

A Method of Lines Based on Immersed Finite Elements for Parabolic Moving Interface Problems

Tao Lin¹, Yanping Lin^{2,3,*} and Xu Zhang¹

¹ Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA

² Department of Applied Mathematics, Hong Kong Polytechnic University, Hung Hom, Hong Kong

³ Department of Mathematical and Statistics Science, University of Alberta, Edmonton AB, T6G 2G1, Canada

Received 13 July 2012; Accepted (in revised version) 8 September 2012

Available online 7 June 2013

Dedicated to Graeme Fairweather on the occasion of his 70th birthday.

Abstract. This article extends the finite element method of lines to a parabolic initial boundary value problem whose diffusion coefficient is discontinuous across an interface that changes with respect to time. The method presented here uses immersed finite element (IFE) functions for the discretization in spatial variables that can be carried out over a fixed mesh (such as a Cartesian mesh if desired), and this feature makes it possible to reduce the parabolic equation to a system of ordinary differential equations (ODE) through the usual semi-discretization procedure. Therefore, with a suitable choice of the ODE solver, this method can reliably and efficiently solve a parabolic moving interface problem over a fixed structured (Cartesian) mesh. Numerical examples are presented to demonstrate features of this new method.

AMS subject classifications: 35R05, 65M20, 65M60

Key words: Immersed finite element, moving interface, method of lines, Cartesian mesh.

1 Introduction

In this article, we consider the following parabolic moving interface problem:

$$u_t - \nabla \cdot (\beta \nabla u) = f(t, X), \quad \text{if } X \in \Omega, \quad t \in (0, T_{end}], \quad (1.1a)$$

$$u(t, X) = g(t, X), \quad \text{if } X \in \partial\Omega, \quad t \in (0, T_{end}], \quad (1.1b)$$

$$u(0, X) = u_0(X), \quad \text{if } X \in \overline{\Omega}, \quad (1.1c)$$

*Corresponding author.

Email: tlin@math.vt.edu (T. Lin), malin@polyu.edu.hk (Y. Lin), xuz@vt.edu (X. Zhang)

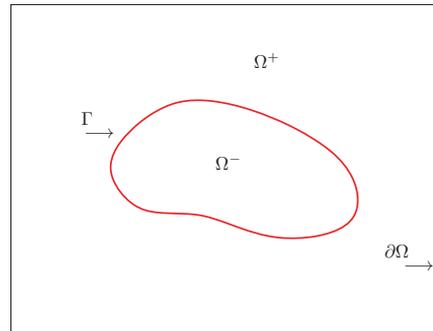


Figure 1: A sketch of the domain for the moving interface problem.

where the domain $\Omega \subset \mathbb{R}^2$ is assumed to be an open rectangle (or a union of open rectangles) that is separated into two sub-domains $\Omega^+(t)$ and $\Omega^-(t)$ by a curve $\Gamma(t)$ defined by a smooth function $\Gamma : [0, T_{end}] \rightarrow \Omega$, see Fig. 1 for an illustration of the solution domain Ω . The diffusion coefficient $\beta(t, X)$ is discontinuous across the interface $\Gamma(t)$. For simplicity's sake, we assume that $\beta(t, X)$ is a piece-wise constant function defined as follows:

$$\beta(t, X) = \begin{cases} \beta^-, & \text{if } X \in \Omega^-(t), \\ \beta^+, & \text{if } X \in \Omega^+(t). \end{cases} \quad (1.2)$$

Across the moving interface $\Gamma(t)$, the solution $u(t, X)$ is required to satisfy the usual jump conditions:

$$[u]|_{\Gamma(t)} = 0, \quad (1.3a)$$

$$[\beta \nabla u \cdot \mathbf{n}]|_{\Gamma(t)} = 0. \quad (1.3b)$$

The moving interface problem described by (1.1a)-(1.3b) appears in many applications, such as field injection problems [14, 15, 33, 35, 42] and Stefan problems [9, 34]. The two-phase Stefan problem consists of this kind of parabolic moving interface problem and an ordinary differential equation based on the physics for tracking the interface location between the two material phases. In this article, we are focusing on the difficulties in solving the parabolic initial boundary value problem with an evolving interface and hope that the method presented here can be extended to more complicated problems.

Conventional finite element (FE) methods can solve the parabolic differential equations (PDEs) satisfactorily [38]. In dealing with interface problems, if the interface does not change its shape and location, then methods such as those discussed in [38] can be straightforwardly utilized provided that the meshes are tailored to match the interface [2, 7, 10]; otherwise, their convergence might be impaired [5]. We call meshes of this type as body-fitting meshes, in which each element is essentially on one side of the interface, see the plot on the left in Fig. 2 for an illustration.

However, the requirement of using body-fitting mesh makes traditional FE methods inefficient for solving moving interface problems. First, for a problem with a moving

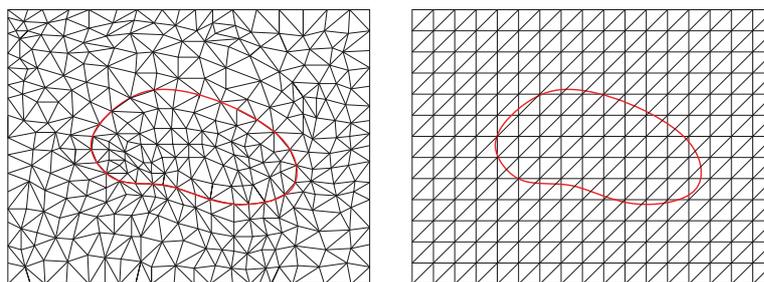


Figure 2: A body-fitting mesh on the left shows how elements are placed along an interface. A non-body-fitting (Cartesian) mesh on the right allows interface to cut through some elements.

interface, the body-fitting restriction requires a new mesh to be generated at each time level. This is a time-consuming task, especially for those applications with complicated moving interface. Secondly, if the interface changes with respect to time in a problem, as the consequence of having to use body-fitting meshes, the number and locations of global degrees of freedom and elements in meshes at two consecutive time levels in a method based on traditional FE functions will usually be different, and this causes many difficulties including, but not limited to, those in the following list:

- Indefinite Solution Dimensions:** Different number of nodes and elements in body-fitting meshes at two different time levels implies that the finite element spaces used at these time levels will have different global degrees of freedom. Consequently, the related FE equations (in either a semi-discrete scheme or a fully discrete scheme) will be defined through a non-square algebraic system which demands more efforts to solve. We note that it is possible to generate body-fitting meshes with the same number of nodes and/or elements at different time levels for a domain with a moving interface, but this usually requests an extra computational cost and has a great potential of losing accuracy unless the geometry of the interface changes in a simple way.
- The Loss of Local Assembling Procedure:** The so-called "local assembling" procedure is an important feature of FE methods. To assemble a global matrix in the algebraic system of a FE method, one can first construct the related local matrix in each element and then assemble its entries into the global matrix. This simplicity is lost for a moving interface problem when assembling a global matrix involves FE functions defined at two different time levels because their meshes usually do not share any common elements. Complicated and time-consuming quadrature procedures have to be developed for assembling matrices in conventional FE methods.
- Inapplicability of Methods of Lines:** The method of lines (MoL) [36, 37, 43] is an efficient technique for solving initial boundary value problems of parabolic PDEs. This technique reduces a PDE initial boundary value problem to an initial value problem of a system of ODEs. One can then solve this ODE system by an ODE solver with desirable features to generate a solution to the PDE problem. The abundant choices of efficient

and robust ODE solvers make the MoL popular for solving the time-dependent PDEs. However, for problems with moving interfaces, the body-fitting restriction on the meshes makes the application of the MoL difficult, if not impossible, in the FE formulation. The main obstacle is the change of global degrees of freedom with respect to the time, possibly in both number and locations, and this forbids a correct formulation of the ODEs in the semi-discretization for a time dependent PDE to be solved.

It is therefore desirable to develop numerical methods that can be carried out on meshes independent of the interface location so that non-body-fitting meshes, such as Cartesian meshes illustrated in the plot on the right in Fig. 2, can be used to solve problems with moving interfaces. Many efforts have been attempted to develop such solvers for interface problems. In the finite difference formulation, the *immersed interface method* [23, 24, 26], the *ghost fluid method* [13, 32], and the *matched interface and boundary method* [44, 45] have been developed. In the FE formulation, elements around the interface have to be treated with special cares. One way is to modify the bilinear form near the interface, such as the *penalty finite element method* [2, 6] and the *unfitted finite element method* [17, 18]. Another approach is to modify those elements cut by the interface in a Cartesian mesh [41]. Modifying basis functions for elements around interface is also investigated, such as the *general finite element method* [3, 4] and the *multi-scale finite element method* [12]. The recently developed *immersed finite element (IFE) methods* [11, 20–22, 25, 27–31, 39, 40] also fall into this framework.

Compared with a conventional FE space, an IFE space has two key features. First, by allowing the mesh to be independent of the interface, an IFE space can be defined on Cartesian meshes for interface problems with a nontrivial geometry without loss of accuracy. Second, instead of universal polynomials in each element of a mesh, an IFE function in each element cut by the interface is a piecewise polynomial of a specified degree constructed according to the interface jump conditions. In particular, this means each IFE function partially solves the interface problem from the point of view of satisfying the interface jump conditions.

Therefore, we can use IFE functions to carry out the discretization in the spatial variables over a fixed structured (Cartesian) mesh for a parabolic PDE whose diffusion coefficient is discontinuous across a moving interface. An immediate benefit of this approach is the avoidance of regenerating meshes through the whole computational procedure, even if the interface changes with respect to time. More importantly, even though the IFE spaces at different time levels are formed according to the location of the interface, the global degrees of freedom as well as their locations in all the IFE spaces used in the whole simulation are maintained the same because the global degrees of freedom are determined by the nodes of the same mesh on which these IFE spaces are constructed. In a recent article [22], we have developed Crank-Nicolson (CN) type IFE schemes for solving the parabolic interface problem (1.1a)-(1.3b). All of these CN-IFE schemes demonstrate $O(\tau^2 + h^2)$ optimal convergence and they are consistent with the standard CN scheme for parabolic problems in the sense that they become the CN scheme if the coefficient

function $\beta(X)$ is continuous or if a body-fitting mesh is used for a problem whose diffusion coefficient is discontinuous across a time independent interface Γ . Our effort here is to develop an IFE-MoL that can work together with a suitably chosen ODE solver to efficiently and reliably solve parabolic moving interface problems over a fixed Cartesian mesh.

The rest of this article is organized as follows. In Section 2, we present some preliminaries of IFEs and introduce new notations to facilitate later discussions. In Section 3, we derive an IFE-MoL based on linear IFEs for the parabolic moving interface problem described by (1.1a)-(1.3b) and discuss its implementation issues. In Section 4, we apply several ODE solvers to the ODE system in the IFE-MoL to generate numerical results that can demonstrate features of this IFE method. Brief conclusions are given in Section 5.

2 Notations and IFE preliminaries

Without loss of generality, we let $\mathcal{T}_h = \{T\}$ be a triangular Cartesian mesh of Ω as illustrated by the plot on the right in Fig. 2. We note that the main ideas in this article can be easily extended to the bilinear IFE space defined on a rectangular Cartesian mesh [19,20]. We call elements whose interiors are cut by the curve $\Gamma(t)$ as interface elements, and we call the rest non-interface elements. Let $\mathcal{T}_h^{i,t}$ and $\mathcal{T}_h^{n,t}$ denote the collections of interface elements and non-interface elements at the time t , respectively. In the discussion from now on, we assume that $\mathcal{T}_h = \mathcal{T}_h^{i,t} \cup \mathcal{T}_h^{n,t}$ does not change with respect to t while $\mathcal{T}_h^{i,t}$ and $\mathcal{T}_h^{n,t}$ may vary according to the interface location.

Define \mathcal{N}_h to be the set of nodes of \mathcal{T}_h . Let \mathcal{N}_h^0 and \mathcal{N}_h^b be the sets of interior nodes and boundary nodes, respectively. Also, we define $\mathcal{N}_h^{i,t}$ to be the set of nodes of all interface elements at time t and let $\mathcal{N}_h^{n,t} = \mathcal{N}_h / \mathcal{N}_h^{i,t}$ denote the set containing the rest of the nodes. Again, since \mathcal{T}_h is time independent, the node set \mathcal{N}_h is also time independent while sets $\mathcal{N}_h^{i,t}$ and $\mathcal{N}_h^{n,t}$ can change with respect to time.

Without loss of generality, we assume, at a given time t , the curve $\Gamma(t)$ intersects the edge of each interface element at no more than two points, and if there are two intersection points, they should be on different edges of that element. Let $T = \triangle A_1 A_2 A_3 \in \mathcal{T}_h^{i,t}$ be an interface element at time t with the intersection points denoted by $D = D(t)$, and $E = E(t)$ such that

$$A_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \quad A_3 = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}, \quad D = \begin{pmatrix} x_D(t) \\ y_D(t) \end{pmatrix}, \quad E = \begin{pmatrix} x_E(t) \\ y_E(t) \end{pmatrix}, \quad (2.1)$$

where the coordinates of D and E depend on t , see the illustration in Fig. 3.

The line segment \overline{DE} , which cuts T into two pieces $T^-(t)$ and $T^+(t)$, is used to approximate the actual interface $\Gamma(t)$ in T . A linear IFE function on an interface element

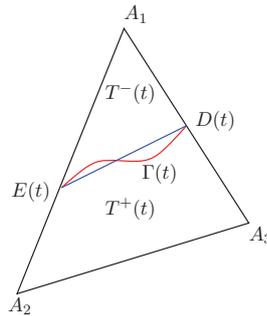


Figure 3: A sketch of local interface element at time t .

such as the one in Fig. 3 can be written in the following form [11, 28],

$$\phi_T^t(x, y) = \begin{cases} \phi_T^{t-}(x, y) = v_1\psi_{1,T} + c_2(t)\psi_{2,T} + c_3(t)\psi_{3,T}, & \text{if } (x, y) \in T^-(t), \\ \phi_T^{t+}(x, y) = c_1(t)\psi_{1,T} + v_2\psi_{2,T} + v_3\psi_{3,T}, & \text{if } (x, y) \in T^+(t). \end{cases} \quad (2.2)$$

Here $\psi_{i,T}$, $i = 1, 2, 3$ are standard linear nodal FE basis functions on T such that

$$\psi_{i,T}(A_j) = \delta_{ij}, \quad 1 \leq i, j \leq 3.$$

In this formulation, v_1, v_2 , and v_3 are nodal values of the IFE function $\phi_T^t(x, y)$ at nodes A_1, A_2 , and A_3 , respectively. The time dependent coefficients $c_1(t), c_2(t)$, and $c_3(t)$ are determined by imposing the interface jump conditions (1.3a) and (1.3b) on $\phi_T^t(x, y)$, see [27, 28], i.e.,

$$\phi_T^{t+}(x_D, y_D) = \phi_T^{t-}(x_D, y_D), \quad \phi_T^{t+}(x_E, y_E) = \phi_T^{t-}(x_E, y_E), \quad (2.3a)$$

$$\beta^+ \nabla \phi_T^{t+}(x, y) \cdot \mathbf{n}_{DE} = \beta^- \nabla \phi_T^{t-}(x, y) \cdot \mathbf{n}_{DE}. \quad (2.3b)$$

It has been shown [28], for each fixed t , the coefficients $c_1(t), c_2(t)$, and $c_3(t)$ are uniquely determined by the nodal values $v_i, i = 1, 2, 3$. To form the nodal IFE basis function $\phi_{i,T}^t$, we let $v_i = 1$, and $v_k = 0$, where $k \neq i$, and solve for $c_j(t), j = 1, 2, 3$ from (2.3a) and (2.3b). A more detailed discussion about constructing these local IFE functions and their derivatives with respect to the time variable t will be presented in Section 3.2. Then, the local FE/IFE space on each element $T \in \mathcal{T}_h$ is defined by

$$S_h^t(T) = \begin{cases} \text{span}\{\psi_{i,T}^t, i = 1, 2, 3\}, & \text{if } T \in \mathcal{T}_h^{n,t}, \\ \text{span}\{\phi_{i,T}^t, i = 1, 2, 3\}, & \text{if } T \in \mathcal{T}_h^{i,t}. \end{cases}$$

Furthermore, the global IFE space on a mesh \mathcal{T}_h can be defined in the following standard form:

$$S_h^t(\Omega) = \text{span}\{\phi_j^t: \phi_j(X_i) = \delta_{ij}, X_i \in \mathcal{N}_h \text{ and } \phi_j^t|_T \in S_h^t(T)\}.$$

We note that, for each interface node $X_j \in \mathcal{N}_h^{i,t}$, the associated global IFE basis function $\phi_j^t(X)$ depends on the interface location, therefore depends on the time t . Otherwise, $\phi_j^t(X)$ is independent of the time t for $X_j \in \mathcal{N}_h^{n,t}$.

3 A method of lines based on immersed finite elements

We note that the superscript t in $S_h^t(T)$ and $S_h^t(\Omega)$ emphasizes that the functions in these spaces actually change with time. Because of the consistency of the IFE functions with the corresponding FE functions [20, 27–29], when the interface moves out of an interface element $T \in \mathcal{T}_h$, the local IFE functions in T become the standard FE functions on T . The consistency further implies that, when the interface moves out of the elements around a node $X_j \in \mathcal{N}_h$, the global IFE basis function ϕ_j^t also become the corresponding standard FE basis function associated with that node. However, the mesh \mathcal{T}_h on which $S_h^t(T)$ and $S_h^t(\Omega)$ are defined does not change with t . When interface changes with t , the global degrees of freedom and their locations associated with $S_h^t(\Omega)$ also remain unchanged. These features enable us to semi-discretize the parabolic moving interface problem (1.1a)-(1.3b).

3.1 IFE method of lines

Since a global IFE basis function $\phi_j^t(X)$ is associated with the fixed node $X_j \in \mathcal{N}_h$, we introduce an unknown function $u_j(t)$ at this node and apply the idea of the MoL to form equations for computing these unknowns $u_j(t), X_j \in \mathcal{N}_h$. Specifically, given a Cartesian mesh \mathcal{T}_h of Ω , we seek an IFE semi-discrete solution to the parabolic moving interface problem (1.1a)-(1.3b) in the following form:

$$u_h(t, X) = \sum_{X_j \in \mathcal{N}_h} u_j(t) \phi_j^t(X). \quad (3.1)$$

Taking the partial derivative with respect to t of the IFE function in (3.1), we have

$$\frac{\partial u_h(t, X)}{\partial t} = \sum_{X_j \in \mathcal{N}_h} \frac{\partial u_j(t)}{\partial t} \phi_j^t(X) + \sum_{X_j \in \mathcal{N}_h^{i,t}} u_j(t) \frac{\partial \phi_j^t(X)}{\partial t}. \quad (3.2)$$

Note that the summation in the second term on the right hand side of (3.2) is only for nodes in $\mathcal{N}_h^{i,t}$ because the time derivative of $\phi_j^t(X)$ is zero if $X_j \notin \mathcal{N}_h^{i,t}$.

Now we turn to the discretization for the moving interface problem starting from the following standard weak form at a given time t :

$$\int_{\Omega} v \frac{\partial u}{\partial t} dX + \int_{\Omega} \nabla v \cdot (\beta \nabla u) dX = \int_{\Omega} v f dX, \quad \forall v \in H_0^1(\Omega), \quad (3.3)$$

which is equivalent to

$$\sum_{T \in \mathcal{T}_h} \int_T v \frac{\partial u}{\partial t} dX + \sum_{T \in \mathcal{T}_h} \int_T \nabla v \cdot (\beta \nabla u) dX = \int_{\Omega} v f dX, \quad \forall v \in H_0^1(\Omega). \quad (3.4)$$

Consequently, this weak form leads to the following spatial discretization: Find $u_h \in S_h^t(\Omega)$, such that

$$\sum_{T \in \mathcal{T}_h} \int_T v_h \frac{\partial u_h}{\partial t} dX + \sum_{T \in \mathcal{T}_h} \int_T \nabla v_h \cdot (\beta \nabla u_h) dX = \int_{\Omega} v_h f dX, \quad \forall v_h \in S_{h,0}^t(\Omega), \quad (3.5)$$

where $S_{h,0}^t(\Omega) = \text{span}\{\phi_j^t \in S_h^t; X_j \in \mathcal{N}_h^0\}$. Plugging (3.1) and (3.2) into (3.5), and substituting $\phi_i^t \in S_{h,0}^t$ for v_h , the above semi-discretization becomes: Find the coefficient functions $u_j(t)$ in $u_h(t, X) = \sum_{X_j \in \mathcal{N}_h} u_j(t) \phi_j^t(X)$ such that

$$\begin{aligned} & \sum_{X_j \in \mathcal{N}_h} u_j'(t) \int_{\Omega} \phi_i^t \phi_j^t dX + \sum_{X_j \in \mathcal{N}_{h,t}^i} u_j(t) \int_{\Omega} \phi_i^t \left(\frac{\partial}{\partial t} \phi_j^t \right) dX \\ & + \sum_{X_j \in \mathcal{N}_h} u_j(t) \int_{\Omega} \beta \nabla \phi_i^t \cdot \nabla \phi_j^t dX = \int_{\Omega} f \phi_i^t dX, \quad \forall \phi_i^t \in S_{h,0}^t. \end{aligned} \quad (3.6)$$

Imposing the boundary condition to the IFE function $u_h(t, X)$ in this semi-discrete scheme leads to

$$\begin{aligned} & \sum_{X_j \in \mathcal{N}_h^0} u_j'(t) \int_{\Omega} \phi_i^t \phi_j^t dX + \sum_{X_j \in \mathcal{N}_{h,t}^{i,0}} u_j(t) \int_{\Omega} \phi_i^t \left(\frac{\partial}{\partial t} \phi_j^t \right) dX + \sum_{X_j \in \mathcal{N}_h^0} u_j(t) \int_{\Omega} \beta \nabla \phi_i^t \cdot \nabla \phi_j^t dX \\ & = \int_{\Omega} f \phi_i^t dX - \sum_{X_j \in \mathcal{N}_h^b} g_j'(t) \int_{\Omega} \phi_i^t \phi_j^t dX - \sum_{X_j \in \mathcal{N}_{h,t}^{i,b}} g_j(t) \int_{\Omega} \phi_i^t \left(\frac{\partial}{\partial t} \phi_j^t \right) dX \\ & - \sum_{X_j \in \mathcal{N}_h^b} g_j(t) \int_{\Omega} \beta \nabla \phi_i^t \cdot \nabla \phi_j^t dX, \quad \forall \phi_i^t \in S_{h,0}^t(\Omega), \end{aligned} \quad (3.7)$$

where $g_j(t) = g(t, X_j)$, for $X_j \in \mathcal{N}_h^b$. We can write (3.7) in the equivalent matrix form as follows

$$M(t) \mathbf{u}'(t) + (K(t) + A(t)) \mathbf{u}(t) = \mathbf{f}(t) - \mathbf{bc}(t), \quad (3.8)$$

with the initial condition

$$\mathbf{u}(0) = \mathbf{u}_0, \quad (3.9)$$

where

- $M(t) = (m_{ij}(t))$ is the mass matrix with $m_{ij} = \int_{\Omega} \phi_i^t \phi_j^t dX$.
- $K(t) = (k_{ij}(t))$ with $k_{ij} = \int_{\Omega} \phi_i^t (\partial \phi_j^t / \partial t) dX$.
- $A(t) = (a_{ij}(t))$ is the stiffness matrix with $a_{ij} = \int_{\Omega} \nabla \phi_i^t \cdot (\beta \nabla \phi_j^t) dX$.
- $\mathbf{f}(t) = (f_i(t))$ is the source term vector with $f_i(t) = \int_{\Omega} f \phi_i^t dX$.
- $\mathbf{u}(t) = (u_j(t))$, $\mathbf{u}'(t) = (u_j'(t))$, and $\mathbf{u}_0 = (u_0(X_j))$ with $X_j \in \mathcal{N}_h^0$.

- $\mathbf{bc}(t)$ is the boundary vector associated with the last three terms in (3.7).

We call Eqs. (3.8) and (3.9) an IFE-MoL for solving the parabolic moving interface problems.

Remark 3.1. Compared with the traditional semi-discrete FE method for the initial boundary value problems of parabolic equations, the IFE-MoL (3.8) contains an extra term involving matrix $K(t)$ that depends on the time derivative of IFE basis functions due to the moving interface. This method is consistent with the standard FE-MoL in the sense that the matrix $K(t)$ is a zero matrix and this method becomes the standard MoL if $\beta(X)$ is continuous, or if the interface is static and a body-fitting mesh is used.

3.2 Implementation of the IFE method of lines

In this subsection, we will discuss some implementation issues for the IFE-MoL for solving parabolic moving interface problems.

At every time t , the process of assembling global matrices from local matrices follows the standard procedure for traditional FE computations. A standard FE assembler can be employed to form local matrices over all non-interface elements; hence, our focus here is the process of generating local matrices on interface elements. The main ideas and more details can be found in [22].

Local matrices of $M(t)$ and $A(t)$

Assembling local mass and stiffness matrices, i.e., quantities to form $M(t)$ and $A(t)$, follows the same procedure as those for the IFE methods for time independent interface problems. The only difference is to update the interface location for a given value of t .

Local matrices of $K(t)$

For the matrix $K(t)$ in (3.8), we note that each of its entries involves the inner product of an IFE basis function and its time derivative function, i.e.,

$$k_{ij}(t) = \int_{\Omega} \phi_i^t(X) \left(\frac{\partial}{\partial t} \phi_j^t(X) \right) dX.$$

Hence, constructing $k_{ij}(t)$ needs the time derivative $\partial \phi_j^t / \partial t$ of the nodal IFE basis function ϕ_j^t . As usual, we only need to derive the time derivative of the local nodal IFE basis functions in interface elements.

Without loss of generality, and in order to simplify the notations, we focus on the derivation of time derivatives of local linear IFE nodal basis functions $\phi_{i,T}^t$, $i = 1, 2, 3$, on the following triangular interface element with vertices

$$A_1 = (x_1, y_1) = (0, 0), \quad A_2 = (x_2, y_2) = (h, 0), \quad A_3 = (x_3, y_3) = (0, h).$$

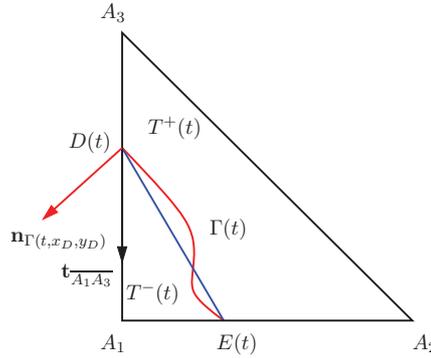


Figure 4: A sketch of the interface configuration in an element at time t .

Assume the intersection points $D(t)$, and $E(t)$ are on $\overline{A_1A_3}$, and $\overline{A_1A_2}$, respectively, as illustrated in Fig. 4. We can write coordinates of $D(t)$ and $E(t)$ in terms of time dependent ratios $d(t)$, and $e(t)$ as follows,

$$x_D(t) = x_1 + d(t)(x_3 - x_1), \quad y_D(t) = y_1 + d(t)(y_3 - y_1), \quad (3.10a)$$

$$x_E(t) = x_1 + e(t)(x_2 - x_1), \quad y_E(t) = y_1 + e(t)(y_2 - y_1), \quad (3.10b)$$

where $0 \leq d(t) \leq 1, 0 \leq e(t) \leq 1$.

As described in Section 2, local linear IFE nodal basis functions $\phi_{i,T}^t, i = 1, 2, 3$, can be obtained by imposing the interface jump conditions (2.3a) and (2.3b) to (2.2). This leads to the following linear system for coefficients $c_1(t), c_2(t), c_3(t)$:

$$\begin{pmatrix} 1-d & 0 & -d \\ 1-e & -e & 0 \\ \beta^+(d+e) & \beta^-d & \beta^-e \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1-d & 0 & -d \\ 1-e & -e & 0 \\ \beta^-(d+e) & \beta^+d & \beta^+e \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad (3.11)$$

where $d=d(t)$, and $e=e(t)$. To obtain each nodal IFE basis function, we choose appropriate nodal values, for instance we choose $(v_1, v_2, v_3) = (0, 1, 0)$ for the basis $\phi_{2,T}^t$, and solve for corresponding coefficients $c_i(t), i = 1, 2, 3$. Then, we put these values $c_i(t), i = 1, 2, 3$, back in (2.2) to obtain the corresponding nodal IFE basis function $\phi_{i,T}^t$.

Using (2.2) for the nodal IFE basis function $\phi_{i,T}^t$, we can calculate their time derivative as follows

$$\frac{\partial}{\partial t} \phi_{i,T}^t(x, y) = \begin{cases} \frac{\partial}{\partial t} \phi_{i,T}^{t-}(x, y) = c'_2(t)\psi_{2,T} + c'_3(t)\psi_{3,T}, & \text{if } (x, y) \in T^-(t), \\ \frac{\partial}{\partial t} \phi_{i,T}^{t+}(x, y) = c'_1(t)\psi_{1,T}, & \text{if } (x, y) \in T^+(t). \end{cases} \quad (3.12)$$

Moreover, derivatives $c'_i(t), i = 1, 2, 3$, can be calculated from the following linear system

obtained by taking the derivative on both sides of (3.11),

$$\begin{aligned} & \begin{pmatrix} 1-d & 0 & -d \\ 1-e & -e & 0 \\ \beta^+(d+e) & \beta^-d & \beta^-e \end{pmatrix} \begin{pmatrix} c'_1 \\ c'_2 \\ c'_3 \end{pmatrix} \\ &= \begin{pmatrix} -d' & 0 & -d' \\ -e' & -e' & 0 \\ \beta^-(d'+e') & \beta^+d' & \beta^+e' \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} - \begin{pmatrix} -d' & 0 & -d' \\ -e' & -e' & 0 \\ \beta^+(d'+e') & \beta^-d' & \beta^-e' \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}. \end{aligned} \quad (3.13)$$

Note that the coefficient matrix of $c'_i(t)$, $i=1,2,3$, in (3.13) is the same as the one of $c_i(t)$, $i=1,2,3$, in (3.11). Hence, the unisolvent property for IFE nodal basis [28] guarantees that $c'_i(t)$, $i=1,2,3$ can be uniquely determined as long as $d'(t)$ and $e'(t)$ exist.

The remaining task is to find $d'(t)$ and $e'(t)$. Let us assume that the moving interface $\Gamma(t)$ is described by the equation $\Gamma(t, x, y) = 0$. Hence, we have

$$\Gamma(t, x_D(t), y_D(t)) = 0, \quad \Gamma(t, x_E(t), y_E(t)) = 0. \quad (3.14)$$

Taking the derivative with respect to t on both sides of these equations leads to equations about $d'(t)$ and $e'(t)$. Then, solving these equations for $d'(t)$ and $e'(t)$ leads to

$$d'(t) = \frac{-\Gamma_t(t, x_D, y_D)}{\Gamma_x(t, x_D, y_D)(x_3 - x_1) + \Gamma_y(t, x_D, y_D)(y_3 - y_1)}, \quad (3.15a)$$

$$e'(t) = \frac{-\Gamma_t(t, x_E, y_E)}{\Gamma_x(t, x_E, y_E)(x_2 - x_1) + \Gamma_y(t, x_E, y_E)(y_2 - y_1)}. \quad (3.15b)$$

More details about the derivation of these formulas can be found in [22].

Remark 3.2. The matrix $K(t)$ is much sparser than the mass matrix $M(t)$ and the stiffness matrix $A(t)$, because only those IFE basis functions associated with interface nodes in $\mathcal{N}_h^{i,t}$ have non-zero time derivatives. When the mesh is fine enough, the majority of nodes are non-interface nodes which belong to $\mathcal{N}_h^{n,t}$. Consequently, it costs little time to assemble the matrix $K(t)$.

Remark 3.3. The procedures developed in this section can be easily extended to assembling matrices for the IFE-MoL with bilinear IFE functions [20] on a rectangular Cartesian mesh.

4 Numerical experiments

In this section, we present numerical examples to demonstrate features of the IFE-MoL for parabolic moving interface problems.

We consider the same example used in [22] for the moving interface problem (1.1a)-(1.3b). The solution domain is $\Omega \times [0, 1]$, where $\Omega = (-1, 1) \times (-1, 1)$ and the interface

$\Gamma(t)$ is a moving circle centered at $(0,0)$ with a radius $r(t)$ that separates Ω into two sub-domains $\Omega^-(t) = \{(x,y) \in \Omega: x^2 + y^2 < r(t)^2\}$ and $\Omega^+(t) = \{(x,y) \in \Omega: x^2 + y^2 > r(t)^2\}$. The exact solution is chosen as:

$$u(t,x,y) = \begin{cases} \frac{1}{\beta^-}(x^2+y^2)^{5/2}\cos(t), & (x,y) \in \Omega^-(t), \\ \frac{1}{\beta^+}(x^2+y^2)^{5/2}\cos(t) + \left(\frac{1}{\beta^-} - \frac{1}{\beta^+}\right)r(t)^5\cos(t), & (x,y) \in \Omega^+(t). \end{cases} \quad (4.1)$$

We use triangular Cartesian meshes \mathcal{T}_h which are formed by partitioning Ω with $N_s \times N_s$ rectangles of size $h = 2/N_s$ and then cutting each rectangle into two triangles along one of its diagonal line, see right plot in Fig. 2 for an illustration.

The IFE-MoL (3.8) and (3.9) can be written in following standard ODE form for $\mathbf{u}(t)$

$$\mathbf{u}'(t) = \mathbf{F}(t, \mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0, \quad (4.2)$$

where $\mathbf{u}_0 = (u_0(X_j))$, with $X_j \in \mathcal{N}_h^0$, and

$$\mathbf{F}(t, \mathbf{u}) = M^{-1}(t) \left(-(K(t) + A(t))\mathbf{u}(t) + \mathbf{f}(t) - \mathbf{bc}(t) \right). \quad (4.3)$$

A preferred ODE solver can then be used to solve this ODE system in the IFE-MoL.

Single step methods

Implicit Runge-Kutta (IRK) methods are good candidates for the IFE-MoL because they are often A-stable and some of them work effectively for stiff problems. A general s -stage IRK method can be described conveniently in the following *Butcher diagram* [8]:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \quad (4.4)$$

However, when a high order multistage fully implicit Runge-Kutta method is used, computing the stage values, denoted by $\mathbf{K}_i, i = 1, \dots, s$, is usually a big hurdle. This is because we have to solve for these vectors from an $(sdim(\mathbf{u})) \times (sdim(\mathbf{u}))$ block linear system whose dimension is very high when a fine mesh is used and the band structure is more complicated than that of each block. One possible way to alleviate this difficulty is to use a so-called Diagonally Implicit Runge-Kutta (DIRK) method [16] for which the coefficient matrix $(a_{ij})_{i,j=1}^s$ in (4.4) is a lower triangular matrix. In a DIRK method, $\mathbf{K}_i, i = 1, \dots, s$ are determined by s decoupled linear systems, each of them is of the size $dim(\mathbf{u}) \times dim(\mathbf{u})$, and they all have the same band structure. Specifically, the s -stage DIRK scheme for solving (4.2) can be described as follows: Given \mathbf{u}^n , and τ , we find \mathbf{u}^{n+1} by

1. Compute \mathbf{K}_1 by solving

$$\left(M^{n+c_1} + a_{11}\tau(K^{n+c_1} + A^{n+c_1})\right)\mathbf{K}_1 = -(K^{n+c_1} + A^{n+c_1})\mathbf{u}^n + \mathbf{f}^{n+c_1} - \mathbf{bc}^{n+c_1}. \quad (4.5)$$

2. Compute \mathbf{K}_i , $i=2, \dots, s$, by solving

$$\begin{aligned} & \left(M^{n+c_i} + a_{ii}\tau(K^{n+c_i} + A^{n+c_i})\right)\mathbf{K}_i \\ &= -(K^{n+c_i} + A^{n+c_i})\left(\mathbf{u}^n + \tau \sum_{j=1}^{i-1} a_{ij}\mathbf{K}_j\right) + \mathbf{f}^{n+c_i} - \mathbf{bc}^{n+c_i}. \end{aligned} \quad (4.6)$$

3. Compute \mathbf{u}^{n+1}

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \tau \sum_{i=1}^s b_i \mathbf{K}_i. \quad (4.7)$$

Here

$$A^{n+c_i} = A(t_n + c_i\tau), \quad K^{n+c_i} = K(t_n + c_i\tau), \quad M^{n+c_i} = M(t_n + c_i\tau), \quad 1 \leq i \leq s,$$

with the matrices $A(t)$, $K(t)$ and $M(t)$ defined in Section 3. The same convention applies to the involved vectors.

Multi-step methods

Compared to the DIRK methods, linear multi-step methods usually require less function evaluations per time step. The family of *Adams Methods* are popular for non-stiff problems, and *Backward Difference Formula* (BDF) methods are effective for stiff systems [1]. Since the ODE system in a MoL for an initial boundary problem of a time dependent PDE is usually stiff, BDF methods are usually preferable. A k -step BDF method [1] can be written as

$$\sum_{i=0}^k \alpha_i \mathbf{u}^{n+1-i} = h\beta_0 \mathbf{F}^{n+1}, \quad (4.8)$$

where $\mathbf{F}^{n+1} = \mathbf{F}(t_{n+1}, \mathbf{u}_{n+1})$. The k -step BDF scheme for solving (4.2) are described as follows: Given \mathbf{u}^{n-k+1} , \mathbf{u}^{n-k+2} , \dots , \mathbf{u}^n and τ , we find \mathbf{u}^{n+1} by solving

$$\left(\alpha_0 M^{n+1} + \tau\beta_0(A^{n+1} + K^{n+1})\right)\mathbf{u}^{n+1} = \tau\beta_0(\mathbf{f}^{n+1} - \mathbf{bc}^{n+1}) - M^{n+1} \sum_{i=1}^k \alpha_i \mathbf{u}^{n+1-i}, \quad (4.9)$$

where

$$A^{n+1} = A(t_{n+1}), \quad K^{n+1} = K(t_{n+1}), \quad M^{n+1} = M(t_{n+1}),$$

and the same convention applies to vectors.

Comparison of single step and multi-step methods

A single step method approximates \mathbf{u}^{n+1} by taking into account only the behavior of $\mathbf{u}(t)$ between t_n and t_{n+1} , while a multi-step method require information from a number of previous time steps. This means a single step method needs nothing except \mathbf{u}_0 to start up the iteration in time. On the other hand, to start a multi-step method, k initial values $\mathbf{u}_0, \dots, \mathbf{u}_{k-1}$ are needed. Usually, an appropriate single step method can be used to generate the rest of the initial values $\mathbf{u}_1, \dots, \mathbf{u}_{k-1}$.

To achieve a comparable high order accuracy, a multi-step method usually requires less matrices assembling and less linear system solving at each time step than a DIRK method. At each time level, the BDF method (4.9) needs to generate $2+\epsilon$ matrices, which are $M(t)$, $A(t)$ and $K(t)$, and solve only one linear system. Here ϵ emphasizes the fact that assembling $K(t)$ costs significantly much less time than $M(t)$, $A(t)$, see Remark 3.2. On the other hand, an s -stage DIRK method needs to assemble $s(2+\epsilon)$ matrices and solve s linear systems at each time step. We also note that single step methods are convenient for the implementation of adaptivity in the time step size which is usually preferred for producing a reliable solution to a complicated ODE system.

Example 4.1. (Second Order ODE Solvers).

We assume the radius of the interface circle is governed by the function

$$r(t) = r_0 \left(\frac{\sin(t) + 3}{4} \right)$$

with $r_0 = \pi/6.28$ in Examples 4.1 and 4.2. The following second order DIRK scheme [1] is used to solve the ODE system in the IFE-MoL:

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & \gamma & 1-\gamma \\ \hline & \gamma & 1-\gamma \end{array}, \quad (4.10)$$

where $\gamma = (2 - \sqrt{2})/2$. Numerical experiments are carried out for both small coefficient jump $(\beta^-, \beta^+) = (1, 2)$ and large coefficient jump $(\beta^-, \beta^+) = (1, 100)$, and in both cases, we choose $\tau = h$. Errors in numerical solutions generated by the IFE-MoL are computed at the final time level $t = 1$ in both L^2 and semi- H^1 norms and they are presented in Table 1. Applying linear regression on these data we can see that the IFE solutions obey the following error estimates:

- **DIRK2 (Small Jump)**

$$\|u_h^n - u(t_n, \cdot)\|_{L^2} \approx 0.9686h^{1.9963}, \quad |u_h^n - u(t_n, \cdot)|_{H^1} \approx 2.9195h^{0.9994},$$

- **DIRK2 (Large Jump)**

$$\|u_h^n - u(t_n, \cdot)\|_{L^2} \approx 0.0412h^{1.8696}, \quad |u_h^n - u(t_n, \cdot)|_{H^1} \approx 0.1219h^{0.9109},$$

Table 1: Errors of 2D linear IFE solution with $\beta^- = 1$ using DIRK2 at time $t = 1$.

h	τ	$\beta^+ = 2$		$\beta^+ = 100$	
		$\ \cdot\ _{L^2}$	$\ \cdot\ _{H^1}$	$\ \cdot\ _{L^2}$	$\ \cdot\ _{H^1}$
1/16	1/16	3.8203E-2	1.8269E-1	2.4420E-4	9.6691E-3
1/32	1/32	9.5852E-4	9.1447E-2	6.0966E-5	5.2348E-3
1/64	1/64	2.4018E-4	4.5738E-2	1.6860E-5	2.7664E-3
1/128	1/128	6.0174E-5	2.2873E-2	4.4807E-6	1.4708E-3
1/256	1/256	1.5081E-5	1.1438E-2	1.3824E-6	7.7637E-4

which indicate the expected error bound:

$$\|u_h^n - u(t_n, \cdot)\|_{k, \Omega} \leq C(h^{2-k} + \tau^2), \quad k=0,1.$$

Example 4.2. (Higher Order ODE Solvers).

One of the motivations to use a MoL for solving time dependent PDEs is the easy employment of higher order schemes for solving the related ODE system. To demonstrate this feature, we present numerical results generated by a higher order single step method and a higher order multi-step method. For the single step method, we use the following fourth order DIRK scheme [16]:

$$\begin{array}{c|ccc}
 \frac{1}{4} & \frac{1}{4} & & \\
 \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & \\
 \frac{11}{20} & \frac{17}{20} & -\frac{1}{25} & \frac{1}{4} \\
 \frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} \\
 1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\
 \hline
 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4}
 \end{array} \tag{4.11}$$

The multi-step method is the following fourth order BDF scheme [1]:

$$\mathbf{u}^{n+1} = \frac{1}{25} (48\mathbf{u}^n - 36\mathbf{u}^{n-1} + 16\mathbf{u}^{n-2} - 3\mathbf{u}^{n-3} + 12\tau\mathbf{F}^{n+1}).$$

Exact initial values $\mathbf{u}^i = (u(t_i, X_j))$, $i=0,1,2,3, X_j \in \mathcal{N}_h^0$ are used to start the time iteration. Errors in IFE solutions generated by both schemes at the final time level $t=1$ are listed in Table 2. Since both schemes are fourth order accurate in time steps, we expect the errors to obey

$$\|u_h^n - u(t_n, \cdot)\|_{k, \Omega} \leq C(h^{2-k} + \tau^4), \quad k=0,1.$$

Therefore, to observe the convergence rate in term of h , we use $h = 8\tau^2$ to make h^2 proportional to τ^4 for the chosen mesh sizes. By linear regression we can see that the data in Table 2 have the following estimates:

Table 2: Errors of 2D linear IFE solution with $\beta^- = 1, \beta^+ = 2$ using 4th order schemes at time $t = 1$.

h	τ	DIRK4		BDF4	
		$\ \cdot\ _{L^2}$	$ \cdot _{H^1}$	$\ \cdot\ _{L^2}$	$ \cdot _{H^1}$
1/8	1/8	1.5087E-2	3.6392E-1	1.5143E-2	3.6392E-1
1/32	1/16	9.3618E-4	9.1447E-2	9.5054E-4	9.1447E-2
1/128	1/32	5.6341E-5	2.2873E-2	5.9474E-5	2.2872E-2
1/512	1/64	3.1699E-6	5.7210E-3	3.7318E-6	5.7209E-3

• **DIRK4**

$$\|u_h^n - u(t_n, \cdot)\|_{L^2} \approx 1.0627h^{2.0352}, \quad |u_h^n - u(t_n, \cdot)|_{H^1} \approx 2.9072h^{0.9987},$$

• **BDF4**

$$\|u_h^n - u(t_n, \cdot)\|_{L^2} \approx 0.9653h^{1.9979}, \quad |u_h^n - u(t_n, \cdot)|_{H^1} \approx 2.9073h^{0.9987},$$

which demonstrate the optimal rates of convergence in both L^2 and semi H^1 norms for the IFE-MoL combined with these higher order ODE solvers.

Example 4.3. (Adaptive ODE Solver).

An advantage to use a MoL is the availability of reliable and efficient adaptive ODE solvers that can automatically adjust the time step size according to the rate of change of the exact solution with respect to t so that the local error can be maintained within a prescribed amount. This adaptivity is particular desirable when one needs to solve a moving interface problem in which the interface changes with respect to the time in a complicated way.

To see the performance of the IFE-MoL combined with an adaptive ODE solver, we consider the moving interface problem described at the beginning of this section in which a moving circular interface has the radius governed by

$$r(t) = \frac{1}{400} \exp\left(\frac{1}{5(0.6-t)^2 + 0.25}\right) + \frac{1}{300} \exp\left(\frac{1}{(1.1-t)^2 + 0.19}\right) + 0.25.$$

It is easy to see that this interface changes with respect to t at a varying rate, as illustrated in the left plot in Fig. 5. The adaptive ODE solver used in our numerical experiments for this problem is the popular embedded DIRK45 scheme [16] described by the following Butcher diagram:

$$\begin{array}{c|cccc}
 \frac{1}{4} & \frac{1}{4} & & & \\
 \frac{3}{4} & \frac{1}{4} & \frac{1}{4} & & \\
 \frac{11}{20} & \frac{17}{20} & -\frac{1}{25} & \frac{1}{4} & \\
 \frac{1}{2} & \frac{371}{1360} & -\frac{137}{2720} & \frac{15}{544} & \frac{1}{4} \\
 1 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\
 \hline
 & \frac{25}{24} & -\frac{49}{48} & \frac{125}{16} & -\frac{85}{12} & \frac{1}{4} \\
 & \frac{59}{48} & -\frac{17}{96} & \frac{225}{32} & -\frac{85}{12} & 0
 \end{array} \tag{4.12}$$

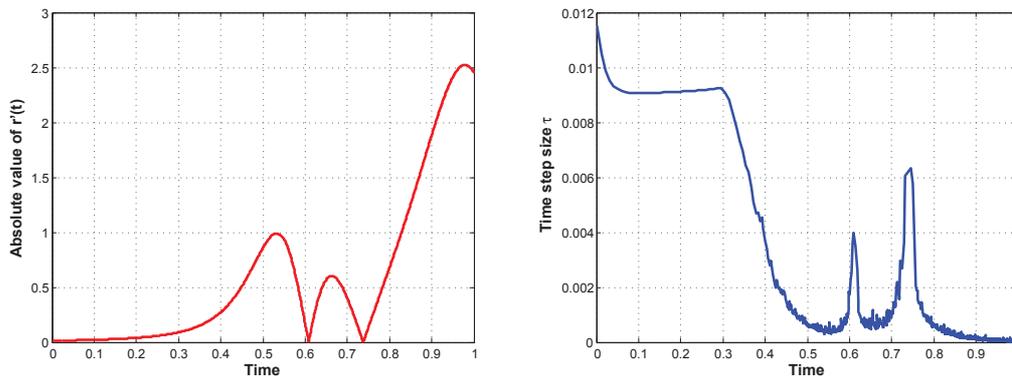


Figure 5: The left plot shows how the radius $r(t)$ of the interface circle $\Gamma(t)$ changes; the plot on the right is for the time step sizes used by the IFE-MoL combined with the adaptive DIRK45 ODE solver.

When we use this DIRK45 scheme to solve the ODE system in the IFE-MoL for this moving interface problem, we set its local tolerance as $tol = h^2$, and choose the maximum time step size $\tau_{\max} = 5h$. The initial step size is set as $\tau_0 = h$.

The right plot in Fig. 5 displays a set of time step sizes automatically determined by the IFE-MoL combined with the adaptive DIRK45 scheme in a computation for solving this moving interface problem. Comparing this plot with the curve of $|\alpha'(t)|$ on the left, we can see that this adaptive IFE-MoL can handle the change in the interface with respect to time very well. The method uses relatively larger time step sizes for $t < 0.3$ where $|\alpha'(t)|$ is small, i.e., the interface location $\alpha(t)$ changes slowly. The time step sizes used by this method decrease in $0.3 < t < 0.55$ since the interface change more rapidly within this time interval. The curve of the time step sizes has two peaks around $t = 0.6$ and $t = 0.75$ where the interface changes at smaller rates; hence larger time steps are allowed. The step sizes become smaller and smaller after $t > 0.8$ due to a faster change of the interface location. All these observations agree with our expectation according to the behavior of interface movement.

Moreover, the adaptive IFE-MoL can produce accurate solutions to moving interface problems by automatic adjustment of time step size according a prescribed error tolerance. To see this, we present some of our numerical results in Table 3 in which errors of IFE solutions at the final time $t = 1$ in both L^2 norm and semi- H^1 norm are listed. The number N in this table denotes the total number of iterations used in each computation.

Table 3: Errors of 2D linear IFE adaptive DIRK45 solutions with $\beta^- = 0.5$, $\beta^+ = 2$.

h	N	$\ \cdot\ _{L^2}$	$\ \cdot\ _{H^1}$
1/8	14	1.7976E-2	4.2420E-1
1/16	49	4.4131E-3	2.1544E-1
1/32	182	1.1625E-3	1.0981E-1
1/64	667	2.9050E-4	5.5990E-2
1/128	2369	7.6317E-5	2.8762E-2

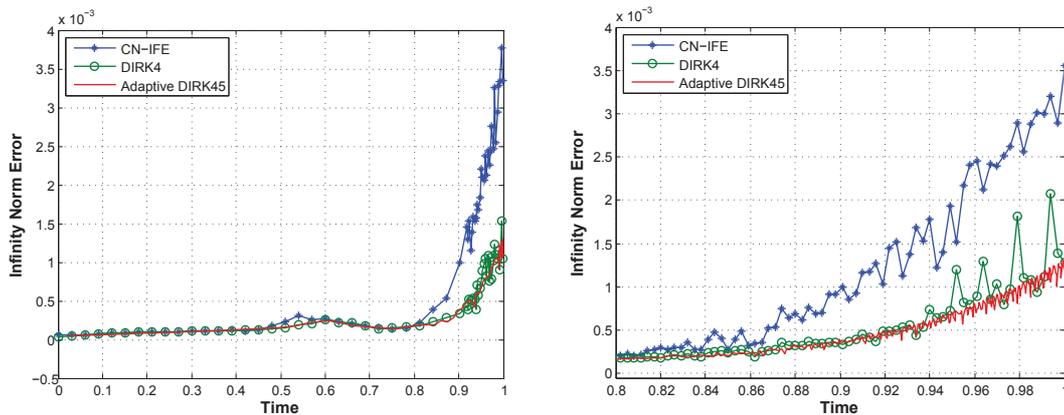


Figure 6: The plot on the left contains curves of L^∞ error for three IFE solutions generated on the same mesh with $h=1/64$. The plot on the right is the enlarged part for time between 0.8 to 1.

By linear regression we can see that these errors obey

- Adaptive DIRK45

$$\|u_h^n - u(t_n, \cdot)\|_{L^2} \approx 1.0592h^{1.9685}, \quad |u_h^n - u(t_n, \cdot)|_{H^1} \approx 3.1846h^{0.9709},$$

which suggest the optimal convergence of the IFE-MoL. Then, we compare the IFE solution generated by the IFE-MoL combined with the adaptive DIRK45 ODE solver on a Cartesian mesh of $h=1/64$ with other IFE solutions produced by methods with a fixed time step size. In the computation to generate this IFE solution, the DIRK45 ODE solver automatically carries out 667 iterations in time. Then, we generate two additional IFE solutions by the IFE-MoL combined with the ODE solver DIRK4 and the Crank-Nicolson Algorithm 1 proposed in [22] on the same mesh, respectively, and we use 667 equally spaced time steps in both of these two methods. The L^∞ norm errors in these three IFE solutions are compared in Fig. 6, from which we can see that the adaptive IFE-MoL has a better control on the error in its solution while errors in those IFE solutions based on a uniform time step size grow faster along with the time. These numerical results indicate that the adaptive IFE-MoL can produce more reliable numerical solutions than methods with a fixed time step size.

5 Conclusions

In this article we develop an IFE-MoL for parabolic moving interface problems. We use immersed finite elements for spatial discretization on a fixed mesh. This IFE method is easy to implement using an existing FE/IFE package developed for static interface problems. Abundant availability of ODE solvers allows us to employ this IFE-MoL to reliably and efficiently solve moving interface problems.

Acknowledgments

This work is partially supported by NSF grant DMS-1016313, GRF grant of Hong Kong (Project No. PolyU 501709), AMA-JRI of PolyU, Polyu grant No. 5020/10P and NSERC (Canada).

References

- [1] U. ASCHER AND L. PETZOLD, *Computer methods for ordinary differential equations and differential-algebraic equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.
- [2] I. BABUŠKA, *The finite element method for elliptic equations with discontinuous coefficients*, *Computing*, 5 (1970), pp. 207–213.
- [3] I. BABUŠKA, C. CALOZ, AND J. OSBORN, *Special finite element methods for a class of second order elliptic problems with rough coefficients*, *SIAM J. Numer. Anal.*, 31(4) (1994), pp. 945–981.
- [4] I. BABUŠKA AND J. OSBORN, *Generalized finite element methods: their performance and their relation to mixed methods*, *SIAM J. Numer. Anal.*, 20(3) (1983), pp. 510–536.
- [5] I. BABUŠKA AND J. OSBORN, *Can a finite element method perform arbitrarily badly?*, *Math. Comput.*, 69(230) (2000), pp. 443–462.
- [6] J. BARRETT AND C. ELLIOTT, *Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces*, *IMA J. Numer. Anal.*, 7(3) (1987), pp. 283–300.
- [7] J. BRAMBLE AND J. KING, *A finite element method for interface problems in domains with smooth boundaries and interfaces*, *Adv. Comput. Math.*, 6(2) (1996), pp. 109–138.
- [8] J. BUTCHER, *Numerical methods for ordinary differential equations*, John Wiley & Sons, Ltd., Chichester, second edition, 2008.
- [9] J. CANNON, J. DOUGLAS, AND C. HILL, *A multi-boundary Stefan problem and the disappearance of phases*, *J. Math. Mech.*, 17 (1967), pp. 21–33.
- [10] Z. CHEN AND J. ZOU, *Finite element methods and their convergence for elliptic and parabolic interface problems*, *Numer. Math.*, 79(2) (1998), pp. 175–202.
- [11] S. CHOU, D. KWAK, AND K. WEE, *Optimal convergence analysis of an immersed interface finite element method*, *Adv. Comput. Math.*, 33(2) (2010), pp. 149–168.
- [12] C. CHU, I. GRAHAM, AND T. HOU, *A new multiscale finite element method for high-contrast elliptic interface problems*, *Math. Comput.*, 79(272) (2010), pp. 1915–1955.
- [13] R. FEDKIW, T. ASLAM, B. MERRIMAN, AND S. OSHER, *A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method)*, *J. Comput. Phys.*, 152(2) (1999), pp. 457–492.
- [14] X. FENG, Z. LI AND L. WANG, *Analysis and numerical methods for some crack problems*, *Int. J. Numer. Anal. Model. Series B*, 2(2-3) (2011), pp. 155–166.
- [15] B. GILDING, *Qualitative mathematical analysis of the Richards equation*, *Transport Porous Med.*, 6(5-6) (1991), pp. 651–666.
- [16] E. HAIRER AND G. WANNER, *Solving ordinary differential equations*, volume II of Springer Series in Computational Mathematics, 14, Springer-Verlag, Berlin, 1996.
- [17] A. HANSBO AND P. HANSBO, *An unfitted finite element method, based on Nitsche's method, for elliptic interface problems*, *Comput. Methods Appl. Mech. Eng.*, 191 (2002), pp. 5537–5552.

- [18] A. HANSBO AND P. HANSBO, *A finite element method for the simulation of strong and weak discontinuities in solid mechanics*, Comput. Methods Appl. Mech. Eng., 193 (2004), pp. 3523–3540.
- [19] X. HE, *Bilinear Immersed Finite Elements For Interface Problems*, PhD thesis, Virginia Tech, 2009.
- [20] X. HE, T. LIN, AND Y. LIN, *Approximation capability of a bilinear immersed finite element space*, Numer. Methods Partial Differential Equations, 24(5) (2008), pp. 1265–1300.
- [21] X. HE, T. LIN, AND Y. LIN, *Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions*, Int. J. Numer. Anal. Model., 8(2) (2011), pp. 284–301.
- [22] X. HE, T. LIN, Y. LIN, AND X. ZHANG, *Immersed finite element methods for parabolic equations with moving interface*, Numer. Methods Partial Differential Equations, 29(2) (2013), 619–646.
- [23] R. LEVEQUE AND Z. LI, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal., 31(4) (1994), pp. 1019–1044.
- [24] Z. LI, *Immersed interface methods for moving interface problems*, Numer. Algorithms, 14(4) (1997), pp. 269–293.
- [25] Z. LI, *The immersed interface method using a finite element formulation*, Appl. Numer. Math., 27(3) (1998), pp. 253–267.
- [26] Z. LI AND K. ITO, *The immersed interface method. Numerical solutions of PDEs involving interfaces and irregular domains*, Frontiers in Applied Mathematics, 33, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.
- [27] Z. LI, T. LIN, Y. LIN, AND R. ROGERS, *An immersed finite element space and its approximation capability*, Numer. Methods Partial Differential Equations, 20(3) (2004), pp. 338–367.
- [28] Z. LI, T. LIN, AND X. WU, *New Cartesian grid methods for interface problems using the finite element formulation*, Numer. Math., 96(1) (2003), pp. 61–98.
- [29] T. LIN, Y. LIN, R. ROGERS, AND M. RYAN, *A rectangular immersed finite element space for interface problems*, Adv. Comput. Theory Pract., 7 (2001), pp. 107–114.
- [30] T. LIN AND D. SHEEN, *The immersed finite element method for parabolic problems with the Laplace transformation in time discretization*, Int. J. Numer. Anal. Model., 10(2) (2013), pp. 298–313.
- [31] T. LIN AND X. ZHANG, *Linear and bilinear immersed finite elements for planar elasticity interface problems*, J. Comput. Appl. Math., 236(18) (2012), pp. 4681–4699.
- [32] X. LIU, R. FEDKIW, AND M. KANG, *A boundary condition capturing method for Poisson’s equation on irregular domains*, J. Comput. Phys., 160(1) (2000), pp. 151–178.
- [33] N. MARHEINEKE, J. LILJO, J. MOHRING, J. SCHNEBELE AND R. WEGENER, *Multiphysics and multimethods problem of rotational glass fiber melt-spinning*, Int. J. Numer. Anal. Model. Series B, 3(3) (2012), pp. 330–344.
- [34] H. MEYER, *Multidimensional Stefan problems*, SIAM J. Numer. Anal., 10 (1973), pp. 522–538.
- [35] R. NOUTCHEUWA AND R. G. OWEN, *A new incompressible smoothed particle hydrodynamics-immersed boundary method*, Int. J. Numer. Anal. Model. Series B, 3(2) (2012), pp. 126–167.
- [36] S. REDDY AND L. TREFETHEN, *Stability of the method of lines*, Numer. Math., 62(2) (1992), pp. 235–267.
- [37] W. E. SCHIESSER AND G. W. GRIFFITHS, *A compendium of partial differential equation models, Method of Lines Analysis with Matlab*, Cambridge University Press, Cambridge, 2009.
- [38] V. THOMEE, *Galerkin finite element methods for parabolic problems*, Springer Series in Computational Mathematics, 25, Springer-Verlag, Berlin, second edition, 2006.
- [39] K. WANG, H. WANG, AND X. YU, *An immersed Eulerian-Lagrangian localized adjoint method for transient advection-diffusion equations with interfaces*, Int. J. Numer. Anal. Model., 9(1) (2012), pp. 29–42.

- [40] C. WU, Z. LI, AND M. LAI, *Adaptive mesh refinement for elliptic interface problems using the non-conforming immersed finite element method*, *Int. J. Numer. Anal. Model.*, 8(3) (2011), pp. 466–483.
- [41] H. XIE, Z. LI, AND Z. QIAO, *A finite element method for elasticity interface problems with locally modified triangulations*, *Int. J. Numer. Anal. Model.*, 8(2) (2011), pp. 189–200.
- [42] M. YE, R. KHALEEL, AND T. J. YEH, *Stochastic analysis of moisture plume dynamics of a field injection experiment*, *Water Resour. Res.*, 41 (2005), W03013.
- [43] A. ZAFARULLAH, *Application of the method of lines to parabolic partial differential equations with error estimates*, *J. ACM*, 17(2) (1970), pp. 294–302.
- [44] Y. ZHOU AND G. WEI, *On the fictitious-domain and interpolation formulations of the matched interface and boundary (mib) method*, *J. Comput. Phys.*, 219(1) (2006), pp. 228–246.
- [45] Y. ZHOU, S. ZHAO, M. FEIG, AND G. WEI, *High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources*, *J. Comput. Phys.*, 213(1) (2006), pp. 1–30.