vol. 96, pp. 61-98.
- Lin, T.; Lin, Y.; Zhang, X.** (2015): Partially penalized immersed finite element methods for elliptic interface problems. *SIAM Journal on Numerical Analysis*, vol. 53, no. 2, pp. 1121-1144.
- Mu, L.; Wang, J.; Wei, G. W.; Ye, X.; Zhao, S.** (2013): Weak galerkin methods for second order elliptic interface problems. *Journal of Computational Physics*, vol. 250, pp. 106-125.
- Peskin, C. S.** (1977): Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, vol. 25, pp. 220-252.
- Radu, A.; Meir, A. J.; Bakker, E.** (2004): Dynamic diffusion model for tracing the real-time potential response of polymeric membrane ion-selective electrodes. *Analytical Chemistry*, vol. 76, no. 21, pp. 6402-6409.
- Wang, L.; Hou, S.; Shi, L.** (2017): A numerical method for solving two-dimensional elliptic interface problems with nonhomogeneous flux jump condition and nonlinear jump condition. *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 18, no. 3-4, pp. 245-254.
- Wang, L.; Shi, L.** (2014): A simple method for matrix-valued coefficient elliptic equations with sharp-edged interfaces. *Applied Mathematics and Computation*, vol. 242, pp. 917-930.
- Wang, L.; Shi, L.** (2015): A numerical method for solving matrix coefficient heat equations with interfaces. *Numerical Mathematics: Theory, Methods and Applications*, vol. 8, no. 4, pp. 475-495.
- Wang, Z.; Li, Z.; Lubkin, S.** (2014): A Robin-Robin domain decomposition method for a Stokes-Darcy structure interaction with a locally modified mesh. *Numerical Mathematics: Theory, Methods and Applications*, vol. 7, pp. 435-446.
- Xia, K.; Zhan, M.; Wei, G. W.** (2011): The matched interface and boundary (MIB) method for multi-domain elliptic interface problems. *Journal of Computational Physics*, vol. 230, pp. 8231-8258.
- Xie, H.; Li, Z.; Qiao, Z.** (2011): A finite element method for elasticity interface problems with locally modified triangulations. *International Journal of Numerical Analysis and Modeling*, vol. 8, no. 2, pp. 189-200.
- Ying, W. J.; Henriquez, C. S.** (2007): A kernel-free boundary integral method for elliptic boundary value problems. *Journal of Computational Physics*, vol. 227, no. 2, pp. 1046-1074.

**Yu, S.; Zhou, Y.; Wei, G. W.** (2007): Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *Journal of Computational Physics*, vol. 224, pp. 729-756.

**Zhou, Y.; Zhao, S.; Feig, M.; Wei, G. W.** (2006): High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, vol. 213, pp. 1-30.