

A symmetric positive definite formulation for monolithic fluid structure interaction

Avi Robinson-Mosher*, Craig Schroeder*, Ronald Fedkiw*

Stanford University, 353 Serra Mall Room 207, Stanford, CA 94305

Abstract

In this paper we consider a strongly coupled (monolithic) fluid structure interaction framework for incompressible flow, as opposed to a loosely coupled (partitioned) method. This requires solving a single linear system that combines the unknown velocities of the structure with the unknown pressures of the fluid. In our previous work, we were able to obtain a symmetric formulation of this coupled system; however, it was also indefinite, making it more difficult to solve. In fact in practice there have been cases where we have been unable to invert the system. In this paper we take a novel approach that consists of factoring the damping matrix of deformable structures and show that this can be used to obtain a symmetric positive definite system, at least to the extent that the uncoupled systems were symmetric positive definite. We use a traditional MAC grid discretization of the fluid and a fully Lagrangian discretization of the structures for the sake of exposition, noting that our procedure can be generalized to other scenarios. For the special case of rigid bodies, where there are no internal damping forces, we exactly recover the system of [4].

1. Introduction

Fluid structure interaction has been of increasing interest to the computational fluids community, due in part to the ability to address more computationally expensive problems with improved hardware. Researchers have found differing approaches useful for qualitatively different regimes of fluid structure interaction. Applications may involve low or high structural deformation. Problems involving low deformation can be approached with aligning fluid meshes using for instance Arbitrary Lagrangian-Eulerian (ALE) schemes. This has been quite successful for studying phenomena such as airplane wing flutter, see e.g. [10]. High structural deformation is sometimes also approached using ALE techniques, but frequent remeshing and the averaging it requires can lead to reduced accuracy and increased computational cost. More often, methods where the fluid and structure meshes are not aligned are employed (e.g. the body of work originating from [23]). In this case the interaction between fluid and structure is usually massaged via penalty methods or Lagrange multiplier formulations.

Anecdotal evidence suggests that problems where either the fluid or the structure dominates the interaction are inherently more stable than those where the effect is more balanced. For example, it is more straightforward to solve problems where a kinematic structure drives the fluid or a structure is passively advected by fluid, as opposed to problems where structures and fluids have nearly equal densities (see e.g. [7]). Partitioned coupling approaches are typically explicit and staggered, for instance first imposing the fluid pressures as a force on the structure and then solving for the fluid using the structure velocities as boundary conditions in a second step. These approaches can also be iterated within a given time step for increased stability, noting that one obtains a monolithic (albeit expensive) approach if the iteration is run to convergence. Other approaches construct strongly coupled systems and then solve them in one of several ways. We follow a monolithically coupled approach, in which we model the fluid structure interaction by formulating and solving a single coupled system of equations; however, we do this in a way which is more

*{avir,cas43,fedkiw}@cs.stanford.edu, Stanford University

efficient as well as more likely to converge than simply iterating the partitioned approach to convergence in a simple Gauss-Seidel manner.

Our approach allows the use of existing discretizations for both the fluid and the structure and requires only minor modification to existing codes. We do require that the fluid and structure solvers have certain properties, though we will argue that they ought to be true of any physically constituted system. The current formulation assumes that the fluid and structure systems are synchronized in time such that we can solve for the coupling at the synchronization points, although it is not inconceivable that one could apply asynchronous subiteration to each system.

The method that we propose in this paper is similar to the previous work in [25] addressing an implicit method for two-way coupling of incompressible fluids and structures, with the chief difference being that the linear system presented in [25] was indefinite. Our main contribution is the observation in Section 5.2 that the structure damping operator can be factored and used to perform a change of variables that allows us to symmetrize the coupled system while maintaining positive definiteness. We examine the properties of the resulting linear system and demonstrate convergence of the method on several example problems from the literature.

2. Fluid Equations

We address the Navier-Stokes equations for incompressible flow,

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \quad (1)$$

and

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

in the presence of structures. Here, \mathbf{u} is the fluid velocity, p is the pressure, μ is the dynamic viscosity, ρ is the density, and \mathbf{f} is any external body force such as gravity.

We use a standard Marker and Cell (MAC, [16]) grid discretization for the fluid. When solving in the presence of structures, some faces may not contain valid velocity samples because they are covered by structures. The vector of fluid velocity samples omitting any covered by the structure is denoted by \mathbf{u} . Pressure samples are defined at cell centers. The divergence operator is defined to act on velocities at cell faces to yield divergences at cell centers using the standard six-point stencil in three spatial dimensions, and the gradient operator is defined as the negative transpose of the divergence.

We apply the projection method of [8] to solve the discretized form of Eqs. (1) and (2), making a first order approximation in time. This splits the time integration into the computation of an intermediate velocity ignoring the pressure terms

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \frac{\Delta t \mu}{\rho} \nabla^2 \mathbf{u} + \frac{\Delta t}{\rho} \mathbf{f} \quad (3)$$

followed by enforcement of incompressibility via an implicit pressure projection

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p. \quad (4)$$

We have omitted a specification of time from the viscosity term, since the viscosity might be applied explicitly or implicitly. We postpone further consideration of viscosity until Section 7, at which point we will include it in the coupling framework. We use a semi-Lagrangian advection scheme (see e.g. [26]) to compute the convective terms in Eq. (3). In the presence of structures, this requires ghost cells or clipping since the back-cast rays may intersect the solid; we use a method based on [14] for semi-Lagrangian advection near

solid surfaces. We advect using the divergence free time n velocities. We enforce incompressibility as usual by plugging Eq. (4) into the discrete form of Eq. (2) to obtain

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p \right) = \nabla \cdot \mathbf{u}^*,$$

which we solve for p using the conjugate gradient method.

3. Structure Equations

We utilize a traditional Lagrangian approach for structures. For deformable structures, we solve the equations $\dot{\mathbf{x}} = \mathbf{v}$ and $\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}$. We take a lumped mass approach where the quantities \mathbf{x} and \mathbf{v} are the vectors containing the position and velocity degrees of freedom for all n of the particles. The mass matrix \mathbf{M} is a $3n \times 3n$ diagonal matrix with each 3×3 diagonal block of the form $m_i \mathbf{I}$, where m_i is the mass of particle i . The forces $\mathbf{F} = \mathbf{F}_e + \mathbf{F}_d$ contain both elastic \mathbf{F}_e (velocity independent) and damping \mathbf{F}_d (velocity dependent) forces. Purely elastic forces (e.g., no plasticity) can be expressed in terms of an energy potential as $\mathbf{F}_e = -\frac{\partial \phi}{\partial \mathbf{x}}$, where the energy potential ϕ depends only on \mathbf{x} . Damping forces are often chosen according to a simple model $\mathbf{F}_d = \mathbf{D}\mathbf{v}$ that is linear in velocities and has symmetric negative semidefinite \mathbf{D} . This is true, for example, for the damping applied to a simple spring and Rayleigh damping.

One particularly simple and stable way of evolving these equations is via backward Euler. We wish to solve

$$\begin{pmatrix} \mathbf{x}^{n+1} \\ \mathbf{v}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^n + \Delta t \mathbf{v}^{n+1} \\ \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}^{n+1} \end{pmatrix}.$$

Taking the first order approximation $\mathbf{F}^{n+1} = \mathbf{F}^n + \frac{\partial \mathbf{F}^n}{\partial \mathbf{x}} (\mathbf{x}^{n+1} - \mathbf{x}^n) + \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} (\mathbf{v}^{n+1} - \mathbf{v}^n)$ this becomes

$$\begin{pmatrix} \mathbf{x}^{n+1} \\ \mathbf{v}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^n + \Delta t \mathbf{v}^{n+1} \\ \mathbf{v}^n + \Delta t \mathbf{M}^{-1} (\mathbf{F}^n + \frac{\partial \mathbf{F}^n}{\partial \mathbf{x}} (\mathbf{x}^{n+1} - \mathbf{x}^n) + \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} (\mathbf{v}^{n+1} - \mathbf{v}^n)) \end{pmatrix}.$$

Using the first equation to eliminate \mathbf{x}^{n+1} in the second equation yields

$$\begin{aligned} \mathbf{v}^{n+1} &= \mathbf{v}^n + \Delta t \mathbf{M}^{-1} (\mathbf{F}^n + \frac{\partial \mathbf{F}^n}{\partial \mathbf{x}} \Delta t \mathbf{v}^{n+1} + \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} (\mathbf{v}^{n+1} - \mathbf{v}^n)) \\ \left(\mathbf{I} - \Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} - \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{x}} \right) \mathbf{v}^{n+1} &= \mathbf{v}^n - \Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}^n \end{aligned} \quad (5)$$

From this we see that we must invert $\mathbf{I} - \Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} - \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{x}}$. The second term, $-\Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} = -\Delta t \mathbf{M}^{-1} \mathbf{D}$, is symmetric positive semidefinite under a mass weighted inner product, which can be used for solving the equation. The third term, however, can cause problems. Note that $\frac{\partial \mathbf{F}^n}{\partial \mathbf{x}} = \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}_d}{\partial \mathbf{x}}$. The first term is symmetric since $\mathbf{F}_e = -\frac{\partial \phi}{\partial \mathbf{x}}$, but it might be indefinite. The second term need not even be symmetric under the mass weighted inner product.

Another approach is a Newmark-style semi-implicit scheme

- $\mathbf{v}_*^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{F}(\mathbf{x}^n, \mathbf{v}_*^{n+1/2})$
- $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}_*^{n+1/2}$
- $\mathbf{v}^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{F}(\mathbf{x}^n, \mathbf{v}^n)$
- $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + \frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{F}(\mathbf{x}^{n+1}, \mathbf{v}^{n+1})$

Each time step of this scheme involves two implicit solves, bulleted steps 1 and 4. These solves differ from the fully implicit backward Euler solve in that the dependence of force on position is explicit. The first solve, if taken for a full Δt , would look like

$$\begin{aligned} \mathbf{v}^{n+1} &= \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}(\mathbf{x}^n, \mathbf{v}^{n+1}) \\ \mathbf{v}^{n+1} &= \mathbf{v}^n + \Delta t \mathbf{M}^{-1} (\mathbf{F}^n + \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} (\mathbf{v}^{n+1} - \mathbf{v}^n)) \\ \left(\mathbf{I} - \Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} \right) \mathbf{v}^{n+1} &= \mathbf{v}^n - \Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{v}} \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}^n \end{aligned} \quad (6)$$

$$\left(\mathbf{I} - \Delta t \mathbf{M}^{-1} \mathbf{D} \right) \mathbf{v}^{n+1} = \mathbf{v}^n - \Delta t \mathbf{M}^{-1} \mathbf{D} \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}^n \quad (7)$$

We see from this that the fully implicit system Eq. (5) differs from the semi-implicit system Eq. (6) by the presence of the extra $-\Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{F}^n}{\partial \mathbf{x}}$ term. This term was the source of the indefiniteness and asymmetry problems in the fully coupled system. This simplification comes at the price of stability, however. The fully implicit backward Euler scheme is unconditionally stable, but the semi-implicit scheme is not. Of course, this is the well-known difference between a fully implicit Newmark method and one that resembles central differencing [18].

The particular scheme used for the simulations in this paper is based on the backward Euler variant of [28], which differs from the previous scheme in that it has been rewritten so that a backward Euler step can be used for updating the final positions. That is

1. $\hat{\mathbf{v}}^* = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{F}_e(\mathbf{x}^n)$
2. $\mathbf{v}^{n+1/2} = \hat{\mathbf{v}}^* + \frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{D}(\mathbf{x}^n) \mathbf{v}^{n+1/2}$
3. $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1/2}$
4. $\mathbf{v}^* = \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}_e(\mathbf{x}^{n+1})$
5. $\mathbf{v}^{n+1} = \mathbf{v}^* + \Delta t \mathbf{M}^{-1} \mathbf{D}(\mathbf{x}^{n+1}) \mathbf{v}^{n+1}$

As before, both solves lead to systems of the form Eq. (7). This scheme is first order accurate in time, limited to first order by the backward Euler step used to compute the final velocities.

Our treatment of rigid bodies follows that of [28]. In general, each particle of a deformable body contributes one block to \mathbf{v} , \mathbf{M} , and \mathbf{F} . Each rigid body contributes two blocks to each, with the first block corresponding to the linear part, \mathbf{v}_i , m_i , and \mathbf{f}_i , and the second block corresponding to the angular part, $\boldsymbol{\omega}_i$, \mathbf{I}_i , and $\boldsymbol{\tau}_i$. With this taken into account, the linear system Eq. (7) is valid for deformable and rigid bodies.

The ability to treat rigid bodies using the same formulas as deformable bodies breaks down somewhat when dealing with advancing rigid body orientation. Fortunately, the coupling formulation presented here is explicit in position and therefore does not directly involve the integration of rigid body orientation, so this complication is avoided. The coupling formulation is valid when bodies are deformable, rigid, or a mixture of both.

4. Discrete Fluid Structure Coupling Equations

Our goal is to write down the equations that need to be solved in full, including the coupling and any other terms such as boundary conditions. In doing this it is more useful to work from the actual discretizations than it is from the continuous equations. Therefore we will consider the MAC grid fluid discretizations and the Lagrangian structure discretizations as defined in Sections 2 and 3 and use them going forward for our notation. However, we stress that one could use other discretizations for the fluids and structures and apply the same methodology that we propose to those sets of equations.

We follow a method similar to that of [25] to merge the time integration schemes described in Sections 2 and 3,

1. $\hat{\mathbf{v}}^* = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{M}^{-1} \mathbf{F}_e(\mathbf{x}^n)$
2. $\hat{\mathbf{u}}^* = \mathbf{u}^n + \frac{\Delta t}{2\rho} \mathbf{f}$

3. Coupled solve with known quantities $\hat{\mathbf{u}}^*$, $\hat{\mathbf{v}}^*$, and \mathbf{x}^n and unknown quantities $\mathbf{u}^{n+1/2}$, \mathbf{p} , and $\mathbf{v}^{n+1/2}$
4. $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1/2}$
5. $\mathbf{v}^* = \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}_e(\mathbf{x}^{n+1})$
6. $\mathbf{u}^* = \mathbf{u}^n - \Delta t (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \frac{\Delta t}{\rho} \mathbf{f}$
7. Coupled solve with known quantities \mathbf{u}^* , \mathbf{v}^* , and \mathbf{x}^{n+1} and unknown quantities \mathbf{u}^{n+1} , \mathbf{p} , and \mathbf{v}^{n+1}

Steps 1 through 2 integrate the explicit structure forces and explicit fluid forces for $\Delta t/2$. We leave out the convective terms in the fluid update because the coupled solve in step 3 takes positions at time n and solves for velocities at time $n + \frac{1}{2}$. The known quantities at the beginning of step 3, which replaces step 2 in Section 3, are $\hat{\mathbf{u}}^*$, $\hat{\mathbf{v}}^*$, and \mathbf{x}^n . The remaining unknown quantities to solve for in Eqs. (2), (4) and (7) are $\mathbf{u}^{n+1/2}$, \mathbf{p} , and $\mathbf{v}^{n+1/2}$. The details of the coupled solve are described later, but note that this solve takes place over only half of the time step (and thus has step size $\frac{\Delta t}{2}$). In step 4, $\mathbf{v}^{n+1/2}$ is used to advance the structure positions to time $n + 1$, and then $\mathbf{u}^{n+1/2}$, \mathbf{p} and $\mathbf{v}^{n+1/2}$ are discarded.

Steps 5 through 6 re-integrate the explicit structure forces and explicit fluid forces for Δt , this time including the convective terms in the fluid update to get time $n + 1$ fluid positions, resulting in \mathbf{v}^* and \mathbf{u}^* . The known quantities at the beginning of step 7, which replaces step 5 from Section 3 and Eq. (4), are \mathbf{u}^* , \mathbf{v}^* , and \mathbf{x}^{n+1} . We use \mathbf{x}^{n+1} to determine the valid degrees of freedom for \mathbf{u}^{n+1} . The remaining unknown quantities to solve for in Eqs. (2), (4) and (7) are \mathbf{u}^{n+1} , \mathbf{p} , and \mathbf{v}^{n+1} . The coupled solve used in this step has the same form as that in step 3.

For the sake of exposition, we consider below only the second of the two implicit steps described above (step 7), and therefore we will discuss the updates from \mathbf{u}^* and \mathbf{v}^* to \mathbf{u}^{n+1} and \mathbf{v}^{n+1} , whereas the updates from $\hat{\mathbf{u}}^*$ and $\hat{\mathbf{v}}^*$ to $\mathbf{u}^{n+1/2}$ and $\mathbf{v}^{n+1/2}$ will be handled independently. One can replace Δt , \mathbf{x}^{n+1} , \mathbf{u}^* , \mathbf{v}^* , \mathbf{u}^{n+1} and \mathbf{v}^{n+1} with $\frac{\Delta t}{2}$, \mathbf{x}^n , $\hat{\mathbf{u}}^*$, $\hat{\mathbf{v}}^*$, $\mathbf{u}^{n+1/2}$ and $\mathbf{v}^{n+1/2}$ in the discussion that follows in order to apply the same treatment to step 3.

4.1. Fluid Structure Coupling Constraints

The modifications to the fluid and structure equations correspond to constraints that capture the behavior at the fluid structure interface. A fluid structure coupling constraint is an enforced constraint on the relative velocity of the fluid and the structure at a point in space. In our implementation, a constraint occurs when a ray cast between two adjacent MAC grid cell centers, at least one of which is in the fluid region, intersects a structure surface. The constraint is placed at the center of the MAC grid face between the two cell centers, and enforces that the projection of the fluid and structure velocities at that point in the ray direction are equal. The impulse transfer between the solid and the fluid which enforces the constraint is called λ . See Figure 1. This simple choice of constraint location is only first order accurate, since the constraint locations are located $O(\Delta x)$ from the structure surface.

4.2. Modified Fluid Equations

For fluid structure interaction, Eq. (4) changes due to the interaction with the structure. In Figure 1 for the x direction we have a mass in that dual cell m and a velocity u , the product of which mu is equal to mu^* plus the contribution from p in the fluid on the left side of the dual cell, plus a contribution from λ due to interaction with the structure. This discards the pressure on the right hand side of the dual cell due to the structure. We can neglect the dual cells that would lose two pressure contributions, those inside the structure, since they are not degrees of freedom. We write this in vector form for all of the fluid velocity terms as

$$\beta \mathbf{u}^{n+1} = \beta \mathbf{u}^* - \hat{\mathbf{G}} \hat{\mathbf{p}} + \mathbf{W}^T \lambda, \quad (8)$$

where β is the diagonal matrix of fluid dual cell masses assembled from the product of the density ρ and the volume V of each dual cell. We let $\hat{\mathbf{p}} = \Delta t \mathbf{p}$. We define $-\hat{\mathbf{G}}^T = -V \nabla^T$, the volume weighted divergence, with the additional modification that we drop rows from $-\hat{\mathbf{G}}^T$ that correspond to cells without fluid. Its negative transpose, $\hat{\mathbf{G}}$, is defined as the volume weighted gradient. $-\hat{\mathbf{G}}^T$ is simpler to describe directly than

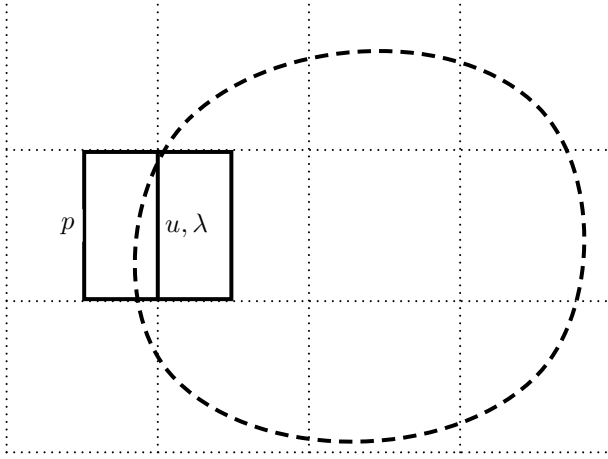


Figure 1: A dual cell requiring an equal velocity constraint is depicted above intersecting the solid depicted by the dashed line. The fluid pressure on the left face of the dual cell will be balanced by the constraint-maintaining impulse λ applied at the equal velocity constraint location.

$\hat{\mathbf{G}}$, since every row of $-\hat{\mathbf{G}}^T$ has the same stencil while the stencils for rows of $\hat{\mathbf{G}}$ differ because some pressure samples fall inside the structure. \mathbf{W}^T is the matrix of 1s and 0s which maps $\boldsymbol{\lambda}$ to the appropriate fluid velocity scalars, which will be discussed in more detail below. The discrete form of the incompressibility constraint is

$$-\hat{\mathbf{G}}^T \mathbf{u}^{n+1} = \mathbf{0} \quad (9)$$

Our discretizations of \mathbf{W} , $\hat{\mathbf{G}}$, and $-\hat{\mathbf{G}}^T$ are first order accurate since the constraint locations used in their definition are only first order accurate. We note that even were $\hat{\mathbf{G}}$ a higher order accurate approximation of the gradient operator it would not necessarily imply that $-\hat{\mathbf{G}}^T$ would be a compatible and higher order accurate approximation of the divergence. Unlike the second order discretizations [11, 22], it is not sufficient in this context to discretize the Poisson operator with higher order accuracy. The discretizations of gradient and divergence in [22] do not satisfy the negative transpose relationship, so using them in this formulation would produce an asymmetric linear system.

4.3. Modified Structure Equations

To write the modified equations for the structure, one needs to consider the fact that the fluid and structure degrees of freedom are defined at different spatial locations. This is quite important as constraints must be applied to co-located fluid and structure velocities, and the equal and opposite forces applied to the fluid and the structure must be applied at the same location in space so as not to introduce a net torque. We define an operator \mathbf{J} which interpolates from structure degrees of freedom to fluid velocity locations. Its transpose \mathbf{J}^T is a conservative redistribution from fluid velocity locations to structure velocity degrees of freedom.

\mathbf{J} can be decomposed into three matrices, $\mathbf{J} = \mathbf{N}\hat{\mathbf{J}}\mathbf{R}$. The matrix \mathbf{R} is comprised of blocks \mathbf{R}_{pb} , where body b is sampled by particle p . For particles within a deforming body, the particles are the samples, and $\mathbf{R}_{pb} = \mathbf{I}$. If b corresponds to a rigid body, then there is a sample point p at each nearby fluid velocity location. If such a sample point p is displaced from the rigid body's center of mass by \mathbf{r}_{pb} , the velocity at the sample point is

$$\mathbf{v}_p = \mathbf{v}_b + \mathbf{r}_{pb}^{*T} \boldsymbol{\omega}_b = (\mathbf{I} \quad \mathbf{r}_{pb}^{*T}) \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix}_b = \mathbf{R}_{pb} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix}_b \quad (10)$$

so that $\mathbf{R}_{pb} = (\mathbf{I} \quad \mathbf{r}_{pb}^{*T})$, where \mathbf{r}^* represents the matrix such that $\mathbf{r}^* \mathbf{u} = \mathbf{r} \times \mathbf{u}$ for any \mathbf{u} .

The operator $\hat{\mathbf{J}}$ interpolates particle velocity samples p to a MAC grid face f as $\sum_p \alpha_{fp} \mathbf{v}_p$, so that the blocks of $\hat{\mathbf{J}}$ are $\hat{\mathbf{J}}_{fp} = \alpha_{fp} \mathbf{I}$. Since $\hat{\mathbf{J}}$ is an interpolation operator, $\sum_p \alpha_{fp} = 1$. The coefficients α_{fp} are defined based on local geometric information associated with the face f and the particle p such as the control volume around f and structure faces incident on p . For the rigid body, \mathbf{R} is used to sample the rigid body velocities at sample points, and the most logical positions for sample points are the fluid velocity locations at the centers of faces. In that sense, if only one rigid body was present near a face, locally the $\hat{\mathbf{J}}$ operator is the identity for that degree of freedom. However multiple rigid bodies or particles may influence the same face, in which case an averaging of the velocities would be required, making $\hat{\mathbf{J}}$ nontrivial.

Finally, \mathbf{N} extracts the appropriate scalar velocity component from the vector velocity constructed via \mathbf{R} and $\hat{\mathbf{J}}$. The matrix \mathbf{N} is block diagonal, where each block \mathbf{N}_{ff} corresponds to a grid face f with normal \mathbf{n}_f . These blocks are formed as $\mathbf{N}_{ff} = \mathbf{n}_f^T$. Our discretization of the operator \mathbf{J} is limited to first order accuracy since it depends on the first order accurate structure interface location.

Since Eq. (8) applies an impulse to the fluid at MAC grid face locations, the equal and opposite impulse must be applied to the structure at those same locations, guaranteeing conservation of angular momentum. For the case of rigid bodies, one can apply impulses to the rigid body from any location in space and therefore the method does conserve angular momentum. However, in the case of deformable bodies $\hat{\mathbf{J}}^T$ contains a spatial redistribution of the impulse which, while being a fully conservative remapping with columns summing to 1 and thus conserving linear momentum, will not conserve angular momentum. In summary, \mathbf{N}^T maps impulse magnitudes from dual cells to impulse vectors normal to the face. Since $\sum_p \alpha_{fp} = 1$, $\hat{\mathbf{J}}^T$ conservatively distributes these impulses to different spatial sample locations in the case of deformable bodies. For rigid bodies the spatial location does not change although the impulse can still be conservatively distributed among multiple rigid bodies influencing that face. Finally \mathbf{R}^T conservatively applies those sample impulses to the structure degrees of freedom. This mapping is trivial for deformable particles, but for rigid bodies it converts the impulse sample into linear and angular impulses at the body's center of mass. We note that [2] proposed a method which allows the solid and fluid domains to be non-coincident, coupled via interpolation and Lagrange multipliers, similar to our use of \mathbf{W} and \mathbf{J} .

The full modified structure equations are

$$\mathbf{M}\mathbf{v}^{n+1} = \mathbf{M}\mathbf{v}^* + \Delta t \mathbf{D}\mathbf{v}^{n+1} - \mathbf{J}^T \mathbf{W}^T \boldsymbol{\lambda} \quad (11)$$

where \mathbf{M} is the structure mass matrix and $\mathbf{D}\mathbf{v}^{n+1}$ is the implicit contribution of structure forces. The last term is the constraint impulse mapped first by \mathbf{W}^T from constraint locations to fluid velocity degrees of freedom and then by \mathbf{J}^T to structure velocity degrees of freedom. The coupling momentum $\mathbf{W}^T \boldsymbol{\lambda}$ is subtracted from the structure equation since it was added to the fluids equation. \mathbf{W} has a number of columns equal to the number of fluid scalar velocity samples and a number of rows equal to the number of constraints. \mathbf{W}^T therefore maps to every MAC grid face but puts zero on every face which is not coupled. As a consequence, \mathbf{J}^T only needs to be nonzero for the faces which are coupled, even though it has a number of columns equal to the number of fluid scalar velocity samples. $\mathbf{J}^T \mathbf{W}^T$ then maps from the fluid degrees of freedom that are coupled to the structure degrees of freedom that are coupled.

Although Eqs. (8) and (11) indicate how momentum is conservatively transferred between the fluid and the structure, this still leaves open the question of how much momentum is transferred. That is, we have introduced a new degree of freedom $\boldsymbol{\lambda}$ and have not yet provided the equations to govern these new degrees of freedom. In order to do this we consider the equal velocity constraints that the fluid and the structure should move with the same velocity, or for inviscid flow the same velocity in the normal direction. One way of governing these degrees of freedom is to set the difference between the fluid velocity and the interpolated structure velocity equal to zero at every fluid velocity degree of freedom as indicated in Figure 1. This can be expressed as the equation

$$\mathbf{W}(\mathbf{u} - \mathbf{J}\mathbf{v}) = 0, \quad (12)$$

noting that \mathbf{J} maps zero to all the face locations that are not influenced by the structure and that \mathbf{W} filters all of these locations out. Note that one can use Eq. (12) and its variants in order to model a wide range

of physical phenomena. For example, one could use $\mathbf{W}(\mathbf{u} - \mathbf{J}(\mathbf{v} - \mathbf{s})) = \mathbf{0}$ in order to model an arbitrary source term with inflow and outflow from the structure. One might also wish to constrain the velocities at other locations than the faces of the MAC grid or to constraint limited components such as the normal component as was done in [24] in order to achieve better slip boundary conditions. In this sense \mathbf{W} may not be simply defined as a trivial restriction matrix and may instead be thought of as a general interpolation operator mapping from the faces of the MAC grid to constraint locations. This would not change the inputs of \mathbf{W} , but the outputs would no longer necessarily correspond to the MAC grid faces but could instead be at arbitrary locations. Note that the formulation of our equations does not rule out any of these approaches, but for the sake of exposition we will deal with Eq. (12) going forward.

5. Numerical Approach

We can write our system in symmetric form as

$$\begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \beta^{-1} \hat{\mathbf{G}} & \mathbf{W} \beta^{-1} \mathbf{W}^T & -\mathbf{W} \mathbf{J} \\ \mathbf{0} & -\mathbf{J}^T \mathbf{W}^T & -\mathbf{M} + \Delta t \mathbf{D} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \boldsymbol{\lambda} \\ \mathbf{v}^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ -\mathbf{W} \mathbf{u}^* \\ -\mathbf{M} \mathbf{v}^* \end{pmatrix}.$$

The first equation,

$$\hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} - \hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T \boldsymbol{\lambda} = \hat{\mathbf{G}}^T \mathbf{u}^* \quad (13)$$

is obtained by substituting Eq. (8) into Eq. (9). The second equation,

$$-\mathbf{W} \beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} + \mathbf{W} \beta^{-1} \mathbf{W}^T \boldsymbol{\lambda} - \mathbf{W} \mathbf{J} \mathbf{v}^{n+1} = -\mathbf{W} \mathbf{u}^* \quad (14)$$

is obtained by substituting Eq. (8) into Eq. (12). The third equation,

$$-\mathbf{J}^T \mathbf{W}^T \boldsymbol{\lambda} - \mathbf{M} \mathbf{v}^{n+1} + \Delta t \mathbf{D} \mathbf{v}^{n+1} = -\mathbf{M} \mathbf{v}^* \quad (15)$$

is a rearrangement of Eq. (11).

If we make some assumptions about \mathbf{W} and \mathbf{J} , we can eliminate $\boldsymbol{\lambda}$. If \mathbf{W} takes on the particularly simple form described in Sections 4.2 and 4.3, then $\mathbf{W} \mathbf{W}^T = \mathbf{I}$. If \mathbf{J} has nonzero rows only for constraint faces, then $\mathbf{W}^T \mathbf{W} \mathbf{J} = \mathbf{J}$. If we also take densities to be constant at all constraint faces, then $\mathbf{W}^T \mathbf{W} \beta^{-1} = \beta^{-1} \mathbf{W}^T \mathbf{W}$. Multiplying Eq. (14) by \mathbf{W}^T produces

$$\begin{aligned} -\mathbf{W}^T \mathbf{W} \beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} + \mathbf{W}^T \mathbf{W} \beta^{-1} \mathbf{W}^T \boldsymbol{\lambda} - \mathbf{W}^T \mathbf{W} \mathbf{J} \mathbf{v}^{n+1} &= \mathbf{W}^T \mathbf{W} \mathbf{u}^* \\ -\mathbf{W}^T \mathbf{W} \beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} + \beta^{-1} \mathbf{W}^T \boldsymbol{\lambda} - \mathbf{J} \mathbf{v}^{n+1} &= \mathbf{W}^T \mathbf{W} \mathbf{u}^* \end{aligned} \quad (16)$$

Adding $\hat{\mathbf{G}}^T$ times Eq. (16) to Eq. (13) yields

$$\begin{aligned} (\hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} - \hat{\mathbf{G}}^T \mathbf{W}^T \mathbf{W} \beta^{-1} \hat{\mathbf{G}}) \hat{\mathbf{p}} + (-\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T + \hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T) \boldsymbol{\lambda} - \hat{\mathbf{G}}^T \mathbf{J} \mathbf{v}^{n+1} &= \hat{\mathbf{G}}^T \mathbf{u}^* - \hat{\mathbf{G}}^T \mathbf{W}^T \mathbf{W} \mathbf{u}^* \\ \hat{\mathbf{G}}^T (\mathbf{I} - \mathbf{W}^T \mathbf{W}) \beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} - \hat{\mathbf{G}}^T \mathbf{J} \mathbf{v}^{n+1} &= \hat{\mathbf{G}}^T (\mathbf{I} - \mathbf{W}^T \mathbf{W}) \mathbf{u}^* \end{aligned}$$

Adding $\mathbf{J}^T \beta$ times Eq. (16) to Eq. (15) yields

$$\begin{aligned} -\mathbf{J}^T \beta \mathbf{W}^T \mathbf{W} \beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} + (-\mathbf{J}^T \mathbf{W}^T + \mathbf{J}^T \mathbf{W}^T) \boldsymbol{\lambda} + (\Delta t \mathbf{D} - \mathbf{M} - \mathbf{J}^T \beta \mathbf{J}) \mathbf{v}^{n+1} &= -\mathbf{M} \mathbf{v}^* + \mathbf{J}^T \beta \mathbf{W}^T \mathbf{W} \mathbf{u}^* \\ -\mathbf{J}^T \hat{\mathbf{G}} \hat{\mathbf{p}} + (\Delta t \mathbf{D} - \mathbf{M} - \mathbf{J}^T \beta \mathbf{J}) \mathbf{v}^{n+1} &= -\mathbf{M} \mathbf{v}^* + \mathbf{J}^T \beta \mathbf{u}^* \end{aligned}$$

These together form a symmetric system similar to that of [25],

$$\begin{pmatrix} \hat{\mathbf{G}}^T (\mathbf{I} - \mathbf{W}^T \mathbf{W}) \beta^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \mathbf{J} \\ -\mathbf{J}^T \hat{\mathbf{G}} & -\mathbf{J}^T \beta \mathbf{J} - \mathbf{M} + \Delta t \mathbf{D} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \mathbf{v}^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{G}}^T (\mathbf{I} - \mathbf{W}^T \mathbf{W}) \mathbf{u}^* \\ -\mathbf{M} \mathbf{v}^* + \mathbf{J}^T \beta \mathbf{u}^* \end{pmatrix}.$$

Both of these systems are symmetric indefinite. Although numerical methods exist for solving such systems, faster methods are available for systems which are symmetric positive definite. In this paper, we propose a novel approach which leads to a symmetric positive definite system.

We begin by forming the system

$$\begin{pmatrix} \hat{\mathbf{G}}^T \boldsymbol{\beta}^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \boldsymbol{\beta}^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \boldsymbol{\beta}^{-1} \hat{\mathbf{G}} & \mathbf{W} \boldsymbol{\beta}^{-1} \mathbf{W}^T + \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & -\Delta t \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{D} \\ \mathbf{0} & \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{I} - \Delta t \mathbf{M}^{-1} \mathbf{D} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \boldsymbol{\lambda} \\ \mathbf{v}^{n+1} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ \mathbf{W} \mathbf{J} \mathbf{v}^* - \mathbf{W} \mathbf{u}^* \\ \mathbf{v}^* \end{pmatrix} \quad (17)$$

where the second equation, $-\mathbf{W} \boldsymbol{\beta}^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} + \mathbf{W} \boldsymbol{\beta}^{-1} \mathbf{W}^T \boldsymbol{\lambda} + \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T \boldsymbol{\lambda} - \Delta t \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{D} \mathbf{v}^{n+1} = \mathbf{W} \mathbf{J} \mathbf{v}^* - \mathbf{W} \mathbf{u}^*$, is obtained by substituting Eq. (8) and Eq. (11) into Eq. (12). The first equation is Eq. (13), and the third equation is Eq. (15) multiplied through by \mathbf{M}^{-1} .

An obvious way to symmetrize this system is to multiply the third equation through by $-\Delta t \mathbf{D}$. Though this system is symmetric it is only positive semidefinite. The conditioning of the original structure system is typically quite good, since $\mathbf{I} - \Delta t \mathbf{M}^{-1} \mathbf{D}$ has eigenvalues clustered near one for small Δt . This symmetrized version looks more like \mathbf{D} , which will often have significantly worse conditioning.

A particularly interesting way of symmetrizing the system is to factor \mathbf{D} .

5.1. Factoring Damping Matrix

If \mathbf{D} is symmetric negative semidefinite, then it can be factored as $-\Delta t \mathbf{D} = \mathbf{C}^T \mathbf{C}$. We rely on this property of \mathbf{D} to apply our method, noting that although there are instances where \mathbf{D} will not have this property and in those cases our approach based on factoring \mathbf{D} is not applicable, there are a wide range of models in which this property holds (see below). If \mathbf{D} is a damping matrix, and the force is given by $\mathbf{f} = \mathbf{M} \mathbf{a} = -\nabla \phi + \mathbf{D} \mathbf{v}$, then the energy is $E = \phi + \frac{1}{2} \mathbf{v}^T \mathbf{M} \mathbf{v}$ and $\dot{E} = (\nabla \phi)^T \mathbf{v} + \mathbf{v}^T \mathbf{M} \mathbf{a} = (\nabla \phi)^T \mathbf{v} + \mathbf{v}^T (-\nabla \phi + \mathbf{D} \mathbf{v}) = \mathbf{v}^T \mathbf{D} \mathbf{v}$. Then, a non-positive change in energy leads to negative definiteness of \mathbf{D} . Furthermore, if \mathbf{D} is decomposed into its symmetric and antisymmetric parts, $\frac{\mathbf{D} + \mathbf{D}^T}{2}$ and $\frac{\mathbf{D} - \mathbf{D}^T}{2}$ respectively, then $\mathbf{v}^T \left(\frac{\mathbf{D} - \mathbf{D}^T}{2} \right) \mathbf{v} = \mathbf{0}$ shows that the antisymmetric part does not contribute damping. Therefore, damping matrices gain little by containing asymmetric parts. These facts lead to the conclusion that for most purposes a damping matrix \mathbf{D} can be considered to be symmetric negative semidefinite.

On the other hand, if you consider the backward Euler formulation at the beginning of Section 3, where \mathbf{D} would be augmented by linearizations of elastic terms $\frac{\partial \mathbf{F}^n}{\partial \mathbf{x}}$ in the process of carrying out the Newton-Raphson iterations, then much of this would not be true. The current factorization is therefore limited to a semi-implicit treatment of the structure, where elastic terms are handled explicitly. Moreover, a fully implicit treatment would have the further complication of moving the locations of the constraints and changing \mathbf{W} during the solve, which we do not address.

For the sake of exposition, consider the case of a linear spring which connects two particles \mathbf{p}_1 and \mathbf{p}_2 . The magnitude of the damping force \mathbf{f} is proportional to the difference in particle velocities along the spring direction \mathbf{d} , multiplied by a damping constant k_d . The force is then applied in opposite directions to \mathbf{p}_1 and \mathbf{p}_2 . The damping matrix for this single spring would be

$$\mathbf{D} = -\Delta t k_d \begin{pmatrix} \mathbf{d} \\ -\mathbf{d} \end{pmatrix} \begin{pmatrix} \mathbf{d} \\ -\mathbf{d} \end{pmatrix}^T,$$

which can be factored into

$$\begin{aligned} \mathbf{C}_i &= \sqrt{\Delta t k_d} \begin{pmatrix} \mathbf{d} \\ -\mathbf{d} \end{pmatrix}^T \\ \mathbf{D} &= -\mathbf{C}_i^T \mathbf{C}_i. \end{aligned}$$

For the case of n springs and an arbitrary number of particles, \mathbf{C} is formed by stacking up rows of the form \mathbf{C}_i and the same factorization applies.

Rayleigh damping is commonly used along with finite element constitutive models. In practice, we have found that our routines for applying damping forces are naturally partially factored. The implementation follows the framework of [20]. Below is a description of what the implementation looks like in the case of a single element and an isotropic constitutive model and how it lends itself to efficient factorization. In this framework, the deformation gradient is factored into its polar singular value decomposition as $\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^T$, where \mathbf{U} and \mathbf{V} are rotations and $\hat{\mathbf{F}}$ is diagonal, in order to compute the elastic forces. Define two operators

$$G \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = (\mathbf{x}_1 - \mathbf{x}_4 \quad \mathbf{x}_2 - \mathbf{x}_4 \quad \mathbf{x}_3 - \mathbf{x}_4) \quad S \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ -\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3 \end{pmatrix}.$$

If \mathbf{x}_m are the material space positions of the element, then $\mathbf{D}_m = G(\mathbf{x}_m)$ and $\mathbf{D}_s = G(\mathbf{x})$ are the locations of three vertices of the element relative to the fourth in material and world space. The damping force application routine consists of the steps

1. Compute the relative velocities, $\hat{\mathbf{D}}_s = G(\mathbf{v})$.
2. Compute the deformation gradient, $\hat{\mathbf{F}} = \hat{\mathbf{D}}_s \mathbf{D}_m^{-1}$.
3. Rotate the deformation gradient into the diagonalized coordinate system, $\hat{\hat{\mathbf{F}}} = \mathbf{U}^T \hat{\mathbf{F}} \mathbf{V}$.
4. Compute the first Piola-Kirchhoff stress in the diagonalized coordinate system, $\hat{\mathbf{P}} = \alpha \text{tr}(\hat{\hat{\mathbf{F}}}) \mathbf{I} + \beta \hat{\hat{\mathbf{F}}} + \beta \hat{\hat{\mathbf{F}}}^T$, where α and β are parameters of the constitutive model.
5. Rotate back to compute the first Piola-Kirchhoff stress, $\mathbf{P} = \mathbf{U} \hat{\mathbf{P}} \mathbf{V}^T$.
6. Compute forces the element applies on three of the tetrahedron faces as $\hat{\mathbf{G}} = -\mathbf{P} \mathbf{N}$, where \mathbf{N} are the area weighted face normals. These normals can be computed as $\mathbf{N} = -\frac{1}{2} \det(\mathbf{D}_m) \mathbf{D}_m^{-T}$. To see that this is correct, note that $\frac{1}{3} \mathbf{N}^T \mathbf{D}_m = -\mathbf{V} \mathbf{I}$, where V is the volume of the tetrahedron. The negative sign is because \mathbf{N} need to point outwards. Note that, if we computed it, $\hat{\mathbf{G}}_4 = -\mathbf{P} \mathbf{N}_4 = -\mathbf{P}(-\mathbf{N}_1 - \mathbf{N}_2 - \mathbf{N}_3) = -\hat{\mathbf{G}}_1 - \hat{\mathbf{G}}_2 - \hat{\mathbf{G}}_3$.
7. Distribute the forces from faces to vertices. The force \mathbf{f}_1 at the first vertex is one third of the forces on the faces around it, or $\mathbf{f}_1 = \frac{1}{3}(\hat{\mathbf{G}}_2 + \hat{\mathbf{G}}_3 + \hat{\mathbf{G}}_4) = -\frac{1}{3} \hat{\mathbf{G}}_1$. Noting that the forces sum to one, the forces are finally obtained as $\mathbf{f} = S(-\frac{1}{3} \hat{\mathbf{G}})$.

Several observations make the process of factoring this force application simpler. The first observation is that each of these steps can be regarded as a linear operator between vector spaces, where the matrices are taken to be flattened into vectors. In this way, the operators S and G are transposes, so that ignoring the factor of $-\frac{1}{3}$, the operator described by step 7 is the transpose of step 1. Similarly, step 6 is $\frac{1}{2} \det(\mathbf{D}_m)$ times the transpose of step 2. Step 3 is the transpose of step 5. Let $P(\alpha, \beta)$ represent the linear operator applied in step 4. The damping force application can be written as steps 1 through 3, followed by $-\frac{1}{6} \det(\mathbf{D}_m) P(\alpha, \beta)$, followed by the transpose of steps 1 through 3. Finally, the operation \mathbf{C} is just steps 1 through 3, followed by the square root of $\frac{1}{6} \Delta t \det(\mathbf{D}_m) P(\alpha, \beta)$. It remains only to determine what this square root is.

Assume that the square root has the form $P(\alpha', \beta')$. Then,

$$\begin{aligned} P(\alpha', \beta') \hat{\hat{\mathbf{F}}} &= \alpha' \text{tr}(\hat{\hat{\mathbf{F}}}) \mathbf{I} + \beta' \hat{\hat{\mathbf{F}}} + \beta' \hat{\hat{\mathbf{F}}}^T \\ P(\alpha', \beta')^2 \hat{\hat{\mathbf{F}}} &= \alpha' \mathbf{I} \text{tr}(\alpha' \text{tr}(\hat{\hat{\mathbf{F}}}) \mathbf{I} + \beta' \hat{\hat{\mathbf{F}}} + \beta' \hat{\hat{\mathbf{F}}}^T) + \beta' (\alpha' \text{tr}(\hat{\hat{\mathbf{F}}}) \mathbf{I} + \beta' \hat{\hat{\mathbf{F}}} + \beta' \hat{\hat{\mathbf{F}}}^T) + \beta' (\alpha' \text{tr}(\hat{\hat{\mathbf{F}}}) \mathbf{I} + \beta' \hat{\hat{\mathbf{F}}} + \beta' \hat{\hat{\mathbf{F}}}^T)^T \\ &= (3\alpha'^2 + 4\alpha' \beta') \text{tr}(\hat{\hat{\mathbf{F}}}) \mathbf{I} + 2\beta'^2 \hat{\hat{\mathbf{F}}} + 2\beta'^2 \hat{\hat{\mathbf{F}}}^T \\ &= P(3\alpha'^2 + 4\alpha' \beta', 2\beta'^2) \hat{\hat{\mathbf{F}}} \\ \beta' &= \sqrt{\frac{1}{6} \Delta t \det(\mathbf{D}_m)} \left(\frac{1}{2} \sqrt{2\beta} \right) \\ \alpha' &= \sqrt{\frac{1}{6} \Delta t \det(\mathbf{D}_m)} \left(\frac{1}{3} (\sqrt{3\alpha + 2\beta} - \sqrt{2\beta}) \right) \end{aligned}$$

Since the operator $P(\alpha', \beta')$ is symmetric, this construction results in the operator \mathbf{C} , and routines to apply both \mathbf{C} and \mathbf{C}^T are readily constructed with only minor modifications to the existing code.

5.2. SPD Formulation

Assuming a factorization of the form $-\Delta t \mathbf{D} = \mathbf{C}^T \mathbf{C}$ allows us to left multiply the third row of Eq. (17) by \mathbf{C} and subsequently factor \mathbf{C} out of the third column into the solution vector to obtain

$$\begin{aligned} \begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \beta^{-1} \hat{\mathbf{G}} & \mathbf{W} \beta^{-1} \mathbf{W}^T + \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & -\Delta t \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{D} \\ \mathbf{0} & \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{I} - \Delta t \mathbf{M}^{-1} \mathbf{D} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \boldsymbol{\lambda} \\ \mathbf{v}^{n+1} \end{pmatrix} &= \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ \mathbf{W} \mathbf{J} \mathbf{v}^* - \mathbf{W} \mathbf{u}^* \\ \mathbf{v}^* \end{pmatrix} \\ \begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \beta^{-1} \hat{\mathbf{G}} & \mathbf{W} \beta^{-1} \mathbf{W}^T + \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{C}^T \mathbf{C} \\ \mathbf{0} & \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{I} + \mathbf{M}^{-1} \mathbf{C}^T \mathbf{C} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \boldsymbol{\lambda} \\ \mathbf{v}^{n+1} \end{pmatrix} &= \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ \mathbf{W} \mathbf{J} \mathbf{v}^* - \mathbf{W} \mathbf{u}^* \\ -\mathbf{v}^* \end{pmatrix} \\ \begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \beta^{-1} \hat{\mathbf{G}} & \mathbf{W} (\beta^{-1} + \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T) \mathbf{W}^T & \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{C}^T \\ \mathbf{0} & \mathbf{C} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{I} + \mathbf{C} \mathbf{M}^{-1} \mathbf{C}^T \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \boldsymbol{\lambda} \\ \hat{\mathbf{v}} \end{pmatrix} &= \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ \mathbf{W} \mathbf{J} \mathbf{v}^* - \mathbf{W} \mathbf{u}^* \\ \mathbf{C} \mathbf{v}^* \end{pmatrix} \end{aligned}$$

where $\hat{\mathbf{v}} = \mathbf{C} \mathbf{v}^{n+1}$ represents the new unknowns. Note that the substitution of Eq. (11) into Eq. (12) to derive Eq. (17) allowed us to place a \mathbf{D} into the third column of the coefficient matrix in Eq. (17), leading to the factorization of \mathbf{C} out of the coefficient matrix and into the unknowns, and this was the reason for that extra substitution. The resulting system is obviously symmetric, and we can recover \mathbf{v}^{n+1} from $\hat{\mathbf{v}}$ by replacing $\Delta t \mathbf{D} \mathbf{v}^{n+1}$ with $-\mathbf{C}^T \mathbf{C} \mathbf{v}^{n+1} = -\mathbf{C}^T \hat{\mathbf{v}}$ in Eq. (11)

$$\mathbf{M} \mathbf{v}^{n+1} = \mathbf{M} \mathbf{v}^* - \mathbf{C}^T \hat{\mathbf{v}} - \mathbf{J}^T \mathbf{W}^T \boldsymbol{\lambda}.$$

The symmetrized system can be expressed as the sum of three matrices that are evidently symmetric positive semidefinite as

$$\begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & -\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \beta^{-1} \hat{\mathbf{G}} & \mathbf{W} \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{C}^T \\ \mathbf{0} & \mathbf{C} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{C} \mathbf{M}^{-1} \mathbf{C}^T \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}$$

from which the symmetric positive semidefiniteness of the coupled system follows.

To consider the rank of this matrix, note that any vector in the nullspace of the coupled system, consisting of three parts $\hat{\mathbf{p}}$, $\boldsymbol{\lambda}$, and $\hat{\mathbf{v}}$ must necessarily be in the nullspace of each of these three matrices. From the third matrix, we conclude that this requires $\hat{\mathbf{v}} = \mathbf{0}$. Assuming $\hat{\mathbf{v}} = \mathbf{0}$ and looking at the second matrix leads one to consider only requirements for $\boldsymbol{\lambda}$. The second matrix factors as

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{W} \mathbf{J} \\ \mathbf{C} \end{pmatrix} \mathbf{M}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{W} \mathbf{J} \\ \mathbf{C} \end{pmatrix}^T$$

It is necessary for $\mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T \boldsymbol{\lambda} = \mathbf{0}$, which implies that the constraint impulses have absolutely no effect on any degrees of freedom of the structure. The first matrix factors as

$$\begin{pmatrix} \hat{\mathbf{G}}^T \\ -\mathbf{W} \\ \mathbf{0} \end{pmatrix} \beta^{-1} \begin{pmatrix} \hat{\mathbf{G}}^T \\ -\mathbf{W} \\ \mathbf{0} \end{pmatrix}^T$$

A nontrivial nullspace of this matrix satisfies $\beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} - \beta^{-1} \mathbf{W}^T \boldsymbol{\lambda} = \mathbf{0}$, which implies that the combination of constraints and pressures cancels to give no effect on the final fluid velocities \mathbf{u}^{n+1} as can be seen by inspecting Eq. (8). This immediately gives the cases $\beta^{-1} \hat{\mathbf{G}} \hat{\mathbf{p}} = \mathbf{0}$ (Neumann region) and $\mathbf{W}^T \boldsymbol{\lambda} = \mathbf{0}$ (redundant constraints), both of which will lead to nullspaces of the full coupled system by placing zeros in the remaining entries.

The right hand side of our equations must be in the range of the coefficient matrix in order for the system to be solvable. Comparing the form of our right hand side to the requirements that a nullspace must satisfy, we find that it is sufficient but not necessary condition for consistency for \mathbf{v}^* to contain no nullspace components of \mathbf{M}^{-1} and \mathbf{u}^* to contain no nullspace components of β^{-1} . A Neumann region will result in inconsistency if there is a net flux in \mathbf{u}^* through the region boundary, so we enforce consistency by summing up the net flux and subtracting it evenly from the \mathbf{u}^* on the region boundary faces. Clearly constraints which apply only to nullspace components of \mathbf{M}^{-1} and β^{-1} will result in inconsistency and must be avoided. Fluid velocities which are constrained only to nullspace components of \mathbf{M}^{-1} are in effect Neumann grid faces and must be taken into account in detection of Neumann regions. Isolated rigid bodies within a fluid region also do not prevent a region from being a Neumann region. A region that contains deformable bodies does not behave as a Neumann region, since the bodies can change volume to accommodate a net flux into the region.

5.3. Note On Forming SPD System

Many problems can be formulated as a KKT system of the form

$$\begin{pmatrix} \mathbf{M} + \mathbf{C}^T \mathbf{C} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}, \quad (18)$$

where \mathbf{M} is an SPD matrix that is easily inverted. Dissipative systems, such as fluid viscosity or structure damping, produce terms of the form $\mathbf{M} + \mathbf{C}^T \mathbf{C}$. Evolving such systems subject to constraints, such as the incompressibility constraint or the constraint that the solid and fluid velocities be equal at their interface, then leads to a KKT system in the form of Eq. (18). Multiplying the first row of Eq. (18) by $\mathbf{C}\mathbf{M}^{-1}$ produces

$$\mathbf{C}\mathbf{u} + \mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T\mathbf{C}\mathbf{u} + \mathbf{C}\mathbf{M}^{-1}\mathbf{B}^T\lambda = \mathbf{C}\mathbf{M}^{-1}\mathbf{b}. \quad (19)$$

Multiplying the first row of Eq. (18) by $\mathbf{B}\mathbf{M}^{-1}$ and simplifying using the second row produces

$$\mathbf{B}\mathbf{M}^{-1}\mathbf{C}^T\mathbf{C}\mathbf{u} + \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T\lambda = \mathbf{B}\mathbf{M}^{-1}\mathbf{b}. \quad (20)$$

These are then combined into the SPD system using the definition $\hat{\mathbf{u}} = \mathbf{C}\mathbf{u}$ to produce

$$\begin{pmatrix} \mathbf{I} + \mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T & \mathbf{C}\mathbf{M}^{-1}\mathbf{B}^T \\ \mathbf{B}\mathbf{M}^{-1}\mathbf{C}^T & \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T \end{pmatrix} \begin{pmatrix} \hat{\mathbf{u}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{C}\mathbf{M}^{-1}\mathbf{b} \\ \mathbf{B}\mathbf{M}^{-1}\mathbf{b} \end{pmatrix}. \quad (21)$$

Finally, \mathbf{u} is obtained from the first row of Eq. (18) using

$$\mathbf{u} = \mathbf{M}^{-1}\mathbf{b} - \mathbf{M}^{-1}\mathbf{C}^T\hat{\mathbf{u}} - \mathbf{M}^{-1}\mathbf{B}^T\lambda. \quad (22)$$

To demonstrate that this is in fact equivalent to Eq. (18), we show that we can derive Eq. (18) from $\hat{\mathbf{u}} = \mathbf{C}\mathbf{u}$, Eq. (21) and Eq. (22). The first row of Eq. (18) is obtained by substituting $\hat{\mathbf{u}} = \mathbf{C}\mathbf{u}$ into Eq. (22). The constraint row is obtained by subtracting the second row of Eq. (21) from \mathbf{B} times Eq. (22). Thus, the indefinite KKT formulation is equivalent to the SPD formulation.

6. Note On Conditioning

In the absence of coupling ($\mathbf{W} = 0$), the structure system looks like

$$\begin{aligned} \mathbf{v}^{n+1} &= \mathbf{v}^* + \Delta t \mathbf{M}^{-1} \mathbf{D} \mathbf{v}^{n+1} \\ \mathbf{v}^{n+1} - \Delta t \mathbf{M}^{-1} \mathbf{D} \mathbf{v}^{n+1} &= \mathbf{v}^* \\ (\mathbf{I} + \mathbf{M}^{-1} \mathbf{C}^T \mathbf{C}) \mathbf{v}^{n+1} &= \mathbf{v}^* \end{aligned}$$

The modified structure system looks like

$$(\mathbf{I} + \mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T)\hat{\mathbf{v}} = \mathbf{C}\mathbf{v}^*$$

These two systems have the same eigenvalues. To see this

$$\begin{aligned}(\mathbf{I} + \mathbf{M}^{-1}\mathbf{C}^T\mathbf{C})\mathbf{v} &= \alpha\mathbf{v} \\ \mathbf{C}(\mathbf{I} + \mathbf{M}^{-1}\mathbf{C}^T\mathbf{C})\mathbf{v} &= \mathbf{C}\alpha\mathbf{v} \\ (\mathbf{I} + \mathbf{C}\mathbf{M}^{-1}\mathbf{C}^T)\hat{\mathbf{v}} &= \alpha\hat{\mathbf{v}}\end{aligned}$$

so that if \mathbf{v} is an eigenvector with eigenvalue α in the original system, then $\hat{\mathbf{v}}$ is an eigenvector of the modified system with the same eigenvalue. Note also that, though modifying the system in this way may introduce new eigenvalues (since the system may be larger), none of these eigenvalues will be visible to a Krylov solver. This is because the right hand side is of the form $\mathbf{C}\mathbf{v}^*$ and application of the matrix results in vectors that are also of this form. The modified system may have fewer eigenvalues, however (caused by rank deficiency of \mathbf{C}). The important point is that this modification to the structure system will not adversely affect its convergence properties with a Krylov solver.

7. Viscosity

Applying viscosity as a separate step in the time evolution does not yield a fully coupled time integration scheme, since viscous forces cannot apply momentum to structures. This can be overcome by adding viscosity directly into the coupled formulation. With this in mind, we redefine the intermediate velocity

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \frac{\Delta t}{\rho}\mathbf{f}. \quad (23)$$

Assume that the fluid viscosity μ is constant and ignore the coupling term. Then, the fluid momentum equation becomes

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + \frac{\Delta t}{\rho}\mathbf{f}\rho\mathbf{u}^{n+1} = \rho\mathbf{u}^* - \Delta t\nabla p + \Delta t\mu\nabla^2\mathbf{u}^{n+1}. \quad (24)$$

The viscous terms introduce a new gradient operator, which acts on face velocities. The gradient operator itself is discretized using central differences. The boundary conditions at structures are Dirichlet conditions, noting that these velocities are constrained implicitly using \mathbf{W} and remain degrees of freedom rather than being moved to the right hand side as would normally be done. The discretization of \mathbf{W} used above is not suitable for viscosity, however, since it does not couple to structures in the tangential direction. This is resolved by locating faces adjacent to those in \mathbf{W} and adding them to \mathbf{W} as shown in Figure 2. Domain boundary conditions can be discretized in the usual way. For simplicity of implementation, however, we treat domain walls as infinitely massive structures and add terms to \mathbf{W} , \mathbf{J} , and \mathbf{M}^{-1} accordingly. This discretization is limited to first order accuracy since it uses the first order accurate constraint locations.

Let $\mathbf{G}_\mu = \sqrt{\Delta t\mu}\nabla$ be the discretized gradient operator that acts on face velocities, where the scaling factors are pulled into the resulting matrix to simplify the presentation. Substituting this into Eq. (24) and adding back the coupling term produces the momentum equation

$$\beta\mathbf{u}^{n+1} = \beta\mathbf{u}^* - \hat{\mathbf{G}}\hat{\mathbf{p}} - \mathbf{G}_\mu^T\mathbf{G}_\mu\mathbf{u}^{n+1} + \mathbf{W}^T\boldsymbol{\lambda}. \quad (25)$$

Define $\hat{\mathbf{u}} = \mathbf{G}_\mu\mathbf{u}^{n+1}$. Then, multiplying Eq. (25) through by both $\hat{\mathbf{G}}^T\beta^{-1}$ (using Eq. (9)) and $\mathbf{G}_\mu\beta^{-1}$ produces

$$\hat{\mathbf{G}}^T\beta^{-1}\hat{\mathbf{G}}\hat{\mathbf{p}} + \hat{\mathbf{G}}^T\beta^{-1}\mathbf{G}_\mu^T\hat{\mathbf{u}} - \hat{\mathbf{G}}^T\beta^{-1}\mathbf{W}^T\boldsymbol{\lambda} = \hat{\mathbf{G}}^T\mathbf{u}^* \quad (26)$$

$$\mathbf{G}_\mu\beta^{-1}\hat{\mathbf{G}}\hat{\mathbf{p}} + (\mathbf{I} + \mathbf{G}_\mu\beta^{-1}\mathbf{G}_\mu^T)\hat{\mathbf{u}} - \mathbf{G}_\mu\beta^{-1}\mathbf{W}^T\boldsymbol{\lambda} = \mathbf{G}_\mu\mathbf{u}^*. \quad (27)$$

Multiplying Eq. (25) through by $\mathbf{W}\beta^{-1}$ and substituting in Eq. (12) followed by Eq. (11), $-\Delta t\mathbf{D} = \mathbf{C}^T\mathbf{C}$ and $\mathbf{C}\mathbf{v}^{n+1} = \hat{\mathbf{v}}$ produces

$$\mathbf{W}\beta^{-1}\hat{\mathbf{G}}\hat{\mathbf{p}} + \mathbf{W}\beta^{-1}\mathbf{G}_\mu^T\hat{\mathbf{u}} - \mathbf{W}(\beta^{-1} + \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)\mathbf{W}^T\boldsymbol{\lambda} - \mathbf{W}\mathbf{J}\mathbf{M}^{-1}\mathbf{C}^T\hat{\mathbf{v}} = \mathbf{W}\mathbf{u}^* - \mathbf{W}\mathbf{J}\mathbf{v}^*. \quad (28)$$

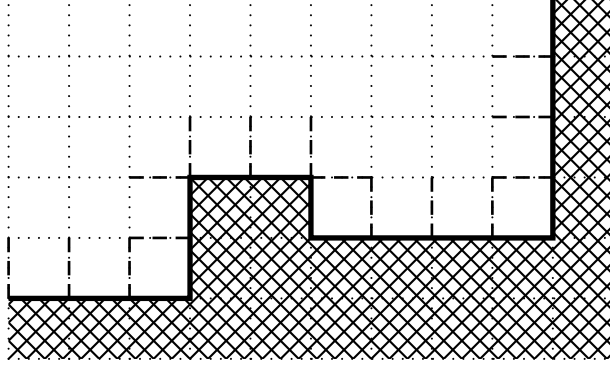


Figure 2: The hashed region is the rasterized structure, and the thick faces at its boundary are the original coupling faces in \mathbf{W} . For viscosity, the dashed faces are added to \mathbf{W} to allow tangential forces to be exchanged between fluid and structure.

Combining these with Eq. (11) produces

$$\begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & \hat{\mathbf{G}}^T \beta^{-1} \mathbf{G}_\mu^T & -\hat{\mathbf{G}}^T \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ \mathbf{G}_\mu \beta^{-1} \hat{\mathbf{G}} & \mathbf{I} + \mathbf{G}_\mu \beta^{-1} \mathbf{G}_\mu^T & -\mathbf{G}_\mu \beta^{-1} \mathbf{W}^T & \mathbf{0} \\ -\mathbf{W} \beta^{-1} \hat{\mathbf{G}} & -\mathbf{W} \beta^{-1} \mathbf{G}_\mu^T & \mathbf{W}(\beta^{-1} + \mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T) \mathbf{W}^T & \mathbf{W} \mathbf{J} \mathbf{M}^{-1} \mathbf{C}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{C} \mathbf{M}^{-1} \mathbf{J}^T \mathbf{W}^T & \mathbf{I} + \mathbf{C} \mathbf{M}^{-1} \mathbf{C}^T \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{u}} \\ \lambda \\ \hat{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ \mathbf{G}_\mu \mathbf{u}^* \\ \mathbf{W} \mathbf{J} \mathbf{v}^* - \mathbf{W} \mathbf{u}^* \\ \mathbf{C} \mathbf{v}^* \end{pmatrix} \quad (29)$$

This equation has a structure very similar to what was obtained without viscosity, and it can be shown to be symmetric positive semidefinite in exactly the same way.

Stokes Flow. When the Reynolds number is very small, the viscosity term dominates the inertial terms. Because viscosity and pressure are fully coupled in a single linear system, our scheme does not experience difficulties at high viscosity. In this case, one may also approximate the Navier-Stokes equations with those of Stokes flow by omitting the inertial terms. We can simulate the unsteady Stokes flow equations directly in our framework by omitting advection terms for the fluid (Step 6) and the structure (Step 4). This renders the entire first half of the update unnecessary, and our evolution scheme reduces to

1. $\mathbf{v}^* = \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{F}_e(\mathbf{x}^{n+1})$
2. $\mathbf{u}^* = \mathbf{u}^n + \frac{\Delta t}{\rho} \mathbf{f}$
3. Coupled solve with known quantities \mathbf{u}^* , \mathbf{v}^* , and \mathbf{x}^{n+1} and unknown quantities \mathbf{u}^{n+1} , \mathbf{p} , and \mathbf{v}^{n+1}

We demonstrate our ability to simulate with high viscosity and Stokes flow in Section 9.2.

As discussed in [25], the indefinite system of equations derived for fluid-structure interaction has many similarities to the Stokes equations as both are dissipative systems combined with constraints (see Section 5.3). Indeed, we produce an SPD formulation for both. If no structures are present then $\mathbf{W} = \mathbf{0}$, $\mathbf{C} = \mathbf{0}$ and $\mathbf{J} = \mathbf{0}$, and Eq. (29) simplifies to

$$\begin{pmatrix} \hat{\mathbf{G}}^T \beta^{-1} \hat{\mathbf{G}} & \hat{\mathbf{G}}^T \beta^{-1} \mathbf{G}_\mu^T \\ \mathbf{G}_\mu \beta^{-1} \hat{\mathbf{G}} & \mathbf{I} + \mathbf{G}_\mu \beta^{-1} \mathbf{G}_\mu^T \end{pmatrix} \begin{pmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{G}}^T \mathbf{u}^* \\ \mathbf{G}_\mu \mathbf{u}^* \end{pmatrix}. \quad (30)$$

This system provides an SPD treatment of unsteady Stokes flow. If convection is handled explicitly, this formulation also provides an SPD treatment of Navier-Stokes.

Fully implicit Navier Stokes. We could take a fully implicit approach to the full nonlinear Navier-Stokes equations by using a Newton-Raphson iteration method, where we linearize at each Newton step to get a system of the form in Eq. (30). The equation to be iterated is

$$\mathbf{u}_{(i+1)}^{n+1} = \mathbf{u}^n - \Delta t (\mathbf{u}_{(i)}^{n+1} \cdot \nabla) \mathbf{u}_{(i)}^{n+1} + \frac{\Delta t}{\rho} (-\nabla \mathbf{p} + \mu \nabla^2 \mathbf{u}_{(i+1)}^{n+1} + \mathbf{f}),$$

where $\mathbf{u}_{(i)}^{n+1}$ is the time t^{n+1} fluid velocity after the i^{th} Newton iteration and we let $\mathbf{u}_{(0)}^{n+1} = \mathbf{u}^n$. The inner iterations are performed by solving the symmetric and positive definite system Eq. (30). For each outer iteration, a new system is set up using the new estimate of the final velocity to recompute the advection term. A similar approach could be explored for treatment of fully implicit coupled fluid-structure evolution.

8. Note on Stability

We have found the proposed monolithic implicit coupling scheme described in this paper to be quite stable. We have never observed stability problems due to this simple choice. In the example from Figure 9, we consider the structure discretization to be exact and refine only the fluid discretization. This results in a grid spacing much smaller than the lengths of edges in the structure mesh, which has never been observed to cause stability problems. We also tried running examples with a fixed resolution fluid mesh, refining the structure so that the grid spacing was over 30 times the lengths of the structure edges, again without noticing any stability problems. We have tried making structures a billion times less dense than the fluid and a billion times more dense than the fluid, and neither modification lead to stability problems.

We also note that solving the coupled linear system and applying the resulting update to the velocities of the structure and the fluid has the property that it never increases total kinetic energy. Since positions are not changed during this process, potential energy is unchanged. Let

$$\mathbf{K} = \begin{pmatrix} \hat{\mathbf{G}}^T & \mathbf{0} \\ \mathbf{G}_\mu & \mathbf{0} \\ -\mathbf{W} & \mathbf{W}\mathbf{J} \\ \mathbf{0} & \mathbf{C} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \beta & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{pmatrix} \quad \mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} \mathbf{u}^* \\ \mathbf{v}^* \end{pmatrix} \quad \mathbf{z} = \begin{pmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{u}} \\ \lambda \\ \hat{\mathbf{v}} \end{pmatrix}$$

Eq. (29) reduces to $(\mathbf{K}\mathbf{B}^{-1}\mathbf{K}^T + \mathbf{P})\mathbf{z} = \mathbf{K}\mathbf{w}$. The new velocities are $\mathbf{w} - \mathbf{B}^{-1}\mathbf{K}^T\mathbf{z}$. Finally, the change in kinetic energy is

$$\begin{aligned} 2KE^{n+1} - 2KE^* &= (\mathbf{w} - \mathbf{B}^{-1}\mathbf{K}^T\mathbf{z})^T \mathbf{B} (\mathbf{w} - \mathbf{B}^{-1}\mathbf{K}^T\mathbf{z}) - \mathbf{w}^T \mathbf{B} \mathbf{w} \\ &= -2\mathbf{w}^T \mathbf{K}^T \mathbf{z} + (\mathbf{K}^T \mathbf{z})^T \mathbf{B}^{-1} (\mathbf{K}^T \mathbf{z}) \\ &= -2\mathbf{z}^T (\mathbf{K}\mathbf{B}^{-1}\mathbf{K}^T + \mathbf{P})\mathbf{z} + \mathbf{z}^T \mathbf{K}\mathbf{B}^{-1}\mathbf{K}^T \mathbf{z} \\ &= -\mathbf{z}^T (\mathbf{K}\mathbf{B}^{-1}\mathbf{K}^T + 2\mathbf{P})\mathbf{z} \leq 0 \end{aligned}$$

This does not imply that we always decrease kinetic energy in a full time step, since elastic forces added to the structures or gravity added to fluids may both increase kinetic energy. In particular, we are still explicit on elastic forces, and this continues to impose a CFL restriction. On the other hand, if there is no potential energy (e.g., rigid bodies without forces between them), and the remaining steps in the time integration scheme make a similar guarantee on kinetic energy, the resulting scheme would be unconditionally stable. We also note that if there is neither damping nor viscosity, then $\mathbf{P} = \mathbf{0}$, and the process of obtaining the final velocities from the intermediate velocities \mathbf{w} is described by the mass-symmetric projection matrix $\mathbf{I} - \mathbf{B}^{-1}\mathbf{K}^T(\mathbf{K}\mathbf{B}^{-1}\mathbf{K}^T)^{-1}\mathbf{K}$, which is exactly the property that would be desired in this case.

9. Examples

The SPD formulation merely gives an analytically equivalent expression of the indefinite form of the equations. What we expect to illustrate in these examples is that the method is numerically stable. We examine several common examples from the fluid structure interaction literature to demonstrate the convergence properties of our method.

We choose our time step size to be the minimum of the time step size required for fluids alone and the time step size required for structures alone. The fluid time step size is computed using a CFL number of 0.9 for all of our examples. Except in the stability test described below, the structures in our examples are not refined, so the time step size computed for the structures will not change significantly from resolution

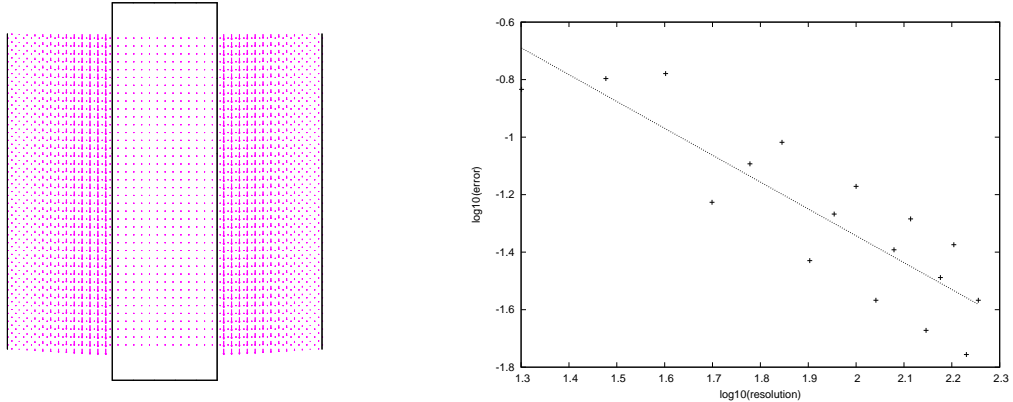


Figure 3: Analytic test with a structure flowing through a channel. The errors are computed as the difference between the steady state velocity of the moving structure and the analytic solution. The error depends on the grid resolution mod 3 since this value determines how the structure aligns with the domain. The convergence order of the errors is fit as .93.

to resolution. For higher resolutions, the fluid time step restriction will dominate. We note that since all of our examples converge to nonzero velocities under refinement, we effectively refine with $\Delta t \propto \Delta x$. Thus, the first order convergence demonstrated in our examples implies that our scheme is at least first order accurate in space and time. We do not test convergence order separately for space and time, since we do not expect to achieve better than first order.

9.1. Analytic example

Figure 3 shows a simple setup designed to mimic an infinite channel with a rigid structure flowing through it and centered in the channel. The density of the fluid is $\rho = 100kg\ m^{-2}$, and the mass of the rigid body is $M = 150kg$. The height of the fluid domain is $h = 1m$, and the total domain width of $1m$ is divided into two fluid regions with width $w = \frac{1}{3}m$ with the central $\frac{1}{3}m$ occupied by the structure. The fluid viscosity is $\mu = 100kg\ s^{-1}$. Both the fluid and structure experience gravity $g = -9.8ms^{-1}$. We elongate the rigid body to extend slightly beyond the domain and periodically reset its position, providing the illusion of an infinitely long rigid body and allowing the simulation to be run long enough to reach steady state. Altering the dimensions of the structure does not affect the analytic solution, provided its mass is not altered.

We derive the analytic solution by considering the structure to be a boundary condition to the fluid, and then solving for the structure's velocity using conservation of energy. The analytic velocity profile for the left fluid region $\Omega = [0, w] \times [0, h]$ is $u = 0$ and $v = \frac{\rho g}{2\mu}x(x - w) + \frac{v_s}{w}x$, where the v is the vertical fluid velocity component and v_s is the vertical component of the structure velocity. The fluid experiences no pressure gradient. The only unknown remaining is v_s , the velocity of the structure. Taking into account the symmetry of the system, conservation of energy implies that the viscous dissipation equals the total work applied by gravity, so that $2 \int_{\Omega} (\tau \cdot \nabla) \mathbf{u} dV = 2 \int_{\Omega} \rho g v dV + M g v_s$. Solving this equation yields the analytic solution $v_s = -(M + \rho w h) \frac{g w}{2 h \mu}$.

9.2. Rigid cylinder falling under gravity

Motivated by [30, 1], a rigid cylinder is allowed to fall under an externally applied gravitational force. The fluid domain has dimensions $2L \times 8L$, and the cylinder has radius r . If the fluid density is ρ_f and the solid density is ρ_s , then the terminal velocity is [30]

$$v_{term} = \frac{(\rho_s - \rho_f) g r^2}{4\mu} \left(-\ln\left(\frac{r}{L}\right) - 0.9157 + 1.7244\left(\frac{r}{L}\right)^2 - 1.7302\left(\frac{r}{L}\right)^4 \right).$$

It is instructive to note the origin of this analytic solution. Consider a fluid domain consisting of an infinite 2D channel with parallel walls separated by a distance of $2L$. In the middle of this channel is a cylinder with

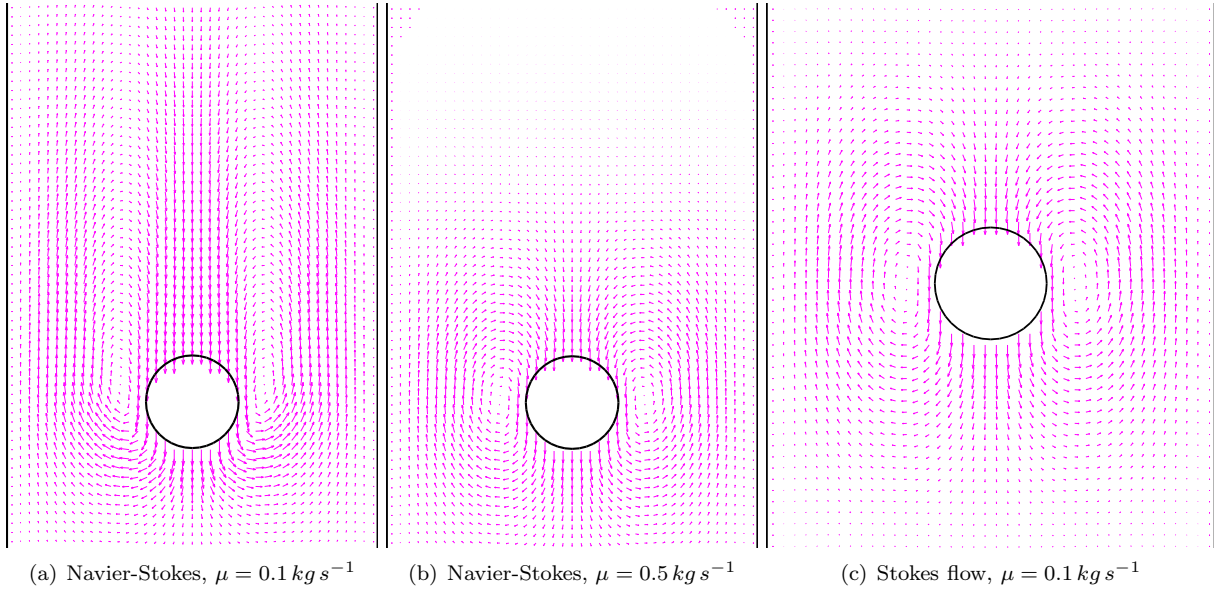


Figure 4: Steady state velocity field.

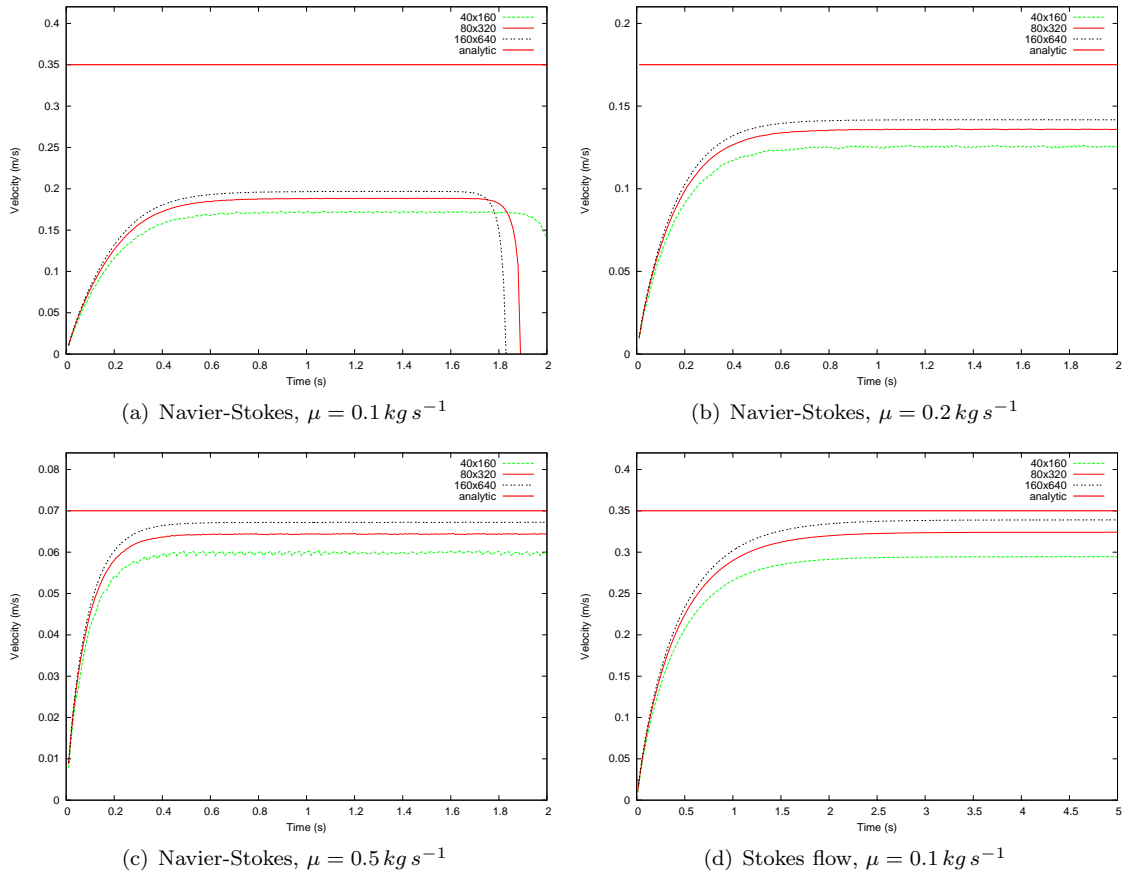


Figure 5: Velocity of a falling cylinder in a channel over time.

Equations	Viscosity ($kg s^{-1}$)	Terminal Velocities (ms^{-1})				Convergence order	
		Analytic	40×160	80×320	160×640		
N-S	0.1	-0.3501	-0.1716	-0.1882	-0.1966	-	-
N-S	0.2	-0.1750	-0.1254	-0.1359	-0.1417	-	-
N-S	0.5	-0.07002	-0.05983	-0.06438	-0.06721	0.85	1.01
N-S	1.0	-0.03501	-0.03052	-0.03264	-0.03399	0.92	1.22
N-S	2.0	-0.01750	-0.01533	-0.01635	-0.01702	0.91	1.26
N-S	5.0	-0.007002	-0.006148	-0.006547	-0.006828	0.91	1.38
N-S	10.0	-0.003501	-0.003013	-0.003279	-0.003417	1.14	1.40
Stokes	0.1	-0.3501	-0.2945	-0.3241	-0.3390	1.10	1.23

Figure 6: Terminal velocity of a falling cylinder and convergence orders.

radius a moving with a constant velocity v_s along the axis of the channel. Under the simplifying assumption of Stokes flow, the resistive force experienced by the cylinder is approximately [15]

$$F = \frac{4\pi\mu v_s}{-\ln\left(\frac{r}{L}\right) - 0.9157 + 1.7244\left(\frac{r}{L}\right)^2 - 1.7302\left(\frac{r}{L}\right)^4}.$$

Note that because of the Stokes flow assumption, fluid advection and movement of the cylinder are assumed to be insignificant and dropped. The steady state solid velocity is obtained by balancing the resistive force against the net gravitational force $F = (\rho_s - \rho_f)g\pi r^2$ on the solid (taking into account buoyancy).

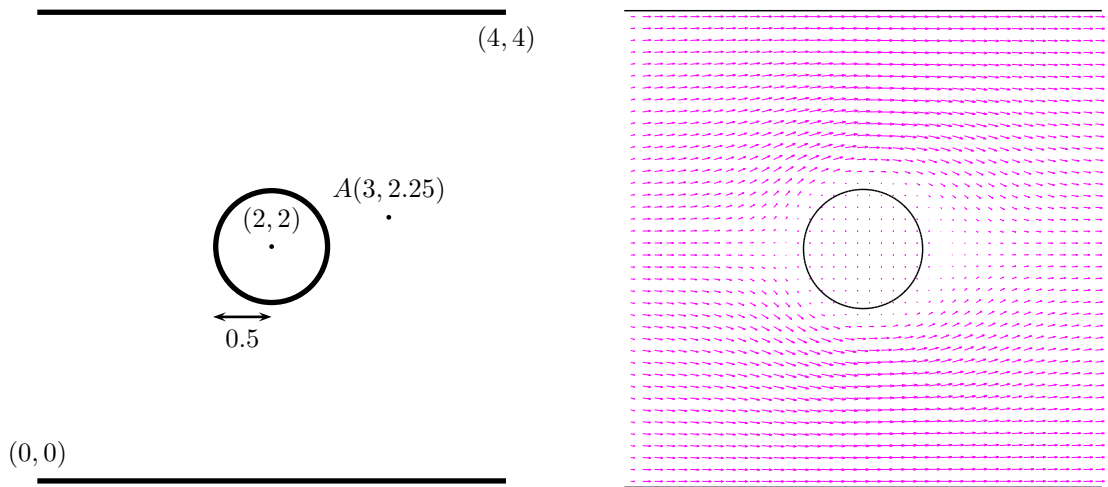
We use the same parameters as [30]: $g = 9.8 ms^{-1}$, $r = 5 \times 10^{-3} m$, $L = 2 \times 10^{-2} m$, $\mu = 0.1 kg s^{-1}$, $\rho_f = 1 \times 10^3 kg m^{-2}$, and $\rho_s = 2 \times 10^3 kg m^{-2}$. The top of the fluid domain has a zero pressure Dirichlet boundary condition, and the sides and bottom are no-slip walls. Our results are shown in Figure 5(a). We note that our results do not agree with those in [30], and we note that their velocity profile does appear to reach a terminal velocity. We converge to a value significantly lower than the analytic solution. Noting that the analytic solution was derived under the assumption of Stokes flow, we also ran with higher viscosities: $\mu = 0.2 kg s^{-1}$ (Figure 5(b)), $\mu = 0.5 kg s^{-1}$ (Figure 5(c)), $\mu = 1.0 kg s^{-1}$, $\mu = 2.0 kg s^{-1}$, $\mu = 5.0 kg s^{-1}$, and $\mu = 10.0 kg s^{-1}$. We obtain close agreement with the analytic solution for $\mu = 0.5 kg s^{-1}$ and higher viscosities. We also repeated the the original test $\mu = 0.1 kg s^{-1}$ using Stokes flow and converge to the analytic terminal velocity as shown in Figure 5(d). The results from all of these simulations are summarized in Figure 6, and the steady state velocities are shown in Figure 4.

The velocity profile shown in Figure 7 of [1] does show a faster initial increase in velocity followed by the velocity profile reaching a plateau, which is consistent with the behavior we observe. Their method, like ours, is first order accurate, and their figure shows two velocity profiles for each of two methods corresponding to a factor-of-four refinement. This suggests that the close fit shown in their highest resolution may be coincidental and that their scheme will actually converge to a value somewhat higher than the predicted value.

9.3. Laminar flow past a cylinder

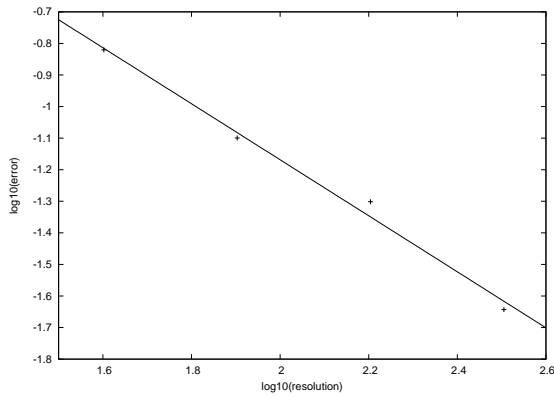
In this example, we consider a simple one-way coupled setup and directly compare our technique against a different scheme. For the comparison scheme, we use standard Chorin splitting [8]. We then choose a sample point at a time and position which exhibits dynamic behavior and demonstrate that the two schemes converge to the same velocity at that point. This demonstrates that we are consistent with an existing scheme from the literature in the simpler case where structures are only one-way coupled to the fluid.

For this comparison we use laminar flow across a fixed cylinder. The domain is $[0 m, 4 m] \times [0 m, 4 m]$ with a cylinder of radius $0.5 m$ located at $(2, 2)$. The cylinder has no-slip boundary conditions. The top and bottom walls have slip boundary conditions. The right wall has the condition $p = 0$, and the left wall has the boundary condition $\mathbf{u}_1 = 1$ and a slipping boundary condition for \mathbf{u}_2 . The density is $\rho = 1 kg m^{-2}$, and the viscosity is $\mu = 0.1 kg s^{-1}$. The initial fluid velocity is $\mathbf{u} = \mathbf{0}$. The setup is illustrated in Figure 7(a).

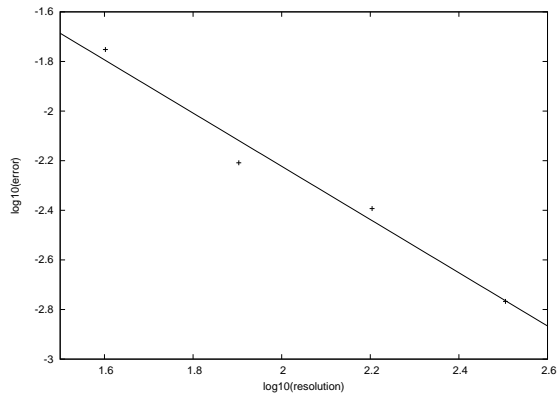


(a) Domain configuration (units in m)

(b) Velocity field



(c) Convergence of u_1 at A .



(d) Convergence of u_2 at A .

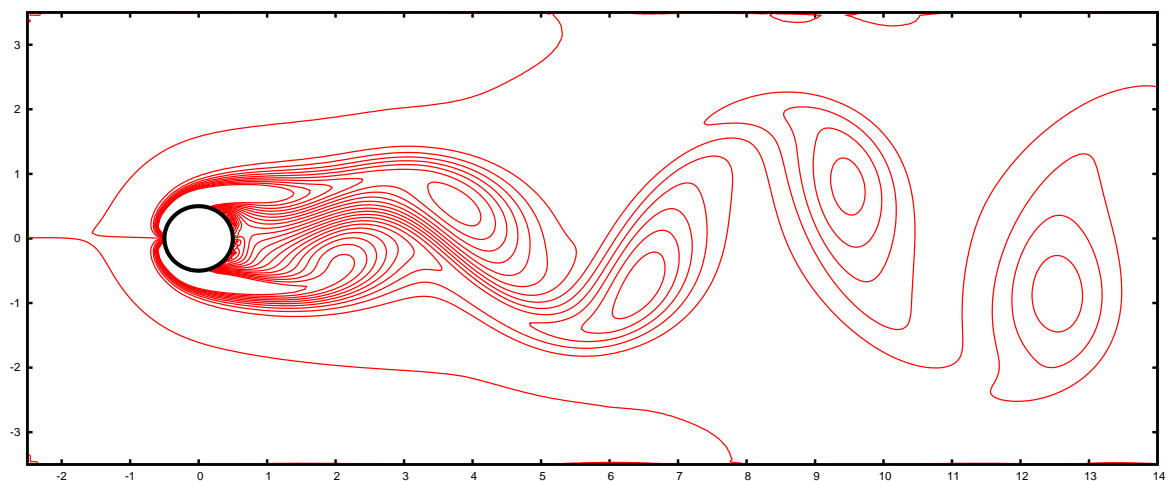
Figure 7: Laminar flow around a fixed cylinder.

The fluid velocity is sampled at the point $(3, 2.25)$ at time $0.5s$ using linear interpolation. The fluid velocity is far from steady state at this time, so this comparison also tests dynamic behavior.

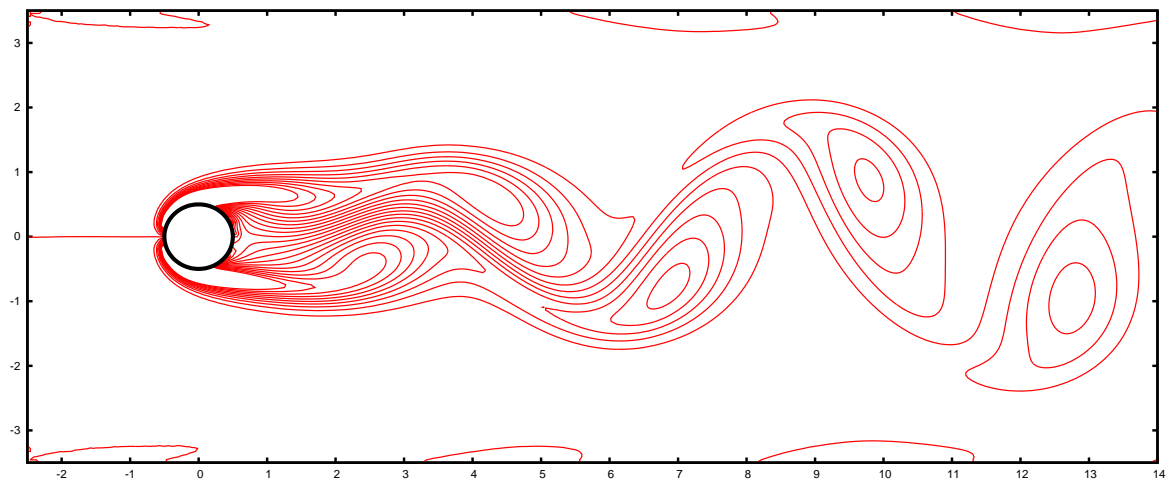
The Chorin splitting implementation at resolution 640×640 is taken to be the exact solution. The example was then run using the proposed method at resolutions 40×40 , 80×80 , 160×160 , and 320×320 . The differences for each velocity component are plotted in Figure 7(c) and Figure 7(d). The convergence order for \mathbf{u}_1 is 0.89, and the convergence order for \mathbf{u}_2 is 1.07. The final velocity profile is shown in Figure 7(b).

9.4. Vortex shedding

In this example we demonstrate Kármán vortex shedding around a cylinder and compare the dynamic behavior of our method in the case of low viscosity and one-way coupling against the results that others have obtained and against our implementation of Chorin splitting. Our setup is based on that of [17, 6], except that we employ slightly different boundary conditions. Our domain is $[-2.5 m, 15.0 m] \times [-3.5 m, 3.5 m]$ with a fixed cylinder of radius $0.5 m$ centered at the origin. The top and bottom of the domain walls have slip



(a) Proposed method



(b) Chorin splitting

Figure 8: Vorticity contours demonstrating vortex shedding. Contours are provided for the range $[-2.5, 2.5]$ in increments of 0.25.

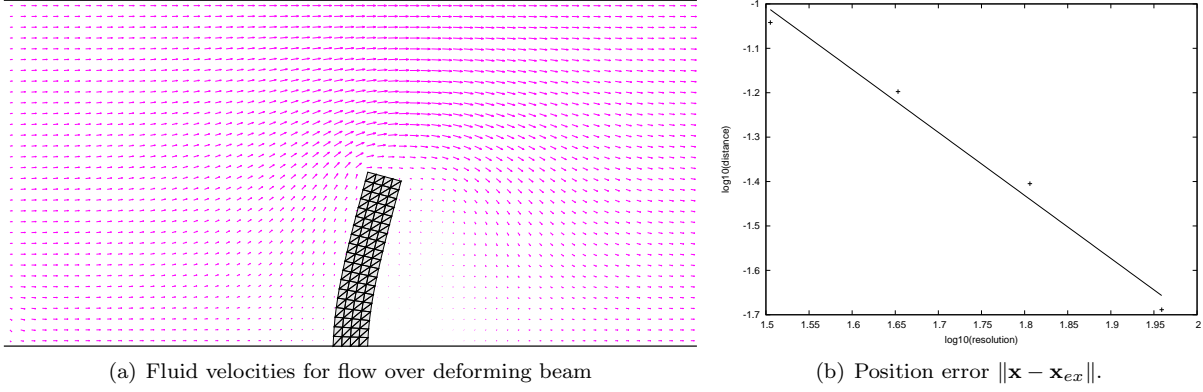


Figure 9: Velocity profile and position error for the top right particle of the deforming beam at time 1 s. The position of the particle at the highest resolution is taken to be the exact value \mathbf{x}_{ex} .

boundary conditions. The left boundary has a source term $\mathbf{u}_1 = 1 \text{ m s}^{-1}$ with a slipping boundary condition for \mathbf{u}_2 . The right boundary has the boundary condition $p = 0$. The flow has $Re = 100$ with $\rho = 1 \text{ kg m}^{-2}$ and $\mu = 0.01 \text{ kg s}^{-1}$. We used a 400×160 grid for our proposed method and for Chorin splitting. Our vorticity contours for our proposed method and Chorin splitting are shown in Figure 8. Our vorticity contours exhibit the same characteristics as those shown in [17, 6] as well as being quite similar to each other. We note that the contour plot for the proposed method has an additional zero contour near the left side. The vorticity in this region is quite small, reaching its peak at roughly 1% of the separation between two contour lines. At resolution 200×80 , the vorticity peaks at roughly 3% of the separation between two contour lines, indicating that the extra contour line will go away under refinement. For the proposed method, the frequency and Strouhal number are $f = 0.160 \text{ Hz}$ and $St = 0.160$. For the Chorin splitting implementation, $f = 0.166 \text{ Hz}$ and $St = 0.166$. These compare favorably with [6], which reported $St = 0.164 - 0.173$ depending on the details of their mesh.

9.5. Flow past deformable beam

This problem deals with a deformable beam immersed in fluid, similar to problems investigated by [3, 31, 29, 30, 9, 27, 19, 32]. The fluid domain was $[0 \text{ m}, 2 \text{ m}] \times [0 \text{ m}, 1 \text{ m}]$ with velocity boundary conditions of 0.25 m s^{-1} on the left side, slip walls on the top and bottom, and a $p = 0$ boundary condition on the right wall. The beam has no-slip boundary conditions. The fluid density is 100 kg m^{-2} , and the coefficient of dynamic viscosity is $\mu = 1 \text{ kg s}^{-1}$. The structure is discretized as a triangle mesh using a mass-spring model with 20 particles in the vertical direction and 5 in the horizontal, each of mass 0.1 kg except for the bottom row of particles, which are treated as infinitely massive. The mesh is initially rectangular, with extent from $[0.95 \text{ m}, 1.05 \text{ m}] \times [0 \text{ m}, 0.5 \text{ m}]$. Edge springs with Young's modulus $E = 166 \text{ kg m}^{-1} \text{ s}^{-2}$ and overdamping fraction 2 along with altitude springs with Young's modulus $E = 400 \text{ kg m}^{-1} \text{ s}^{-2}$ and overdamping fraction 2 were used as the structure constitutive model. (If considered in isolation, a spring can be made critically damped by choosing a specific damping constant. When we refer to an overdamping fraction of 2, we mean that the damping coefficient on the spring is twice the value that would result in a critically damped spring. Isolated springs with an overdamping fraction greater than one are overdamped, and isolated springs with an overdamping fraction less than one are underdamped. The analogy does not extend to systems of springs, but we find this approach to be more intuitive for describing the amount of damping present.) Altitude springs [21] create a spring between each particle of a triangle and a virtual node projected onto the opposite face and are used to give volumetric structure and prevent inversion.

Note for the purposes of our convergence analysis we take the structure to be a discrete system and refine only the fluid grid resolution. We examine convergence with fluid grid resolution and time step size in Figure 9(b). The error is computed assuming the simulation with resolution 256×128 is the exact solution. We plot the convergence of $\|\mathbf{x} - \mathbf{x}_{ex}\|$, where \mathbf{x}_{ex} is the position in the exact solution configuration, for the

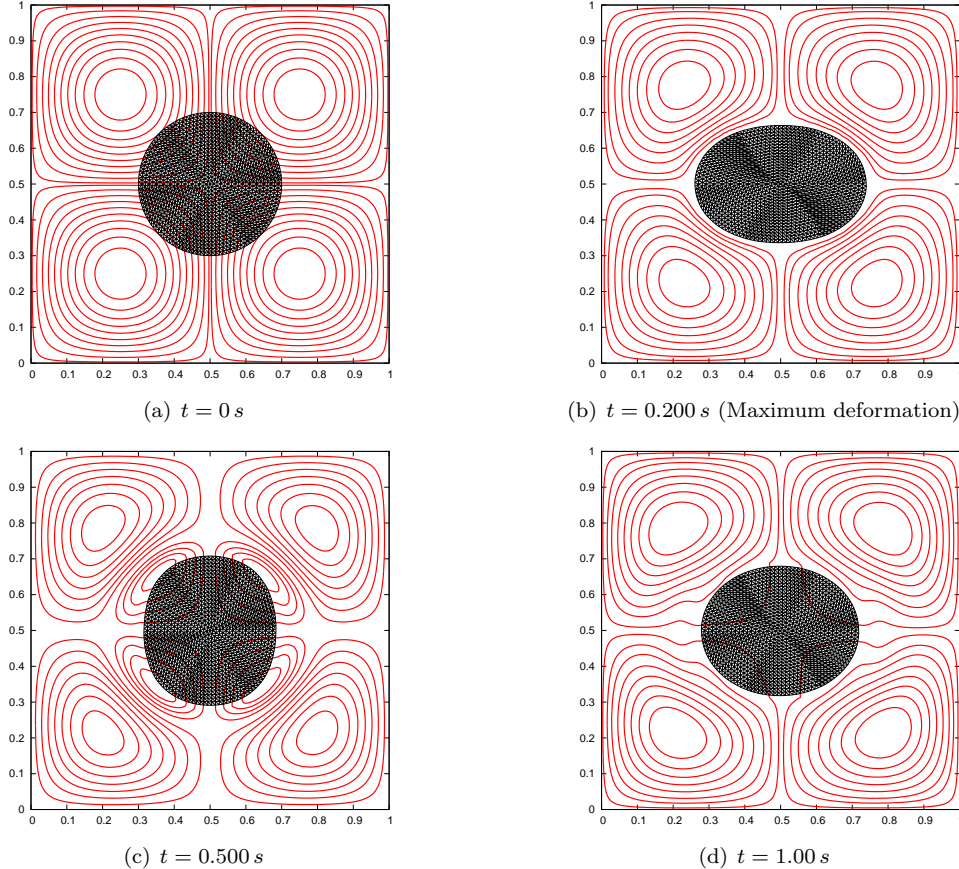


Figure 10: Streamlines in the oscillating disk example. The fluid grid resolution is 128×128 and the structure is composed of 2400 triangles.

top right particle of the beam at time 1 s in Figure 9(b). The convergence order is 1.42. The velocity profile is shown in Figure 9(a).

9.6. Oscillating disk

We have implemented the oscillating disk example from [32], where a circular deformable body is placed within a fluid domain of dimensions $[0 m, 1 m] \times [0 m, 1 m]$ with periodic boundary conditions and an initial velocity defined for both the fluid and the structure by the stream function $\psi = \psi_0 \sin(k_x x) \sin(k_y y)$, with $\psi_0 = 0.05 m^2 s^{-1}$ and $k_x = k_y = 2\pi m^{-1}$. We used a neo-Hookean constitutive model as described in [5] for the structure with Poisson's ratio $\nu = 0.475$, shear modulus $G = 1 Pa$, and Rayleigh damping coefficient 0.005. The structure and fluid densities were set to $\rho = 1 kg m^{-2}$ and the coefficient of dynamic viscosity was $\mu = 0.001 kg s^{-1}$. We ran the simulation for a period of 1 s. For comparison to [32] we rasterize the structure velocities onto the fluid grid and display the streamlines and structure deformation (Figure 10). The frequency of the oscillator is similar to that in [32] as can be seen from the energy plots in Figure 11. We note that our simulation is more damped than theirs. Our extra damping is due to our use of semi-Lagrangian advection on the fluid and Rayleigh damping on the structure.

10. Conclusions and Future Work

We have presented a method for formulating strongly coupled fluid structure interaction problems as a symmetric positive definite linear system. The method is based on factoring the structure damping matrix

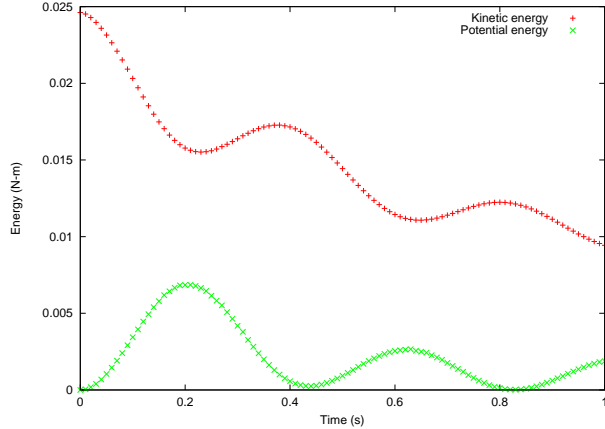


Figure 11: Energy over time for the oscillating disk shown in Figure 10.

and performing a change of variables in the structure unknowns. It can be applied to arbitrary linearized structure constitutive models, but requires a semi-implicit time discretization for the structure equations where elastic terms are handled explicitly and damping terms are handled implicitly. For the special case of only rigid structures and no other rigid-rigid forces we recover exactly the method of [4]. The numerical experiments on problems common in the fluid structure interaction literature demonstrate that the method is convergent with grid resolution.

Second order accuracy for fluid structure interaction has been achieved by some authors, such as the formally second order methods [13, 12]. We note, however, that second order accuracy came well after the original method and that the additional accuracy comes at the cost of additional complexity. An approach which attempts to be second order accurate in space might resolve the boundary between the fluid and structure more accurately by placing constraints at the structure surface rather than at MAC face centers or otherwise take advantage of the available geometric information. Second order accuracy in time would be more difficult to achieve, as it would require finding second order accurate integration schemes for the fluid and the structure that retain that accuracy when coupled. Instead of attempting to address second order accuracy in this paper, we leave the problem of higher order accuracy for future work.

Compared to standard Chorin splitting (without structures), our evolution will be significantly slower. Chorin splitting with implicit viscosity requires the solution of one well-conditioned viscosity system per dimension and a single poorly-conditioned Poisson system for which preconditioning has been well studied. We replace these with a single linear system that, due to the Poisson matrix it contains, is also poorly conditioned. We have observed its conditioning to be significantly worse at high viscosity than low viscosity, and we have also observed poor conditioning to result from underresolved regions. In particular, we have observed poor convergence when the structure is much coarser than the fluid grid and we add extra coupling faces to \mathbf{W} as described in Section 7, possibly due to overconstraint. Increasing the resolution of the structure improves convergence in these cases, but a more careful treatment of boundary conditions might avoid the issue entirely. Conditioning with deformable structures becomes worse with lower density and higher damping. The conditioning improves significantly when all structures are one-way coupled.

The system is symmetric positive definite, making it suitable for solution with methods like conjugate gradient and generally easier to precondition and solve than other types of linear systems. The solution of this system is still expensive in our implementation as we have not developed an effective preconditioner for it and have not made significant effort to optimize our routines. We leave the problem of finding a good preconditioner for future work. Solving the linear systems dominates the cost of our time evolution. The algorithm compares favorably with [25], which has an indefinite system that we have observed to converge much slower in practice.

Acknowledgments

Research supported in part by a Packard Foundation Fellowship, an Okawa Foundation Research Grant, ONR N0014-06-1-0393, ONR N0014-06-1-0505, ONR N0014-05-1-0479 for a computing cluster, NIH U54-GM072970, NSF ACI-0323866, NSF IIS-0326388, and King Abdullah University of Science and Technology (KAUST) 42959. C.S. was supported in part by a Stanford Graduate Fellowship.

References

- [1] S.V. Apte, M. Martin, and N.A. Patankar. A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows. J. Comput. Phys., 228(8):2712–2738, 2009.
- [2] F.P.T. Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. Intl. J. for Num. Meth. in Fluids, 35(7):743–761, 2001.
- [3] K.J. Bathe and H. Zhang. Finite element developments for general fluid flows with structural interactions. International Journal for Numerical Methods in Engineering, 60(1), 2004.
- [4] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. ACM Trans. Graph. (SIGGRAPH Proc.), 26(3):100, 2007.
- [5] J. Bonet and R. Wood. Nonlinear continuum mechanics for finite element analysis. Cambridge University Press, Cambridge, 1997.
- [6] T.M. Burton and J.K. Eaton. Analysis of a fractional-step method on overset grids. J. of Comput. Phys., 177(2):336–364, 2002.
- [7] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. Comp. Meth. Appl. Mech. Engng., 194(42-44), 2005.
- [8] A. Chorin. A numerical method for solving incompressible viscous flow problems. J. Comput. Phys., 2:12–26, 1967.
- [9] D. Dos Santos, N. Miguel, J.F. Gerbeau, and J.F. Bourgat. A partitioned fluid-structure algorithm for elastic thin valves with contact. Computer Methods in Applied Mechanics and Engineering, 2006.
- [10] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. Computer methods in applied mechanics and engineering, 157(1-2):95–114, 1998.
- [11] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. J. Comput. Phys., 176:205–227, 2002.
- [12] B.E. Griffith, R.D. Hornung, D.M. McQueen, and C.S. Peskin. An adaptive, formally second order accurate version of the immersed boundary method. J. of Comput. Phys., 223(1):10–49, 2007.
- [13] B.E. Griffith and C.S. Peskin. On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems. J. Comput. Phys., 208(1):75–105, 2005.
- [14] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. ACM Trans. Graph. (SIGGRAPH Proc.), 24(3):973–981, 2005.
- [15] J. Happel and H. Brenner. Low Reynolds number hydrodynamics: with special applications to particulate media. Kluwer Academic Print on Demand, 1991.

- [16] F. Harlow and J. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. Phys. Fluids, 8:2182–2189, 1965.
- [17] W.D. Henshaw. A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. J. of Comput. Phys., 113(1):13–25, 1994.
- [18] T. J. R. Hughes. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover, 2000.
- [19] SR Idelsohn, J. Marti, A. Limache, and E. Onate. Unified lagrangian formulation for elastic solids and incompressible fluids: Application to fluid–structure interaction problems via the PFEM. Computer Methods in Applied Mechanics and Engineering, 197(19-20):1762–1776, 2008.
- [20] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In Proc. of the ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 131–140, 2004.
- [21] N. Molino, R. Bridson, J. Teran, and R. Fedkiw. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In 12th Int. Meshing Roundtable, pages 103–114, 2003.
- [22] Y.T. Ng, C. Min, and F. Gibou. An efficient fluid-solid coupling algorithm for single-phase flows. J. of Comput. Phys., 228(23):8807–8829, 2009.
- [23] C. Peskin. Numerical analysis of blood flow in the heart. J. Comput. Phys., 25:220–252, 1977.
- [24] A. Robinson-Mosher, R.E. English, and R. Fedkiw. Accurate tangential velocities for solid fluid coupling. In Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 227–236, 2009.
- [25] A. Robinson-Mosher, T. Shinar, J.T. Grétarsson, J. Su, and R. Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. ACM Trans. on Graphics, 27(3):46:1–46:9, August 2008.
- [26] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. An unconditionally stable MacCormack method. J. of Sci. Comp., 35(2):350–371, 2008.
- [27] X. Shi and S.P. Lim. A LBM–DLM/FD method for 3D fluid–structure interactions. J. Comput. Phys., 226(2):2028–2043, 2007.
- [28] T. Shinar, C. Schroeder, and R. Fedkiw. Two-way coupling of rigid and deformable bodies. In SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 95–103, 2008.
- [29] H. Wang and T. Belytschko. Fluid-structure interaction by the discontinuous-galerkin method for large deformations. International Journal for Numerical Methods in Engineering, 77(1), 2009.
- [30] H. Wang, J. Chessa, W.K. Liu, and T. Belytschko. The immersed/fictitious element method for fluid–structure interaction: Volumetric consistency, compressibility and thin members. Int. J. Numer. Meth. Engng, 74:32–55, 2008.
- [31] Z. Yu. A DLM/FD method for fluid/flexible-body interactions. J. Comput. Phys., 207(1):1–27, 2005.
- [32] H. Zhao, J.B. Freund, and R.D. Moser. A fixed-mesh method for incompressible flow–structure systems with finite solid deformations. J. Comput. Phys., 2007.