

A PARALLEL COMPUTING APPROACH FOR FLUID-SOLID INTERACTION PROBLEMS

by

Milad Rakhsha

A preliminary report submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Mechanical Engineering)

Under the Supervision of Professor Dan Negrut

at the

UNIVERSITY OF WISCONSIN - MADISON

March 2017

Contents

I	Introduction	2
II	Background	4
1	Fluid Dynamics via SPH	6
1.1	Kernel Function	8
1.2	Balance Equations and Their Discretization	9
2	Rigid Body Frictional Contact via Differential Variational Inequalities	13
2.1	Preamble	13
2.2	The DVI Formulation of the Frictional Contact Problem	14
3	Nonlinear Flexible Body Dynamics via ANCF	16
3.1	An 1D, Cable, ANCF Element	18
3.2	A 2D, Shell, ANCF Element	20
4	The Fluid-Solid Coupling	22
5	Time Integration Considerations	24
6	High Performance Computing Aspects	26
III	Proposed Directions of Research	28
7	Second-Order Accurate SPH	28
8	Incompressible SPH via Poisson Pressure Equation	31
9	Incompressible SPH via Kinematic Constraints	33
10	Variable Resolution SPH	36
11	Simulation Framework for Large Scale SPH	41
12	The Performance-Gain Project	43
IV	Preliminary Work	44
13	Numerical Methods	44
13.1	Incompressible SPH with Semi-Implicit Time Integration	44
14	Advanced Computing Aspects	47
14.1	Multi-Physics Framework	47
14.2	Parallelization	48

15 Validation Efforts	49
15.1 IISPH and CFSPH	49
15.1.1 Dam Break	49
15.1.2 Elastic Gate Experiment	50
16 Application Focus	52
16.1 Multi-Physics Model of Articular Cartilage	52
16.2 Underwater Robotics	54
17 References	55

List of Figures

1 Illustration of the computational domains for non-conforming and conforming mesh methods	5
2 Illustration of the kernel W and neighboring markers of marker i . SPH markers are shown as small circles. The radius of the support domain is defined as κh	8
3 Schematic of contact between two bodies	14
4 ANCF cable element's schematic. Each node features a global position vector and a position vector gradient along the axis of the element (6DOF). Using shape functions and knowing ξ one can interpolate the degrees of freedom to any point P within the element	19
5 ANCF shell element's schematic. Global position vector \mathbf{r}^j and fiber's direction $\mathbf{r}_z^j = \frac{\partial \mathbf{r}^j}{\partial z^i}(\xi^i, \eta^j)$ are the nodal coordinates of the j^{th} node (6DOF). Using shape functions and knowing ξ and η one can interpolate the degrees of freedom to any point within the element	20
6 BCE and fluid markers, key for the coupling between fluid and solid, are represented by black and white circles, respectively. A section of the rigid body is shown herein as the gray area. The BCE markers positioned in the interior of the body (markers g and f in the figure) are placed at a depth less than or equal to the size of the compact support associated with the kernel function W	23
7 Numerical results of the IISPH method for a lid-driven cavity problem at $Re = 1$, $U_{wall} = 1$, $D = 1$ and $\nu = 1$, where D is the length of each side, and ν is the kinematic viscosity. The Lagrangian data was mapped to an Eulerian grid for visualization purposes	47
8 Fluid-solid interaction with deformable ANCF shells (Full Video)	48
9 A cylinder drops into a tank of water and goes down under the gravity effect. For visualization purposes, the Lagrangian data were mapped to an Eulerian grid on the mid-plane parallel to $x - z$ plane. (Full Video)	49
10 Comparison of water-front propagation between IISPH, CFSPH, and experimental results of Martin and Moyce [55]	50
11 Comparison of the horizontal position of the tip of the elastic gate from the numerical results of the present study, the numerical results of the study by Yang et al. [95], and the experimental results of Antoci et al. [5].	51
12 Multi-scale representation of knee articular cartilage	53
13 Comparison of the fiber alignment predicted by this study for different scenarios with experimental results ([9]) on the femur condyle.	54

Executive Summary

Solving real world problems demands engineers to take advantage of all the available resources including the power of computers for calculation. Computational methods have made it possible to conduct research without laborious experimental work. In terms of fluid mechanics, current numerical techniques allow for solving large problems with millions of degrees of freedom in both Eulerian and Lagrangian frameworks. In terms of solid mechanics, solving frictional contact problems between millions of rigid-bodies has been made possible. In addition, many finite element-based methods allow for resolving the stress-strain relationship for both large and small deformations in deformable bodies. However, combining all the physics together in one framework is still a challenging and ongoing research topic. The two general approaches to tackle the Fluid-Solid Interaction (FSI) problems are monolithic and non-monolithic (partitioned) approaches. In the former approach, the governing equations of the fluid and the solid phases are solved simultaneously, whereas in the later approach the two phases are solved separately and the solutions are explicitly coupled together. Each of these methods come with advantages and disadvantages. What makes the partitioned approach appealing is the fact the it allows for solving each phase with more efficient techniques and to make the most out of the state-of-the-art approaches in modeling, numerical solvers and hardware architectures.

In the current research proposal, the idea is to couple all the physics in one framework by making use of the state-of-the-art numerical methods and potentially newer ones to solve problems of interest. Resolving the internal micro-structure of articular cartilage and similar biological tissues is an example of such problems. In order to do so, the friction and contact problem between rigid bodies is to be resolved via complementarity-based rigid body dynamics, the deformable-body dynamics is to be captured in Absolute Nodal Coordinate Framework, and the fluid dynamics is to be simulated in Lagrangian framework using Smoothed Particle Hydrodynamics. Lastly, the frictional-contact problem between deformable and solid bodies is to be resolved with penalty-based formulation.

Part I

Introduction

Since the advent of computers, computational methods have appeared in many physical sciences. By making use of computers, performing billions of operations at a fraction of second have been made possible. This feature of computers has influenced the way that many researchers pursue their ideas to a large extent. So far, significant progress has been made in modeling single-phase problems. However, solving multidisciplinary problems that involve coupling between fluids and flexible/rigid bodies are still a challenging research topic.

In the present PhD work, it is attempted to make the most out of the state-of-the-art numerical techniques in fluid dynamics as well as flexible/rigid body dynamics to create an *accurate* and *efficient* FSI framework. More specifically, *accurate* modeling of FSI problems demands that the deformable solid phase be modeled over any arbitrary and potentially large deformation; and the fluid phase be modeled under standard conditions which are required by the Navier-Stokes equations. Nonetheless, the concept of accuracy often is tightly coupled with the numerical resolution. Especially for fluid simulation in Lagrangian framework, a more accurate solution would call for higher computational costs. Therefore, besides using advanced numerical techniques to solve the problem more *accurately*, there is a need to exploit all the available computational resources to be able to solve these types of problems *efficiently*. Consequently, parallelizing the computations is necessary.

Motivation

Modeling FSI problems in which the interaction of fluids and solids is captured in a Lagrangian-Lagrangian framework is advantageous for many reasons. Lagrangian frameworks for fluids map more closely to solid mechanics approaches, and consequently allow for robust coupling between the two phases in partitioned methods. However, a Lagrangian approach for fluids

demands more computational costs when compared to the Eulerian approach, if the same level of accuracy is expected for both. This shortcoming can be overshadowed by many factors, including but not limited to, power of computers when parallelized, and more advanced numerical methods such as higher-order accurate schemes, and variable/multi-resolution methods which allow for more accurate solutions in regions of interest while keeping the computational costs at a reasonable level.

On the other hand, accurate modeling of FSI problems encountered in real life such as modeling the internal micro-structure of Articular Cartilage (AC) with fluids and deformable/rigid bodies, requires accurate, realistic and fast simulation on the solid phase as well as the fluid phase. For instance, the solid solver must allow for large deformation in deformable bodies, resolving friction and contact between deformable/rigid bodies and so on. As far as the speed is concerned, often the speed of Finite Element (FE) solvers is not the bottleneck when compared with the mesh-less fluid solvers. However, improving upon the efficiency of rigid/flexible body dynamics solvers will allow for larger and/or higher resolution simulations.

Hence, taking advantage of the latest *numerical methods* and *computational resources* to increase the fidelity of an FSI framework is necessary. Subsequently, this framework will be used to solve the problem that is of interest in the current work; resolving the internal micro-structure of articular cartilage using flexible, and rigid bodies as well as fluids.

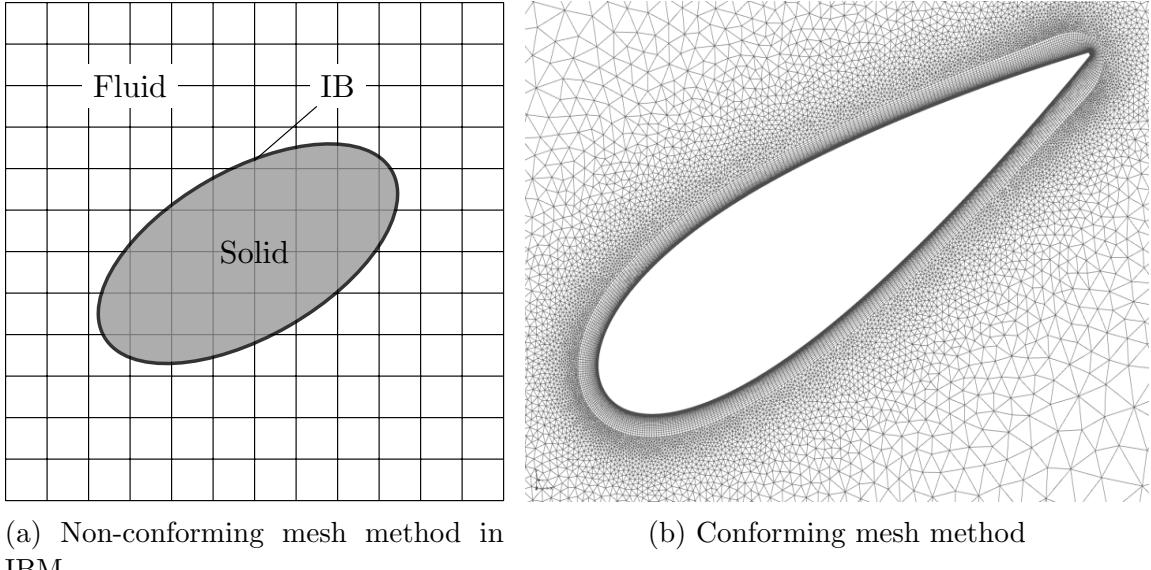
Part II

Background

Many engineering applications, e.g., fluid sloshing, fording operations, particles in suspension, etc., require the solution of a Fluid-Solid Interaction (FSI) problem. Some of these problems involve large deformations, in which case the fluid-solid coupling remains challenging to solve [75]. For most FSI problems, it is nearly impossible to obtain analytical solutions, when laboratory experiments are also limited in this scope [37]. Thus, numerical simulations must be employed. Different classifications exist in the context of numerical methods for FSI problems.

One classification is to consider the computational domain and divide the methods into *conforming mesh methods* and *non-conforming mesh methods*. In the conforming mesh methodology, the computational domains of sub-systems match at the interface, while in non-conforming mesh methods the computational domain of the fluid sub-system serves more as a background mesh for the solid phase. As an example of non-conforming mesh methods, in Immersed Boundary Method (IBM) [70, 60], the immersed boundary is tracked in a Lagrangian fashion; the fluid is tracked in an Eulerian framework on a non-conforming and regular grids; lastly, the presence of the immersed boundary on the non-conforming grid is included in Navier-Stokes equations as extra source terms. Fig. 1 illustrates the difference between the computational domains of aforementioned methods. The approach to be taken in the current work does not clearly fit into this classification since a mesh-less method is to be used for the fluid phase. However, the methodology is more similar to conforming mesh methods.

Another classification is to consider whether or not the different sub-systems are solved simultaneously in the same framework, and hence to group the methods into *monolithic* and *partitioned* approaches. In the former [41, 75, 59], the governing equations of the fluid



(a) Non-conforming mesh method in IBM

(b) Conforming mesh method

Figure 1: Illustration of the computational domains for non-conforming and conforming mesh methods.

and the solid phases are solved simultaneously; the interfacial conditions are implicit in the solution procedure; and potentially a better accuracy can be obtained for a multidisciplinary problem [37]. In contrast, in the latter approach the two phases are solved separately and the solutions are usually explicitly coupled together. Each method has its own advantages and drawbacks. What makes the partitioned approach appealing is the fact that it solves each phase with efficient techniques that make the most out of the state-of-the-art methods in modeling, numerical solvers, and hardware architectures. Among partitioned methods, the Arbitrary Lagrangian-Eulerian (ALE) approach for the fluid and Lagrangian one for the solid are commonly used to treat strongly-coupled FSI problems [42, 83]. In the ALE approach, the fluid mesh is deformed to adapt to the solid domain deformation [30]. Although this approach has proven capable of handling FSI problems with large-deformations, it is computationally expensive due to continuous mesh adaptation, a task that is not needed in meshless Lagrangian methods such as SPH [95]. This is the main reason why SPH, a Lagrangian method, is embraced here. In addition, we rely on the Absolute Nodal Coordinate Formulation (ANCF) [79, 78] to characterize the time evolution of the solid phase undergoing

large deformations.

The SPH-FEM coupling for FSI problems was discussed in [7] and further improved in [30, 23, 32]. In [7], the contact forces were calculated based on an iterative master-slave scheme which finds the best penalty forces required for the no-penetration condition. A similar but non-iterative scheme was used in [32]. In [30, 69] a dummy-particles scheme was used to compute the pressure from the fluid side. The present work follows in the footsteps of this last approach.

In the present research, the Fluid-Solid Interaction method will be resolved in a Lagrangian-Lagrangian framework. SPH is a Lagrangian numerical method for the solution of the fluid flow governing equations. SPH comes in many different flavors as far as time integration, compressibility level and boundary condition treatment are concerned. On the other hand, rigid multi-body dynamics and nonlinear flexible-body dynamics methods have been conventionally developed in Lagrangian frameworks. The methods that deal with frictional contact between rigid bodies fall into two general categories; (*i*) Complementarity (DEM-C) and (*ii*) Penalty (DEM-P) methods. In the present work both are used, depending on whether a hard or a soft contact is required. As far as the flexible multi-body dynamics is concerned, ANCF is mainly used in the present work since the method has shown its capabilities in capturing large deformations in many different problems. In what follows detailed background information of each phase is described.

1 Fluid Dynamics via SPH

SPH was introduced in 1977 by Gingold and Monaghan [31] and independently by Lucy [54]. In the SPH approach, the governing equations of motion of the fluid are discretized by (Lagrangian) moving material points (markers) which carry mass, momentum, and possibly other physical properties such as temperature. As explained before, this Lagrangian perspective matches very well with the framework which discretizes the solid phase. Moreover, due

to the fact that the identity is given to the material not the space, the non-linear advection term in the momentum equations vanishes automatically.

In SPH formulation a kernel function W is defined over a support domain S to approximate physical properties of each marker. The approximation is second-order accurate provided that the following conditions are met.

1. In the limit, as the size of the support domain goes to zero, the kernel function must approach the Dirac delta function δ , i.e., $\lim_{h \rightarrow 0} W(\mathbf{r}, h) = \delta(\mathbf{r})$.
2. The kernel function must be symmetric i.e., $W(\mathbf{r}, h) = W(-\mathbf{r}, h)$.
3. The normalization condition holds over the support domain i.e., $\int_S W(\mathbf{r}, h) dV = 1$,

where h is the kernel characteristic length. In practice, however, the second-order accuracy is deteriorated due to issues that will be discussed later in Section 7. Having the kernel function W in hand, by making use of the identity $f(\mathbf{r}) = \int_{-\infty}^{+\infty} f(\mathbf{r}') \delta(\mathbf{r} - \mathbf{r}') dV$, and approximating the delta function with a kernel function which meets the above criteria and has finite support domain of κh ($W(\mathbf{r} - \mathbf{r}', h) = 0$ when $|\mathbf{r} - \mathbf{r}'| > \kappa h$), the SPH approximation of a function f is expressed as follows,

$$f(\mathbf{r}) = \int_V f(\mathbf{r}') W(\mathbf{r} - \mathbf{r}') dV + O(h^2) , \quad (1)$$

where κ is a parameter that controls the cut-off length. The above integral can be approximated for marker i with

$$f(\mathbf{r}_i) = \int_V \frac{f(\mathbf{r}_j)}{\rho(\mathbf{r}_j)} W(\mathbf{r} - \mathbf{r}_j) \rho(\mathbf{r}_j) dV + O(h^2) \approx \sum_{j \in \mathcal{S}(i)} \frac{m_j}{\rho_j} f(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h) , \quad (2)$$

where $\rho(r_j)$ is the density, $m_j = \rho(r_j)dV$ is the marker mass, and i and j denote markers i and j , respectively. Eq. 2 is the basic SPH equation which is used to obtain higher-order derivatives. Higher-order derivatives that are derived from the above equation, however, are

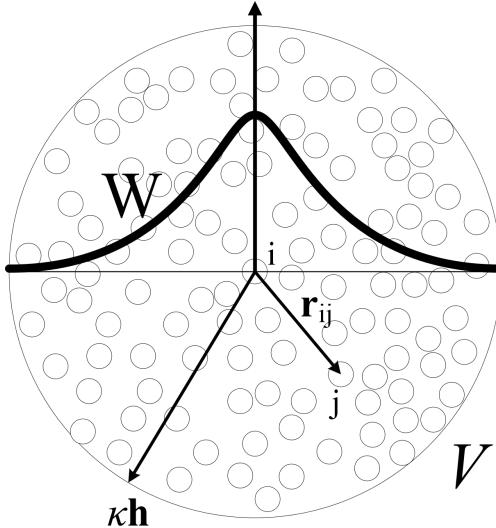


Figure 2: Illustration of the kernel W and neighboring markers of marker i . SPH markers are shown as small circles. The radius of the support domain is defined as κh .

prone to loss of accuracy, and in practice if special modification is not made, the second-order accuracy is not guaranteed anymore.

1.1 Kernel Function

There are many different kernel functions that meet the criteria mentioned in Section 1. In the original work of Gingold and Monaghan [31], a Gaussian kernel function is used. Lucy [54] used a 4^{th} -order polynomial function. M_n splines are also popular as they allow for compact support in space [76]. Examples of M_2 and M_4 splines are given by:

$$M_2(q) = \begin{cases} 1 - q, & 0 \leq q < 1 \\ 0, & q \geq 1 \end{cases}, \quad (3)$$

$$M_4(q) = \frac{1}{6} \times \begin{cases} (2 - q)^3 - 4(1 - q)^3, & 0 \leq q < 1 \\ (2 - q)^3, & 1 \leq q < 2 \\ 0, & q \geq 2 \end{cases}, \quad (4)$$

where $q = |\mathbf{r}|/h$. $W(\mathbf{r}, h) = M_n(q) \times c/h^d$ is defined based on the M_n spline where d is the dimension of the problem, and c is a constant that ensures that the normalization condition of the kernel is satisfied. For instance, in Eq. 4, $c = \frac{30}{14\pi}$ in 3D problems and $c = \frac{6}{4\pi}$ in 2D.

1.2 Balance Equations and Their Discretization

The governing equations of Newtonian fluids in incompressible flows, which is of interest in the current work by and large, are continuity and momentum, Navier-Stokes (NS), equations which are described as:

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v} , \quad (5)$$

and

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f} , \quad (6)$$

where \mathbf{v} is the fluid velocity, p is the pressure, ν is the kinematic viscosity, and \mathbf{f} is the body force per unit mass. These equations should be discretized in space based on the available information about Lagrangian markers and be integrated in time in order to give the transient solution of the flow. Higher-order derivatives are approximated based on the modified version of Eq. 2 which is read as follows

$$\langle f(\mathbf{r}_i) \rangle = \sum_{j \in \mathcal{S}(i)} V_j f(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h) , \quad (7)$$

where $V_j = m_j/\rho_j$ and $\langle f(\mathbf{r}_i) \rangle$ operator indicates the SPH approximate of $f(\mathbf{r}_i)$, which does not necessarily hold the second-order accuracy due to inconsistency associated with conversion of integral over volume to summation over discrete markers with random distribution.

There are many formulas to approximate the first-order derivative of a field f . The

scheme introduced by Monaghan [31] is:

$$\langle \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j f_j \nabla_i W_{ij} , \quad (8)$$

where $\langle \nabla \rangle_i$ indicates the approximated gradient of the field associated with the marker i , $W_{ij} = W(\mathbf{r}_i - \mathbf{r}_j, h)$, and ∇_i denotes the differentiation in space with respect to the coordinates on particle i . For the sake of simplicity $\nabla_i W_{ij}$ is replaced by ∇W_{ij} , herein. The modified version of the Eq. 8 that satisfies the zeroth-order consistency¹ is expressed as:

$$\langle \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (f_j - f_i) \nabla_i W_{ij} . \quad (9)$$

The symmetric form of Eq. 9 is read as follows

$$\langle \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (f_j + f_i) \nabla_i W_{ij} , \quad (10)$$

which does not hold the zeroth-order consistency, but preserves the local momentum conservation if used to discretize force terms. Eq. 10 is usually used to discretize the pressure gradient term in Eq. 6 [25].

Regarding the approximation of second-order derivatives, one can apply the first-order operators in Eq. 8-10 to the gradient of a field to derive $\langle \nabla \cdot \nabla (\cdot) \rangle_i$ operator and obtain,

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j \langle \nabla f \rangle_j \nabla_i W_{ij} , \quad (11a)$$

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (\langle \nabla f \rangle_j - \langle \nabla f \rangle_i) \nabla_i W_{ij} , \quad (11b)$$

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (\langle \nabla f \rangle_j + \langle \nabla f \rangle_i) \nabla_i W_{ij} . \quad (11c)$$

¹First derivative of a constant function should exactly be zero for the zeroth-order consistency to be satisfied.

However, these forms require double summations, are expensive to calculate and show oscillatory solutions similar to grid-based methods [28]. One way to remove the double summations is to permute the ∇ operator to the kernel function similar to Eq. 8 and obtain

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j f_j \nabla \cdot \nabla W_{ij} , \quad (12a)$$

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (f_j - f_i) \nabla \cdot \nabla W_{ij} , \quad (12b)$$

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j (f_j + f_i) \nabla \cdot \nabla W_{ij} . \quad (12c)$$

Different forms of Eq.12 have been used by many researchers [17, 87]. Another well-known discretization scheme for the Laplacian operator was proposed in [15] and is expressed as

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} 2V_j \frac{(f_i - f_j)}{r_{ij}} \mathbf{e}_{ij} \cdot \nabla_i W_{ij} , \quad (13)$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$, and $\mathbf{e}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)/r_{ij}$. It has been shown that the above equation performs better than Eqs. 11-12.

Knowing how to discretize different terms in Eq. 5-6, one can solve the NS equations. Depending on how the pressure gradient term is coupled to the continuity equation, different methods such as Weakly Compressible SPH (WCSPH), and Incompressible SPH (ISPH) could emerge. In WCSPH methods, the fluid is allowed to have some amount of compressibility. Pressure is obtained from an equation of state with the knowledge of the density. In fact, if there is no density variation inside the fluid domain, the use of the equation of state leads to a constant pressure inside the domain which eliminates the effect of the pressure gradient term in Eq. 6. An example of an equation of state used in WCSPH methods is expressed as follows:

$$p = c^2 \rho , \quad (14)$$

where c can be adjusted in order to reach the desirable level of compressibility. The lower the

desirable compressibility level, the higher the stiffness, c , of the equation of state. However, one has to decrease the integration time step when increasing the value of c , and this is the price that is paid when pressure-velocity coupling is overshadowed by making use of the equation of state.

In contrast to WCSPH methods, Incompressible SPH methods couple the continuity and NS equations, Eq. 5-6, with similar approaches to the one introduced by Chorin [18]. Coupling the pressure and the velocity using this methodology results in semi-implicit time integration where velocity is predicted using a part of momentum balance terms with the information of the current time step, and then corrected using a pressure gradient term. The pressure is obtained by solving a linear system of equation derived from continuity equation at the new time step. Semi-implicit methods guarantee larger time-integration steps, with the price of solving a linear system at each time step. If the resulting system of equations is well-conditioned and fast linear solvers are used, then incompressible semi-implicit methods could be viable alternatives to weakly compressible and explicit methods.

In the current work, four major thrusts are perused as far as the fluid solver is concerned. (i) Increasing the stability and speed of the fluids' solvers by making use of semi-implicit methods as explained above, (ii) decreasing the computational costs by regaining the second-order accuracy of the Eqs. 7-13 and/or by using variable resolution methods where h in Eq. 7 varies from marker to marker, and (iii) increasing the speed of the solver by parallelizing the calculations with the latest available hardware architecture. More details about (i) will be discussed in Section 5. Regarding (iii), details of the parallelization paradigm of the fluid solver will be explained in Section 6

2 Rigid Body Frictional Contact via Differential Variational Inequalities

The Differential Variational Inequality (DVI) approach [68] is one of the two major approaches to deal with frictional contact in many-body dynamics simulations. The main idea in this approach is to modify the equations of motion to include the differential inclusion [29] and to impose non-penetration between rigid bodies through constraints enforced by applying impulses. After discretization this problem is posed as an optimization problem with complementarity and equilibrium constraints. More specifically, the three-dimensional Coulomb friction assumption leads to a Non-linear Complementarity Problem (NCP). In what follows, more details about the theory of the complementarity approach is explained.

2.1 Preamble

“Rigid body” refers to a 3D object that can translate and rotate in space. The set of generalized coordinates that describe the position and orientation of a body in the 3D Euclidean space are $\mathbf{r}_j \in \mathbb{R}^3$ and $\boldsymbol{\epsilon}_j \in \mathbb{R}^4$, which are respectively the absolute position of the center of mass, and Euler parameters associated with orientation of body j . Combining the set of generalized coordinates of different bodies for a system of n_b bodies, one can write the set of generalized coordinates describing the system at position level as $\mathbf{x} = [\mathbf{r}_1^T, \boldsymbol{\epsilon}_1^T, \dots, \mathbf{r}_{n_b}^T, \boldsymbol{\epsilon}_{n_b}^T]^T \in \mathbb{R}^{7n_b}$, and at velocity level as $\dot{\mathbf{x}} = [\dot{\mathbf{r}}_1^T, \dot{\boldsymbol{\epsilon}}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \dot{\boldsymbol{\epsilon}}_{n_b}^T]^T \in \mathbb{R}^{7n_b}$. One can choose to use angular velocities instead of the time derivative of the Euler parameters to describe the system at the velocity level by $\mathbf{v} = [\dot{\mathbf{r}}_1^T, \bar{\boldsymbol{\omega}}_1^T, \dots, \dot{\mathbf{r}}_{n_b}^T, \bar{\boldsymbol{\omega}}_{n_b}^T]^T \in \mathbb{R}^{6n_b}$, which reduces the problem size. The transformation from the derivatives of Euler parameters, $\dot{\boldsymbol{\epsilon}}_B$, to angular velocities at the body-fixed frame, $\bar{\boldsymbol{\omega}}_B$, for each body is governed by $\dot{\boldsymbol{\epsilon}}_B = \frac{1}{2}\mathbf{G}^T(\boldsymbol{\epsilon}_B)\bar{\boldsymbol{\omega}}_B$, where matrix $\mathbf{G} \in \mathbb{R}^{3 \times 4}$ depends linearly on the Euler parameters $\boldsymbol{\epsilon}_B$. Therefore, if one chooses to work with Euler angles at the velocity level, the block diagonal matrix $\mathbf{L}(\mathbf{q}) \equiv \text{diag}[\mathbf{I}_{3 \times 3}, \frac{1}{2}\mathbf{G}^T(\boldsymbol{\epsilon}_1), \dots, \mathbf{I}_{3 \times 3}, \frac{1}{2}\mathbf{G}^T(\boldsymbol{\epsilon}_{n_b})] \in \mathbb{R}^{7n_b \times 6n_b}$ can be used to obtain

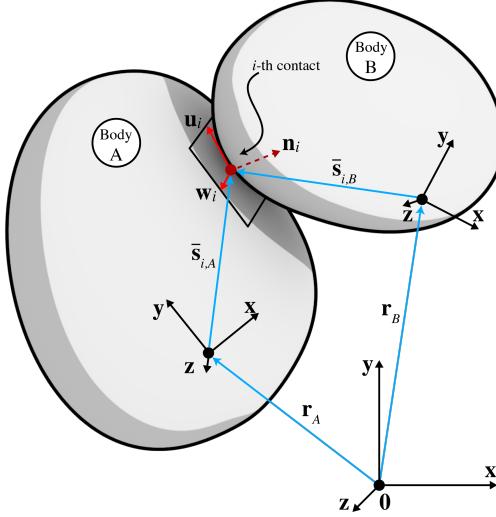


Figure 3: Schematic of contact between two bodies.

$\dot{\mathbf{x}} = \mathbf{L}(\mathbf{q})\mathbf{v}$, the time derivative of the set of generalized coordinate describing the system, where $\mathbf{I}_{3 \times 3}$ is the identity matrix [35].

2.2 The DVI Formulation of the Frictional Contact Problem

Consider a contact between bodies A and B as shown in Fig. 3. The tangent plane at the contact point for the contact i is defined by vectors \mathbf{u}_i , \mathbf{w}_i and the normal \mathbf{n}_i in Fig. 3. A local reference frame can be defined for each body at the contact point based on these vectors. For body A , the normal to the tangent plane, $\mathbf{n}_{i,A}$, points toward body B . The two mutually orthogonal vectors $\mathbf{u}_{i,A}$, and $\mathbf{w}_{i,A}$ are defined using the Gram-Schmidt method. The same methodology is used to build the local reference frame at the contact point i , for body B with $\mathbf{w}_{i,B}$, $\mathbf{u}_{i,B}$, and $\mathbf{n}_{i,B} \in \mathbb{R}^3$.

If the gap (distance) between bodies A and B at the contact point is defined by Φ , a complementarity condition can be defined as $0 \leq \hat{\gamma}_{i,n}^c \perp \Phi \geq 0$, where $\hat{\gamma}_{i,n}^c$ is the Lagrange multiplier associated with the contact i . The complementarity condition states that at least one of the $\hat{\gamma}_{i,n}^c$, or Φ is zero; when the gap function is zero, the normal contact force is greater than zero and when the normal contact force is zero the gap function is greater than zero

(there is no contact between body A and B). The contact force associated with contact i can be expressed as $\mathbf{f}_{i,N} = \hat{\gamma}_{i,n}^c \mathbf{n}_i$, and $\mathbf{f}_{i,T} = \hat{\gamma}_{i,u}^c \mathbf{u}_i + \hat{\gamma}_{i,w}^c \mathbf{w}_i$ which are the normal and tangential forces, respectively. $\hat{\gamma}_{i,w}^c$, $\hat{\gamma}_{i,u}^c$, and $\hat{\gamma}_{i,n}^c$ are the magnitude of the contact forces in each direction. The Coulomb dry-friction model based on the friction forces is expressed as [85, 84]

$$\sqrt{(\hat{\gamma}_{i,u}^c)^2 + (\hat{\gamma}_{i,w}^c)^2} \leq \mu_i^f \hat{\gamma}_{i,n}^c, \quad (15a)$$

$$\|\mathbf{v}_{i,T}\| \left(\sqrt{(\hat{\gamma}_{i,u}^c)^2 + (\hat{\gamma}_{i,w}^c)^2} - \mu_i^f \hat{\gamma}_{i,n}^c \right) = 0, \quad (15b)$$

$$\langle \mathbf{f}_{i,T}, \mathbf{v}_{i,T} \rangle = -\|\mathbf{f}_{i,T}\| \|\mathbf{v}_{i,T}\|, \quad (15c)$$

where $\mathbf{v}_{i,T}$ denotes the relative tangential velocity between bodies A and B at the contact point. More specifically, Eq. 15a states that the the friction force is less than the normal force times the friction coefficient. Eq. 15b states a complementarity condition where equality condition of Eq. 15a holds if the $\mathbf{v}_{i,T} \neq 0$, and inequality of Eq. 15a holds if $\mathbf{v}_{i,T} = 0$. Lastly, Eq. 15c states that the friction force is in the opposite direction of $\mathbf{v}_{i,T}$. If now one considers the following constraint minimization problem,

$$(\hat{\gamma}_{i,u}^c, \hat{\gamma}_{i,w}^c) = \underset{\sqrt{(\hat{\gamma}_{i,u}^c)^2 + (\hat{\gamma}_{i,w}^c)^2} \leq \mu_i^f \hat{\gamma}_{i,n}^c}{\operatorname{argmin}} \mathbf{v}_{i,T}^T (\hat{\gamma}_{i,u}^c \mathbf{u}_i + \hat{\gamma}_{i,w}^c \mathbf{w}_i), \quad (16)$$

then Eq. 15a-15c represent the first order Karush-Kuhn-Tucker optimality condition for the above optimization problem. Hence, the Coulomb friction model is implemented as a constraint optimization problem. Finally, the contact force at the i^{th} contact point is expressed as $\mathbf{f}_i = \mathbf{f}_{i,N} + \mathbf{f}_{i,T} = \hat{\gamma}_{i,n}^c \mathbf{n}_i + \hat{\gamma}_{i,w}^c \mathbf{u}_i + \hat{\gamma}_{i,w}^c \mathbf{w}_i \in \Upsilon_i$, where Υ_i is a 3D cone of slope $\tan^{-1}(\mu_i^f)$ and oriented along \mathbf{n}_i , i.e., $\Upsilon_i = \{[x, y, z]^T \in \mathbb{R}^3 | \sqrt{y^2 + z^2} \leq \mu_i^f x\}$.

The Newton-Euler equations of motion for the system [85] are expressed as:

$$\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{L}(\mathbf{q})\mathbf{v} \\
\mathbf{M}\dot{\mathbf{v}} &= \mathbf{f}(t, \mathbf{q}, \mathbf{v}) + \sum_{i \in \mathcal{A}(\mathbf{q}, \delta)} (\hat{\gamma}_{i,n} \mathbf{D}_{i,n} + \hat{\gamma}_{i,u} \mathbf{D}_{i,u} + \hat{\gamma}_{i,w} \mathbf{D}_{i,w}) \\
0 &= \mathbf{g}(\mathbf{q}, t)
\end{aligned} \tag{17}$$

$i \in \mathcal{A}(\mathbf{q}(t), \delta) \quad : \quad 0 \leq \hat{\gamma}_{i,n} \perp \Phi_i(\mathbf{q}) \geq 0$

$$(\hat{\gamma}_{i,u}, \hat{\gamma}_{i,w}) = \underset{\sqrt{(\hat{\gamma}_{i,u}^c)^2 + (\hat{\gamma}_{i,w}^c)^2} \leq \mu_i^f \hat{\gamma}_{i,n}^c}{\operatorname{argmin}} \mathbf{v}^T (\hat{\gamma}_{i,u}^c \mathbf{D}_{i,u} + \hat{\gamma}_{i,w}^c \mathbf{D}_{i,w}),$$

where $\mathbf{f}(t, \mathbf{q}, \mathbf{v})$ are the external forces, \mathbf{M} is the system mass Matrix, $\mathcal{A}(\mathbf{q}, \delta)$ is the set of active and potential unilateral constraints based on the bodies that are mutually less than δ apart, and $\mathbf{g}(\mathbf{q}, t)$ is the set of bilateral constraints acting on the system. Moreover, the tangent space generators $\mathbf{D}_i = [\mathbf{D}_{i,n}, \mathbf{D}_{i,u}, \mathbf{D}_{i,w}] \in \mathbb{R}^{6n_b \times 3}$ are defined as

$$\mathbf{D}_i^T = [\mathbf{0} \ \dots \ -\mathbf{A}_{i,p}^T \ \mathbf{A}_{i,p}^T \mathbf{A}_A \tilde{\mathbf{s}}_{i,A} \ \mathbf{0} \ \dots \ \mathbf{0} \ \mathbf{A}_{i,p}^T \ -\mathbf{A}_{i,p}^T \mathbf{A}_B \tilde{\mathbf{s}}_{i,B} \ \dots \ \mathbf{0}], \tag{18}$$

where $\mathbf{A}_{i,p} = [\mathbf{n}_i, \mathbf{u}_i, \mathbf{w}_i] \in \mathbb{R}^{3 \times 3}$ is the orientation matrix associated with contact i , $\mathbf{A}_A = \mathbf{A}(\epsilon_A)$ and $\mathbf{A}_B = \mathbf{A}(\epsilon_B)$ are the rotation matrices of bodies A and B respectively; the vectors $\tilde{\mathbf{s}}_{i,A}$ and $\tilde{\mathbf{s}}_{i,B} \in \mathbb{R}^3$ represent the contact point positions in body-relative coordinates as shown in Fig. 3. More details about the solution algorithm, and time-stepping scheme of the this DVI problem may be found in [85, 3, 4].

3 Nonlinear Flexible Body Dynamics via ANCF

The nonlinear flexible body dynamics formulation used in the current work draws on ANCF, a nonlinear finite element formulation introduced by Shabana [80] to describe large deformation of moving bodies. The salient feature of ANCF is the use of position vector gradients to describe the rotation of the body. ANCF uses nodal global position and nodal position vector

gradient vectors to describe the nonlinear dynamics of flexible bodies that can undergo large deformation. In general, the position field of i^{th} ANCF element may be defined as:

$$\underbrace{\mathbf{r}^i(\xi, \eta, \zeta, t)}_{\substack{\text{Position of an arbitrary} \\ \text{point within the element}}} = \underbrace{\mathbf{S}(\xi, \eta, \zeta)}_{\substack{\text{Space-dependent} \\ \text{shape function}}} \times \underbrace{\mathbf{q}^i(t)}_{\substack{\text{Time-dependent vector of} \\ \text{nodal degrees of freedom}}}, \quad (19)$$

which simply gives the position of any point $(\xi, \eta, \zeta) \in [-1, 1]$ inside the element at time t based on interpolation ($\mathbf{S}(\xi, \eta, \zeta)$) of the nodal coordinates ($\mathbf{q}^i(t)$). Due to the fact that description of elements is in global coordinates, the inertia forces have a simple form in ANCF elements. The velocity of any point within an element i may be written as

$$\underbrace{\dot{\mathbf{r}}^i(\xi, \eta, \zeta, t)}_{\substack{\text{Velocity of an arbitrary} \\ \text{point within the element}}} = \underbrace{\mathbf{S}(\xi, \eta, \zeta)}_{\substack{\text{Space-dependent} \\ \text{shape function}}} \times \underbrace{\dot{\mathbf{q}}^i(t)}_{\substack{\text{Time-dependent vector of} \\ \text{generalized velocities}}}. \quad (20)$$

The kinetic energy of a finite element i can be obtained as

$$T = \frac{1}{2} \int_V \rho \dot{\mathbf{r}}^{iT} \dot{\mathbf{r}}^i \, dV = \frac{1}{2} \dot{\mathbf{q}}^{iT} \mathbf{M} \dot{\mathbf{q}}^i, \quad (21)$$

where the mass matrix \mathbf{M} is defined as $\mathbf{M} = \int_A \rho A \mathbf{S}^T \mathbf{S} \, dx$, which is time-independent. The equations of motion assume the form [77]

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{Q}_e = \mathbf{Q}_a, \quad (22)$$

where \mathbf{Q}_e and \mathbf{Q}_a are the generalized element elastic and applied forces, respectively. The description of these elements and calculation of the internal forces are described in §3.1, and §3.2.

ANCF elements may be classified based on the number of position vector gradients defined at each node. (i) **Fully parameterized** ANCF elements use position vector $r \in \mathbb{R}^3$, and 3 position vector gradients, \mathbf{r}_x , \mathbf{r}_y , and $\mathbf{r}_z \in \mathbb{R}^3$ where x , y , and z are the natural coordinates of the element. Fully parameterized elements allow for easy implementation of the continuum mechanics approach to calculate the deformation gradient (\mathbf{F}). (ii) **Gradient Deficient** ANCF elements

use position vector $r \in \mathbb{R}^3$, and fewer than 3 position vector gradients, when using fewer position vector gradients is sufficient to define the volume used in continuum mechanics approach. Using fewer position gradient vectors has been shown to eliminate various locking problems.

Development of ANCF elements is still an ongoing research topic, yet in the current work elements that have been shown robust and acceptable accuracy are chosen for the simulations of the flexible bodies. More specifically, ANCF cable element, ANCF shell element, and hexahedron brick elements, which are appropriate respectively for modeling 1D, 2D, and 3D bodies, are to be used in the FE analysis of this present work.

3.1 An 1D, Cable, ANCF Element

The gradient-deficient ANCF cable element introduced by Berzeri and Shabana [10] is used in the present work to simulate 1D flexible bodies such as fibers inside the articular cartilage micro-structure, as illustrated in Fig. 12. As shown in Fig. 4, the coordinates of this element at each node are a position vector and a position vector gradient along the beam center axis. The position gradient vectors normal to the cable axis are not defined, hence the element is gradient deficient. Subsequently, torsion and shear deformation cannot be captured with this set of degrees of freedom. The coordinates (nodal degree of freedom) of the j^{th} node is expressed as the 6×1 matrix $\mathbf{q}^j(t) = [\mathbf{r}^{jT} \quad \mathbf{r}_x^{jT}]^T$. The position of any point inside the i^{th} element may be interpolated from the nodal degrees of freedom of its nodes as follows

$$\mathbf{r}^i = \begin{bmatrix} s_1 \mathbf{I} & s_2 \mathbf{I} & s_3 \mathbf{I} & s_4 \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{q}^{1T} & \mathbf{q}^{2T} \end{bmatrix}^T = \mathbf{S}(\xi) \mathbf{q}^i, \quad (23)$$

where \mathbf{I} is the 3×3 identity matrix, $\mathbf{S}(\xi)$ is a 3×12 matrix, \mathbf{q}^1 , are \mathbf{q}^2 are the nodal coordinates of the two nodes forming element i , as defined before, and finally \mathbf{q}^i is a 12×1 matrix combining the nodal coordinates of the element i . The interpolation functions are

defined as

$$\begin{aligned}
s_1 &= 1 - 2\xi^2 + 2\xi^3, \\
s_2 &= l(\xi - 2\xi^2 + \xi^3), \\
s_3 &= 3\xi^2 - 2\xi^3, \\
s_4 &= l(-\xi^2 + \xi^3),
\end{aligned} \tag{24}$$

where $0 < \xi < 1$ is the non-dimensional parameter defined over the natural coordinates of the element locates a point along the cable centerline ($\xi = 0$ at the first node, and $\xi = l$ at the second node), and l is the length of the element.

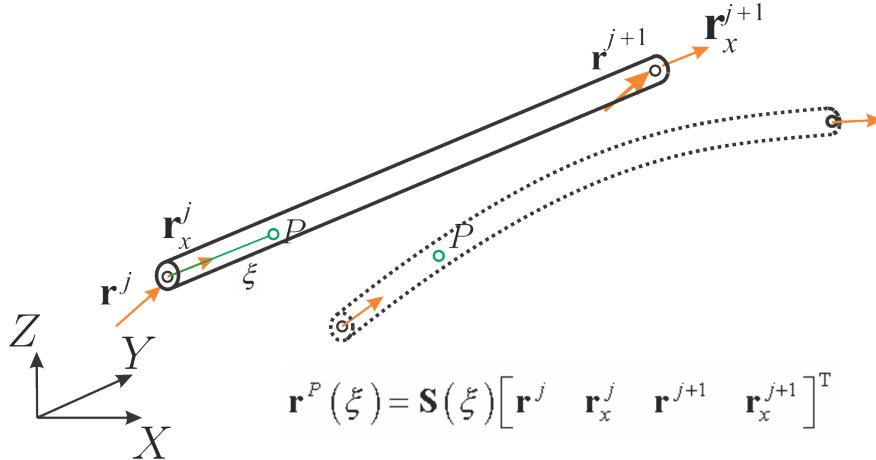


Figure 4: ANCF cable element's schematic. Each node features a global position vector and a position vector gradient along the axis of the element (6DOF). Using shape functions and knowing ξ one can interpolate the degrees of freedom to any point P within the element.

Knowing the axial and bending strains, one can define the internal loads of this element. The generalized element elastic forces are calculated as follows:

$$\mathbf{Q}_e^i = \int_L \left[EA\varepsilon_x \left(\frac{\partial \varepsilon_x}{\partial \mathbf{q}} \right)^T + EI\kappa \left(\frac{\partial \kappa}{\partial \mathbf{q}} \right)^T \right] dx, \tag{25}$$

where E , A , and I are the modulus of elasticity, the cross section area, and the area moment of inertia, respectively. The axial strain and curvature are defined as follows

$$\varepsilon_x = \frac{1}{2} (\mathbf{r}_x^T \mathbf{r}_x - 1) \text{ and } \kappa = \frac{|\mathbf{r}_x \times \mathbf{r}_{xx}|}{|\mathbf{r}_x|^3},$$

where $\mathbf{r}_{xx} = \partial^2 \mathbf{r} / \partial x^2$.

3.2 A 2D, Shell, ANCF Element

The gradient-deficient ANCF shell element studied in [94] is used in the present work to simulate 2D flexible bodies such as articular surface of the articular cartilage, as illustrated in Fig. 12. The nodal global position vector (\mathbf{r}^i) and global position vector transverse gradient ($\mathbf{r}_z^i = \frac{\partial \mathbf{r}^i}{\partial z^i}(\xi^i, \eta^i)$) are chosen as the nodal degrees of freedom, as shown in Fig. 5.

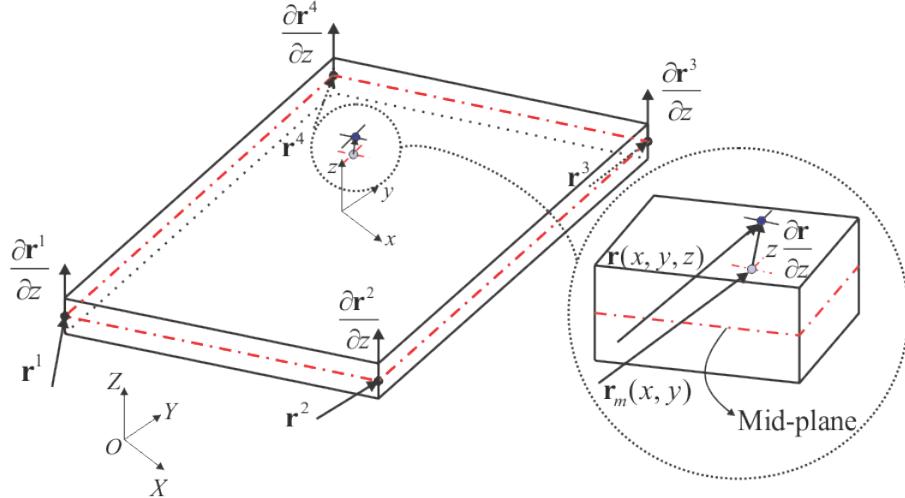


Figure 5: ANCF shell element's schematic. Global position vector \mathbf{r}^j and fiber's direction $\mathbf{r}_z^j = \frac{\partial \mathbf{r}^j}{\partial z^j}(\xi^j, \eta^j)$ are the nodal coordinates of the j^{th} node (6DOF). Using shape functions and knowing ξ and η one can interpolate the degrees of freedom to any point within the element.

The positions and gradients on the mid-plane for any point inside the i^{th} element can be interpolated from the positions and gradients of its nodes as follows

$$\mathbf{r}_m^i(\xi^i, \eta^i) = \mathbf{S}_m^i(\xi^i, \eta^i) \mathbf{e}_p^i, \quad \frac{\partial \mathbf{r}^i}{\partial z^i}(\xi^i, \eta^i) = \mathbf{S}_m^i(\xi^i, \eta^i) \mathbf{e}_g^i, \quad (26)$$

where ξ^i and η^i refer to i^{th} element's natural coordinates in the parametric space, $\mathbf{S}_m^i = [S_1^i \mathbf{I} \ S_2^i \mathbf{I} \ S_3^i \mathbf{I} \ S_4^i \mathbf{I}]$ is a bilinear shape function matrix, $\mathbf{e}_p^{ij} = \mathbf{r}^{ij}$ is the position vector of j^{th} node of the element i , and $\mathbf{e}_g^{ij} = \partial \mathbf{r}^{ij} / \partial z^i$ is the position vector gradient of node j of element i , and \mathbf{I} is the 3×3 identity matrix. The bilinear shape functions of the ANCF shell element are given by the following expressions

$$\begin{aligned} S_1^i &= \frac{1}{4}(1 - \xi^i)(1 - \eta^i), S_2^i = \frac{1}{4}(1 + \xi^i)(1 - \eta^i), \\ S_3^i &= \frac{1}{4}(1 + \xi^i)(1 + \eta^i), S_4^i = \frac{1}{4}(1 - \xi^i)(1 + \eta^i). \end{aligned}$$

The position of an arbitrary point in the i^{th} element may be described as

$$\mathbf{r}^i(\xi^i, \eta^i, z^i) = \mathbf{S}^i(\xi^i, \eta^i, z^i)\mathbf{e}^i, \quad (27)$$

where $\mathbf{S}^i = [\mathbf{S}_m^i \ z^i \mathbf{S}_m^i]_{3 \times 24}$ is the combined shape function matrix, and $\mathbf{e}^i = [(\mathbf{e}_p^i)^T \ (\mathbf{e}_g^i)^T]_{1 \times 24}^T$ is the coordinates of the i^{th} element grouped together. Eq. 27 allows for interpolating points along the element thickness by incorporating the element natural coordinate z^i .

The Green-Lagrange strain tensor, which is expressed as follows,

$$\mathbf{E}^i = \frac{1}{2} \left((\mathbf{F}^i)^T \mathbf{F}^i - \mathbf{I} \right), \quad (28)$$

is used to obtain the strains, where \mathbf{F}^i is the deformation gradient matrix defined as the Jacobian of the current configuration over the reference configuration, which is expressed as

$$\mathbf{F}^i = \frac{\partial \mathbf{r}^i}{\partial \mathbf{X}^i} = \frac{\partial \mathbf{r}^i}{\partial \mathbf{x}^i} \left(\frac{\partial \mathbf{X}^i}{\partial \mathbf{x}^i} \right)^{-1}. \quad (29)$$

The strain tensor can then be expressed in vector form as follows

$$\boldsymbol{\varepsilon}^i = \left[\varepsilon_{xx}^i \ \varepsilon_{yy}^i \ \gamma_{xy}^i \ \varepsilon_{zz}^i \ \gamma_{xz}^i \ \gamma_{yz}^i \right]^T \quad (30)$$

where $\boldsymbol{\varepsilon}^i$ is the engineering strain vector in the deformed configuration. The elastic internal forces are obtained by integration over the element volume using Gaussian quadrature as follows

$$\mathbf{Q}_e^i = \int_{V_0^i} \left(\frac{\partial \boldsymbol{\varepsilon}^i}{\partial \mathbf{e}^i} \right)^T \boldsymbol{\sigma}^i dV_0^i, \quad (31)$$

where $\boldsymbol{\sigma}^i$ is the vector of the second Piola-Kirchhoff stresses and dV_0^i is the infinitesimal volume at the reference configuration of the element i .

In order to alleviate the locking of the element, two modifications are performed on the strain's field of the element. The bilinear quadrilateral ANCF shell elements suffer from the in-plane shear/normal and transverse shear lockings, which can be eliminated using the ANS approach. Moreover, the use of position vector transverse gradient in this element causes thickness locking, which can be alleviated using the EAS approach. More details about the aforementioned approaches were provided in [94].

4 The Fluid-Solid Coupling

The fluid-solid coupling is obtained by making use of ghost Boundary Condition Enforcing (BCE) SPH particles that are placed on and close to the flexible bodies as shown in Fig. 6. This idea is similar to what was discussed in [30, 69].

More specifically, the position, velocity and acceleration of each BCE marker is dictated by the flexible body to which it belongs. In other words, these BCE particles are attached to the solid. The shape functions of Eqs. 24 and 3.2 are used to interpolate the position, velocity, and acceleration of each BCE particle based on nodal values of the corresponding ANCF element. The velocity and the acceleration of the BCE markers factor in the solution of the fluid phase equations of motion when imposing the no-slip and no-penetration boundary conditions. Herein, the velocity \mathbf{v}_a of an SPH BCE marker a is calculated based on [1]

$$\mathbf{v}_a = 2\mathbf{v}_a^p - \tilde{\mathbf{v}}_a, \quad (32)$$

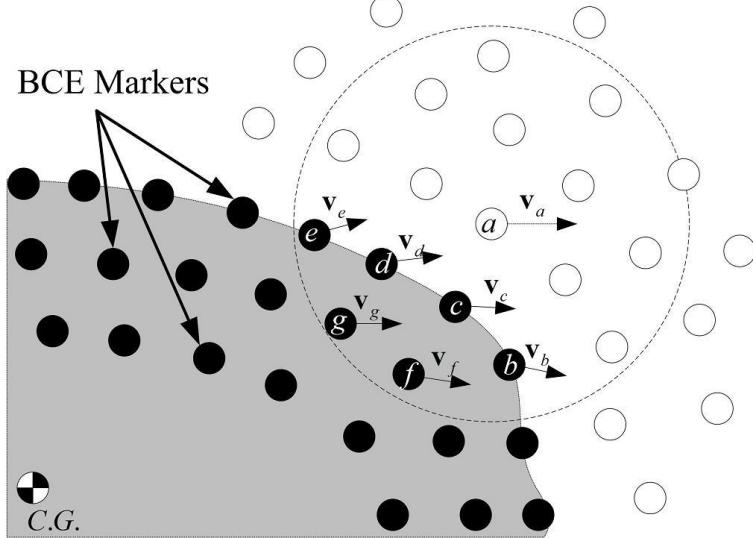


Figure 6: BCE and fluid markers, key for the coupling between fluid and solid, are represented by black and white circles, respectively. A section of the rigid body is shown herein as the gray area. The BCE markers positioned in the interior of the body (markers g and f in the figure) are placed at a depth less than or equal to the size of the compact support associated with the kernel function W .

where \mathbf{v}_a^p is the prescribed wall velocity obtained from the interpolation of nodal velocities for each FE, and $\tilde{\mathbf{v}}_a$ is an extrapolation of the smoothed velocity field of the fluid phase to the BCE markers

$$\tilde{\mathbf{v}}_a = \frac{\sum_{b \in \mathbf{F}} \mathbf{v}_b W_{ab}}{\sum_{b \in \mathbf{F}} W_{ab}}.$$

Above, \mathbf{F} denotes a set of “fluid” markers that are within the compact support of the BCE marker a . The pressure of a BCE marker is calculated via a force balance condition at the wall interface, which leads to [1] :

$$p_a = \frac{\sum_{b \in \mathbf{F}} p_b W_{ab} + (\mathbf{g} - \mathbf{a}_w) \cdot \sum_{b \in \mathbf{F}} \rho_b \mathbf{r}_{ab} W_{ab}}{\sum_{b \in \mathbf{F}} W_{ab}}, \quad (33)$$

where \mathbf{g} is the gravity and \mathbf{a}_w is the prescribed acceleration of the solid phase at the location of SPH BCE particle a , location obtained by interpolating from the nodal accelerations.

The overall procedure for coupling the fluid phase with the solid phase(s) may be sum-

marized as follows in Algorithm 1.

Algorithm 1 Fluid-Solid coupling procedure at each time step

- 1: The fluid phase is integrated in time.
 - 2: The pressure and viscous forces exerted by the fluid markers onto the BCE markers are obtained via Eqs. 58 and 59.
 - 3: The viscous and pressure generalized forces are calculated from BCE markers and applied to the nodal coordinates.
 - 4: Knowing the forces impressed by the fluid phase onto the solid phase, the solid phase is integrated in time.
 - 5: The new positions, velocities, and acceleration of the BCE markers are interpolated from the new nodal values.
 - 6: The velocities and pressure of the BCE markers required for the no-slip and no-penetration conditions are obtained respectively from Eq. 32 and 33.
-

5 Time Integration Considerations

The unsteady solution to the set of PDEs described by Eq. 6 and DAEs described by Eq. 17 must be integrated in time to obtain an approximate solution at discrete instants in time for each sub-system. The fluid and solid phases are then explicitly coupled with the scheme described in §4. The general approaches to solve the sets of equations for each phase are explicit or implicit integration. In explicit integration methods, the informations of the new time step are obtained solely based on the information of the current time step, yet this approach will usually require very small time steps. In contrast, in implicit integration schemes, the information of the current time step is somehow dependent to the information of the previous time step. Hence, another complexity is added to the solution of the equations at each time step, with the benefits that come with the implicit methods, such as larger time steps. Generally speaking, implicit integrators tend to have larger stability regions

and are more stable than explicit schemes. In the current work, the idea is to use implicit integrators for both the fluid and the solid phases. This, however, comes at the cost of solving a (non)linear system of equations at each time step.

For the fluid phase, the implicit nature comes into play in the velocity-pressure (continuity-momentum) coupling [28, 93]. Depending on the flow characteristics, one may choose to calculate the velocity at the new time step with the information of the pressure (and perhaps velocity) at the new time step. The original Projection method introduced by Chorin [18], evaluates new velocities based on the pressure at the new time step. In this formulation, new pressures obtained by making use of divergence-free velocity condition for incompressible flows. These types of formulations are semi-implicit as they make use of only *some* information from the new time step and some information from the current step. For highly viscous flows, another restriction on the time step comes into play that inversely depends on the viscosity [39, 40]. For such flows, it becomes vital to treat the viscous term of Eq. 6 implicitly too [86, 48]. Nonetheless, even the implicit treatment of the pressure gradient term (and viscous term) does not still guarantee infinitely large time steps because of the nonlinearities and coupling of equations.

On the other hand, for the solid phase, the Backward Differentiation Formula (BDF) method, the Newmark and its modification, the Hilber-Hughes-Taylor (HHT) method, are the most widely used approaches for implicit integration. The order of the BDF method is defined depending on how many previous time steps are taken into account [14, 53]. The HHT methods, which works similar to the Newmark method, was designed to increase the amount of numerical damping without degrading the order of accuracy [45, 36, 64]. In all these methods, the position and velocity in Eq. 17 at the new time step are written as a function of acceleration of the new time step and informations of previous time step(s). The unknown of the equations will be the set of accelerations and Lagrange multipliers associated with the constraints. Due to the presence of the bilateral constraints, this approach will lead to a nonlinear system of equations at each time step. The solution of this nonlinear system

is usually obtained via Newton or modified Newton, and require that the Jacobian matrix be evaluated and a linear sub-system be solved for a few times until convergence is achieved. After the convergence of the Newton method, the new positions and velocities are recovered from the information of the accelerations.

6 High Performance Computing Aspects

The speed of solving Eqs. 6 and 17 depends primarily on the size (resolution) of the system. Realistic simulation of the fluid phase requires smaller kernel length, and therefore larger number of markers need to be simulated. The same is true for the flexible bodies; using more elements to mesh a flexible body results in a more accurate representation of the underlying physics. Nonetheless, this is not achieved in a linear fashion. For instance, if one halves the kernel length in the fluid solver to obtain a doubly-higher resolution, the number of markers, in 3D, goes up by a factor of 8. Hence, the computational costs very rapidly go up with the resolution of the problem. Subsequently, making use of more computational resources becomes necessary.

In the past, it used to be the case that increasing clock frequencies was the method to achieve better performance in serial programming applications. However, this trend is loosing speed and cannot be relied on for further gains due to the associated power consumption. Moreover, the instruction level parallelism (ILP), which relies on compilers or chip hardware to identify parallelizable code segments, does not results in further significant gains. ILP uses mainly three approaches for parallelization; out-of-order execution, speculative execution, and branch prediction. Further gains from ILP would require longer horizons of branch prediction, and more power consumption from unused computation results.

The aforementioned difficulties have steered the multiprocessor industry away from single core CPUs to multi-core and many-core architectures. This approach, parallel computing, allows for higher Floating-point Operations Per Second (FLOPS). However, parallel com-

puting requires special programming practices to leverage the compute power. There exist three main models for parallel computing; distributed-memory multiprocessor architecture, shared-memory multiprocessor architecture, and compute accelerator model.

In distributed-memory multiprocessor architecture, a set of nodes are connected by a high-speed network technology such as Infiniband. Each node consists of a processor and some discrete memory, which is only accessible within each node. This drawback is justified for very large scale simulations where use of 10-40 threads is not sufficient. Message Passing Interface (MPI) is one way to leveraged this methodology.

In the shared-memory multiprocessor architecture, a set of compute cores (usually 4-20) reside on the same address space. This facilitates memory access process and is considered useful for moderately large simulation where use of small number of cores is sufficient. Open-MP is one way to leverage this methodology.

Lastly, making use of compute accelerators, devices separated from the CPU, is another way to achieve parallelism. Graphics Processing Unit (GPU) architecture, which features its own memory and many lightweight parallel compute cores, is an example of this model. Compute Unified Device Architecture (CUDA) is an example Application Programming Interface (API) to leverage GPU computing.

The aim of the present work is to make use of the last two model architectures to leverage parallel computing on compute accelerators and multiprocessor on shared-memory architecture.

Part III

Proposed Directions of Research

Lagrangian fluid solvers such as SPH are well-known to be computationally expensive despite their natural capabilities in FSI simulations. Hence, as stated in Part I, the main objective of the present work is to make use of the latest developments in the numerical methods in SPH, and to apply new modifications where necessary, to solve the fluid phase of a FSI problem *faster* and more *accurate*. Furthermore, parallelizing the computations is necessary as explained in Part I, and Section §6. Below, we provide details of the proposed directions of research. These proposals either aim at improving the state-of-the-art SPH in order to achieve a more accurate and efficient fluid solver, or target hardware architectures to leverage parallelization in order to achieve more efficient solvers.

7 Second-Order Accurate SPH

The original SPH formulation expressed in Eq. 2 has second-order accuracy. In other words, the associated error scales by $O(h^2)$. More specifically, in 1D, the expectation is that if one doubles the number of SPH particles (halves the kernel length), the error will go down by a factor of four. The same improvement in the quality of the solution would call for a particle count increase by a factor of four and eight for 2D and 3D problems, respectively. In addition, if a second-order accurate scheme is compared with a first-order accurate scheme, the associated error will be smaller for the second-order scheme for the same problem size. Therefore, in numerical analysis a second-order accurate scheme is often preferred over first-order accurate schemes. In the current work, it is proposed that by taking advantage of consistent discretization for the first order (∇) and the second order (∇^2) operators the second-order accuracy of the SPH formulation is retained, which ultimately leads to a more accurate and efficient fluid solver.

The second-order accuracy of the original SPH formulation is not held in practice when the integral is approximated with a summation of discrete markers. Hence, special modifications are required to alleviate this issue. Elaborating on the concept of convergence and accuracy, if the numerical solution matches the first m terms of the Taylor expansion of the solution, then the numerical approximation is said to be $(m + 1)$ th-order accurate and has m th-order consistency. In practice, the consistency of the SPH formulation can be deteriorated due to factors such as (i) truncation of the support domain of the kernel near the boundaries, or (ii) irregular distribution of the particles. More specifically, if the Taylor series expansion of the kernel approximation is written with two terms, it will emerge that the SPH approximation is second-order accurate (or first-order consistent) under the kernel normalization condition $\int_S W(|x - x'|, h)dx' = 1$, where S is the support domain of the kernel function. However, when the kernel has a compact support, κh , the discrete form of the normalization condition is written over n Lagrangian particles that are inside the support domain as $(\sum_{b=1}^n W_{ab}\Delta x)$. The error associated with the inconsistency of this discretization is scaled by $O(1/\sqrt{n})$ for truly-random distribution, and $O(1/n)$ for quasi-ordered distribution of the n particles. These scaling show (i) the error is reduced by increasing the number of particles in the support domain regardless of particle distribution and most importantly (ii) how particle distribution in quasi-ordered setting leads to better error scaling in comparison to truly-random distribution. Therefore, in practical applications where Lagrangian markers are randomly distributed, renormalization techniques should be used to improve upon such defects and retain the consistency of the normalization condition. Ultimately, this procedure will retain the second-order accuracy of the formulation.

If the reason for loss of accuracy is irregular particle distribution, the renormalization technique to retain the second-order accuracy of the gradient operator draws on Eq. 9 as follows:

$$\langle \nabla f \rangle_i = \sum_{j \in \mathcal{S}(i)} V_j(f_j - f_i) \mathbf{G}_i \nabla_i W_{ij}, \quad (34)$$

where \mathbf{G}_i (for 3D problems) is a 3×3 symmetric normalization matrix whose mn element of its inverse is expressed as [51, 73, 25]:

$$(\mathbf{G}_i^{-1})^{mn} = - \sum_j r_{ij}^m \nabla_{i,n} W_{ij} V_j , \quad (35)$$

where r_{ij}^m , and $\nabla_{i,n} W_{ij}$ are the m th and n th components of vectors \mathbf{r}_{ij} and $\nabla_i W_{ij}$ respectively. Eq. 34 guarantees that the approximation of the gradient is accurate for a linear tensor and may be obtained with various methods [73, 51].

Moreover, following Eq. 13 and applying the correction factor \mathbf{L}_i , the consistent second-order operator is expressed as [88]:

$$\langle \nabla \cdot \nabla f \rangle_i = \sum_j 2V_j (\mathbf{L}_i : \mathbf{e}_{ij} \otimes \nabla_i W_{ij}) \left(\frac{f_i - f_j}{r_{ij}} - \mathbf{e}_{ij} \cdot \nabla f_i \right) , \quad (36)$$

where “ \otimes ” represents the dyadic product of the two vectors \mathbf{e}_{ij} and $\nabla_i W_{ij}$, and “ $:$ ” represents the double dot product of two matrices. In Eq. 36, \mathbf{L}_i (for 3D problems) is a 3×3 symmetric matrix with 6 unknowns which may be obtained by solving a linear system of equations [25]. The required 6 independent equations may be obtained by expanding the following equation for the upper/lower triangular elements of a 3×3 matrix, e.g. $m = 1, n = 1, 2, 3$, $m = 2, n = 2, 3$, and $m = 3, n = 3$.

$$-\delta^{mn} = \sum_j (A_i^{kmn} e_{ij}^k + r_{ij}^m e_{ij}^n) (L_i^{op} e_{ij}^o \nabla_{i,p} W_{ij} V_j) , \quad (37)$$

where the elements of the third order tensor A_i can be obtained from

$$A_i^{kmn} = \sum_j r_{ij}^m r_{ij}^n G_i^{kq} \nabla_{i,q} W_{ij} V_j . \quad (38)$$

Detailed information about the procedure of obtaining the elements of the \mathbf{L}_i is explained in [38].

Ultimately, the above formulations would retain the second-order accuracy of a SPH solver and result in a more accurate solution which can reduce the computational costs.

8 Incompressible SPH via Poisson Pressure Equation

Solving fluid dynamics equations using the SPH method is generally done in two ways [39]. The first one is the weakly incompressible SPH that approximates incompressibility by assuming a small Mach number ($Ma < 0.1$) [62, 61]. The second one [39] is incompressible SPH where incompressibility is enforced by solving a Poisson equation for pressure which involves a source term proportional to either the velocity divergence ($\nabla \cdot \mathbf{v}$) [21], or density variation ($D\rho/dt$) [81, 43]. Using incompressible SPH is more efficient when it comes to problems with moderate to high Raynolds number (Re), since the use of the weakly compressible SPH would require a very small time steps. However, there does not still exist an Implicit SPH method that is globally adopted due to deficiencies associated with existing methods.

More specifically, if the pressure equation is formed with the density-invariance condition, $D\rho/dt = 0$, the solver can maintain a uniform particle distribution, although the pressure distribution features high unrealistic fluctuations. In contrast, if the pressure equation is formed via the divergence-free velocity condition, $\nabla \cdot \mathbf{v} = 0$, smoother pressure distribution may be obtained, yet particle clustering and large density errors will occur [6].

Hence, there have been investigations to resolve these issues by combining the above schemes and taking advantage of their assets while shadowing their drawbacks. In such modified methods, both the divergence-free velocity, and the density-invariance conditions are aimed to be satisfied. In [71], it was proposed that two Poisson equations, one to satisfy zero-density-variation and one to satisfy divergence-free-velocity condition, be solved at two intermediate states for each time step. In [40, 39], intermediate iterations are performed at each time step to reduce the density error while maintaining accurate pressure distribution. In [6], both the zero-density-variation and the divergence-free-velocity conditions are com-

bined with a relaxation coefficient which weighs each part based on the initial separation of particles. This last study differs from the previous studies in that it solves only one Poisson equation at each time step, hence is thought to be more efficient.

As a consequence of solving a Poisson equation to form a (Semi-)Implicit solution for Navier-Stokes equations, one has to solve a linear system of equation of form $\mathbf{Ax} = \mathbf{b}$ whose size scales linearly with number of particles. These linear systems are sparse with no constant sparsity pattern; as the interacting particles redistribute, the coefficients of the matrix \mathbf{A} changes. Hence, in contrast to Eulerian methods where the connectivity matrix is constant, additional cost is to be paid in Lagrangian methods when forming the coefficient matrix of the linear system. Ultimately, the efficiency of such incompressible SPH methods are usually tied to the efficiency of forming and solving the resulting linear system of equations. It is worth mentioning that this procedure can be eliminated if a matrix-free method can be used for solving the system of equations similar to what studied in [43, 82].

To the best of our knowledge, a matrix-free approach for forming and solving linear system of equations resulting from the modified Poisson equation discussed in [6, 71, 39, 40] has not been investigated. Moreover, it was stated in [39] that the iteration count to obtain a compressibility as tight as 1% is less than $O(10)$. For large problems with millions of particles, forming a sparse matrix, storing data, and moving data around at each time step is very inefficient when only a few iterations are required to solve the system. Furthermore, it is a very well established fact that with today's computational capabilities, the computation is cheap while moving around data is about two orders of magnitude more expensive. Hence, it is proposed in the present work that the use of matrix-free methods in forming and solving such linear systems would be beneficial and may be considered.

9 Incompressible SPH via Kinematic Constraints

Constrained Fluids SPH (CFSPH) is a different flavor of the SPH methodology that tackles the fluid dynamics problem from a different angle and via different tools. In this methodology, the constant density is enforced by imposing constraints between SPH markers rather than obtaining a pressure field which results in a velocity field that satisfies the continuity equation. More specifically, holonomic constraints ² are enforced in order to match the current density at the location of marker i , with the rest density ρ_0 . The density constraint on marker i is expressed as follows:

$$C_i^f = \frac{\rho_i - \rho_0}{\rho_0} = 0 . \quad (39)$$

Now if one decides to satisfy the density constraint at the velocity level, $\dot{C}_i^f = 0$, which is equivalent to constraining $D\rho/Dt$ to be 0, a very interesting property will emerge as follows:

$$\begin{aligned} \dot{C}_i^f &= \frac{d}{dt} \left(\frac{\rho_i}{\rho_0} - 1 \right) \\ &\approx \sum_j \frac{m_j}{\rho_0} \frac{dW_{ij}}{dt} \\ &= \sum_j \frac{m_j}{\rho_0} \frac{dW_{ij}}{d\mathbf{x}_{ij}} \cdot \frac{d\mathbf{x}_{ij}}{dt} \\ &= \sum_j \frac{m_j}{\rho_0} \nabla W_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) \\ &= - \sum_j \frac{m_j}{\rho_0} \nabla W_{ij} \cdot (\mathbf{v}_j - \mathbf{v}_i) = - \langle \nabla \cdot \mathbf{v} \rangle_i . \end{aligned} \quad (40)$$

Above, the last equality is followed from Eq. 9, by replacing the f with \mathbf{v} [52]. The first and the last equality in Eq. 40 form the mass conservation, $(1/\rho(D\rho/Dt)) = -\nabla \cdot \mathbf{v}$, which for incompressible flows ($D\rho/Dt = 0$) reduces to $\nabla \cdot \mathbf{v} = 0$. Hence, by imposing a density constraint at the velocity level it is guaranteed that $\nabla \cdot \mathbf{v} = 0$ is satisfied. Satisfying both the $D\rho/Dt = 0$ and the $\nabla \cdot \mathbf{v} = 0$ is a feature that comes naturally with constrained fluids, while

²Constraints that are only a function of time and position level coordinates.

special treatment must be performed if the problem is approached from the traditional SPH angle, as explained in §8. This future of the constrained fluids implies a natural advantage of this method over traditional SPH method and, to the best of our knowledge, has not been clearly exploited in the literature of the subject.

Elements of each row of the Jacobian matrix (Φ_q or D in $\Phi_q \dot{q} = D\dot{q} = \nu$, where $\Phi = \dot{C}_f$ for this constraint), is simply the coefficient of the velocities (time derivative of the generalized coordinate, \dot{q}) in Eq. 40, and can be obtained from the diagonal D_{ii}^f and off diagonal D_{ij}^f terms of Eq. 40 as follows:

$$\dot{C}_i^f = \sum_j \frac{m_j}{\rho_0} \nabla W_{ij} \mathbf{v}_i - \sum_j \frac{m_j}{\rho_0} \nabla W_{ij} \mathbf{v}_j , \quad (41)$$

where

$$\begin{aligned} \mathbf{D}_{ii}^f &= \frac{1}{\rho_0} \sum_{k \neq i} m_k \nabla W_{ik} \\ \mathbf{D}_{ij}^f &= -\frac{m_j}{\rho_0} \nabla W_{ij}. \end{aligned} \quad (42)$$

Each density constraint on a marker contributes to a single row in the full Jacobian matrix which has $3n$ columns, where n is the total number of fluid markers. This matrix is sparse and its rows have three values at the columns corresponding to the current marker i and three values for each marker j within the support domain of i .

In terms of boundary handling in the constrained fluids, in the introductory work of the subject [12] it was suggested that additional constraints be imposed on the markers close to the boundaries. However, the constraint was only posed as a no-penetration condition, while a no-slip condition, which substantially distinguishes fluids' behavior from behaviors of granular materials at the wall, has not been investigated. Reformulation of the problem to incorporate the no-slip condition could significantly increase the usefulness of this methodology, especially for internal flows where the fluid-wall interaction determines the flow characteristics.

Imposing the no-slip condition is potentially possible using different approaches. (i)

Imposing additional constraints at the velocity level in a way that the extrapolation of the velocity of the fluid particles at the wall is the prescribed wall velocity. (ii) Making use of BCE markers as discussed in §4 and imposing constraints on the velocities in order to set the relative velocity at the wall to zero. (iii) Imposing a high friction coefficient to the near-wall contacts (interactions). (iv) Making use of BCE markers, yet finding the velocity and the pressure of the BCE markers from Eqs. 32 and 33 which guarantees no-slip and no-penetration conditions. Cases (i) and (ii) have similar natures, but the accuracy of (ii) should be higher since the error due to domain truncation is relatively lower in this case, although it would be more expensive since additional markers have to be created. Furthermore, the first two proposed methods have a more implicit nature in calculating velocities than the last two. Hence, it is predicted that they would result in a more stable solution, yet they are more computationally expensive as more constraints have to be satisfied at each time step.

Another modification to constrained fluids methodology is to incorporate the consistent formulation introduced in §7 in the way that constraints are imposed. Specifically, if one aims at satisfying $\nabla \cdot \mathbf{v} = 0$, a more accurate expansion of Eq. 41 can be written from Eq. 34 as:

$$\dot{C}_i^f = \sum_j \frac{m_j}{\rho_0} \mathbf{G}_i \nabla W_{ij} \cdot \mathbf{v}_i - \sum_j \frac{m_j}{\rho_0} \mathbf{G}_i \nabla W_{ij} \cdot \mathbf{v}_j , \quad (43)$$

where \mathbf{G}_i is obtained from Eq. 35. Hence, the elements of the Jacobian matrix in Eq. 42 will be modified in a way that second-order accuracy in first-order spatial operator is obtained.

Finally, the last modification to constrained fluids formulation is to evaluate the effectiveness of different methods when including viscous forces into equations. To the best of our knowledge, the viscous forces have not been fully incorporated in the constrained fluids methodology.

10 Variable Resolution SPH

Variable resolution methods are adopted in mature numerical schemes in order to improve the efficiency of simulations. For instance, in Eulerian Computational Fluid Dynamics (CFD), Adaptive Mesh Refinement (AMR) is used to reduce the computation costs. Although there have been similar early attempts in SPH, a method which shows a good trade-off between accuracy and efficiency is yet to be proposed. Generally speaking, in variable, or adaptive, resolution SPH, the idea is to decrease the characteristic length of Eq. 2 at specific regions inside the domain. This refinement improves the local accuracy of the solution while having larger characteristic length away from the region of interest decreases the computational costs. This is important because in many physical problems there exist zones where higher accuracy is required due to the presence of large field gradients, and large regions where high resolution would unjustifiably burden the computational effort. For instance, the solution of fluid flows shows a large velocity gradient near walls where the velocity has to become zero, and accordingly high resolution is needed only close to the wall. Hence, using adaptive resolution methods becomes essential.

There have been two major research directions when it comes to variable resolution SPH: *i*) variable smoothing length methods [13, 67], and *ii*) particle refinement methods [49, 26, 74, 91, 8]. The difficulty in dealing with variable resolution methods is handling extra terms that appear due to dh/dt , where h is the kernel length, while the difficulty in particle refinement methods is to conserve mass and momentum during the refinement and de-refinement procedures. Schemes of type (*ii*) may further be categorized as : *a*) schemes in which particles are added or deleted depending on how much resolution is required at a particular location; *b*) schemes where particles are coalesced when lower resolution is acceptable and are split when higher resolution is necessary; and, *c*) schemes which solve two fluid systems with different resolutions and couple the two simulations by making use of a hypothetical overlap zone. By and large, all the aforementioned schemes in this category share the common concept of *particle addition and deletion*, which is necessary for the conservation

of transport quantities.

Chronologically, research in variable resolution SPH methods started in astrophysical applications [65, 47]. Later, adaptive particle insertion and deletion, which is categorized in schemes of type (*ii*), was studied by Lastiwka et al. [49]. The basic ideas that are almost common in similar approaches are as follow:

1. Some criteria for adding and deleting particles are defined, and addition and deletion take place based on those criteria. For instance, the magnitude of the velocity gradient can be used as a criteria; where the local velocity gradient is large particles are inserted and where the local velocity gradient is low particles are deleted.
2. For newly added particles, mass and field variables should be interpolated with some schemes while preserving the linear and angular momentums, and other transport quantities.
3. The smoothing length and the mass of particles must be readjusted for the new distribution while preserving the mass.

A major step toward Adaptive Particle Refinement (APR) method was taken by Feldman and Bonet [26], whose work features split and merge procedures. In their work a large (*mother*) particle may split into some smaller (*daughter*) particles, or some *daughter* particles may coalesce to form a *mother* particle, depending on flow conditions. In their work a *global density refinement error* parameter was introduced which should be minimized in order to find an optimum configuration of the splitting process. This parameter depends on the separation (ϵh) and smoothing length (αh) and mass of the daughter particles where $\alpha, \epsilon \in [0, 1]$. It was also shown in their work that daughters particles' velocity should be the same as that of the mother particle in order to preserve the linear momentum. Angular momentum may be preserved only if mass is distributed symmetrically, but the lowest error may not be obtained in such a case.

López et al. [74] used the same procedure as Feldman and Bonet [26], but in their work α and ϵ were determined such that the error in gradient of the kernel is minimized. More specifically, the error in their work was defined based on rate of density change ($D\rho/dt$), rather than the density itself. More attention was given to the coalescing procedure part of the same formulation by Vacondio et al. [91]. In that work, the coalescing operation is done in pairs; each particle is merged with its nearest particle. This procedure happens once for each particle per time step. Consequently, the rate of coalescing might become smaller than the rate of splitting, if large particles are to split into more than two particles. Subsequently, Barcarolo et al. [8] proposed a new de-refinement procedure where the *mother* particle is not erased, but is turned off and excluded from the SPH summations. This ghost-like mother particle, however, follows the flow dynamics until it exits the refinement zone, when its state switched to “on” again. This idea was claimed to remedy the problem with the different rates of splitting and coalescing in the work by Vacondio et al. [91], however has higher computational cost.

The formulation of APR in the work by Feldman and Bonet [26] starts with calculating the local refinement error in density. Following Eq. 7, one can approximate the density of each marker by replacing $f(\mathbf{r})$ by ρ as:

$$\rho_i = \sum_{j \in \mathcal{S}(i)}^M \rho_j W(\mathbf{r} - \mathbf{r}_j, h) . \quad (44)$$

If a particle M inside the domain undergoes refinement and is split into D daughter particles, the new density estimation may be expressed as :

$$\rho_i^* = \sum_{j \in \mathcal{S}(i), j \neq M} \rho_j W(\mathbf{r} - \mathbf{r}_j, h) + \sum_{k \in \mathcal{S}(i)}^D \rho_k W(\mathbf{r} - \mathbf{r}_k, \alpha h) , \quad (45)$$

where mass of daughter particles are expressed as $m_d = \lambda_d m_M$, and $\sum_j^D \lambda_j = 1$ must hold in order to preserve mass. The *local refinement error* in formulations described above is

obtained as:

$$e_{\rho_i} = \rho_i - \rho_i^* = m_M W(\mathbf{r} - \mathbf{r}_M, h) + \sum_{k \in \mathcal{S}(i)}^D \rho_k W(\mathbf{r} - \mathbf{r}_k, \alpha h), \quad (46)$$

where α defines the smoothing length of the daughter particles and $\rho_k = m_k(\alpha h)^3$. The *global refinement error* in density is written as :

$$E_\rho = \int_{\Omega} e_\rho^2 dV, \quad (47)$$

which depends on smoothing length (αh), placement (ϵh) and mass (λm_M) of the daughter particles [26, 8]. On the other hand, the improvement of López et al. [74] aims at minimizing the rate of change of the gradient where the error is defined as:

$$\begin{aligned} e_{\nabla f_i} &= \langle \nabla f_i \rangle - \langle \nabla f_i^* \rangle; = V_M f_M \nabla_i W_{iM} - \sum_{k \in \mathcal{S}(i)}^D V_k f_k \nabla_i W_{ik} \\ &= m_M \left(\frac{f_M}{\rho_M} \nabla_i W_{iM} - \sum_{k \in \mathcal{S}(i)}^D \lambda_k \frac{f_k}{\rho_k} \nabla_i W_{ik} \right), \end{aligned} \quad (48)$$

where $\langle \nabla f \rangle_i$ is defined from Eq. 8. Above, a similar approach that was used to derive Eq. 45 and 46 has been used to derive the above equation but for gradient of a general function such as f . The global gradient error may be expressed similar to Eq. 47 as

$$E_{\nabla f} = \int_{\Omega} e_{\nabla f}^2 dV. \quad (49)$$

Using chain rule for differentiation, $D\rho/Dt$ can be related to the spatial variation of density ($\nabla \rho$), and following Eq. 48 the local error of the rate of change in density may be written as:

$$e_{\frac{D\rho_i}{Dt}} = \langle \frac{D\rho_i}{Dt} \rangle - \langle \frac{D\rho_i^*}{Dt} \rangle = m_M (\mathbf{v}_i - \mathbf{v}_M) \left(\nabla_i W_{iM} - \sum_{k \in \mathcal{S}(i)}^D \lambda_k \nabla_i W_{ik} \right), \quad (50)$$

where daughter particle's velocity (\mathbf{v}_d) was assumed to be the same as that of their mother particle (\mathbf{v}_M) to conserve the momentum. Finally, considering that the velocities do not change at each time step one can write the global error of the rate of change in density as :

$$E_{\frac{D\rho}{Dt}} = \int_{\Omega} \left(\frac{f_M}{\rho_M} \nabla_i W_{iM} - \sum_{k \in \mathcal{S}(i)}^D \lambda_k \frac{f_k}{\rho_k} \nabla_i W_{ik} \right)^2 dV . \quad (51)$$

The error formulation of Eq. 51 is believed to be more accurate than the error estimation based on Eq. 47. This is because of the fact that the term $\frac{D\rho}{Dt}$ explicitly appears in the continuity equation and minimizing this error would make more sense than minimizing the error in density.

By and large, all the above studies have been conducted for 2D configurations. However, recently Vacondio et al. [90] conducted a study for 3D configuration and found the optimum configuration of the daughter particles to be an icosahedron-shaped arrangement.

Lastly, in the work by Bian et al. [11], a type (iii) domain decomposition approach is used to separate different regions of the domain with different resolutions. In this work, each sub-domain has its own resolution, and the connection (coupling) between two sub-domains is achieved through an overlap region where the Lagrangian interpolation of the *state variables* inside the two sub-domains are matched through an iterative process.

In conclusion, there have not been a large number of research studies in the APR area and many new ideas and schemes may be tested in order to improve the accuracy and efficiency of this method. For instance, the local error in particles' arrangement expressed in Eq. 48 uses the crudest form of the gradient operator (see Eq. 8) to find the optimum arrangement of the particles. The question to be answered is whether or not using an approximation as simple as Eq. 8 would in fact give the optimum configuration of the daughter particles. As mentioned before in §1.2, Eqs. 9 and 10 give better approximation for the gradient operator and using the modified method discussed in §7, Eq. 34 results in even a more accurate estimation. Therefore, there might be a configuration that indeed is more accurate than the

ones studied before and is consistent with the second-order accuracy of SPH formulation, which needs to be investigated.

11 Simulation Framework for Large Scale SPH

When working with particle methods such as SPH, it is always the case that the computational cost of solving engineering problems is high. Therefore, it is imperative that the computational cost be split into many processing units in order to speed up the simulations. This is possible using both CPUs and GPUs.

If one decides to use only CPUs, parallelization is usually done either in shared memory architecture via OpenMP, or on distributed memory architecture over many nodes, which are interconnected via a computer networking technology such as Infiniband, using a Message Passing Interface (MPI). Parallelization of SPH methods on shared memory architecture is a relatively easier task than making an efficient parallelized solver over many nodes via MPI. For instance, in [56] simulations with up to 124 million particles were conducted via MPI over 1204 cores, or in [27] simulations with up to 2 million particles were carried out on a few hundred CPUs. The main drawback of this parallelization model is its high maintenance cost, and energy consumption.

On the other hand, considerable benefits may be obtained from using Graphic Processing Units (GPU), since their hardware architecture closely matches the computational demands of Lagrangian methods such as SPH. Since the arrival of Compute Unified Device Architecture (CUDA) by NVIDIA in 2007 to scientific computing communities, many SPH solvers were implemented on GPUs due to their high computing power and comparatively low energy costs per FLOP. For instance, a GPU implementation of SPH was reported in [34] to be one order of magnitude faster than CPU implementation. However, use of only one GPU may not be enough for solving large scale engineering applications which should consist of several million particles to accurately describe the physical processes. This limitation is

mainly due to the memory limitation on GPU devices. Hence, parallelizing the SPH solvers over multiple GPUs is essential.

Recently, multi-GPU SPH solvers were investigated in [24, 92]. Both of these studies took advantage of the single-GPU implementation of the DualSPHysics solver [22, 20]. Apart from these studies, limited work has been carried out in multi-GPU SPH implementation. This is in contrast to other types of particle-based simulations such as molecular dynamics (MD) which has already made use of this technology. For instance, many studies have been conducted on implementing multi-GPU version of the code LAMMPS [2, 89, 16].

Although SPH and MD share many similar computational concepts such as algorithms and underlying data structures, the two methods diverge when load balancing, density variation, and types of interaction come into play. SPH solvers have to deal with larger density variations and more extreme load imbalance. Hence, further treatment is required when parallelizing SPH code over multiple GPUs.

Spatial decomposition is the common technique that is used when parallelizing particle-based simulation. In this approach, which was used in [24, 92], sub-domains of the physical system are assigned to different processing units (GPUs). Communication between units is obtained with MPI routines. However, when increasing the number of the MPI processes, two sources of efficiency loss enter the problem; (*i*) data exchange and (*ii*) synchronization. The poor weak-scaling observed in [92], was attributed to aforementioned factors [24]. To alleviate this drawback, asynchronous communication was used in [24] to overlap communication with computation. Moreover, two dynamic load balancing methods were used in [24].

In summary, it is proposed in the current study that a multi-GPU framework be created to allow for solving problems with millions of particles. This framework would include the features discussed in §7, and §10. Hence, it would advance the state-of-the-art HPC at this subject.

12 The Performance-Gain Project

Increasing the efficiency of a multi-physics solver is possible by using *new numerical methods* and/or leveraging emerging *hardware and software solutions*. Each of these two paths will be pursued and resulting outcomes implemented in the current framework. In establishing *The Performance-Gain Project*, my goal is as follows: establish a collection of problems that will be monitored for two decades; i.e., for longer than my PhD program. The goal is to understand, in terms of performance gains, what gains are due to emerging hardware architectures (GPU, Phi, FPGA, etc.), to new software approaches (Charm++ [46], CUDA [66], etc.), to new modeling techniques (modeling friction and contact via DVI, for instance), or to new numerical methods (APGD vs. Gauss-Jacobi, for instance [57]). To this end, I plan to establish the set of benchmark problems, which fall into one of two categories: standard problems and real-world applications.

Regarding *standard problems*, by which we may evaluate the effectiveness of a *single* strategy, we are to rely on fluid mechanics' basic problems such as dam break, Poiseuille flow, lid-driven cavity, and flow around cylinder. As far as the flexible body dynamics solver is concerned, tests such as cantilever beam and plate are to be used.

Regarding *real-world applications*, by means of which we may evaluate the overall performance of the multi-physics framework studied here, we propose certain challenging applications such as fording simulation over rigid and deformable terrains. These scenarios require a mature multi-physics solver and stretch its fluid dynamics and deformable and rigid body dynamics support.

Part IV

Preliminary Work

13 Numerical Methods

13.1 Incompressible SPH with Semi-Implicit Time Integration

The preliminary implementation of the fluid solver of the current work draws on the work in [43]. In this method, the continuity equation, Eq. 5, is discretized in time using forward Euler method, and in space using the SPH discretization method of Eq. 9. Therefore, the discretized form of the continuity equation is written as :

$$\frac{\rho_i(t + \Delta t) - \rho_i(t)}{\Delta t} = \sum_{j \in \mathcal{S}(i)} m_j \mathbf{v}_{ij}(t + \Delta t) \cdot \nabla W_{ij}, \quad (52)$$

where incompressibility is imposed by setting $\rho_i(t + \Delta t) = \rho_0$.

The NS equations, Eq. 6, are discretized by making use of Chorin's two-step projection method [18]: first, the intermediate velocity, u^* , is obtained with the known values of the current velocities and the values of the RHS terms of Eq. 6 except the pressure term $\nabla p/\rho$; then, the new velocity is obtained by correcting the intermediate velocities u^* with pressure information at new time step. In IISPH the pressure at the new time step is obtained by solving a Poisson equation with a source term that is related to the zero-density-invariant condition, $d\rho/dt = 0$. More specifically, the RHS of the momentum equations is decomposed into the pressure and the non-pressure components of the force acting at the location of marker i as:

$$m_i \frac{\mathbf{v}_i(t_l + \Delta t) - \mathbf{v}_i(t_l)}{\Delta t} = \mathbf{f}_i^p(t_l) + \mathbf{f}_i^{np}(t_l + \Delta t), \quad (53)$$

where \mathbf{f}_i^p and \mathbf{f}_i^{np} represent the pressure forces and non-pressure forces acting on marker i ,

respectively. By manipulating the above equation it will emerge that:

$$\mathbf{v}_{i,l+1} = \left(\mathbf{v}_{i,l} + \frac{\mathbf{f}_{i,l}^{np}}{m_i} \Delta t \right) + \frac{\mathbf{f}_{i,l+1}^p}{m_i} \Delta t = \mathbf{v}_{i,l}^{np} + \mathbf{v}_{i,l+1}^p, \quad (54)$$

where $\mathbf{v}_{i,l}^{np} = \mathbf{v}_{i,l} + \mathbf{f}_i^{np}/m_i \Delta t$ is the intermediate velocity, $\mathbf{v}_{i,l+1}^p = \mathbf{f}_{i,l+1}^p \Delta t/m_i$ is the velocity correction, and subscripts l and $l+1$ are selected to indicate the discretized values at time t_l and $t_l + \Delta t$, respectively.

In contrast to the original Chorin's projection method and other ISPH flavors, which obtain the pressure equation by imposing the divergence-free velocity condition, $\nabla \cdot \mathbf{v} = 0$, in IISPH the pressure equation is obtained by imposing $d\rho/dt = 0$. The choice of density-invariance condition leads to a more uniform particle distribution and lower density error, although smoother pressure field is obtained if one uses the divergence-free velocity condition to create the velocity-pressure coupling [6]. Hence, in Eq. 52, density is also divided into the pressure and non pressure parts. Subsequently, the intermediate density resulting from the intermediate velocity $\mathbf{v}_{ij,l}^{np}$ is expressed as:

$$\rho_{i,l}^{np} = \rho_{i,l} + \Delta t \sum_{j \in \mathcal{S}(i)} m_j \mathbf{v}_{ij,l}^{np} \nabla W_{ij}. \quad (55)$$

By subtracting the above equation from Eq. 52 it can be concluded that:

$$\rho_{i,l+1} - \rho_{i,l}^{np} = \Delta t \sum_{j \in \mathcal{S}(i)} m_j (\mathbf{v}_{ij,l+1} - \mathbf{v}_{ij,l}^{np}) \nabla W_{ij}, \quad (56)$$

where $\rho_{i,l} = \sum_j m_j W_{ij}$. In order to preserve incompressibility, $\rho_{i,l+1} = \rho_0$ must be satisfied within the margin of numerical accuracy. Moreover, using Eq. 54 and the definition of $\mathbf{v}_{ij,l+1}^p$, the above equation is further simplified to the following form, which implicitly gives

the pressure equation:

$$\begin{aligned}
\rho_0 - \rho_{i,l}^{np} &= \Delta t \sum_{j \in \mathcal{S}(i)} m_j \mathbf{v}_{ij,l+1}^p \nabla W_{ij} \\
&= \Delta t \sum_{j \in \mathcal{S}(i)} m_j \left(\frac{\mathbf{f}_{ij,l+1}^p}{m_i} \Delta t \right) \nabla W_{ij} \\
&= \Delta t^2 \sum_{j \in \mathcal{S}(i)} m_j \left(\frac{\mathbf{f}_{i,l+1}^p}{m_i} - \frac{\mathbf{f}_{j,l+1}^p}{m_j} \right) \nabla W_{ij} ,
\end{aligned} \tag{57}$$

where pressure forces may be obtained from

$$\mathbf{f}_{i,l+1}^p = -m_i \sum_{j \in \mathcal{S}(i)} m_j \left(\frac{p_{i,l+1}}{\rho_{i,l}^2} + \frac{p_{j,l+1}}{\rho_{j,l}^2} \right) \nabla W_{ij}, \tag{58}$$

and the non-pressure forces from

$$\mathbf{f}_{i,l}^{np} = m_i \sum_{j \in \mathcal{S}(i)} m_j \Pi_{ij} + \mathbf{f}_b. \tag{59}$$

Above, \mathbf{f}_b represents the body forces and $\Pi_{ij} = -\frac{(\mu_i + \mu_j)\mathbf{x}_{ij}\nabla W_{ij}}{\bar{\rho}_{ij}^2(\mathbf{x}_{ij}^2 + \epsilon\bar{h}_{ij}^2)}$, in which $\bar{h}_{ij} = (h_i + h_j)/2$ is defined based on kernel function's characteristic length, and $\bar{\rho}_{ij} = (\rho_i + \rho_j)/2$.

Finally, by substituting Eqs. 55 and 58 into 57, a linear system of equations, with pressure as unknown, is obtained. A new set of velocities is obtained by correcting the intermediate velocity, $\mathbf{v}_{i,l}^{np}$, using Eq. 54. Position of particles can also be obtained using $\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_{i,l+1}\Delta t$. More detailed information about the pressure equation and its solution was explained in [72].

This implementation was subsequently used to solve several standard CFD problems such as the Poiseuille flow and the lid-driven cavity. The numerical results of the lid-driven cavity at $Re = 1$ is shown in Fig. 7. The CFD results are slightly differs from other numerical results obtained from Finite Volume approach. We attribute this discrepancy to several factors such the viscosity model as well as boundary condition treatments.

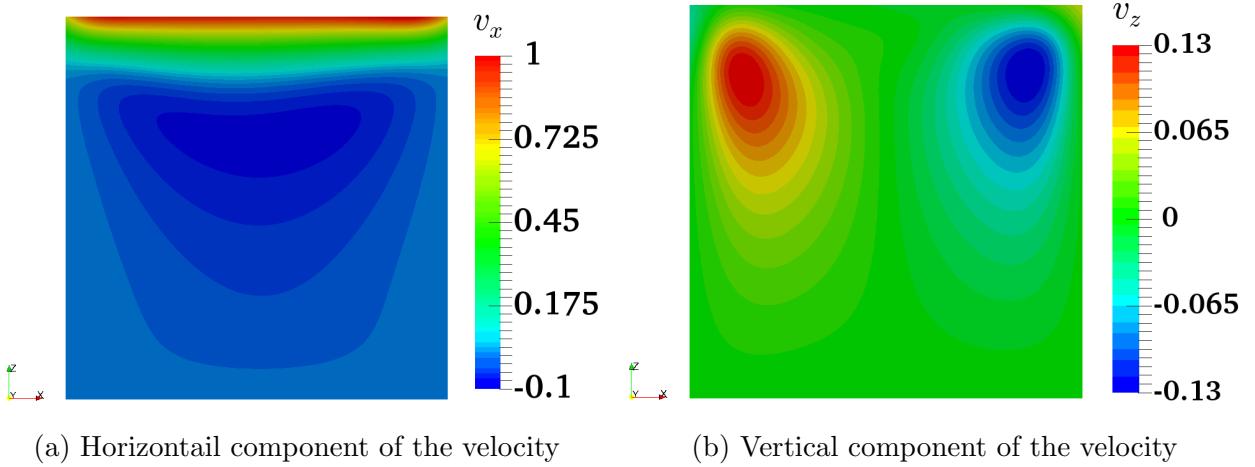


Figure 7: Numerical results of the IISPH method for a lid-driven cavity problem at $Re = 1$, $U_{wall} = 1$, $D = 1$ and $\nu = 1$, where D in the length of each side, and ν is the kinematic viscosity. The Lagrangian data was mapped to an Eulerian grid for visualization purposes.

14 Advanced Computing Aspects

14.1 Multi-Physics Framework

In order to solve fluid-solid interaction problems, a co-simulation framework was created in the present work. This framework features the fluid dynamics solver described in §13.1, rigid body dynamics described in §2 as well as 1D and 2D flexible bodies described in §3.1 and §3.2. The fluid-solid interaction is handled using the methodology described in §4. Hence, the current framework allows for solving many types of FSI problems.

A sample FSI simulation which includes the interaction of fluids and thin shells was shown in Fig. 8.

Another FSI simulation that features rigid body dynamics was demonstrated in Fig. 9.

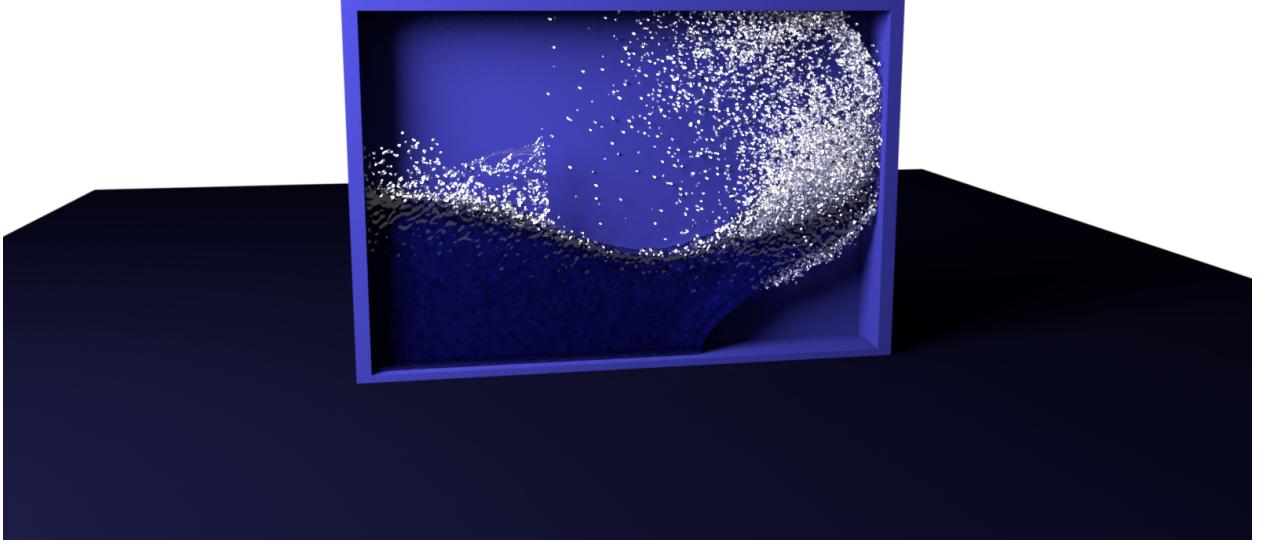


Figure 8: Fluid-solid interaction with deformable ANCF shells ([Full Video](#))

14.2 Parallelization

The FSI framework described in §14.1, was parallelized on both CPU and GPU. Regarding the flexible body dynamics part, the ANCF implementations of cable and shell elements were parallelized over multiple threads using OpenMP directives which aim at parallelizing the calculation of the internal forces and the Jacobian. Furthermore, AVX intrinsics were used for vectorization of the underlying lower level matrix operations at places where these operations may be optimized. These strategies result in 6-7x speed-up in calculation of the flexible body dynamics using 12 threads. Regarding the fluid dynamics solver, the IISPH method described in §13.1 was implemented on a single GPU. The use of a GPU for parallelization of the fluid phase results in about one order of magnitude faster solver when compared with the same solver parallelized over multi-thread CPU.

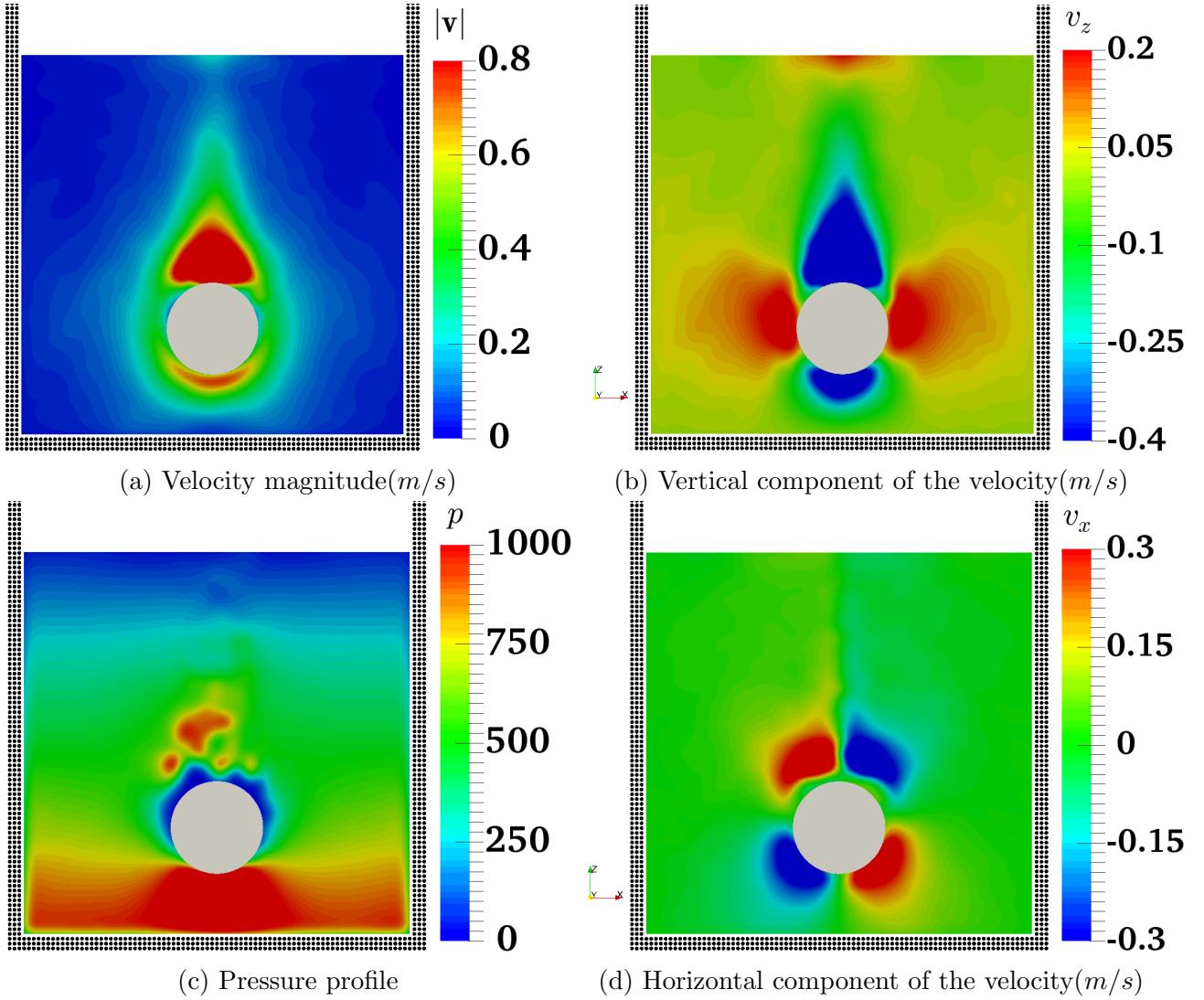


Figure 9: A cylinder drops into a tank of water and goes down under the gravity effect. For visualization purposes, the Lagrangian data were mapped to an Eulerian grid on the mid-plane parallel to $x - z$ plane. ([Full Video](#))

15 Validation Efforts

15.1 IISPH and CFSPH

15.1.1 Dam Break

The collapse of a liquid column is a standard test performed to validate an SPH implementation. To that end, we simulated the water-front propagation of a $L = 1$ cube of water placed on the leftmost side of a cubic container and released over a dry bed at $t(gH)^{1/2} = 0$. In the

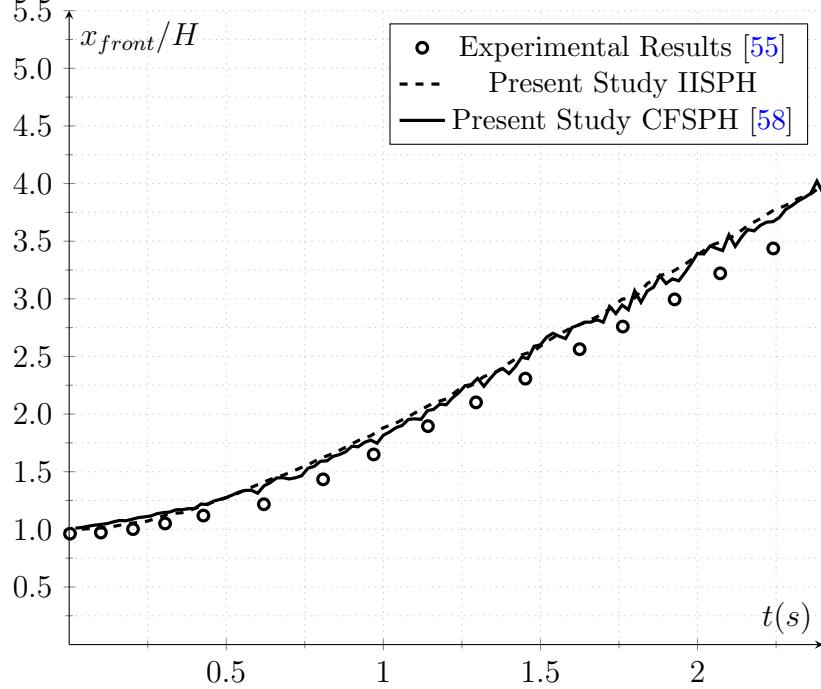


Figure 10: Comparison of water-front propagation between IISPH, CFSPH, and experimental results of Martin and Moyce [55].

3D case studied the flow is inviscid and $g = 1$. In Fig. 10, we compare the non-dimensional front position x_{front}/H as a function of the time scale, $t(gH)^{1/2}$, against experimental [55] and simulation results [1]. Numerical simulations were performed based on both the CFSPH and IISPH methods. At early stages, CFSPH agrees well with the experiment, yet it gradually over-predicts the experimental results after $t(gH)^{1/2} \approx 2$. We attribute this discrepancy to the inviscid flow assumption which becomes inaccurate as the flow speed (\propto tangent line in Fig. 10) increases and the effect of the viscosity term, $\nabla^2\mathbf{v}$, in the Navier-Stokes equations becomes significant. Moreover, smaller front propagation speeds are expected for viscous fluids. Thus, it stands to a reason that the numerical results over-estimate the experimental results.

15.1.2 Elastic Gate Experiment

Elastic gate simulation is a test used to validate the coupling between the fluid and solid phases. In this experiment, water is encapsulated in a cubic container. The container consists

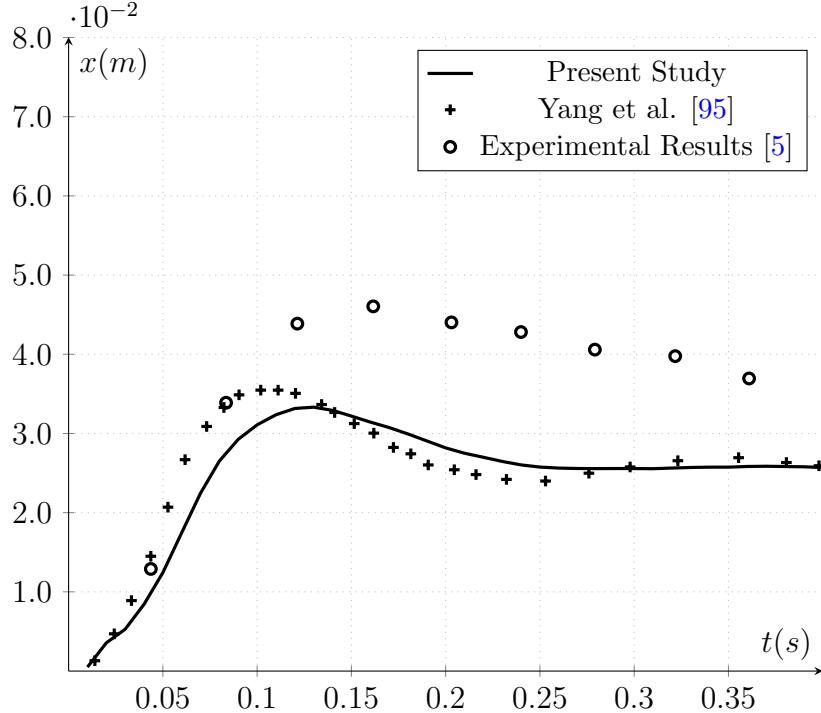


Figure 11: Comparison of the horizontal position of the tip of the elastic gate from the numerical results of the present study, the numerical results of the study by Yang et al. [95], and the experimental results of Antoci et al. [5].

of three rigid sides and one rectangular elastic side which is released at $t = 0$. Due to the hydrostatic pressure of the water, the elastic gate is deflected and subsequently the water is released. The position of the tip of the gate was measured in the experiment reported in [5]. The numerical simulation presented here resembles the configuration of the same experimental study. The results were shown in Fig. 11 along with another numerical study by Yang et al. [95].

The discrepancy between the numerical results and the experiment was attributed to the fact that the elasticity of the gate in the experiment decreases when it deforms [95, 44]. By using a constant Young modulus in the numerical simulation, the result of the deflection will under-predict the actual deformation of the gate as shown in Fig. 11.

16 Application Focus

16.1 Multi-Physics Model of Articular Cartilage

One of the goals of the current study is to develop a multi-physics model of the micro-structure of the articular cartilage. Articular cartilage is a biological tissue that provides wear-resistant, low-friction, force distribution between subchondral bone surfaces at synovial joints. The performance of this tissue is key to proper, painless joint motion. Structurally, cartilage consists of two distinct phases: a fluid phase, composed of water and electrolytes, and a solid phase, composed of collagen fibrils, chondrocytes, and proteoglycans [63].

The mechanics of AC has been modeled in the literature through continuum-based representations using either biphasic or triphasic formulations relying on constitutive properties derived from experimental observations [50, 33]. These formulations use the methods of composite materials for giving directional properties to the cartilage. While these approaches provide a holistic view of the tissue, they do not explicitly model the discrete fluid-solid interactions (FSI) on the ultra-scale level. The present investigation proposes an FSI approach to model the microstructure of the articular cartilage using the fundamental elements, as shown in Fig. 12.

In this approach :

- Collagen fibrils are modeled using absolute nodal coordinate formulation (ANCF) elastic beams where shear deformation is neglected.
- The articular cartilage's surface membrane is represented by compliant nonlinear ANCF shell elements which distribute external loading.
- The interstitial fluid is modeled in a Lagrangian fashion through SPH markers.

Modeling articular cartilage on the ultra-scale level through a discrete, fully coupled representation of fluid and solids provides more accurate insight into the biomechanical properties

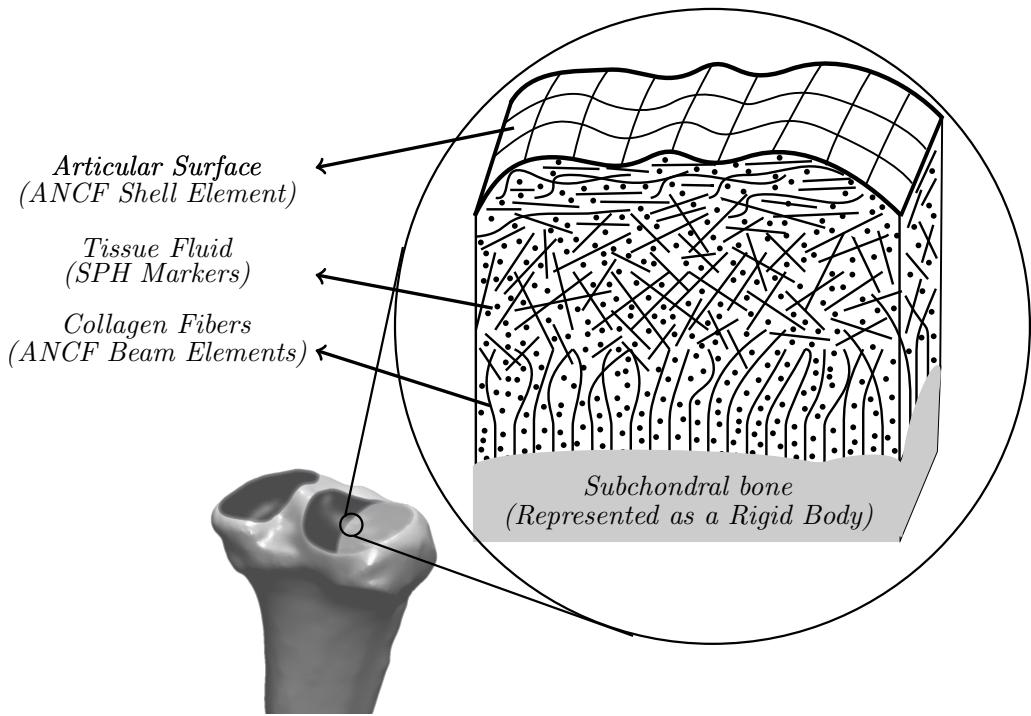


Figure 12: Multi-scale representation of knee articular cartilage

of the AC. In addition, we expect that this investigation will deepen our understanding of how macroscopic behavior is affected by the interplay of microstructural elements.

As a first step, we studied cartilage surface loading during walking to investigate if split line orientations correspond with principal strain directions in the cartilage surface. The two-step framework uses a multibody musculoskeletal model to predict tibiofemoral kinematics which are then imposed on a deformable surface model to predict surface strains. The deformable surface model uses absolute nodal coordinate formulation (ANCF) shell elements to represent the articular surface and a system of spring-dampers and internal pressure to represent the underlying cartilage. Simulations were performed to predict surface strains due to osmotic pressure, loading induced by walking, and the combination of both loading due to pressure and walking. Time-averaged magnitude-weighted first principal strain directions agreed well with split line maps from the literature for both the osmotic pressure and combined cases, as shown in Fig. 13.

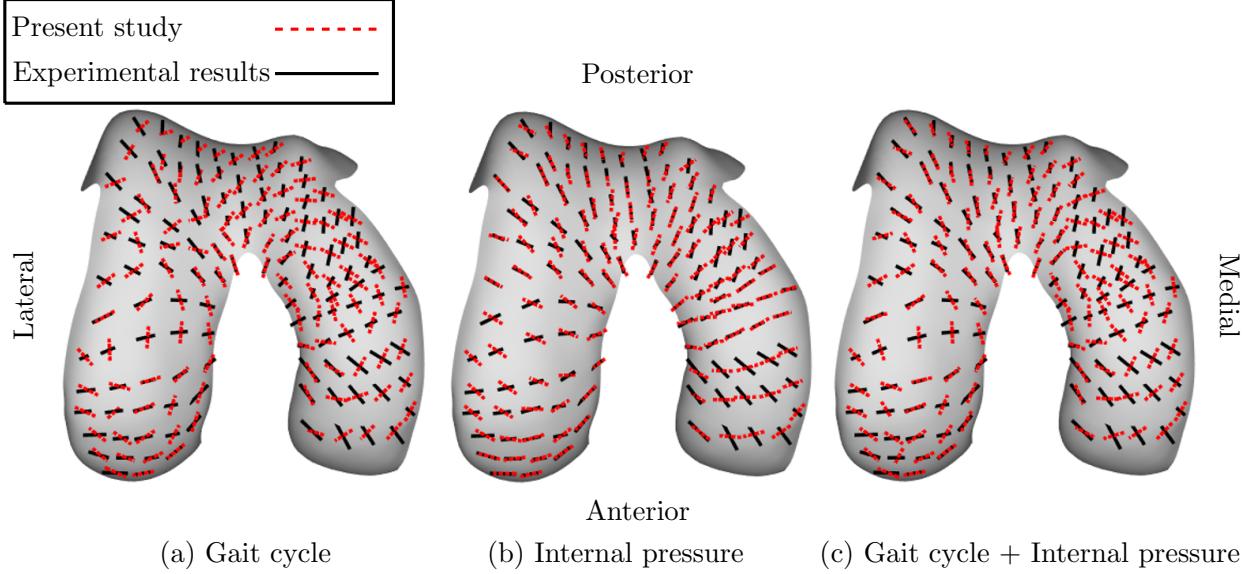


Figure 13: Comparison of the fiber alignment predicted by this study for different scenarios with experimental results ([9]) on the femur condyle.

As the next step, we are planning to generate 3D geometries of knee articular cartilage from medical images and have meshed it with ANCF shell elements. This mesh of shell elements encloses water and collagen fibrils. We expect that the mechanical interaction among beams, shells, and SPH markers determine the overall response of the AC. This, to the best of our knowledge, is a novel approach to model articular cartilage.

16.2 Underwater Robotics

As discussed earlier, the current framework allows for capturing the interaction of different physics, including rigid and flexible bodies with fluids. I plan to use this framework to simulate fully-resolved models of underwater robots. This fully-resolved approach is in contrast to conventional underwater vehicle simulations where the hydrodynamic effects acting on the vehicle are “modeled” via added mass, viscous damping, restoring and propulsion forces and moments [19]. I hope that the high-fidelity approach embrace for modeling the two-way coupling between the robot and fluid will allow better design and analysis of underwater robots engaged in complex maneuvers on rocky/sandy sea floor.

17 References

References

- [1] S Adami, XY Hu, and NA Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012.
- [2] Pratul K Agarwal, Scott Hampton, Jeffrey Poznanovic, Arvind Ramanthan, Sadaf R Alam, and Paul S Crozier. Performance modeling of microsecond scale biological molecular dynamics simulations on heterogeneous architectures. *Concurrency and Computation: Practice and Experience*, 25(10):1356–1375, 2013.
- [3] M. Anitescu. Optimization-based simulation of nonsmooth rigid multibody dynamics. *Mathematical Programming*, 105(1):113–143, 2006.
- [4] M. Anitescu and G. D. Hart. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *International Journal for Numerical Methods in Engineering*, 60(14):2335–2371, 2004.
- [5] Carla Antoci, Mario Gallati, and Stefano Sibilla. Numerical simulation of fluid–structure interaction by SPH. *Computers and Structures*, 85(11):879–890, 2007.
- [6] Mitsuteru Asai, Abdelraheem M Aly, Yoshimi Sonoda, and Yuzuru Sakai. A stabilized incompressible sph method by relaxing the density invariance condition. *Journal of Applied Mathematics*, 2012, 2012.
- [7] SW Attaway, MW Heinstein, and JW Swegle. Coupling of smooth particle hydrodynamics with the finite element method. *Nuclear engineering and design*, 150(2-3):199–205, 1994.
- [8] Daniel A Barcarolo, David Le Touzé, Guillaume Oger, and Florian De Vuyst. Adaptive particle refinement and derefinement applied to the smoothed particle hydrodynamics method. *Journal of Computational Physics*, 273:640–657, 2014.
- [9] Steven Below, Steven P. Arnoczky, Julie Dodds, Cynthia Kooima, and Norman Walter. The split-line pattern of the distal femur: A consideration in the orientation of autologous cartilage grafts. *Arthroscopy*, 18(6):613–617, 2002.
- [10] M. Berzeri and A.A. Shabana. Development of simple models for the elastic forces in the absolute nodal co-ordinate formulation. *J. of Sound and Vibration*, 235(4):539–565, 2000.
- [11] Xin Bian, Zhen Li, and George Em Karniadakis. Multi-resolution flow simulations by smoothed particle hydrodynamics via domain decomposition. *Journal of Computational Physics*, 297:132–155, 2015.
- [12] K. Bodin, C. Lacoursiere, and M. Servin. Constraint fluids. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):516–526, 2012.
- [13] Javier Bonet and Miguel X Rodríguez-Paz. Hamiltonian formulation of the variable-h sph equations. *Journal of Computational Physics*, 209(2):541–558, 2005.
- [14] K. Brenan and B. E. Engquist. Backward differentiation approximations of nonlinear differential/algebraic systems. *Mathematics of Computation*, 51(184):659–676, 1988.

- [15] L Brookshaw. A method of calculating radiative heat diffusion in particle simulations. In *Proceedings of the Astronomical Society of Australia*, volume 6, pages 207–210, 1985.
- [16] W Michael Brown, Peng Wang, Steven J Plimpton, and Arnold N Tharrington. Implementing molecular dynamics on hybrid high performance computers—short range forces. *Computer Physics Communications*, 182(4):898–911, 2011.
- [17] AK Chaniotis, D Poulikakos, and P Koumoutsakos. Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. *Journal of Computational Physics*, 182(1):67–90, 2002.
- [18] Alexandre Joel Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968.
- [19] T.S. Chuan. Modeling and simulation of the autonomous underwater vehicle. Master thesis, Department of Ocean Engineerig, Massachusetts Institute of Technology, 1999.
- [20] Alejandro JC Crespo, José M Domínguez, Benedict D Rogers, Moncho Gómez-Gesteira, S Longshaw, R Canelas, R Vacondio, A Barreiro, and O García-Feal. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH). *Computer Physics Communications*, 187:204–216, 2015.
- [21] Sharen J Cummins and Murray Rudman. An sph projection method. *Journal of computational physics*, 152(2):584–607, 1999.
- [22] RA Dalrymple, M Gomez-Gesteira, B Rogers, A Crespo, M Narayanaswamy, S Zou, and A Panizzo. SPHysics. https://wiki.manchester.ac.uk/sphysics/index.php/Main_Page. Accessed: 2015-02-12.
- [23] Tom De Vuyst, Rade Vignjevic, and JC Campbell. Coupling between meshless and finite element methods. *International Journal of Impact Engineering*, 31(8):1054–1064, 2005.
- [24] José M Domínguez, Alejandro JC Crespo, Daniel Valdez-Balderas, Benedict D Rogers, and Moncho Gómez-Gesteira. New multi-gpu implementation for smoothed particle hydrodynamics on heterogeneous clusters. *Computer Physics Communications*, 184(8):1848–1860, 2013.
- [25] R Fatehi and MT Manzari. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Computers & Mathematics with Applications*, 61(2):482–498, 2011.
- [26] J Feldman and J Bonet. Dynamic refinement and boundary contact forces in sph with applications in fluid flow problems. *International Journal for Numerical Methods in Engineering*, 72(3):295–324, 2007.
- [27] Angela Ferrari, Michael Dumbser, Eleuterio F Toro, and Aronne Armanini. A new 3d parallel sph scheme for free surface flows. *Computers & Fluids*, 38(6):1203–1217, 2009.
- [28] Joel H Ferziger and Milovan Peric. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.
- [29] AF Filippov. Classical solutions of differential equations with multi-valued right-hand side. *SIAM Journal on Control*, 5(4):609–621, 1967.

- [30] G Fourey, G Oger, D Le Touzé, and B Alessandrini. Violent fluid-structure interaction simulations using a coupled sph/fem method. In *IOP conference series: materials science and engineering*, volume 10, page 012041. IOP Publishing, 2010.
- [31] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(1):375–389, 1977.
- [32] Paul HL Groenenboom and Bruce K Cartwright. Hydrodynamics and fluid-structure interaction by coupled sph-fe method. *Journal of Hydraulic Research*, 48(S1):61–73, 2010.
- [33] JP Halloran, S Sibole, CC Van Donkelaar, MC Van Turnhout, CWJ Oomens, Jeffrey A Weiss, F Guilak, and A Erdemir. Multiscale mechanics of articular cartilage: potentials and challenges of coupling musculoskeletal, joint, and microscale computational models. *Annals of biomedical engineering*, 40(11):2456–2474, 2012.
- [34] Takahiro Harada, Seiichi Koshizuka, and Yoichiro Kawaguchi. Smoothed particle hydrodynamics on gpus. In *Computer Graphics International*, pages 63–70. SBC Petropolis, 2007.
- [35] E. J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume-I*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
- [36] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Eng. and Struct. Dynamics*, 5:283–292, 1977.
- [37] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interactiona review. *Communications in Computational Physics*, 12(02):337–377, 2012.
- [38] Wei Hu, Wenxiao Pan, Milad Rakhsha, and Dan Negrut. An Overview of an SPH Technique to Maintain Second-Order Convergence for 2D and 3D Fluid Dynamics. *Technical Report TR-2016-14*, 2017.
- [39] X. Y. Hu and N. A. Adams. An incompressible multi-phase SPH method. *Journal of Computational Physics*, 227(1):264–278, Nov 10 2007.
- [40] XY Hu and NA Adams. A constant-density approach for incompressible multi-phase sph. *Journal of Computational Physics*, 228(6):2082–2091, 2009.
- [41] Björn Hübner, Elmar Walhorn, and Dieter Dinkler. A monolithic approach to fluid–structure interaction using space–time finite elements. *Computer methods in applied mechanics and engineering*, 193(23):2087–2104, 2004.
- [42] Thomas JR Hughes, Wing Kam Liu, and Thomas K Zimmermann. Lagrangian-eulerian finite element formulation for incompressible viscous flows. *Computer methods in applied mechanics and engineering*, 29(3):329–349, 1981.
- [43] Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. Implicit incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):426–435, 2014.
- [44] Ebrahim Jahanbakhsh, Christian Vessaz, Audrey Maertens, and François Avellan. Finite volume particle method for fluid-structure interaction. In *Proceedings of the 10th International SPHERIC Workshop*, 2015.

- [45] L. O. Jay and D. Negrut. Extensions of the HHT- α method to Differential-Algebraic Equations in mechanics. *Electronic Transactions on Numerical Analysis*, 26:190–208, 2007.
- [46] Laxmikant V. Kale and Sanjeev Krishnan. CHARM++: A portable concurrent object oriented system based on c++. In *Proceedings of the Conference on Object Oriented Programming Systems, Languages and Applications*, pages 91–108, 1993.
- [47] S Kitsionas and A P Whitworth. Smoothed particle hydrodynamics with particle splitting, applied to self-gravitating collapse. *Monthly Notices of the Royal Astronomical Society*, 330(1):129–136, 2002.
- [48] Hans Petter Langtangen, Kent-Andre Mardal, and Ragnar Winther. Numerical methods for incompressible viscous flow. *Advances in Water Resources*, 25(8):1125–1146, 2002.
- [49] M Lastiwka, N Quinlan, and M Basa. Adaptive particle distribution for smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 47(10–11):1403–1409, 2005.
- [50] LePing Li, JTM Cheung, and W Herzog. Three-dimensional fibril-reinforced finite element model of articular cartilage. *Medical & biological engineering & computing*, 47(6):607, 2009.
- [51] LD Libersky, AG Petschek, TC Carney, JR Hipp, and FA Allahdadi. High strain Lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response. *Journal of Computational Physics*, 109(1):67–75, 1993.
- [52] SJ Lind, Rui Xu, PK Stansby, and Benedict D Rogers. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4):1499–1523, 2012.
- [53] C. Lötstedt and L. Petzold. Numerical solution of nonlinear differential equations with algebraic constraints I: Convergence results for backward differentiation formulas. *Mathematics of Computation*, 174:491–516, 46.
- [54] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, 1977.
- [55] J. C. Martin and W. J. Moyce. Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 244(882):312–324, mar 1952.
- [56] Pierre Maruzewski, David Le Touzé, Guillaume Oger, and François Avellan. Sph high-performance computing simulations of rigid solids impacting the free-surface of water. *Journal of Hydraulic Research*, 48(S1):126–134, 2010.
- [57] H. Mazhar, T. Heyn, A. Tasora, and D. Negrut. Using Nesterov’s method to accelerate multibody dynamics with friction and contact. *ACM Trans. Graph.*, 34(3), 2015.
- [58] H. Mazhar, A. Pazouki, M. Rakhsha, P. Jayakumar, and D. Negrut. A differential variational approach for handling fluid-solid interaction problems via Smoothed Particle Hydrodynamics. *Journal of Computational Physics (under review)*, 0:0, 2017.
- [59] C Michler, SJ Hulshoff, EH Van Brummelen, and René De Borst. A monolithic approach to fluid–structure interaction. *Computers & fluids*, 33(5):839–848, 2004.

- [60] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.
- [61] J. J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399–399, 1994.
- [62] J. P. Morris, P. J. Fox, and Y. Zhu. Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136(1):214–226, 1997.
- [63] Van C Mow, Anthony Ratcliffe, and A Robin Poole. Cartilage and diarthrodial joints as paradigms for hierarchical materials and structures. *Biomaterials*, 13(2):67–97, 1992.
- [64] D. Negrut, R. Rampalli, G. Ottarsson, and A. Sajdak. On the use of the HHT method in the context of index 3 Differential Algebraic Equations of Multibody Dynamics. *ASME Journal of Computational and Nonlinear Dynamics*, 2, 2007.
- [65] R. P. Nelson and J. C. B. Papaloizou. Variable smoothing lengths and energy conservation in smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 270(1):1–20, 1994.
- [66] NVIDIA Corporation. NVIDIA CUDA Developer Zone. Available online at <https://developer.nvidia.com/cuda-downloads>, 2012.
- [67] Guillaume Oger, Mathieu Doring, Bertrand Alessandrini, and Pierre Ferrant. Two-dimensional sph simulations of wedge water entries. *Journal of computational physics*, 213(2):803–822, 2006.
- [68] Jong-Shi Pang and David Stewart. Differential variational inequalities. *Mathematical Programming*, 113(2):345–424, 2008.
- [69] Arman Pazouki, Radu Serban, and Dan Negrut. A high performance computing approach to the simulation of fluid-solid interaction problems with rigid and flexible components. *Archive of Mechanical Engineering*, 61(2):227–251, 2014.
- [70] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.
- [71] Jacek Pozorski and Arkadiusz Wawreńczuk. SPH computation of incompressible viscous flows. *Journal of Theoretical and Applied Mechanics*, 40(4):917–937, 2002.
- [72] M. Rakhsha, A. Pazouki, and D. Negrut. Incompressible implicit SPH. Technical Report TR-2016-14: <http://sbel.wisc.edu/documents/TR-2016-14.pdf>, Simulation-Based Engineering Laboratory, University of Wisconsin-Madison, 2016.
- [73] P. W Randles and L. D. Libersky. Smoothed Particle Hydrodynamics: some recent improvements and applications. *Computer methods in applied mechanics and engineering*, 139(1-4):375–408, 1996.
- [74] Yaidel Reyes López, Dirk Roose, and Carlos Recarey Morfa. Dynamic particle refinement in sph: application to free surface flow and non-cohesive soil simulations. *Computational Mechanics*, pages 1–11, 2013.
- [75] PB Ryzhakov, Riccardo Rossi, SR Idelsohn, and E Oñate. A monolithic lagrangian approach for fluid–structure interaction problems. *Computational mechanics*, 46(6):883–899, 2010.

- [76] Isaac Jacob Schönberg. Contributions to the problem of approximation of equidistant data by analytic functions. In *IJ Schoenberg Selected Papers*, pages 3–57. Springer, 1988.
- [77] A. A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge, England, fourth edition, 2013.
- [78] A. A. Shabana and R. Y. Yakoub. Three dimensional absolute nodal coordinate formulation for beam elements: Implementation and applications. *ASME Journal of Mechanical Design*, 123:614–621, 2001.
- [79] A. A. Shabana and R. Y. Yakoub. Three dimensional absolute nodal coordinate formulation for beam elements: Theory. *ASME Journal of Mechanical Design*, 123:606–613, 2001.
- [80] Ahmed A Shabana. Definition of the slopes and the finite element absolute nodal coordinate formulation. *Multibody System Dynamics*, 1(3):339–348, 1997.
- [81] Songdong Shao and Edmond Lo. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources*, 26(7):787–800, JUL 2003.
- [82] Barbara Solenthaler and Renato Pajarola. Predictive-corrective incompressible sph. In *ACM transactions on graphics (TOG)*, volume 28, page 40. ACM, 2009.
- [83] M Souli, A Ouahsine, and L Lewin. Ale formulation for fluid–structure interaction problems. *Computer methods in applied mechanics and engineering*, 190(5):659–675, 2000.
- [84] David E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.
- [85] David E. Stewart and Jeffrey C. Trinkle. An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and Coulomb friction. *International Journal for Numerical Methods in Engineering*, 39:2673–2691, 1996.
- [86] Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C Lin. Implicit formulation for sph-based viscous fluids. In *Computer Graphics Forum*, volume 34, pages 493–502. Wiley Online Library, 2015.
- [87] Hidenori Takeda, Shoken M Miyama, and Minoru Sekiya. Numerical simulation of viscous flow by smoothed particle hydrodynamics. *Progress of Theoretical Physics*, 92(5):939–960, 1994.
- [88] Nathaniel Trask, Martin Maxey, Kyungjoo Kim, Mauro Perego, Michael L. Parks, Kai Yang, and Jinchao Xu. A scalable consistent second-order {SPH} solver for unsteady low reynolds number flows. *Computer Methods in Applied Mechanics and Engineering*, 289:155 – 178, 2015.
- [89] Christian R Trott, Lars Winterfeld, and Paul S Crozier. General-purpose molecular dynamics simulations on gpu-based clusters. *arXiv preprint arXiv:1009.4330*, 2010.
- [90] R Vacondio, BD Rogers, PK Stansby, and P Mignosa. Variable resolution for sph in three dimensions: Towards optimal splitting and coalescing for dynamic adaptivity. *Computer Methods in Applied Mechanics and Engineering*, 300:442–460, 2016.

- [91] R Vacondio, BD Rogers, PK Stansby, P Mignosa, and J Feldman. Variable resolution for sph: a dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering*, 256:132–148, 2013.
- [92] Daniel Valdez-Balderas, José M Domínguez, Benedict D Rogers, and Alejandro JC Crespo. Towards accelerating smoothed particle hydrodynamics simulations for free-surface flows on multi-gpu clusters. *Journal of Parallel and Distributed Computing*, 73(11):1483–1493, 2013.
- [93] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- [94] Hiroki Yamashita, Antti I Valkeapää, Paramsothy Jayakumar, and Hiroyuki Sugiyama. Continuum mechanics based bilinear shear deformable shell element using absolute nodal coordinate formulation. *Journal of Computational and Nonlinear Dynamics*, 10(5):051012, 2015.
- [95] Qing Yang, Van Jones, and Leigh McCue. Free-surface flow interactions with deformable structures using an sph-fem model. *Ocean engineering*, 55:136–147, 2012.