Journal of Computational Mathematics, Vol.21, No.3, 2003, 257–276.

AN ITERATIVE HYBRIDIZED MIXED FINITE ELEMENT METHOD FOR ELLIPTIC INTERFACE PROBLEMS WITH STRONGLY DISCONTINUOUS COEFFICIENTS*[1)]

Dao-qi Yang
(*Department of Mathematics, Wayne State University, Detroit, MI 48202, USA*)

Jennifer Zhao
(*Department of Mathematics and Statistics, University of Michigan-Dearborn, Dearborn, MI 48128, USA*)

Abstract

An iterative algorithm is proposed and analyzed based on a hybridized mixed finite element method for numerically solving two-phase generalized Stefan interface problems with strongly discontinuous solutions, conormal derivatives, and coefficients. This algorithm iteratively solves small problems for each single phase with good accuracy and exchange information at the interface to advance the iteration until convergence, following the idea of Schwarz Alternating Methods. Error estimates are derived to show that this algorithm always converges provided that relaxation parameters are suitably chosen. Numeric experiments with matching and non-matching grids at the interface from different phases are performed to show the accuracy of the method for capturing discontinuities in the solutions and coefficients. In contrast to standard numerical methods, the accuracy of our method does not seem to deteriorate as the coefficient discontinuity increases.

Key words: Mixed finite element method, Interface problems, Discontinuous solutions.

1. Introduction

Interface problems occur in many physical applications. Below is a description of alloy solidification [10, 11, 12, 9] that shows the importance and characteristics of interface problems.

In alloy solidification problems, the melting temperature is not known in advance, which is different from classical Stefan problems such as ice-melting in water. The melting temperature depends on the composition of the alloy. Typically, an alloy is considered to comprise a pure substance containing a small concentration of one or more secondary substances, called impurities. The solidification of an alloy calls for a simultaneous study of the processes of heat flow and the diffusion of impurities. We now describe the mathematical model of a simple two-phase alloy solidification process in one, two or three space dimensions, with x denoting the space coordinate vector [10, 12, 11, 9]. Let $\Omega_1(t)$ denote the solid (alloy) region and $\Omega_2(t)$ the liquid (impurity) region, which are separated by the interface denoted by $\Gamma(t)$, where t represents time. Note the solid and liquid regions and the interface change with time t. Let u_1, c_1, K_1, and D_1 be the temperature, concentration of impurity, heat conductivity, and mass diffusion coefficient, respectively, in the solid region $\Omega_1(t)$, and u_2, c_2, K_2, and D_2 be the corresponding quantities in the liquid region $\Omega_2(t)$; see Figure 1.1. Then the partial differential equations modeling the process can be expressed as:

$$\rho_1 \frac{\partial u_1}{\partial t} = \nabla \cdot (K_1 \nabla u_1) + q_1, \quad \frac{\partial c_1}{\partial t} = \nabla \cdot (D_1 \nabla c_1), \quad \text{in } \Omega_1(t), \tag{1}$$

$$\rho_2 \frac{\partial u_2}{\partial t} = \nabla \cdot (K_2 \nabla u_2) + q_2, \quad \frac{\partial c_2}{\partial t} = \nabla \cdot (D_2 \nabla c_2), \quad \text{in } \Omega_2(t), \tag{2}$$

* Received September 5, 2000; final revised April 5, 2002.

$$u_1 = u_2, \quad K_2 \frac{\partial u_2}{\partial \nu} - K_1 \frac{\partial u_1}{\partial \nu} = -L v_\nu, \quad \text{on } \Gamma(t), \tag{3}$$

$$c_1 = c_S(u_1),\ c_2 = c_L(u_2), \quad D_2 \frac{\partial c_2}{\partial \nu} - D_1 \frac{\partial c_1}{\partial \nu} = (c_1 - c_2) v_\nu, \quad \text{on } \Gamma(t), \tag{4}$$

together with appropriate boundary and initial conditions, where v_ν denotes the speed at which the interface is moving along its normal direction ν, q_1, q_2 and ρ_1, ρ_2 are sources or sinks and specific heat in Ω_1, Ω_2, respectively, L is the latent heat. Equations (3) and (4) are the so-called generalized Stefan conditions.

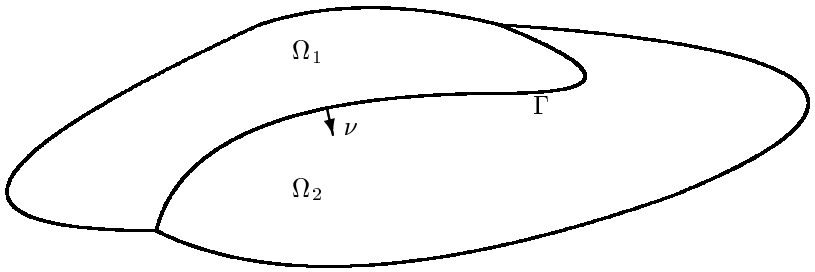

Figure 1.1: Two-phase alloy solidification with a moving interface $\Gamma(t)$ between the solid region $\Omega_1(t)$ and the liquid region $\Omega_2(t)$. Concentration of impurity is discontinuous across the interface.

Another example is multiphase immiscible flow of incompressible fluids with different densities and viscosities and surface tension. The governing equations in each fluid are the Navier-Stokes equations. The effect of surface tension is to balance the jump of the normal stress along the fluid interface, which gives rise to a free boundary condition for the discontinuity of the normal stress across the interface of the fluids. In the case of inviscid flows, the above jump condition is reduced to a discontinuity in pressure across the interface proportional to the curvature.

Interface problems such as the two mentioned above are difficult to solve by using conventional numerical methods since the the coefficients in different phases can be strongly discontinuous across the interface. Standard numerical methods such as finite element and mixed finite element algorithms are mainly designed to deal with problems with continuous or moderately discontinuous coefficients. For problems with strongly discontinuous coefficients, their accuracy can become arbitrarily inaccurate; see [31, 24] for some explanations and numerical examples. In particular, Nielsen [24] gave an example using standard finite element method whose accuracy deteriorates from 0.0044 to 0.0290 (or the error increased 559 percent) when the coefficient jump increases from 2 to 16. Vavasis [31] gave examples on which standard finite element methods fail on current computers. Note that modern preconditioners based on domain decomposition and multigrid cannot expect to improve the accuracy, although they may dramatically improve the efficiency of the solution process.

On the other hand, discontinuities in the solution or its normal derivatives can also present another difficulty; see [19, 20, 18, 32]. Note that the standard finite element theory [7] requires the solution be continuous (in $H^1(\Omega)$) and the mixed finite element theory [4] requires the normal derivative of the solution be continuous. Here we face a class of problems whose solution and its normal derivative can be discontinuous in the physical domain. Thus standard finite

element and mixed finite element theories do not apply directly to these interface problems.

We herein present an iterative hybridized mixed finite element method for linear second order elliptic interface problems in general domains, to which time-dependent interface problems can be reduced after a finite-differencing in time. The idea is to employ the Schwarz alternating method for the mixed formulation of the problem by introducing Lagrange multipliers on the interface so that the original problem can be approximated iteratively through solving each individual phase problem.

The organization of this paper is as follows. In §, the algorithm is described for general elliptic interface problems. In §, a convergence analysis of the method is carried out. Finally in §, numerical examples are provided to check the performance of the method, and in §, some concluding remarks are given.

2. An Iterative Hybridized Mixed Finite Element Method

Let Ω be a smooth bounded domain or a convex polygon or polyhedron in $\mathcal{R}^2$ or $\mathcal{R}^3$ with boundary $\partial\Omega$. Suppose that one phase of a steady state two-phase generalized Stefan problem occupies Ω_1 and the other phase occupies Ω_2, where Ω_1 and Ω_2 are two nonoverlapping subregions of Ω such that $\bar{\Omega}_1 \cup \bar{\Omega}_2 = \bar{\Omega}$, and $\Omega_1 \cap \Omega_2 = 0$. Let $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ denote the interface between the two phases or subregions, and denote $\Gamma_i = \partial\Omega_i \cap \partial\Omega$, $i = 1, 2$. Consider the linear boundary value interface problem of finding $u = \{u_1, u_2\} : \Omega_1 \times \Omega_2 \to \mathcal{R}$ such that

$$-\text{div}\,(a_i \nabla u_i) + c_i u_i = f_i \text{ in } \Omega_i,\ i = 1, 2, \tag{5}$$

$$u_i = g_i \text{ on } \Gamma_i,\ i = 1, 2, \tag{6}$$

$$u_1 - u_2 = \mu \text{ on } \Gamma, \tag{7}$$

$$-(a_1 \nabla u_1) \cdot \nu_1 - (a_2 \nabla u_2) \cdot \nu_2 = \eta \text{ on } \Gamma, \tag{8}$$

where a_i and c_i are the diffusion tensor and reaction coefficient, defined on Ω_i, $i = 1, 2$, respectively, μ and η are given regular functions on the interface Γ, and ν_1 and ν_2 are the unit outward normals at the interface of Ω_1 and Ω_2, respectively. ∇ and div denote the gradient and divergence operators respectively.

The problem (5)-(8) is a slight generalization from normal interface problems in which $\mu = \eta = 0$; see [6, 14, 40]. Our main concern in this paper is how to solve it accurately when the coefficient jumps are large. In many applications, the coefficient jump functions on the interface:

$$a_1(x)\,|_{x\in\Gamma} - a_2(x)\,|_{x\in\Gamma} \quad \text{and} \quad c_1(x)\,|_{x\in\Gamma} - c_2(x)\,|_{x\in\Gamma}$$

are very large. For example, in fluid flow in heterogeneous and anisotropic porous media, the permeability jump can be 10^{11} or larger (permeability in shale is in the order of 10^{-6} to 10^{-11} and in sandstone is in the order of 1 to 10^{-4}). In computational electromagnetics [5, 29], large discontinuity in material properties can occur; the jump in magnetic permeability can be in the range of 10^5 and jump in electrical conductivity can be in the range of 10^{15} or more (for example, the conductivity of pure copper is 6.4×10^5 while conductivity of pure water is only 4.0×10^{-8} at temperature of 291K). In fluid dynamics, the viscosities of fluids can be in the order of 10^5 or more (viscosities in $Pa \cdot s$ of glycerine and hydrogen are 1.49 and 8.7×10^{-6} respectively at temperature of $20C$). In dealing with micro (at atom or molecule levels) and celestial objects, discontinuities in larger scales can also occur. With such large coefficient jumps, standard numerical methods can be arbitrarily inaccurate; see [31, 24, 13] for numerical examples. A clear deterioration in accuracy can be observed for standard finite element and volume methods even for very small discontinuities in the coefficients [24, 13]. Vavasis and Nielsen [31, 24] are able to improve the accuracy of standard numerical methods for special interface problems under the assumption that, for example, the source term is zero in the region where the coefficient is small.

Recently, in [19, 18], the immersed interface method based on finite differences was considered for the problem (5)-(8) with a scalar diffusion coefficient. Applications to time-dependent interface problems were presented in [20] and [32]. In this approach, interface conditions are combined into finite difference equations in each subregion to form a global system of algebraic equations. The resulting system is can be singular in general. The design of efficient algorithms for solving such a system is not easy; see Li [18] for a special case. Due to the ill-conditioning of the global system, such methods are not expected to improve the accuracy of the numerical solutions for strongly discontinuous coefficients.

Denote $q_i = -a_i \nabla u_i$, $i = 1, 2$. Throughout this paper, we will call u the pressure variable and q the flux variable. Then, under reasonable assumptions, the interface problem (5)-(8) is equivalent to the mixed formulation given by

$$a_i^{-1} q_i + \nabla u_i = 0, \text{ in } \Omega_i, \ i = 1, 2, \tag{9}$$

$$\text{div } q_i + c_i u_i = f_i \text{ in } \Omega_i, \ i = 1, 2, \tag{10}$$

$$u_i = g_i \text{ on } \Gamma_i, \ i = 1, 2, \tag{11}$$

$$u_1 - u_2 = \mu \text{ on } \Gamma, \tag{12}$$

$$q_1 \cdot \nu_1 + q_2 \cdot \nu_2 = \eta \text{ on } \Gamma. \tag{13}$$

Define $V_i = H(\text{div}, \Omega_i) = \{v : v \in L^2(\Omega_i) \text{ and div } v \in L^2(\Omega_i)\}$ and $W_i = L^2(\Omega_i)$, $i = 1, 2$. Denote $(\phi, \psi)_\Omega = \int_\Omega \phi \cdot \psi dx$ and $\langle \phi, \psi \rangle_\Gamma = \int_\Gamma \phi \cdot \psi ds$. When the integrals do not exist, corresponding duality should be used to interpret them. Then the weak form of (9)-(13) is given by seeking $\{q_i, u_i\} \in V_i \times W_i$ such that

$$(a_i^{-1} q_i, v)_{\Omega_i} - (u_i, \text{div } v)_{\Omega_i} + \langle u_i, v \cdot \nu_i \rangle_\Gamma = -\langle g_i, v \cdot \nu_i \rangle_{\Gamma_i}, \quad \forall v \in V_i, \tag{14}$$

$$(\text{div } q_i, w)_{\Omega_i} + (c_i u_i, w)_{\Omega_i} = (f_i, w)_{\Omega_i}, \quad \forall w \in W_i, \tag{15}$$

$$u_1 - u_2 = \mu, \text{ on } \Gamma, \tag{16}$$

$$q_1 \cdot \nu_1 + q_2 \cdot \nu_2 = \eta, \text{ on } \Gamma. \tag{17}$$

Note the global system (14)-(17) can be solved by eliminating the pressure and flux unknowns from one side on the interface. However, this global system is neither symmetric nor positive definite (when $\mu \neq 0$ and $\eta \neq 0$), and in general is hard to solve. A popular method is the GMRES method, that requires large memory storage and a vector orthogonalization process. It can be very slow in execution time and exceed one's computer memory for large size computations. Even if the global system can be solved efficiently with a good preconditioner, its accuracy will deteriorate very rapidly when the coefficient jumps are getting large.

Now we sketch the idea of our algorithm that iteratively solves problems defined for each phase and exchange pressure and flux data at the interface to advance the iterative process. The single-phase problems are smaller and thus are much easier to solve than the global system (14)-(17). The approximations to flux and pressure variables will be of the same order as in standard mixed finite element methods. The algorithm could be written as follows. Select an initial guess $\{q_i^0, u_i^0\} \in V_i \times W_i$, $i = 1, 2$, and compute $\{q_i^{n+1}, u_i^{n+1}\} \in V_i \times W_i$ such that

$$\begin{cases} (a_1^{-1} q_1^{n+1/2}, v)_1 - (u_1^{n+1/2}, \text{div } v)_1 + \langle u_1^{n+1/2}, v \cdot \nu_1 \rangle_\Gamma = -\langle g_1, v \cdot \nu_1 \rangle_{\Gamma_1}, \quad \forall v \in V_1, \\ (\text{div } q_1^{n+1/2}, w)_1 + (c_1 u_1^{n+1/2}, w)_1 = (f_1, w)_1, \quad \forall w \in W_1, \\ q_1^{n+1/2} \cdot \nu_1 = \alpha q_1^n \cdot \nu_1 - (1 - \alpha) q_2^n \cdot \nu_2 + (1 - \alpha)\eta \quad \text{on} \quad \Gamma; \end{cases} \tag{18}$$

$$\begin{cases} (a_2^{-1} q_2^{n+1/2}, v)_2 - (u_2^{n+1/2}, \text{div } v)_2 + \langle u_2^{n+1/2}, v \cdot \nu_2 \rangle_\Gamma = -\langle g_2, v \cdot \nu_2 \rangle_{\Gamma_2}, \quad \forall v \in V_2, \\ (\text{div } q_2^{n+1/2}, w)_2 + (c_2 u_2^{n+1/2}, w)_2 = (f_2, w)_2, \quad \forall w \in W_2, \\ q_2^{n+1/2} \cdot \nu_2 = -\alpha q_1^n \cdot \nu_1 + (1 - \alpha) q_2^n \cdot \nu_2 + \alpha\eta \quad \text{on} \quad \Gamma; \end{cases} \tag{19}$$

and

$$\begin{cases} (a_1^{-1} q_1^{n+1}, v)_1 - (u_1^{n+1}, \text{div } v)_1 = \\ \quad -\langle \beta u_1^{n+1/2} + (1 - \beta) u_2^{n+1/2} + (1 - \beta)\mu, v \cdot \nu_1 \rangle_\Gamma - \langle g_1, v \cdot \nu_1 \rangle_{\Gamma_1}, \ \forall v \in V_1, \\ (\text{div } q_1^{n+1}, w)_1 + (c_1 u_1^{n+1}, w)_1 = (f_1, w)_1, \quad \forall w \in W_1; \end{cases} \tag{20}$$

$$\begin{cases} (a_2^{-1}q_2^{n+1}, v)_2 - (u_2^{n+1}, \operatorname{div} v)_2 = \\ \qquad -\langle \beta u_1^{n+1/2} + (1-\beta)u_2^{n+1/2} - \beta\mu, v\cdot\nu_2\rangle_\Gamma - \langle g_2, v\cdot\nu_2\rangle_{\Gamma_2}, \ \forall v \in V_2, \\ (\operatorname{div} q_2^{n+1}, w)_2 + (c_2 u_2^{n+1}, w)_2 = (f_2, w)_2, \quad \forall w \in W_2; \end{cases} \tag{21}$$

where $\alpha, \beta \in [0,1]$ are relaxation parameters that will be determined to ensure and to accelerate the convergence of the iterative procedure. Note when this algorithm converges, the iterates will converge to the solution to the system (14)-(17).

However, there is a technical difficulty with this formulation of the algorithm, since the restriction of an L^2 function on the interface Γ is not clear. For this reason, we introduce Lagrange multiplier λ_i [1, 16, 17] in place of u_i on Γ to obtain:

$$\begin{cases} (a_1^{-1}q_1^{n+1/2}, v)_1 - (u_1^{n+1/2}, \operatorname{div} v)_1 + \langle \lambda_1^{n+1/2}, v\cdot\nu_1\rangle_\Gamma = -\langle g_1, v\cdot\nu_1\rangle_{\Gamma_1}, \quad \forall v \in V_1, \\ (\operatorname{div} q_1^{n+1/2}, w)_1 + (c_1 u_1^{n+1/2}, w)_1 = (f_1, w)_1, \quad \forall w \in W_1, \\ q_1^{n+1/2}\cdot\nu_1 = \alpha q_1^n\cdot\nu_1 - (1-\alpha)q_2^n\cdot\nu_2 + (1-\alpha)\eta \quad \text{on} \quad \Gamma; \end{cases} \tag{22}$$

$$\begin{cases} (a_2^{-1}q_2^{n+1/2}, v)_2 - (u_2^{n+1/2}, \operatorname{div} v)_2 + \langle \lambda_2^{n+1/2}, v\cdot\nu_2\rangle_\Gamma = -\langle g_2, v\cdot\nu_2\rangle_{\Gamma_2}, \quad \forall v \in V_2, \\ (\operatorname{div} q_2^{n+1/2}, w)_2 + (c_2 u_2^{n+1/2}, w)_2 = (f_2, w)_2, \quad \forall w \in W_2, \\ q_2^{n+1/2}\cdot\nu_2 = -\alpha q_1^n\cdot\nu_1 + (1-\alpha)q_2^n\cdot\nu_2 + \alpha\eta \quad \text{on} \quad \Gamma; \end{cases} \tag{23}$$

and

$$\begin{cases} (a_1^{-1}q_1^{n+1}, v)_1 - (u_1^{n+1}, \operatorname{div} v)_1 = \\ \qquad -\langle \beta\lambda_1^{n+1/2} + (1-\beta)\lambda_2^{n+1/2} + (1-\beta)\mu, v\cdot\nu_1\rangle_\Gamma - \langle g_1, v\cdot\nu_1\rangle_{\Gamma_1}, \ \forall v \in V_1, \\ (\operatorname{div} q_1^{n+1}, w)_1 + (c_1 u_1^{n+1}, w)_1 = (f_1, w)_1, \quad \forall w \in W_1; \end{cases} \tag{24}$$

$$\begin{cases} (a_2^{-1}q_2^{n+1}, v)_2 - (u_2^{n+1}, \operatorname{div} v)_2 = \\ \qquad -\langle \beta\lambda_1^{n+1/2} + (1-\beta)\lambda_2^{n+1/2} - \beta\mu, v\cdot\nu_2\rangle_\Gamma - \langle g_2, v\cdot\nu_2\rangle_{\Gamma_2}, \ \forall v \in V_2, \\ (\operatorname{div} q_2^{n+1}, w)_2 + (c_2 u_2^{n+1}, w)_2 = (f_2, w)_2, \quad \forall w \in W_2; \end{cases} \tag{25}$$

This algorithm can be discretized in a standard way. Let $\mathcal{T}_h$ be a union of triangulations in subregions Ω_1 and Ω_2 that matches at the interface Γ, and let $V_i^h \times W_i^h \subset V_i \times W_i$ be a mixed finite element space [4, 2, 3, 25], defined on Ω_i, $i = 1, 2$. In each of the mixed finite element spaces cited, the functions in W_i^h are allowed to be discontinuous across element boundaries. We now need to introduce a space for the Lagrange multipliers [1, 16, 17]. For $q_i^h \in V_i^h$, assume that its normal component $q_i^h \cdot \nu_i \in \mathcal{P}_\tau$ on each edge in Γ, where $\mathcal{P}_\tau$ denotes the set of polynomials of degree τ in the arc length on the edge; for simplicity we assume τ to be the same on all edges of all elements. Let Y^h consist of two copies of $\mathcal{P}_\tau$ over each interior edge in Γ, with one copy associated with the edge as a part of $\partial\Omega_1$ and the other as a part of $\partial\Omega_2$. Denote the collection of copies of $\mathcal{P}_\tau$ associated with Ω_i by Y_i^h. Then, the selectively hybridized mixed finite element method is given by seeking $\{q_i^h, u_i^h, \lambda_i^h\} \in V_i^h \times W_i^h \times Y_i^h$ such that

$$(a_i^{-1}q_i^h, v)_{\Omega_i} - (u_i^h, \operatorname{div} v)_{\Omega_i} + \langle \lambda_i^h, v\cdot\nu_i\rangle_\Gamma = -\langle g_i, v\cdot\nu_i\rangle_{\Gamma_i}, \qquad \forall v \in V_i^h, \tag{26}$$

$$(\operatorname{div} q_i^h, w)_{\Omega_i} + (c_i u_i^h, w)_{\Omega_i} = (f, w)_{\Omega_i}, \qquad \forall w \in W_i^h, \tag{27}$$

$$\lambda_1^h - \lambda_2^h = \mu, \qquad \text{on} \quad \Gamma, \tag{28}$$

$$q_1^h\cdot\nu_1 + q_2^h\cdot\nu_2 = \eta, \qquad \text{on} \quad \Gamma. \tag{29}$$

If Γ includes every interior edge of all elements, then the method above is said to be totally hybridized; however, it should be emphasized that the partition $\{\Omega_1, \Omega_2\}$ is fixed and does not depend on the mesh parameter h.

The discrete version of our iterative algorithm can be stated as follows. Let $\{Q_i^n, U_i^n, \Lambda_i^n\}$ denote the finite-dimensional approximations of $\{q_i^n, u_i^n, \lambda_i^n\}$. Choose an initial guess

$$\{Q_i^0, U_i^0, \Lambda_i^0\} \in V_i^h \times W_i^h \times Y_i^h$$

arbitrarily. For $n = 0, 1, 2, \ldots$, construct pairs $\{Q_i^{n+1}, U_i^{n+1}\} \in V_i^h \times W_i^h$ satisfying:

$$\begin{cases} (a_1^{-1}Q_1^{n+1/2}, v)_1 - (U_1^{n+1/2}, \operatorname{div} v)_1 + \langle \Lambda_1^{n+1/2}, v\cdot\nu_1\rangle_\Gamma = -\langle g_1, v\cdot\nu_1\rangle_{\Gamma_1}, \quad \forall v \in V_1^h, \\ (\operatorname{div} Q_1^{n+1/2}, w)_1 + (c_1 U_1^{n+1/2}, w)_1 = (f_1, w)_1, \quad \forall w \in W_1^h, \\ Q_1^{n+1/2}\cdot\nu_1 = \alpha Q_1^n\cdot\nu_1 - (1-\alpha)Q_2^n\cdot\nu_2 + (1-\alpha)\eta \quad \text{on} \quad \Gamma; \end{cases} \tag{30}$$

$$\begin{cases} (a_2^{-1}Q_2^{n+1/2}, v)_2 - (U_2^{n+1/2}, \operatorname{div} v)_2 + \langle \Lambda_2^{n+1/2}, v\cdot\nu_2\rangle_\Gamma = -\langle g_2, v\cdot\nu_2\rangle_{\Gamma_2}, \quad \forall v \in V_2^h, \\ (\operatorname{div} Q_2^{n+1/2}, w)_2 + (c_2 U_2^{n+1/2}, w)_2 = (f_2, w)_2, \quad \forall w \in W_2^h, \\ Q_2^{n+1/2}\cdot\nu_2 = -\alpha Q_1^n\cdot\nu_1 + (1-\alpha)Q_2^n\cdot\nu_2 + \alpha\eta \quad \text{on} \quad \Gamma, \end{cases} \tag{31}$$

and

$$\begin{cases} (a_1^{-1}Q_1^{n+1}, v)_1 - (U_1^{n+1}, \operatorname{div} v)_1 = \\ \qquad -\langle \beta\Lambda_1^{n+1/2} + (1-\beta)\Lambda_2^{n+1/2} + (1-\beta)\mu, v\cdot\nu_1\rangle_\Gamma - \langle g_1, v\cdot\nu_1\rangle_{\Gamma_1}, \ \forall v \in V_1^h, \\ (\operatorname{div} Q_1^{n+1}, w)_1 + (c_1 U_1^{n+1}, w)_1 = (f_1, w)_1, \quad \forall w \in W_1^h; \end{cases} \tag{32}$$

$$\begin{cases} (a_2^{-1}Q_2^{n+1}, v)_2 - (U_2^{n+1}, \operatorname{div} v)_2 = \\ \qquad -\langle \beta\Lambda_1^{n+1/2} + (1-\beta)\Lambda_2^{n+1/2} - \beta\mu, v\cdot\nu_2\rangle_\Gamma - \langle g_2, v\cdot\nu_2\rangle_{\Gamma_2}, \ \forall v \in V_2^h, \\ (\operatorname{div} Q_2^{n+1}, w)_2 + (c_2 U_2^{n+1}, w)_2 = (f_2, w)_2, \quad \forall w \in W_2^h. \end{cases} \tag{33}$$

This iterative hybridized mixed finite element algorithm reduces the original indefinite problem to iteratively solving well-conditioned small problems defined for individual phases. Since quantities corresponding to each phase, such as heat conductivity, density, and concentration, are usually continuous functions, our subproblems have continuous solutions and coefficients and thus can easily be solved with good accuracy. The idea of the iterative algorithm is motivated by the Schwarz Alternating Method [8, 28, 30, 15, 26, 35, 38, 39, 37, 40].

3. Error Analysis

In this section, we analyze the convergence of the iterative scheme (30)-(33). Let

$$r_i^k = q_i^h - Q_i^k, \quad e_i^k = u_i^h - U_i^k, \quad \delta_i^k = \lambda_i^h - \Lambda_i^k, \qquad i = 1, 2.$$

Then, by combining (26)-(29) and (30)-(33), we have the following error equations:

$$(a_i^{-1}r_i^{n+1/2}, v)_{\Omega_i} - (e_i^{n+1/2}, \operatorname{div} v)_{\Omega_i} + \langle \delta_i^{n+1/2}, v\cdot\nu_i\rangle_\Gamma = 0, \ \forall v \in V_i^h, \tag{34}$$

$$(\operatorname{div} r_i^{n+1/2}, w)_{\Omega_i} + (c_i e_i^{n+1/2}, w)_{\Omega_i} = 0, \qquad \forall w \in W_i^h, \tag{35}$$

$$r_i^{n+1/2}\cdot\nu_i = \alpha r_1^n\cdot\nu_i + (1-\alpha)r_2^n\cdot\nu_i, \qquad \text{on} \quad \Gamma; \tag{36}$$

and

$$(a_i^{-1}r_i^{n+1}, v)_{\Omega_i} - (e_i^{n+1}, \operatorname{div} v)_{\Omega_i} = -\langle \beta\delta_1^{n+1/2} + (1-\beta)\delta_2^{n+1/2}, v\cdot\nu_i\rangle_\Gamma, \quad \forall v \in V_i^h, \tag{37}$$

$$(\operatorname{div} r_i^{n+1}, w)_{\Omega_i} + (c_i e_i^{n+1}, w)_{\Omega_i} = 0, \qquad \forall w \in W_i^h. \tag{38}$$

Let $\Phi = \{\psi\cdot\nu_i|_\Gamma : \psi \in V_i^h, i = 1, 2\}$, and define the extension operators

$$R_i : \phi \in \Phi \to \{R_i^1\phi, R_i^2\phi, R_i^3\phi\} \in V_i^h \times W_i^h \times Y_i^h$$

by

$$(a_i^{-1}R_i^1\phi, v)_{\Omega_i} - (R_i^2\phi, \operatorname{div} v)_{\Omega_i} + \langle R_i^3\phi, v\cdot\nu_i\rangle_\Gamma = 0, \quad \forall v \in V_i^h, \tag{39}$$

$$(\operatorname{div} R_i^1\phi, w)_{\Omega_i} + (c_i R_i^2\phi, w)_{\Omega_i} = 0, \qquad \forall w \in W_i^h, \tag{40}$$

$$R_i^1\phi\cdot\nu_i = \phi, \qquad \text{on} \quad \Gamma. \tag{41}$$

Then, define the linear operators $T_{km} : \Phi \to \Phi, \ k, m = 1, 2$, by

$$T_{km}\phi = \xi\cdot\nu_m \in \Phi,$$

where $\{\xi, \zeta\} \in V_m^h \times W_m^h$ satisfies

$$(a_m^{-1}\xi, v)_{\Omega_m} - (\zeta, \operatorname{div} v)_{\Omega_m} = -\langle R_k^3\phi, v\cdot\nu_m\rangle_\Gamma, \quad \forall v \in V_m^h, \tag{42}$$

$$(\operatorname{div}\xi, w)_{\Omega_m} + (c_m\zeta, w)_{\Omega_m} = 0, \qquad \forall w \in W_m^h. \tag{43}$$

Let γ_0 denote the restriction operator on Γ. From (34)-(36), we have

$$\gamma_0 r_1^{n+1/2}\cdot\nu_1 = \alpha\gamma_0 r_1^n\cdot\nu_1 - (1-\alpha)\gamma_0 r_2^n\cdot\nu_2, \tag{44}$$

$$\gamma_0 r_2^{n+1/2}\cdot\nu_2 = -\alpha\gamma_0 r_1^n\cdot\nu_1 + (1-\alpha)\gamma_0 r_2^n\cdot\nu_2. \tag{45}$$

From (37)-(43), we have

$$\gamma_0 r_1^{n+1}\cdot\nu_1 = \beta\gamma_0 r_1^{n+1/2}\cdot\nu_1 + (1-\beta)T_{21}(\gamma_0 r_2^{n+1/2}\cdot\nu_2), \tag{46}$$

$$\gamma_0 r_2^{n+1}\cdot\nu_2 = \beta T_{12}(\gamma_0 r_1^{n+1/2}\cdot\nu_1) + (1-\beta)\gamma_0 r_2^{n+1/2}\cdot\nu_2. \tag{47}$$

For convenience, we define the following norms on the interface Γ :

$$|\phi|_i^2 = (a_i^{-1}R_i^1\phi, R_i^1\phi)_{\Omega_i} + (c_iR_i^2\phi, R_i^2\phi)_{\Omega_i}, \quad \forall\ \phi \in \Phi,\ \ i = 1, 2, \tag{48}$$

and introduce the two quantities

$$\sigma = \sup_{\phi\in\Phi} \frac{|\phi|_1^2}{|\phi|_2^2}, \qquad \tau = \sup_{\phi\in\Phi} \frac{|\phi|_2^2}{|\phi|_1^2}. \tag{49}$$

Note that $|\cdot|_i$ are equivalent discrete norms of the space $H_{00}^{1/2}(\Gamma)$. Thus σ and τ are positive and finite numbers.

We are now in a position to show the convergence of our algorithm.

Theorem 3.1.

$$|\gamma_0 r_1^{n+1}\cdot\nu_1|_1^2 \le \left[\beta^2 - \frac{2\beta(1-\beta)}{\sigma} + (1-\beta)^2\tau^2\right] |\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2. \tag{50}$$

Proof. By (46) and (48), we have

$$\begin{aligned} |\gamma_0 r_1^{n+1}\cdot\nu_1|_1^2 = \quad & \beta^2|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2 + (1-\beta)^2|T_{21}\gamma_0 r_2^{n+1/2}\cdot\nu_2|_1^2 \\ & +2\beta(1-\beta)\big(a_1^{-1}R_1^1\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_1^1T_{21}\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_1} \\ & +2\beta(1-\beta)\big(c_1R_1^2\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_1^2T_{21}\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_1}. \end{aligned} \tag{51}$$

In order to estimate (51) term by term, we need some preliminary bounds. From (39)-(41) and (42)-(43), we have, $\forall\phi \in \Phi$,

$$\begin{aligned} & (a_1^{-1}R_1^1T_{21}\phi, v)_{\Omega_1} - (R_1^2T_{21}\phi, \operatorname{div} v)_{\Omega_1} \\ & \quad = (a_2^{-1}R_2^1\phi, R_2^1(\gamma_0 v\cdot\nu_1))_{\Omega_2} - (R_2^2\phi, \operatorname{div} R_2^1(\gamma_0 v\cdot\nu_1))_{\Omega_2}, \quad \forall v \in V_1^h, \end{aligned} \tag{52}$$

$$(\operatorname{div} R_1^1T_{21}\phi, w)_{\Omega_1} + (c_1R_1^2T_{21}\phi, w)_{\Omega_1} = 0, \quad \forall w \in W_1^h, \tag{53}$$

$$(\operatorname{div} R_2^1\phi, R_2^2(\gamma_0 v\cdot\nu_1))_{\Omega_2} + (c_2R_2^2\phi, R_2^2(\gamma_0 v\cdot\nu_1))_{\Omega_2} = 0, \quad \forall v \in V_1^h. \tag{54}$$

Letting $v = R_1^1T_{21}\phi$ in (52) and (54) and $w = R_1^2T_{21}\phi$ in (53) and adding, we obtain

$$\begin{aligned} & (a_1^{-1}R_1^1T_{21}\phi, R_1^1T_{21}\phi)_{\Omega_1} + (c_1R_1^2T_{21}\phi, R_1^2T_{21}\phi)_{\Omega_1} \\ & \quad = (a_2^{-1}R_2^1\phi, R_2^1T_{21}\phi)_{\Omega_2} + (c_2R_2^2\phi, R_2^2T_{21}\phi)_{\Omega_2},\ \forall\phi \in \Phi, \end{aligned} \tag{55}$$

where we used the fact that

$$(R_2^2\phi, \operatorname{div} R_2^1T_{21}\phi)_{\Omega_2} = (\operatorname{div} R_2^1\phi, R_2^2T_{21}\phi)_{\Omega_2}. \tag{56}$$

Indeed, the equality (56) can be proved by combining (40) and (43). Combining (55) and (49) yields

$$|T_{21}\phi|_1 \le \sqrt{\tau}|\phi|_2, \qquad \forall\phi \in \Phi. \tag{57}$$

From (57), (49), (44), and (45), we obtain immediately

$$|T_{21}\gamma_0 r_2^{n+1/2}\cdot\nu_2|_1 \le \sqrt{\tau}|\gamma_0 r_2^{n+1/2}\cdot\nu_2|_2 \le \sqrt{\tau}|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2 \le \tau|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1. \tag{58}$$

Now, from (52), we have, $\forall\phi, \psi \in \Phi$,

$$(a_1^{-1}R_1^1T_{21}\psi, R_1^1\phi)_{\Omega_1} - (R_1^2T_{21}\psi, \operatorname{div} R_1^1\phi)_{\Omega_1} = (a_2^{-1}R_2^1\psi, R_2^1\phi)_{\Omega_2} - (R_2^2\psi, \operatorname{div} R_2^1\phi)_{\Omega_2}. \tag{59}$$

Since $(R_1^2\phi, \operatorname{div} R_1^1T_{21}\psi)_{\Omega_1} = (R_1^2T_{21}\psi, \operatorname{div} R_1^1\phi)_{\Omega_1}$, then (59) becomes

$$(a_1^{-1}R_1^1\phi, R_1^1T_{21}\psi)_{\Omega_1} - (R_1^2\phi, \operatorname{div} R_1^1T_{21}\psi)_{\Omega_1} = (a_2^{-1}R_2^1\psi, R_2^1\phi)_{\Omega_2} - (R_2^2\psi, \operatorname{div} R_2^1\phi)_{\Omega_2}. \tag{60}$$

In view of (53) and (40), we have

$$(\operatorname{div} R_1^1T_{21}\psi, R_1^2\phi)_{\Omega_1} + (c_1R_1^2T_{21}\psi, R_1^2\phi)_{\Omega_1} = 0, \tag{61}$$

$$(\operatorname{div} R_2^1\phi, R_2^2\psi)_{\Omega_2} + (c_2R_2^2\phi, R_2^2\psi)_{\Omega_2} = 0. \tag{62}$$

Combining (60), (61) and (62) yields

$$\begin{aligned} & (a_1^{-1}R_1^1\phi, R_1^1T_{21}\psi)_{\Omega_1} + (c_1R_1^2\phi, R_1^2T_{21}\psi)_{\Omega_1} \\ & \quad = (a_2^{-1}R_2^1\phi, R_2^1\psi)_{\Omega_2} + (c_2R_2^2\phi, R_2^2\psi)_{\Omega_2}, \quad \forall\phi,\ \psi \in \Phi. \end{aligned} \tag{63}$$

Thus, by (63)

$$\begin{aligned}&\big(a_1^{-1}R_1^1\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_1^1T_{21}\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_1}+\big(c_1R_1^2\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_1^2T_{21}\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_1}\\&=\big(a_2^{-1}R_2^1\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^1\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2}+\big(c_2R_1^2\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_1^2\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2}\\&=-|\gamma_0 r_2^{n+1/2}\cdot\nu_2|_2^2\le-\frac{1}{\sigma}|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2.\end{aligned}\tag{64}$$

Substituting (58) and (64) into (51) gives (50).

Theorem 3.2.

$$|\gamma_0 r_2^{n+1}\cdot\nu_2|_2^2\le\left[\beta^2-\frac{2\beta(1-\beta)}{\sigma}+(1-\beta)^2\tau^2\right]|T_{12}\gamma_0 r_2^{n+1/2}\cdot\nu_2|_2^2.\tag{65}$$

Proof. From (47) and (48), we have

$$\begin{aligned}|\gamma_0 r_2^{n+1}\cdot\nu_2|_2^2=\;&\beta^2\big|T_{12}(\gamma_0 r_1^{n+1/2}\cdot\nu_1)\big|_2^2+(1-\beta)^2\big|\gamma_0 r_2^{n+1/2}\cdot\nu_2\big|_2^2\\&+2\beta(1-\beta)\big(a_2^{-1}R_2^1T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^1\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2}\\&+2\beta(1-\beta)\big(c_2R_2^2T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^2\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2}.\end{aligned}\tag{66}$$

Instead of bounding $|\gamma_0 r_2^{n+1}\cdot\nu_2|_2^2$ in terms of $\big|\gamma_0 r_2^{n+1/2}\cdot\nu_2\big|_2^2$ in the same fashion as for (51), we estimate (66) differently. The idea is to bound all terms on the right side of (66) in terms of $\big|T_{12}\gamma_0 r_2^{n+1/2}\cdot\nu_2\big|_2^2$. From (39)-(41) and (42)-(43) we have, $\forall\phi\in\Phi$,

$$\begin{aligned}&(a_2^{-1}R_2^1T_{12}\phi, v)_{\Omega_2}-(R_2^2T_{12}\phi,\operatorname{div} v)_{\Omega_2}\\&\quad=(a_1^{-1}R_1^1\phi, R_1^1(\gamma_0 v\cdot\nu_2))_{\Omega_1}-(R_1^2\phi,\operatorname{div} R_1^1(\gamma_0 v\cdot\nu_2))_{\Omega_1},\quad\forall v\in V_2^h,\end{aligned}\tag{67}$$

$$(\operatorname{div} R_2^1T_{12}\phi, w)_{\Omega_2}+(c_2R_2^2T_{12}\phi, w)_{\Omega_2}=0\quad\forall w\in W_2^h,\tag{68}$$

$$(\operatorname{div} R_1^1\phi, R_1^2(\gamma_0 v\cdot\nu_2))_{\Omega_1}+(c_1R_1^2\phi, R_1^2(\gamma_0 v\cdot\nu_2))_{\Omega_1}=0,\quad\forall v\in V_2^h.\tag{69}$$

Let $v=R_2^1T_{12}\phi$ in (67), (69) and $w=R_2^2T_{12}\phi$ in (68) and add; then,

$$\begin{aligned}&(a_2^{-1}R_2^1T_{12}\phi, R_2^1T_{12}\phi)_{\Omega_2}+(c_2R_2^2T_{12}\phi, R_2^2T_{12}\phi)_{\Omega_2}\\&=(a_1^{-1}R_1^1\phi, R_1^1T_{12}\phi)_{\Omega_1}+(c_1R_1^2\phi, R_1^2T_{12}\phi)_{\Omega_1},\quad\forall\phi\in\Phi,\end{aligned}\tag{70}$$

where we used the fact that

$$(R_1^2\phi,\operatorname{div} R_1^1T_{12}\phi)_{\Omega_1}=(\operatorname{div} R_1^1\phi, R_1^2T_{12}\phi)_{\Omega_1}.$$

Combining (70) and (49) yields

$$|T_{12}\phi|_2\le\sqrt{\sigma}|\phi|_1,\qquad\forall\phi\in\Phi.\tag{71}$$

In particular,

$$|T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2\le\sqrt{\sigma}|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1,\qquad\forall\phi\in\Phi.\tag{72}$$

Now from (67) we have, $\forall\phi,\psi\in\Phi$,

$$\begin{aligned}&(a_2^{-1}R_2^1T_{12}\psi, R_2^1\phi)_{\Omega_2}-(R_2^2T_{12}\psi,\operatorname{div} R_2^1\phi)_{\Omega_2}\\&\quad=(a_1^{-1}R_1^1\psi, R_1^1\phi)_{\Omega_1}-(R_1^2\psi,\operatorname{div} R_1^1\phi)_{\Omega_1}.\end{aligned}\tag{73}$$

Since $(R_2^2\phi,\operatorname{div} R_2^1T_{12}\psi)_{\Omega_2}=(R_2^2T_{12}\psi,\operatorname{div} R_2^1\phi)_{\Omega_2}$, then (73) becomes

$$\begin{aligned}&(a_2^{-1}R_2^1\phi, R_2^1T_{12}\psi)_{\Omega_2}-(R_2^2\phi,\operatorname{div} R_2^1T_{12}\psi)_{\Omega_2}\\&\quad=(a_1^{-1}R_1^1\psi, R_1^1\phi)_{\Omega_1}-(R_1^2\psi,\operatorname{div} R_1^1\phi)_{\Omega_1}.\end{aligned}\tag{74}$$

In view of (68) and (69) we have

$$(\operatorname{div} R_2^1T_{12}\psi, R_2^2\phi)_{\Omega_2}+(c_2R_2^2T_{12}\psi, R_2^2\phi)_{\Omega_2}=0,\tag{75}$$

$$(\operatorname{div} R_1^1\phi, R_1^2\psi)_{\Omega_1}+(c_1R_1^2\phi, R_1^2\psi)_{\Omega_1}=0.\tag{76}$$

Combining (74), (75) and (76) yields

$$\begin{aligned}&(a_2^{-1}R_2^1\phi, R_2^1T_{12}\psi)_{\Omega_2}+(c_2R_2^2\phi, R_2^2T_{12}\psi)_{\Omega_2}\\&\quad=(a_1^{-1}R_1^1\phi, R_1^1\psi)_{\Omega_1}+(c_1R_1^2\phi, R_1^2\psi)_{\Omega_1},\quad\forall\phi,\ \psi\in\Phi.\end{aligned}\tag{77}$$

Thus, by (77), (72), (44), and (45),

$$\begin{aligned}
&\big(a_2^{-1}R_2^1T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^1\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2} + \big(c_2R_2^2T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^2\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2}\\
&= \big(a_1^{-1}R_1^1\gamma_0 r_2^{n+1/2}\cdot\nu_2, R_1^1\gamma_0 r_1^{n+1/2}\cdot\nu_1\big)_{\Omega_1} + \big(c_1R_1^2\gamma_0 r_2^{n+1/2}\cdot\nu_2, R_1^2\gamma_0 r_1^{n+1/2}\cdot\nu_1\big)_{\Omega_1}\\
&= -|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2 \le -\frac{1}{\sigma}|T_{12}\gamma_0 r_2^{n+1/2}\cdot\nu_2|_2^2. \qquad (78)
\end{aligned}$$

The formula (78) also gives

$$\begin{aligned}
&|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2\\
&= -\big(a_2^{-1}R_2^1T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^1\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2} - \big(c_2R_2^2T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1, R_2^2\gamma_0 r_2^{n+1/2}\cdot\nu_2\big)_{\Omega_2}\\
&\le |T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2|\gamma_0 r_2^{n+1/2}\cdot\nu_2|_2\\
&\le |T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2\\
&\le \sqrt{\tau}|T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1,
\end{aligned}$$

which leads to

$$|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1 \le \sqrt{\tau}|T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2. \qquad (79)$$

Thus

$$|\gamma_0 r_2^{n+1/2}\cdot\nu_2|_2^2 \le \tau|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2 \le \tau^2|T_{12}\gamma_0 r_1^{n+1/2}\cdot\nu_1|_2^2. \qquad (80)$$

Combining (66), (78), and (80) gives (65).

Theorem 3.3.

$$|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2 \le \left[\alpha^2 - \frac{2\alpha(1-\alpha)}{\tau} + (1-\alpha)^2\sigma^2\right]|\gamma_0 r_1^n\cdot\nu_1|_1^2. \qquad (81)$$

Proof. By (44) and (48), we have

$$\begin{aligned}
|\gamma_0 r_1^{n+1/2}\cdot\nu_1|_1^2 = \;& \alpha^2|\gamma_0 r_1^n\cdot\nu_1|_1^2 + (1-\alpha)^2|\gamma_0 r_2^n\cdot\nu_2|_1^2 \qquad (82)\\
&-2\alpha(1-\alpha)\big(a_1^{-1}R_1^1\gamma_0 r_1^n\cdot\nu_1, R_1^1\gamma_0 r_2^n\cdot\nu_2\big)_{\Omega_1}\\
&-2\alpha(1-\alpha)\big(c_1R_1^2\gamma_0 r_1^n\cdot\nu_1, R_1^2\gamma_0 r_2^n\cdot\nu_2\big)_{\Omega_1}.
\end{aligned}$$

We now estimate (82) term by term. From (37)-(38) and (40), we have

$$\begin{aligned}
&(a_1^{-1}r_1^n, v)_{\Omega_1} - (e_1^n, \operatorname{div} v)_{\Omega_1} \qquad (83)\\
&\quad = (a_2^{-1}r_2^n, R_2^1(\gamma_0 v\cdot\nu_1))_{\Omega_2} - (e_2^n, \operatorname{div} R_2^1(\gamma_0 v\cdot\nu_1))_{\Omega_2},\ \forall v\in V_1^h,\\
&(\operatorname{div} r_1^n, w)_{\Omega_1} + (c_1e_1^n, w)_{\Omega_1} = 0, \quad \forall w\in W_1^h, \qquad (84)\\
&(\operatorname{div} R_2^1\gamma_0 v\cdot\nu_1, w)_{\Omega_2} + (c_2R_2^2\gamma_0 v\cdot\nu_1, w)_{\Omega_2} = 0, \quad \forall v\in V_1^h,\ w\in W_2^h. \qquad (85)
\end{aligned}$$

Letting $v = r_1^n$ in (83) and (85), $w = e_1^n$ in (84), and $w = e_2^n$ in (85) and adding, we obtain

$$(a_1^{-1}r_1^n, r_1^n)_{\Omega_1} + (c_1e_1^n, e_1^n)_{\Omega_1} = (a_2^{-1}r_2^n, R_2^1\gamma_0 r_1^n\cdot\nu_1)_{\Omega_2} + (c_2e_2^n, R_2^2\gamma_0 r_1^n\cdot\nu_1)_{\Omega_2}. \qquad (86)$$

Thus

$$|\gamma_0 r_1^n\cdot\nu_1|_1^2 \le |\gamma_0 r_2^n\cdot\nu_2|_2|\gamma_0 r_1^n\cdot\nu_1|_2 \le \sqrt{\tau}|\gamma_0 r_2^n\cdot\nu_2|_2|\gamma_0 r_1^n\cdot\nu_1|_1,$$

which gives

$$|\gamma_0 r_1^n\cdot\nu_1|_1 \le \sqrt{\tau}|\gamma_0 r_2^n\cdot\nu_2|_2. \qquad (87)$$

Similarly,

$$|\gamma_0 r_2^n\cdot\nu_2|_2 \le \sqrt{\sigma}|\gamma_0 r_1^n\cdot\nu_1|_1. \qquad (88)$$

Now combining (37)-(38) and (40) and noting that $R_2^1\gamma_0(R_1^1\gamma_0 r_2^n\cdot\nu_2)\cdot\nu_1 = R_2^1\gamma_0(r_2^n\cdot\nu_2) = r_2^n$, we obtain

$$\begin{aligned}
&(a_1^{-1}r_1^n, R_1^1\gamma_0 r_2^n\cdot\nu_2)_{\Omega_1} - (e_1^n, \operatorname{div} R_1^1\gamma_0 r_2^n\cdot\nu_2)_{\Omega_1} \qquad (89)\\
&\quad = (a_2^{-1}r_2^n, r_2^n)_{\Omega_2} - (e_2^n, \operatorname{div} r_2^n)_{\Omega_2},\\
&(\operatorname{div} R_1^1\gamma_0 r_2^n\cdot\nu_2, w)_{\Omega_1} + (c_1R_1^2\gamma_0 r_2^n\cdot\nu_2, w)_{\Omega_1} = 0, \quad \forall w\in W_1^h, \qquad (90)\\
&(\operatorname{div} r_2^n, w)_{\Omega_2} + (c_2e_2^n, w)_{\Omega_2} = 0,\ w\in W_2^h. \qquad (91)
\end{aligned}$$

Letting $w = e_1^n$ in (90) and $w = e_2^n$ in (91), and combining with (89) we see that

$$(a_1^{-1} r_1^n, R_1^1 \gamma_0 r_2^n \cdot \nu_2)_{\Omega_1} + (c_1 e_1^n, R_1^2 \gamma_0 r_2^n \cdot \nu_2)_{\Omega_1} = (a_2^{-1} r_2^n, r_2^n)_{\Omega_2} + (c_2 e_2^n, e_2^n)_{\Omega_2}. \quad (92)$$

Thus, from (92), (87), (88), and (82),

$$\begin{aligned} |\gamma_0 r_1^{n+1/2} \cdot \nu_1|_1^2 &= \alpha^2 |\gamma_0 r_1^n \cdot \nu_1|_1^2 + (1-\alpha)^2 |\gamma_0 r_2^n \cdot \nu_2|_1^2 \quad (93) \\ &\quad -2\alpha(1-\alpha)\big[(a_2^{-1} r_2^n, r_2^n)_{\Omega_2} + \big(c_1 e_2^n, e_2^n\big)_{\Omega_2}\big] \\ &\le \left[\alpha^2 - \frac{2\alpha(1-\alpha)}{\tau} + (1-\alpha)^2 \sigma^2\right] |\gamma_0 r_1^n \cdot \nu_1|_1^2. \quad (94) \end{aligned}$$

This completes the proof.

Theorem 3.4.

$$|T_{12} \gamma_0 r_2^{n+1/2} \cdot \nu_2|_2^2 \le \left[\alpha^2 - \frac{2\alpha(1-\alpha)}{\tau} + (1-\alpha)^2 \sigma^2\right] |\gamma_0 r_2^n \cdot \nu_2|_2^2. \quad (95)$$

Proof. From (37)-(38) and (39)-(41), we see that

$$R_1^3(\gamma_0 r_1^n \cdot \nu_1) = \beta \delta_1^{n-1/2} + (1-\beta)\delta_2^{n-1/2}. \quad (96)$$

By the definition of the operator T_{12}, (96), and (37)-(38) we have

$$\gamma_0 r_2^n \cdot \nu_2 = T_{12} \gamma_0 r_1^n \cdot \nu_1. \quad (97)$$

By (45), (48), and (97), we have

$$\begin{aligned} |T_{12} \gamma_0 r_2^{n+1/2} \cdot \nu_2|_2^2 &= |T_{12}[-\alpha \gamma_0 r_1^n \cdot \nu_1 + (1-\alpha)\gamma_0 r_2^n \cdot \nu_2]|_2^2 \\ &= |-\alpha \gamma_0 r_2^n \cdot \nu_2 + (1-\alpha) T_{12} \gamma_0 r_2^n \cdot \nu_2|_2^2 \\ &= \alpha^2 |\gamma_0 r_2^n \cdot \nu_2|_2^2 + (1-\alpha)^2 |T_{12} \gamma_0 r_2^n \cdot \nu_2|_2^2 \quad (98) \\ &\quad -2\alpha(1-\alpha)\big(a_2^{-1} R_2^1 \gamma_0 r_2^n \cdot \nu_2, R_2^1 T_{12} \gamma_0 r_2^n \cdot \nu_2\big)_{\Omega_2} \\ &\quad -2\alpha(1-\alpha)\big(c_2 R_2^2 \gamma_0 r_2^n \cdot \nu_2, R_2^2 T_{12} \gamma_0 r_2^n \cdot \nu_2\big)_{\Omega_2}. \end{aligned}$$

In view of (71) and (49) we obtain

$$|T_{12} \gamma_0 r_2^n \cdot \nu_2|_2^2 \le \sigma |\gamma_0 r_2^n \cdot \nu_2|_1^2 \le \sigma^2 |\gamma_0 r_2^n \cdot \nu_2|_2^2. \quad (99)$$

From (77) and (49) we have

$$\begin{aligned} &\big(a_2^{-1} R_2^1 \gamma_0 r_2^n \cdot \nu_2, R_2^1 T_{12} \gamma_0 r_2^n \cdot \nu_2\big)_{\Omega_2} + \big(c_2 R_2^2 \gamma_0 r_2^n \cdot \nu_2, R_2^2 T_{12} \gamma_0 r_2^n \cdot \nu_2\big)_{\Omega_2} \\ &= \big(a_1^{-1} R_1^1 \gamma_0 r_2^n \cdot \nu_2, R_1^1 \gamma_0 r_2^n \cdot \nu_2\big)_{\Omega_1} + \big(c_1 R_1^2 \gamma_0 r_2^n \cdot \nu_2, R_1^2 \gamma_0 r_2^n \cdot \nu_2\big)_{\Omega_1} \quad (100) \\ &= |\gamma_0 r_2^n \cdot \nu_2|_1^2 \ge \frac{1}{\tau} |\gamma_0 r_2^n \cdot \nu_2|_2^2. \end{aligned}$$

Combining (98), (99), and (100) leads to (95).

Theorem 3.5. *Let*

$$\rho(\alpha, \beta, \sigma, \tau) = \left[\alpha^2 + (1-\alpha)^2 \sigma^2 - \frac{2\alpha(1-\alpha)}{\tau}\right] \left[\beta^2 + (1-\beta)^2 \tau^2 - \frac{2\beta(1-\beta)}{\sigma}\right].$$

Then

$$(a_1^{-1} r_1^{n+1}, r_1^{n+1})_{\Omega_1} + (c_1 e_1^{n+1}, e_1^{n+1})_{\Omega_1} \le \rho(\alpha, \beta, \sigma, \tau) \left[(a_1^{-1} r_1^n, r_1^n)_{\Omega_1} + (c_1 e_1^n, e_1^n)_{\Omega_1}\right], \quad (101)$$

$$(a_2^{-1} r_2^{n+1}, r_2^{n+1})_{\Omega_2} + (c_2 e_2^{n+1}, e_2^{n+1})_{\Omega_2} \le \rho(\alpha, \beta, \sigma, \tau) \left[(a_2^{-1} r_2^n, r_2^n)_{\Omega_2} + (c_2 e_2^n, e_2^n)_{\Omega_2}\right], \quad (102)$$

where the errors r_i^n and e_i^n are defined by (34)-(38).

Proof. From (37)-(38), it follows that

$$\{r_i^{n+1}, e_i^{n+1}\} = \{R_i^1(\gamma_0 r_i^{n+1} \cdot \nu_i), R_i^2(\gamma_0 r_i^{n+1} \cdot \nu_i)\}. \quad (103)$$

Thus, by (48), we have

$$|\gamma_0 r_i^{n+1} \cdot \nu_i|_i^2 = (a_i^{-1} r_i^{n+1}, r_i^{n+1})_{\Omega_i} + (c_i e_i^{n+1}, e_i^{n+1})_{\Omega_i}. \quad (104)$$

Combining Theorems 3.1, 3.2, 3.3, and 3.4 we then finish the proof of this theorem.

Corollary 3.1. *The iterative algorithm (30)-(33) converges if α and β are chosen such that*

$$\max\{0, 1 - \frac{2(\tau+1)}{\tau\sigma^2+\tau+2}\} < \alpha < 1, \quad \max\{0, 1 - \frac{2(\sigma+1)}{\tau^2\sigma+\sigma+2}\} < \beta < 1.$$

Corollary 3.2. *The optimal relaxation parameters are $\alpha = \frac{\sigma^2\tau+1}{\tau\sigma^2+\tau+2}$ and $\beta = \frac{\sigma\tau^2+1}{\tau^2\sigma+\sigma+2}$.*

Corollary 3.3. *When $\sigma = \tau = 1$, which is true if Ω_1 and Ω_2 are symmetric with respect to the interface and the coefficients $a_1 = a_2$ and $c_1 = c_2$ are constants, then (30)-(33) with $\beta = 1/2$ and $\alpha \in (0,1)$ converges in one iteration no matter what the initial guess is.*

Note that Theorem 3.2 could be changed such that $|\gamma_0 r_2^{n+1} \cdot \nu_2|_2$ was bounded in terms of $|\gamma_0 r_2^{n+1/2} \cdot \nu_2|_2$ instead of $|T_{12}\gamma_0 r_2^{n+1/2} \cdot \nu_2|_2$. But this would lead to a recursive relation that is hard to analyze. In particular, it was not obvious how to obtain optimal relaxation parameters like the ones given Corollary 3.2. In fact, such recursive relation would not guarantee convergence no matter how α and β were chosen. In contrast, our Theorem 3.5 guarantees convergence no matter how large the jumps are in the solutions and coefficients across the interface, provided α and β are suitably chosen. Besides, an expression for optimal relaxation parameters is obtained. Note that the optimal values of α and β in Corollary 3.2 can be approximated numerically and they can be used to achieve nearly optimal convergence.

4. Numeric Examples

In this section, we present some numerical experiments for our iterative procedure using triangular lowest order Raviart-Thomas mixed finite elements in two dimensions. Numeric examples will show that our iterative method is insensitive to discontinuous solutions and variable coefficients even with big jumps across the interface. We will apply approximated optimal relaxation parameters α and β in all of our numerical tests, which would not converge for wildly behaved coefficients with $\alpha = \beta = 1/2$. Sharp interfaces of the true solution can also be captured fairly easily and accurately. For simplicity, we will test our algorithm on problems with a simple interface, although our theory permits rather arbitrary ones when a sophisticated grid generator is available. Subregion problems are solved by GMRES with symmetric SOR preconditioning. Only nonzero entries of subregion matrices are stored in the compressed sparse row format [27]. This means that all neighboring nodes for each node in a subregion triangulation need be found and sorted. Since the number of neighboring nodes varys with each node and with different triangulations, allocating a large array for each node would unnecessarily increase memory storage. We use a linked list for finding and sorting the neighboring nodes. Our implementation is in C++ with double precision on an SGI machine.

For the lowest order Raviart-Thomas mixed finite elements, pressure is constant on each triangle and is discontinuous across different triangles. The flux on each triangle is a vector function $\phi(x) = [a + cx, b + cy]^T$ in two dimensions. Let p_i be the middle point on side i of a reference triangle and ν_i the unit outward normal at p_i, $i = 1, 2, 3$. Three basis functions $\phi_i (i = 1, 2, 3)$ can be constructed corresponding to p_i such that $\phi_i \cdot \nu_i(p_j) = \delta_{ij}$. Then the approximate flux function $q(x)$ on the triangle can be written as a linear combination of the basis functions: $q(x) = q_1\phi_1(x) + q_2\phi_2(x) + q_3\phi(x)$, where $q_i = q \cdot \nu_i$ is the flux unknown at node p_i, $i = 1, 2, 3$. Note that the global flux is continuous across the middle points p_i from one triangle to another.

In all of our test, we let the domain $\Omega = (0,1) \times (0,1)$ and the interface Γ be at the line $x = y$, which divides the domain into two subregions $\Omega_1 = \{(x,y) : 0 < x, y < 1 \text{ and } x < y\}$ and $\Omega_2 = \{(x,y) : 0 < x, y < 1 \text{ and } x > y\}$. Uniform triangular grids are used in each of the subregions. All initial guesses are chosen to be zero. Optimal relaxation parameters α and β are obtained by an iterative algorithm that generates random numbers on the interface and extends them to subregions:

Algorithm 4.1. *optipms(M,ϵ)*

$\sigma = \tau = \sigma p = \tau p = 0$;

for $(n = 0, 1, \cdots, M)$ *do*

generate random numbers for vector ϕ *on the interface;*

Extend ϕ *to* Ω_1 *and compute* $r_1 = |\phi|_1^2$;

Extend ϕ *to* Ω_2 *and compute* $r_2 = |\phi|_2^2$;

$\sigma = \max(\sigma, r_1/r_2)$;

$\tau = \max(\tau, r_2/r_1)$;

if $(\max(|\sigma - \sigma p|, |\tau - \tau p|) \le \epsilon \quad and \quad n \ge 1)$ *break;*

$\sigma p = \sigma$;

$\tau p = \tau$;

enddo

$\alpha = (\sigma\tau^2 + 1)/(\tau^2\sigma + \sigma + 2)$;

$\beta = (\tau\sigma^2 + 1)/(\sigma^2\tau + \tau + 2)$;

output $\alpha,\ \beta$;

In particular, we will use *optipms*$(100, 10^{-4})$. That is, we choose $M = 100$ and $\epsilon = 10^{-4}$ in the algorithm. In reality, the *for* loop in the algorithm is broken within a few iterations (1 to 4 in our examples) for obtaining fairly good approximations α and β. The algorithm is very fast due to the random vectors ϕ generated on the interface, and the stronger the discontinuity in the coefficients, the faster convergence it has.

We consider the following four examples with continuous and strongly discontinuous coefficients across the interface.

Example 4.1.

$$
\begin{aligned}
&-\frac{\partial}{\partial x}(e^x \frac{\partial u_1}{\partial x}) - \frac{\partial}{\partial y}(e^y \frac{\partial u_1}{\partial y}) + \frac{1}{1+x+y} u_1 = f_1, \quad in\ \Omega_1,\\
&-\delta(\frac{\partial^2 u_2}{\partial x^2} + \frac{\partial^2 u_2}{\partial y^2}) = f_2, \quad in\ \Omega_2,\\
&-\nabla u_1 \cdot \nu_1 = g_1,\ on\ \partial\Omega_1 \cap \partial\Omega,\\
&u_2 = g_2,\ on\ \partial\Omega_2 \cap \partial\Omega,
\end{aligned}
$$

where ν_1 is the unit outward normal along $\partial\Omega_1$. The value of δ stands for the strength of discontinuity in the coefficients and will be given in the tables below. The functions $f_1,\ f_2,\ g_1,\ g_2$ $(\mu = \eta = 0)$ are chosen such that the exact solution is discontinuous across the interface and given by:

$$u_1(x, y) = 10x + y,\ in\ \Omega_1, \quad u_2(x, y) = sin(x + y),\ in\ \Omega_2.$$

Example 4.2.

$$
\begin{aligned}
&-\nabla \cdot \left(\begin{bmatrix} \delta e^x & \delta x \\ \delta x & \delta e^y \end{bmatrix} \nabla u_1 \right) = f_1(x, y), \quad in\ \Omega_1,\\
&-\nabla \cdot \left(\begin{bmatrix} x+1 & \sin(xy) \\ \sin(xy) & y+1 \end{bmatrix} \nabla u_2 \right) + (2 + \sin(x) + \cos(y))u_2 = f_2(x, y), \quad in\ \Omega_2,\\
&u_k = g_k, \quad on\ \partial\Omega_k \cap \partial\Omega, k = 1, 2.
\end{aligned}
$$

The value of δ stands for the strength of discontinuity in the coefficients and will be given in the tables below. The functions $f_1,\ f_2,\ g_1,\ g_2$ ($\mu = \eta = 0$) are chosen such that the exact solution is

$$u_1(x,y) = 2e^{xy}sin(3x)cos(3y),\ in\ \Omega_1, \quad u_2(x,y) = 2e^{xy}sin(3x)cos(3y),\ in\ \Omega_2.$$

Example 4.3.

$$-\nabla\cdot\left(\left[\begin{array}{cc}\delta e^x & \delta x\\ \delta x & \delta e^y\end{array}\right]\nabla u_1\right) + \delta\frac{u_1}{1+x+y} = f_1(x,y), \quad in\ \Omega_1,$$
$$-\nabla\cdot\left(\left[\begin{array}{cc} e^x & x\\ x & e^y\end{array}\right]\nabla u_1\right) + \frac{u_1}{1+x+y} = f_2(x,y), \quad in\ \Omega_2,$$
$$u_k = g_k, \quad on\ \partial\Omega_k\cap\partial\Omega, k = 1,2.$$

The value of δ stands for the strength of discontinuity in the coefficients and will be given in the tables below. The functions $f_1,\ f_2,\ g_1,\ g_2,\ \mu,\ \eta$ are chosen such that the exact solution has a jump discontinuity:

$$u_1(x,y) = e^{xy},\ in\ \Omega_1, \quad u_2(x,y) = 5e^{xy},\ in\ \Omega_2.$$

Example 4.4.

$$-\nabla\cdot\left(\left[\begin{array}{cc}\delta e^x & \delta x\\ \delta x & \delta e^y\end{array}\right]\nabla u_1\right) = e^{x+y}, \quad in\ \Omega_1,$$
$$-\nabla\cdot\left(\left[\begin{array}{cc} e^x & x\\ x & e^y\end{array}\right]\nabla u_1\right) + \delta\frac{u_1}{1+x+y} = \sin(x+y)\cos(13xy), \quad in\ \Omega_2,$$
$$u_1 = 5, \quad on\ \partial\Omega_1\cap\partial\Omega,$$
$$u_2 = x+y, \quad on\ \partial\Omega_2\cap\partial\Omega.$$

The value of δ stands for the strength of discontinuity in the coefficients and will be given in the tables below. The exact solution is unknown so that we only check the speed of convergence.

Our numerical results will show true errors for both pressure and flux between the numerical solution and the given exact solution at the time the iterative process is stopped when the iterative error is less than 10^{-5}. At the m-th iteration level, they are defined in Sobolev space norms as:

$$F_m = \max\left\{\frac{\|Q_m - Q_{m-1}\|_{H(\mathrm{div};\Omega_1)}}{\|Q_m\|_{H(\mathrm{div};\Omega_1)}}, \frac{\|Q_m - Q_{m-1}\|_{H(\mathrm{div};\Omega_2)}}{\|Q_m\|_{H(\mathrm{div};\Omega_2)}}\right\},$$
$$P_m = \max\left\{\frac{\|U_m - U_{m-1}\|_{L^2(\Omega_1)}}{\|U_m\|_{L^2(\Omega_1)}}, \frac{\|U_m - U_{m-1}\|_{L^2(\Omega_2)}}{\|U_m\|_{L^2(\Omega_2)}}\right\},$$
$$\text{Iterative error} = \max\{F_m, P_m\}, \tag{105}$$
$$\text{True flux error} = \max\left\{\frac{\|Q_m - q\|_{H(\mathrm{div};\Omega_1)}}{\|q\|_{H(\mathrm{div};\Omega_1)}}, \frac{\|Q_m - q\|_{H(\mathrm{div};\Omega_2)}}{\|q\|_{H(\mathrm{div};\Omega_2)}}\right\}, \tag{106}$$
$$\text{True pressure error } = \max\left\{\frac{\|U_m - u\|_{L^2(\Omega_1)}}{\|u\|_{L^2(\Omega_1)}}, \frac{\|U_m - u\|_{L^2(\Omega_2)}}{\|u\|_{L^2(\Omega_2)}}\right\}, \tag{107}$$

where U_m and Q_m denote approximate solution and flux at iteration m, u and q denote exact solution and flux, and $\|\cdot\|_{H(\mathrm{div};\Omega_k)}$ and $\|\cdot\|_{L^2(\Omega_k)}$ denote the discrete norms of

$$\|v\|_{H(\mathrm{div};\Omega_k)} = \left(\|v\|^2_{L^2(\Omega_k)} + \|\mathrm{div}\ v\|^2_{L^2(\Omega_k)}\right)^{1/2},$$
$$\|v\|_{L^2(\Omega_k)} = \left(\int_{\Omega_k}|v|^2dx\right)^{1/2}.$$

The iterative process will be stopped when the iterative error, defined by (105), is less than 10^{-5}. The the number of iterations and true flux and pressure errors, defined by (106) and

(107), will be reported for Examples 4.1, 4.2, and 4.3. However, for Example 4.4, only the iterative error, defined by (105), is reported to show the speed of convergence.

4.1. Numeric Experiments on Matching Grids

In this section, we conduct numerical experiments with the grids from different subregions matching at the interface; that is, they form a valid triangulation in the whole domain Ω.

We first test the dependency of the convergence upon the grid size. Two uniform triangular grid sizes $\frac{1}{10} \times \frac{1}{10}$ and $\frac{1}{20} \times \frac{1}{20}$ are applied to Examples 4.1, 4.2, 4.3, and 4.4, with results shown in Tables 4.1, 4.2, 4.3, and 4.4, respectively. Note that our algorithm contains a fractional step, at which the iterative errors are also shown in Table 4.4. From these results, we see the iterative errors in the relative Sobolev norms are essentially independent of the grid size. The same conclusion has been claimed for similar methods [21, 22, 39, 38].

We now observe the accuracy of our method in capturing strong discontinuities in the solution, its conormal derivative and the coefficients. From Table 4.1 for Example 4.1, Table 4.2 for Example 4.2, and Table 4.3 for Example 4.3, we see that our method is very accurate after a few iterations for both the pressure and flux variables. The accuracy of our method is observed to be independent of the coefficient jump, which is the greatest advantage of our method. No other numerical methods are known to share the same property for the general problem (5)-(8). Notice that the methods in [24, 31] have accuracy independent of the coefficient jumps only for a special class of problems. Furthermore, a first order accuracy is observed for both flux and pressure variables. This is consistent with standard results of mixed finite element methods [4, 25] for well-behaved coefficients.

Finally we observe the dependence of convergence upon the coefficient jumps. From Tables 4.1, 4.2, 4.3, and 4.4, our method also converges very fast even for very large coefficient jumps. The convergence tends to be faster when the jump gets larger from $\delta = 1$ up to $\delta = 10^{-12}$. This can be explained theoretically by our error reduction formula in Theorem 3.5. When the coefficient jump is getting larger, one of σ and τ, defined by (49), will go to infinity and the other goes to zero. With the optimal relaxation parameters chosen according to Corollary 3.2, one of α and β will go to 1 while the other go to 0. Then the error reduction factor $\rho(\alpha, \beta, \sigma, \tau)$ in Theorem 3.5 will approach zero. In other words, in the absence of approximation errors, the larger the coefficient jump is, the faster convergence we should have. However, this is not true when the coefficient jump is $\delta = 10^{-15}$. This could be caused by the mixed finite element method. Notice that we need to compute the inverse matrix a_i^{-1} in the iterative method and δ appears in the denominator of this inverse. When $\delta = 10^{-15}$, this could cause a lot of numerical errors in the iterative method.

4.2. Numeric Experiments on Non-Matching Grids

In this section, we conduct numerical experiments with the grids from different subregions not matching at the interface; that is, they do not form a triangulation in the whole domain Ω. See Figure 4.2 for an example of a non-matching grid. Thus the grids in different subregions can be generated separately. This provides great flexibility in grid generation as the interface can be of arbitrary shape and it is much more difficult to generate grids in the subregions that match each other at the interface. Our current theory in the paper needs to be extended to this situation, but we want to see how well our method performs numerically. When the grids do not match at the interface, a linear interpolation is used to interpolate from one grid to another at the two sides of the interface.

For non-matching grids, the convergence is much slower. Thus the stopping criterion is now changed to when the iterative error, defined by (105), is less than 10^{-2}. As soon as this criterion is met, the iteration is stopped and the true flux and pressure errors are computed according to (106) and (107). Table 4.5 shows the results for Example 4.2 in the cases of grid size $\frac{1}{10} \times \frac{1}{10}$ in subregion Ω_1 and grid size $\frac{1}{15} \times \frac{1}{15}$ in subregion Ω_2 (denoted by $\frac{1}{10} \times \frac{1}{15}$ for short; see Figure 4.2 for the grid), and grid size $\frac{1}{15} \times \frac{1}{15}$ in subregion Ω_1 and grid size $\frac{1}{20} \times \frac{1}{20}$ in subregion Ω_2

(denoted by $\frac{1}{15} \times \frac{1}{21}$ for short). In Table 4.6 we show the results for Example 4.3 in the cases of grid size $\frac{1}{15} \times \frac{1}{15}$ in subregion Ω_1 and grid size $\frac{1}{10} \times \frac{1}{10}$ in subregion Ω_2 (denoted by $\frac{1}{15} \times \frac{1}{10}$ for short) and grid size $\frac{1}{20} \times \frac{1}{20}$ in subregion Ω_1 and grid size $\frac{1}{15} \times \frac{1}{15}$ in subregion Ω_2 (denoted by $\frac{1}{20} \times \frac{1}{15}$ for short).

Table 4.1: True flux and true pressure errors, as defined in (106)-(107), and number of iterations when the iterative process is stopped for Example 4.1 with different grid sizes $\frac{1}{10} \times \frac{1}{10}$ and $\frac{1}{20} \times \frac{1}{20}$, and jump discontinuities $\delta = 1, 10^{-2}, 10^{-5}, 10^{-10}, 10^{-15}$, respectively.

	Grid size $= \frac{1}{10} \times \frac{1}{10}$			Grid size $= \frac{1}{20} \times \frac{1}{20}$		
Jumps	Iterations	flux error	pressure error	Iterations	flux error	pressure error
$\delta = 1$	5	3.39e-02	5.27e-02	5	1.64e-02	2.63e-02
$\delta = 10^{-2}$	3	3.59e-02	5.27e-02	3	1.67e-02	2.63e-02
$\delta = 10^{-5}$	2	3.59e-02	5.27e-02	2	1.67e-02	2.63e-02
$\delta = 10^{-10}$	2	3.59e-02	5.27e-02	2	1.67e-02	2.63e-02
$\delta = 10^{-15}$	4	3.59e-02	5.27e-02	7	1.67e-02	2.63e-02

Table 4.2: True flux and true pressure errors, as defined in (106)-(107), and number of iterations when the iterative process is stopped for Example 4.2 with different grid sizes $\frac{1}{10} \times \frac{1}{10}$ and $\frac{1}{20} \times \frac{1}{20}$, and jump discontinuities $\delta = 1, 10^{-2}, 10^{-5}, 10^{-10}, 10^{-15}$, respectively.

	Grid size $= \frac{1}{10} \times \frac{1}{10}$			Grid size $= \frac{1}{20} \times \frac{1}{20}$		
Jumps	Iterations	flux error	pressure error	Iterations	flux error	pressure error
$\delta = 1$	4	1.37e-01	1.03e-01	3	6.90e-02	5.19e-02
$\delta = 10^{-2}$	3	1.37e-01	1.03e-01	3	6.90e-02	5.19e-02
$\delta = 10^{-5}$	3	1.37e-01	1.03e-01	2	6.90e-02	5.18e-02
$\delta = 10^{-10}$	3	1.37e-01	1.03e-01	2	6.90e-02	5.19e-02
$\delta = 10^{-15}$	3	1.37e-01	1.03e-01	4	6.90e-02	5.18e-02

Table 4.3: True flux and true pressure errors, as defined in (106)-(107), and number of iterations when the iterative process is stopped for Example 4.3 with different grid sizes $\frac{1}{10} \times \frac{1}{10}$ and $\frac{1}{20} \times \frac{1}{20}$, and jump discontinuities $\delta = 1, 10^{-3}, 10^{-6}, 10^{-12}, 10^{-15}$, respectively.

	Grid size $= \frac{1}{10} \times \frac{1}{10}$			Grid size $= \frac{1}{20} \times \frac{1}{20}$		
Jumps	Iterations	flux error	pressure error	Iterations	flux error	pressure error
$\delta = 1$	2	8.58e-02	2.63e-02	2	4.29e-02	1.31e-02
$\delta = 10^{-3}$	2	8.64e-02	2.65e-02	2	4.30e-02	1.31e-02
$\delta = 10^{-6}$	2	8.64e-02	2.65e-02	2	4.30e-02	1.31e-02
$\delta = 10^{-12}$	2	8.64e-02	2.65e-02	2	4.30e-02	1.31e-02
$\delta = 10^{-15}$	3	8.64e-02	2.65e-02	4	4.30e-02	1.31e-02

Table 4.4: Iterative errors between current iteration and previous iteration, as defined in (105), at each iteration for Example 4.4 with different grid sizes $\frac{1}{10} \times \frac{1}{10}$ and $\frac{1}{20} \times \frac{1}{20}$, and jump discontinuities $\delta = 1, 10^{-3}, 10^{-6}, 10^{-12}, 10^{-15}$, respectively. Notice that each iteration contains two fractional steps and the iterative error at iteration 0 is always 1 since zero initial guess is used.

		Grid size $= \frac{1}{10} \times \frac{1}{10}$	Grid size $= \frac{1}{20} \times \frac{1}{20}$
Jumps	Iteration	iterative error	iterative error
$\delta = 1$	0	1.00e+00	1.00e+00
	0.5	1.02e+00	1.01e+00
	1.0	1.47e-02	1.92e-02
	1.5	3.04e-04	4.43e-04
	2.0	6.57e-06	1.14e-05
	2.5	–	2.87e-07
$\delta = 10^{-3}$	0	1.00e+00	1.00e+00
	0.5	1.41e+00	1.44e+00
	1.0	1.65e-03	1.71e-03
	1.5	3.58e-07	3.35e-07
$\delta = 10^{-6}$	0	1.00e+00	1.00e+00
	0.5	1.51e+00	1.54e+00
	1.0	1.56e-03	1.59e-03
	1.5	3.93e-09	7.06e-09
$\delta = 10^{-12}$	0	1.00e+00	1.00e+00
	0.5	1.51e+00	1.54e+00
	1.0	1.56e-03	1.60e-03
	1.5	1.12e-04	4.32e-04
	2.0	2.35e-07	7.41e-07
$\delta = 10^{-15}$	0	1.00e+00	1.00e+00
	0.5	1.54e+00	1.64e+00
	1.0	2.50e-02	7.72e-02
	1.5	7.33e-03	5.11e-02
	2.0	1.01e-04	2.41e-03
	2.5	4.46e-05	1.29e-03
	3.0	7.64e-07	8.12e-06

Table 4.5: True flux and true pressure errors, as defined in (106)-(107), and number of iterations when the iterative process is stopped for Example 4.2 with different non-matching grid sizes $\frac{1}{10} \times \frac{1}{15}$ and $\frac{1}{15} \times \frac{1}{20}$, and jump discontinuities $\delta = 1, 10^{-2}, 10^{-5}, 10^{-10}, 10^{-15}$, respectively.

	Grid size $= \frac{1}{10} \times \frac{1}{15}$			Grid size $= \frac{1}{15} \times \frac{1}{20}$		
Jumps	Iterations	flux error	pressure error	Iterations	flux error	pressure error
$\delta = 1$	2	1.38e-01	9.97e-02	2	9.20e-02	6.64e-02
$\delta = 10^{-2}$	2	1.38e-01	9.96e-02	2	9.20e-02	6.64e-02
$\delta = 10^{-5}$	2	1.38e-01	9.96e-02	2	9.20e-02	6.64e-02
$\delta = 10^{-10}$	2	1.38e-01	9.96e-02	2	9.20e-02	6.64e-02
$\delta = 10^{-15}$	3	1.38e-01	9.96e-02	3	9.20e-02	6.64e-02

A notable difference between matching and non-matching grids is the slow convergence when the grids do not match. However, the accuracy of the method, i.e. the error between the approximate solution and true solution for both flux and pressure variables, is about the same as in the matching grid case.

Table 4.6: True flux and true pressure errors, as defined in (106)-(107), and number of iterations when the iterative process is stopped for Example 4.3 with different non-matching grid sizes $\frac{1}{15} \times \frac{1}{10}$ and $\frac{1}{20} \times \frac{1}{15}$, and jump discontinuities $\delta = 1, 10^{-3}, 10^{-5}, 10^{-10}, 10^{-15}$, respectively.

	Grid size $= \frac{1}{15} \times \frac{1}{10}$			Grid size $= \frac{1}{20} \times \frac{1}{15}$		
Jumps	Iterations	flux error	pressure error	Iterations	flux error	pressure error
$\delta = 1$	5	8.02e-02	2.62e-02	1	5.36e-02	1.75e-02
$\delta = 10^{-3}$	2	8.03e-02	2.62e-02	2	5.36e-02	1.75e-02
$\delta = 10^{-5}$	2	8.03e-02	2.62e-02	2	5.36e-02	1.75e-02
$\delta = 10^{-10}$	2	8.03e-02	2.62e-02	2	5.36e-02	1.75e-02
$\delta = 10^{-15}$	2	8.03e-02	2.62e-02	3	5.36e-02	1.75e-02

The algorithm with non-matching grids is worth further investigation, since it offers an opportunity for easily and accurately tackling realistic interfaces such as in alloy solidification problems. In such problems, the interface between the two phases can be of arbitrary shape and grid generation can be applied in parallel and separately in subregions. Note that grid generation usually consumes a large portion of CPU time in a typical numerical simulation, and generating non-matching grids in parallel in subregions can dramatically improve the efficiency.

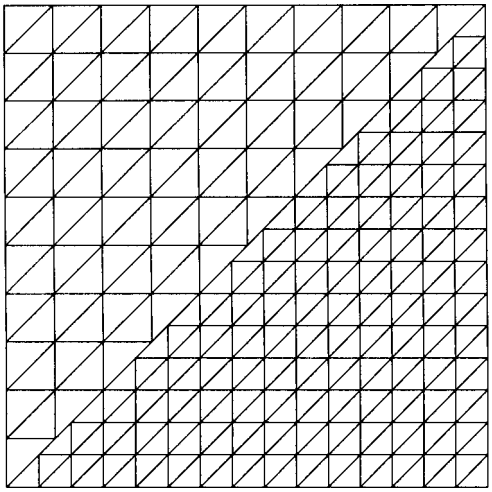

Fig 4.2. Non-matching grids at the interface with grid size $\frac{1}{10} \times \frac{1}{10}$ in subregion Ω_1 and grid size $\frac{1}{15} \times \frac{1}{15}$ in subregion Ω_2.

5. Concluding Remarks

We have proposed a numerical method to solve two-phase interface problems with possibly strongly discontinuous coefficients by iteratively solving smaller and well conditioned problems corresponding to each phase using the mixed finite element method with Lagrange multipliers at the interface. When the solution and its conormal derivative have known jumps at the interface, our method still works well without additional computational cost. Notice that a sophisticated technique is developed in [19, 20, 18, 32] to handle problems with discontinuous solutions and conormal derivatives at the interface, whose accuracy should decrease when the coefficient jump is large.

The advantage of our method is that its accuracy does not seem to deteriorate when the

coefficient jump is getting increasingly larger, provided the jump can be accurately represented on a given computer. Experimentally, it seems the coefficient jump $\delta(x)$ or its inverse (scaled to $0 < \delta(x) \leq 1$) must be larger than the machine epsilon. In double precision on most workstations, this machine epsilon is approximately 2.22×10^{-16}. Another advantage of our method is that it converges very fast even for strongly discontinuous coefficients. Notice that the conditioning of problems with strongly discontinuous coefficients is very bad and standard numerical methods converge very slowly on them, unless some sophisticated preconditioners are properly constructed and implemented. Besides, our method is guaranteed to converge, provided the relaxation parameters are chosen according to Corollary 3.2. In fact, these parameters may not have to be chosen exactly optimally. An approximation such as by Algorithm 4.1 with just a few iterations provides values for the parameters that lead to very fast convergence for our method (30)-(33).

Our numerical experiments have included the case in which the grids in different subregions do not match on the interface, but the convergence is faster when they match. The accuracy, however, does not seem to be affected by non-matching grids. This method is particularly suitable for object oriented programming [38, 41] and collaborative PDE solvers [23]. The accuracy, ease of implementation, and extendibility to more complex interface geometry make this method very competitive compared to methods studied previously [9, 10, 12, 11, 19, 18, 6, 31, 24]. In general, the interface can be of arbitrary geometry. Applying non-matching grids is a good feature and grids in different subregions can be generated separately and in parallel.

It should be noted that our algorithm does not seem to directly apply to more than two phases, in which more than two subdomains should be employed to tackle strong discontinuities in order to obtain accurate numerical solution.

In the future, we may try to solve two and three dimensional application problems with more complex geometry and interface. For some elliptic interface problems, the position of the interface is unknown and the iterative algorithm has to include a procedure to find an approximate location of the interface. For some time-dependent problems, the interface may move with time. Thus the subregion that each phase occupies and the finite element grid on it are varying dynamically. In this respect, dynamic finite element methods [33, 34, 36] may be applied.

References

[1] D. N. Arnold and F. Brezzi, Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates, *RAIRO Modél. Math. Anal. Numér.*, **19** (1985), 7-32.

[2] F. Brezzi, J. Douglas, Jr., R. Durán, and M. Fortin, Mixed finite elements for second order elliptic problems in three variables, *Numer. Math.*, **51** (1987), 237-250.

[3] F. Brezzi, J. Douglas, Jr., and L. D. Marini, Two families of mixed finite elements for second order elliptic problems, *Numer. Math.*, **47** (1985), 217-235.

[4] F. Brezzi and M. Fortin, Mixed and Hybrid Finite Element Methods, Springer-Verlag, New York, 1991.

[5] K. J. Binns, P. J. Lawrenson, and C. W. Trowbridge, The analytical and numerical solution of electric and magnetic fields, J.Wiley and Sons, New York, 1992.

[6] Y. Z. Cao and M. D. Gunzburger, Least-squares finite element approximations to solutions of interface problems. *SIAM J. Numer. Anal.*, **35** (1998), 393-405.

[7] P. G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland, Amsterdam, 1978.

[8] T. F. Chan and T. P. Mathew, Domain Decomposition Algorithms, volume 3 of *Acta Numerica*, pages 61–143. Cambridge University Press, Cambridge, 1994.

[9] A. B. Crowley and J. R. Ockendon, On the numerical solution of an alloy solidification problem, *Int. J. Heat Mass Transfer*, **22** (1979), 941–946.

[10] J. Crank, Free and Moving Boundary Problems, Oxford University Press, Oxford, 1984.

[11] A. B. Crowley, Alloy solidification problems, *Control Cybern.*, **14** (1985), 97–113.

[12] A. B. Crowley, Numerical solution of alloy solidification problems revisited, In A. Bossavit, *et al*, editor, Free boundary problems: applications and theory, Research Notes in Mathematics, volume 120, pages 122–131. Pitman, Boston, 1985.

[13] P. I. Crumpton, G. J. Shaw, and A. F. Ware, Discretisation and multigrid solution of elliptic equations with mixed derivative terms and strongly discontinuous coefficients, *J. Comput. Phys.*, **116** (1995), 343–358.

[14] Zhiming Chen and Jun Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, *Numer. Math.*, **79** (1998), 175–202.

[15] J. Douglas, Jr. and Daoqi Yang, Numerical experiments of a nonoverlapping domain decomposition method for partial differential equations. In D. Griffiths and G. A. Watson, editors, Numerical Analysis: A. R. Mitchell 75th Birthday Volume, pages 85–97. World Scientific, Singapore, 1996.

[16] B. X. Fraeijs de Veubeke, Displacement and equilibrium models in the finite element method, In O. C. Zienkiewicz and G. Holister, editors, Stress Analysis, New York, 1965, John Wiley.

[17] B. X. Fraeijs de Veubeke, Stress function approach, In International Congress on the Finite Element Method in Structural Mechanics, Bournemouth, 1975.

[18] Z. L. Li, A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, **35** (1998), 230–254.

[19] R. J. LeVeque and Z. L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, **31** (1994), 1019–1044.

[20] R. J. LeVeque and Z. L. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, *SIAM J. Sci. Comput.*, **18** (1997), 709–735.

[21] L. D. Marini and A. Quarteroni, An iterative procedure for domain decomposition methods: a finite element approach, In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, pages 129–143, Philadelphia, 1988, SIAM.

[22] L. D. Marini and A. Quarteroni, A relaxation procedure for domain decomposition methods using finite elements, *Numer. Math.*, **55** (1989), 575–598.

[23] M. Mu and J. R. Rice, Modeling with collaborating PDE solvers: theory and practice, *Computing Systems in Engigeering*, **6** (1995), 87–95.

[24] B. F. Nielsen, Finite element discretizations of elliptic problems in the presence of arbitrarily small ellipticity: An error analysis. *SIAM J. Numer. Anal.*, **36** (1999), 368–392.

[25] P.-A. Raviart and J.-M. Thomas, A mixed finite element method for second order elliptic problems, In I. Galligani and E. Magenes, editors, Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics 606, pages 92–315. Springer-Verlag, Berlin and New York, 1977.

[26] J. R. Rice, E. A. Vavalis, and Daoqi Yang, Convergence analysis of a nonoverlapping domain decomposition method for elliptic PDEs, *J. Comput. Appl. Math.*, **87** (1997), 11–19, 1997.

[27] Y. Saad, m Iterative Methods for Sparse Linear Systems, PWS Publishing Co., Boston, 1996.

[28] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, New York, 1996.

[29] J. Simkin and C. W. Trowbridge, On the use of the total scalar potential in the numerical solution of field problems in electromagnetics, *Internat. J. Numer. Methods Engrg.*, **14** (1979), 423–440.

[30] X. C. Tai and M. Espedal, Rate of convergence of some space decomposition methods for linear and nonlinear problems, *SIAM J. Numer. Anal.*, **35** (1998), 1558–1570.

[31] S. A. Vavasis, Stable finite elements for problems with wild coefficients, *SIAM J. Numer. Anal.*, **33** (1996), 890–916.

[32] A. Wiegmann and K. P. Bube, The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, **35** (1998), 177–200.

[33] Daoqi Yang, Mixed methods with dynamic finite-element spaces for miscible displacement in porous media, *J. Comput. Appl. Math.*, **30** (1990), 313–328.

[34] Daoqi Yang, Grid modification for second order hyperbolic problems, *Math. Comput.*, **64** (1995), 1495–1509.

[35] Daoqi Yang, A parallel iterative nonoverlapping domain decomposition procedure for elliptic problems, *IMA J. Numer. Anal.*, **16** (1996), 75–91.

[36] Daoqi Yang, Dynamic domain decomposition and grid modification for parabolic problems, *Computers and Mathematics with Applications*, **33** (1997), 89–103.

[37] Daoqi Yang, A parallel nonoverlapping Schwarz domain decomposition method for elliptic interface problems, Technical report, IMA preprint 1508, University of Minnesota, Minneapolis, 1997.

[38] Daoqi Yang, A nonoverlapping subdomain algorithm with Lagrange multipliers and its object oriented implementation for interface problems, In X. Cai, J. Mandel, and C. Farhat, editors, Domain Decomposition Methods 10, Contemporary Mathematics, Vol 218, pages 365–373, Providence, RI, 1998. American Mathematical Society.

[39] Daoqi Yang, A parallel iterative domain decomposition algorithm for elliptic problems, *J. Comput. Math.*, **16** (1998), 141–151.

[40] Daoqi Yang, Finite elements for elliptic problems with wild coefficients, *Mathematics and Computers in Simulation*, **54** (2000), 383–395.

[41] Daoqi Yang, C++ and Object Oriented Numeric Computing for Scientists and Engineers, Springer-Verlag, New York, 2001.