

# An energy-stable scheme for a 2D simple fluid-particle interaction problem <sup>☆</sup>

Xiang Li <sup>a,b</sup>, Qiang Du <sup>c</sup>, Li Luo <sup>d</sup>, Xiao-Ping Wang <sup>b,\*</sup>

<sup>a</sup> Applied and Computational Mathematics Division, Beijing Computational Science Research Center, Beijing, PR China

<sup>b</sup> Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong

<sup>c</sup> Department of Applied Physics and Applied Mathematics and Data Science Institute, Columbia University, New York, NY 10027, USA

<sup>d</sup> Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, PR China

## ARTICLE INFO

### Article history:

Received 29 March 2019

Received in revised form 1 June 2020

Accepted 12 September 2020

Available online 18 September 2020

### Keywords:

Fluid-particle interaction

Energy-stable scheme

Temporary arbitrary Lagrangian-Eulerian

Extended finite element method

## ABSTRACT

We develop an energy-stable scheme for simulating fluid-particle interaction problems governed by a coupled system consisting of the incompressible Navier-Stokes (NS) equations defined in a time-dependent fluid domain and Newton's second law for particle motion. A modified temporary arbitrary Lagrangian-Eulerian (tALE) method is designed based on a bijective mapping between the fluid regions at different time steps. In the proposed numerical scheme, the tALE mesh velocity, the incompressible NS equations, and Newton's second law are solved simultaneously. We prove that under certain conditions, the new time discretization scheme satisfies an energy law. For the space discretization, the extended finite element method (XFEM) is used to solve the problem on a fixed Cartesian mesh. The developed method is first-order accurate in time and space without being momentum conservative. To verify the accuracy and stability of our numerical scheme, we present numerical experiments including the fitting of the Jeffery orbit by rotating of an ellipse and the free-falling of an elliptic particle in water.

© 2020 Elsevier Inc. All rights reserved.

## 1. Introduction

Fluid-particle interaction plays an important role in scientific and engineering problems such as crude oil emulsions, fluidized suspensions and sedimentation, ship maneuvering, and underwater vehicles. In this paper, we consider interaction between a rigid body and the surrounding incompressible fluid. In the past decades, two categories of models have been proposed to describe the particle surface and the hydrodynamic interaction between the fluid and the particle, including the continuum approach [23,36,47,48] and the models using the direct numerical simulation (DNS) technique. The continuum approach views solid particles and fluids as interpenetrating mixtures with different viscosities that are governed by the laws of conservation. In this approach, the motion of particles and that of fluids are combined in the same mathemati-

<sup>☆</sup> **Funding:** The work of Xiang Li is partially supported by the NSFC program for Scientific Research Center under program No. U1530401. The work of Li Luo is partially supported by the NSFC program for Scientific Research Center under program No. 11701547. The work of Xiao-Ping Wang is partially supported by the Hong Kong Research Grants Council (GRF grants 16302715 and 16324416, CRF grant C6004-14G, and NSFC-RGC joint research grant N-HKUST620/15). The work of Qiang Du is partially supported by US National Science Foundation grant DMS-1719699 and DMS-2012562.

\* Corresponding author.

E-mail addresses: lixiang@csrcc.ac.cn (X. Li), qd2125@columbia.edu (Q. Du), li.luo@siat.ac.cn (L. Luo), mawang@ust.hk (X.-P. Wang).

cal framework to form a single system of equations and the boundary conditions are imposed implicitly on the particle surface. The continuum approach is flexible and can efficiently solve various kinds of fluid-particle interaction problems. However, the false response from the viscous material used to mimic the rigid objects might produce undesirable hydrodynamic effects in the case when the particle concentration is dense, or when particle-wall and particle-particle interactions are considered. Alternatively, the DNS technique treats the fluid and the particle separately and computes hydrodynamic interactions as a part of the solution. The boundary conditions that capture precisely the momentum and pressure response between the fluid and the rigid particle are imposed explicitly on the solid surfaces. The DNS technique gives a clear understanding of the mechanisms between the fluid and the particle and is well designed for many problems involving nonlinear and geometrically complicated phenomena.

For fluid-particle interaction simulations based on the DNS technique, the fluid domain is time-dependent as the particle moves inside the fluid. In order to track the particle and its surrounding fluid, several numerical simulation techniques have been developed, including the arbitrary Lagrangian-Eulerian (ALE) method, the temporary arbitrary Lagrangian-Eulerian (tALE) method, and the fictitious domain method. For the ALE method [21,20,22], a boundary fitted mesh is used to describe the fluid domain. The particle surface is aligned with element boundaries. The ALE mesh is updated after each time step using the ALE mapping. Moreover, when the mesh becomes too distorted, a new mesh must be generated. In order to simulate the fluid-particle problem on a fixed Eulerian mesh, the tALE method is proposed [10]. In the tALE approach, time derivatives in the fluid region are discretized by an inverse tALE mapping. The interface of the particle is approximated with the help of the extended finite element method (XFEM) [12,40], which is a generalization of the standard Galerkin finite element method with additional degrees of freedom near the particle surface. An alternative approach is the fictitious domain method developed by Glowinski et al. [15–17,13]. The basic idea of this approach is to extend the problem in a geometrically complex domain to a simple and regular domain by generalizing the weak formulation of flows from the fluid domain to a fictitious domain that represents the particles. This approach enforces the constraint of rigid-body motion with a distributed Lagrange multiplier to introduce an additional body force to the interior of the region of particles. The fictitious domain method is widely used to simulate fluid-particle interaction problems, including the interaction between the fluid and a large amount of rigid particles [14,41,42].

The main purpose of this paper is to develop an energy-stable scheme for the simulation of fluid-particle interaction problems. Energy stability becomes crucial in numerical simulations when large time step sizes are used and long-time behavior is simulated. Ensuring numerical stability for fluid-particle interaction is far from trivial. Wood et al. [45] indicated that numerical schemes based on sequential computation of fluid and particle dynamics are often unstable if there are no sub-iteration steps between computations of the fluid and the particle. To improve the stability, certain sub-iteration steps are performed between the two sets of computations [8,39,43]. Efforts have been made to study the stability of numerical methods such as the immersed boundary methods [26,27,38,46], the block-factorization methods [3,35], and the weak-coupling scheme using the summation-by-parts (SBP) operators [32,28]. The SBP-based schemes are proven to be energy-stable for both time and space discretization, and they have been applied successfully in fluid-structure interaction [31]. Other studies of the stable numerical schemes are also available [34,44]. Recently, a stable partitioned algorithm [4–6] has been presented to simulate the fluid-particle interaction problems with light or zero-mass rigid particles.

In this paper, we develop an energy-stable time discretization scheme for simulating the fluid-particle interaction problem. The energy-stable scheme is based on a DNS-type model. The incompressible Navier-Stokes equations and Newton's second law are used to describe the fluid and the particle motion respectively. To track the time-dependent fluid domain, our numerical scheme develops a bijective mapping based on the tALE mapping [10]. To ensure energy stability, the tALE mesh velocity, the fluid equations, and Newton's second law are solved implicitly and simultaneously. We prove that under a mild condition that is usually omitted in practical implementation, the new time discretization scheme satisfies a discrete energy law. The discrete energy law is similar to the energy law of the continuous system. For the discretization in space, we use the XFEM [12,40] to handle discontinuities near the particle surface. Only a fixed Eulerian mesh is required for the entire computation, without the need to regenerate or deform the meshes.

Numerical tests are carried out to verify the stability and accuracy of the energy-stable scheme. We simulate the Jeffery orbit of a fixed ellipse rotating in a shear flow and compare the results with those obtained from the standard tALE-XFEM method [49]. It is shown that our energy-stable scheme allows the usage of large time step sizes such as  $\delta t = \delta x$ , while the standard tALE-XFEM method soon blows up even with a much smaller time step size. Moreover, our energy-stable scheme remains stable in long time simulation. Tests show that our algorithm is first-order accurate in both time and space. We also simulate a thin ellipse sinking in water which displays fluttering and tumbling motions. The proposed numerical scheme is shown to be energy-stable for both cases.

The rest of the paper is organized as follows. In Section 2, we introduce the governing equations and derive the energy law for the continuous system. In Section 3 we introduce the tALE method with a bijective tALE mapping between the different time steps. The energy-stable scheme is developed in Section 4. The discrete energy law of the time discretization scheme is also proved in this section. Implementation details of the proposed scheme are described in Section 5. Numerical experiments are presented in Section 6 to support the theoretical results. The paper is concluded in Section 7.

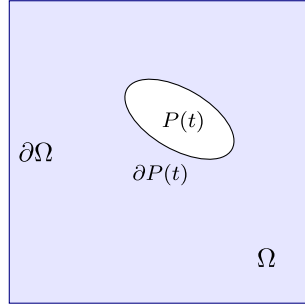


Fig. 1. Fluid-particle interaction problem.

## 2. Model of the fluid-particle system

### 2.1. Governing equations

We consider the motion of a solid particle in the fluid, as illustrated in Fig. 1. Let  $\Omega$  be the entire computational domain with boundary  $\partial\Omega$ . Let  $P(t)$  be the interior of the solid particle, and let  $\partial P(t)$  denote its surface. Equations (2.1)–(2.6) are the governing equations of the fluid-particle system:

The fluid motion is governed by the incompressible NS equations in  $\Omega \setminus P(t)$ :

$$\mathcal{Re} \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}, \quad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (2.2)$$

Here  $\rho$  stands for the fluid mass density and  $\eta$  is the fluid viscosity coefficient.  $\mathbf{u}$  is the vector of fluid velocity,  $p$  stands for the pressure, and  $\mathbf{f}$  denotes the external force on the fluid. For incompressible flow, tensor  $\boldsymbol{\sigma}$  is given by  $\boldsymbol{\sigma} := \nabla \mathbf{u}$ . Parameter  $\mathcal{Re}$  is the particle Reynolds number, which is given by  $\mathcal{Re} := \frac{\rho V_0 L_0}{\eta}$ .  $V_0$  and  $L_0$  are the characteristic velocity and characteristic length of the particle, which are the particle diameter and the particle maximal velocity norm, respectively.

The motion of the solid particle is given by Newton's law:

$$\mathcal{Re} M_s \frac{d\mathbf{U}_s}{dt} = - \int_{\partial P(t)} (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n} dS + \mathbf{F}_s, \quad (2.3)$$

$$\mathcal{Re} (I_s \frac{d\boldsymbol{\omega}_s}{dt} + \boldsymbol{\omega}_s \times I_s \boldsymbol{\omega}_s) = - \int_{\partial P(t)} \mathbf{r} \times ((-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n}) dS. \quad (2.4)$$

Here  $\mathbf{U}_s$  is the velocity of the particle's center of gravity,  $\boldsymbol{\omega}_s$  is the particle angular velocity,  $\mathbf{I}$  denotes the unit matrix,  $\mathbf{n}$  is the inner normal of  $P(t)$  on its surface (thus it is the outer normal of  $\Omega \setminus P(t)$  on  $\partial P(t)$ ),  $\mathbf{r} = \mathbf{x} - \mathbf{X}_s$  is the vector from the particle's gravity center  $\mathbf{X}_s$  to a point  $\mathbf{x}$  on the particle surface, and  $\mathbf{F}_s$  is the external force on the particle. In this model, the particle is assumed to be rigid and is made of a homogeneous material of density  $\rho_s$ . Dimensionless mass  $M_s$  and inertia  $I_s$  of the particle are defined as:

$$\begin{aligned} M_s &:= (\rho_s / \rho) \int_{P(t)} dx, \\ I_s &:= (\rho_s / \rho) \int_{P(t)} |\mathbf{r}|^2 dx \quad (2D \text{ case}), \\ I_s &:= (\rho_s / \rho) \int_{P(t)} (|\mathbf{r}|^2 \mathbf{I} - \mathbf{r} \otimes \mathbf{r}) dx \quad (3D \text{ case}), \end{aligned}$$

where  $\mathbf{r} \otimes \mathbf{r}$  denotes the Kronecker product of  $\mathbf{r}$  and  $\mathbf{r}^T$ . If the fluid-particle motion occurs in 2D, or if the particle is a spherical body made of a homogeneous material, the nonlinear term  $\boldsymbol{\omega}_s \times I_s \boldsymbol{\omega}_s$  in (2.4) becomes zero.

The boundary conditions on  $\partial P(t)$  are

$$\frac{1}{L_s} (\mathbf{u} - \mathbf{u}_s) \cdot \boldsymbol{\tau} = -(\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}, \quad (\mathbf{u} - \mathbf{u}_s) \cdot \mathbf{n} = 0. \quad (2.5)$$

Here  $\mathbf{u}_s = \mathbf{U}_s + \boldsymbol{\omega}_s \times \mathbf{r}$ , and  $\boldsymbol{\tau}$  is the tangent vector on  $\partial P(t)$ ,  $L_s$  is the slip length coefficient.

The boundary conditions on  $\partial\Omega$  are

$$\frac{1}{L_s}(\mathbf{u} - \mathbf{u}_w) \cdot \boldsymbol{\tau} = -(\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}, \quad \mathbf{u} \cdot \mathbf{n} = 0. \quad (2.6)$$

Here  $\mathbf{n}$  is the outer normal of  $\Omega$ , and  $\mathbf{u}_w$  denotes the wall speed whose direction is parallel to  $\partial\Omega$ .

The total energy  $E$  of the governing system consists of the kinetic energy of the fluid and the particle, which is defined as:

$$E := \mathcal{R}e E_k + \mathcal{R}e M_s E_{pm} + \mathcal{R}e I_s E_{pr}, \quad (2.7)$$

where

$$E_k := \frac{1}{2} \int_{\Omega \setminus P(t)} |\mathbf{u}|^2 dx, \quad E_{pm} := \frac{1}{2} |\mathbf{U}_s|^2, \quad E_{pr} := \frac{1}{2} |\boldsymbol{\omega}_s|^2.$$

## 2.2. Energy law of the governing system

In this section, we derive the energy law for the governing system. For simplicity we assume the wall velocity  $\mathbf{u}_w = 0$ . We denote  $(F, G)_S$  as the integral in the region  $S$  of the product, inner product, or Kronecker product of  $F$  and  $G$ , when  $F$  and  $G$  are scalars, vectors, or matrices respectively.  $\|F\|_S$  denotes the  $L^2$  norms of  $F$ . Subscript  $S$  is omitted when  $S$  is chosen as  $\Omega \setminus P(t)$ .  $\langle F, G \rangle_{\partial S}$  denotes the corresponding boundary integral and  $\|F\|_{\partial S}$  the corresponding  $L^2$  norm on  $\partial S$ . The following theorem gives the energy law of the continuous system:

**Theorem 2.1** (Energy law of the continuous system). Assume that  $\mathbf{u}_w = 0$ . The governing equations (2.1) - (2.6) satisfy the following energy law:

$$\frac{dE}{dt} = -\|\nabla \mathbf{u}\|^2 - \frac{1}{L_s} \|\mathbf{u} \cdot \boldsymbol{\tau}\|_{\partial\Omega}^2 - \frac{1}{L_s} \|(\mathbf{u} - \mathbf{u}_s) \cdot \boldsymbol{\tau}\|_{\partial P(t)}^2 + (\mathbf{f}, \mathbf{u}) + \mathbf{F}_s \cdot \mathbf{U}_s. \quad (2.8)$$

**Proof.** Using Green's formula and incompressibility equation (2.2), we obtain

$$\begin{aligned} \frac{dE_k}{dt} &= \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{u} \right) \\ &= \left( \frac{\partial \mathbf{u}}{\partial t}, \mathbf{u} \right) + \langle \frac{1}{2} |\mathbf{u}|^2 (\mathbf{u} \cdot \mathbf{n}), \mathbf{u} \rangle_{\partial\Omega} + \langle \frac{1}{2} |\mathbf{u}|^2 (\mathbf{u} \cdot \mathbf{n}), \mathbf{u} \rangle_{\partial P(t)}. \end{aligned} \quad (2.9)$$

Taking inner product of (2.1) with  $\mathbf{u}$  and using Green's formula and (2.2), we obtain

$$\begin{aligned} \mathcal{R}e \frac{dE_k}{dt} &= -(\boldsymbol{\sigma}, \nabla \mathbf{u}) + \langle \boldsymbol{\sigma} \cdot \mathbf{n}, \mathbf{u} \rangle_{\partial\Omega} + \langle \boldsymbol{\sigma} \cdot \mathbf{n}, \mathbf{u} \rangle_{\partial P(t)} - \langle p, \mathbf{u} \cdot \mathbf{n} \rangle_{\partial\Omega} - \langle p, \mathbf{u} \cdot \mathbf{n} \rangle_{\partial P(t)} + (\mathbf{f}, \mathbf{u}) \\ &= -\|\nabla \mathbf{u}\|^2 + (\mathbf{f}, \mathbf{u}) + \langle (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n}, \mathbf{u} \rangle_{\partial\Omega} + \langle (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n}, \mathbf{u} \rangle_{\partial P(t)}. \end{aligned} \quad (2.10)$$

Taking the inner product of (2.3) with  $\mathbf{U}_s$  and the inner product of (2.4) with  $\boldsymbol{\omega}_s$ , then summing up these two equations, we obtain

$$\frac{d}{dt} (\mathcal{R}e M_s E_{pm} + \mathcal{R}e I_s E_{pr}) = \mathcal{R}e M_s \mathbf{U}_s \cdot \frac{d\mathbf{U}_s}{dt} + \mathcal{R}e I_s \boldsymbol{\omega}_s \cdot \frac{d\boldsymbol{\omega}_s}{dt} = -\langle (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n}, \mathbf{u}_s \rangle_{\partial P(t)} + \mathbf{F}_s \cdot \mathbf{U}_s. \quad (2.11)$$

Summing (2.10) and (2.11) and using the boundary conditions (2.5) and (2.6), we obtain

$$\begin{aligned} \frac{dF}{dt} &= -\|\nabla \mathbf{u}\|^2 + \langle (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n}, \mathbf{u} \rangle_{\partial\Omega} + \langle (-p\mathbf{I} + \boldsymbol{\sigma}) \cdot \mathbf{n}, \mathbf{u} - \mathbf{u}_s \rangle_{\partial P(t)} + (\mathbf{f}, \mathbf{u}) + \mathbf{F}_s \cdot \mathbf{U}_s \\ &= -\|\nabla \mathbf{u}\|^2 + \langle (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}, \mathbf{u} \cdot \boldsymbol{\tau} \rangle_{\partial\Omega} + \langle (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}, (\mathbf{u} - \mathbf{u}_s) \cdot \boldsymbol{\tau} \rangle_{\partial P(t)} + (\mathbf{f}, \mathbf{u}) + \mathbf{F}_s \cdot \mathbf{U}_s \\ &= -\|\nabla \mathbf{u}\|^2 - \frac{1}{L_s} \|\mathbf{u} \cdot \boldsymbol{\tau}\|_{\partial\Omega}^2 - \frac{1}{L_s} \|(\mathbf{u} - \mathbf{u}_s) \cdot \boldsymbol{\tau}\|_{\partial P(t)}^2 + (\mathbf{f}, \mathbf{u}) + \mathbf{F}_s \cdot \mathbf{U}_s. \quad \square \end{aligned}$$

**Remark 2.1.** If the wall velocity  $\mathbf{u}_w \neq 0$ , we introduce  $W(t)$  as the accumulated work done by the shear force on the wall,

$$W(t) := \int_0^t w(\tilde{t}) d\tilde{t}, \quad w(t) := -\langle (\boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n}) \cdot \boldsymbol{\tau}, \mathbf{u}_w(t) \cdot \boldsymbol{\tau} \rangle_{\partial\Omega}. \quad (2.12)$$

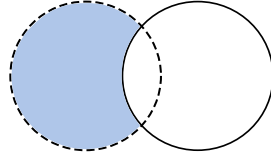


Fig. 2. Particle motion from time step  $t^n$  (dashed line) to  $t^{n+1}$  (solid line).

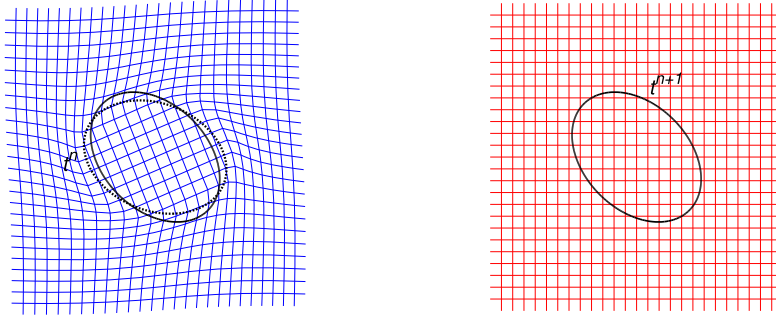


Fig. 3. Computational mesh at  $t^{n+1}$  (right) and the fictitious tALE mesh constructed at  $t^n$  (left). The ellipse at  $t^{n+1}$  is shown by a solid line and the ellipse at  $t^n$  is shown by a dotted line.

Then the energy law becomes

$$\frac{d}{dt}(E + W) = -\|\nabla \mathbf{u}\|^2 - \frac{1}{L_s} \|(\mathbf{u} - \mathbf{u}_w) \cdot \boldsymbol{\tau}\|_{\partial\Omega}^2 - \frac{1}{L_s} \|(\mathbf{u} - \mathbf{u}_s) \cdot \boldsymbol{\tau}\|_{\partial P(t)}^2 + (\mathbf{f}, \mathbf{u}) + \mathbf{F}_s \cdot \mathbf{U}_s.$$

**Remark 2.2.** The governing system conserves mass since the fluid is incompressible and the particle is rigid. If there is no external force acting on the fluid and the particle, the fluid-particle system (2.1) - (2.6) conserves momentum, which could be derived by combining the momentum equations for the fluid and the particle.

### 3. Time discretization with a bijective tALE mapping

We will construct an energy-stable time discretization based on the tALE method. Before introducing our tALE time discretization, we first review the standard ALE and tALE methods. Then we construct a tALE-based time discretization with a bijective mapping in 2D and in 3D.

#### 3.1. A brief review of the ALE and tALE methods

As a particle moves, field variables near its boundary at the previous time step  $t^n$  may become undefined at time  $t^{n+1}$  since there may not be any fluid flow (as shown by the dashed region in Fig. 2). One way to resolve this issue is to use the ALE method. In the ALE method, each mesh point moves at a mesh velocity so that it can track the motion of the particle. For instance, for a point  $\mathbf{x}^{n+1}$  at time  $t^{n+1}$ , we map this point to a point  $\mathbf{x}^n$  at the previous time step  $t^n$  by

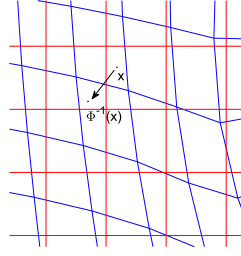
$$\mathbf{x}^{n+1} = B(\mathbf{x}^n) := \mathbf{x}^n + \delta t \bar{\mathbf{u}}^n(\mathbf{x}^n).$$

Here  $\bar{\mathbf{u}}^n$  is the mesh velocity field of the ALE method. An example of the ALE mesh motion is shown in Fig. 3. However, the ALE mesh may become too distorted after a few time steps. In this case, mesh regeneration is needed.

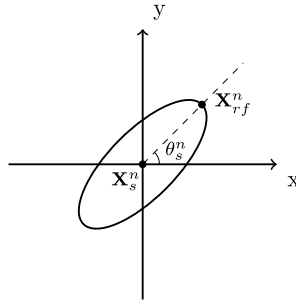
To solve the system on a fixed Eulerian mesh, Choi et al. [10] introduced the tALE method. Different from the ALE method, mesh points in the tALE method do not move. For the time discretization of the tALE method, we find the previous position of the fixed mesh points by an inverse flow mapping  $\tilde{B}$  (see Fig. 4) such that

$$\mathbf{x}^n = \tilde{B}(\mathbf{x}^{n+1}) := \mathbf{x}^{n+1} - \delta t \bar{\mathbf{u}}^{n+1}(\mathbf{x}^{n+1}).$$

Here  $\bar{\mathbf{u}}^{n+1}$  is the mesh velocity field of the tALE method. The values of the physical quantities at such points in the previous time step are calculated by interpolation. Unlike the ALE method, the tALE method maps the fixed mesh at the current time step, to a fictitious mesh at the previous time step (see Fig. 3). Mesh regeneration is not needed since only a fixed mesh is used in the practical implementation. Similar to the ALE method, the inverse mapping may be constructed from “mesh velocity”. However, the “mesh velocity” in tALE is only used to find the previous position of a mesh point, thus it does not change the mesh in the calculation.



**Fig. 4.** Inverse mapping  $\Phi^{-1}$  between the computational mesh and the tALE mesh. The mesh in red is the fixed mesh in computation, while the mesh in blue is the tALE fictitious mesh in the previous time step. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)



**Fig. 5.** Illustration of  $(\mathbf{X}_s^n, \theta_s^n)$  for an ellipse.

For both theoretical analysis and practical implementation of the tALE method, one of the key issues is to ensure the bijectivity of the fluid region at different time steps. Without such bijectivity, the position of a point inside the fluid in the current time step might be tracked to a position inside the particle or outside the entire domain in the previous time step. Such poor tracking will make the algorithm not well-posed. Moreover, such bijectivity is crucial to the stability of the numerical scheme. The issue of bijectivity also appears in the standard ALE analysis [30].

**Remark 3.1.** The tALE method allows discretization on a fixed mesh, and so it is less costly compared to the standard ALE method. However, one of the drawbacks of the tALE method is that it cannot be written in a conservative form in a discrete setting.

### 3.2. Bijective tALE mapping: 2D case

For our model problem, we show how to construct a bijective mapping between time steps  $t^n$  and  $t^{n+1}$ . We start from a simple case where the particle does not rotate. We show that the ALE/tALE method can form a bijective mapping. However, the ALE/tALE method fails in bijectivity as soon as the particle starts to rotate. Then we introduce our bijective map to resolve this problem.

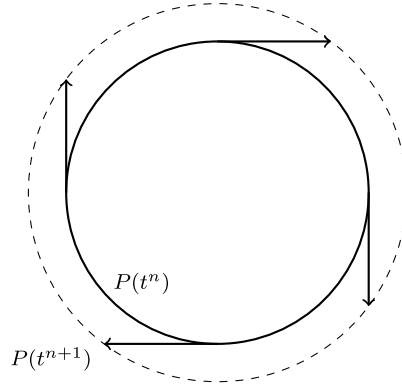
Assume that the particle is circular and does not rotate. Denote by  $\mathbf{X}_s^n$  and  $\theta_s^n$  the position of the particle's center of gravity and the angular orientation of the particle respectively. Here the angular orientation of the particle is defined by the angle from the x axis to a particular half-line. The half line starts from the particle's center of gravity and goes through a fixed reference point on the particle interface. Fig. 5 shows an example of  $(\mathbf{X}_s^n, \theta_s^n)$  for an ellipse at  $t = t^n$ , where the fixed reference point is positioned at  $\mathbf{X}_{r,f}^n$ . Using the ALE method for example, the bijective mapping  $(\mathbf{x}^n, \mathbf{X}_s^n) \rightarrow (\mathbf{x}^{n+1}, \mathbf{X}_s^{n+1})$  can be constructed in a simple way:

$$\begin{cases} \mathbf{x}^{n+1} = \mathbf{x}^n + \delta t \bar{\mathbf{u}}^n(\mathbf{x}^n), \\ \mathbf{X}_s^{n+1} = \mathbf{X}_s^n + \delta t \mathbf{U}_s^n, \end{cases}$$

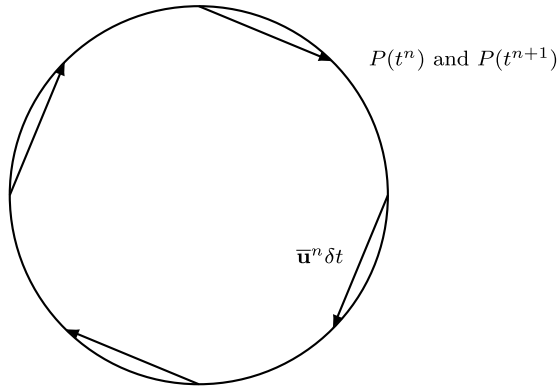
where the tALE mesh velocity  $\bar{\mathbf{u}}^n$  is given by

$$\begin{cases} -\Delta \bar{\mathbf{u}}^n = 0, & \text{in } \Omega \setminus P(t^n), \\ \bar{\mathbf{u}}^n = \mathbf{U}_s^n, & \text{on } \partial P(t^n), \\ \bar{\mathbf{u}}^n = 0, & \text{on } \partial \Omega. \end{cases}$$

Denote  $B(\mathbf{x}^n) = \mathbf{x}^n + \delta t \bar{\mathbf{u}}^n(\mathbf{x}^n)$ . Assume that  $\bar{\mathbf{u}}^n$  is Lipschitz continuous, that is,



**Fig. 6.** Example of non-bijective mapping between different time steps. The solid circle is  $P(t^n)$ , which is rotating. The dashed circle is  $P(t^{n+1})$ , which is the particle boundary at  $t = t^{n+1}$  calculated by the tALE mapping using (3.1).



**Fig. 7.** Example of bijective mapping between different time steps. The solid line circle is both  $P(t^n)$  and  $P(t^{n+1})$ . The modified tALE mapping is shown by arrows.

$$|\bar{\mathbf{u}}^n(\mathbf{x}_2) - \bar{\mathbf{u}}^n(\mathbf{x}_1)| < L|\mathbf{x}_2 - \mathbf{x}_1|, \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \Omega \setminus P(t^n), \quad \mathbf{x}_1 \neq \mathbf{x}_2.$$

Then the discrete mapping  $B$  is a bijection from  $\Omega \setminus P(t^n)$  to  $\Omega \setminus P(t^{n+1})$ , for all  $\delta t < 1/L$ , since (i) the mapping on particle surface is bijective, (ii) the mapping and inverse mapping are continuous. However, when the particle rotates, we cannot construct a bijective mapping using the ALE mesh velocity field.

To further illustrate this, for an arbitrary point with coordinate  $\mathbf{x}^n$  on the particle boundary,  $\mathbf{r}^n$  is defined by  $\mathbf{r}^n := \mathbf{x}^n - \mathbf{X}_s^n$ . If we define a mapping  $B$  using the above tALE velocity field, the tangential velocity on the particle boundary will distort the particle surface. We show an example in Fig. 6, where the particle is a rotating sphere with a fixed center of gravity. In this example, since mesh velocity on the sphere surface is in the tangential direction of the sphere, for the mapping  $B(\mathbf{x}^n) = \mathbf{x}^n + \delta t \bar{\mathbf{u}}^n(\mathbf{x}^n)$ , a point on the particle surface at time  $t^n$  will not be mapped onto the particle surface at time  $t^{n+1}$ . The same problem also occurs in the tALE method since the mesh velocity field is similarly derived.

$$\begin{cases} -\Delta \bar{\mathbf{u}}^n = 0, & \text{in } \Omega \setminus P(t^n), \\ \bar{\mathbf{u}}^n = \mathbf{U}_s^n + \boldsymbol{\omega}_s^n \times \mathbf{r}^n, & \text{on } \partial P(t^n), \\ \bar{\mathbf{u}}^n = 0, & \text{on } \partial \Omega. \end{cases} \quad (3.1)$$

In order to resolve this issue, a modified mesh velocity field needs to be designed. Denote by  $\mathbf{x}^{n+1}$  the position of an arbitrary point on the particle surface at time  $t^{n+1}$ , and denote by  $\mathbf{X}_s^{n+1}$  the position of the center of gravity of the particle at time  $t^{n+1}$ . Define  $\mathbf{r}^{n+1} := \mathbf{x}^{n+1} - \mathbf{X}_s^{n+1}$ . Denote by  $\theta^{n+1}$  and  $\theta^n$  the angular orientation of  $\mathbf{x}^{n+1}$  and  $\mathbf{x}^n$  with respect to the particle's center of gravity. In order to ensure bijectivity, we need to map each point  $\mathbf{x}^n = \mathbf{X}_s^n + (|\mathbf{r}^n| \cos \theta^n, |\mathbf{r}^n| \sin \theta^n)$  to  $\mathbf{x}^{n+1} = \mathbf{X}_s^{n+1} + (|\mathbf{r}^{n+1}| \cos \theta^{n+1}, |\mathbf{r}^{n+1}| \sin \theta^{n+1})$ . This is done by developing the boundary condition of the tALE mesh velocity field by  $\bar{\mathbf{u}}^n = (\mathbf{x}^{n+1} - \mathbf{x}^n)/\delta t$  for each point  $\mathbf{x}^n$  on  $\partial P(t^n)$ . We can see from the same example that the modified tALE mesh velocity helps maintain the shape of the particle (see Fig. 7). Note that  $|\mathbf{r}^n| = |\mathbf{r}^{n+1}|$  for each pair of  $\mathbf{x}^n$  and  $\mathbf{x}^{n+1}$  above. We have

$$\mathbf{x}^{n+1} - \mathbf{x}^n = \mathbf{U}_s^n \delta t + |\mathbf{r}^n|(\cos(\theta^n + \omega_s^n \delta t) - \cos \theta^n, \sin(\theta^n + \omega_s^n \delta t) - \sin \theta^n).$$

Let  $\boldsymbol{\omega}_s^n = (0, 0, \omega_{s,z}^n)$ . If the angular velocity of the particle is not 0, define

$$\mathbf{r}_f^n := (\omega_{s,z}^n \delta t)^{-1} |\mathbf{r}^n| (\sin(\theta^n + \omega_s^n \delta t) - \sin \theta^n, -(\cos(\theta^n + \omega_s^n \delta t) - \cos \theta^n), 0),$$

otherwise let  $\mathbf{r}_f^n := \mathbf{r}^n$  (this can also be obtained by letting  $\omega_s \rightarrow 0$ ). Using Taylor expansion, we can see that  $\mathbf{r}_f^n$  is a first-order approximation of  $\mathbf{r}^n$  w.r.t.  $\delta t$ . The boundary condition of the mesh velocity field can be written as

$$\bar{\mathbf{u}}^n = (\mathbf{x}^{n+1} - \mathbf{x}^n)/\delta t = \mathbf{U}_s^n + \boldsymbol{\omega}_s^n \times \mathbf{r}_f^n \text{ on } \partial P(t). \quad (3.2)$$

The modified tALE velocity is obtained by solving the following equations:

$$\begin{cases} -\Delta \bar{\mathbf{u}}^n = 0, & \text{in } \Omega \setminus P(t^n), \\ \bar{\mathbf{u}}^n = \mathbf{U}_s^n + \boldsymbol{\omega}_s^n \times \mathbf{r}_f^n, & \text{on } \partial P(t^n), \\ \bar{\mathbf{u}}^n = 0, & \text{on } \partial \Omega. \end{cases} \quad (3.3)$$

Using the modified tALE velocity field  $\bar{\mathbf{u}}^n$ , we can construct a bijective mapping from the time step  $t^n$  to the time step  $t^{n+1}$ :

$$\begin{cases} \mathbf{x}^{n+1} = \mathbf{x}^n + \delta t \bar{\mathbf{u}}^n(\mathbf{x}^n), \\ \mathbf{X}_s^{n+1} = \mathbf{X}_s^n + \delta t \mathbf{U}_s^n, \\ \theta_s^{n+1} = \theta_s^n + \delta t \omega_s^n. \end{cases} \quad (3.4)$$

Define a mapping  $B^n : \Omega \setminus P(t^n) \rightarrow \Omega \setminus P(t^{n+1})$  such that  $B^n(\mathbf{x}^n) = \mathbf{x}^n + \delta t \bar{\mathbf{u}}^n(\mathbf{x}^n)$ , where  $\bar{\mathbf{u}}^n$  is given by equation (3.3). For all  $\delta t < 1/L$ , where  $L$  is the Lipschitz constant of  $\bar{\mathbf{u}}^n$ , such discrete mapping  $B^n$  is bijective from  $\Omega \setminus P(t^n)$  to  $\Omega \setminus P(t^{n+1})$ , since (i) the mapping on the particle surface is bijective, (ii) the mapping and inverse mapping is continuous. Moreover, for all functions  $f^n$  defined in  $\Omega \setminus P(t^n)$ , we can define the corresponding function  $f_\star^n$  by

$$f_\star^n(\mathbf{x}^{n+1}) := f^n((B^n)^{-1} \mathbf{x}^{n+1}), \quad \forall \mathbf{x}^{n+1} \in \Omega \setminus P(t^{n+1}). \quad (3.5)$$

It is easy to see that

$$\int_{\Omega \setminus P(t^n)} f^n(\mathbf{x}^n) d\mathbf{x}^n = \int_{\Omega \setminus P(t^{n+1})} f_\star^n(\mathbf{x}^{n+1}) (J^n)^{-1} d\mathbf{x}^{n+1} \quad (3.6)$$

where  $J^n$  the Jacobian of  $B$ .

Similarly, the bijective mesh could also be constructed from  $t^{n+1}$  to  $t^n$ :

$$\begin{cases} \mathbf{x}^n = \mathbf{x}^{n+1} - \delta t \bar{\mathbf{u}}^{n+1}(\mathbf{x}^{n+1}), \\ \mathbf{X}_s^n = \mathbf{X}_s^{n+1} - \delta t \mathbf{U}_s^{n+1}, \\ \theta_s^n = \theta_s^{n+1} - \delta t \omega_s^{n+1}, \end{cases} \quad (3.7)$$

where

$$\begin{cases} -\Delta \bar{\mathbf{u}}^{n+1} = 0, & \text{in } \Omega \setminus P(t^{n+1}), \\ \bar{\mathbf{u}}^{n+1} = \mathbf{U}_s^{n+1} + \boldsymbol{\omega}_s^{n+1} \times \mathbf{r}_f^{n+1}, & \text{on } \partial P(t^{n+1}), \\ \bar{\mathbf{u}}^{n+1} = 0, & \text{on } \partial \Omega. \end{cases} \quad (3.8)$$

$$\mathbf{r}_f^{n+1} := (\omega_{s,z}^{n+1} \delta t)^{-1} |\mathbf{r}^{n+1}| (\sin \theta^{n+1} - \sin(\theta^{n+1} - \omega_s^{n+1} \delta t), -(\cos \theta^{n+1} - \cos(\theta^{n+1} - \omega_s^{n+1} \delta t)), 0).$$

Define a mapping  $\hat{B}^{n+1} : \Omega \setminus P(t^{n+1}) \rightarrow \Omega \setminus P(t^n)$  such that  $\hat{B}^{n+1}(\mathbf{x}^{n+1}) = \mathbf{x}^{n+1} - \delta t \bar{\mathbf{u}}^{n+1}(\mathbf{x}^{n+1})$ , and define  $f_\star^n(\mathbf{x}^{n+1}) := f^n(\hat{B}^{n+1} \mathbf{x}^{n+1})$ . Denote  $J^{n+1}$  as the Jacobian of  $\hat{B}^{n+1}$ , we also have

$$\int_{\Omega \setminus P(t^n)} f^n(\mathbf{x}^n) d\mathbf{x}^n = \int_{\Omega \setminus P(t^{n+1})} f_\star^n(\mathbf{x}^{n+1}) J^{n+1} d\mathbf{x}^{n+1}. \quad (3.9)$$

**Remark 3.2.** In the above discussion, to make  $B^n$  bijective, we require  $\delta t$  to be small enough such that  $\delta t < 1/L$ . Such requirement for  $\delta t$  is rarely explicitly mentioned in numerical papers based on algorithms using the ALE method, but it is required to guarantee well-posedness. According to the actual implementation experience, such constraint is very mild in numerical simulation and is naturally satisfied in most cases.



### 3.3. Bijective tALE mapping: 3D case

In this paper we mainly consider 2D problems. Meanwhile, the bijective tALE mapping can also be extended to 3D without extra difficulties. We first consider the case where the particle gravity center is fixed. In this case, we use fixed Eulerian coordinates. As the particle rotates, velocity of an arbitrary point  $\mathbf{x}$  on the particle surface can be expressed as

$$\frac{d\mathbf{x}}{dt} = \boldsymbol{\omega}_s \times \mathbf{r}.$$

For standard tALE, this leads to an explicit discretization

$$\mathbf{x}^{n+1} - \mathbf{x}^n = \delta t (\boldsymbol{\omega}_s^n \times \mathbf{r}^n),$$

which is not bijective between  $t^n$  and  $t^{n+1}$ , as shown in the previous section. To obtain a bijective tALE mapping, similarly we find an  $\mathbf{r}_f^{n+1}$  such that the following identity holds exactly:

$$\mathbf{x}^{n+1} - \mathbf{x}^n = \delta t (\boldsymbol{\omega}_s^{n+1} \times \mathbf{r}_f^{n+1}). \quad (3.10)$$

Given  $\mathbf{x}^{n+1}$ ,  $\mathbf{x}^n$ ,  $\boldsymbol{\omega}_s^{n+1}$ , and  $\boldsymbol{\omega}_s^n$ , (3.10) becomes a group of linear equations of  $\mathbf{r}_f^{n+1}$ . Similar to the 2D case,  $\mathbf{r}_f^{n+1}$  can be explicitly expressed by the position and angular velocity at the current and previous time steps by solving the above linear system. Using the bijective mapping on the particle surface, we can calculate the tALE mesh velocity field  $\bar{\mathbf{u}}^n$  and construct a bijection from  $\Omega \setminus P(t^n)$  to  $\Omega \setminus P(t^{n+1})$ .

If the particle simultaneously translates and rotates, consider Newton-Euler coordinates which are always centered at the particle's gravity center. Motion of the particle is decomposed into translation and rotation. Rotation of the particle can be described in the Newton-Euler coordinates and  $\mathbf{r}_f^{n+1}$  can be solved in a similar way.

For the 3D case, the Jacobian determinant  $J^n$  of linear transformation  $B^n$  is

$$J^n = \begin{vmatrix} 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_1} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_2} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_3} \\ \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_1} & 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_2} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_3} \\ \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_1} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_2} & 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_3} \end{vmatrix} \\ = 1 + \delta t \nabla \cdot \bar{\mathbf{u}}^n + \delta t^2 (J_{12} + J_{23} + J_{31}) + \delta t^3 \det(\nabla \bar{\mathbf{u}}^n),$$

where

$$J_{12} = \begin{vmatrix} 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_1} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_2} \\ \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_1} & 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_2} \end{vmatrix}, \\ J_{23} = \begin{vmatrix} 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_2} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_2}{\partial x_3} \\ \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_2} & 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_3} \end{vmatrix}, \\ J_{31} = \begin{vmatrix} \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_1} & 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_3}{\partial x_3} \\ 1 + \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_1} & \delta t \frac{\partial(\bar{\mathbf{u}}^n)_1}{\partial x_3} \end{vmatrix}.$$

Equation (3.6) also holds in 3D using the above  $\mathbf{r}_f^n$  and  $J^n$ .

## 4. Energy-stable time discretization

In this section, we derive the energy-stable time discretization for our model problem. As mentioned earlier, one of the main difficulties in constructing an energy-stable time discretization comes from the time-dependent shape of the fluid domain. In order to overcome this difficulty, we first modify the tALE method with the bijective mapping introduced in section 3. Then, we construct an implicit numerical scheme in which the tALE mesh velocity, the NS equations for the fluid and Newton's law for the particle are solved in a coupled system. We introduce the Jacobian determinant  $J^{n+1}$  in the numerical scheme of the NS equations, which enable us to balance the fluid kinetic energy in different fluid regions in the energy analysis. From this section,  $(\cdot, \cdot)$  is the abbreviation for  $(\cdot, \cdot)_{\Omega \setminus P(t^{n+1})}$  and  $\|\cdot\|$  denotes the corresponding  $L^2$  norm in  $\Omega \setminus P(t^{n+1})$ .

### 4.1. Energy-stable time discretization

Using the bijective tALE mapping designed in the previous section, we design an energy-stable scheme for the fluid-particle interaction problem:

**Algorithm 4.1** (Energy-stable time discretization).

**Step I:** Given  $(\mathbf{u}^n, p^n, \mathbf{U}_s^n, \boldsymbol{\omega}_s^n, \mathbf{X}_s^n, \theta_s^n)$  at time step  $t^n$ , solve the following coupled system (a)-(c) simultaneously to obtain  $(\tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{U}}_s^{n+1}, \tilde{\boldsymbol{\omega}}_s^{n+1}, \mathbf{X}_s^{n+1}, \theta_s^{n+1})$  at  $t^{n+1}$ :

(a) Calculate particle position and angular orientation by solving

$$\begin{cases} \mathbf{X}_s^{n+1} = \mathbf{X}_s^n + \tilde{\mathbf{U}}_s^{n+1} \delta t, \\ \theta_s^{n+1} = \theta_s^n + \tilde{\boldsymbol{\omega}}_{s,z}^{n+1} \delta t. \end{cases}$$

The region  $P(t^{n+1})$  and  $\Omega \setminus P(t^{n+1})$  are determined by  $\mathbf{X}_s^{n+1}$  and  $\theta_s^{n+1}$  in this step. Here  $\tilde{\boldsymbol{\omega}}_s^{n+1} = (0, 0, \tilde{\boldsymbol{\omega}}_{s,z}^{n+1})$ .

(b) Solve the bijective tALE velocity field:

$$\begin{cases} -\Delta \tilde{\mathbf{u}}^{n+1} = 0, & \text{in } \Omega \setminus P(t^{n+1}), \\ \tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{U}}_s^{n+1}, & \text{on } \partial P(t^{n+1}), \\ \tilde{\mathbf{u}}^{n+1} = 0, & \text{on } \partial \Omega, \end{cases} \quad (4.1)$$

where

$$\begin{aligned} \tilde{\mathbf{u}}_s^{n+1} &= \tilde{\mathbf{U}}_s^{n+1} + \tilde{\boldsymbol{\omega}}_s^{n+1} \times \mathbf{r}_f^{n+1}, \\ \mathbf{r}_f^{n+1} &= (\tilde{\boldsymbol{\omega}}_{s,z}^{n+1} dt)^{-1} |\mathbf{r}^{n+1}| (\sin \theta^{n+1} - \sin(\theta^{n+1} - \tilde{\boldsymbol{\omega}}_{s,z}^{n+1} \delta t), -(\cos \theta^{n+1} - \cos(\theta^{n+1} - \tilde{\boldsymbol{\omega}}_{s,z}^{n+1} \delta t)), 0). \end{aligned}$$

(c) Solve the NS equations for the fluid in  $\Omega \setminus P(t^{n+1})$  and Newton's law for the particle:

$$\mathcal{Re} \left[ \frac{\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^{n+1}}{\delta t} + ((\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^{n+1}) \cdot \nabla) \tilde{\mathbf{u}}^{n+1} \right] = \nabla \cdot \tilde{\boldsymbol{\sigma}}^{n+1} - \frac{1}{2} \mathcal{Re}(\nabla \cdot (\tilde{\mathbf{u}}^{n+1} - \mathbf{u}^{n+1})) \tilde{\mathbf{u}}^{n+1} + \mathbf{f}^{n+1}, \quad (4.2)$$

$$\mathcal{Re} M_s \frac{\tilde{\mathbf{U}}_s^{n+1} - \mathbf{U}_s^n}{\delta t} = - \int_{\partial P(t^{n+1})} \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n} ds + \mathbf{F}_s^{n+1}, \quad (4.3)$$

$$\mathcal{Re} I_s \frac{\tilde{\boldsymbol{\omega}}_s^{n+1} - \boldsymbol{\omega}_s^n}{\delta t} + \boldsymbol{\omega}_s^{n+1} \times I_s \boldsymbol{\omega}_s^n = - \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) ds, \quad (4.4)$$

where  $\mathbf{u}_{**}^n := \mathbf{u}_{**}^n (J^{n+1})^{\frac{1}{2}}$ ,  $\mathbf{u}_{**}^n(\mathbf{x}) := \mathbf{u}^n(\hat{B}^{n+1} \mathbf{x})$ .  $J^{n+1}$  is the Jacobian determinant of bijective mapping  $\hat{B}^{n+1}$ , which is defined by

$$\hat{B}^{n+1} : \Omega \setminus P(t^{n+1}) \rightarrow \Omega \setminus P(t^n), \quad \hat{B}^{n+1}(\mathbf{x}^{n+1}) = \mathbf{x}^{n+1} - \tilde{\mathbf{u}}^{n+1}(\mathbf{x}^{n+1}).$$

The boundary conditions on  $\partial \Omega$  are

$$\tilde{\mathbf{u}}^{n+1} \cdot \mathbf{n} = 0, \quad \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_w) \cdot \boldsymbol{\tau} = -(\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}. \quad (4.5)$$

The boundary conditions on  $\partial P(t^{n+1})$  are

$$(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{U}}_s^{n+1}) \cdot \mathbf{n} = 0, \quad \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{U}}_s^{n+1}) \cdot \boldsymbol{\tau} = -(\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}. \quad (4.6)$$

**Step II:** Given  $(\tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{U}}_s^{n+1}, \tilde{\boldsymbol{\omega}}_s^{n+1}, \mathbf{X}_s^{n+1}, \theta_s^{n+1})$ , solve  $(\mathbf{u}^{n+1}, p^{n+1}, \mathbf{U}_s^{n+1}, \boldsymbol{\omega}_s^{n+1})$  in  $\Omega \setminus P(t^{n+1})$  by pressure correction (see e.g. [18]):

$$\mathcal{Re} \frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}}{\delta t} = -\nabla p^{n+1}, \quad (4.7)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (4.8)$$

$$\mathcal{Re} M_s \frac{\mathbf{U}_s^{n+1} - \tilde{\mathbf{U}}_s^{n+1}}{\delta t} = - \int_{\partial P(t^{n+1})} -p^{n+1} \mathbf{I} \cdot \mathbf{n} ds, \quad (4.9)$$

$$\mathcal{Re} I_s \frac{\boldsymbol{\omega}_s^{n+1} - \tilde{\boldsymbol{\omega}}_s^{n+1}}{\delta t} = - \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (-p^{n+1} \mathbf{I} \cdot \mathbf{n}) ds. \quad (4.10)$$

The boundary condition on  $\partial\Omega$  is

$$\mathbf{u}^{n+1} \cdot \mathbf{n} = 0. \quad (4.11)$$

The boundary condition on  $\partial P(t^{n+1})$  is

$$(\mathbf{u}^{n+1} - \mathbf{u}_s^{n+1}) \cdot \mathbf{n} = 0, \quad (4.12)$$

where  $\mathbf{u}_s^{n+1} = \mathbf{U}_s^{n+1} + \boldsymbol{\omega}_s^{n+1} \times \mathbf{r}_f^{n+1}$ .

**Remark 4.1.** From the definition,  $J^{n+1}$  is of order 1 +  $O(\delta t)$ . Multiplying  $J^{n+1}$  by  $\mathbf{u}_\star^n$  brings an  $O(\delta t)$  truncation error in the discrete scheme, so that the time discretization is consistent with the continuous system, while the accuracy is limited to  $O(\delta t)$ .

Similar to the continuous model, we may also define total energy of the discrete system. At time step  $t^n$ , the total energy  $E^n$  of the governing system is defined as:

$$E^n := \mathcal{R}e E_k^n + \mathcal{R}e M_s E_{pm}^n + \mathcal{R}e I_s E_{pr}^n \quad (4.13)$$

where

$$E_k^n := \frac{1}{2} \int_{\Omega \setminus P(t^n)} |\mathbf{u}^n|^2 dx, \quad E_{pm}^n := \frac{1}{2} |\mathbf{U}_s^n|^2, \quad E_{pr}^n := \frac{1}{2} |\boldsymbol{\omega}_s^n|^2.$$

#### 4.2. Energy law of the time discretization

Assuming that wall velocity is zero, we have the following energy law for the time discretization Algorithm 4.1:

**Theorem 4.2** (Energy law of the time discretization). Assume that the wall velocity  $\mathbf{u}_w = 0$ , Algorithm 4.1 satisfies the following energy law:

$$\begin{aligned} E^{n+1} - E^n &\leq -\delta t \|\nabla \tilde{\mathbf{u}}^{n+1}\|^2 + \delta t (\mathbf{f}^{n+1}, \tilde{\mathbf{u}}^{n+1}) + \delta t \mathbf{F}_s^{n+1} \cdot \tilde{\mathbf{U}}_s^{n+1} \\ &\quad - \delta t \frac{1}{L_s} \|\tilde{\mathbf{u}}^{n+1} \cdot \boldsymbol{\tau}\|_{\partial\Omega}^2 - \delta t \frac{1}{L_s} \|(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau}\|_{\partial P(t^{n+1})}^2. \end{aligned} \quad (4.14)$$

**Proof.** Using property (3.9) of the bijective mapping  $\hat{B}^{n+1}$  and a simple inequality  $\frac{1}{2}(a^2 - b^2) \leq a(a - b)$ , we have

$$\mathcal{R}e(\tilde{E}_k^{n+1} - E_k^n) = \frac{1}{2} \mathcal{R}e \left( \frac{1}{2} |\tilde{\mathbf{u}}^{n+1}|^2 - |\mathbf{u}_\star^n|^2 J^{n+1}, 1 \right) \leq \mathcal{R}e(\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_\star^n, \tilde{\mathbf{u}}^{n+1}). \quad (4.15)$$

Taking the inner product of (4.2) with  $\tilde{\mathbf{u}}^{n+1} \delta t$ , we obtain

$$\mathcal{R}e(\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_\star^n, \tilde{\mathbf{u}}^{n+1}) \leq I_1 + I_2 + I_3 \quad (4.16)$$

where

$$\begin{aligned} I_1 &= -\delta t \mathcal{R}e((\tilde{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^{n+1}) \cdot \nabla) \tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{u}}^{n+1}) - \frac{1}{2} \delta t \mathcal{R}e(\nabla \cdot (\tilde{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^{n+1}), |\tilde{\mathbf{u}}^{n+1}|^2), \\ I_2 &= \delta t (\nabla \cdot \tilde{\boldsymbol{\sigma}}^{n+1}, \tilde{\mathbf{u}}^{n+1}), \quad I_3 = \delta t (\mathbf{f}^{n+1}, \tilde{\mathbf{u}}^{n+1}). \end{aligned}$$

Using Green's formula and boundary conditions (4.5), (4.6) and (4.1), we obtain

$$I_1 = \delta t < \frac{1}{2} |\tilde{\mathbf{u}}^{n+1}|^2, (\tilde{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^{n+1}) \cdot \mathbf{n} >_{\partial\Omega} + \delta t < \frac{1}{2} |\tilde{\mathbf{u}}^{n+1}|^2, (\tilde{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^{n+1}) \cdot \mathbf{n} >_{\partial P(t)} = 0.$$

Using Green's formula we also have

$$\begin{aligned} I_2 &= -\delta t (\tilde{\boldsymbol{\sigma}}^{n+1}, \nabla \tilde{\mathbf{u}}^{n+1}) + \delta t < \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}^{n+1} >_{\partial\Omega} + \delta t < \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}^{n+1} >_{\partial P(t)} \\ &= -\delta t \|\nabla \tilde{\mathbf{u}}^{n+1}\|^2 + \delta t < \boldsymbol{\sigma}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}^{n+1} >_{\partial\Omega} + \delta t < \boldsymbol{\sigma}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}^{n+1} >_{\partial P(t)}. \end{aligned}$$

Plugging the above two equalities in (4.16) and using (4.15), we obtain

$$\mathcal{R}e(\tilde{E}_k^{n+1} - E_k^n) \leq -\delta t \|\nabla \tilde{\mathbf{u}}^{n+1}\|^2 + (\mathbf{f}^{n+1}, \tilde{\mathbf{u}}^{n+1}) + \delta t < \boldsymbol{\sigma}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}^{n+1} >_{\partial\Omega} + \delta t < \boldsymbol{\sigma}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}^{n+1} >_{\partial P(t)}. \quad (4.17)$$

Taking the inner product of (4.3) with  $\tilde{\mathbf{U}}_s^{n+1}\delta t$  and the inner product of (4.4) with  $\boldsymbol{\omega}_s^{n+1}\delta t$ , then summing these two equations, we obtain

$$\mathcal{R}eM_s(\tilde{E}_{pm}^{n+1} - E_{pm}^n) + \mathcal{R}eI_s(\tilde{E}_{pr}^{n+1} - E_{pr}^n) \leq -\delta t < \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}, \tilde{\mathbf{u}}_s^{n+1} >_{\partial P(t^{n+1})} + \mathbf{F}_s^{n+1} \cdot \tilde{\mathbf{U}}_s^{n+1}. \quad (4.18)$$

Similar to the proof of the continuous case, summing (4.17) and (4.18) and using boundary conditions (4.5) and (4.6), we obtain

$$\begin{aligned} \tilde{E}^{n+1} - E^n &\leq -\delta t \|\nabla \tilde{\mathbf{u}}^{n+1}\|^2 + \delta t (\mathbf{f}^{n+1}, \tilde{\mathbf{u}}^{n+1}) + \delta t \mathbf{F}_s^{n+1} \cdot \tilde{\mathbf{U}}_s^{n+1} \\ &\quad - \delta t \frac{1}{L_s} \|\tilde{\mathbf{u}}^{n+1} \cdot \boldsymbol{\tau}\|_{\partial\Omega}^2 - \delta t \frac{1}{L_s} \|(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau}\|_{\partial P(t^{n+1})}^2. \end{aligned} \quad (4.19)$$

For the pressure correction step, taking the inner product of (4.7) with  $\delta t \mathbf{u}^{n+1}$ , and using the inequality  $\frac{1}{2}(a^2 - b^2) \leq a(a - b)$ , we have

$$\mathcal{R}e(E_k^{n+1} - \tilde{E}_k^{n+1}) \leq \mathcal{R}e(\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}, \mathbf{u}^{n+1}) = -(\nabla p^{n+1}, \mathbf{u}^{n+1}).$$

Using Green's formula and boundary conditions (4.11) and (4.12), we have

$$\mathcal{R}e(E_k^{n+1} - \tilde{E}_k^{n+1}) \leq -\delta t < p^{n+1}, \mathbf{u}^{n+1} \cdot \mathbf{n} >_{\partial P(t^{n+1})}. \quad (4.20)$$

Taking the inner product of (4.9) with  $\mathbf{U}_s^{n+1}\delta t$  and the inner product of (4.10) with  $\boldsymbol{\omega}_s^{n+1}\delta t$ , then summing up the two identities, we obtain

$$E^{n+1} - \tilde{E}^{n+1} \leq 0. \quad (4.21)$$

Energy law (4.14) is then straightforward combining (4.19) and (4.21)  $\square$

**Remark 4.2.** When the wall velocity  $\mathbf{u}_w \neq 0$ , define  $W^n$  as the accumulated work done by the wall:

$$W^n := \sum_{i=0}^n w^i, \quad w^i := \int_{\partial\Omega} \delta t ((\tilde{\boldsymbol{\sigma}}^{n+1}(\mathbf{x}, t^i) \cdot \mathbf{n}) \cdot \boldsymbol{\tau})(\mathbf{u}_w \cdot \boldsymbol{\tau}) ds,$$

and we have

$$\begin{aligned} (E^{n+1} + W^{n+1}) - (E^n + W^n) &\leq -\delta t \|\nabla \tilde{\mathbf{u}}^{n+1}\|^2 + \delta t (\mathbf{f}^{n+1}, \tilde{\mathbf{u}}^{n+1}) + \delta t \mathbf{F}_s^{n+1} \cdot \tilde{\mathbf{U}}_s^{n+1} \\ &\quad - \delta t \frac{1}{L_s} \|(\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_w) \cdot \boldsymbol{\tau}\|_{\partial\Omega}^2 - \delta t \frac{1}{L_s} \|(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau}\|_{\partial P(t^{n+1})}^2. \end{aligned}$$

**Remark 4.3.** Since we use the bijective mapping for fluid domains, the volume of the fluid domain remains constant at each time step, thus time discretization in Algorithm 4.1 conserves mass. Conservation of mass is also true for the fully discrete scheme since we use a level-set function to describe the particle surface. The time discretization does not conserve momentum. The fully discrete scheme introduced in the next section does not conserve momentum either.

Using techniques introduced elsewhere [19], we may also obtain an equivalent form of Algorithm 4.1 by eliminating  $\mathbf{u}^{n+1}$  in the numerical scheme and view  $\tilde{\mathbf{u}}^{n+1}$  as the solution at time step  $t^{n+1}$ . Such equivalent algorithm is given in Appendix A.

## 5. Numerical implementation

In this section, we describe the implementation of Algorithm 4.1. We solve the coupled system by iteration, and use the XFEM method to handle variable discontinuities across the particle surface.

### 5.1. The extended finite element method

The XFEM method [12,40] is widely used for handling the jump in unknowns across elements. In order to handle the discontinuities across  $\partial P(t^{n+1})$ , we enrich the FEM basis functions by adding Heaviside function  $H(s)$ , which is defined by a scalar level set function  $s$ :

$$H(s) = \begin{cases} 1, & s \geq 0, \\ 0, & s < 0, \end{cases} \quad \text{where } s(\mathbf{x}) \begin{cases} > 0, & \mathbf{x} \in \Omega \setminus P(t^n), \\ < 0, & \mathbf{x} \in P(t^n), \\ = 0, & \mathbf{x} \in \partial P(t^n). \end{cases} \quad (5.1)$$

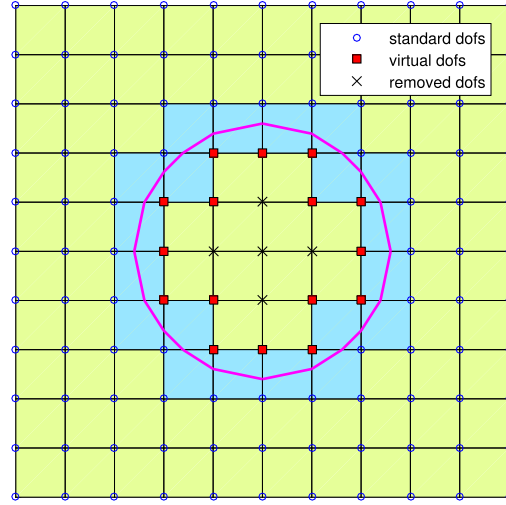


Fig. 8. Standard, virtual, and removed DOFs in XFEM.

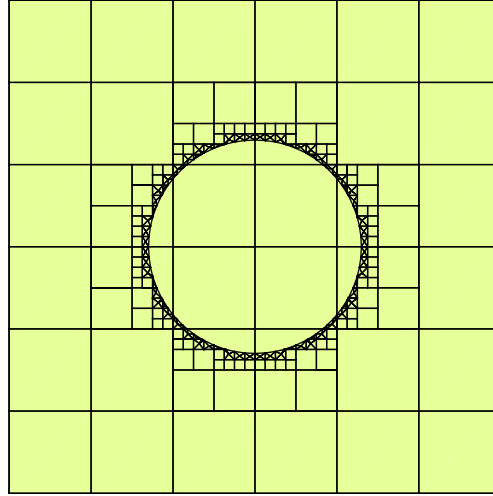


Fig. 9. Element subdivisions near the particle surface.

We choose  $s(\mathbf{x})$  as a signed distance function to the particle boundary. For example, for an ellipse with major radius  $a$ , minor radius  $b$  located at  $(x_0, y_0)$  and with the major axis parallel to the  $x$  axis, the level-set function  $s(\mathbf{x})$  for  $\mathbf{x} = (x, y)$  is defined by

$$s(\mathbf{x}) := \left( \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} \right)^{\frac{1}{2}} - 1.$$

Denote by  $\{\phi_i\}$  a standard set of FEM basis functions in an element  $T$ . In XFEM we use basis functions  $\{H(s)\phi_i\}$  instead of the standard FEM basis.

A simple interpretation of the XFEM enrichment procedure for the fluid-particle interaction system is shown in Fig. 8. We classify the degrees of freedom (DOFs) into standard DOFs, virtual DOFs, and removed DOFs by the relationship between the element and the particle surface. For the elements intersected by the interface  $\partial P(t^n)$ , we mark the DOFs inside the fluid region as standard DOFs and the DOFs inside the particle region  $P(t)$  as virtual DOFs. For the remaining DOFs, if the DOFs lie in an element entirely inside the fluid domain, then the DOFs are standard DOFs. Otherwise, the element is entirely inside the particle region, and the DOFs in that element are removed since they do not have any contribution to the system.

It is well known that the accuracy of a numerical quadrature is crucial for the robustness of XFEM [10]. To achieve better approximation, we perform local mesh subdivisions near the interface to calculate the weights of the numerical quadrature. The local mesh subdivisions are only used in calculating the numerical quadrature and do not add extra DOFs to the system of equations. An example of the subdivision near the interface is shown in Fig. 9.

## 5.2. The iteration scheme

In this section, we discuss how to apply step I in Algorithm A.1 in practice. We discuss Algorithm A.1 since it is the algorithm we use in the real calculation. Implementation of Algorithm 4.1 is similar. Since equations (a)-(c) in the numerical scheme are strongly coupled, we solve the equations (a)-(c) by iterating over the particle velocity  $\mathbf{U}_s^{n+1}$  and angular velocity  $\tilde{\omega}_s^{n+1}$ . The iteration proceeds are as follows:

### Algorithm 5.1 (Nonlinear iteration).

1. Given an initial guess of  $\mathbf{U}_s^{n+1}$  and  $\omega_s^{n+1}$ , let  $\tilde{\mathbf{U}}_{s,old}^{n+1}$ ,  $\tilde{\omega}_{s,old}^{n+1}$  be equal to the initial guess of  $\mathbf{U}_s^{n+1}$  and  $\omega_s^{n+1}$ . For example, we may choose the initial guess as  $\tilde{\mathbf{U}}_{s,old}^{n+1} = \mathbf{U}_s^n$  and  $\tilde{\omega}_{s,old}^{n+1} = \omega_s^n$ .

2. Solve the following decoupled equations in (a)-(c) respectively:

(a) Calculate particle position and angular orientation by

$$\begin{cases} \mathbf{X}_s^{n+1} = \mathbf{X}_s^n + \tilde{\mathbf{U}}_{s,old}^{n+1} \delta t, \\ \theta_s^{n+1} = \theta_s^n + \tilde{\omega}_{s,old,z}^{n+1} \delta t, \end{cases}$$

where  $\tilde{\omega}_{s,old}^{n+1} = (0, 0, \tilde{\omega}_{s,old,z}^{n+1})$ .

(b) Given  $\tilde{\mathbf{U}}_{s,old}^{n+1}$  and  $\tilde{\omega}_{s,old}^{n+1}$ , solve the tALE mesh velocity in  $\Omega \setminus P(t^{n+1})$ :

$$\begin{cases} -\Delta \tilde{\mathbf{u}}^{n+1} = 0, & \text{in } \Omega \setminus P(t^{n+1}), \\ \tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}}_s^{n+1}, & \text{on } \partial P(t^{n+1}), \\ \tilde{\mathbf{u}}^{n+1} = 0, & \text{on } \partial \Omega, \end{cases} \quad (5.2)$$

where

$$\begin{aligned} \tilde{\mathbf{u}}_s^{n+1} &= \tilde{\mathbf{U}}_{s,old}^{n+1} + \tilde{\omega}_{s,old}^{n+1} \times \mathbf{r}_f^{n+1}, \\ \mathbf{r}_f^{n+1} &= (\tilde{\omega}_{s,old,z}^{n+1} \delta t)^{-1} |\mathbf{r}^{n+1}| (\sin \theta^{n+1} - \sin(\theta^{n+1} - \tilde{\omega}_{s,old,z}^{n+1} \delta t), -(\cos \theta^{n+1} - \cos(\theta^{n+1} - \tilde{\omega}_{s,old,z}^{n+1} \delta t)), 0). \end{aligned}$$

(c) Solve the NS equations and Newton's law in  $\Omega \setminus P(t^{n+1})$  with the tALE mesh velocity in (b):

$$\text{Re} \left[ \frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_*^{n+1}}{\delta t} + ((\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1}) \cdot \nabla) \tilde{\mathbf{u}}^{n+1} \right] = -(\nabla p^n)_* J^{\frac{1}{2}} + \nabla \cdot \tilde{\boldsymbol{\sigma}}^{n+1} - \frac{1}{2} \text{Re}(\nabla \cdot (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1})) \tilde{\mathbf{u}}^{n+1}, \quad (5.3)$$

$$\text{Re} M_s \frac{\tilde{\mathbf{U}}_s^{n+1} - \tilde{\mathbf{U}}_s^n}{\delta t} = - \int_{\partial P(t^n)} -p^n \mathbf{I} \cdot \mathbf{n} ds - \int_{\partial P(t^{n+1})} \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n} ds, \quad (5.4)$$

$$\text{Re} I_s \frac{\tilde{\omega}_s^{n+1} - \tilde{\omega}_s^n}{\delta t} = - \int_{\partial P(t^n)} \mathbf{r}_f^n \times (-p^n \mathbf{I} \cdot \mathbf{n}) ds - \int_{\partial P(t^{n+1})} \mathbf{r}_{f,old}^{n+1} \times (\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) ds. \quad (5.5)$$

The boundary conditions on  $\partial \Omega$  are

$$\tilde{\mathbf{u}}^{n+1} \cdot \mathbf{n} = 0, \quad \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_w) \cdot \boldsymbol{\tau} = -(\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}. \quad (5.6)$$

The boundary conditions on  $\partial P(t^{n+1})$  are

$$(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \mathbf{n} = 0, \quad \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau} = -(\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}. \quad (5.7)$$

3. Let  $\tilde{\mathbf{U}}_{s,new}^{n+1} = \tilde{\mathbf{U}}_s^{n+1}$ ,  $\tilde{\omega}_{s,new}^{n+1} = \tilde{\omega}_s^{n+1}$ . If

$$|\tilde{\mathbf{U}}_{s,new}^{n+1} - \tilde{\mathbf{U}}_{s,old}^{n+1}| / |\tilde{\mathbf{U}}_{s,old}^{n+1}| < \text{tol} \text{ and } |\tilde{\omega}_{s,new}^{n+1} - \tilde{\omega}_{s,old}^{n+1}| / |\tilde{\omega}_{s,old}^{n+1}| < \text{tol},$$

then output  $(\tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{U}}_s^{n+1}, \tilde{\omega}_s^{n+1}, \mathbf{X}_s^{n+1}, \theta_s^{n+1})$  as the solution at time  $t^{n+1}$ , else set

$$\tilde{\mathbf{U}}_{s,old}^{n+1} = \tilde{\mathbf{U}}_{s,new}^{n+1}, \quad \tilde{\omega}_{s,old}^{n+1} = \tilde{\omega}_{s,new}^{n+1},$$

then repeat step 2.

We can see that if  $\tilde{\mathbf{U}}_{s,new}^{n+1} = \tilde{\mathbf{U}}_{s,old}^{n+1}$  and  $\tilde{\omega}_{s,new}^{n+1} = \tilde{\omega}_{s,old}^{n+1}$  hold exactly after step 2, then we obtain exactly the solution of the tALE step I in Algorithm A.1, i.e. solution of step I in Algorithm A.1 is the fixed point of the above iteration.

We also introduce the FEM weak form that we use in practical implementation for (c). Penalty terms with parameter  $\epsilon_p$  are introduced to deal with the essential boundary conditions on  $\partial P(t^{n+1})$  and  $\partial\Omega$ . For example, we apply a Robin type boundary condition on  $\partial P(t^{n+1})$ ,

$$(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \mathbf{n} + \epsilon_p (\tilde{\boldsymbol{\sigma}} \cdot \mathbf{n}) \cdot \mathbf{n} = 0,$$

instead of the Dirichlet boundary condition on  $\partial P(t^{n+1})$ :

$$(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \mathbf{n} = 0.$$

Similar Robin-type boundary conditions are applied on  $\partial\Omega$ .

Denote by  $\Omega_h$  the set of FEM elements that we use in computation, i.e. the set of elements with only standard/virtual DOFs. Denote by  $\partial P_h(t^{n+1})$  the approximated particle boundary  $\partial P(t^{n+1})$ . Define  $(\cdot, \cdot)$  as the  $L^2$  inner products in  $\Omega_h$ . Denote  $\langle \cdot, \cdot \rangle_{\partial P_h(t^{n+1})}$  and  $\langle \cdot, \cdot \rangle_{\partial\Omega}$  as the  $L^2$  inner product on  $\partial P_h(t^{n+1})$  and  $\partial\Omega$  respectively. We use a Q1-Q1 element to discretize  $\tilde{\mathbf{u}}^{n+1}$  and denote the corresponding FEM discrete space by  $(Q^1(\Omega_h))^2$ . Define  $\mathbb{R}_{001} := \text{span}\{(0, 0, 1)\}$ . The FEM weak form of step (c) is as follows:

Find  $(\tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{u}}_s^{n+1}, \tilde{\omega}_s^{n+1}) \in (Q^1(\Omega_h))^2 \times \mathbb{R} \times \mathbb{R}_{001}$ , such that for all  $(\mathbf{v}, \mathbf{V}_s, \boldsymbol{\phi}_s) \in (Q^1(\Omega_h))^2 \times \mathbb{R} \times \mathbb{R}_{001}$ ,

$$\begin{aligned} & \mathcal{R}e(\tilde{\mathbf{u}}^{n+1}, \mathbf{v}) + \delta t \mathcal{R}e((\tilde{\mathbf{u}}^{n+1} - \bar{\mathbf{u}}^{n+1}) \cdot \nabla) \mathbf{u}_*^n, \mathbf{v}) + \delta t (\tilde{\boldsymbol{\sigma}}^{n+1}, \nabla \mathbf{v}) \\ & + \delta t \epsilon_p^{-1} \langle \tilde{\mathbf{u}}^{n+1} \cdot \mathbf{n}, \mathbf{v} \cdot \mathbf{n} \rangle_{\partial\Omega} + \delta t \langle \frac{1}{L_s} \tilde{\mathbf{u}}^{n+1} \cdot \boldsymbol{\tau}, \mathbf{v} \cdot \boldsymbol{\tau} \rangle_{\partial\Omega} \\ & + \delta t \epsilon_p^{-1} \langle (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \mathbf{n}, (\mathbf{v} - \mathbf{V}_s) \cdot \mathbf{n} \rangle_{\partial P_h(t^{n+1})} \\ & + \delta t \langle \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau}, (\mathbf{v} - \mathbf{V}_s) \cdot \boldsymbol{\tau} \rangle_{\partial P_h(t^{n+1})} \\ & + \frac{1}{2} \delta t \mathcal{R}e((\nabla \cdot (\tilde{\mathbf{u}}_s^{n+1} - \bar{\mathbf{u}}^{n+1})) \mathbf{u}_*^n, \mathbf{v}) + \mathcal{R}e M_s \tilde{\mathbf{u}}_s^{n+1} \cdot \mathbf{V}_s + \mathcal{R}e I_s \tilde{\omega}_s^{n+1} \cdot \boldsymbol{\phi}_s \\ = & \mathcal{R}e(\mathbf{u}_*^n J^{\frac{1}{2}}, \mathbf{v}) - \delta t ((\nabla p^n)_{\bar{u}} J^{\frac{1}{2}}, \mathbf{v}) + \delta t \langle \frac{1}{L_s} \mathbf{u}_w \cdot \boldsymbol{\tau}, \mathbf{v} \cdot \boldsymbol{\tau} \rangle_{\partial\Omega} \\ & - \delta t \langle -p^n \mathbf{I} \cdot \mathbf{n}, \mathbf{V}_s + \boldsymbol{\phi}_s \times \mathbf{r}_f^{n+1} \rangle_{\partial P_h(t^n)} + \mathcal{R}e M_s \mathbf{U}_s^n \cdot \mathbf{V}_s + \mathcal{R}e I_s \omega_s^n \cdot \boldsymbol{\phi}_s \\ & + \delta t \mathbf{F}_s^{n+1} \cdot \mathbf{V} + \delta t (\mathbf{f}^{n+1}, \mathbf{v}). \end{aligned}$$

## 6. Numerical experiments

We test the accuracy and stability of our numerical scheme with two numerical experiments. In the first example, we simulate the Jeffery orbit of a fixed ellipse rotating in a shear flow and compare the results with those obtained from the standard tALE-XFEM method. In the second example, we simulate the dynamic behavior of an elliptical free-falling particle in water.

### 6.1. The Jeffery orbits

The phenomena of suspension and dynamics of a particle in a viscous flow have attracted intense interest in the scientific community. In this numerical example, we simulate a fixed ellipse rotating in a shear flow, as shown in Fig. 10. It has been shown experimentally that the ellipse undergoes a periodic revolution [37]. The particle motion consists of a spin about the axis of symmetry and a precession of this axis about the vorticity of the undisturbed flow. The rate of spin is equal to the component of the vorticity in the direction of the axis of symmetry. In the well-known paper of Jeffery [24], the spin rate of the ellipse is described in terms of polar angles. Bretherton [9] demonstrated the general validity of the Jeffery orbit for particles with rotational symmetry. The effect of the boundary slip on the Jeffery orbit has also been studied in [48]. In this section, we simulate a Jeffery orbit with our energy-stable scheme.

The computational domain  $\Omega$  is  $[-1, 1] \times [-1, 1]$ . Boundary conditions on the top and bottom wall are Navier-slip boundary conditions with the wall speeds  $\mathbf{u}_w = 2$  and  $\mathbf{u}_w = -2$  respectively. A periodic boundary condition is applied on the left and right boundaries of  $\Omega$ . The center of the ellipse is set at  $(0, 0)$  with zero initial velocity. Major and minor axes of the ellipse are  $a = 0.25, b = 0.15$ . In the absence of gravity and slip on the particle surface, Jeffery [24] showed that for a two-dimensional ellipse in the Stokes flow, the ellipse undergoes a periodic tumbling which is determined by the Jeffery orbit equation

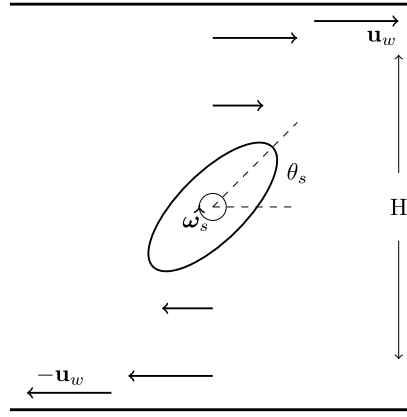


Fig. 10. An illustration of the Jeffery Orbit.

Table 1

Fitted  $k$  and  $\gamma$  solved by the energy-stable scheme.

| mesh             | $k$    | $\gamma$ |
|------------------|--------|----------|
| $50 \times 50$   | 0.4723 | 1.9624   |
| $100 \times 100$ | 0.4700 | 1.9627   |
| $200 \times 200$ | 0.4695 | 1.9625   |
| Theoretical      | 0.4706 | 2        |

Table 2

 $L^2$  norm of the error and convergence rate for x-velocity component  $u$ , y-velocity component  $v$  and pressure  $p$  at  $T = 0.2$  with different grids.

| mesh                                  | $u$      | rate  | $v$      | rate  | $p$      | rate  |
|---------------------------------------|----------|-------|----------|-------|----------|-------|
| $(50 \times 50) - (800 \times 800)$   | 5.94e-02 |       | 7.59e-04 |       | 8.84e-04 |       |
| $(100 \times 100) - (800 \times 800)$ | 1.20e-02 | 2.307 | 3.25e-04 | 1.223 | 3.01e-04 | 1.554 |
| $(200 \times 200) - (800 \times 800)$ | 4.68e-03 | 1.358 | 1.05e-04 | 1.630 | 1.05e-04 | 1.519 |
| $(400 \times 400) - (800 \times 800)$ | 1.83e-03 | 1.354 | 3.56e-05 | 1.761 | 3.43e-05 | 1.614 |

$$\omega_s = \frac{\gamma}{2}(-1 + k \cos 2\theta_s), \quad (6.1)$$

where  $\gamma = 2\mathbf{u}_w/H$  and  $k = \frac{1-e^2}{1+e^2}$ .  $H = 2$  is the distance between walls, and  $e = b/a = 0.6$ . When the inertial effect is considered, Ding et al. [11] showed that the system approximates the Jeffery orbits for a sufficiently small Reynolds number.

In this numerical experiment, we choose  $\rho_s = \rho$ . Other parameters in our numerical experiments are

$$\mathcal{Re} = 1, L_s = 0.0025, \epsilon_p = 10^{-10}.$$

We compute the solution with three different grids,  $50 \times 50$ ,  $100 \times 100$  and  $200 \times 200$  with the same time step  $\delta t = 0.001$ . Best fitted  $k$  and  $\gamma$  are calculated by linear regression. A comparison with the theoretical result is given in Table 1.

To check the accuracy of the numerical scheme, we simulate the Jeffery orbit problem with different grids and a sufficiently small time step size  $\delta t = 5 \times 10^{-5}$ . The meshes used are  $50 \times 50$ ,  $100 \times 100$ ,  $200 \times 200$ ,  $400 \times 400$ , and  $800 \times 800$ , all of which are uniform rectangular meshes. We use bilinear FEM elements for all the variables in the fluid domain. Denote by  $T^N$  the set of all mesh points of an  $N \times N$  mesh. We compute the error at  $t_0 = 0.2$  by taking the difference between the numerical solutions using the  $N \times N$  mesh and a reference solution using an  $800 \times 800$  mesh for  $N = 50, 100, 200$ , and 400. The  $L^2$  errors  $E_N$  of the fluid field unknowns are approximated by numerical integration in the fluid region with an  $800 \times 800$  mesh via the following steps: first, interpolate FEM solutions on the  $N \times N$  mesh into a piecewise bilinear function space defined on an  $800 \times 800$  mesh. Such interpolations are identical since the  $800 \times 800$  mesh can be viewed as a refined mesh from  $N \times N$ , where  $N = 50, 100, 200$ , and 400. Next, compute the error by calculating

$$E_N = \left[ \sum_{x \in \Omega \setminus P(t_0) \cap T^{1600}} |(\cdot)_N(x) - (\cdot)_{800}(x)|^2 \delta x^2 \right]^{\frac{1}{2}}, \quad (6.2)$$

where  $(\cdot)_N$  is the numerical solution calculated using the  $N \times N$  mesh. The convergence rate is calculated by  $\log_2(E_N/E_{2N})$ . Table 2 shows the results using the energy-stable scheme. Numerical results show that the velocity and pressure are convergent. Accuracy of velocity and pressure is of first order.



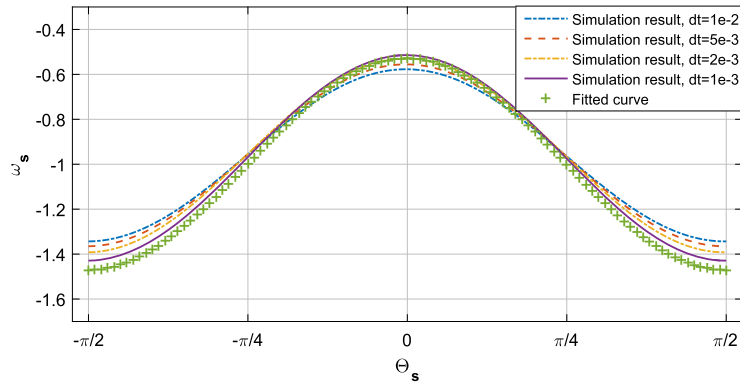


Fig. 11. Fitted curve with different time steps in the energy-stable scheme using a  $200 \times 200$  mesh.

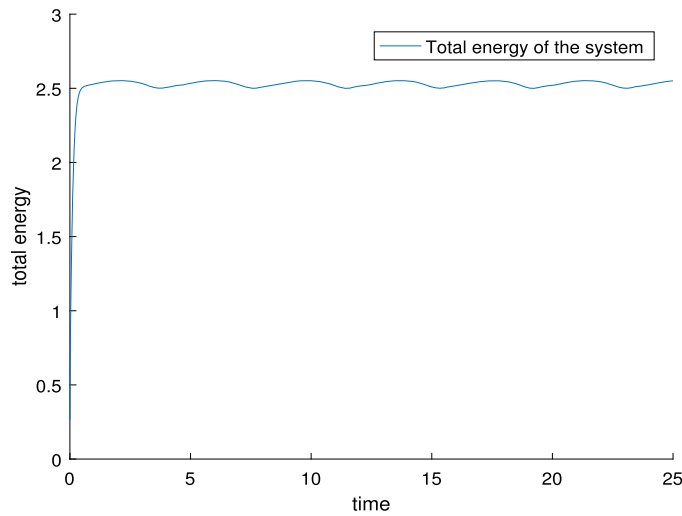


Fig. 12. Total energy of the system with a  $200 \times 200$  mesh,  $\delta t = 5 \times 10^{-3}$ .

To see if the energy-stable time discretization improves numerical stability in practical implementation, a comparison is made between our energy-stable scheme and the standard explicit tALE scheme with XFEM (see section 3). We use the same model and parameters with a fixed  $200 \times 200$  mesh and run the code with increasing the time step size  $\delta t$ . For the standard tALE method, the numerical calculation blows up when we increase time steps to  $\delta t = 3 \times 10^{-3}$ , while our energy-stable scheme is still stable for time steps as large as  $\delta t = 10^{-2}$  ( $\delta t = \delta x$ ). Fig. 11 shows  $\omega_s - \theta_s$  fitted curves obtained from the energy-stable scheme with different time steps. We show the total energy over time for the energy-stable scheme in Fig. 12 using time step size  $\delta t = 5 \times 10^{-3}$  and on a  $200 \times 200$  mesh. The energy of the entire system undergoes a periodical change, thus remains stable for 5,000 steps of calculation (25 seconds for the simulated problem). For the momentum of the continuous system, the total momentum of the solution is zero in both x and y directions due to symmetry. For the numerical results, we show the history of total momentum in the x-direction with different time step sizes in Fig. 13. It can be seen that total momentum is stably controlled in a small region and approaches to zero as the time step size becomes smaller.

Comparing between the fitted curve of the standard tALE scheme and that of our energy-stable scheme, it can be shown that the energy-stable scheme gives a less oscillative  $\theta_s - \omega_s$  curve. The fitted curves for the standard tALE scheme and our energy-stable scheme with a  $100 \times 100$  mesh are shown in Fig. 14. Time step size for both methods is chosen as  $\delta t = 10^{-3}$ . According to the numerical results, both numerical schemes fit well with the theoretical result. Meanwhile, the standard tALE scheme produces oscillations in the simulation results (similar oscillations also appear in the numerical results in e.g. [48]), while the energy-stable scheme still produces a near-perfect cosine curve. We show in Fig. 15 a comparison of the error of  $\omega_s$  between the numerical result and the fitted curve calculated by the standard tALE scheme and our energy-stable scheme.

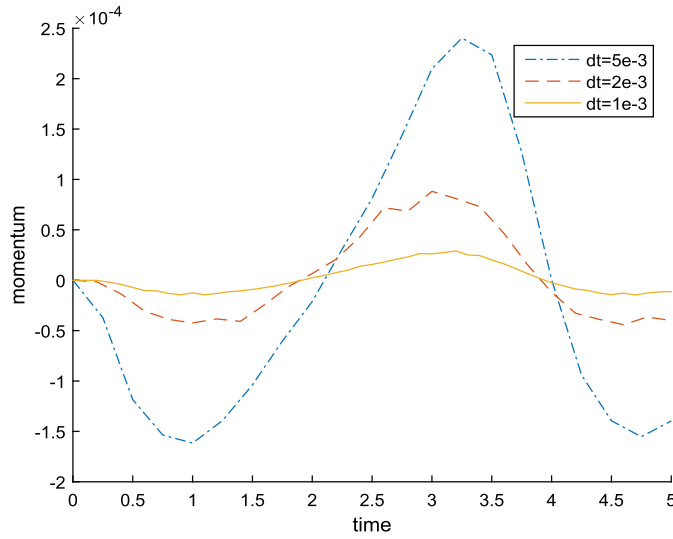


Fig. 13. Total momentum in the x direction for different time step sizes, using  $200 \times 200$  mesh.

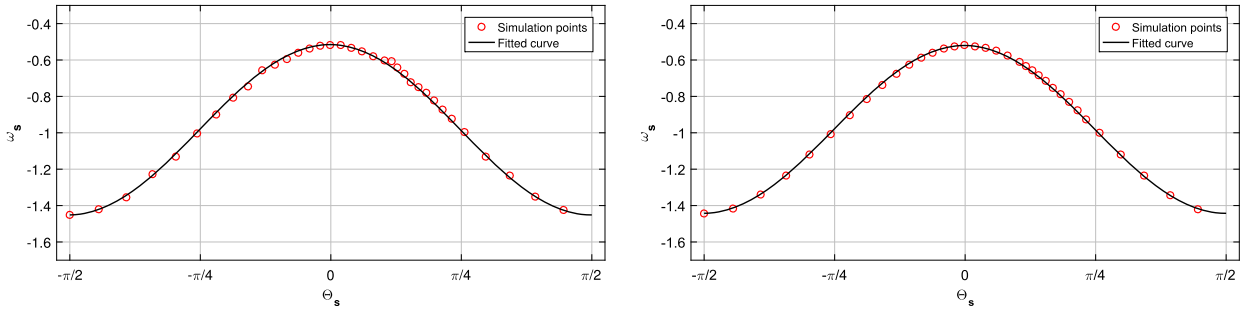


Fig. 14. Fitted curves of  $\omega_s$  and  $\theta_s$  by the standard tALE method (left) and our energy-stable scheme (right), using a  $100 \times 100$  mesh,  $\delta t = 1e - 3$ . Both results are well-fitted by a cosine curve, while the standard tALE produces more oscillations.

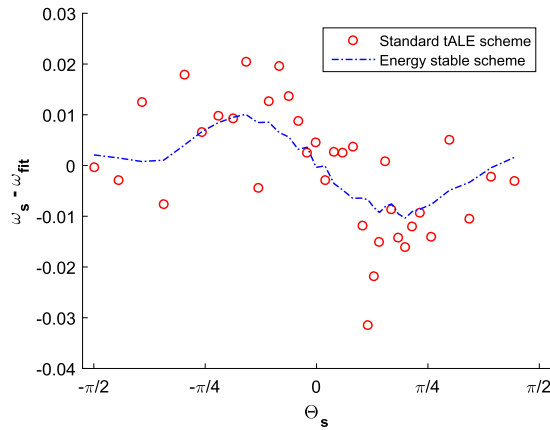


Fig. 15. Error between the fitted curve and numerical simulation, where the latter is performed using the standard tALE scheme and our energy-stable scheme with a  $100 \times 100$  mesh,  $\delta t = 1e - 3$ .

## 6.2. Elliptical particle free falling in water

In this experiment, we simulate the dynamics of a free-falling elliptical particle inside water (see Fig. 16). Under the action of gravity, the free-falling elliptical particle turns, flutters, and tumbles, and therefore its downward trajectory is complicated. The dynamical behavior of the particle is highly sensitive to the initial position and velocity of the particle,

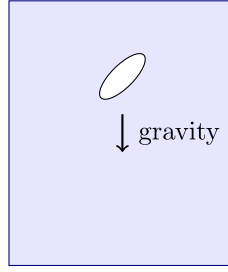


Fig. 16. Illustration of a falling elliptical particle inside water.

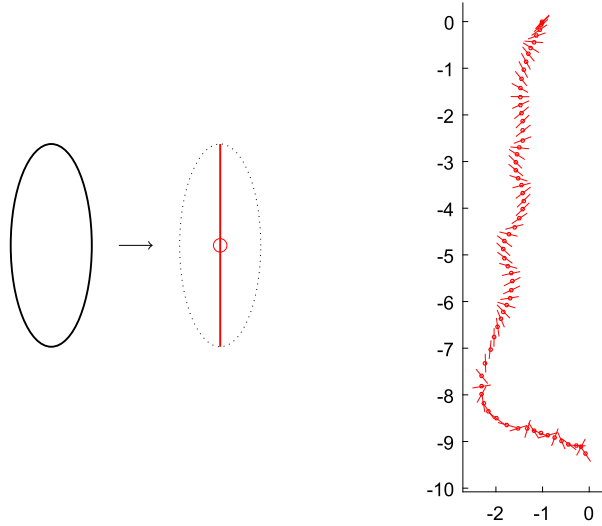


Fig. 17. Trajectory of the free-falling ellipse. Hollow circle indicates the center while the line segment indicates the long axis of the ellipse, as shown on the left.

as well as the boundary conditions on the boundary of the region. The behavior of objects free-falling in fluids has been studied extensively both experimentally and numerically [7,29,33,2,1,25]. We carry out a numerical experiment to simulate this problem using our energy-stable scheme. The computational domain is  $(-3, 1) \times (-11, 1)$  with a  $200 \times 600$  rectangular mesh. The ellipse is released at  $(0, 0)$ , with angular orientation  $\theta_s = \pi/4$ . Major and minor axes of the ellipse are chosen as  $r_a = 0.25$ ,  $r_b = 0.1$ . The gravity is  $\mathbf{g} = (0, -9.8)$  and the external force terms are

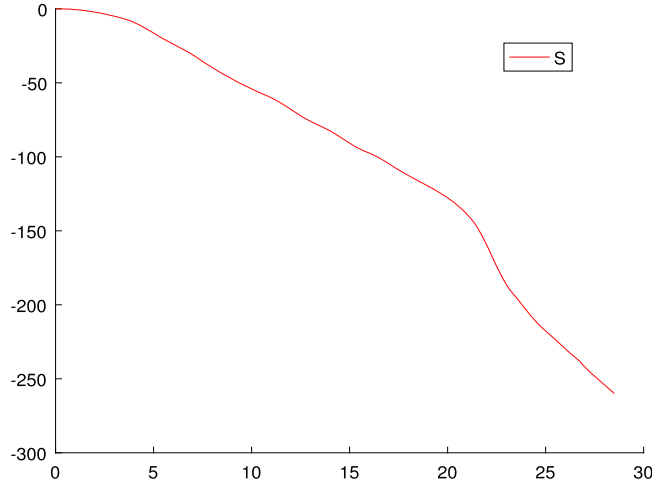
$$\mathbf{f} = \text{Reg}, \quad \mathbf{F}_s = \text{Re}M_s\mathbf{g}.$$

Densities of the fluid and the ellipse are  $\rho = 1$  and  $\rho_s = 2.7$ . In the previous numerical experiments, for simulations for the problem of a leaf or business card falling in air [33,25], the particle Reynolds number are around 1000 (for example as  $Re = 837, 1100$  in [25]). In the numerical experiment in this section, we simulate a problem with Reynolds number  $Re = 600$ . Other parameters in the system of equations are

$$L_s = 0.005, \quad \epsilon_p = 10^{-10}.$$

Fig. 17 shows the trajectory of the falling ellipse. We draw a hollow circle to represent the center, and a line segment to represent the long axis of the ellipse (see the left of Fig. 17 for a simple illustration). It is shown in Fig. 17 that the ellipse flutters for a while, then starts to tumble when it is near the bottom of the water. According to Theorem 4.2, the discrete system satisfies the following energy law:

$$\begin{aligned} E^{n+1} - E^n \leq & -\delta t \int_{\Omega \setminus P(t^{n+1})} \frac{1}{2} \|\tilde{\sigma}^{n+1}\|_F^2 dx - \delta t \int_{\partial\Omega} \frac{1}{L_s} |(\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_w) \cdot \boldsymbol{\tau}|^2 ds \\ & - \delta t \int_{\partial P(t^{n+1})} \frac{1}{L_s} |(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau}|^2 ds + \delta t \text{Re} \int_{\Omega \setminus P(t^{n+1})} \mathbf{g} \cdot \mathbf{u}^{n+1} dx + \delta t \text{Re} M_s \mathbf{g} \cdot \mathbf{U}_s^{n+1}, \end{aligned} \quad (6.3)$$



**Fig. 18.** History of  $S(t)$  over time.  $S(t)$  stands for the total energy minus the total work done by external forces at time  $t$ .

where  $y_s^{n+1}$  and  $y_s^n$  are the  $y$ -coordinates of the center of the ellipse at  $t^{n+1}$  and  $t^n$  respectively. Since there are external forces in this problem, by the energy law of the discrete system, total energy minus total work should decay over time. Define

$$S(t^n) = E^n - \delta t \operatorname{Re} \sum_{k=1}^n \int_{\Omega \setminus P(t^k)} \mathbf{g} \cdot \mathbf{u}^k dx - \delta t \operatorname{Re} M_s \mathbf{g} \cdot \mathbf{U}_s^k,$$

which is the total energy minus the total work done by external forces. It is shown in Fig. 18 that  $S$  decays over time throughout the calculation, which is consistent with the theoretical analysis.

## 7. Conclusions

We have developed an energy-stable time discretization scheme for the fluid-particle interaction problems. Theoretical analysis shows that the time discretization satisfies an energy law similar to that of the continuous system. The fully-discrete algorithm uses the XFEM method to solve the coupled system on a fixed mesh. The numerical method is first order accurate in time and space without being momentum conservative. Numerical experiments show that the algorithm is energy-stable and allows large time steps in practical application.

## CRediT authorship contribution statement

**Xiang Li:** Conceptualization, Methodology, Software, Writing - review & editing. **Qiang Du:** Conceptualization, Methodology, Writing - review & editing. **Li Luo:** Methodology, Writing - review & editing. **Xiao-Ping Wang:** Conceptualization, Methodology, Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. An equivalent numerical scheme by elimination

Applying mapping  $\hat{B}^{n+1}$  to both sides of the equation (4.7) in  $t^n$ , then multiplying both sides by  $(J^{n+1})^{\frac{1}{2}}$ , we have

$$\operatorname{Re} \frac{\mathbf{u}_{**}^n - \tilde{\mathbf{u}}_{**}^n}{\delta t} = \operatorname{Re} \frac{\mathbf{u}_{**}^n - \tilde{\mathbf{u}}_{**}^n}{\delta t} (J^{n+1})^{\frac{1}{2}} = -(\nabla p^n)_* (J^{n+1})^{\frac{1}{2}} \text{ in } \Omega \setminus P(t^{n+1}). \quad (\text{A.1})$$

From (4.9) and (4.10) we have

$$\operatorname{Re} M_s \frac{\mathbf{U}_s^n - \tilde{\mathbf{U}}_s^n}{\delta t} = - \int_{\partial P(t^n)} -p^n \mathbf{I} \cdot \mathbf{n} ds, \quad (\text{A.2})$$

$$\mathcal{R}e I_s \frac{\omega_s^n - \tilde{\omega}_s^n}{\delta t} = - \int_{\Omega \setminus P(t^n)} \mathbf{r}_f^n \times (-p^n \mathbf{I} \cdot \mathbf{n}) ds. \quad (\text{A.3})$$

Summing (A.1) and (4.2), (A.2) and (4.3) and (A.3) and (4.4) respectively, we can eliminate  $\mathbf{u}^{n+1}$  in the NS equations and Newton's law:

$$\begin{aligned} & \mathcal{R}e \left[ \frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_\star^n}{\delta t} + ((\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1}) \cdot \nabla) \tilde{\mathbf{u}}^{n+1} \right] \\ &= -(\nabla p^n)_\star (J^{n+1})^{\frac{1}{2}} + \nabla \cdot \tilde{\boldsymbol{\sigma}}^{n+1} - \frac{1}{2} \mathcal{R}e (\nabla \cdot (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1})) \tilde{\mathbf{u}}^{n+1} + \mathbf{f}^{n+1}, \\ & \mathcal{R}e M_s \frac{\tilde{\mathbf{u}}_s^{n+1} - \tilde{\mathbf{u}}_s^n}{\delta t} = - \int_{\partial P(t^n)} -p^n \mathbf{I} \cdot \mathbf{n} ds - \int_{\partial P(t^{n+1})} \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n} ds + \mathbf{F}_s^{n+1}, \\ & \mathcal{R}e I_s \frac{\tilde{\omega}_s^{n+1} - \tilde{\omega}_s^n}{\delta t} = - \int_{\partial P(t^n)} \mathbf{r}_f^n \times (-p^n \mathbf{I} \cdot \mathbf{n}) ds - \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) ds. \end{aligned}$$

Similarly, we can eliminate  $\mathbf{u}^{n+1}$  in step II of Algorithm 4.1. Apply a divergence operator to both sides of (4.7). Since  $\mathbf{u}^{n+1}$  is incompressible, we have

$$-\Delta p^{n+1} = -\frac{\mathcal{R}e}{\delta t} \nabla \cdot \tilde{\mathbf{u}}^{n+1}.$$

Multiplying equation (4.7) by  $\mathbf{n}$  on  $\partial P(t^{n+1})$  yields the following boundary condition of the above Poisson's equation:

$$-\frac{\partial p^{n+1}}{\partial n} = \frac{\mathcal{R}e}{\delta t} (\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}) \cdot \mathbf{n} \text{ on } \partial P(t^{n+1}).$$

According to the boundary conditions (4.6) and (4.12), and equations (4.9) and (4.10), we can derive the boundary condition of  $p^{n+1}$  on  $\partial P(t^{n+1})$ :

$$\begin{aligned} -\frac{\partial p^{n+1}}{\partial n} &= \frac{\mathcal{R}e}{\delta t} (\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^{n+1}) \cdot \mathbf{n} = \frac{\mathcal{R}e}{\delta t} (\mathbf{u}_s^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \mathbf{n} \\ &= -\frac{1}{M_s} \left( \int_{\partial P(t^{n+1})} -p^{n+1} \mathbf{I} \cdot \mathbf{n} ds \right) \cdot \mathbf{n} \\ &\quad - \frac{1}{I_s} \left( \left( \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (-p^{n+1} \mathbf{I} \cdot \mathbf{n}) ds \times \mathbf{r}_f^{n+1} \right) \cdot \mathbf{n} \right). \end{aligned} \quad (\text{A.4})$$

The boundary condition of  $p^{n+1}$  on  $\partial \Omega$  can be similarly derived:

$$\frac{\partial p^{n+1}}{\partial n} = 0.$$

Collecting the above equations, we have an equivalent algorithm to Algorithm 4.1.

**Algorithm A.1** (Modified energy-stable scheme with  $\mathbf{u}^{n+1}$  eliminated).

**Step I:** Given  $(\tilde{\mathbf{u}}^n, p^n, \tilde{\mathbf{U}}_s^n, \tilde{\omega}_s^n, \mathbf{X}_s^n, \theta_s^n)$  at time step  $t^n$ , solve the following coupled system (a)-(c) simultaneously to obtain  $(\tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{U}}_s^{n+1}, \tilde{\omega}_s^{n+1}, \mathbf{X}_s^{n+1}, \theta_s^{n+1})$  at  $t^{n+1}$ :

(a) Calculate particle position and angular orientation by

$$\begin{cases} \mathbf{X}_s^{n+1} = \mathbf{X}_s^n + \tilde{\mathbf{U}}_s^{n+1} \delta t, \\ \theta_s^{n+1} = \theta_s^n + \tilde{\omega}_{s,z}^{n+1} \delta t. \end{cases}$$

(b) Calculate the tALE velocity field in  $\Omega \setminus P(t^{n+1})$ :

$$\begin{cases} -\Delta \tilde{\mathbf{u}}^{n+1} = 0, & \text{in } \Omega \setminus P(t^{n+1}), \\ \tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}}_s^{n+1}, & \text{on } \partial P(t^{n+1}), \\ \tilde{\mathbf{u}}^{n+1} = 0, & \text{on } \partial \Omega. \end{cases} \quad (\text{A.5})$$

Bijjective mapping  $\hat{B} : \Omega \setminus P(t^{n+1}) \rightarrow \Omega \setminus P(t^n)$ , and functions  $\tilde{\mathbf{u}}_\star^n$  and  $\tilde{\mathbf{u}}_{\star\star}^n$  are defined in the same way as they are in Algorithm 4.1.

(c) Solve the NS equations and Newton's law:

$$\begin{aligned} & \mathcal{R}e \left[ \frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_\star^n}{\delta t} + ((\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1}) \cdot \nabla) \tilde{\mathbf{u}}^{n+1} \right] \\ &= -(\nabla p^n)_\star J^{\frac{1}{2}} + \nabla \cdot \tilde{\boldsymbol{\sigma}}^{n+1} - \frac{1}{2} \mathcal{R}e(\nabla \cdot (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1})) \tilde{\mathbf{u}}^{n+1} + \mathbf{f}^{n+1}, \end{aligned} \quad (\text{A.6})$$

$$\mathcal{R}e M_s \frac{\tilde{\mathbf{u}}_s^{n+1} - \tilde{\mathbf{u}}_s^n}{\delta t} = - \int_{\partial P(t^n)} -p^n \mathbf{I} \cdot \mathbf{n} ds - \int_{\partial P(t^{n+1})} \tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n} ds + \mathbf{F}_s^{n+1}, \quad (\text{A.7})$$

$$\mathcal{R}e I_s \frac{\tilde{\boldsymbol{\omega}}_s^{n+1} - \tilde{\boldsymbol{\omega}}_s^n}{\delta t} = - \int_{\partial P(t^n)} \mathbf{r}_f^n \times (-p^n \mathbf{I} \cdot \mathbf{n}) ds - \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) ds. \quad (\text{A.8})$$

Boundary conditions on  $\partial\Omega$  are

$$\tilde{\mathbf{u}}^{n+1} \cdot \mathbf{n} = 0, \quad \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \mathbf{u}_w) \cdot \boldsymbol{\tau} = -(\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}. \quad (\text{A.9})$$

Boundary conditions on  $\partial P(t^{n+1})$  are

$$(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \mathbf{n} = 0, \quad \frac{1}{L_s} (\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}_s^{n+1}) \cdot \boldsymbol{\tau} = -(\tilde{\boldsymbol{\sigma}}^{n+1} \cdot \mathbf{n}) \cdot \boldsymbol{\tau}. \quad (\text{A.10})$$

**Step II:** Given  $(\tilde{\mathbf{u}}^{n+1}, \tilde{\mathbf{u}}_s^{n+1}, \tilde{\boldsymbol{\omega}}_s^{n+1}, \mathbf{X}_s^{n+1}, \theta_s^{n+1})$ , solve  $p^{n+1}$  in  $\Omega \setminus P(t^{n+1})$ :

$$-\Delta p^{n+1} = -\frac{\mathcal{R}e}{\delta t} \nabla \cdot \tilde{\mathbf{u}}^{n+1}. \quad (\text{A.11})$$

The boundary condition on  $\partial\Omega$  is

$$\frac{\partial p^{n+1}}{\partial n} = 0. \quad (\text{A.12})$$

The boundary condition on  $\partial P(t^{n+1})$  is

$$\begin{aligned} -\frac{\partial p^{n+1}}{\partial n} &= -\frac{1}{M_s} \left( \int_{\partial P(t^{n+1})} -p^{n+1} \mathbf{I} \cdot \mathbf{n} ds \right) \cdot \mathbf{n} \\ &\quad - \frac{1}{I_s} \left( \left( \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (-p^{n+1} \mathbf{I} \cdot \mathbf{n}) ds \right) \times \mathbf{r}_f^{n+1} \right) \cdot \mathbf{n}. \end{aligned} \quad (\text{A.13})$$

**Remark A.1.** For an arbitrary test function  $q$ , boundary condition (A.4) yields

$$\begin{aligned} \int_{\partial P(t^{n+1})} -\frac{\partial p^{n+1}}{\partial n} q ds &= \frac{1}{M_s} \left( \int_{\partial P(t^{n+1})} -p^{n+1} \mathbf{I} \cdot \mathbf{n} ds \right) \cdot \left( \int_{\partial P(t^{n+1})} -q^{n+1} \mathbf{I} \cdot \mathbf{n} ds \right) \\ &\quad + \frac{1}{I_s} \left( \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (-p^{n+1} \mathbf{I} \cdot \mathbf{n}) ds \right) \cdot \left( \int_{\partial P(t^{n+1})} \mathbf{r}_f^{n+1} \times (-q^{n+1} \mathbf{I} \cdot \mathbf{n}) ds \right). \end{aligned}$$

Thus the Robin-like boundary condition (A.4) may be applied using the above symmetric inner product in the FEM weak form.

## References

- [1] A. Andersen, U. Pesavento, Z.J. Wang, Analysis of transitions between fluttering, tumbling and steady descent of falling cards, *J. Fluid Mech.* 541 (2005) 91–104, <https://doi.org/10.1017/S0022112005005847>.
- [2] A. Andersen, U. Pesavento, Z.J. Wang, Unsteady aerodynamics of fluttering and tumbling plates, *J. Fluid Mech.* 541 (2005) 65–90, <https://doi.org/10.1017/S002211200500594X>.

- [3] S. Badia, A. Quaini, A. Quarteroni, Splitting methods based on algebraic factorization for fluid-structure interaction, *SIAM J. Sci. Comput.* 30 (2008) 1778–1805, <https://doi.org/10.1137/070680497>.
- [4] J.W. Banks, W.D. Henshaw, D.W. Schwendeman, Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part I: model problem analysis, *J. Comput. Phys.* 343 (2017) 432–468, <https://doi.org/10.1016/j.jcp.2017.01.015>.
- [5] J.W. Banks, W.D. Henshaw, D.W. Schwendeman, Q. Tang, A stable partitioned FSI algorithm for rigid bodies and incompressible flow. Part II: general formulation, *J. Comput. Phys.* 343 (2017) 469–500, <https://doi.org/10.1016/j.jcp.2017.01.015>.
- [6] J.W. Banks, W.D. Henshaw, B. Sjögreen, A stable FSI algorithm for light rigid bodies in compressible flow, *J. Comput. Phys.* 245 (2013) 399–430, <https://doi.org/10.1016/j.jcp.2013.02.050>.
- [7] A. Belmonte, H. Eisenberg, E. Moses, From flutter to tumble: inertial drag and Froude similarity in falling paper, *Phys. Rev. Lett.* 81 (1998) 345, <https://doi.org/10.1103/PhysRevLett.81.345>.
- [8] I. Borazjani, L. Ge, F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies, *J. Comput. Phys.* 227 (2008) 7587–7620, <https://doi.org/10.1016/j.jcp.2008.04.028>.
- [9] F.P. Bretherton, The motion of rigid particles in a shear flow at low Reynolds number, *J. Fluid Mech.* 14 (1962) 284–304, <https://doi.org/10.1017/S002211206200124X>.
- [10] Y.J. Choi, M.A. Hulsen, H.E. Meijer, An extended finite element method for the simulation of particulate viscoelastic flows, *J. Non-Newton. Fluid Mech.* 165 (2010) 607–624, <https://doi.org/10.1016/j.jnnfm.2010.02.021>.
- [11] E. Ding, C.K. Aidun, The dynamics and scaling law for particles suspended in shear flow with inertia, *J. Fluid Mech.* 423 (2000) 317–344.
- [12] A. Gerstenberger, W.A. Wall, An extended finite element method/Lagrange multiplier based approach for fluid-structure interaction, *Comput. Methods Appl. Mech. Eng.* 197 (2008) 1699–1714, <https://doi.org/10.1016/j.cma.2007.07.002>.
- [13] R. Glowinski, Finite element methods for incompressible viscous flow, *Handb. Numer. Anal.* 9 (2003) 3–1176.
- [14] R. Glowinski, T. Pan, T.I. Hellsa, D.D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, *J. Comput. Phys.* 169 (2001) 363–426.
- [15] R. Glowinski, T. Pan, J. Periaux, Fictitious domain methods for incompressible viscous flow around moving rigid bodies, *Math. Finite Elem. Appl.* 9 (1996) 155–174.
- [16] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, *Int. J. Multiph. Flow* 25 (1999) 755–794, [https://doi.org/10.1016/S0301-9322\(98\)00048-2](https://doi.org/10.1016/S0301-9322(98)00048-2).
- [17] R. Glowinski, T.-W. Pan, J. Periaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, *Comput. Methods Appl. Mech. Eng.* 151 (1998) 181–194, [https://doi.org/10.1016/S0045-7825\(97\)00116-3](https://doi.org/10.1016/S0045-7825(97)00116-3).
- [18] J.-L. Guermond, P. Mineev, J. Shen, An overview of projection methods for incompressible flows, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 6011–6045, <https://doi.org/10.1016/j.cma.2005.10.010>.
- [19] J.-L. Guermond, A. Salgado, A splitting method for incompressible flows with variable density based on a pressure Poisson equation, *J. Comput. Phys.* 228 (2009) 2834–2846, <https://doi.org/10.1016/j.jcp.2008.12.036>.
- [20] H.H. Hu, Direct simulation of flows of solid-liquid mixtures, *Int. J. Multiph. Flow* 22 (1996) 335–352, [https://doi.org/10.1016/0301-9322\(95\)00068-2](https://doi.org/10.1016/0301-9322(95)00068-2).
- [21] H.H. Hu, D.D. Joseph, M.J. Crochet, Direct simulation of fluid particle motions, *Theor. Comput. Fluid Dyn.* 3 (1992) 285–306, <https://doi.org/10.1007/BF00717645>.
- [22] H.H. Hu, N.A. Patankar, M. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, *J. Comput. Phys.* 169 (2001) 427–462, <https://doi.org/10.1006/jcph.2000.6592>.
- [23] M. Ishii, T. Hibiki, *Thermo-Fluid Dynamics of Two-Phase Flow*, Springer Science & Business Media, 2010.
- [24] G.B. Jeffery, The motion of ellipsoidal particles immersed in a viscous fluid, *Proc. R. Soc. A* 102 (1922) 161–179, <https://doi.org/10.1098/rspa.1922.0078>.
- [25] C. Jin, K. Xu, Numerical study of the unsteady aerodynamics of freely falling plates, *Commun. Comput. Phys.* 3 (2008) 834–851.
- [26] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, *J. Comput. Phys.* 231 (2012) 3663–3684, <https://doi.org/10.1016/j.jcp.2012.01.021>.
- [27] U. Lācis, K. Taira, S. Bagheri, A stable fluid–structure-interaction solver for low-density rigid bodies using the immersed boundary projection method, *J. Comput. Phys.* 305 (2016) 300–318, <https://doi.org/10.1016/j.jcp.2015.10.041>.
- [28] T. Lundquist, J. Nordstrom, The SBP-SAT technique for initial value problems, *J. Comput. Phys.* 270 (2014) 86–104.
- [29] L. Mahadevan, W.S. Ryu, A.D. Samuel, Tumbling cards, *Phys. Fluids* 11 (1999) 1–3, <https://doi.org/10.1063/1.869919>.
- [30] F. Nobile, L. Formaggia, A stability analysis for the arbitrary Lagrangian: Eulerian formulation with finite elements, *East-West J. Numer. Math.* 7 (1999) 105–132.
- [31] J. Nordstrom, S. Eriksson, Fluid structure interaction problems: the necessity of a well posed, stable and accurate formulation, *Commun. Comput. Phys.* 8 (2010) 1111–1138.
- [32] J. Nordstrom, T. Lundquist, Summation-by-parts in time, *J. Comput. Phys.* 251 (2013) 487–499.
- [33] U. Pesavento, Z.J. Wang, Falling paper: Navier-Stokes solutions, model of fluid forces, and center of mass elevation, *Phys. Rev. Lett.* 93 (2004) 144501, <https://doi.org/10.1103/PhysRevLett.93.144501>.
- [34] O. Pironneau, An energy stable monolithic Eulerian fluid-structure numerical scheme, *Chin. Ann. Math., Ser. B* 39 (2018) 213–232, <https://doi.org/10.1007/s11401-018-1061-9>.
- [35] A. Robinson-Mosher, C. Schroeder, R. Fedkiw, A symmetric positive definite formulation for monolithic fluid structure interaction, *J. Comput. Phys.* 230 (2011) 1547–1566, <https://doi.org/10.1016/j.jcp.2010.11.021>.
- [36] H. Tanaka, T. Araki, Simulation method of colloidal suspensions with hydrodynamic interactions: fluid particle dynamics, *Phys. Rev. Lett.* 85 (2000) 1338, <https://doi.org/10.1103/PhysRevLett.85.1338>.
- [37] G.I. Taylor, The motion of ellipsoidal particles in a viscous fluid, *Proc. R. Soc. Lond. Ser. A* 103 (1923) 58–61, <https://doi.org/10.1098/rspa.1923.0040>.
- [38] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2005) 448–476, <https://doi.org/10.1016/j.jcp.2005.03.017>.
- [39] J. Vierendeels, K. Dumont, E. Dick, P. Verdonck, Analysis and stabilization of fluid-structure interaction algorithm for rigid-body motion, *AIAA J.* 43 (2005) 2549–2557, <https://doi.org/10.2514/1.3660>.
- [40] G. Wagner, N. Moës, W. Liu, T. Belytschko, The extended finite element method for rigid particles in Stokes flow, *Int. J. Numer. Methods Eng.* 51 (2001) 293–313, <https://doi.org/10.1002/nme.169>.
- [41] D. Wan, S. Turek, Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method, *Int. J. Numer. Methods Fluids* 51 (2006) 531–566.
- [42] D. Wan, S. Turek, Fictitious boundary and moving mesh methods for the numerical simulation of rigid particulate flows, *J. Comput. Phys.* 222 (2007) 28–56.
- [43] C. Wang, J.D. Eldredge, Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method, *J. Comput. Phys.* 295 (2015) 87–113, <https://doi.org/10.1016/j.jcp.2015.04.005>.
- [44] Y. Wang, P.K. Jimack, M.A. Walkley, An energy stable one-field fictitious domain method for fluid-structure interactions, *arXiv preprint*, arXiv:1801.09264, 2018.

- [45] C. Wood, A. Gil, O. Hassan, J. Bonet, Partitioned block-Gauss-Seidel coupling for dynamic fluid-structure interaction, *Comput. Struct.* 88 (2010) 1367–1382, <https://doi.org/10.1016/j.compstruc.2008.08.005>.
- [46] J. Yang, F. Stern, A non-iterative direct forcing immersed boundary method for strongly-coupled fluid-solid interactions, *J. Comput. Phys.* 295 (2015) 779–804, <https://doi.org/10.1016/j.jcp.2015.04.040>.
- [47] D. Zhang, A. Prosperetti, Averaged equations for inviscid disperse two-phase flow, *J. Fluid Mech.* 267 (1994) 185–219, <https://doi.org/10.1017/S0022112094001151>.
- [48] J. Zhang, X. Xu, T. Qian, Anisotropic particle in viscous shear flow: Navier slip, reciprocal symmetry, and Jeffery orbit, *Phys. Rev. E* 91 (2015) 033016, <https://doi.org/10.1103/PhysRevE.91.033016>.
- [49] X. Zhang, L. Luo, X. Wang, A numerical study of fluid-particle interaction with slip boundary condition, *Numer. Math., Theory Methods Appl.* 11 (2018) 795–809, <https://doi.org/10.4208/nmtma.2018.s07>.