"Fluids tell solids how to bend, solids tell fluids how to move."

S. Succi, readapted from J. Wheeler

1

Introduction

From cell division to space exploration, interactions between solids and fluids are present across all scales in many natural, biological, and engineering systems. The coupled dynamics between immersed solids and surrounding fluids is known as fluid–structure interaction (FSI). Surprisingly, or perhaps not, FSI is everywhere. We see it in beating cilia [1], flowing cells [2], falling seeds [3], flapping flags [4], flying birds [5], and swimming fish [6]. We live because of it, starting from embryo morphogenesis to modern technologies such as microfluidic devices and international flights.

However, the natural complexity of FSI has left many intriguing yet unsolved scientific puzzles. Finding answers to questions like how the oscillatory motion of microswimmers leads to propulsion and col-

lective motion [7, 8], how the geometric difference between cancerous and healthy cells affects the hemodynamics [9, 10], and how fish school and utilize vortex to improve thrust efficiency [11, 12], would not only deepen our comprehension of the physical world, but shed insights applicable to diverse research fields, such as active matter, mammalian fertilization, medical devices, and bio-inspired engineering.

The essence of FSI is elegantly captured by the readaption of S. Succi [13] after J. Wheeler's famous quote: "Spacetime tells matter how to move; matter tells spacetime how to curve" [14]. While Wheeler's original words describe how matter and spacetime are interconnected in general relativity, this interplay is equivalently critical in FSI. For instance, let us consider a simple scenario: boiling one tangyuan in water, where a squishy solid is immersed in fluids. This soft glutinous rice ball can be deformed under the influence of surrounding water; meanwhile, the geometric shape change can induce stress onto fluids and change the fluid motion in return. Yet, even this simple configuration is already challenging to study both theoretically and experimentally (perhaps even culinarily), let alone considering additional physics like heat transfer and multi-body contact. After all, we never really just boil one tangyuan to eat.

With the rapid evolution of computational science, multiphysics simulations have become attractive complements to studying the coupling between solids and fluids, as well as other relevant physics [15]. Many commercial and open-source software are available for simulating FSI, such as `Ansys Fluent`, `COMSOL Multiphysics`, `OpenFOAM`, `AMReX`, `FEniCS`, and `Project Chrono`, among others. While these off-the-shelf software packages can generate reliable simulations, researchers often encounter FSI problems beyond the functionality of these packages, leading to the development of custom numerical meth-

ods tailored to address their particular research questions.

However, developing FSI methods is a nontrivial task. The main challenge stems from the intrinsic dichotomy in their preferred simulation approach. Solid stress is induced by strain, and it is easier to compute solid stress via a Lagrangian mesh which conforms to the solid geometry and moves with it. Therefore, solid simulations often use Lagrangian approaches [16–19]. In contrast, fluid stress is induced by strain rate, favoring the use of a fixed Eulerian mesh to compute fluid stress. Consequently, fluid simulations often use Eulerian methods [20–22]. Other simulation challenges include modeling material and geometric nonlinearity in solids, modeling multi-body interactions, and integrating additional physics like heat transfer, surface tension, and experimental protocols. In the era of high-performance and exascale computing [15], considerations such as parallelization, scalability, robustness, and generalizability to many multiphysics problems are also pivotal in the development of FSI methods.

1.1 State-of-the-art numerical methods for fluid–structure interaction

Many numerical methods have successfully addressed the aforementioned simulation challenges, suitable for a range of FSI applications, such as microswimmer biofluid mechanics [1, 23], blood cell simulation [2, 24, 25], and biolocomotion modeling [5, 26]. Although solids and fluids have their preferred discretization frameworks—moving Lagrangian or fixed Eulerian—they do not necessarily have to be discretized in that framework. Depending on their chosen framework for discretizing continuum fields, FSI methods can be loosely categorized into four types (Fig. 1.1), which are (A) mesh-free, (B) Lagrangian, (C) Eulerian–Lagrangian, and (D) Eulerian.

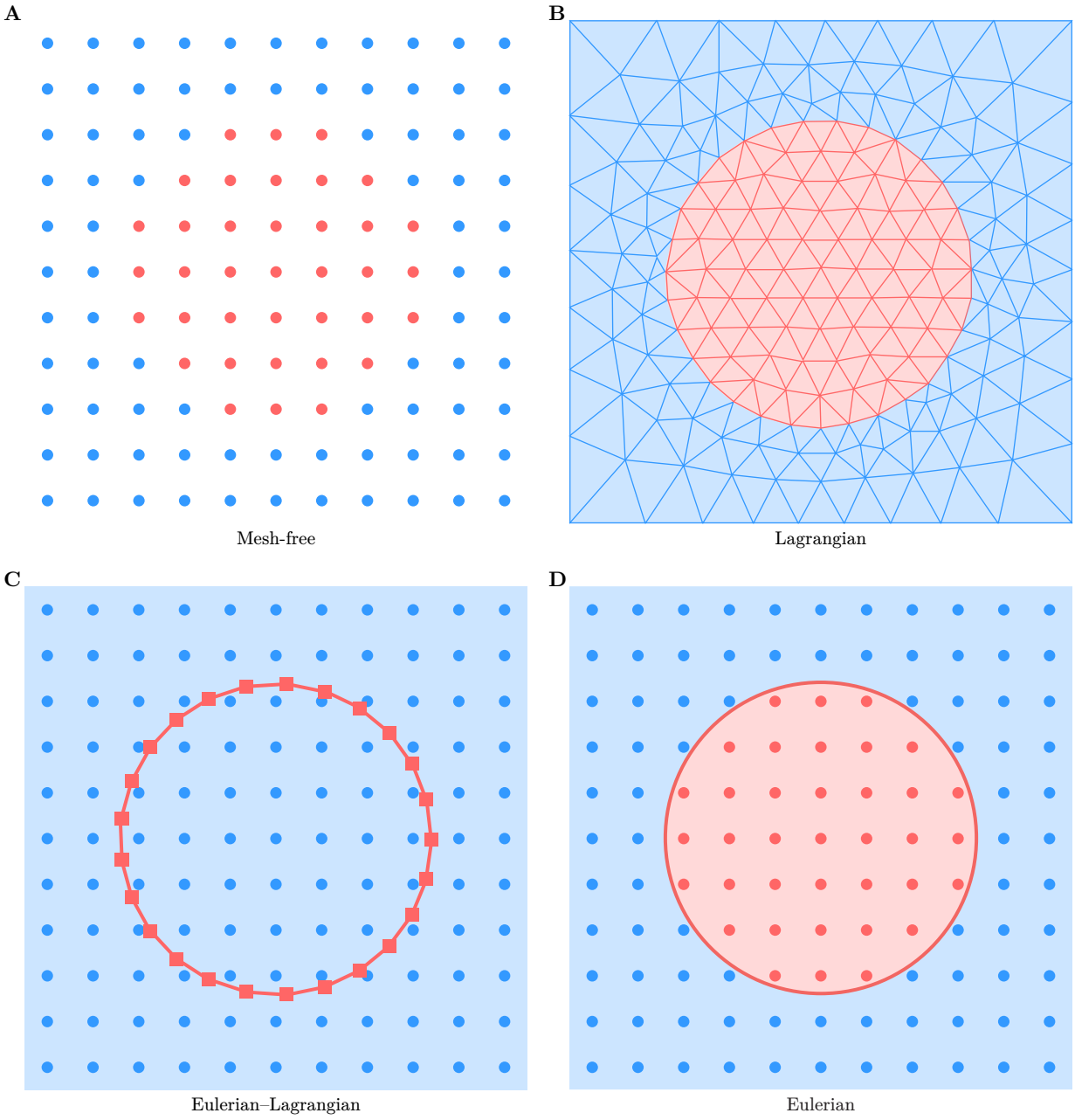

Figure 1.1: Illustration of fluid–structure interaction (FSI) numerical methods. Types of FSI numerical methods based on different choices of solid and fluid discretization frameworks. **(A)** Mesh-free methods use particles to represent both phases. **(B)** Lagrangian methods use unstructured adaptive meshes for both solids and fluids. **(C)** Eulerian–Lagrangian methods use a fixed Eulerian mesh for fluids but moving Lagrangian markers or finite-element meshes for solids. **(D)** Eulerian methods only use one fixed computational grid for both phases.

The first type is *mesh-free* (Fig. 1.1A). It uses points to represent both solids and fluids without relying on a mesh. These points, distinct from actual atoms or molecules as seen in molecular dynamics (MD) simulations [27, 28], constitute a set of Lagrangian elements forming a continuum. While these points may be referred to by various names, such as "particles", "material points", or "elements", they operate on the same principle, carrying the physical properties of the fluid or solid and being free to move within the simulation domain. Mesh-free methods [29] offer advantages such as adaptive resolution and numerical mass conservation, making them attractive for handling large unbounded domains with significant density variations, free surfaces, or extreme deformations. They are well-suited for parallelization, as each point operates independently and only interacts with its neighboring points. One popular example is smoothed particle hydrodynamics (SPH) [30–32], which uses a kernel function to smoothly convolve the physical properties of neighboring particles to compute the continuum fields. Another example, originally proposed for computational solid mechanics but perhaps more known for its appearance in animated movies such as Disney's *Frozen* and *Moana*, is the material point method (MPM) [17, 33–37]. It employs an additional on-the-fly Eulerian grid to extrapolate forces between material points and the grid, enabling physically accurate simulations of snow, water, and other granular materials. However, constructing the connectivity between points to compute stress fields or determining the interface between phases can present a computational hurdle for mesh-free methods.

The second type is *Lagrangian* (Fig. 1.1B). It uses unstructured adaptive Lagrangian meshes to discretize both phases. Lagrangian methods typically employ a moving non-overlapping computational

grid, with a finite-element representation for the solid and an adaptive Lagrangian mesh for the fluid. Examples include the arbitrary Lagrangian–Eulerian (ALE) method [38–40] and finite-element procedures [41, 42]. Consequently, they provide a body-fitted representation of solid geometry and can adaptively update the mesh based on solid motion. However, this adaptivity is also a double-edged sword, as the computational cost of remeshing at each time step can be significant, particularly for very large deformation or slender structures. Additionally, implementing the fluid incompressibility constraint can be tricky, given that the fluid mesh is not a regular fixed grid. Special handling is also required to regularize the mesh for numerical stability and to prevent mesh entanglement.

The third type is *Eulerian–Lagrangian* (Fig. 1.1C). As the name suggests, it combines the two frameworks. It uses a fixed background Eulerian grid for fluids, overlaid with Lagrangian markers or finite-element meshes for solids [43–45]. By keeping both phases in their preferred discretization frameworks, Eulerian–Lagrangian methods are often used to track deformable solids or solids with prescribed kinematics in fluids. One widely used example is the family of immersed methods [45], such as the immersed boundary method (IBM) [44–48] and immersed interface method (IIM) [49–51]. One interesting historical note is that the IBM was introduced by C. Peskin in 1972 as part of his PhD thesis to model flow in heart valves [52–54]. The IBM enables two-way coupling between the Eulerian and Lagrangian variables via interpolation kernel functions on the integral equations of solid and fluid dynamics. The IIM is a discretized form of the IBM integral formulation, suitable for modeling problems with large jump conditions across the solid–fluid interface. With more than half a century of development, the immersed

methods have been widely used in various applications, such as simulating swimming and flying locomotion [55–59], cardiovascular dynamics [60–62], and flows of cells and capsules [63–65]. Some vanilla versions of the IBM suffer from volume conservation issues [66, 67], whereby fluids *leak* at the interface of enclosed pressurized solid chambers, causing their volume to decrease over time. These issues have also been further addressed in recent years [68, 69]. Eulerian–Lagrangian methods have become very popular in biological and medical applications, for prediction, prevention, and design. However, they are still less straightforward to implement given there are two frameworks, Eulerian and Lagrangian, that we need to keep track of. In addition, the communication via the interpolation kernel functions can be computationally expensive, especially for a large number of markers or elements.

The last type is *Eulerian* (Fig. 1.1D). By coupling solids and fluids on one fixed computational grid, Eulerian methods offer several advantages. They eliminate the need for extra computations in remeshing solid geometries or communication between Lagrangian and Eulerian frameworks. Additionally, they facilitate easier implementation of convergence and stability analysis, parallelization, and the incompressibility constraint. Stress, force, and velocity fields can be computed directly on the fixed grid, also simplifying the implementation of multi-body contact. The primary numerical challenge for a fully-Eulerian FSI method lies in describing the solid in an Eulerian framework, as Eulerian methods are well-established in computational fluid dynamics (CFD), such as the family of finite-difference (FDM) [70, 71], finite-element (FEM) [72, 73], and finite-volume methods (FVM) [74, 75], as well as the lattice Boltzmann method (LBM) [13, 76]. One approach is to describe the solid geometry via a level set function to track

its topological changes, such as the level set method (LSM) [77, 78]. Other approaches consider the solid under two limits: small-strain or finite-strain. The small-strain approximation of linear elasticity can be used to model hypoelastic solids [79], as seen in modeling sharp interfaces with shock-capturing [80] and shear transformation zones in bulk metallic glass [81–83]. To model finite-strain solids undergoing large deformation, a hyperelastic formulation of the solid stress is a typical approach [84]. Many methods choose different key simulation variables to formulate the Eulerian solid stress, such as the deformation gradient tensor [85–87], the left Cauchy–Green deformation tensor [88, 89], the displacement field [90–92], and the reference map field [93, 94]. Later in Chapter 3, we focus on the reference map technique (RMT) [95–102], which uses the reference map field to compute solid stress. Incidentally, the RMT was proposed by K. Kamrin in 2008 in his PhD thesis [93] to model granular flow. Eulerian methods are simple to implement and can be computationally efficient, but they still face challenges in modeling free surfaces or maintaining an accurate body-fitted representation of the solid geometry.

FSI methods will continue to evolve and develop, especially with the yearly upgrade of computing hardware and the growing popularity of data-driven approaches [103]. The aforementioned methods are not an exhaustive list of all FSI methods, but they highlight the basic idea of computational physics, *i.e.* discretizing solids and fluids using frameworks such as mesh-free, Lagrangian, Eulerian, or a combination of both. Each method has its own strengths and weaknesses, and the choice of method depends on the specific FSI problem of interest. Complementary to one another, these methods form a powerful toolbox for researchers to explore the rich phenomena in FSI, from the microscopic to the macroscopic scale.

1.2 Contribution of this thesis

This thesis explores multiphysics simulation of fluid–structure interaction from three perspectives: experimental, numerical, and artistic. Each chapter constitutes a self-contained project focusing on one of these aspects. Given the simulation-centric nature of this thesis, we develop computational frameworks to model one-way and two-way FSI problems of interest. Specifically, these methods fall into the *Eulerian* category (Fig. 1.1D), where solids and fluids are coupled on one fixed computational grid.

In Chapter 2, we examine one-way FSI in the context of cryo-plunging experiments. In cryogenic electron microscopy (cryo-EM), plunge freezing is a critical step to preserve thin biological samples in vitreous ice for imaging. The cooling rate has been indicative of vitrification quality. However, the role of fluid dynamics has long been neglected in this step. To investigate its coupling with thermodynamics during cryo-plunging, we develop a computational framework, `cryoflo`, that prescribes a moving rigid hot solid with experimental protocols and models its heat transfer with cold fluid. This framework, integrated with block-structured adaptive mesh refinement software and experimental measurements, enables a three-dimensional (3D) "digital twin" of cryo-plunging. It is highly parallelizable and validated against experimental thermocouple data. Our 3D simulations also provide a virtual probe to visualize heat transfer and explore the effects of plunging protocols on cooling rate. `cryoflo` is the first step towards building a time-resolved monitoring system and engineering fluid dynamics for improved cryo-vitrification.

In Chapter 3, we introduce the LBRMT, a fully-integrated lattice Boltzmann (LB) method to model two-way FSI of deformable solids in fluids. The LB method, widely used in computational physics, mod-

els fluid motion via a mesoscopic probabilistic view of particle motion based on kinetic theory. Despite its popularity, there is no well-established method to model finite-strain solid mechanics on the same fixed Eulerian grid. To address this gap, we extend the LB method to model finite-strain solids with the reference map technique (RMT). An integral part of the LBRMT is a new LB boundary condition for moving deformable interfaces across different densities. The LBRMT represents a new capability for LB simulations to couple solids undergoing large deformation with fluids, adept at simulating soft solids rotating and settling. Being fully explicit and well-suited for parallelization, the LBRMT enables us to solve highly challenging FSI problems, such as the mixing of several hundred flexible objects, using modest computational resources. It is a powerful tool for studying many-body interactions and complex geometries, exemplified in collective behavior in active matter and biofluid dynamics.

In Chapter 4, we explore the hydrodynamics of marbling art. Marbling is an ancient art where marblers create intricate patterns by dropping paints onto a viscosified fluid surface and then using thin tools to make designs. It has fascinated artists and scientists alike, but the underlying fluid dynamics remain poorly understood. Through a gallery of marbling patterns, we develop a physical explanation of why marbling works and investigate the roles of interfacial tension and surfactants. We also propose a computational framework inspired by the RMT to simulate marbling while capturing sharp color interfaces.

1.2.1 Manuscripts and open-source software code

Software code is an indispensable component of any simulation project. All the numerical methods developed in this thesis are implemented in `C++` and visualized in `Python`. We believe that open-source

software is crucial to the ecosystem of scientific research. The manuscripts, open-source software code, and movies included in this thesis are:

Chapter 2: `cryoflo`: https://github.com/yue-sun/cryoflo

Three-dimensional simulation of sample vitrification in cryo-plunging,

Yue Sun, Simon Merminod, Arvind K. Srinivasan, Maxim B. Prigozhin, and Chris H. Rycroft,

in preparation (2024).

Cryo-vitrification with millisecond time-resolution,

in preparation (2024).

Chapter 3: `LBRMT`: https://github.com/yue-sun/lbrmt

A fully-integrated lattice Boltzmann method for fluid–structure interaction,

Yue Sun and Chris H. Rycroft,

under review at Journal of Computational Physics (2024). (arXiv:2402.12696)

Chapter 4: `marbling-fluids`: https://github.com/yue-sun/marbling-fluids

The hydrodynamics of marbling art,

Yue Sun, John W. M. Bush, Saverio E. Spagnolie, and Chris H. Rycroft,

in preparation for an invited short paper in a special collection of Physics Review Fluids (2024).

1.2.2 Movies and visualizations

If a picture is worth a thousand words, what is a video worth? Scientific visualization has always been a key focus of our research. As another indispensable component of any simulation project, it not only helps us to visualize and contextualize the complex phenomena in FSI, but also makes FSI research more approachable to a broader audience. For each chapter, we provide simulation movies via their respective GitHub repositories (see `README.md` or `videos` directory).

To celebrate the beauty of fluid dynamics, we have submitted two entries to the Gallery of Fluid Motion (GFM), held each November during the American Physical Society's Division of Fluid Dynamics annual meeting. The first video focuses on the 3D visualization of soft immersed rods. Additionally, we have developed a pipeline to render simulation data using animation software Autodesk Maya. The second video highlights the ancient art of marbling. We showcase the marbling steps and patterns, while studying the hydrodynamics behind how to create stable patterns on a viscosified fluid surface.

Settling down: simulations of soft immersed rods with the reference map technique,

Yue Sun, Yuexia Luna Lin, Nicholas J. Derr, and Chris H. Rycroft,

video entry (2021). (doi:10.1103/APS.DFD.2021.GFM.V0045)

The hydrodynamics of marbling art,

Yue Sun, John W. M. Bush, Saverio E. Spagnolie, and Chris H. Rycroft,

Milton van Dyke Award video winner (2023). (doi:10.1103/APS.DFD.2023.GFM.V0002)