Journal of Scientific Computing (2024) 98:19
https://doi.org/10.1007/s10915-023-02414-z

Solving Navier–Stokes Equations with Stationary and Moving Interfaces on Unfitted Meshes

Yuan Chen[1] · **Xu Zhang**[2]

Received: 5 January 2023 / Revised: 12 September 2023 / Accepted: 12 November 2023

Abstract
This paper introduces a high-order immersed finite element (IFE) method to solve two-phase incompressible Navier–Stokes equations on interface-unfitted meshes. In spatial discretization, we use the newly developed immersed $\mathcal{P}_2$-$\mathcal{P}_1$ Taylor-Hood finite element. The unisolvency of new IFE basis functions is theoretically established. We introduce an enhanced partially penalized IFE method which includes the penalization on both interface edges and the interface itself. Ghost penalties are also added for pressure robustness. In temporal discretization, θ-schemes and backward differentiation formulas are adopted. Newton's method is used to handle the nonlinear advection. The proposed method completely circumvent re-meshing in tackling moving-interface problems. Thanks to the isomorphism of our IFE spaces with the standard finite element spaces, the new method enables efficient updates of global matrices, which significantly reduces the overall computational cost. Comprehensive numerical experiments show that the proposed method is third-order convergent for velocity and second-order for pressure in both stationary and moving interface cases.

Keywords Immersed finite element · Navier–Stokes equation · Interface problem · Moving interface · Unfitted mesh

Mathematics Subject Classification 35R05 · 65N15 · 65N30

1 Introduction

Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain, which is separated into $\Omega^+(t)$ and $\Omega^-(t)$ by an evolving interface $\Gamma(t)$. Consider the following unsteady Navier–Stokes interface problem

✉ Xu Zhang
xzhang@okstate.edu

Yuan Chen
chen.11050@buckeyemail.osu.edu

[1] Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA

[2] Department of Mathematics, Oklahoma State University, Stillwater, OK 74078, USA

Published online: 08 December 2023

in deformation tensor formulation:

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nabla\cdot\sigma(\mathbf{u}, p) = \mathbf{f}, \quad \text{on } \Omega\times[0, L], \tag{1.1a}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \text{on } \Omega\times[0, L], \tag{1.1b}$$

$$\mathbf{u} = \mathbf{0}, \quad \text{on } \partial\Omega\times[0, L], \tag{1.1c}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \ p(\mathbf{x}, 0) = p_0, \quad \text{on } \Omega. \tag{1.1d}$$

Here, the unknown $\mathbf{u}$ and p represent the velocity and the pressure of an incompressible fluid motion, respectively. $\mathbf{f}$, $\mathbf{u}_0$, and p_0 denote the given surface tension, initial velocity, and the initial pressure of the fluid. The stress tensor σ is defined by

$$\sigma(\mathbf{u}, p) = 2\nu\boldsymbol{\epsilon}(\mathbf{u}) - p\mathbf{I}, \tag{1.2}$$

where $\mathbf{I}$ is the identity tensor, and $\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ denotes the strain tensor. The fluid viscosity coefficient $\nu(\mathbf{x}, t)$ is assumed to be discontinuous across the interface $\Gamma(t)$. Without loss of generality, we assume that ν is a piecewise-constant function as follows

$$\nu(\mathbf{x}, t) = \begin{cases} \nu^+, \ \mathbf{x}\in\Omega^+(t), \\ \nu^-, \ \mathbf{x}\in\Omega^-(t). \end{cases} \tag{1.3}$$

We assume that the movement of the fluid interface $\Gamma(t)$ is governed by a velocity field $\mathbf{v}(\mathbf{x}, t)$, which is either given or part of unknowns, described as follows

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}, t), \quad \mathbf{x}\in\Gamma(t). \tag{1.4}$$

Across the interface $\Gamma(t)$, the homogeneous fluid interface jump conditions are imposed

$$[\![\mathbf{u}]\!]_{\Gamma(t)} = \mathbf{0}, \quad \text{on } \Gamma(t)\times[0, L], \tag{1.5a}$$

$$[\![\sigma(\mathbf{u}, p)\mathbf{n}]\!]_{\Gamma(t)} = \mathbf{0}, \quad \text{on } \Gamma(t)\times[0, L], \tag{1.5b}$$

where the jump $[\![\cdot]\!]_{\Gamma(t)}$ is defined as $[\![\mathbf{v}]\!]_{\Gamma(t)} = (\mathbf{v}|_{\Omega^+})_{\Gamma(t)} - (\mathbf{v}|_{\Omega^-})_{\Gamma(t)}$, and $\mathbf{n}$ is the unit normal vector pointing from Ω^- to Ω^+.

The two-phase flow interface problems, such as the Navier–Stokes interface problem (1.1), have wide applications in fluid mechanics and computational fluid dynamics, such as dispersion of bubbles [7], oil slick transportation [37], blood flows [41], etc. For computational simulations of these fluid flow interface problems, re-meshing and global interpolations between meshes are inevitable for conventional finite element methods due to the large deformation of the fluid interfaces. Therefore, numerical methods based on unfitted meshes are particularly desirable for these simulations, such as immersed interface method (IIM) [31, 32, 42], extended finite element method (XFEM) [10, 16, 38, 40], CutFEM [6, 8, 13, 15, 23], fictitious domain FEM [9, 14, 17], etc.

The immersed finite element method (IFEM) is a class of unfitted-mesh finite element methods for solving interface problems. The key idea of IFEM is to locally modify approximation functions to capture the non-smooth behavior of the solution at the interface. A distinctive feature of IFEM from other unfitted finite element methods mentioned above is that its approximating space is isomorphic to the standard finite element space on the same mesh. Not only is the mesh independent of the interface, but also the number and the location of degrees of freedom are intact. This distinguished feature enables a significant saving of computational cost when applying IFEM for solving moving interface problems because global interpolations between meshes are very efficient. The IFEM has been developed for many PDE interface models including elliptic equations [11, 19, 21, 22, 34], linear elasticity

equations [20, 35, 36], parabolic moving interface problems [18, 24, 33], hyperbolic moving interface problems [4, 5], to name only a few.

Recently, IFEM has been extended for fluid flow interface problems governed by Stokes equations. In [1], a $\mathcal{Q}_1$-$\mathcal{Q}_0$ IFE space was constructed and was used in the interior penalty discontinuous Galerkin formulation for Stokes interface problems. A class of low-order nonconforming IFE spaces was recently introduced in [28]. In [26], a priori error estimation for Crouzeix-Raviart IFE scheme was provided. A $\mathcal{Q}_1$-$\mathcal{Q}_0$ immersed discontinuous Galerkin scheme [2] and a mixed conforming-nonconforming IFEM [29, 30] have been presented for Stokes moving interface problems. For the high-order approximation of Stokes interface problems, we developed a high-order Taylor-Hood $\mathcal{P}_2$-$\mathcal{P}_1$ IFE space for Stokes interface problems [12].

For more general multiphase Navier–Stokes equations, a Crouzeix-Raviart nonconforming IFEM was introduced in [44], in which the backward Euler method was used for temporal discretization. Despite its simple construction and algorithm, this method is of low-order accuracy (first-order in time and second-order in space). In this paper, we develop a high-order IFEM for solving the Navier–Stokes moving interface problem. The proposed method is third-order accurate in velocity and second-order accurate in pressure. Since the velocity and stress interface jump conditions (1.5) are the same for the Navier–Stokes equations as for Stokes equations [27], we use the Taylor-Hood $\mathcal{P}_2$-$\mathcal{P}_1$ IFE spaces recently introduced in [12]. Some important properties of these IFE spaces including the unisolvency and partition of unity of IFE functions is proved in this paper. We introduce an enhanced partially penalized immersed finite element (EPPIFE) scheme, in which penalty terms are added on both interface edges and the interface itself. Penalization on the interface itself is critical for high-order polynomial approximation [3, 12, 19]. In addition, ghost penalty terms for both velocity and pressure are also added on interface edges and interface curves which yield robust pressure approximation. We find that these new penalty terms are crucial in high-order approximations for Navier–Stokes interface problems, which are carefully examined through extensive numerical experiments.

For the time discretization, we adopt the θ-schemes and high-order Backward Differentiation Formulas (BDF) methods. The Newton's method is used to handle the nonlinear advection terms. Due to the virtue of IFEM, re-meshing is not required throughout the whole computational process. Thanks to the isomorphism of IFE space with the standard FE space, the number and the location of degrees of freedom are independent of the interface. Only a small portion of elements change the interface configuration in two consecutive time steps; hence, only local modification from the previous global matrices is required instead of regenerating whole new matrices. This is a prominent feature for IFEM and enables efficient global matrices assembly.

The rest of this paper is organized as follows. In Sect. 2, we recall immersed $\mathcal{P}_2$-$\mathcal{P}_1$ finite element spaces. In Sect. 3, we analyze the unisolvency and partition of unity of the immersed $\mathcal{P}_2$-$\mathcal{P}_1$ finite element spaces. In Sect. 4, we introduce a semi-discrete scheme using the enhanced partially-penalized IFE method. Newton's method and full-discrete schemes are presented in Sect. 5. In Sect. 6, we provide some numerical experiments to test the convergences and accuracies of our methods for both stationary and moving interface cases. A brief summary is given in Sect. 7.

2 Immersed Finite Element Spaces

In this section, we introduce some preliminaries and notations, and then recall Taylor-Hood $\mathcal{P}_2$-$\mathcal{P}_1$ IFE spaces developed in [12].

2.1 Notations and Preliminaries

From now on, we assume that the domain Ω is polygonal, and that $\mathcal{T}_h$ is a shape regular triangulation of the domain Ω with mesh size h. Note that $\mathcal{T}_h$ does not need to align with the interface $\Gamma(t)$ at any time t. As a result, the mesh $\mathcal{T}_h$ are divided into two subsets: the set of interface elements $\mathcal{T}_h^i(t) = \{T \in \mathcal{T}_h \mid T \cap \Gamma(t) \neq \varnothing\}$ and the set of non-interface elements $\mathcal{T}_h^n(t) = \mathcal{T}_h \backslash \mathcal{T}_h^i(t)$. Although the background mesh $\mathcal{T}_h$ is time independent, $\mathcal{T}_h^n(t)$ and $\mathcal{T}_h^i(t)$ are time dependent due to the movement of the interface. For each interface element $T \in \mathcal{T}_h^i(t)$, we denote the interface segment by $\Gamma_T(t) = \Gamma(t) \cap T$. Define $\mathcal{F}_h^i(t) = \{\Gamma_T(t) \mid T \in \mathcal{T}_h^i(t)\}$ to be the set of interface segments on the mesh $\mathcal{T}_h$.

Let $\mathcal{E}_h$ be the collection of all edges of $\mathcal{T}_h$. Let $\mathcal{E}_h^i(t) = \{e \in \mathcal{E}_h \mid e \cap \Gamma(t) \neq \varnothing\}$ be the set of interface edges at time t, and $\mathcal{E}_h^n(t) = \mathcal{E}_h \backslash \mathcal{E}_h^i(t)$ be the set of non-interface edges. Again, $\mathcal{E}_h^i(t)$ and $\mathcal{E}_h^n(t)$ evolve with the moving interface. The collections of boundary edges and interior edges are denoted by $\mathcal{E}_h^b$ and $\mathcal{E}_h^o$, respectively. For each edge $e \in \mathcal{E}_h$, a unit normal vector is designated by $\mathbf{n}_e$. Note that the direction of the normal vector of an interior edge has no effect under our context. The direction of the normal vector $\mathbf{n}_e$ for a boundary edge $e \in \mathcal{E}_h^b$ is defined to be outward of the domain Ω. For an interface segment $\Gamma_T \in \mathcal{F}_h^i(t)$, the normal $\mathbf{n}_\Gamma$ is assigned to point from $\Omega^-(t)$ to $\Omega^+(t)$.

We also define the jump operator $[\![\cdot]\!]$ and the average operator $\{\cdot\}$ of a vector function $\mathbf{w}$. On an interior edge $e \in \mathcal{E}_h^o$ shared by two elements T_{e1} and T_{e2}, we define

$$[\![\mathbf{w}]\!]_e = \mathbf{w}|_{T_{e1}} - \mathbf{w}|_{T_{e2}}, \quad \{\mathbf{w}\}_e = \frac{1}{2}(\mathbf{w}|_{T_{e1}} + \mathbf{w}|_{T_{e2}}). \tag{2.1}$$

Here, the neighboring elements T_{e1} and T_{e2} are assigned such that the normal vector $\mathbf{n}_e$ points from T_{e1} to T_{e2}. On an interface segment $\Gamma_T \in \mathcal{F}_h^i(t)$, the jump and average operators are defined as:

$$[\![\mathbf{w}]\!]_{\Gamma_T} = \mathbf{w}|_{T^-} - \mathbf{w}|_{T^+}, \quad \{\mathbf{w}\}_{\Gamma_T} = \frac{1}{2}(\mathbf{w}|_{T^+} + \mathbf{w}|_{T^-}), \tag{2.2}$$

where $T^+ = T \cap \Omega^+(t)$ and $T^- = T \cap \Omega^-(t)$. On a boundary edge $e \in \mathcal{E}_h^b$ associated with the element T_e, the jump and average are defined as:

$$[\![\mathbf{w}]\!]_e = \{\mathbf{w}\}_e = \mathbf{w}|_e. \tag{2.3}$$

We use standard Sobolev spaces notation in this paper. Let V be a normed vector space equipped with norm $\|\cdot\|_V$. Then we define time dependent Sobolev spaces as follows: for $1 < p < \infty$

$$L^p(0, L; V) = \left\{ v : [0, L] \mapsto V \;\middle|\; \int_0^L \|v(\cdot, t)\|_V^p dt < \infty \right\}. \tag{2.4}$$

The space $H^1(0, L; V)$ is defined as follows

$$H^1(0, L; V) = \left\{ v : [0, L] \mapsto V \;\middle|\; v \in L^2(0, L; V) \text{ and } \partial_t v \in L^2(0, L; V) \right\}. \tag{2.5}$$

The space $H^k(0, L; V)$ with $k > 1$ can be defined in a similar way. We also introduce the mesh dependent space $\mathbf{V}_h(t)$:

$$\mathbf{V}_h(t) = \big\{(\mathbf{u}, p) \in [L^2(\Omega)]^3 : \mathbf{u}|_{\partial\Omega} = \mathbf{0} \text{ and } (\mathbf{u}, p) \text{ satisfies the following conditions } \mathbf{(A1)} - \mathbf{(A2)}\big\}. \tag{2.6}$$

(A1) $(\mathbf{u}, p)|_T \in [H^1(T)]^3$, if $T \in \mathcal{T}_h^n(t)$; $(\mathbf{u}, p)|_{T^\pm} \in [H^1(T^\pm)]^3$, for all $T \in \mathcal{T}_h^i(t)$.
(A2) $(\mathbf{u}, p)$ is continuous on every non-interface edge $e \in \mathcal{E}_h^n(t)$.

To ensure the mesh sufficiently resolves the interface $\Gamma(t)$, we assume that the triangulation $\mathcal{T}_h$ satisfies the following hypotheses [3, 10, 12, 44]:

(H1) The interface $\Gamma(t)$ can only intersect an element with two points, and these two points must be on different edges.
(H2) The interface $\Gamma(t)$ is a piecewise C^2 function, and the mesh $\mathcal{T}_h$ is formed such that on every interface element $T \in \mathcal{T}_h^i(t)$, the interface segment Γ_T is C^2.

2.2 Fictitious Elements

We use fictitious elements to construct local IFE functions [3, 12, 45]. Given a time t, let $T = \triangle A_1A_2A_3 \in \mathcal{T}_h^i$ be an interface triangle with the homothetic center O. A fictitious element of T, denoted by T^λ, is defined by $T^\lambda = \{X \in \mathbb{R}^2 : \exists Y \in T \text{ such that } \overrightarrow{OX} = (1+\lambda)\overrightarrow{OY}\}$ where λ is a scaling constant. We denote $\Gamma_T^\lambda = \Gamma(t) \cap T^\lambda$. These definitions are illustrated in Fig. 1 (Right), where $\triangle A_1^\lambda A_2^\lambda A_3^\lambda$ is the corresponding fictitious element of $\triangle A_1A_2A_3$, and $\Gamma_T = \widetilde{DE}$, $\Gamma_T^\lambda = \widetilde{D^\lambda E^\lambda}$. To this point, we need two more hypotheses under the context of fictitious elements [19]:

(H3) At any time t, the interface can only intersect each interface element T and its fictitious element T^λ at two distinct points. These intersection points must locate on two different edges of T and T^λ.
(H4) At any time t, there exists an integer N such that for each $K \in \mathcal{T}_h$, the number of elements in the set $\{T \in \mathcal{T}_h^i(t) : K \cap T^\lambda \neq \varnothing\}$ is bounded above by N.

2.3 Taylor-Hood IFE Spaces

We briefly recall the $\mathcal{P}_2$-$\mathcal{P}_1$ vector-valued IFE spaces [12] for Stokes interface problems. For a triangular element $T \in \mathcal{T}_h$ with vertices A_1, A_2, and A_3, let λ_i, $i = 1, 2, 3$ be the barycentric coordinates. Define three midpoints by $A_4 = \frac{1}{2}(A_1 + A_2)$, $A_5 = \frac{1}{2}(A_2 + A_3)$, $A_6 = \frac{1}{2}(A_3 + A_1)$. Let ψ_i, $i = 1, 2, \cdots, 6$ be the quadratic Lagrange shape functions on T. Then we have $\psi_i = (2\lambda_i - 1)\lambda_i$, for $i = 1, 2, 3$ and $\psi_4 = 4\lambda_1\lambda_2$, $\psi_5 = 4\lambda_2\lambda_3$, $\psi_6 = 4\lambda_3\lambda_1$. The vector-valued $\mathcal{P}_2$-$\mathcal{P}_1$ shape functions $\{\boldsymbol{\psi}_{i,T}, i \in \mathcal{I}\}$ with $\mathcal{I} = \{1, 2, ..., 15\}$ can be written as

$$\boldsymbol{\psi}_{i,T} = \begin{pmatrix} \psi_i \\ 0 \\ 0 \end{pmatrix} \quad 1 \le i \le 6, \qquad \boldsymbol{\psi}_{i,T} = \begin{pmatrix} 0 \\ \psi_{i-6} \\ 0 \end{pmatrix} \quad 7 \le i \le 12,$$

$$\boldsymbol{\psi}_{i,T} = \begin{pmatrix} 0 \\ 0 \\ \lambda_{i-12} \end{pmatrix} \quad 13 \le i \le 15.$$

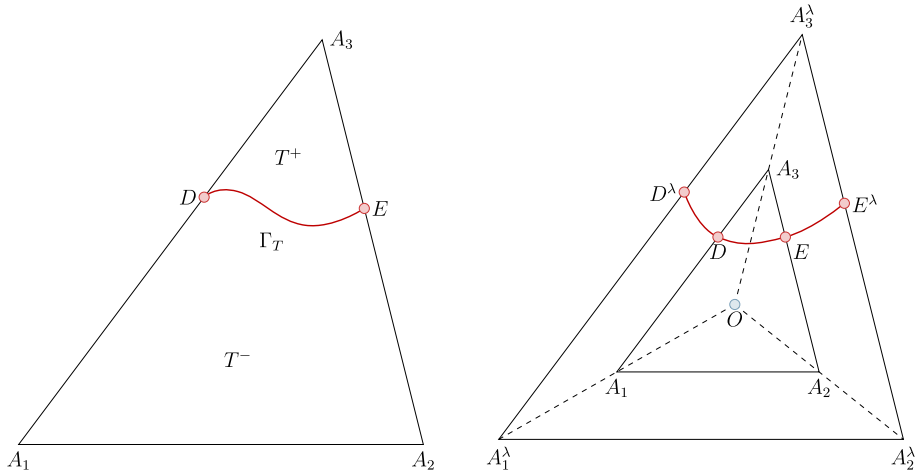

Fig. 1 An typical interface element (Left) and a fictitious element (Right)

These shape functions $\{\boldsymbol{\psi}_{i,T}, i \in \mathcal{I}\}$ are used on non-interface elements $T \in \mathcal{T}_h^n(t)$.

On an interface element $T \in \mathcal{T}_h^i(t)$, we construct piecewise $\mathcal{P}_2$-$\mathcal{P}_1$ shape functions according to the interface conditions (1.5). We define the nodes N_i such that $N_i = A_i$ for $1 \le i \le 6$, $N_{i+6} = A_i$ for $1 \le i \le 6$, and $N_{i+12} = A_i$ for $1 \le i \le 3$. Every interface element is divided into $T^+ := T \cap \Omega^+(t)$ and $T^- := T \cap \Omega^-(t)$ by the interface, as illustrated in the left plot of Fig. 1. The index set $\mathcal{I}$ is split into $\mathcal{I}^+ := \{i \in \mathcal{I}, N_i \in T^+\}$ and $\mathcal{I}^- := \{i \in \mathcal{I}, N_i \in T^-\}$. We define the piecewise $\mathcal{P}_2$-$\mathcal{P}_1$ function space on T by:

$$\begin{aligned}\mathcal{P}(T) = \big\{(\mathbf{u}, p) : (\mathbf{u}, p)|_{T^+} \in [\mathcal{P}_2\left(T^+\right)]^2 \times \mathcal{P}_1\left(T^+\right) \text{ and} \\ (\mathbf{u}, p)|_{T^-} \in [\mathcal{P}_2\left(T^-\right)]^2 \times \mathcal{P}_1\left(T^-\right)\big\}.\end{aligned} \tag{2.7}$$

For the convenience of construction, we introduce the following tensor-product space $\mathcal{S}(T)$:

$$\mathcal{S}(T) = [[\mathcal{P}_2(T)]^2 \times \mathcal{P}_1(T)] \times [[\mathcal{P}_2(T)]^2 \times \mathcal{P}_1(T)]. \tag{2.8}$$

It is obvious that $\mathcal{S}(T)$ is an isomorphism of $\mathcal{P}(T)$. A set of basis functions $\{\boldsymbol{\xi}_{i,T}\}_{i\in\mathcal{I}} \cup \{\boldsymbol{\eta}_{i,T}\}_{i\in\mathcal{I}}$ of $\mathcal{S}(T)$ can be written as follows

$$\boldsymbol{\xi}_{i,T} = \begin{cases} \left(\boldsymbol{\psi}_{i,T}, \mathbf{0}\right), & \text{if } i \in \mathcal{I}^+, \\ \left(\mathbf{0}, \boldsymbol{\psi}_{i,T}\right), & \text{if } i \in \mathcal{I}^-, \end{cases} \quad \boldsymbol{\eta}_{i,T} = \begin{cases} \left(\mathbf{0}, \boldsymbol{\psi}_{i,T}\right), & \text{if } i \in \mathcal{I}^+, \\ \left(\boldsymbol{\psi}_{i,T}, \mathbf{0}\right), & \text{if } i \in \mathcal{I}^-. \end{cases} \tag{2.9}$$

The following approximated interface conditions are used in the construction of IFE function. As stated in the Sect. 2.2, the interface conditions are imposed on the fictitious interface Γ_T^λ. Two physical jump conditions (1.5) imply the continuity of velocity and viscous stress tensor in normal direction:

$$[\![\mathbf{u}]\!]_{\Gamma_T^\lambda} = [\![\sigma(\mathbf{v}, q)\mathbf{n}]\!]_{\Gamma_T^\lambda} = \mathbf{0}. \tag{2.10}$$

Since the true solution $\mathbf{u}$ is divergence free, the finite element approximation of u is not necessarily divergence free, but it is within each element. We also impose the continuity of $\nabla \cdot \mathbf{u}$ across the interface:

$$[\![\nabla \cdot \mathbf{u}]\!]_{\Gamma_T^\lambda} = 0. \tag{2.11}$$

As we use high-order approximation for the velocity, two physical jump conditions (1.5) are insufficient to uniquely determine the IFE function; hence, we will need some additional jump conditions. As in the Laplacian extended jump conditions for elliptic equations [3], we use the following extended jump condition

$$[\![\nabla \cdot \sigma(\mathbf{u}, p)]\!]_{\Gamma_T^\lambda} = \mathbf{0}. \tag{2.12}$$

Finally, when the surface tension is small, we impose the continuity of pressure across the interface [25, 43]:

$$[\![p]\!]_{\Gamma_T^\lambda} = 0. \tag{2.13}$$

Combining these conditions (2.10) - (2.13), we define the following least-squares functional $\mathcal{J}_\lambda(\cdot,\cdot) : \mathcal{S}(T) \times \mathcal{S}(T) \mapsto \mathbb{R}^+$ to weakly enforce these approximating interface jump conditions across the fictitious interface Γ_T^λ:

$$\begin{aligned}\mathcal{J}_\lambda((\mathbf{u}, p), (\mathbf{v}, q)) = {} & \omega_0 \int_{\Gamma_T^\lambda} [\![\mathbf{u}]\!]\,[\![\mathbf{v}]\!]\, ds + \omega_1 h^2 \int_{\Gamma_T^\lambda} [\![\nabla \cdot \mathbf{u}]\!]\,[\![\nabla \cdot \mathbf{v}]\!]\, ds \\ & + \omega_2 h^2 \int_{\Gamma_T^\lambda} [\![\sigma(\mathbf{u}, p)\mathbf{n}]\!]\,[\![\sigma(\mathbf{v}, q)\mathbf{n}]\!]\, ds \\ & + \omega_3 h^4 \int_{\Gamma_T^\lambda} [\![\nabla \cdot \sigma(\mathbf{u}, p)]\!]\,[\![\nabla \cdot \sigma(\mathbf{v}, q)]\!]\, ds + \omega_4 h^2 \int_{\Gamma_T^\lambda} [\![p]\!]\,[\![q]\!]\, ds.\end{aligned} \tag{2.14}$$

where $\omega_i > 0, i = 0, \cdots, 4$ are the weights and the factors of h are used for scaling balance.

We construct the local IFE function $\boldsymbol{\phi}_T$ (in tensor form) as follows

$$\boldsymbol{\phi}_T|_{\mathbf{v},\mathbf{c}} = \sum_{i\in\mathcal{I}} v_i \boldsymbol{\xi}_{i,T} + \sum_{i\in\mathcal{I}} c_i \boldsymbol{\eta}_{i,T}, \tag{2.15}$$

where v_i denotes the given function value at nodes N_i. The coefficients c_i are solved from the following linear system

$$\mathbf{A}\mathbf{c} = -\mathbf{B}\mathbf{v}, \tag{2.16}$$

where

$$\mathbf{A} = \left(\mathcal{J}_\lambda\left(\boldsymbol{\eta}_{j,T}, \boldsymbol{\eta}_{i,T}\right)\right)_{i,j\in\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}, \qquad \mathbf{B} = \left(\mathcal{J}_\lambda\left(\boldsymbol{\xi}_{j,T}, \boldsymbol{\eta}_{i,T}\right)\right)_{i,j\in\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}. \tag{2.17}$$

It has been shown in [12] that c_i obtained from (2.16) minimizes $|\boldsymbol{\phi}_T|_{\mathcal{J}}$ with given values of v_i, where $|\cdot|_{\mathcal{J}}$ is the semi-norm induced by the bilinear form $\mathcal{J}(\cdot,\cdot)$. This semi-norm $|\cdot|_{\mathcal{J}}$ is a measure of fitness of interface jump conditions.

To find piecewise IFE function, we map functions in pairing form in space $\mathcal{S}(T)$ into piecewise function form in space $\mathcal{P}(T)$ by an isomorphic mapping $\mathcal{H}_T : \mathcal{S}(T) \mapsto \mathcal{P}(T)$:

$$\mathcal{H}_T(\boldsymbol{\phi}_T^+, \boldsymbol{\phi}_T^-) = \begin{cases} \boldsymbol{\phi}_T^+, & \text{on } T^+ \\ \boldsymbol{\phi}_T^-, & \text{on } T^-, \end{cases} \quad \text{for all } \boldsymbol{\phi}_T = (\boldsymbol{\phi}_T^+, \boldsymbol{\phi}_T^-) \in \mathcal{S}(T). \tag{2.18}$$

Finally, these piecewise IFE functions on the interface element $T \in \mathcal{T}_h^i$ can be written as:

$$\boldsymbol{\phi}_{i,T} = \mathcal{H}_T(\boldsymbol{\phi}_T|_{\mathbf{e}_i,\mathbf{c}_i}), \quad i \in \mathcal{I}. \tag{2.19}$$

With the preparation of local IFE shape functions $\boldsymbol{\phi}_{i,T}$, we construct the local IFE space $\mathbf{S}_{h,T}(t)$ for $T \in \mathcal{T}_h$:

$$\mathbf{S}_{h,T}(t) = \begin{cases} \text{Span}\{\boldsymbol{\psi}_{i,T},\ i \in \mathcal{I}\} & \text{if } T \in \mathcal{T}_h^n(t), \\ \text{Span}\{\boldsymbol{\phi}_{i,T},\ i \in \mathcal{I}\} & \text{if } T \in \mathcal{T}_h^i(t). \end{cases} \tag{2.20}$$

Accordingly, the global IFE space $\mathbf{S}_h(t)$ is defined by

$$\mathbf{S}_h(t) = \left\{(\mathbf{u},\, p) \in [L^2(\Omega)]^3 :\ (\mathbf{u},\, p) \text{ satisfies the following conditions } \mathbf{(C1)} - \mathbf{(C4)}\right\}. \tag{2.21}$$

(C1) $(\mathbf{u},\, p)|_T \in \mathbf{S}_{h,T}(t)$, for all $T \in \mathcal{T}_h$.
(C2) $(\mathbf{u},\, p)$ is continuous on every non-interface edge $e \in \mathcal{E}_h^n(t)$.
(C3) $\mathbf{u}$ is continuous at all vertices and midpoints of $T \in \mathcal{T}_h$.
(C4) p is continuous at all vertices of $T \in \mathcal{T}_h$.

For more details of Taylor-Hood $\mathcal{P}_2$-$\mathcal{P}_1$ IFE spaces, we refer readers to [12].

3 Unisolvency of Immersed Finite Element Spaces

In this section, we discuss the unisolvency and partition of unity of Taylor-Hood IFE spaces. Let $T \in \mathcal{T}_h^i(t)$ be an interface element. Let $\mathcal{V}_1 = \text{Span}\{\boldsymbol{\xi}_{i,T} : i \in \mathcal{I}\}$ and $\mathcal{V}_2 = \text{Span}\{\boldsymbol{\eta}_{i,T} : i \in \mathcal{I}\}$ where $\boldsymbol{\xi}_{i,T}$ and $\boldsymbol{\eta}_{i,T}$ are defined in (2.9). Clearly, $\mathcal{S}(T)$ can be decomposed to $\mathcal{S}(T) = \mathcal{V}_1 \oplus \mathcal{V}_2$. We denote the kernel of $\mathcal{J}_\lambda$ by $\mathcal{K}_\lambda = \{\boldsymbol{\phi} \in \mathcal{S}(T) : |\boldsymbol{\phi}|_{\mathcal{J}_\lambda} = 0\}$.

Theorem 3.1 (Uniqueness) *Let $T \in \mathcal{T}_h^i(t)$ be an interface element. Assume that the interface segment $T_\lambda \cap \Gamma(t)$ is a non-degenerate curve, i.e., not a straight line segment. Then each IFE function $\boldsymbol{\phi}_T$ is uniquely determined by given nodal values.*

Proof It suffices to show that the coefficient matrix $\mathbf{A}$ defined in (2.16) is nonsingular. We prove this result in two cases of interface curves.

Case 1 *$T_\lambda \cap \Gamma$ is a non-algebraic curve or an algebraic curve with order larger than two.*

Suppose $((\mathbf{u}_1,\, p_1),\, (\mathbf{u}_2,\, p_2)) \in \mathcal{K}_\lambda \cap \mathcal{V}_2$, then, $\mathbf{u}_1 - \mathbf{u}_2 = \mathbf{0}$ and $p_1 - p_2 = 0$ on $T_\lambda \cap \Gamma$. Since $\mathbf{u}_1 - \mathbf{u}_2 \in \mathcal{P}_2 \times \mathcal{P}_2$ and $p_1 - p_2 \in \mathcal{P}_1$, they cannot vanish on an algebraic curve of order larger than two or a non-algebraic curve. Thus, $\mathbf{u}_1 - \mathbf{u}_2 = \mathbf{0}$ and $p_1 - p_2 = 0$ on T_λ. Therefore, $\mathbf{u}_1$, $\mathbf{u}_2$ must vanish on all nodal points A_i for $1 \le i \le 6$, and p_1, p_2 vanishes on all A_i for $1 \le i \le 3$. This proves $|\cdot|_{\mathcal{J}_\lambda}$ is an inner product on $\mathcal{V}_2$. Thus, the coefficient matrix $\mathbf{A}$ is positive definite.

Case 2 *$T_\lambda \cap \Gamma$ is an algebraic curve of order two.*

Suppose $((\mathbf{u}_1,\, p_1),\, (\mathbf{u}_2,\, p_2)) \in \mathcal{K}_\lambda \cap \mathcal{V}_2$. It remains valid that $p_1 = p_2 \equiv 0$ due to $p_1 - p_2 \in \mathcal{P}_1$. Since $T_\lambda \cap \Gamma$ is an algebraic curve of order two, it can be written as $L(x,\, y) = 0$ where L is an irreducible second-order polynomial. Since $\mathbf{u}_1 - \mathbf{u}_2 = 0$ on $T_\lambda \cap \Gamma$, we can express $\mathbf{u}_1 - \mathbf{u}_2 = (k_1 L, k_2 L)$, where k_1 and k_2 are constants. The fact $((\mathbf{u}_1,\, p_1),\, (\mathbf{u}_2,\, p_2)) \in \mathcal{K}_\lambda$ implies $[\![\nabla \cdot \mathbf{u}]\!] = 0$ on $T_\lambda \cap \Gamma$. That is $k_1 L_x + k_2 L_y = 0$ on the curve $T_\lambda \cap \Gamma$. Thus, the vector $(L_x,\, L_y)$ is orthogonal to the constant vector $(k_1,\, k_2)$ on the interface curve. Since L is irreducible, we have $k_1 = k_2 = 0$, and therefore $\mathbf{u}_1 - \mathbf{u}_2 = \mathbf{0}$ on T_λ. Thus, the coefficient matrix $\mathbf{A}$ is positive definite. □

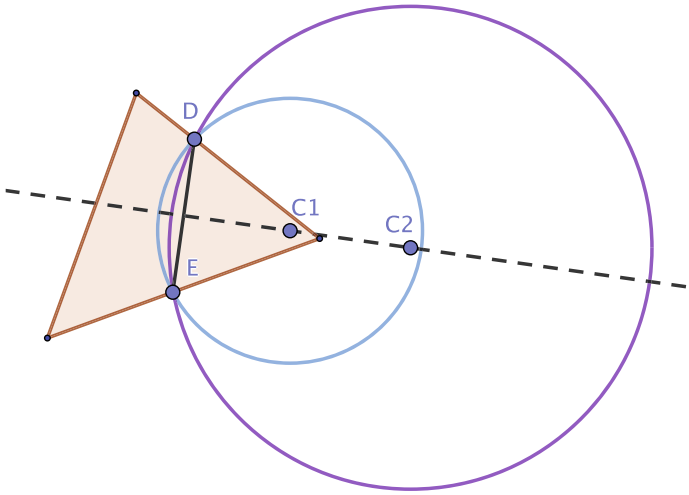

Fig. 2 Use the circle arc to approximate line

Remark 3.1 If $T_\lambda \cap \Gamma(t)$ is a degenerate curve, i.e., a straight line segment, the existence of IFE functions still holds [12]. However, the construction of IFE functions might not be unique. In fact, we find that the coefficient matrix $\mathbf{A}$ can be singular under certain configuration of interface location and viscosity coefficients. In particular, when the intersection points are $D = (0.5, 0)$, $E = (0.6403882032022076, 0.3596117967977924)$, and the viscosity coefficients are $\nu^+ = 0.03835954587269348$, $\nu^- = 1$ in the reference triangular element, the null space of $\mathbf{A}$ has a dimension of one. In this case, local IFE functions can still be constructed but they are not unique. Moreover, in this case, there exist non-zero functions in the IFE space such that they vanish at all degrees of freedom and satisfy the interface conditions (1.5) exactly.

Remark 3.2 In the case of the straight line interface with degenerate rank of coefficient matrix $\mathbf{A}$, we can use the pseudo inverse to uniquely solve the linear system (2.16). Another approach is to use a sequence of curves to geometrically approximate of a straight line segment. For example, we may construct a family of circles whose centers C_i locates in the perpendicular bisector of the line segment $\overline{DE}$ with the radius $r = |\overline{C_i D}|$. The curvature of these circles can be controlled by its radius. As increasing the value of r, the arc $\widetilde{DE}$ will approach to the line segment $\overline{DE}$. See the illustration in Fig. 2. Since the construction of IFE functions on any arc $\widetilde{DE}$ is unique, the construction of IFE function on the limiting line segment is also unique by letting the radius $r \to \infty$.

Remark 3.3 In the linear interface case, it requires 16 equations to enforce all interface conditions exactly (i.e. $\mathcal{J} \equiv 0$). However, there are only 15 degrees of freedom for P_2-P_1 elements, which results in an overdetermined linear system. Moreover, all the 16 conditions can be linear independent. This means, in some cases, there exists no piecewise polynomial functions exactly satisfying all interface conditions. For example, in Fig. 3, we plot an IFE shape function for the velocity on the reference triangle cut by a straight line interface $\Gamma_T : 2x + y + 27 - 20\sqrt{2} = 0$. We can see that the two pieces are not continuous across the straight line interface. More precisely, the least-squares functional $\mathcal{J}$ in this configuration

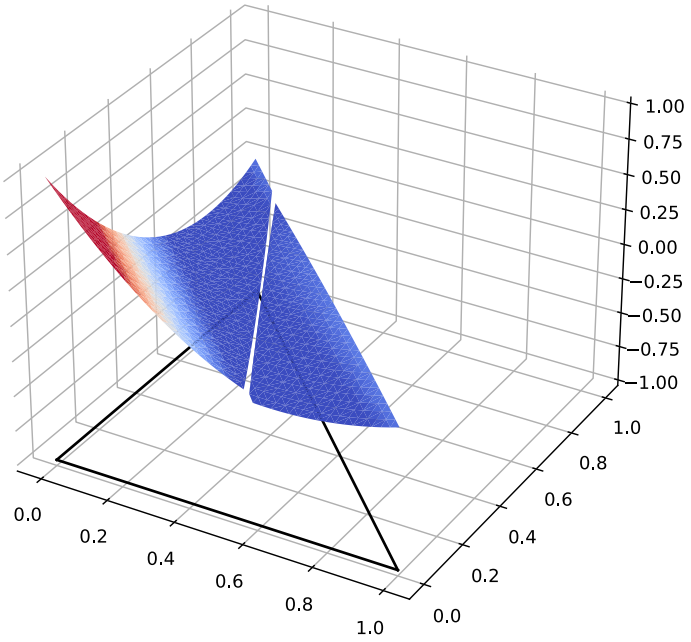

Fig. 3 An IFE shape function with the straight line interface $2x + y + 27 - 20\sqrt{2} = 0$

returns to the value of 0.13821871726173554, which is nonzero. In other words, the continuity condition $[\![u]\!]_{\Gamma_T} = 0$ is not satisfied exactly. This phenomenon is further explained in Numerical Example 6.1.

Remark 3.4 The least-squares construction can be extended to the construction of higher-order P_k-P_{k-1} Taylor-Hood IFE spaces with $k \geq 3$. However, this requires the further investigation on specifying of high-order extended interface jump conditions in addition to (2.12).

Theorem 3.2 (Partition of Unity) *Let $T \in \mathcal{T}_h^i(t)$ be an interface element. Under the assumptions of Theorem 3.1, the IFE shape functions $\boldsymbol{\phi}_{i,T}$ satisfy the following partition of unity property:*

$$\sum_{i\in\mathcal{I}} \boldsymbol{\phi}_{i,T} \equiv \mathbf{1}. \tag{3.1}$$

Proof Let $\boldsymbol{\phi}_T = \sum_{i\in\mathcal{I}} \boldsymbol{\phi}_{i,T}$, so $\mathcal{H}_T^{-1}\boldsymbol{\phi}_T \in \mathcal{S}(T)$. We then write $\boldsymbol{\phi}_T$ in the component form $\boldsymbol{\phi}_T = (\phi_{1,T}, \phi_{2,T}, \phi_{3,T})$. By direct computation, $\phi_{1,T}(A_i) = 1$ for $1 \leq i \leq 6$, $\phi_{2,T}(A_i) = 1$ for $1 \leq i \leq 6$, and $\phi_{3,T}(A_i) = 1$ for $1 \leq i \leq 3$. This is to say $\mathbf{v} = \mathbf{1}$. By the existence of the IFE functions [12], there exists a solution $\mathbf{c}$, such that

$$\boldsymbol{\psi}_T = \mathcal{H}_T^{-1}\boldsymbol{\phi}_T, \quad \boldsymbol{\phi}_T = \boldsymbol{\phi}_T|_{\mathbf{1},\mathbf{c}} = \sum_{i\in\mathcal{I}} \boldsymbol{\xi}_{i,T} + \sum_{i\in\mathcal{I}} c_i \boldsymbol{\eta}_{i,T}. \tag{3.2}$$

We note that $\mathbf{c} = \mathbf{1}$ is a solution due to the partition of unity of standard finite element shape functions $\boldsymbol{\psi}_{i,T}$. By the uniqueness result in Theorem 3.1, $\mathbf{c} = \mathbf{1}$ is the only solution. Thus

$$\boldsymbol{\phi}_T = \mathcal{H}_T\left(\sum_{i\in\mathcal{I}} \boldsymbol{\xi}_{i,T} + \sum_{i\in\mathcal{I}} \boldsymbol{\eta}_{i,T}\right) \equiv \mathbf{1}, \quad \text{on } T. \tag{3.3}$$

□

4 Semi-Discrete Schemes

In this section, we develop a semi-discrete scheme of the Navier–Stokes interface problem (1.1)–(1.5) using IFE spaces introduced above. We write the vector IFE space $\mathbf{V}_h(t) = \mathbf{W}_h(t) \times Q_h(t)$ and $\mathbf{S}_h(t) = \mathbf{U}_h(t) \times P_h(t)$. It is obvious that $\mathbf{S}_h(t) \subset \mathbf{V}_h(t)$ at any time t.

For a fixed time t, multiplying (1.1a) and (1.1b) by test functions $\mathbf{v} \in \mathbf{W}_h(t)$ and $q \in Q_h(t)$, respectively, and integrating over an element $T \in \mathcal{T}_h$, we obtain

$$\int_T \partial_t \mathbf{u} \cdot \mathbf{v} d\mathbf{x} + \int_T (\mathbf{u} \cdot \nabla)\mathbf{u} \cdot \mathbf{v} d\mathbf{x} - \int_T (\nabla \cdot \sigma(\mathbf{u}, p)) \cdot \mathbf{v} d\mathbf{x} - \int_T (\nabla \cdot \mathbf{u})\, q d\mathbf{x} = \int_T \mathbf{f} \cdot \mathbf{v} d\mathbf{x}. \tag{4.1}$$

If $T \in \mathcal{T}_h^n(t)$ is a non-interface element, by the divergence theorem, we get

$$\int_T (\nabla \cdot \sigma(\mathbf{u}, p)) \cdot \mathbf{v} d\mathbf{x} = \int_{\partial T} (\sigma(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} ds - \int_T \sigma(\mathbf{u}, p) : \nabla \mathbf{v} d\mathbf{x}. \tag{4.2}$$

If $T \in \mathcal{T}_h^i(t)$ is an interface element, we apply the divergence theorem separately on $T^{\pm}$ and sum up with respect to $s = \pm$ to obtain

$$\begin{aligned}\int_T (\nabla \cdot \sigma(\mathbf{u}, p)) \cdot \mathbf{v} d\mathbf{x} = &\int_{\partial T} (\sigma(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v} ds \\ &+ \int_{\Gamma_T} [\![(\sigma(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v}]\!] ds - \int_T \sigma(\mathbf{u}, p) : \nabla \mathbf{v} d\mathbf{x}\end{aligned} \tag{4.3}$$

Note on an interface edge $e \in \mathcal{E}_h^i(t)$ or an interface segment Γ_T, we have

$$\int_l [\![(\sigma(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{v}]\!] ds = \int_l \{\sigma(\mathbf{u}, p)\mathbf{n}\} \cdot [\![\mathbf{v}]\!] ds, \quad l = e \text{ or } \Gamma_T, \tag{4.4}$$

since $[\![fg]\!] = [\![f]\!]\{g\} + \{f\}[\![g]\!]$ and the fact $[\![\sigma(\mathbf{u}, p)\mathbf{n}]\!]_l = 0$. Then, summing up (4.2) over all non-interface elements $T \in \mathcal{T}_h^n(t)$, summing up (4.3) over all interface elements $T \in \mathcal{T}_h^i(t)$, and using the fact (4.4) we obtain

$$\begin{aligned}-\sum_{T\in\mathcal{T}_h} \int_T (\nabla \cdot \sigma(\mathbf{u}, p)) \cdot \mathbf{v} d\mathbf{x} = &\sum_{T\in\mathcal{T}_h} \int_T 2\nu\boldsymbol{\epsilon}(\mathbf{u}) : \nabla \mathbf{v} d\mathbf{x} + \sum_{T\in\mathcal{T}_h} \int_T p\nabla \cdot \mathbf{v} d\mathbf{x} \\ &- \sum_{e\in\mathcal{E}_h^i(t)} \int_e \{2\nu\boldsymbol{\epsilon}(\mathbf{u})\mathbf{n}\} \cdot [\![\mathbf{v}]\!] ds \\ &- \sum_{T\in\mathcal{T}_h^i(t)} \int_{\Gamma_T} \{2\nu\boldsymbol{\epsilon}(\mathbf{u})\mathbf{n}\} \cdot [\![\mathbf{v}]\!] ds \\ &+ \sum_{e\in\mathcal{E}_h^i(t)} \int_e \{p\mathbf{n}\} [\![\mathbf{v}]\!] ds + \sum_{T\in\mathcal{T}_h^i(t)} \int_{\Gamma_T} \{p\mathbf{n}\} [\![\mathbf{v}]\!] ds.\end{aligned} \tag{4.5}$$

Now we sum up (4.1) over all elements $T \in \mathcal{T}_h$ and use (4.5), then we obtain the weak formulation of the Navier–Stokes interface problem: find $(\mathbf{u}, p) \in H^1(0, L; \mathbf{W}_h(t)) \times L^2(0, L; Q_h(t))$ such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + C(\mathbf{u}, \mathbf{u}, \mathbf{v}) + K((\mathbf{u}, p), (\mathbf{v}, q)) = L_{\mathbf{f}}(\mathbf{v}), \quad \forall (\mathbf{v}, q) \in \mathbf{W}_h(t) \times Q_h(t). \tag{4.6}$$

In (4.6), the bilinear form $K(\cdot, \cdot) : \mathbf{V}_h(t) \times \mathbf{V}_h(t) \mapsto \mathbb{R}$ is defined as:

$$K((\mathbf{u}, p), (\mathbf{v}, q)) = A(\mathbf{u}, \mathbf{v}) + B(\mathbf{v}, p) + B(\mathbf{u}, q) + J_{\mathbf{u}}(\mathbf{u}, \mathbf{v}) - J_p(p, q) \tag{4.7}$$

where $A(\cdot, \cdot)$, $B(\cdot, \cdot)$, $C(\cdot, \cdot, \cdot)$, $L_{\mathbf{f}}(\cdot)$ are specified as follows:

$$\begin{aligned} A(\mathbf{u}, \mathbf{v}) = & \sum_{T \in \mathcal{T}_h} \int_T 2\nu \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) d\mathbf{x} - \sum_{e \in \mathcal{E}_h^i(t)} \int_e \{2\nu \boldsymbol{\epsilon}(\mathbf{u})\mathbf{n}\} \cdot [\![\mathbf{v}]\!] ds \\ & - \sum_{e \in \mathcal{E}_h^i(t)} \int_e \{2\nu \boldsymbol{\epsilon}(\mathbf{v})\mathbf{n}\} \cdot [\![\mathbf{u}]\!] ds \\ & + \frac{\sigma_e^0}{|e|} \sum_{e \in \mathcal{E}_h^i(t)} \int_e [\![\mathbf{u}]\!] \cdot [\![\mathbf{v}]\!] ds + \epsilon_1 \sum_{T \in \mathcal{T}_h^i(t)} \int_{\Gamma_T} \{2\nu \boldsymbol{\epsilon}(\mathbf{u})\mathbf{n}\} \cdot [\![\mathbf{v}]\!] ds \\ & + \epsilon_1 \sum_{T \in \mathcal{T}_h^i(t)} \int_{\Gamma_T} \{2\nu \boldsymbol{\epsilon}(\mathbf{v})\mathbf{n}\} \cdot [\![\mathbf{u}]\!] ds + \frac{\sigma_e^1}{h_T} \sum_{T \in \mathcal{T}_h^i(t)} \int_{\Gamma_T} [\![\mathbf{u}]\!] \cdot [\![\mathbf{v}]\!] ds, \end{aligned} \tag{4.8}$$

$$B(\mathbf{v}, p) = - \sum_{T \in \mathcal{T}_h} \int_T p \nabla \cdot \mathbf{v} d\mathbf{x} + \sum_{e \in \mathcal{E}_h^i(t)} \int_e \{p\mathbf{n}\} [\![\mathbf{v}]\!] ds - \epsilon_1 \sum_{T \in \mathcal{T}_h^i(t)} \int_{\Gamma_T} \{p\mathbf{n}\} [\![\mathbf{v}]\!] ds, \tag{4.9}$$

$$C(\mathbf{w}, \mathbf{u}, \mathbf{v}) = \sum_{T \in \mathcal{T}_h} \int_T (\mathbf{w} \cdot \nabla)\mathbf{u} \cdot \mathbf{v} d\mathbf{x}, \tag{4.10}$$

$$L_{\mathbf{f}}(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} d\mathbf{x}. \tag{4.11}$$

The ghost penalty terms $J_{\mathbf{u}}(\cdot, \cdot)$ and $J_p(\cdot, \cdot)$ in (4.7) are defined as:

$$J_{\mathbf{u}}(\mathbf{u}, \mathbf{v}) = \sum_{1 \le j \le k_{\mathbf{u}}} h_T^{2j-1} \left[\sigma_{\mathbf{u},j}^0 \sum_{e \in \mathcal{E}_h^i(t)} \int_e \{\nu\} \left[\!\!\left[\partial_{\mathbf{n}}^j \mathbf{u} \right]\!\!\right] : \left[\!\!\left[\partial_{\mathbf{n}}^j \mathbf{v} \right]\!\!\right] ds + \sigma_{\mathbf{u},j}^1 \sum_{T \in \mathcal{T}_h^i(t)} \int_{\Gamma_T} \{\nu\} \left[\!\!\left[\partial_{\mathbf{n}}^j \mathbf{u} \right]\!\!\right] : \left[\!\!\left[\partial_{\mathbf{n}}^j \mathbf{v} \right]\!\!\right] ds \right], \tag{4.12}$$

$$J_p(p, q) = \sum_{0 \le j \le k_p} h_T^{2j+1} \left[\sigma_{p,j}^0 \sum_{e \in \mathcal{E}_h^i(t)} \int_e \frac{1}{\{\nu\}} \left[\!\!\left[\partial_{\mathbf{n}}^j p \right]\!\!\right] \cdot \left[\!\!\left[\partial_{\mathbf{n}}^j q \right]\!\!\right] ds + \sigma_{p,j}^1 \sum_{T \in \mathcal{T}_h^i(t)} \int_{\Gamma_T} \frac{1}{\{\nu\}} \left[\!\!\left[\partial_{\mathbf{n}}^j p \right]\!\!\right] \cdot \left[\!\!\left[\partial_{\mathbf{n}}^j q \right]\!\!\right] ds \right] \tag{4.13}$$

In the above formulas, $\epsilon_1, \sigma_e^0, \sigma_e^1, \sigma_{\mathbf{u},j}^0, \sigma_{p,j}^0, \sigma_{\mathbf{u},j}^1$ and $\sigma_{p,j}^1$ are penalty constants. $k_{\mathbf{u}}$ and k_p are the order of polynomials for $\mathbf{u}$ and p. For $\mathcal{P}_2$-$\mathcal{P}_1$ element pair, $k_{\mathbf{u}} = 2$ and $k_p = 1$.

Using the vector IFE space $\mathbf{S}_h(t) = \mathbf{U}_h(t) \times P_h(t)$ to approximate the broken Sobolev space $\mathbf{V}_h(t)$, we obtain the **semi-discrete IFE scheme**: find $(\mathbf{u}_h, p_h) \in H^1(0, L; \mathbf{U}_h(t)) \times$

$L^2(0, L; P_h(t))$ such that

$$\begin{aligned}&(\partial_t \mathbf{u}_h, \mathbf{v}_h) + C(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + K((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h))\\ &= L_{\mathbf{f}}(\mathbf{v}_h), \quad \forall (\mathbf{v}_h, q_h) \in \mathbf{U}_h(t) \times P_h(t),\end{aligned} \tag{4.14}$$

subjected to the initial conditions:

$$\mathbf{u}_h(\cdot, 0) = \mathbf{u}_{0h}, \quad p_h(\cdot, 0) = p_{0h} \tag{4.15}$$

where $(\mathbf{u}_{0h}, p_{0h})$ is an approximation of $(\mathbf{u}_0, p_0)$ in $\mathbf{S}_h(\Omega)$, such as the Lagrange interpolation [12].

The proposed method (4.14)–(4.15) is an enhanced partially penalized immersed finite element (EPPIFE) method. Comparing to classical PPIFE method [34], additional penalty terms are added on interface segments Γ_T in (4.8) and (4.9), aiming to penalize the discontinuities of solution on the edges and interface introduced by our high order IFE spaces. In addition, ghost penalties terms (4.12)–(4.13) are used to improve the stability of the scheme especially for pressure terms. The effect of ghost penalties has been explored for Stokes interface problems [12]. In Sect. 6, we further discuss the necessity of these penalty terms for Navier–Stokes interface problems.

5 Newton's Method and Full-Discrete Schemes

In this section, we first present the Newton's method to handle the nonlinear advection term $C(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h)$. Then we introduce several temporal discrete schemes of (4.14).

5.1 Newton's Method

For the convenience of the notation, we use the steady-state Navier–Stokes interface problem to derive Newton's method. Numerical examples for this steady-state case are also provided in Sect. 6. To this end, we write the semi-discrete system (4.14) in the following operator form: find $(\mathbf{u}_h, p_h)$ as the root of the following nonlinear operator equation:

$$G(\mathbf{u}_h, p_h) = \mathbf{0} \tag{5.1}$$

where $G : \mathbf{S}_h(t) \mapsto \mathbf{S}_h^*(t)$ is the mapping $\mathbf{S}_h(t)$ to its dual space defined by

$$[G(\mathbf{u}, p)](\mathbf{v}, q) = C(\mathbf{u}, \mathbf{u}, \mathbf{v}) + K((\mathbf{u}, p), (\mathbf{v}, q)) - L_{\mathbf{f}}(\mathbf{v}).$$

Note that $\mathbf{0}$ in (5.1) denotes the zero operator. The Newton iteration can be written as: given $(\mathbf{u}^{(l-1)}, p^{(l-1)})$, seek for $(\mathbf{u}^{(l)}, p^{(l)})$ in

$$(\mathbf{u}^{(l)}, p^{(l)}) = (\mathbf{u}^{(l-1)}, p^{(l-1)}) - \left[G'(\mathbf{u}^{(l-1)}, p^{(l-1)})\right]^{-1} \left(G(\mathbf{u}^{(l-1)}, p^{(l-1)})\right) \tag{5.2}$$

where $G'(\mathbf{u}^{(l-1)}, p^{(l-1)})\left(\mathbf{h}_{\mathbf{u}}, h_p\right) \in \mathbf{S}_h^*(t)$ is the Gâteaux derivative of G at $(\mathbf{u}^{(l-1)}, p^{(l-1)})$ in the direction of $(\boldsymbol{h}_{\boldsymbol{u}}, h_p)$, i.e.,

$$\begin{aligned}G'(\mathbf{u}^{(l-1)}, p^{(l-1)})\left(\mathbf{h}_{\mathbf{u}}, h_p\right) &= \lim_{\varepsilon \to 0} \frac{G(\mathbf{u}^{(l-1)} + \varepsilon \mathbf{h}_{\mathbf{u}}, p^{(l-1)} + \varepsilon h_p) - G(\mathbf{u}^{(l-1)}, p^{(l-1)})}{\varepsilon}\\ &= K\left((\mathbf{h}_{\mathbf{u}}, h_p), (\mathbf{v}, q)\right) + C\left(\mathbf{h}_{\mathbf{u}}, \mathbf{u}^{(l-1)}, \mathbf{v}\right) + C\left(\mathbf{u}^{(l-1)}, \mathbf{h}_{\mathbf{u}}, \mathbf{v}\right).\end{aligned} \tag{5.3}$$

The Newton iteration (5.2) can be written as:

$$G'(\mathbf{u}^{(l-1)}, p^{(l-1)})\left(\mathbf{u}^{(l)}, p^{(l)}\right) = G'(\mathbf{u}^{(l-1)}, p^{(l-1)})\left(\mathbf{u}^{(l-1)}, p^{(l-1)}\right) - G\left(\mathbf{u}^{(l-1)}, p^{(l-1)}\right). \tag{5.4}$$

Substituting (5.3) in (5.4) with $(\boldsymbol{h_u}, h_p) = (\mathbf{u}^{(l)}, p^{(l)})$ and $(\boldsymbol{h_u}, h_p) = (\mathbf{u}^{(l-1)}, p^{(l-1)})$, respectively, we have the following iteration process: given $(\mathbf{u}^{(l-1)}, p^{(l-1)})$, compute $(\mathbf{u}^{(l)}, p^{(l)})$ in

$$\begin{aligned} &C(\mathbf{u}_h^{(l)}, \mathbf{u}_h^{(l-1)}, \mathbf{v}_h) + C(\mathbf{u}_h^{(l-1)}, \mathbf{u}_h^{(l)}, \mathbf{v}_h) + K\left((\mathbf{u}_h^{(l)}, p_h^{(l)}), (\mathbf{v}_h, q_h)\right) \\ &\quad = C(\mathbf{u}_h^{(l-1)}, \mathbf{u}_h^{(l-1)}, \mathbf{v}_h) + L_{\mathbf{f}}(\mathbf{v}_h). \end{aligned} \tag{5.5}$$

Once $(\mathbf{u}^{(l)}, p^{(l)})$ and $(\mathbf{u}^{(l-1)}, p^{(l-1)})$ are sufficiently close, we recognize it as the solution of Navier–Stokes interface problem.

5.2 θ-Schemes

For temporal discretization, we partition the time domain $[0, L]$ uniformly as follows

$$0 = t^0 < t^1 < \ldots < t^N = L, \quad \tau = t^n - t^{n-1}, \quad n = 1, 2, \ldots, N. \tag{5.6}$$

Denote $\mathbf{u}_h^n = \mathbf{u}_h(\mathbf{x}, t^n)$ and define

$$\begin{aligned} &\partial \mathbf{u}_h^{n+1/2} = \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\tau}, \quad \mathbf{u}_h^{n+\theta} = \theta \mathbf{u}_h^{n+1} + (1-\theta)\mathbf{u}_h^n, \\ &C^{n+\theta}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) = \theta C\left(\mathbf{u}_h^{n+1}, \mathbf{u}_h^{n+1}, \mathbf{v}_h\right) + (1-\theta) C\left(\mathbf{u}_h^n, \mathbf{u}_h^n, \mathbf{v}_h\right) \end{aligned} \tag{5.7}$$

where $\theta \in (0, 1)$. The θ-schemes of the Navier–Stokes interface problem is written as: Find $(\mathbf{u}_h^n, p_h^n) \in \mathbf{S}_h(t^n)$ such that

$$\left(\partial \mathbf{u}_h^{n+1/2}, \mathbf{v}_h\right) + C^{n+\theta}(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + K\left((\mathbf{u}_h^{n+\theta}, p_h^{n+\theta}), (\mathbf{v}_h, q_h)\right) = L_{\mathbf{f}^{n+\theta}}(\mathbf{v}_h) \tag{5.8}$$

for all $(\mathbf{v}_h, q_h) \in \mathbf{S}_h(t^n)$. We expand (5.8) and use Newton's Method (5.5) to linearize it at each time step:

$$\begin{aligned} &\left(\mathbf{u}_h^{n+1}, \mathbf{v}_h\right) + \tau\theta\left[C\left(\mathbf{u}_h^{n+1}, \mathbf{u}_h^{n+1}, \mathbf{v}_h\right) + K\left((\mathbf{u}_h^{n+1}, p_h^{n+1}), (\mathbf{v}_h, q_h)\right)\right] \\ &\quad = \tau L_{\mathbf{f}^{n+\theta}}(\mathbf{v}_h) + \left(\mathbf{u}_h^n, \mathbf{v}_h\right) - \tau(1-\theta)\left[C\left(\mathbf{u}_h^n, \mathbf{u}_h^n, \mathbf{v}_h\right) + K\left((\mathbf{u}_h^n, p_h^n), (\mathbf{v}_h, q_h)\right)\right]. \end{aligned} \tag{5.9}$$

We denote the related matrices and vectors as follows

$$\begin{aligned} (\mathbf{M}_n)_{ij} &= \left(\boldsymbol{\phi}_{\mathbf{u},j}^n, \boldsymbol{\phi}_{\mathbf{u},i}^n\right) \\ (\mathbf{K}_n)_{ij} &= A\left(\boldsymbol{\phi}_{\mathbf{u},j}^n, \boldsymbol{\phi}_{\mathbf{u},i}^n\right) + B\left(\boldsymbol{\phi}_{\mathbf{u},i}^n, \boldsymbol{\phi}_{p,j}^n\right) + B\left(\boldsymbol{\phi}_{\mathbf{u},j}^n, \boldsymbol{\phi}_{p,i}^n\right) \\ &\quad + J_{\mathbf{u}}\left(\boldsymbol{\phi}_{\mathbf{u},j}^n, \boldsymbol{\phi}_{\mathbf{u},i}^n\right) - J_p\left(\boldsymbol{\phi}_{p,j}^n, \boldsymbol{\phi}_{p,i}^n\right) \\ (\mathbf{F}_{\mathbf{n}+\theta})_i &= \theta L_{\mathbf{f}^{\mathbf{n}+1}}(\boldsymbol{\phi}_{\mathbf{u},i}^n) + (1-\theta) L_{\mathbf{f}^{\mathbf{n}}}(\boldsymbol{\phi}_{\mathbf{u},i}^n) \\ (\mathbf{KN}_n^{(l)})_{ij} &= C(\boldsymbol{\phi}_{\mathbf{u},j}^n, \mathbf{u}^{n,(l)}, \boldsymbol{\phi}_{\mathbf{u},i}^n) + C(\mathbf{u}^{n,(l)}, \boldsymbol{\phi}_{\mathbf{u},j}^n, \boldsymbol{\phi}_{\mathbf{u},i}^n) \\ (\mathbf{FN}_n^{(l)})_i &= C(\mathbf{u}^{n,(l)}, \mathbf{u}^{n,(l)}, \boldsymbol{\phi}_{\mathbf{u},i}^n) \\ (\mathbf{FN}_n)_i &= C(\mathbf{u}^n, \mathbf{u}^n, \boldsymbol{\phi}_{\mathbf{u},i}^n), \end{aligned}$$

where, $\boldsymbol{\phi}^n_{\mathbf{u},i}$ and $\boldsymbol{\phi}^n_{p,i}$ represent velocity and pressure components of the global IFE basis function $\boldsymbol{\phi}_i$ at time level t^n, and $(\cdot,\cdot)$ represents standard L^2 inner product. At each time level, Newton's method is used to find the solution $\mathbf{U}_{n+1}$ with initial value $\mathbf{U}^{(0)}_{n+1} = \mathbf{U}_n$. The **fully-discrete θ-scheme with Newton's iteration** can be written in the matrix form as follows: compute $\mathbf{U}^{(l)}_{n+1}$, $l = 1, 2, ..., L$ in

$$\begin{aligned}\left[\mathbf{M}_{n+1} + \tau\theta\left(\mathbf{KN}^{(l-1)}_{n+1} + \mathbf{K}_{n+1}\right)\right]\mathbf{U}^{(l)}_{n+1} \\ = \tau\mathbf{F}_{n+\theta} + [\mathbf{M}_n - \tau(1-\theta)\mathbf{K}_n]\,\mathbf{U}_n - \tau(1-\theta)\mathbf{FN}_n + \tau\theta\mathbf{FN}^{(l-1)}_{n+1}.\end{aligned} \tag{5.10}$$

Once $\|\mathbf{U}^{(l)}_{n+1} - \mathbf{U}^{(l-1)}_{n+1}\| \le tol$, a prescribed tolerance, we set $\mathbf{U}_{n+1} = \mathbf{U}^{(l)}_{n+1}$. The complete procedure is summarized in Algorithm 1. Note that when $\theta = 1$, this scheme is called Backward Euler method. When $\theta = 1/2$, this method is called Crank-Nicolson method.

Algorithm 1: θ-scheme for Navier-Stokes interface problem

Result: Solve $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_N$.
1 Compute $\mathbf{U}_0, \mathbf{K}_0, \mathbf{M}_0, \mathbf{F}_0, \mathbf{FN}_0$ from initial condition;
2 **for** $n = 0, 1, 2, ..., N-1$ **do**
3 Update $\mathbf{K}_{n+1}, \mathbf{M}_{n+1}, \mathbf{F}_{n+1}$;
4 $\mathbf{F}_{n+\theta} \leftarrow \theta\mathbf{F}_{n+1} + (1-\theta)\mathbf{F}_n$;
5 $\mathbf{U}^{(0)} \leftarrow \mathbf{U}_n$; $l \leftarrow 0$;
6 **while** $\|\mathbf{U}^l - \mathbf{U}^{l-1}\| \ge tol$ ***or*** $l \le L$ **do**
7 $l \leftarrow l + 1$;
8 Compute $\mathbf{KN}^{(l-1)}_{n+1}, \mathbf{FN}^{(l-1)}_{n+1}$;
9 $\mathbf{A}^{(l)}_{n+1} \leftarrow \mathbf{M}_{n+1} + \tau\theta\left(\mathbf{KN}^{(l-1)}_{n+1} + \mathbf{K}_{n+1}\right)$;
10 $\mathbf{b}^{(l)}_{n+1} \leftarrow \tau\mathbf{F}_{n+\theta} + (\mathbf{M}_n - \tau(1-\theta)\mathbf{K}_n)\,\mathbf{U}_n - \tau(1-\theta)\mathbf{FN}_n + \tau\theta\mathbf{FN}^{(l-1)}_{n+1}$;
11 Solve $\mathbf{A}^{(l)}_{n+1}\mathbf{U}^{(l)}_{n+1} = \mathbf{b}^{(l)}_{n+1}$;
12 **end**
13 $\mathbf{U}_{n+1} \leftarrow \mathbf{U}^{(l)}_{n+1}$;
14 Compute $\mathbf{FN}_{n+1}$;
15 **end**

Remark 5.1 This algorithm is set under general assumption, i.e. the moving interface case. For a fixed interface case, there is no need to repeatedly compute $\mathbf{K}_{n+1}$ and $\mathbf{M}_{n+1}$ in the step line 3 since these two matrices are independent with n.

Remark 5.2 We could slightly modify the above algorithm to save the computational cost when certain θ value is taken. For example, when $\theta = 1$, we do not need to compute $\mathbf{FN}_n$.

Remark 5.3 One apparent advantage of unfitted-mesh methods is that there is no need to regenerate mesh for a new interface location when assembling global matrices and vectors (line 3 of Algorithm 1). Moreover, with careful coding, one could further reduce the computational cost since it is unnecessary to re-assemble matrices $\mathbf{K}_{n+1}$ and $\mathbf{M}_{n+1}$ from scratch. In fact, only $\mathcal{O}(h^{-1})$ elements need to be updated when moving from time step t_n to t_{n+1}, so that we only need to update those local matrices and substitute them in $\mathbf{K}_{n+1}$ and $\mathbf{M}_{n+1}$, compared with $\mathcal{O}(h^{-2})$ elements computation under body-fitted mesh methods. This process

is written in Algorithm 2. Note that in real computation, we have $\mathbf{K}_n = \mathbf{A}_n + \mathbf{P}_n$ where $\mathbf{A}_n$ is non-penalized stiff matrix, $\mathbf{P}_n$ is the penalization matrix. It is easy to see from t_n to t_{n+1}, we need only to update interface elements at two time steps.

Algorithm 2: Fast Updating of $\mathbf{K}_{n+1}$ and $\mathbf{M}_{n+1}$

Result: $\mathbf{K}_{n+1}$ and $\mathbf{M}_{n+1}$.
$id \leftarrow \{\mathcal{T}_h^i \text{ at } t_n\} \cup \{\mathcal{T}_h^i \text{ at } t_{n+1}\}$;
for *element* ***in*** id **do**
  Assemble $\widetilde{\mathbf{A}}_n$ and $\widetilde{\mathbf{M}}_n$ for local DOF of *element* under basis function at t_n;
  Assemble $\widetilde{\mathbf{A}}_{n+1}$ and $\widetilde{\mathbf{M}}_{n+1}$ for local DOF of *element* under basis function at t_{n+1};
end
Compute Penalization Matrix $\mathbf{P}_{n+1}$ at t_{n+1};
$\mathbf{K}_{n+1} \leftarrow \mathbf{A}_n - \widetilde{\mathbf{A}}_n + \widetilde{\mathbf{A}}_{n+1} + \mathbf{P}_{n+1}$;
$\mathbf{M}_{n+1} \leftarrow \mathbf{M}_n - \widetilde{\mathbf{M}}_n + \widetilde{\mathbf{M}}_{n+1}$;

5.3 Backward Differentiation Formulas

Backward Differentiation Formulas (BDF) can also be used for temporal discretization of NS equation. In particular, the 2-step scheme BDF2 is

$$D_2\mathbf{v}^n = \frac{3\mathbf{v}^{n+1} - 4\mathbf{v}^n + \mathbf{v}^{n-1}}{2\tau}. \tag{5.11}$$

Approximating the derivative of $\partial_t\mathbf{u}$ by $D_2\mathbf{u}$, the BDF2 scheme is written as: for $n = 1, 2, ..., N-1$, for $\forall\,(\mathbf{v}_h, q_h) \in \mathbf{S}_h(t)$,

$$\left(D_2\mathbf{u}^n, \mathbf{v}_h\right) + C^{n+1}\left(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h\right) + K\left((\mathbf{u}_h^{n+1}, p_h^{n+1}), (\mathbf{v}_h, q_h)\right) = L_{\mathbf{f}^{n+1}}\left(\mathbf{v}_h\right). \tag{5.12}$$

We linearize this system using Newton's method at each time step n as above, then we can write the full-discrete scheme in the matrix form: find $\mathbf{U}_{n+1}^{(l)}$ in

$$\begin{aligned}&\left[3\mathbf{M}_{n+1} + 2\tau\left(\mathbf{KN}_{n+1}^{(l-1)} + \mathbf{K}_{n+1}\right)\right]\mathbf{U}_{n+1}^{(l)}\\ &\quad = 2\tau\mathbf{F}_{n+1} + 4\mathbf{M}_n\mathbf{U}_n - \mathbf{M}_{n-1}\mathbf{U}_{n-1} + 2\tau\mathbf{FN}_{n+1}^{(l-1)}.\end{aligned} \tag{5.13}$$

When $n = 0$, we can use the Backward Euler method to compute $\mathbf{U}_1$. Since the local truncation error is $\mathcal{O}(\tau^2)$, it provides sufficiently accurate initial approximations to ensure the global second-order convergence in time. The complete procedure is written in Algorithm 3.

For higher-order approximation in time, we can use the following 3-step scheme BDF3:

$$D_3\mathbf{v}^n = \frac{11\mathbf{v}^{n+1} - 18\mathbf{v}^n + 9\mathbf{v}^{n-1} - 2\mathbf{v}^{n-2}}{6\tau}. \tag{5.14}$$

We approximate derivative of $\partial_t\mathbf{u}$ by $D_3\mathbf{u}$, then the BDF3 scheme can be written as

$$\begin{aligned}&\left[11\mathbf{M}_{n+1} + 6\tau\left(\mathbf{KN}_{n+1}^{(l-1)} + \mathbf{K}_{n+1}\right)\right]\mathbf{U}_{n+1}^{(l)}\\ &\quad = 6\tau\mathbf{F}_{n+1} + 18\mathbf{M}_n\mathbf{U}_n - 9\mathbf{M}_{n-1}\mathbf{U}_{n-1} + 2\mathbf{M}_{n-2}\mathbf{U}_{n-2} + 6\tau\mathbf{FN}_{n+1}^{(l-1)}.\end{aligned} \tag{5.15}$$

Algorithm 3: BDF2 scheme for Navier-Stokes interface problem

Result: Solve $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_N$.
Compute $\mathbf{U}_0, \mathbf{K}_0, \mathbf{M}_0, \mathbf{F}_0, \mathbf{FN}_0$ from initial condition;
Compute $\mathbf{U}_1, \mathbf{M}_1$, from last step using Algorithm 1 with $\theta = 0.5$ or 1.0 and $N = 1$;
for $n = 1, 2, ..., N-1$ **do**
  Update $\mathbf{K}_{n+1}, \mathbf{M}_{n+1}, \mathbf{F}_{n+1}$;
  $\mathbf{U}^{(0)} \leftarrow \mathbf{U}_n; l \leftarrow 0;$
  while $\|\mathbf{U}^l - \mathbf{U}^{l-1}\| \geq tol$ ***or*** $l \leq L$ **do**
    $l \leftarrow l + 1;$
    Compute $\mathbf{KN}_{n+1}^{(l-1)}, \mathbf{FN}_{n+1}^{(l-1)}$;
    $\mathbf{A}_{n+1}^{(l)} \leftarrow 3\mathbf{M}_{n+1} + 2\tau\left(\mathbf{KN}_{n+1}^{(l-1)} + \mathbf{K}_{n+1}\right)$;
    $\mathbf{b}_{n+1}^{(l)} \leftarrow 2\tau\mathbf{F}_{n+1} + 4\mathbf{M}_n\mathbf{U}_n - \mathbf{M}_{n-1}\mathbf{U}_{n-1} + 2\tau\mathbf{FN}_{n+1}^{(l-1)}$;
    Solve $\mathbf{A}_{n+1}^{(l)}\mathbf{U}_{n+1}^{(l)} = \mathbf{b}_{n+1}^{(l)}$;
  end
  $\mathbf{U}_{n+1} \leftarrow \mathbf{U}_{n+1}^{(l)};$
end

To start the iteration, we use Crank-Nicolson method to find $\mathbf{U}_1$ and $\mathbf{U}_2$. The Crank-Nicolson method has a local truncation error of order $\mathcal{O}(\tau^3)$ which provides sufficient accurate initial approximations to ensure the overall third-order convergence. This whole procedure is summerized in Algorithm 4.

Algorithm 4: BDF3 scheme for Navier-Stokes interface problem

Result: We would like to solve $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_N$.
Compute $\mathbf{U}_0, \mathbf{K}_0, \mathbf{M}_0, \mathbf{F}_0, \mathbf{FN}_0$ from initial condition;
Compute $\mathbf{U}_1, \mathbf{K}_1, \mathbf{M}_1, \mathbf{F}_1, \mathbf{FN}_1$ and $\mathbf{U}_2, \mathbf{M}_2$ from last step using Algorithm 1 with $\theta = 0.5$ and $N = 2$;
for $n = 2, ..., N-1$ **do**
  Update $\mathbf{K}_{n+1}, \mathbf{M}_{n+1}, \mathbf{F}_{n+1}$;
  $\mathbf{U}^{(0)} \leftarrow \mathbf{U}_n; l \leftarrow 0;$
  while $l == 0$ ***or*** $\|\mathbf{U}^l - \mathbf{U}^{l-1}\| \geq tol$ ***and*** $l \leq L$ **do**
    $l \leftarrow l + 1;$
    Compute $\mathbf{KN}_{n+1}^{(l-1)}, \mathbf{FN}_{n+1}^{(l-1)}$;
    $\mathbf{A}_{n+1}^{(l)} \leftarrow 11\mathbf{M}_{n+1} + 6\tau\left(\mathbf{KN}_{n+1}^{(l-1)} + \mathbf{K}_{n+1}\right)$;
    $\mathbf{b}_{n+1}^{(l)} \leftarrow 6\tau\mathbf{F}_{n+1} + 18\mathbf{M}_n\mathbf{U}_n - 9\mathbf{M}_{n-1}\mathbf{U}_{n-1} + 2\mathbf{M}_{n-2}\mathbf{U}_{n-2} + 6\tau\mathbf{FN}_{n+1}^{(l-1)}$;
    Solve $\mathbf{A}_{n+1}^{(l)}\mathbf{U}_{n+1}^{(l)} = \mathbf{b}_{n+1}^{(l)}$;
  end
  $\mathbf{U}_{n+1} \leftarrow \mathbf{U}_{n+1}^{(l)};$
end

6 Numerical Examples

In this section, we report the performance of the proposed IFE schemes. All numerical experiments are carried out on the rectangular domain $\Omega = [-1, 1]^2$ and the time interval

$[0, 1]$. We use Cartesian triangular meshes which are obtained by first partitioning Ω into $N \times N$ uniform rectangles, and then cut each rectangle into two triangles by the diagonal connecting the upper left and lower right vertices. The mesh size $h = 2/N$. The parameters in computing the least-squares IFE basis functions (2.14) are $\omega_0 = \max(\nu^+, \nu^-)$, $\omega_1 = 1$, $\omega_2 = \max(\nu^+, \nu^-)$, $\omega_3 = 1$, $\omega_4 = 100 \max(\nu^+, \nu^-)$. The fictitious element parameter is set to be $\lambda = 1.0$, and the threshold of Newton method is $tol = 10^{-8}$.

In the first experiment, we consider a steady-state Navier–Stokes interface equations with a straight line interface. We would like to see if the new EPPIFE scheme can recover the exact solution if it is a piecewise quadratic polynomial in velocity and piecewise linear in pressure.

Example 6.1 (*Steady problem with a straight line interface*) In this example, we consider a straight line interface $\Gamma(\mathbf{x}) : 2x + y - c = 0$ which separates the domain Ω into two subdomains $\Omega^+ = \{(x, y) \in \Omega : 2x + y - c > 0\}$ and $\Omega^- = \{(x, y) \in \Omega : 2x + y - c < 0\}$. We choose $c = \sqrt{2}$ and the viscosity coefficient $(\nu^+, \nu^-) = (100, 1)$. The exact solution of this problem is

$$\mathbf{u}(\mathbf{x}) = \begin{cases} \begin{pmatrix} (2x + y - c)^2/(2\nu^+) \\ -(2x + y - c)^2/\nu^+ \end{pmatrix}, \mathbf{x} \in \Omega^+ \\ \begin{pmatrix} (2x + y - c)^2/(2\nu^-) \\ -(2x + y - c)^2/\nu^- \end{pmatrix}, \mathbf{x} \in \Omega^-, \end{cases}$$

and

$$p(\mathbf{x}) = x - y.$$

One can verify that the interface jump conditions (1.5) are satisfied. Note that the exact solution $(\mathbf{u}, p) \in \mathbf{S}_h(\Omega)$; hence, we expect our EPPIFE method recover the exact solution. For comparison, we also solve this problem using "non-penalized" and "reduced-penalized" IFE schemes. Table 1 reports numerical results using the classical IFE scheme in which all consistency terms and penalty terms excluded. Table 2 contains the classical PPIFE solution with only penalization on interface edges, but not on the interface segment or ghost penalty, i.e., $\epsilon_1, \sigma_e^1, \sigma_{\mathbf{u},j}^0, \sigma_{\mathbf{u},j}^1, \sigma_{p,j}^0, \sigma_{p,j}^1$ are taken to be 0. The full EPPIFE scheme (4.7) with ghost penalty is used to generate Table 3. We can see that errors in Table 3 are apparently much less than Tables 1 and 2. In fact, only round-off errors are observed when using enhanced PPIFE scheme; hence, the enhanced PPIFE scheme recovers the exact solution if the solution is in the IFE space. Neither the "reduced" PPIFE scheme nor the classical "non-penalized" IFE scheme can recover the true solution, as indicated in Table 3 and Table 1. The accuracy of the PPIFE solution is still better than that of classical IFE solutions, and this indicates the significance of penalization in IFE scheme. However, for high-order approximations ($\mathcal{P}_2$ for velocity), penalization over the interface curve itself is also necessary. We note that for even for straightline interface, the penalty on interface itself is necessary, since the construction of IFE spaces using least-squares does not guarantee pointwise continuity of the interface conditions, as stated in Remark 3.3.

Example 6.2 (*Steady problem with a curved interface*) In this example, we test our enhanced PPIFE method on a curved interface. This example was also used in [44]. The domain Ω is split into Ω^+ and Ω^- by a circular interface $\Gamma = \{(x, y) \in \Omega : x^2 + y^2 - r^2 = 0\}$ such that $\Omega^+ = \{(x, y) \in \Omega : x^2 + y^2 - r^2 > 0\}$ and $\Omega^- = \{(x, y) \in \Omega : x^2 + y^2 - r^2 < 0\}$, where $r = \sqrt{0.3}$. The viscosity coefficient is chosen as $(\nu^+, \nu^-) = (10, 1)$. The exact solutions are

Table 1 Classical "non-penalized" IFE solution of a line interface in Example 6.1

λ	N	$\|u_h^1-u^1\|_{L^2}$	order	$\|u_h^2-u^2\|_{L^2}$	order	$\|p_h-p\|_{L^2}$	order
1.0	10	1.06×10^{-2}		6.20×10^{-3}		5.80×10^{-1}	
	20	8.70×10^{-3}	0.28	5.27×10^{-3}	0.24	1.24×10^{0}	−1.10
	40	2.69×10^{-3}	1.69	1.74×10^{-3}	1.60	3.16×10^{-1}	1.97
	N	$\|u_h^1-u^1\|_{H^1}$	order	$\|u_h^2-u^2\|_{H^1}$	order	$\|p_h-p\|_{H^1}$	order
	10	1.14×10^{-1}		4.95×10^{-2}		1.72×10^{1}	
	20	1.34×10^{-1}	−0.23	8.73×10^{-2}	−0.82	7.39×10^{1}	−2.10
	40	4.85×10^{-2}	1.47	5.58×10^{-2}	0.65	3.49×10^{1}	1.08

Table 2 Classical PPIFE solution of a line interface in Example 6.1

λ	N	$\|u_h^1-u^1\|_{L^2}$	order	$\|u_h^2-u^2\|_{L^2}$	order	$\|p_h-p\|_{L^2}$	order
1.0	10	1.59×10^{-3}		1.07×10^{-3}		1.33×10^{-1}	
	20	5.87×10^{-4}	1.44	3.95×10^{-4}	1.43	3.99×10^{-2}	1.73
	40	1.41×10^{-4}	2.06	9.46×10^{-5}	2.06	1.82×10^{-2}	1.14
	N	$\|u_h^1-u^1\|_{H^1}$	order	$\|u_h^2-u^2\|_{H^1}$	order	$\|p_h-p\|_{H^1}$	order
	10	1.23×10^{-2}		8.12×10^{-3}		3.55×10^{0}	
	20	7.65×10^{-3}	0.68	5.75×10^{-3}	0.50	1.66×10^{0}	1.10
	40	4.46×10^{-3}	0.78	3.75×10^{-3}	0.62	1.63×10^{0}	0.02

Table 3 EPPIFE solution of a line interface in Example 6.1

λ	N	$\|u_h^1-u^1\|_{L^2}$	order	$\|u_h^2-u^2\|_{L^2}$	order	$\|p_h-p\|_{L^2}$	order
1.0	10	2.64×10^{-13}		4.27×10^{-13}		1.77×10^{-10}	
	20	4.23×10^{-13}	-	8.44×10^{-13}	-	1.10×10^{-11}	-
	40	1.93×10^{-12}	-	3.87×10^{-12}	-	5.40×10^{-12}	-
	N	$\|u_h^1-u^1\|_{H^1}$	order	$\|u_h^2-u^2\|_{H^1}$	order	$\|p_h-p\|_{H^1}$	order
	10	6.35×10^{-12}		5.72×10^{-12}		4.66×10^{-9}	
	20	1.29×10^{-12}	-	1.46×10^{-12}	-	4.78×10^{-10}	-
	40	2.03×10^{-12}	-	2.66×10^{-12}	-	4.69×10^{-10}	-

chosen as

$$\mathbf{u}(\mathbf{x})=\begin{cases}\begin{pmatrix} y(x^2+y^2-r^2)/\nu^+ \\ -x(x^2+y^2-r^2)/\nu^+\end{pmatrix}, & \mathbf{x}\in\Omega^+\\ \begin{pmatrix} y(x^2+y^2-r^2)/\nu^- \\ -x(x^2+y^2-r^2)/\nu^-\end{pmatrix}, & \mathbf{x}\in\Omega^-,\end{cases}$$

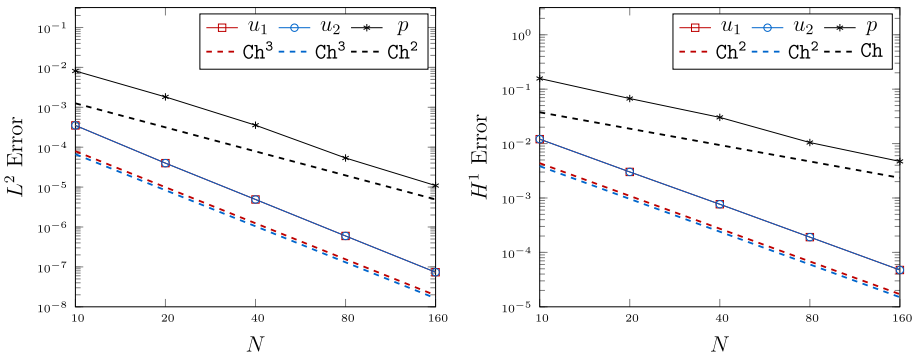

Fig. 4 Convergence of the EPPIFE solution of a circular interface in Example 6.2

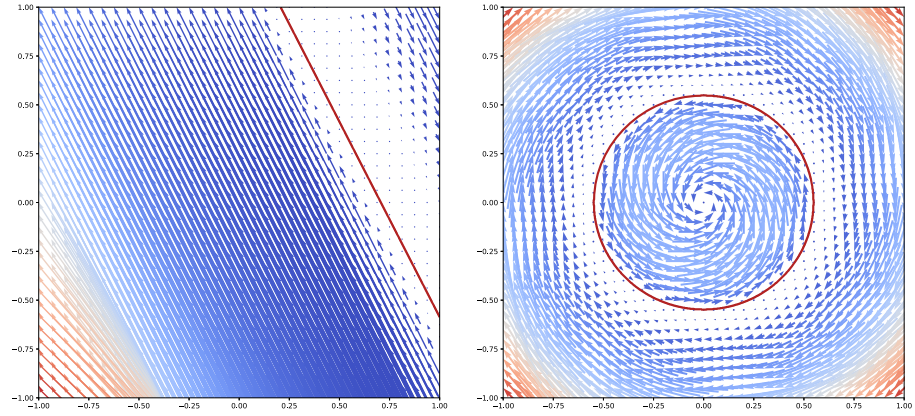

Fig. 5 The velocity fields of line and circular interface in Examples 6.1 and 6.2

and

$$p(\mathbf{x}) = \frac{1}{10}(x^3 - y^3).$$

The convergence results are presented in Fig. 4. The dash lines are reference line indicating the expected orders of convergence. These results indicate that our enhanced PPIFE scheme for steady-state Navier–Stokes interface problem has optimal convergence rates, i.e.,

$$\|u_h^1 - u^1\|_{L^2} + \|u_h^2 - u^2\|_{L^2} + h\|p_h - p\|_{L^2} \approx \mathcal{O}(h^3),$$

and

$$|u_h^1 - u^1|_{H^1} + |u_h^2 - u^2|_{H^1} + h|p_h - p|_{H^1} \approx \mathcal{O}(h^2).$$

Illustration of the velocity fields for Examples 6.1 and 6.2 are depicted in Fig. 5.

Example 6.3 (*Unsteady problem with a stationary curved interface*) In this example, we consider an unsteady Navier–Stokes interface problem with a stationary circular interface.

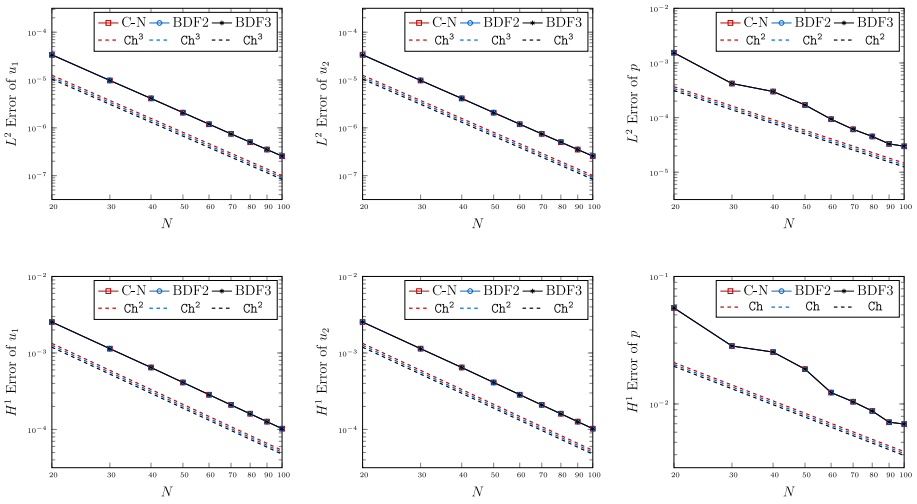

Fig. 6 Convergence of the enhanced PPIFE solutions in L^2 norm (top) and H^1 norm (bottom) at $t = 1$ of a circular interface in Example 6.3

The exact solutions are given as

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} \begin{pmatrix} y(x^2 + y^2 - r^2)\sin(t)/\nu^+ \\ -x(x^2 + y^2 - r^2)\sin(t)/\nu^+ \end{pmatrix}, \ \mathbf{x} \in \Omega^+ \\ \begin{pmatrix} y(x^2 + y^2 - r^2)\sin(t)/\nu^- \\ -x(x^2 + y^2 - r^2)\sin(t)/\nu^- \end{pmatrix}, \ \mathbf{x} \in \Omega^-, \end{cases}$$

and

$$p(\mathbf{x}, t) = \frac{1}{10}(x^3 - y^3)\sin(t).$$

We apply three full-discrete IFE scheme: Crank-Nicolson, BDF2, BDF3. For second-order time discrete schemes, Crank-Nicolson and BDF2, we set $\tau = \mathcal{O}(h^{3/2})$. More precisely, we divide the time interval $[0, 1]$ into $M = \lceil 1/h^{3/2} \rceil$ subintervals, and set $\tau = 1/M$ in our computation. For the third-order BDF3 scheme, we set $\tau = h$. This is to ensure the error in the temporal discretization is compatible with the error in the spatial discretization. All errors are computed at the final time $t = 1$. Convergence of full-discrete IFE solutions are reported in the Fig. 6. We observe optimal convergence rates in all three temporal-discretizing schemes are similar in this example.

In the next three examples, we test our EPPIFE schemes for Navier–Stokes moving interface problems. Three typical types of interface movements: translation, stretch, and deformation, are considered in Examples 6.4, 6.5, and 6.6, respectively. Figure 7 illustrates these interface movements in which the dashed lines represent initial interface state while the solid lines represent the final interface state. The EPPIFE-BDF3 scheme is used in the following three examples. All experiments are carried out on fixed Cartesian triangular meshes without re-meshing.

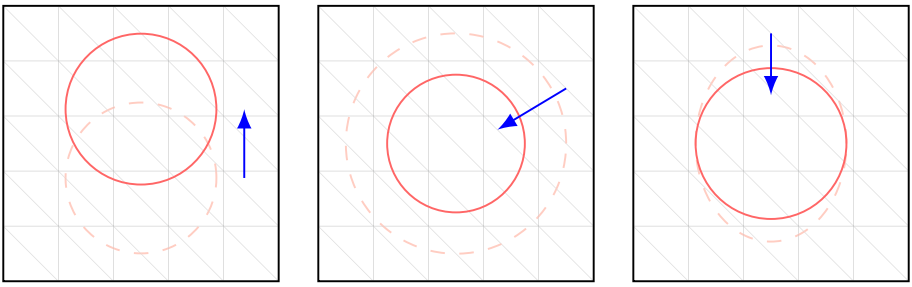

Fig. 7 Interface Movements: translation (left), stretch (middle), deformation (right)

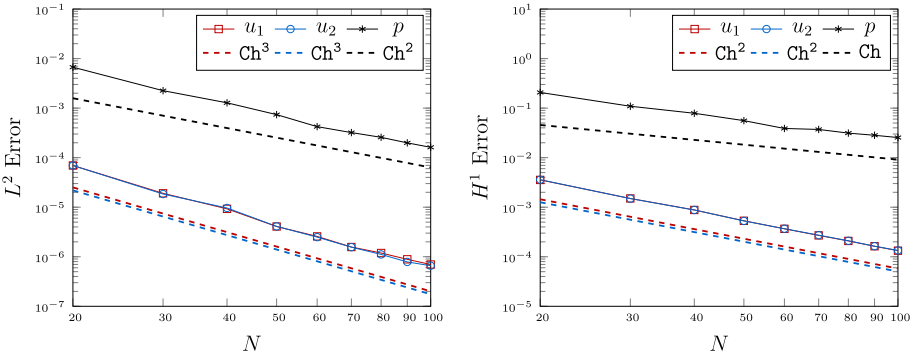

Fig. 8 BDF3-PPIFE error of translating circular interface $t = 1$ for Example 6.4

Example 6.4 (*Unsteady problem with moving interface: translation*) In this example, the interface is set to be a rising circle along y axis. The exact solution is follows

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} \begin{pmatrix} (y-q)[(x-s)^2 + (y-q)^2 - r^2]/\nu^+ \\ -(x-s)[(x-s)^2 + (y-q)^2 - r^2]/\nu^+ \end{pmatrix}, \mathbf{x} \in \Omega^+(t), \\ \begin{pmatrix} (y-q)[(x-s)^2 + (y-q)^2 - r^2]/\nu^- \\ -(x-s)[(x-s)^2 + (y-q)^2 - r^2]/\nu^- \end{pmatrix}, \mathbf{x} \in \Omega^-(t), \end{cases}$$

and

$$p(\mathbf{x}, t) = \frac{1}{10}(x^3 - y^3)$$

where

$$q(t) = q_a t + q_b, \quad s(t) = s_a t + s_b$$

with $r^2 = 0.3$, $q_a = -0.25$, $q_b = 0.5$, $s_a = s_b = 0$. The viscosity coefficients are set to be $(\nu^+, \nu^-) = (10, 1)$.

Optimal convergence rates for velocity and pressure are observed in both L^2 and H^1 norms. The convergence plots are shown in Fig. 8. The velocity fields at times $t = 0$, 0.5, and 1.0 are depicted in Fig. 9.

Example 6.5 (*Unsteady problem with moving interface: Stretch*) In this example, the interface is set to be a circle centered in $(0, 0)$ with varying radius. We set the radius $r(t)$ of circle to

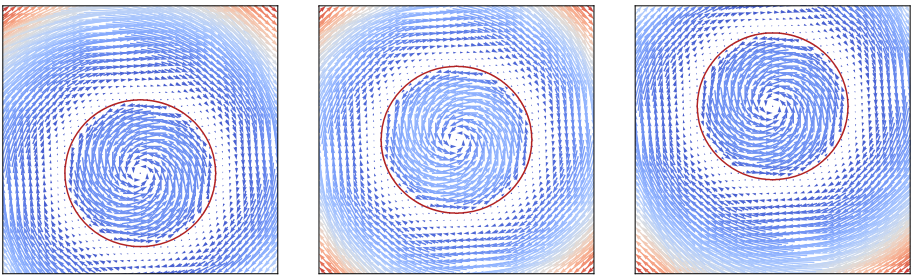

Fig. 9 The velocity fields of translating circular interface when $T = 0,\ 0.5,\ 1.0$ for Example 6.4

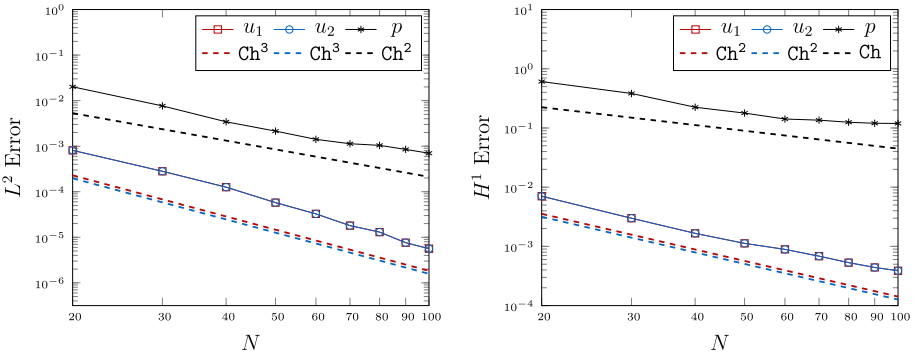

Fig. 10 BDF3-PPIFE error of stretching circular interface for Example 6.5

be a periodic function of time t. The exact solution is set to be

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} \begin{pmatrix} y(x^2+y^2-r^2)/\nu^+ \\ -x(x^2+y^2-r^2)/\nu^+ \end{pmatrix}, & \mathbf{x} \in \Omega^+(t), \\ \begin{pmatrix} y(x^2+y^2-r^2)/\nu^- \\ -x(x^2+y^2-r^2)/\nu^- \end{pmatrix}, & \mathbf{x} \in \Omega^-(t), \end{cases}$$

and

$$p(\mathbf{x}, t) = \frac{1}{10}(x^3 - y^3),$$

where

$$r(t) = r_0\sqrt{s \cdot \sin(2\pi t) + q},$$

with $r_0^2 = 0.3$, $s = 0.238$, $q = 2$. The viscosity coefficients are set to be $(\nu^+, \nu^-) = (10, 1)$.

The convergence rates are optimal which can be seen in Fig. 10. We also plot the velocity fields at times $t = 0, 0.3, 0.8$ in Fig. 11.

Example 6.6 (*Unsteady problem with moving interface: Deformation*) In this example, we test the case of interface deformation. The initial state of the interface is an ellipse. As time goes on, it becomes a circle and reaches equilibrium. The true solution of this example is set

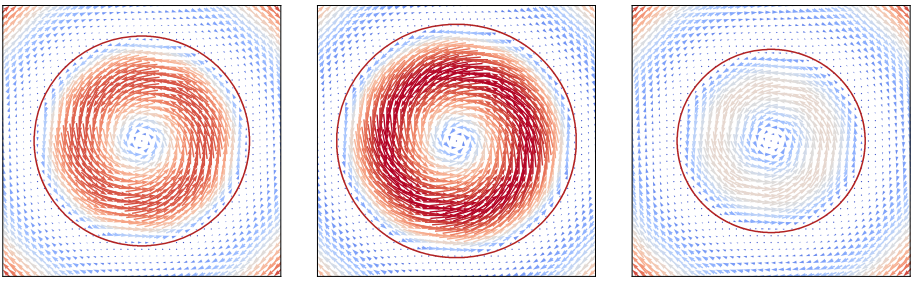

Fig. 11 The velocity fields of a stretching circular interface when $t = 0,\ 0.3,\ 0.8$ for Example 6.5

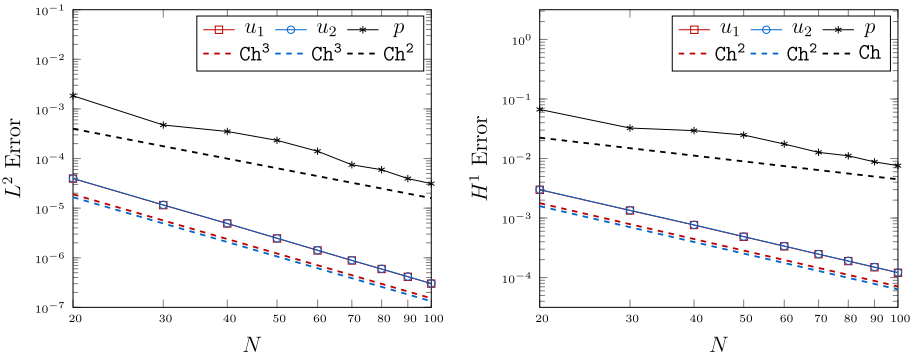

Fig. 12 BDF3-PPIFE error of deforming elliptical interface at $t = 1$ for Example 6.6

to be

$$\mathbf{u}(\mathbf{x}, t) = \begin{cases} \begin{pmatrix} k(t)y(x^2 + ky^2 - r^2)/\nu^+ \\ -x(x^2 + ky^2 - r^2)/\nu^+ \end{pmatrix}, \ \mathbf{x} \in \Omega^+(t), \\ \begin{pmatrix} k(t)y(x^2 + ky^2 - r^2)/\nu^- \\ -x(x^2 + ky^2 - r^2)/\nu^- \end{pmatrix}, \ \mathbf{x} \in \Omega^-(t), \end{cases}$$

and

$$p(\mathbf{x}, t) = \frac{1}{10}(x^3 - y^3)$$

where

$$k(t) = 1 - k_b e^{-k_a t}$$

with $r^2 = 0.3, k_a = 4, k_b = 0.41$. The viscosity coefficients are set to be $(\nu^+, \nu^-) = (10, 1)$.

The convergence plots are shown in Fig. 12, and the velocity fields at times $T = 0, 0.3, 1.0$ are shown in Fig. 13. Once again, we observe optimal convergence rates in this deformation case.

Example 6.7 (*Moving interface: Deformation of Flower-shape Interface*) In this example, we consider the relaxation of a flower-shape interface. The level-set function of the interface is

$$\phi(x, y, t) = (x^2 + y^2)[a + b(t)\sin(c \arctan(y/x))] - d, \tag{6.1}$$

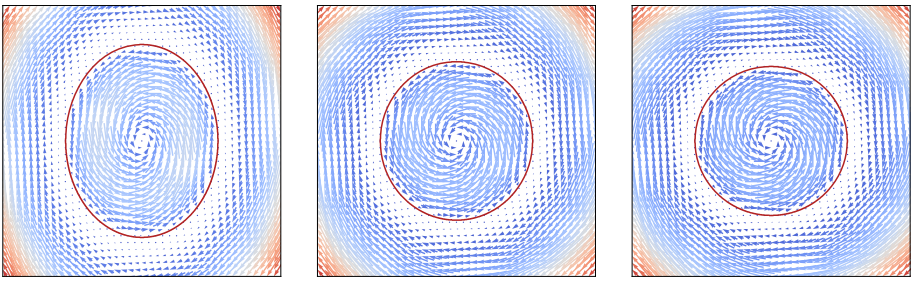

Fig. 13 Velocity field of deforming elliptical interface when $t = 0$, 0.3, and 1.0 for Example 6.6

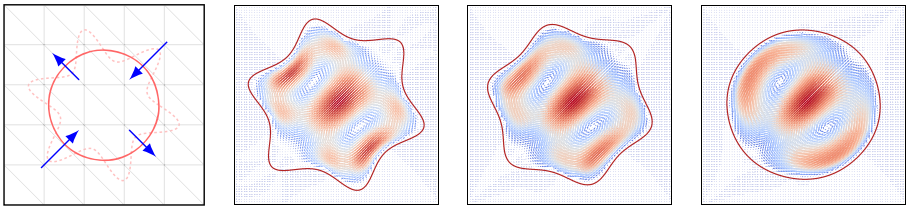

Fig. 14 The evolution of interface and the velocity field of deforming flower-shape interface when $t = 0.05$, 0.1, and 1.0 for Example 6.7 (Right three subfigures)

where $a = 1.0, c = 6, d = 0.3164$ are constants, and the deformation parameter $b(t) = 0.6e^{-4t}$. We define $\Omega^+(t) = \{(x, y) \in \Omega : \phi(x, y, t) > 0\}$ and $\Omega^-(t) = \{(x, y \in \Omega) : \phi(x, y, t) < 0\}$.

In previous examples, we test our scheme on problems with the analytic expression of interface available. We realize that this is relatively difficult in the real world scenarios. In this example, we restrict ourselves with interface is implicitly defined by level set function ϕ in (6.1). Moreover, we are only accessible to the function value of $\phi(x, y, t)$ on a set of grid points, for example $\{(x_i, y_j) : 0 \le i, j \le 100\} \times \{t_0, t_1, ..., t_n\}$, where $x_i = -1 + i/50$, $y_j = -1 + j/50$.

To handle this case, for each t_n fixed, we interpolate $\phi(x_i, y_j, t_n)$ using piecewise linear polynomials. This continuous level set function could be used to construct the immersed $\mathcal{P}_2$-$\mathcal{P}_1$ spaces. For numerical quadratures on curved polygons, we employ the technique introduced by [39].

When $t = 0$, the initial state of the interface is a flower, then it is relaxed to a circle as $b(t)$ decays exponentially to 0. See the left one in Fig. 14. It is hard to construct the exact solution for this example, so we directly solve this problem with homogenous Dirichlet boundary condition and initial conditions $\mathbf{u}_0(x, y, t) = \mathbf{0}$, $p_0(x, y, t) = (x^3 - y^3)/10$. It is also assumed that there exists an external force $\mathbf{f}(x, y, t) = 100(x^2 + y^2)$ in the system. We test our method with BDF2 time discretization on this example up to $t = 1.0$ with parameters with $\nu^+ = 1000$ and $\nu^- = 1$. The quiver plots of simulated velocity field at time points 0.05, 0.1, 1.0 are shown in the Fig. 14, while the deformation of velocity field can be observed with the evolution of interfaces.

7 Conclusion

In this paper, we develop a $\mathcal{P}_2$-$\mathcal{P}_1$ enhanced partially penalized immersed finite element method to solve Navier–Stokes moving interface problems. We establish the unisolvency of

the $\mathcal{P}_2$-$\mathcal{P}_1$ vector-valued IFE functions and the proved the partition of unity. Both θ-scheme and BDF schemes are utilized for the time discretization. The resulting nonlinear system is handled by the Newton method. Numerical simulations are carried out for both static and moving interface cases. Optimal convergence rates of our method are observed from multiple example with different viscosity jumps and various interface dynamics.

Acknowledgements Xu Zhang is partially supported by the National Science Foundation Grant DMS-2110833.

Funding Xu Zhang is partially supported by the National Science Foundation Grant DMS-2110833.

Data Availibility All data generated or analyzed during this study are included in this manuscript.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Adjerid, S., Chaabane, N., Lin, T.: An immersed discontinuous finite element method for stokes interface problems. Comput. Methods Appl. Mech. Eng. **293**, 170–190 (2015)
2. Adjerid, S., Chaabane, N., Lin, T., Yue, P.: An immersed discontinuous finite element method for the Stokes problem with a moving interface. J. Comput. Appl. Math. **362**, 540–559 (2019)
3. Adjerid, S., Guo, R., Lin, T.: High degree immersed finite element spaces by a least squares method. Int. J. Numer. Anal. Model. **14**(4–5), 604–626 (2017)
4. Adjerid, S., Lin, T., Zhuang, Q.: Error estimates for an immersed finite element method for second order hyperbolic equations in inhomogeneous media. J. Sci. Comput. **84**(2), 35 (2020)
5. Adjerid, S., Moon, K.: An immersed discontinuous Galerkin method for acoustic wave propagation in inhomogeneous media. SIAM J. Sci. Comput. **41**(1), A139–A162 (2019)
6. Anselmann, M., Bause, M.: Cut finite element methods and ghost stabilization techniques for space-time discretizations of the Navier–Stokes equations. Int. J. Numer. Methods Fluids **94**(7), 775–802 (2022)
7. Biesheuvel, A., Wijngaarden, L.: Two-phase flow equations for a dilute dispersion of gas bubbles in liquid. J. Fluid Mech. **148**, 301–18 (1984)
8. Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A.: CutFEM: discretizing geometry and partial differential equations. Int. J. Numer. Methods Eng. **104**(7), 472–501 (2015)
9. Burman, E., Hansbo, P.: Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem. ESAIM Math. Model. Numer. Anal. **48**(3), 859–874 (2014)
10. Cattaneo, L., Formaggia, L., Iori, G.F., Scotti, A., Zunino, P.: Stabilized extended finite elements for the approximation of saddle point problems with unfitted interfaces. Calcolo **52**(2), 123–152 (2015)
11. Chen, Y., Hou, S., Zhang, X.: A bilinear partially penalized immersed finite element method for elliptic interface problems with multi-domains and triple-junction points. Results Appl. Math. **8**, 100100 (2020)
12. Chen, Y., Zhang, X.: A P_2-P_1 partially penalized immersed finite element method for stokes interface problems. Int. J. Numer. Anal. Model. **18**(1), 120–141 (2021)
13. Claus, S., Kerfriden, P.: A cutfem method for two-phase flow problems. Comput. Methods Appl. Mech. Eng. **348**, 185–206 (2019)
14. Court, S., Fournié, M.: A fictitious domain finite element method for simulations of fluid-structure interactions: the Navier–Stokes equations coupled with a moving solid. J. Fluids Struct. **55**, 398–408 (2015)
15. Frachon, T., Zahedi, S.: A cut finite element method for incompressible two-phase Navier–Stokes flows. J. Comput. Phys. **384**, 77–98 (2019)
16. Fries, T.-P.: The intrinsic xfem for two-fluid flows. Int. J. Numer. Meth. Fluids **60**(4), 437–471 (2009)
17. Glowinski, R., Pan, T.-W., Periaux, J.: A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations. Comput. Methods Appl. Mech. Eng. **112**(1–4), 133–148 (1994)
18. Guo, R.: Solving parabolic moving interface problems with dynamical immersed spaces on unfitted meshes: fully discrete analysis. SIAM J. Numer. Anal. **59**(2), 797–828 (2021)
19. Guo, R., Lin, T.: A higher degree immersed finite element method based on a Cauchy extension for elliptic interface problems. SIAM J. Numer. Anal. **57**(4), 1545–1573 (2019)
20. Guo, R., Lin, T., Lin, Y.: Error estimates for a partially penalized immersed finite element method for elasticity interface problems. ESAIM Math. Model. Numer. Anal. **54**(1), 1–24 (2020)

21. Guo, R., Lin, T., Zhuang, Q.: Improved error estimation for the partially penalized immersed finite element methods for elliptic interface problems. Int. J. Numer. Anal. Model. **16**(4), 575–589 (2019)
22. Guo, R., Zhang, X.: Solving three-dimensional interface problems with immersed finite elements: a-priori error analysis. J. Comput. Phys. **441**, 110445 (2021)
23. Guzmán, J., Olshanskii, M.: Inf-sup stability of geometrically unfitted Stokes finite elements. Math. Comput. **87**(313), 2091–2112 (2018)
24. He, X., Lin, T., Lin, Y., Zhang, X.: Immersed finite element methods for parabolic equations with moving interface. Numer. Methods Partial Differ. Equ. **29**(2), 619–646 (2013)
25. Huang, J., Carrica, P.M., Stern, F.: Coupled ghost fluid/two-phase level set method for curvilinear body-fitted grids. Int. J. Numer. Methods Fluids **55**(9), 867–897 (2007)
26. Ji, H., Wang, F., Chen, J., Li, Z.: An immersed CR-P0 element for Stokes interface problems and the optimal convergence analysis. Comput. Methods Appl. Mech. Eng. **399**, 115306 (2022)
27. John, V.: A numerical study of a posteriori error estimators for convection-diffusion equations. Comput. Methods Appl. Mech. Eng. **190**(5–7), 757–781 (2000)
28. Jones, D., Zhang, X.: A class of nonconforming immersed finite element methods for Stokes interface problems. J. Comput. Appl. Math. **392**, 113493 (2021)
29. Jones, D., Zhang, X.: A conforming-nonconforming mixed immersed finite element method for unsteady stokes equations with moving interfaces. Electron. Res. Arch. (2021). https://doi.org/10.3934/era.2021032
30. Jones, D.T.: A Class of Immersed Finite Element Methods for Stokes Interface Problems. ProQuest LLC, Ann Arbor. Ph.D., thesis, Mississippi State University (2021)
31. Lee, L., LeVeque, R.J.: An immersed interface method for incompressible Navier–Stokes equations. SIAM J. Sci. Comput. **25**(3), 832–856 (2003)
32. Li, Z., Lai, M.-C.: The immersed interface method for the Navier–Stokes equations with singular forces. J. Comput. Phys. **171**(2), 822–842 (2001)
33. Lin, T., Lin, Y., Zhang, X.: A method of lines based on immersed finite elements for parabolic moving interface problems. Adv. Appl. Math. Mech. **5**(4), 548–568 (2013)
34. Lin, T., Lin, Y., Zhang, X.: Partially penalized immersed finite element methods for elliptic interface problems. SIAM J. Numer. Anal. **53**(2), 1121–1144 (2015)
35. Lin, T., Sheen, D., Zhang, X.: A locking-free immersed finite element method for planar elasticity interface problems. J. Comput. Phys. **247**, 228–247 (2013)
36. Lin, T., Zhang, X.: Linear and bilinear immersed finite elements for planar elasticity interface problems. J. Comput. Appl. Math. **236**(18), 4681–4699 (2012)
37. Nagheeby, M., Kolahdoozan, M.: Numerical modeling of two-phase fluid flow and oil slick transport in estuarine water. Int. J. Environ. Sci. Technol. **7**(4), 771–784 (2010)
38. Sauerland, H., Fries, T.-P.: The stable xfem for two-phase flows. Comput. Fluids **87**, 41–49 (2013)
39. Saye, R.I.: High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. SIAM J. Sci. Comput. **37**(2), A993–A1019 (2015)
40. Schott, B., Wall, W.A.: A new face-oriented stabilized xfem approach for 2d and 3d incompressible Navier–Stokes equations. Comput. Methods Appl. Mech. Eng. **276**, 233–265 (2014)
41. Sharan, M., Popel, A.S.: A two-phase model for flow of blood in narrow tubes with increased effective viscosity near the wall. Biorheology **38**(5, 6), 415–428 (2001)
42. Tan, Z., Le, D.-V., Lim, K.M., Khoo, B.C.: An immersed interface method for the incompressible Navier–Stokes equations with discontinuous viscosity across the interface. SIAM J. Sci. Comput. **31**(3), 1798–1819 (2009)
43. Vukčević, V., Jasak, H., Gatin, I.: Implementation of the ghost fluid method for free surface flows in polyhedral finite volume framework. Comput. Fluids **153**, 1–19 (2017)
44. Wang, J., Zhang, X., Zhuang, Q.: An immersed Crouzeix–Raviart finite element method for Navier–Stokes interface problems. Int. J. Numer. Anal. Model. **19**(4), 563–586 (2022)
45. Zhuang, Q., Guo, R.: High degree discontinuous Petrov–Galerkin immersed finite element methods using fictitious elements for elliptic interface problems. J. Comput. Appl. Math. **362**, 560–573 (2019)