Journal of Computational Physics (2021)

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Boundary pressure projection for partitioned solution of fluid-structure interaction with incompressible Dirichlet fluid domains

Muzaffer Akbay[a], Craig Schroeder[a], Tamar Shinar[a,*]

[a]*Computer Science and Engineering, University of California, Riverside, 351 Winston Chung Hall, Riverside, CA 92521-0429*

ARTICLE INFO

Article history:

ABSTRACT

Partitioned solutions to fluid-structure interaction problems often employ a Dirichlet-Neumann decomposition, where the fluid equations are solved subject to Dirichlet boundary conditions on velocity from the structure, and the structure equations are solved subject to forces from the fluid. In some scenarios, such as an elastic balloon filling with air, an incompressible fluid domain may have pure Dirichlet boundary conditions, leading to two related issues which have been described as the incompressibility dilemma. First, the Dirichlet boundary conditions must satisfy the incompressibility constraint for a solution to exist. However, the structure solver is unaware of this constraint and may supply the fluid solver with incompatible velocities. Second, the constant fluid pressure mode lies in the null space of the fluid pressure solver, but must be determined to apply to the structure. Previously proposed solutions to the incompressibility dilemma have included modifying the fluid solver, the structure solver, or the Dirichlet-Neumann coupling interface between them. In this paper, we present a boundary pressure projection method which alleviates the incompatibility while maintaining the Dirichlet-Neumann structure of the decomposition and without modification of the fluid or solid solvers. Our method takes incompatible velocities from the structure solver and projects them to be compatible while in the process computing the constant pressure modes for the Dirichlet regions. The compatible velocities are then used as Dirichlet boundary conditions for the fluid while the constant pressure modes are added to the fluid-solver-computed pressures to be applied to the structure. The intermediate computation performed in the boundary pressure projection method is small, with the number of unknowns equal to the number of Dirichlet regions. We show that the boundary pressure projection method can be used to solve a variety of scenarios including inflation of an elastic balloon and the action of a hydraulic press. We also demonstrate the method on multiple coupled Dirichlet regions. The method offers a simple approach to overcome the incompressibility dilemma using a small intermediate computation that requires no additional knowledge of the black box fluid and solid solvers.

1. Introduction

Partitioned numerical schemes for fluid-structure interaction problems are appealing as they allow for the coupling of existing fluid and solid solvers. Partitioned schemes often employ a Dirichlet-Neumann decomposition, where the fluid equations are solved subject to Dirichlet boundary conditions on velocity from the structure, and the structure equations are solved subject to forces from the fluid [15, 31, 34, 13, 30, 2, 12]. When the fluid is incompressible, the velocity field must satisfy the divergence-free condition everywhere. Consequently, the net flow through the boundary of any fluid region must also be zero. If the solution of the fluid equations in a region is subject to pure Dirichlet boundary conditions, it is necessary that the boundary conditions satisfy the constraint of zero net flow. Otherwise, they are incompatible with the flow equations, and no solution exists. Examples of such a scenario include the inflation of an elastic balloon and the action of a hydraulic press. More generally, enclosed fluid regions can arise spontaneously in free boundary fluid-structure interaction problems.

While monolithic approaches to fluid-structure interaction problems naturally deal with enclosed fluid regions [6, 38, 40, 7, 37, 20, 24, 10], partitioned schemes with Dirichlet-Neumann decomposition require special treatment of this case for robustness [29, 2]. In the latter approach, the structure is unaware of the incompressibility constraint on the fluid and will generally supply the fluid with incompatible velocities. Relatedly, the constant pressure mode in the fluid is not typically determined by the fluid solver, as it is in the null space of the pressure solver. However, this constant pressure mode should be applied to the structure in the coupling.

This problem has been described by Küttler et al. [29] as the *incompressibility dilemma*, and subsequently noted in several works [7, 6, 8, 2, 25, 27, 26, 32, 17, 9, 14, 33, 10]. Küttler et al. [29] proposed several approaches to addressing the dilemma: modifying the structure solver to incorporate the constraint, modifying the fluid solver to relax the constraint, or reversing the Dirichlet-Neumann decomposition between the solvers for the constant pressure mode. In the first solution method, the boundary constraint is incorporated into the structure equations and solver, requiring their modification. Note that the fluid-structure interface may comprise only a subset of the boundary of the enclosed fluid region, implying that the structure cannot independently determine the constraint on the interfacial velocities. In the second approach, artificial compressibility [11, 36, 23, 9] is incorporated into the fluid equations to eliminate the issue of incompatibility while iterating the solution in pseudo-time toward an incompressible state. The third solution approach suggested by [29] is to replace one mode of the Dirichlet-Neumann decomposition with a Neumann-Dirichlet decomposition where the structure imposes forces on the fluid and the fluid imposes interfacial displacements on the structure.

In a similar vein as [29], [41, 42, 21] addressed the incompressibility dilemma by introducing the constraint into the fluid solver, solid solver, or both and showed that the Lagrange multiplier associated with the compatibility constraint on the solid was the excess fluid pressure [42, 21]. The incompressibility dilemma was also shown to be naturally handled in a partitioned scheme employing Robin transmission conditions [3, 17, 16, 18]. For the similar case of the partitioned simulation of lid-driven flow in a cavity, the dilemma was addressed by removing the Dirichlet boundary condition on the velocity along a small subset of the surface [43, 19, 26, 33].

In this work, we present a boundary pressure projection (BPP) method which alleviates boundary velocity incompatibility and computes the constant pressure mode, while maintaining the Dirichlet-Neumann structure of the decomposition and without requiring modification of the fluid or solid solvers. Our method takes potentially incompatible velocities from the structure solver, projects them to be compatible while in the process computing the constant pressure mode for the Dirichlet region. The compatible velocities are then used as Dirichlet boundary conditions for the fluid while the constant pressure mode is added to the fluid solver pressures to be applied to the structure. This idea naturally extends to multiple coupled Dirichlet fluid regions, by associating a separate constant pressure mode with each Dirichlet fluid region. The intermediate computation performed in the BPP method is small, with the number of unknowns equal to the number of Dirichlet regions. We demonstrate our approach in a variety of scenarios including inflation of an elastic balloon and action of a hydraulic press. We also demonstrate the method on multiple, coupled Dirichlet regions. Our approach overcomes the incompressibility dilemma using a small intermediate computation while respecting the Dirichlet-Neumann decomposition and black-box nature of the solvers. The scheme is summarized in Figure 1, and an illustrative example is shown in Figure 2.

*Corresponding authors: email: shinar@cs.ucr.edu

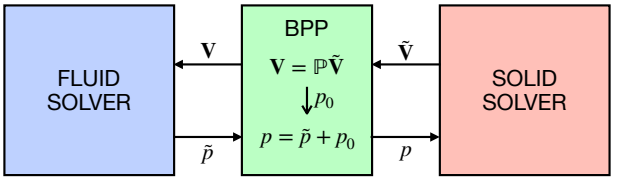


Fig. 1: Schematic representation of the boundary pressure projection method. A small intermediate computation projects the solid velocities $\tilde{\mathbf{V}}$ to be compatible with the fluid incompressibility constraint, while determining the constant fluid pressure mode p_0 as the associated Lagrange multiplier. Compatible velocities are passed on to the fluid solver, while the constant pressure mode is added to the fluid pressures and is passed to the solid solver.

We note that we first introduced a boundary pressure projection approach in [1]. Here, we derive an improved formulation that applies to a broader range of mass density ratios and extend the BPP to multiple enclosed regions. In [1], the constant pressure mode was determined by considering the resulting change of momentum of the *fluid*. While this works for scenarios where the solid density was close to the fluid density, it fails in cases where the solid becomes more massive, as we demonstrate in this work. Here, we correct the formulation to use the change of momentum of the *solid* to determine the correct magnitude of the constant pressure mode. This allows the fluid to support a solid with large relative density. We further formulate the approach for the case of underrelaxed partitioned iterations. Finally, we extend the BPP to multiple enclosed regions, and study its behavior for multiple coupled enclosed regions separated by a thin shell or rigid body. We demonstrate the method in a strongly-coupled partitioned framework, where the fluid and solid solvers are iterated to convergence in every time step; the boundary pressure projection can also be incorporated into weakly-coupled partitioned frameworks where the fluid and solid solvers are not iterated to convergence.

2. Equations and Algorithm

2.1. Fluid Equations

The fluid equations are the incompressible Navier-Stokes equations

$$\rho\left(\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{f}, \quad \mathbf{x}\in\Omega \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where ρ is the fluid density, $\mathbf{u}$ is the velocity, p is the pressure, μ is the viscosity, and $\mathbf{f}$ the force density. The fluid is subject to Dirichlet boundary conditions

$$\mathbf{u} = \mathbf{u}_{bc}, \quad \mathbf{x}\in\partial\Omega. \tag{3}$$

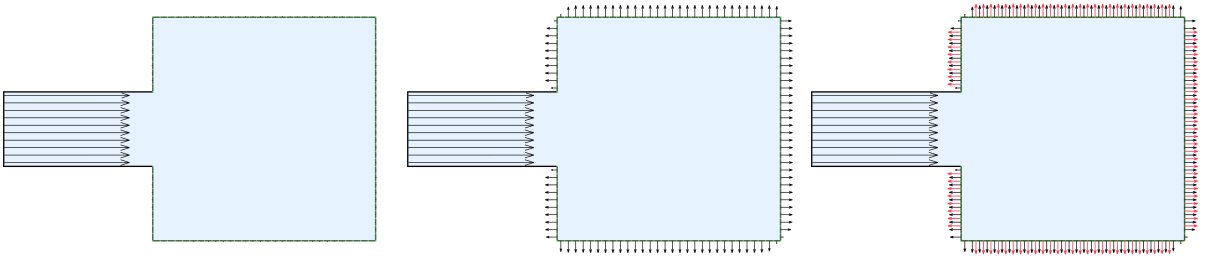


Fig. 2: The boundary pressure projection method demonstrated on a representative thin-shell example. Left: Before the BPP, the fluid boundary velocities (black) are incompatible as there is a net inflow from the source on the left. The BPP is used to project the fluid boundary velocities and to determine the constant pressure mode to be applied to the solid. Center: Fluid boundary velocities (black) after compatibility projection. Right: Solid velocities (red) after application of the constant pressure mode.

2.2. Structure Equations

We consider both elastic and rigid structures. The equations of motion for an elastic structure are given by

$$\rho_s \eta_{tt} = \nabla \cdot \sigma_s + \mathbf{f}, \tag{4}$$

where ρ_s is the density, η is the structure displacement, σ_s is the stress tensor and $\mathbf{f}$ accounts for external forces, including gravity and fluid forces. In our examples, we use the neo-Hookean constitutive model for solid elasticity for all of the volumetric deformable solids. The two-dimensional elastic solids are discretized into a triangular mesh and simulated using a finite volume method.

One-dimensional thin shells are discretized into segmented curves and simulated as a mass-spring system. Stretching springs are applied between neighboring vertices. Bending springs connect pairs of nodes two apart (every other node), resulting in a mesh connecting all odd nodes and a mesh connecting all even nodes. Bending springs are weaker than the stretching springs and tend to resist bending in the solid. Both spring forces take the form

$$\mathbf{f}_1 = -\mathbf{f}_2 = -\frac{k}{\ell_0}\left(\frac{\ell}{\ell_0} - 1\right)\mathbf{w} - \frac{b}{\ell_0}(\mathbf{w} \cdot (\mathbf{v}_1 - \mathbf{v}_2))\mathbf{w}, \tag{5}$$

where $\ell = \|\mathbf{x}_1 - \mathbf{x}_2\|$ is the current spring length, ℓ_0 is the spring length at the beginning of the simulation (the rest length), and $\mathbf{w} = \frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$ is the spring direction. Here k is elastic stiffness and b is the strength of the damping force. The BPP is agnostic to these choices; other approaches may be used as long as the interfaces support the Dirichlet-Neumann decomposition in the partitioned scheme.

For rigid bodies, the equations of motion are

$$\frac{d}{dt}\begin{pmatrix}\mathbf{P}(t)\\ \mathbf{L}(t)\end{pmatrix} = \begin{pmatrix}\mathbf{f}(t)\\ \tau(t)\end{pmatrix}, \tag{6}$$

where $\mathbf{P}$ is the linear momentum of the body, $\mathbf{L}$ is the angular momentum of the body, $\mathbf{f}$ is the net force on the body, and τ is the net torque on the body. The position and orientation of the body are updated as

$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}(t)\\ R(t)\end{pmatrix} = \begin{pmatrix}\mathbf{v}(t)\\ \omega^\star(t)R(t)\end{pmatrix}, \tag{7}$$

where $\mathbf{x}$ is the position of the rigid body center of mass, $R(t)$ is the orientation of the body, I_0 is the body space inertia tensor, $I(t) = R(t)I_0R(t)^T$ is the world space inertia tensor, M is the mass, $\mathbf{v} = M^{-1}\mathbf{P}(t)$ is the velocity of the rigid body center of mass, and $\omega(t) = I(t)^{-1}\mathbf{L}(t)$ is the angular velocity of the body.

2.3. Black-box Interface

We regard the fluid and solid solvers as black boxes. We denote the solid solver by

$$(\mathbf{X}, \tilde{\mathbf{V}}) = S(\mathbf{f}_{bc}, t), \tag{8}$$

where $\mathbf{f}_{bc}$ contains the forces on the solid degrees of freedom, and $\mathbf{X}$ and $\tilde{\mathbf{V}}$ are the position and velocity degrees of freedom of the solid. Similarly, the fluid solver is given by

$$(\mathbf{U}, \tilde{p}) = F(\mathbf{U}_{bc}, t), \tag{9}$$

where $\mathbf{U}_{bc}$ are the velocity Dirichlet boundary conditions applied to the fluid, $\mathbf{U}$ is the fluid velocity, and $\tilde{p}$ is the fluid pressure. Note that there is an implicit dependence of F on the location of the solid since the location of the solid defines in part the fluid region. For the sake of exposition, we take the convention that $\mathbf{U}$ includes its boundary conditions $\mathbf{U}_{bc}$ and that all fluid velocity variables ($\mathbf{U}$, $\mathbf{U}_{bc}$, and the fluid sources $\mathbf{U}_s$) are padded with zeros to have the same representation.

These interfaces for the solid and fluid solvers support the Dirichlet-Neumann coupling of the solvers, where the fluid accepts as input the Dirichlet boundary conditions for velocity from the solid, and the solid accepts as input the fluid stress-based forces. Coupling between the fluid and solid solvers is mediated by the interpolation operator W, which interpolates solid velocity degrees of freedom to the fluid boundary conditions

$$\mathbf{U}_{bc} = W\tilde{\mathbf{V}} + \mathbf{U}_s, \tag{10}$$

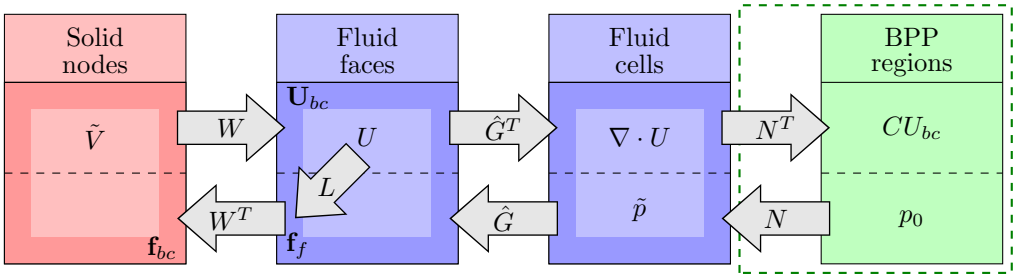


Fig. 3: Solid, fluid, and BPP quantities are located at nodes, faces and cells, and regions, respectively. The operators W, $\hat{G}$, N, L, and $C = N^T\hat{G}^T$ transfer information between the solid, fluid, and BPP representations. The solid and fluid boxes and the operators mapping between them correspond to a standard partitioned solver. Augmenting the solver with the BPP involves the addition of the components in the dashed box.

where $\mathbf{U}_s$ represents any prescribed fluid velocities at portions of the fluid boundary that are not directly coupled to solids. The force boundary conditions for the solid are computed with

$$\mathbf{f}_{bc} = W^T(\hat{G}\tilde{p} + L\mathbf{U}), \tag{11}$$

where $\hat{G}\tilde{p} + L\mathbf{U}$ is the boundary force computed from the fluid solver at the fluid velocity degrees of freedom. The operator W^T distributes the total force to the solid degrees of freedom. The operators W, $\hat{G}$, and L encapsulate discretization details for the solid-fluid interface, and are depicted in the left side of Fig. 3. Iterating (8), (10), (9), and (11) yields a standard partitioned solver.

In the case of a pure Dirichlet fluid region, the fluid solver typically has a null space in the pressure equations. Hence, $\mathbf{U}_{bc}$ in Eq. (10) must satisfy the compatibility condition arising from the incompressibility constraint. Furthermore, for a pure Dirichlet fluid region, Eq. (11) does not account for all fluid forces on the solid. We address this in the next section.

2.4. *Fluid Boundary Constraint*

The pressure field determined by the fluid solver is only defined up to a scalar in each region due to the pressure null space of the fluid discretization. However, the pressure for the corresponding fluid-structure interaction problem does not contain a null space. In the fluid-structure interaction problem, the missing constant pressure p_0 is the Lagrange multiplier associated with the fluid constraint

$$\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}\, dS = 0 \tag{12}$$

over each closed surface $\partial\Omega$, where $\mathbf{n}$ is the outward unit normal to the surface. We denote the discretized boundary constraints in Eq. (12) as

$$C\mathbf{U}_{bc} = \mathbf{0}. \tag{13}$$

Correspondingly, we modify the solid boundary conditions in Eq. (11) accordingly,

$$\mathbf{f}_{bc} = W^T(\hat{G}\tilde{p} + L\mathbf{U} + C^T p_0). \tag{14}$$

Note that C and p_0 contain exactly one row for each separate Dirichlet fluid region. A natural choice for C is

$$C = N^T\hat{G}^T, \tag{15}$$

where each column of N is the null space vector for the pressure Poisson matrix for one of the fluid regions. The entries of N are either 0 or 1, and the matrix N distributes the components of p_0 to the pressures in the appropriate regions. With this choice, Eq. (14) becomes

$$\mathbf{f}_{bc} = W^T(\hat{G}(\tilde{p} + Np_0) + L\mathbf{U}). \tag{16}$$

The addition of the BPP and the associated operators to the standard partitioned solver is depicted in the dashed box in Fig. 3.

2.5. Boundary Pressure Projection

The fluid solver should be called with boundary conditions that satisfy Eq. (13). Therefore, we modify the fluid boundary conditions in Eq. (10) accordingly,

$$\mathbf{U}_{bc} = W\mathbf{V} + \mathbf{U}_s, , \tag{17}$$

where the modified solid velocities $\mathbf{V}$ satisfy

$$C\mathbf{U}_{bc} = C(W\mathbf{V} + \mathbf{U}_s) = \mathbf{0}. \tag{18}$$

The solid velocities $\mathbf{V}$ are computed by correcting the output of the solid solver $(\mathbf{X}, \tilde{\mathbf{V}}) = S(\mathbf{f}_{bc}, t^n)$, which is a black box and unknown to us. If S predicts a $\tilde{\mathbf{V}} = S_2(\tilde{\mathbf{f}}_{bc}, t^n)$ that does not satisfy the constraint, then we must predict a modified $\mathbf{f}_{bc} = \tilde{\mathbf{f}}_{bc} + \Delta\mathbf{f}_{bc}$ so that Eq. (18) holds. We may approximate

$$\mathbf{V} = S_2(\tilde{\mathbf{f}}_{bc} + \Delta\mathbf{f}_{bc}, t^n) = S_2(\tilde{\mathbf{f}}_{bc}, t^n) + \triangle t M^{-1}\Delta\mathbf{f}_{bc} = \tilde{\mathbf{V}} + \triangle t M^{-1}\Delta\mathbf{f}_{bc}, \tag{19}$$

since S is consistent with forward Euler. This $\Delta\mathbf{f}_{bc}$ should be obtained from a change to $p_0 = \tilde{p}_0 + \Delta p_0$, since this is the Lagrange multiplier that enforces Eq. (18). Then

$$\Delta\mathbf{f}_{bc} = \mathbf{f}_{bc} - \tilde{\mathbf{f}}_{bc} = W^T(\tilde{G}\tilde{p} + L\mathbf{U} + C^T p_0) - W^T(\tilde{G}\tilde{p} + L\mathbf{U} + C^T\tilde{p}_0) = W^T C^T \Delta p_0. \tag{20}$$

Combining Eqs. (18), (19), and (20) we get the linear system

$$\triangle t CWM^{-1}W^T C^T \Delta p_0 = -C(W\tilde{\mathbf{V}} + \mathbf{U}_s). \tag{21}$$

This system is $m \times m$, where m is the number of Dirichlet fluid regions subject to the boundary constraint. Hence, it will typically remain small and inexpensive to solve.

We note that we differ from [1], which instead solved

$$\triangle t C M_f^{-1} C^T \Delta p_0 = -C(W\tilde{\mathbf{V}} + \mathbf{U}_s) \tag{22}$$

using the fluid inverse mass M_f^{-1} as an approximation for the effective inverse mass imposed by the solid $WM^{-1}W^T$. While both formulations work for relative densities close to one, Eq. 22 fails when the relative density of the solid is large. This is because the impulse needed to accelerate the fluid is smaller than the impulse needed to accelerate a heavier solid. Hence the formulation with M_f underestimates the impulse. Conversely, if the fluid was heavier, the use of M_f rather than M would overestimate the impulse. The formulation in Eq. (21) is more accurate, because the force due to the constant pressure mode needs to accelerate the *solid* to be compatible. A similar correction was made in [37].

Consistent with our construction above, the association of the Lagrange multiplier enforcing compatibility on the solid velocities with excess fluid pressure was proven in the continuous formulation of a Stokes flow coupled to a Kirchhoff-Love shell model [42], which modified the structural solver to incorporate the constraint.

2.5.1. Boundary Pressure Projection Operator

Solving Eq. (21) for the Lagrange multiplier $\triangle p_0^{k+1}$ and substituting it back into Eqs. (20) and (19) while assuming $\mathbf{U}_s = \mathbf{0}$, Eq. (19) can be written as

$$\mathbf{V} = \mathbb{P}\tilde{\mathbf{V}}, \tag{23}$$

where

$$\mathbb{P} = I - M^{-1}W^T C^T (CWM^{-1}W^T C^T)^{-1} CW \tag{24}$$

$$= I - M^{-1}C_s^T (C_s M^{-1} C_s^T)^{-1} C_s, \tag{25}$$

where $C_s = CW$ is the constraint expressed on the solid degrees of freedom. Thus $\mathbb{P}$ is the mass-orthogonal projection operator that projects solid velocities to be compatible with the fluid boundary velocity constraint. We note that the projection can also be written directly on the fluid boundary velocities. Specifically, the fluid velocity is

$$\mathbf{U}_{bc} = \mathbb{P}_f \tilde{\mathbf{U}}_{bc}, \tag{26}$$

Algorithm 1 Single time step $(t^n \rightarrow t^{n+1})$.

$\mathbf{p}^0 = \mathbf{p}^n$ {Initialize fluid pressure field}
2: $p_0^0 = p_0^n$ {Initialize constant pressure}
for $k = 0, 1, 2, \ldots, k_{\max}$ **do**
4: {**Solid Solve**}
$\mathbf{f}_{bc}^k = W^T(\hat{G}p^k + L\mathbf{U}^k + C^T p_0^k)$ {Compute interface forces}
6: $(\mathbf{X}^{k+1}, \tilde{\mathbf{V}}^{k+1}) = S(\mathbf{f}_{bc}^k, t^{n+1})$ {Call solid solver}
{Compute underrelaxation factor ω}
8: $\mathbf{X}^{k+1} \leftarrow \omega\mathbf{X}^{k+1} + (1-\omega)\mathbf{X}^k$ {Underrelaxation of solid positions}
$\tilde{\mathbf{V}}^{k+1} \leftarrow \omega\tilde{\mathbf{V}}^{k+1} + (1-\omega)\mathbf{V}^k$ {Underrelaxation of solid velocities}
10: {Compute interpolation operator W}
{**Boundary Pressure Projection**}
12: $\left(\triangle t CWM^{-1}W^TC^T\right)\triangle p_0^{k+1} = -CW\tilde{\mathbf{V}}^{k+1}$ {Solve for $\triangle p_0^{k+1}$}
$p_0^{k+1} = p_0^k + \triangle p_0^{k+1}$ {Update constant pressure}
14: $\mathbf{V}^{k+1} = \tilde{\mathbf{V}}^{k+1} + \Delta t M^{-1}W^TC^T\triangle p_0^{k+1}$ {Correct solid velocity}
{**Fluid Solve**}
16: $\mathbf{U}_{bc}^{k+1} = W\mathbf{V}^{k+1}$ {Compute interface velocity}
$(\mathbf{U}^{k+1}, \mathbf{p}^{k+1}) = F(\mathbf{U}_{bc}^{k+1}, t^{n+1})$ {Call fluid solver}
18: {Break if converged}
end for

where

$$\mathbb{P}_f = I - M_f^{-1}C^T(CM_f^{-1}C^T)^{-1}C. \tag{27}$$

Here, $\tilde{\mathbf{U}}_{bc} = W\tilde{\mathbf{V}}$, and $M_f^{-1} = WM^{-1}W^T$ defines the effective fluid mass. $\mathbb{P}_f$ is a mass-orthogonal projection with respect to M_f.

2.6. Algorithm

We use a partitioned approach to solve the fluid-structure interaction problem. In each time step, we iteratively apply the fluid and solid black-box solvers. Algorithm 1 outlines the steps in the basic algorithm. We begin the time step using the previous fluid pressure (Line 1) and p_0 (Line 2). Each iteration begins by computing the interaction forces for the solid solver (Eq. (14); Line 5). We then call the black-box solid solver (Eq. (8); Line 6) to get candidate positions and velocities for the solids. To aid convergence, we use underrelaxation on the position and velocity of the solids. We compute the underrelaxation parameter adaptively using the method of Aitken [30]. After computing the relaxation factor (Line 7), we apply it to the solid state (Lines 8 and 9).

Moving the solid changes the location of the interface between the solid and the fluid and also the fluid domain itself. The next step is to recompute the fluid domain, determine boundary condition locations and coupling degrees of freedom for the fluid solver, and recompute the coupling interpolation weights W (Line 10; See Section 3.3). Next, we apply BPP projection (Eq. (21); Lines 12, 13, and 14) so that solid velocities respect the compatibility condition for fluid incompressibility. Finally, we interpolate the fluid boundary conditions from the solid (Eq. (17); Line 16) and call the black-box fluid solver (Eq. (9); Line 17). Having updated the solid and the fluid, we have a candidate configuration for time t^{n+1}. Next, we evaluate the convergence criteria (Line 18; See Section 4) and, if appropriate, terminate the iteration.

3. Discretization

3.1. Fluid Solver

In our examples, we discretize the fluid equations on a standard, uniform MAC grid [22] and solve them using the fractional-step projection method [11] as follows. First, we advect the fluid using semi-Lagrangian advection

$$\frac{\mathbf{u}^{(1)} - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^n \cdot \nabla)\mathbf{u}^n = 0 \tag{28}$$

and apply body forces (gravity)

$$\frac{\mathbf{u}^{(2)} - \mathbf{u}^{(1)}}{\Delta t} = \frac{\mathbf{f}^n}{\rho}. \tag{29}$$

Since we use a Laplacian formulation of implicit viscosity, we must ensure that the velocity field is incompressible, which we enforce using a pressure projection step

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla \tilde{p}^{(1)} \right) = \nabla \cdot \mathbf{u}^{(2)}, \qquad \frac{\mathbf{u}^{(3)} - \mathbf{u}^{(2)}}{\Delta t} = -\frac{1}{\rho} \nabla \tilde{p}^{(1)}. \tag{30}$$

Next, we apply implicit viscosity

$$\frac{\mathbf{u}^{(4)} - \mathbf{u}^{(3)}}{\Delta t} = \frac{\mu}{\rho} \nabla^2 \mathbf{u}^{(4)}. \tag{31}$$

Although viscosity (analytically) preserves the incompressibility of the velocity, this is not true discretely. To correct this, we repeat the pressure projection to guarantee a divergence-free velocity field at the end of the fluid solver

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla \tilde{p}^{(2)} \right) = \nabla \cdot \mathbf{u}^{(4)} \qquad \frac{\mathbf{u}^{n+1} - \mathbf{u}^{(4)}}{\Delta t} = -\frac{1}{\rho} \nabla \tilde{p}^{(2)}. \tag{32}$$

We compute the fluid pressure as the sum of the pressures from the two projection steps

$$\tilde{p}^{n+1} = \tilde{p}^{(1)} + \tilde{p}^{(2)}. \tag{33}$$

Although we have chosen to use a voxelized fluid solver as the black-box solver, we are able to obtain first order convergence in most cases. We do, however, observe convergence difficulties due to the voxelized (and thus discontinuous) nature of the black-box fluid solver. We note that this choice of fluid solver was made for convenience; a non-voxelized or higher order solver could be used in its place [35, 20]. A second order accurate FSI solver should be possible by using second order accurate black box solvers for the solid and fluid as well as corresponding second order accurate discretizations of the operators used to couple them ($\hat{G}$, L, W, N, and C). We do not pursue higher order accuracy further in this paper.

3.2. Solids Solver

We use a semi-implicit solver [39] to evolve the deformable solids and rigid bodies. The solver is first order and uses a finite volume discretization for volumetric forces. We use collocated velocity degrees of freedom at the vertices of elements for the deformable solver.

3.3. Interface Operators

We summarize our discretization of the interface in Fig. 4. For the purposes of coupling, the solids are represented by their boundary curves. We first classify all cell centers and nodes of the fluid grid as inside the fluid or outside the fluid. We use these inside/outside flags for all of the calculations that follow, which ensures that all decisions are made in a consistent way. We note that the inside/outside classification is done using visibility checks, which allows us to handle thin shells and degenerate configurations.

Next, we identify the fluid faces that are boundary conditions for the fluid pressure and viscosity discretizations. If a pair of adjacent cell *centers* is cut by the solid boundary, then the MAC face between them is a coupled face in the *normal* direction. If a pair of adjacent *nodes* is cut by the solid boundary, then the MAC face between them is a coupled face in the *tangential* direction. In the case of slip boundary conditions, tangential coupling faces are ignored. All remaining MAC faces are fully inside (fluid degrees of freedom) or fully outside. Pressure degrees of freedom are located at cell centers that are marked inside.

A MAC face is a coupling face if segments connecting the adjacent nodes or cell centers are intersected by the solid boundary. To interpolate solid velocity to this coupling MAC face, we evaluate the solid's velocity at these intersection points. If a MAC face is both normally and tangentially coupled, we will compute two solid velocity estimates for the face, which we average. This interpolation process defines the operator W. Note that this simple procedure applies equally to rigid bodies and both volumetric and thin shell deformable bodies.

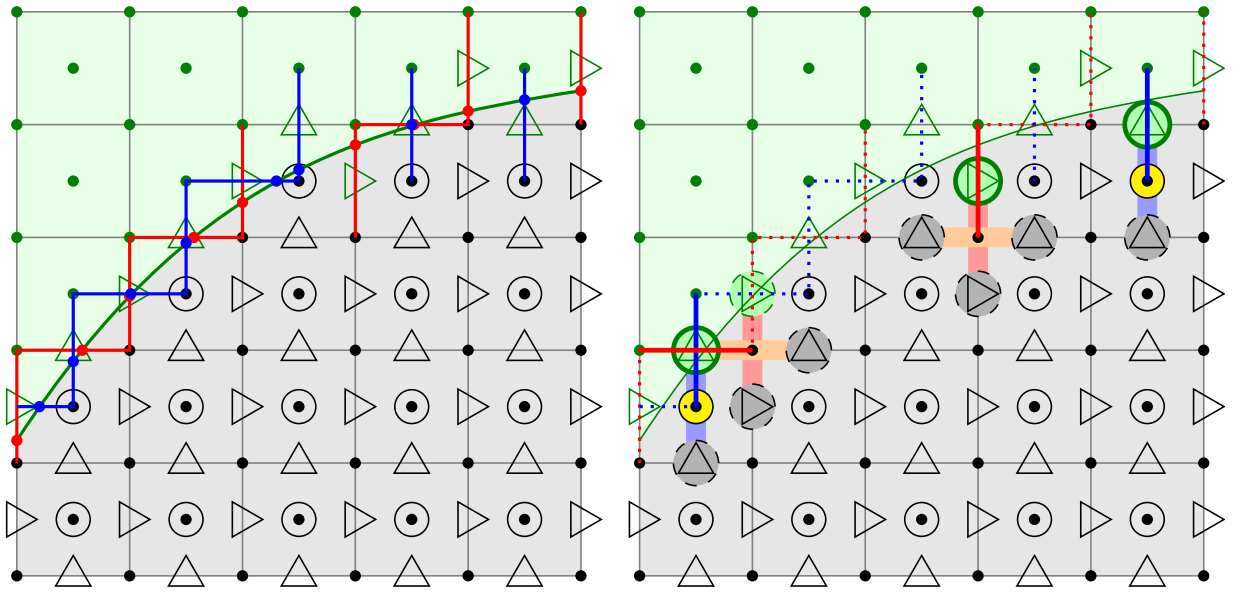


Fig. 4: **(Left) Discretization of the interface.** The boundary of a solid (—) divides space into solid (light green) and fluid (gray). Cell centers and nodes are classified as inside (●) or outside (●) the fluid. Adjacent *centers* of opposite color are connected with (—) and indicate coupling in the *normal* direction. The face at the midpoint is a coupling fluid face (▷△). Adjacent *nodes* of opposite color are connected with (—) and indicate coupling in the *tangential* direction. The face at the midpoint is a coupling fluid face (▷△) when no-slip boundary conditions are being applied. Faces inside the fluid region that are not coupling faces are degrees of freedom for the fluid solver (▷△). Pressure degrees of freedom are located at cell centers that are inside the fluid (◯). The locations where the segments (—) and (—) intersect the solid boundary (—) are the coupling locations (● ●) for the solid for both deformable and rigid solids. **(Right) Stencils for L and $\hat{G}$.** The L operator calculates the boundary force due to viscosity at coupling faces. Stencils are shown for the three coupling faces marked with (◯). The L stencil is comprised of two types of stencils. Coupling in the normal direction uses either a $\frac{\partial u}{\partial x}$ or $\frac{\partial v}{\partial y}$ stencil ( ). Coupling in the tangential direction uses both $\frac{\partial u}{\partial y}$ ( ) and $\frac{\partial v}{\partial x}$ ( ) stencils. Faces coupled in both the normal and tangential direction receive both stencils. The $\hat{G}$ operator applies fluid pressures (◯) to the coupling faces (◯).

The $\hat{G}$ operator takes fluid pressures and returns the associated forces on coupling faces. We use Δx as the area over which the pressure is applied. Note that pressures only couple in the normal direction. A face is normally coupled only if a neighboring cell is inside, which guarantees that it will receive a pressure contribution.

The L operator takes fluid velocities and calculates the forces at coupling faces due to viscosity. The L stencil is comprised of two types of stencils. Coupling in the *normal* direction requires a $\frac{\partial u}{\partial x}$ or $\frac{\partial v}{\partial y}$ stencil, which is naturally computed with central differencing at the cell center and applied to the face. Coupling in the *tangential* direction uses both $\frac{\partial u}{\partial y}$ and $\frac{\partial v}{\partial x}$ stencils, which are naturally computed by central differencing to the inside node and applied to the face. Faces coupled in both the normal and tangential direction receive both stencils. Under these definitions, we are guaranteed valid velocity data for the viscosity stencils. A normally coupled face must have an adjacent cell center inside the fluid, which similarly guarantees valid data at the adjacent faces. A tangentially coupled face has a neighboring node that is inside, which guarantees that the four faces adjacent to this node are either degrees of freedom or velocity boundary conditions and thus have valid velocity values.

In the case of thin shells, the L and $\hat{G}$ stencils on each side are computed independently as above. The stencils share only the velocities stored at coupling faces.

Note that any distribution/interpolation scheme for such transfers can be used, as this choice is orthogonal to BPP.

4. Examples

We take the time step size to be the minimum of the individually computed fluid and solid solver time step sizes $\Delta t = \min(\Delta t_f, \Delta t_s)$, unless otherwise noted. We use several convergence criteria for the subiterations, which we

Description	Criterion	Default value
Solid positions	$\max_p \|\mathbf{X}_p^{k+1} - \mathbf{X}_p^k\| \leq \tau_x \Delta x$	$\tau_x = 10^{-2}$
Solid velocities	$\max_p \|\mathbf{V}_p^{k+1} - \mathbf{V}_p^k\| \leq \tau_v$	$\tau_v = 10^{-3}$
Fluid velocities	$\max_f \|\mathbf{u}_f^{k+1} - \mathbf{u}_f^k\| \leq \tau_u$	$\tau_u = 10^{-3}$
Fluid pressures	$\max_i \|\tilde{p}_i^{k+1} - \tilde{p}_i^k\| \leq \tau_p$	$\tau_p = 10^{-3}$
BPP pressure (p_0)	$\max_r \|p_{0r}^{k+1} - p_{0r}^k\| \leq \tau_0$	$\tau_0 \approx 10^{-3} \max(p_0)$
Constraint error	$\max_r \frac{\|(C\tilde{U}_{bc})_r\|}{\text{vol}_r} \leq \tau_e$	$\tau_e = 10^{-3}$

Table 1: Summary of the subiteration convergence criteria.

summarize in Table 1. All of the convergence criteria are based on L_∞ error measures, which are calculated over solid vertices (p), MAC grid faces (f), MAC grid cells (i), and fluid regions (r). We use the default value in the table unless otherwise noted. The tolerance τ_0 was chosen as a power of ten that is approximately three orders of magnitude less than the typical value for p_0 for that simulation. A minimum of two iterations of the solver is always performed, since two iterates are required to evaluate most of the criteria.

Due to the voxelized nature of our fluid solver, cell crossings can cause significant discontinuities in pressure, which make convergence difficult when the solid boundary lies near the discontinuity. We address this by freezing the fluid interface (including W) when the solid positions change by less than some tolerance τ_m (measured in the same way as the solid position convergence criterion). We use the default value of $\tau_m = 10^{-8} \ll \tau_x$ so that the error introduced by this approximation is very small compared to the convergence error.

Aitken relaxation may compute a relaxation parameter ω that is arbitrarily large or small, even negative. We clamp $10^{-2} \leq \omega \leq 10$ unless otherwise noted. In particular, both underrelaxation and overrelaxation occur frequently in practice. We found the use of a dynamic relaxation parameter to significantly improve the convergence rate of our partitioned scheme, both by underrelaxing when needed to prevent divergence and overrelaxing where possible to accelerate the convergence rate. We do not perform relaxation during the first iteration since we only have one iterate available. During the second iteration, we use the relaxation parameter ω from the previous time step. Once three iterates are available, we are able to apply Aitken relaxation to compute ω.

4.1. Supported Rigid Body

We show an example of a single rigid body supported by an enclosed fluid, as illustrated in Fig. 5. The body is subject to gravitational forces and fluid forces. Therefore, the fluid pressure forces should exactly balance the force of gravity on the body, giving

$$-M\mathbf{g} + Ap_h\mathbf{n} = \mathbf{0}, \tag{34}$$

where M is the total mass of the body, A is the surface area of the interface, p_h is the fluid pressure at the interface, and $\mathbf{n}$ is the fluid region outward normal at the interface.

The rigid body has dimensions $0.6\,m \times 0.15\,m$, and is placed on the top of an enclosed, incompressible fluid domain of dimensions $0.6\,m \times 0.8\,m$. The fluid is inviscid and has a density of $1\,kg/m^2$. The solid density is $10^4\,kg/m^2$. The gravitational acceleration acting on both the solid and the fluid is $9.8\,m/s^2$ in the negative y-direction. Thus, the analytical solution for the interface pressure is $p_h = 14700\,N/m$. Since the rigid body is more dense than the fluid, this system is in an unstable equilibrium. If symmetry is broken (even numerically), the rigid body will rapidly fall to one side. We prevent this by eliminating the horizontal and rotational degrees of freedom from the rigid body. The rigid body remains free to translate in the vertical direction.

The fluid is simulated on a uniform grid of resolution 192×256. The solvers are called iteratively as in Algorithm (1) until the convergence criteria are met. In this example we use $\tau_0 = 1$. We ran the simulation until time $t = 3\,s$ with no underrelaxation, since it always converges in two iterations after the first time step.

The fluid solver uses a fractional step method [11], wherein the pressure is computed subject to pure Neumann boundary conditions. This leads to a linear system of equations for the pressure which has a null space of dimension

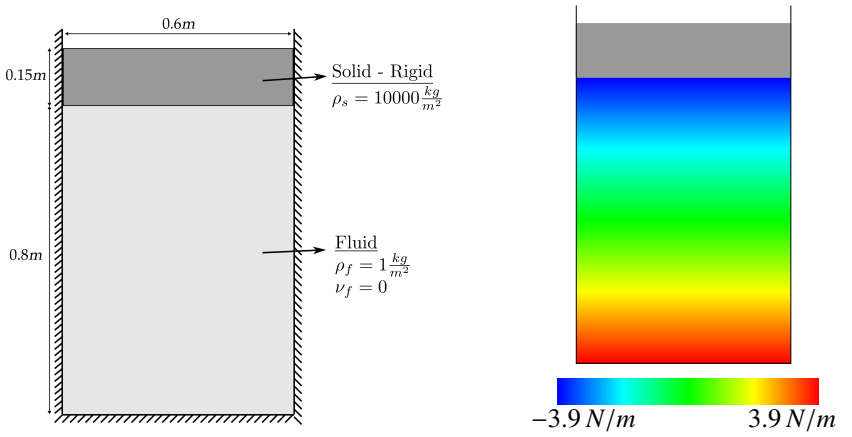


Fig. 5: (Left) Supported rigid solid example setup. (Right) Fluid solver pressures, $\tilde{p}$, for the supported rigid solid example. The fluid solver pressures are on the order of $10^{-4}p_0$. This is the steady state solution computed by the fluid solver, which does not see the constant pressure mode p_0, as it lies in the null space of the fluid pressure solver.

one, corresponding to the constant pressure mode. The fluid solver uses zero for this component of the pressure, and thus computes only the variation in the pressure field, $\tilde{p}$. In this case, the fluid solver gives cell pressures of $\tilde{p} \in [-3.90469, 3.90469]\,N/m$, which matches the analytical solution up to six digits. The fluid solver pressure field is shown in Fig. 5.

Unlike the fluid solution, the solid solution *does* depend on p_0. The BPP determines a value of $p_0 \approx 1.470390469\times 10^4$, computing a steady value of p_0 up to fluctuations of $10^{-15}p_0$. Combined with the fluid pressure values, this gives the analytically correct value of $p_h = 14700\,N/m$, up to the accuracy of the fluid solver. We note also that if the fluid solver returned an arbitrary nonzero value for p_0, the BPP would compensate for that value, determining the Δp_0 needed to achieve the correct constant pressure mode.

4.2. Supported Deformable Body

We study a similar validation example with a deformable solid. The collision with the fluid is computed only on the interface nodes of the solid, and will be propagated throughout the solid by the solid's constitutive model. The steady state solution should be close to the above rigid example,

$$-M\mathbf{g} + Ap_h\mathbf{n} \approx \mathbf{0}, \tag{35}$$

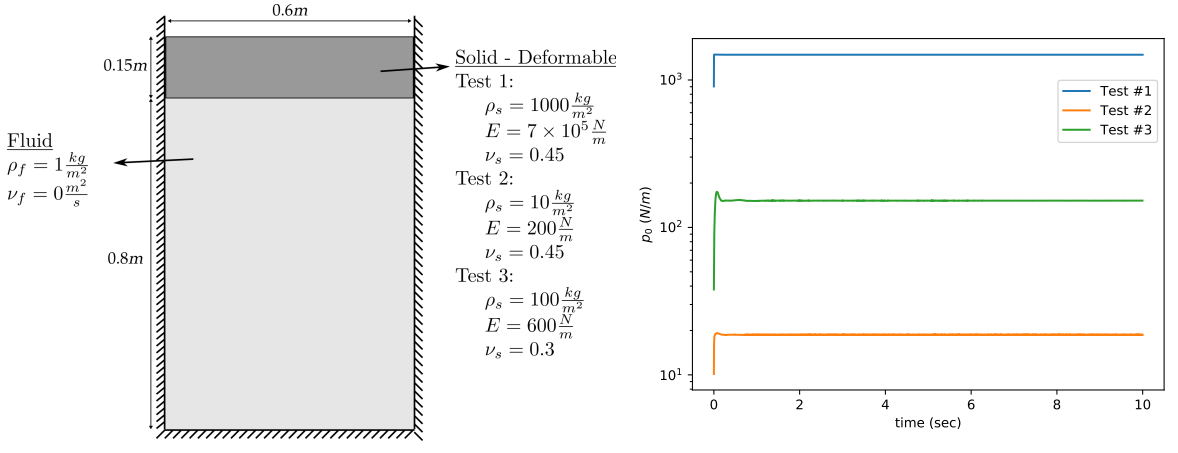


Fig. 6: **(Left)** Supported deformable solid example setup. **(Right)** Constant pressure mode p_0 throughout the simulation of the supported deformable body example (Test #1-3). Note that it takes time for the fluid to feel the full mass of the solid through the constitutive model.

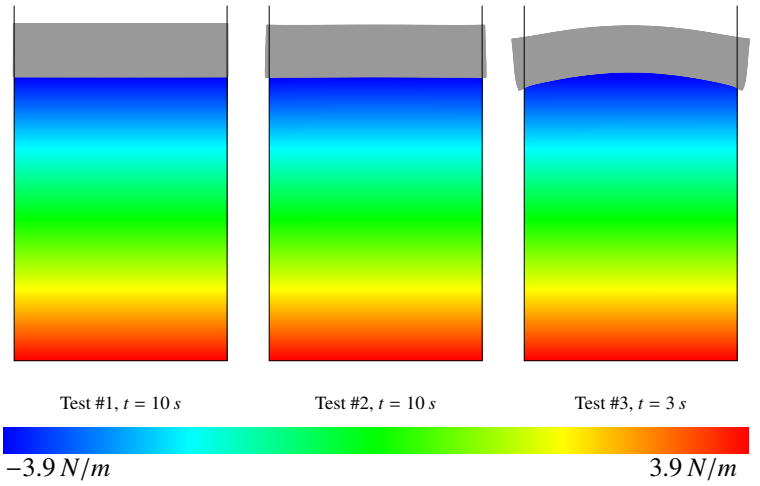


Fig. 7: Supported deformable examples $\tilde{p}$ values. These are the solutions computed by the fluid solver, which does not see the constant pressure mode p_0.

though it will deviate more for the softer test cases .

The example setup is shown in Fig. 6, which also gives the parameters for the three test cases we consider. In all cases, the fluid is inviscid and has the density of $1\,kg/m^2$. There are no collisions between the deformable body and the fluid domain walls. Since the solid is more dense than the fluid, this system also settles into an unstable equilibrium. We avoid the instability by projecting off the total horizontal momentum of the solid, thus preventing the solid from sliding off to one side.

Figure 7 shows the $\tilde{p}$ fields computed by the fluid solver for the three test cases. As in the supported rigid body case, the fluid attains at steady state a pressure profile similar to a hydrostatic solution in all three cases.

The boundary pressure projection computes the constant pressure component p_0, shown in Fig. 6. The interface pressure is given by $p_h \approx Mg/A$, where $p_h = p_0 + \tilde{p}_h$. The variation in pressure in the fluid depends on the density of the fluid and the height of the fluid domain, which is similar in all three test cases. Hence, in all cases, the fluid solver computes an interface pressure $\tilde{p}_h \approx -3.9\,N/m$. Thus, the method should compute $p_0 \approx Mg/A - \tilde{p}_h$. Figure 6 shows good agreement with these expected values of p_0.

In Fig. 8, we plot the number of iterations per time step for the third test case. Figure 8 shows the decrease in the volume error metric over the iterations during the first time step. Note that this example tests extreme scenarios with

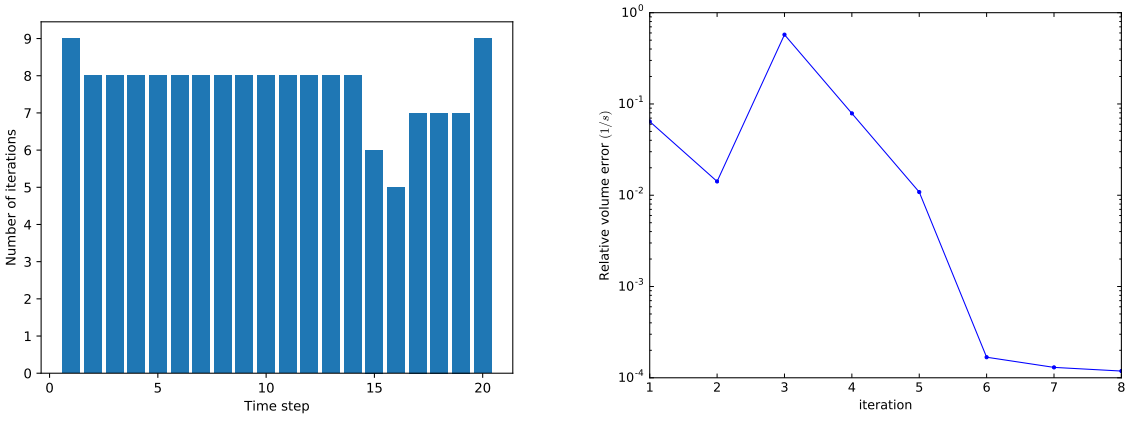


Fig. 8: **(Left)** Number of iterations for all substeps of the supported deformable example (Test #3). **(Right)** Iteration versus $\log(\text{Vol}_{\text{err}})$ during the first time step of supported deformable example (Test #3).

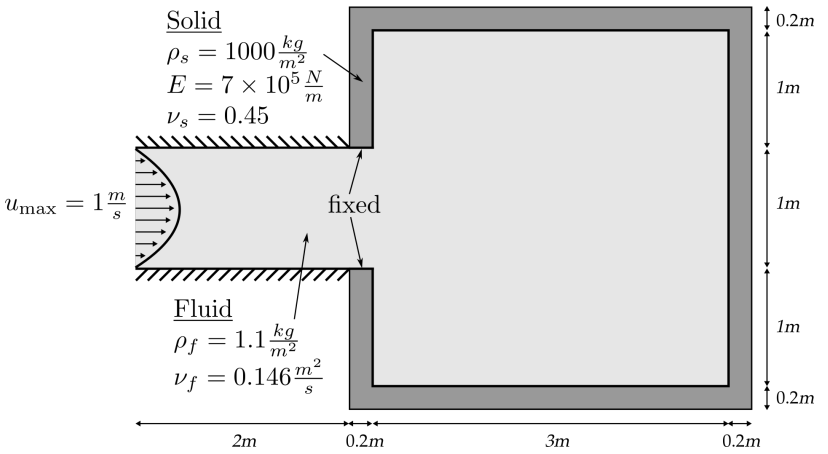


Fig. 9: Setup of a simple balloon-like problem with a volumetric solid.

high density ratios and does not use acceleration techniques [1] in the iterations other than Aitken relaxation.

4.3. Simple balloon-like problem

Similar to the examples in previous works addressing the incompressibility dilemma [29, 9], we study the inflation of a thin, volumetric balloon by a source on the left, as depicted in Fig. 9. Similar balloon inflation problems have been studied with monolithic methods, which do not have an inherent difficulty with such conditions [40, 7, 27, 28], and in a partitioned scheme with Robin-Robin preconditioner [3].

The balloon has a density of $1000\,kg/m^2$ and a thickness of $0.2\,m$. It has a $1\,m$ wide opening, which is fixed to the walls of a static rigid tube that leads to the source. A neo-Hookean constitutive model is used for the solid, with Young's modulus $E = 7{\times}10^5\,N/m$, and Poisson's ratio $\nu_s = 0.45$. The solid volume is constructed using a triangulated mesh with minimum side length of $0.02\,m$ at rest. The total mass is distributed uniformly over the triangle vertices. The fluid has density $\rho_f = 1.1\,kg/m^2$ and kinematic viscosity $\nu_s = 0.146\,m^2/s$.

The source is depicted in Fig. 9. The source velocity has a parabolic profile in y, with a centered maximum value that ramps up gradually from $0\,m/s$ at time $t = 0\,s$ to $u_{\max}$ at time $t = 1\,s$ according to $\frac{1}{2}(1 + \sin(\pi t + \frac{3}{2}\pi))\,u_{\max}$. The

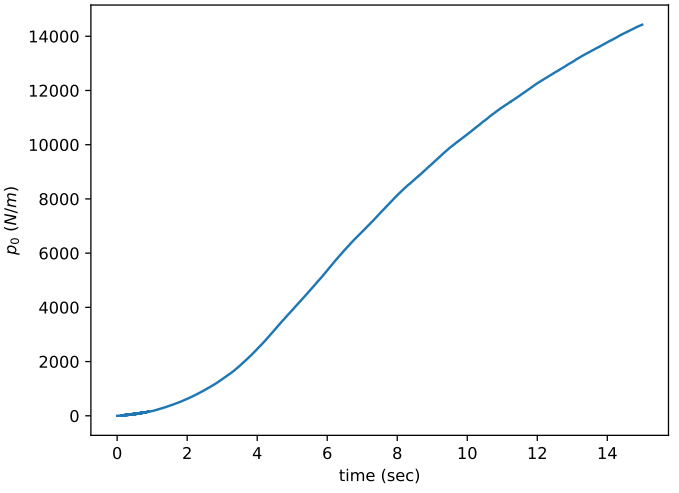


Fig. 10: Volumetric simple balloon-like example with p_0 plotted over time throughout the simulation.

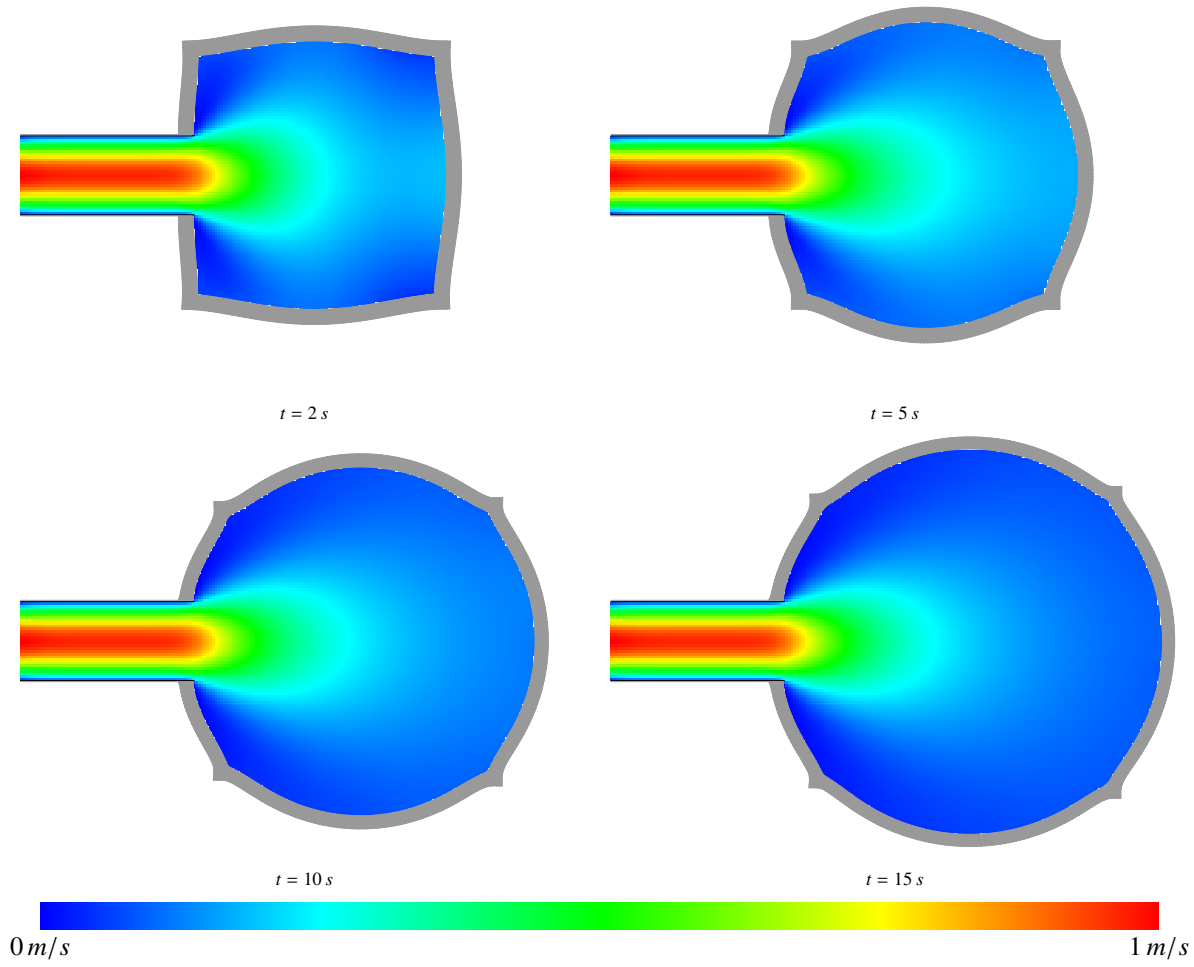


Fig. 11: Snapshots of the fluid velocity magnitudes for the volumetric simple balloon-like example.

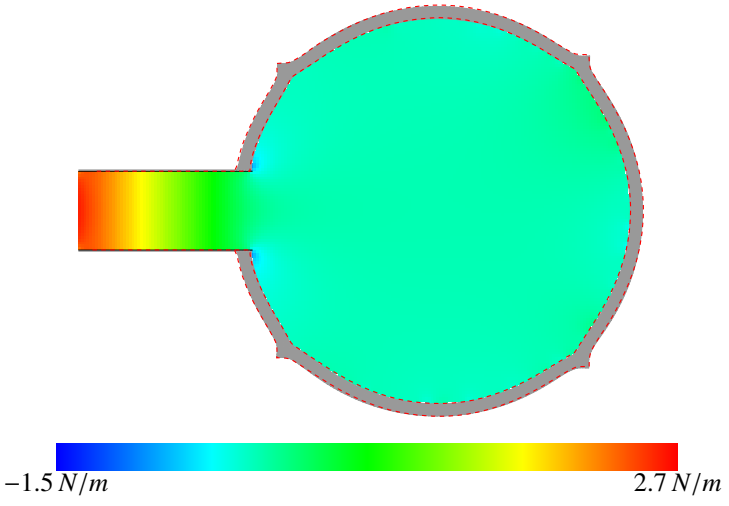


Fig. 12: Volumetric simple balloon-like example. $\tilde{p}$ values at $t = 15$. The outline of the result from [29] is overlaid for comparison (dashed red line).

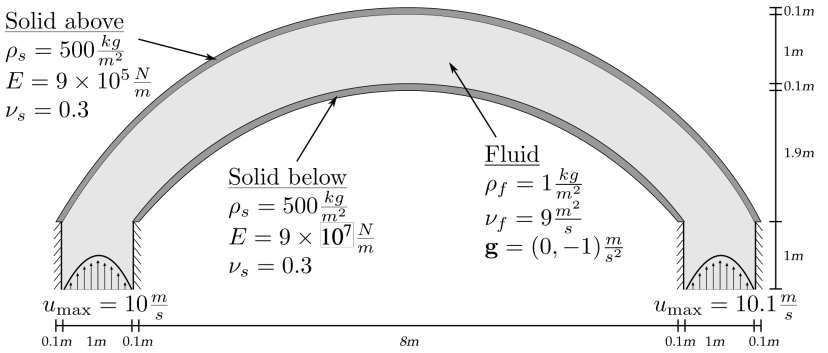


Fig. 13: Damped structural instability problem.

fluid domain is discretized with a cell size of $0.02\,m \times 0.02\,m$, or 270 cells over the domain width and 150 cells over the domain height.

We use $\tau_0 = 10^{-2}$. The simulation is run until time $t = 15\,s$, and the state of the balloon and fluid velocity field at various times are illustrated in Fig. 11. Fig. 12 depicts the fluid pressures without the constant mode, and Fig. 10 plots the evolution of the constant pressure mode, p_0, throughout the simulation. In Fig. 12, we overlay in red the contour of the balloon at time $t = 15\,s$ given in [29], which shows good agreement with our result. While qualitatively our plot in Fig. 10 is similar to those in [29] and [9], the quantitative results differ (note that [29] and [9] also differ from each other). This may be due to different parameters, as the parameters of [29] are given in three-dimensional units, and it is not clear how to determine the equivalent parameters for our two-dimensional simulation. The velocity fields in Fig. 11 are in good agreement with those given in [29, 9].

4.4. Damped Structural Instability

We study an example with a nonsymmetric structure and fluid setup, similar to previous works [29, 17, 9, 16, 18], as illustrated in Fig. 13. A solid tube is constructed by attaching static, rigid tubes to both ends of a free deformable, volumetric tube. A source is placed at the entrance to each rigid tube, and the asymmetry in dynamics is achieved by choosing differing source velocities at either end.

Both the top and bottom bands of the deformable tube have a thickness of $0.1\,m$ and a density of $500\,kg/m^2$. A neo-Hookean constitutive model is used with Poisson's ratio $\nu_s = 0.3$. The top band has a Young's modulus of

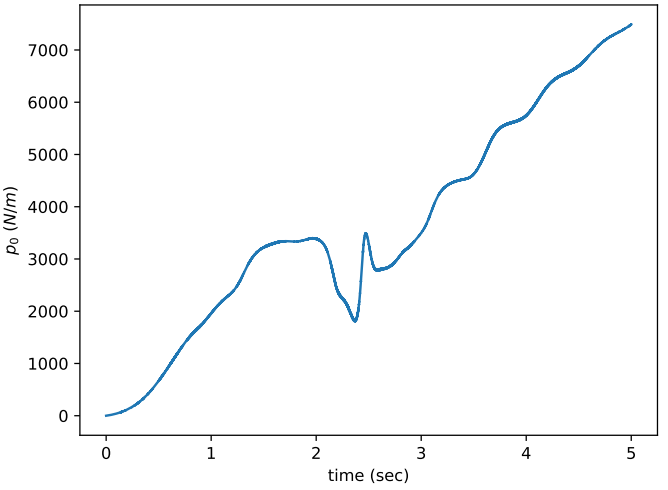


Fig. 14: The change in p_0 over time in the damped structural instability example. p_0 gradually increases as the tube is inflated, with drops in p_0 corresponding to buckling of the bottom band.

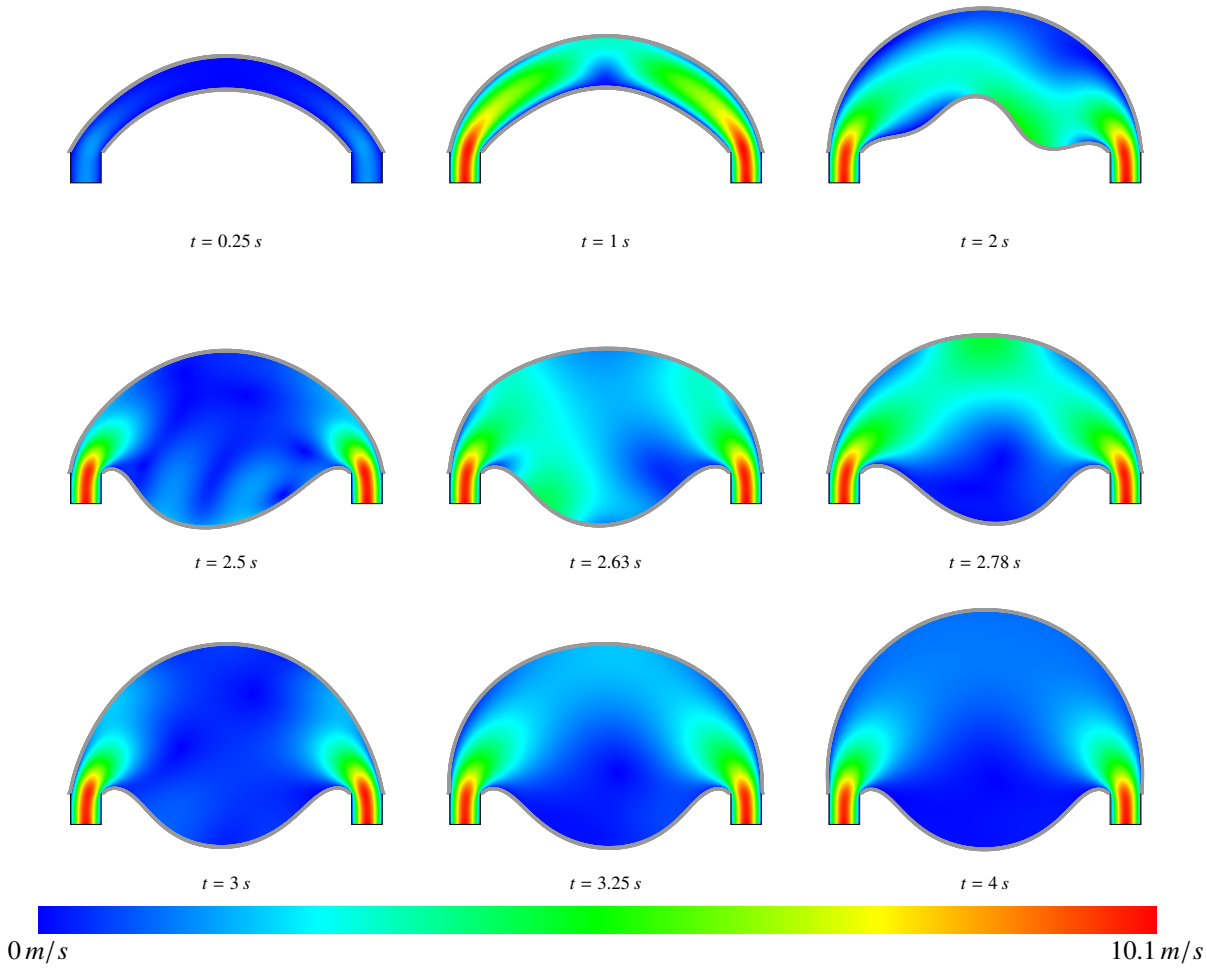


Fig. 15: The fluid velocity field magnitude $\|\mathbf{u}\|$ at different stages of the simulation for the damped structural instability problem.

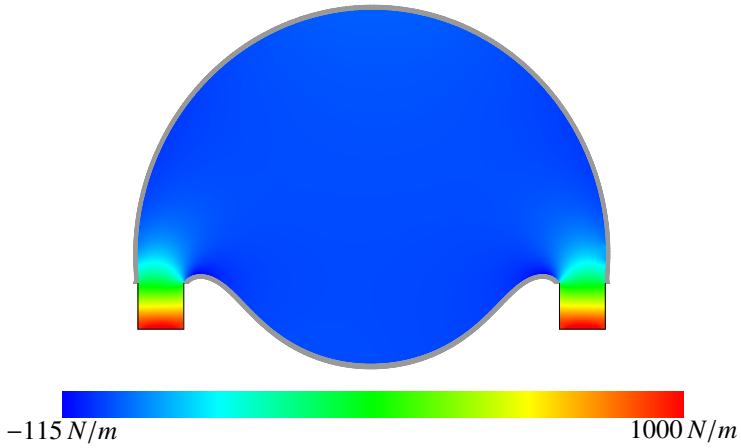


Fig. 16: Fluid solver pressure field $\tilde{p}$ at time=4 s in the simulation of the damped structural instability.

$9 \times 10^5\, N/m$, and the bottom band is significantly stiffer with a Young's modulus of $9 \times 10^7\, N/m$. We note that we used a Young's modulus value one order of magnitude lower than the published value in [29] since we do not observe buckling at the higher stiffness. The fluid density is $1\, kg/m^2$ and the kinematic viscosity is $9\, m^2/s$.

Both fluid sources have parabolic profiles in x, where the center of each source starts at $0\, m/s$ at time $t = 0\, s$ and ramps up to $u_{\max}$ at time $t = 1\, s$ according to the sinusoidal $\frac{1}{2}(1 + \sin(\pi t + \frac{3}{2}\pi))\, u_{\max}$. The $u_{\max}$ values are $10\, m/s$ and $10.1\, m/s$ for the left and right sources, respectively.

The solid mesh is built with triangulated areas with a minimum side length of $0.05\, m$. We observed convergence problems on this example due to the voxelized fluid solver which required the use of modified tolerances: $10^{-2} \leq \omega \leq 1$, $\tau_m = 10^{-6}$, $\tau_u = \tau_v = 10^{-1}$, $\tau_p = \tau_0 = 1$, $\tau_e = 5 \times 10^{-3}$.

A few snapshots capturing key frames of the simulation are displayed in Fig. 15, where the fluid velocity field magnitude is shown along with the solid bands. Similar to the examples in previous works [29, 9], initially the top band moves up as the tube is inflated, while the stiffer bottom band does not displace significantly. Once a critical pressure is reached, the bottom band starts to give near the side with higher source velocity, creating a flow inside the tube toward the other side. Once the bottom band stabilizes, the inner flow calms down, and the tube regains and maintains a more symmetrical shape while continuing to inflate until the end of the simulation. This behavior is also captured on the p_0 plot in Fig. 14, where the sudden decrease in p_0 can be observed after the bottom band buckles around $t = 2\, s$. The value of p_0 resumes its gradually increasing trend once the solid settles. The fluid solver pressure field $\tilde{p}$ at the end of the simulation is shown in Fig. 16.

4.5. Inflating a Thin-shell Balloon

A thin-shell solid is one that is modeled as a surface rather than a volume, so that each point on an immersed thin shell is affected by fluid forces on both sides of the surface. In this example, a thin-shell balloon is attached to a static rigid tube that leads to a source as shown in Fig. 17. The balloon is inflated by the fluid source for 15 seconds.

The solid has one-dimensional density of $200\, kg/m$, with the same total mass as in the volumetric case above. A mass-spring constitutive model that resists stretching and bending is used to model the balloon-like behavior with the following stiffness (k) and damping (b) parameters: $k_{stretch}$=$5 \times 10^4\, kg/s^2$, $b_{stretch}$=$1500\, kg/s$, k_{bend}=$5 \times 10^4\, kg/s^2$, b_{bend}=$800\, kg/s$. The solid is at rest initially, with spring rest lengths of $0.02\, m$.

The fluid density and kinematic viscosity are $1.1\, kg/m^2$ and $0.146\, m^2/s$, respectively. The source profile is parabolic in y, with its peak centered in the source and ramping up from $0\, m/s$ at time $t = 0\, s$ to $u_{\max} = 1\, m/s$ at time $t = 1\, s$ according to $\frac{1}{2}(1 + \sin(\pi\, t + \frac{3}{2}\pi))\, u_{\max}$. A uniform grid with cell width $dx = 0.02\, m$ is used for the fluid domain inside the closed region. We use $\tau_0 = 10^{-2}$.

The spring parameters used in this example result in balloon behavior that is less stiff than the volumetric one, as can be observed in the velocity field snapshots shown in Fig. 18. The effect of the fluid motion on the solid is more localized in this example, and the balloon bounces back and forth as it is inflated. These movements create fluctuations in the calculated p_0 values throughout the simulation as illustrated in the plot in Fig. 19. The initial values of p_0 are relatively small, but they eventually increase significantly, especially after the balloon loses its initial square-like shape around $t = 4\, s$.

The fluid solver pressures $\tilde{p} \sim 10^{-4} p_0$ are relatively small compared to p_0. We provide a representative snapshot of the fluid solver pressure field $\tilde{p}$ at the end of the simulation in Fig. 20.

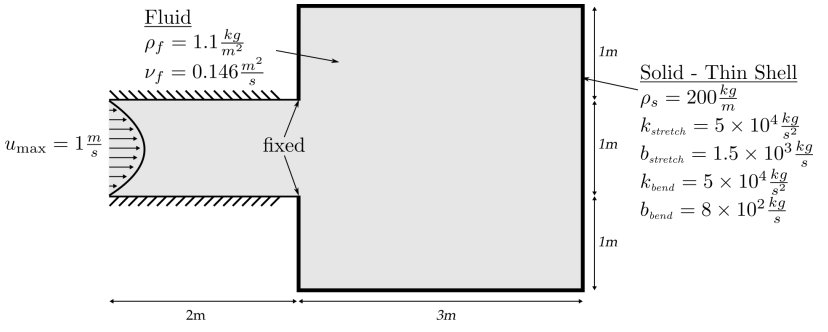


Fig. 17: Setup of a simple balloon-like problem with a thin-shell solid.

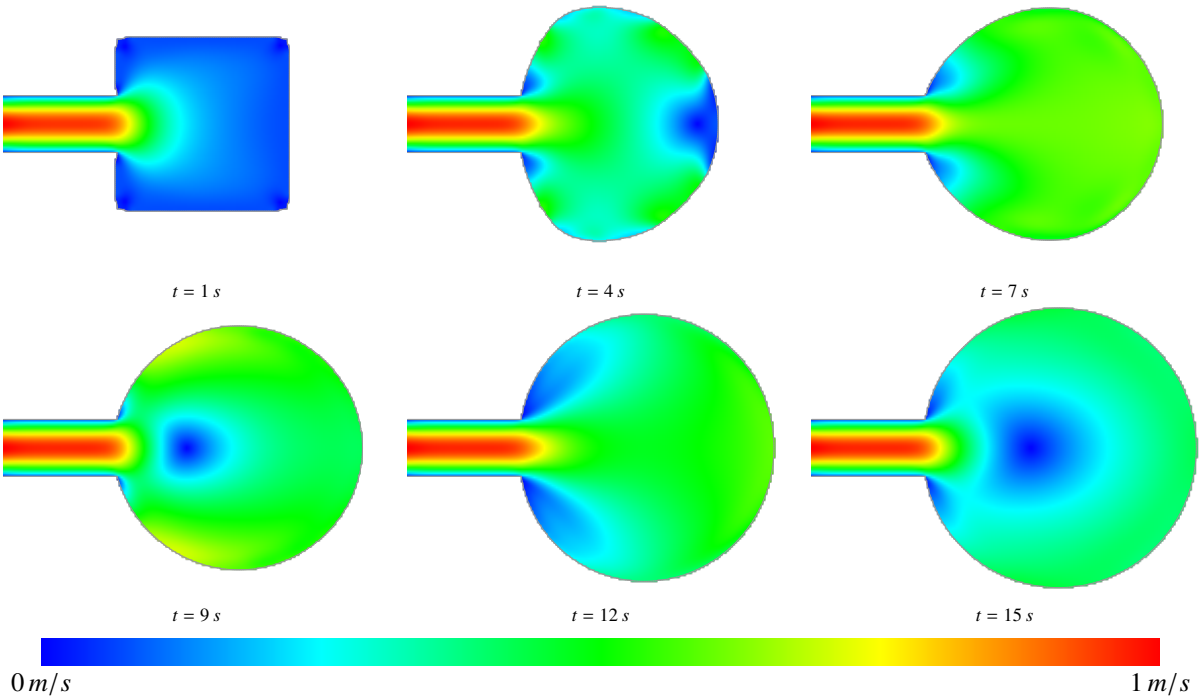


Fig. 18: Snapshots at various stages of the simulation demonstrating the interaction between the fluid and the thin-shell solid in the simple balloon-like example. Color coding represents the fluid velocity field magnitude, $\|\mathbf{u}\|$.

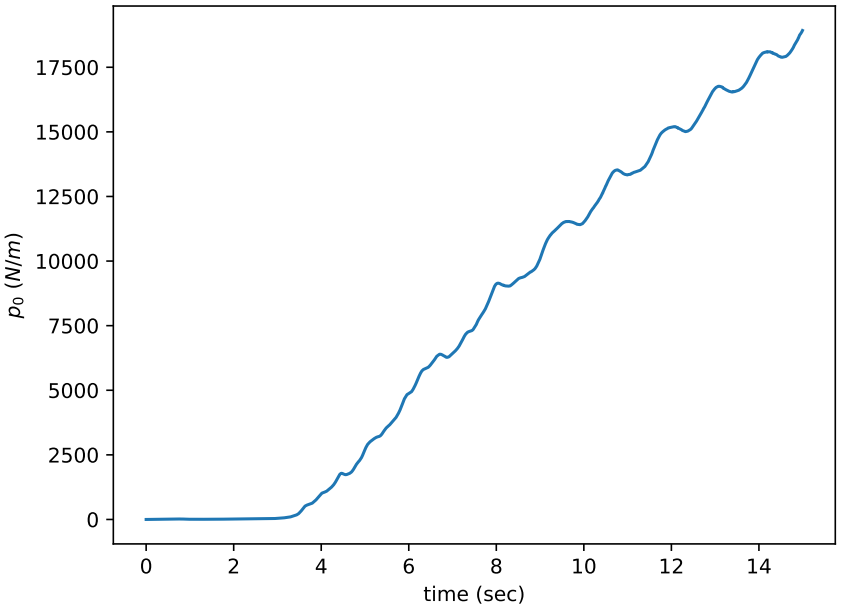


Fig. 19: Constant pressure mode p_0 of the enclosed fluid domain plotted against time for the duration of the simple balloon-like example with a thin-shell solid.

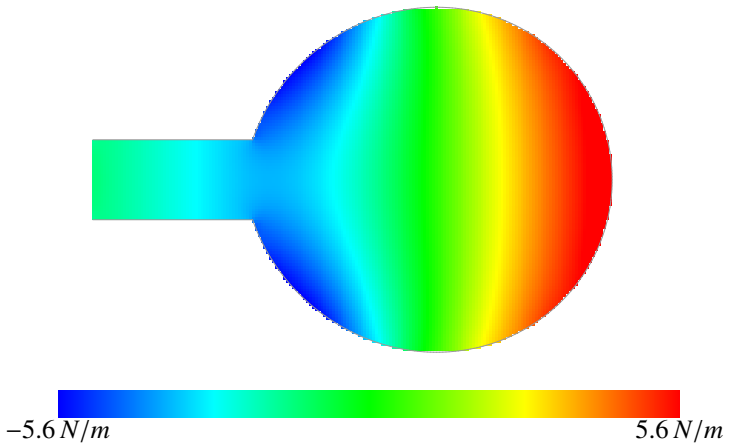


Fig. 20: Fluid solver pressures ($\tilde{p}$) in the fluid pressure at time $t = 10\,s$ are illustrated with a color map for the simple balloon-like example with a thin-shell solid.

4.6. Hydraulic Press at Equilibrium

We demonstrate our method on an example that models a hydraulic press at equilibrium, as illustrated in Fig. 21. The fluid is inviscid with a density of $\rho_f = 1\,kg/m^2$, while the solids are significantly more dense with $\rho_s = 1.5 \times 10^5\,kg/m^2$. The fluid grid cell width is $dx = 0.02\,m$ and we use $\tau_0 = 10$. Since the subiterations converge in two iterations for all time steps except the first (which takes three), we disable relaxation for this test.

As in the example of Section 4.1, at equilibrium the gravitational and fluid pressure forces on each rigid piston are exactly balanced, giving

$$-\rho_b A_b h_b g + A_b p_b = 0, \quad b \in [l, r], \tag{36}$$

where ρ_b is the density of the body, h_b is height of the body, A_b and p_b are the surface area and fluid pressure, respectively, at the interface of the body with the fluid, and the index b indicates the left (l) or right (r) body. In this example, $\rho_l = \rho_r = \rho_s$ and $h_l = h_r = h_s$, so that after cancellation of A_b and rearranging we get

$$p_b = \rho_s h_s g = 1.47 \times 10^5 N/m, \quad b \in [l, r], \tag{37}$$

for the interfacial fluid pressure on both bodies at equilibrium. As shown below, we compute a value of $p_0 \approx 147003.57\,N/m$, and $\Delta p \approx -3.57\,N/m$ for the fluid solver pressure at the interface, giving the predicted result for $p_b = p_0 + \Delta p$ at the interface.

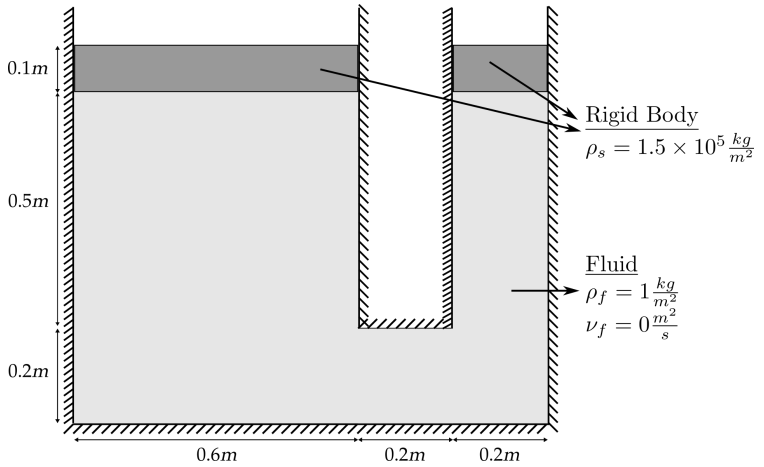


Fig. 21: Setup of the hydraulic press example.

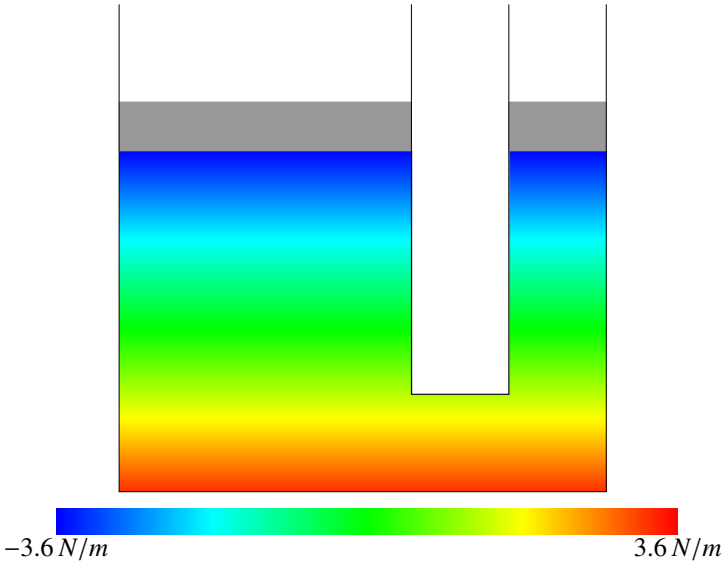


Fig. 22: Color-mapped illustration of the fluid solver pressure field $\tilde{p}$ for the hydraulic press at equilibrium ($t = 10\,s$). These values are added to the constant pressure mode p_0 determined by our method to obtain the interfacial pressures.

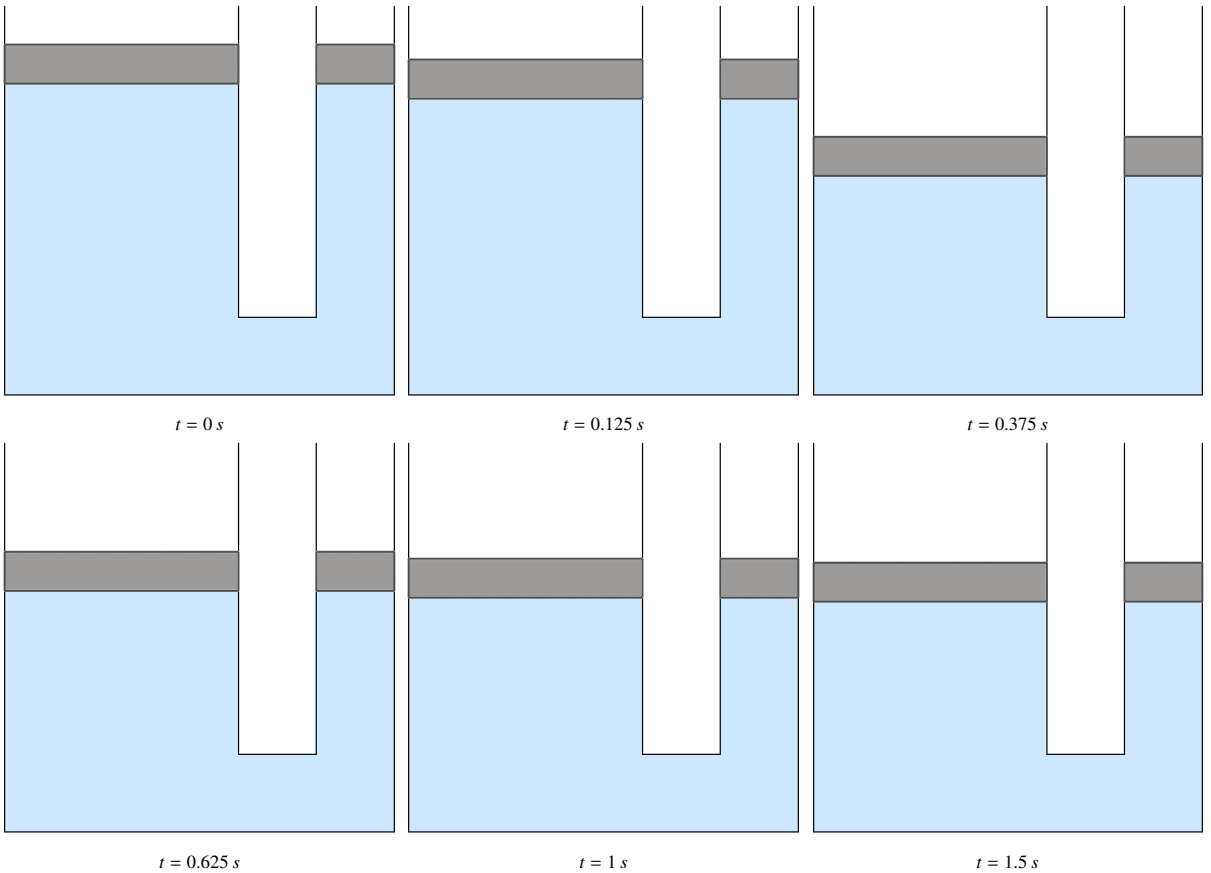


Fig. 23: Failed piston example using the fluid momentum based formulation of [1] is depicted at various times t in the simulation. Since the density ratio $\rho_s/\rho_f \sim 1500$ is large, use of the fluid momentum equation leads to poor convergence and underestimation of p_0, allowing both piston heads to sink down and compress the fluid.

The example is simulated until time $t = 10\,s$. The value of p_0 is steady over time, with maximum deviation of $3.4 \times 10^{-15} p_0$. The fluid solver pressure field $\tilde{p}$ is shown in Fig. 22. Again, the p_0 computation compensates for the arbitrary constant mode in the fluid solver pressure field.

4.6.1. Comparison with Fluid Momentum Formulation

We compare our formulation with the fluid-momentum-based formulation of [1] by studying the case of the hydraulic press with a large mass density ratio of $\rho_s/\rho_f \approx 1500$. As shown in Fig. 23, using the formulation in [1], both piston heads fall downward because the constant pressure mode that is required to hold the system in equilibrium is underestimated due to poor convergence. Although the accumulated p_0 value increases as the simulation progresses and approaches the physically correct value, the solution does not achieve equilibrium as the piston heads continue fluctuating.

4.7. Driven Hydraulic Press

We next demonstrate the hydraulic press driven by an initial external force, as illustrated in Fig. 24. The fluid has density of $\rho_f = 1\,kg/m^2$ and kinematic viscosity $\nu_f = 0.146\,m^2/s$, while the rigid bodies have density $\rho_s = 15\,kg/m^2$. The fluid grid cell width is $dx = 0.02\,m$. For this test we use $\tau_0 = 10^{-2}$. Since the solids are more dense than the fluid in this example, we eliminate the horizontal and rotational degrees of freedom for the rigid bodies to stabilize the dynamics and the equilibrium.

Initially, the solids are displaced away from the equilibrium state as depicted in Fig. 24. For the system to be at equilibrium in this state, the fluid and external forces on the left and right pistons must be balanced, giving

$$-\rho_s A_l h_s g + A_l p_l = 0 \tag{38}$$

$$-\rho_s A_r h_s g + A_r p_r + f_{\max} = 0, \tag{39}$$

where the subscripts l and r denote the left and right piston, respectively, h_s is the height of each piston, $A_{l,r}$ is the surface area of the piston, $p_{l,r}$ is the fluid pressure at the piston interface, and $f_{\max}$ is the external force on the right piston. At equilibrium, the pressure satisfies

$$p = \rho g(y - y_0) + p_0 \tag{40}$$

$$p(y_0) = p_0. \tag{41}$$

Substituting the expression for pressure into the force balance equations above and solving for $f_{\max}$, we get

$$f_{\max} = A_r \rho g(y_l - y_r), \tag{42}$$

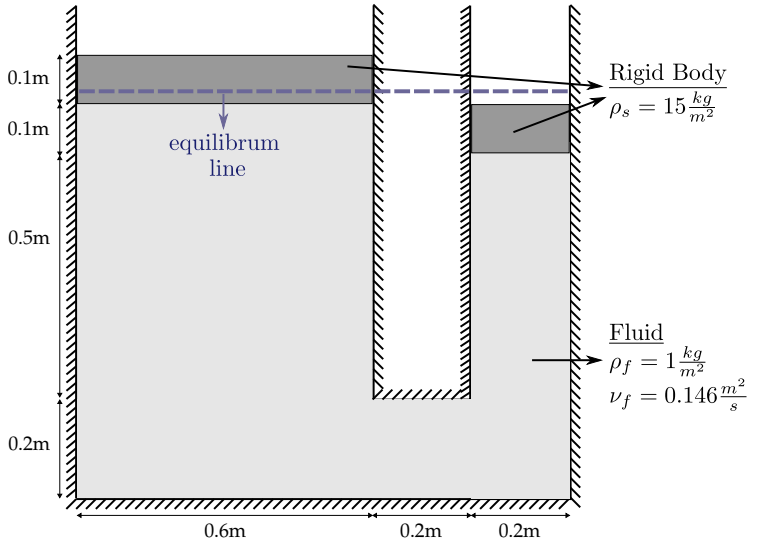


Fig. 24: Setup of the driven hydraulic press example. Initially, an external downward force is applied to the right piston.

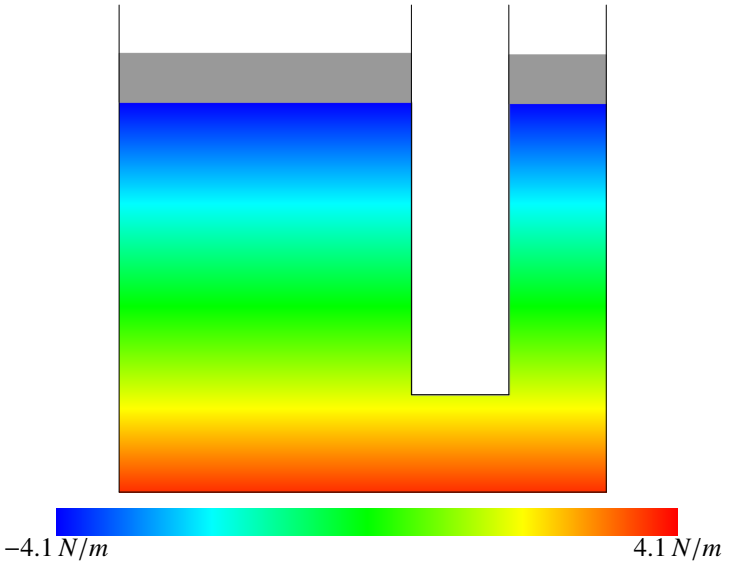


Fig. 25: Color-mapped illustration of the pressure values computed by the fluid solver, $\tilde{p}$, for the driven hydraulic press example at time $t \approx 80\,s$.

where y_l and y_r are the heights of the left and right pistons, respectively.

We apply the force $f_{\max}$ until time 1 s, keeping the system in equilibrium. We then release the force. The fluid pressure field $\tilde{p}$ is shown in Fig. 25. After release of the external force, the constant pressure mode drops slowly as it approaches the state of equilibrium as shown in Fig. 26. The p_0 computation compensates for the arbitrary constant mode in the fluid solver pressure field. Note that p_0 is roughly piecewise constant in this figure with frequent small jumps and less frequent (by about a factor of three) large jumps. These are caused by the voxelized nature of our fluid solver. Small jumps occur when the small rigid body crosses cell centers (thus covering or uncovering an entire layer

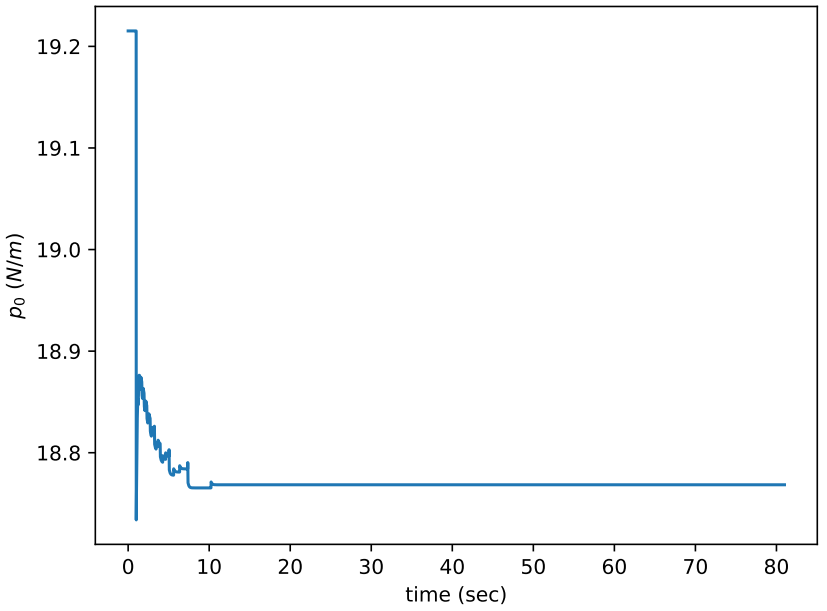


Fig. 26: Constant pressure mode p_0 computed by the BPP during the simulation of the driven hydraulic press example.

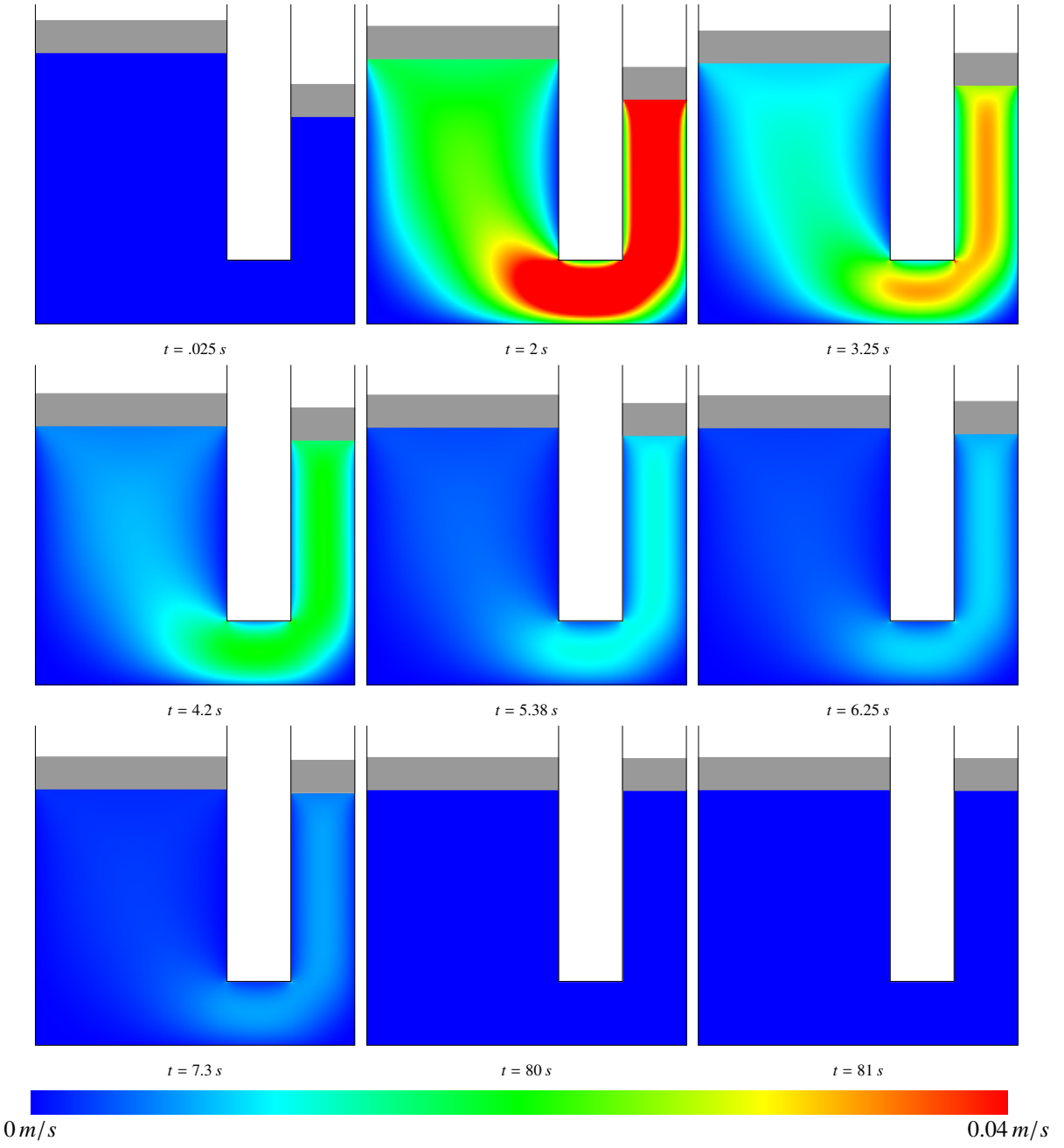


Fig. 27: Snapshots of the fluid velocity field magnitude, $\|\mathbf{u}\|$, at various stages of the simulation of the driven hydraulic press.

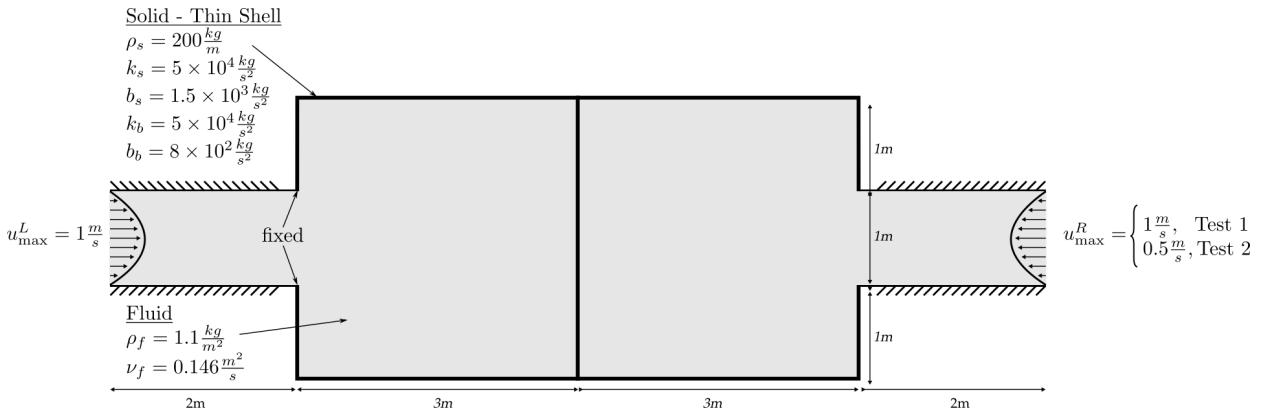


Fig. 28: Setup of the coupled, neighboring thin-shell balloons example. The setup is geometrically symmetric, with a dynamic asymmetry introduced by varying the right source velocity.

of fluid cells at once). Large jumps occur when the large rigid body crosses cell centers (which is three times wider and affects three times as many grid cells). The extra layer of fluid cells has lower hydrostatic pressure than the rest of the cells. The resulting shift in average pressure is picked up by p_0. Jumps occur less frequently as the solids slow down.

Snapshots of the fluid velocity field magnitude throughout the simulation are shown in Fig. 27. At $t = 3.25\ s$, the pistons are displacing away from their initial positions. Images in the bottom row depict the velocities as the system is slowing down.

4.8. Inflating Two Conjoined Thin-shell Balloons

Neighboring enclosed fluid regions separated by a thin membrane will generally have two different values for p_0 that are instantaneously coupled through their shared interface. Our formulation extends naturally to solve for the coupled constant pressure modes of neighboring regions. For the first example with multiple neighboring regions, the simple thin-shell balloon-like problem is extended by adding another balloon and source that mirrors the original one across the y-axis. As in the simple balloon example, the density of the fluid is $1.1\,kg/m^2$ and its kinematic viscosity is $0.146\,m^2/s$, while the one-dimensional solid density is $200\,kg/m$. The parameters of the mass-spring system along with a detailed depiction of the example setup are given in Fig. 28. The same parabolic profile is used for both fluid sources along their respective y-axes. The source velocities are gradually increased during the first second of the simulation, so that the maximum value of the parabola is given by $u^*(t) = \frac{1}{2}(1 + \sin(\pi\, t + \frac{3}{2}\pi))\, u^*_{\max}$, with $u^L_{\max} = 1\ m/s$ for the left source and $u^R_{\max}$ is specified for each example below for the right source.

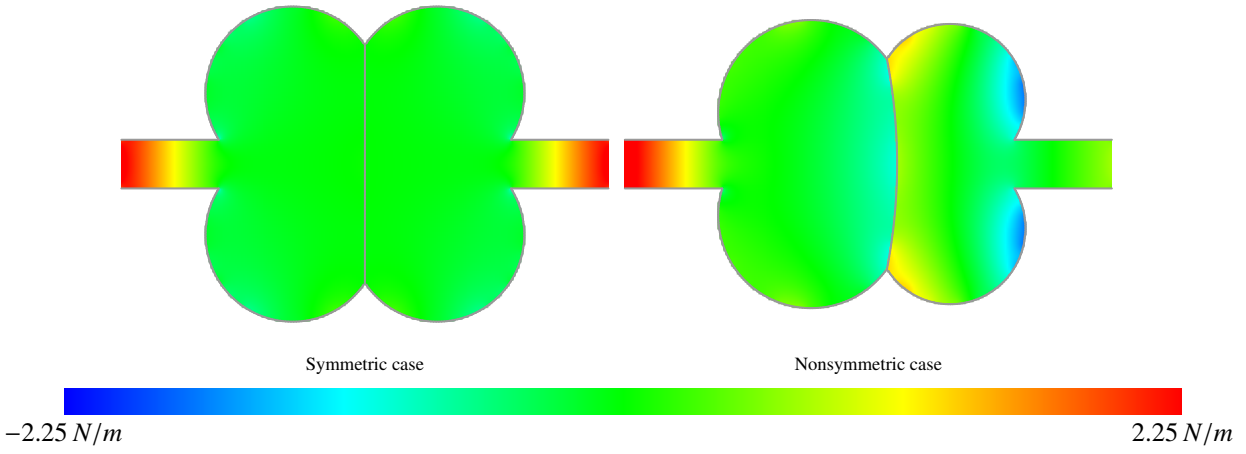


Fig. 29: Color map of the fluid solver pressure field $\tilde{p}$ for the symmetric **(left)** and nonsymmetric **(right)** tests of two coupled thin-shell balloons.

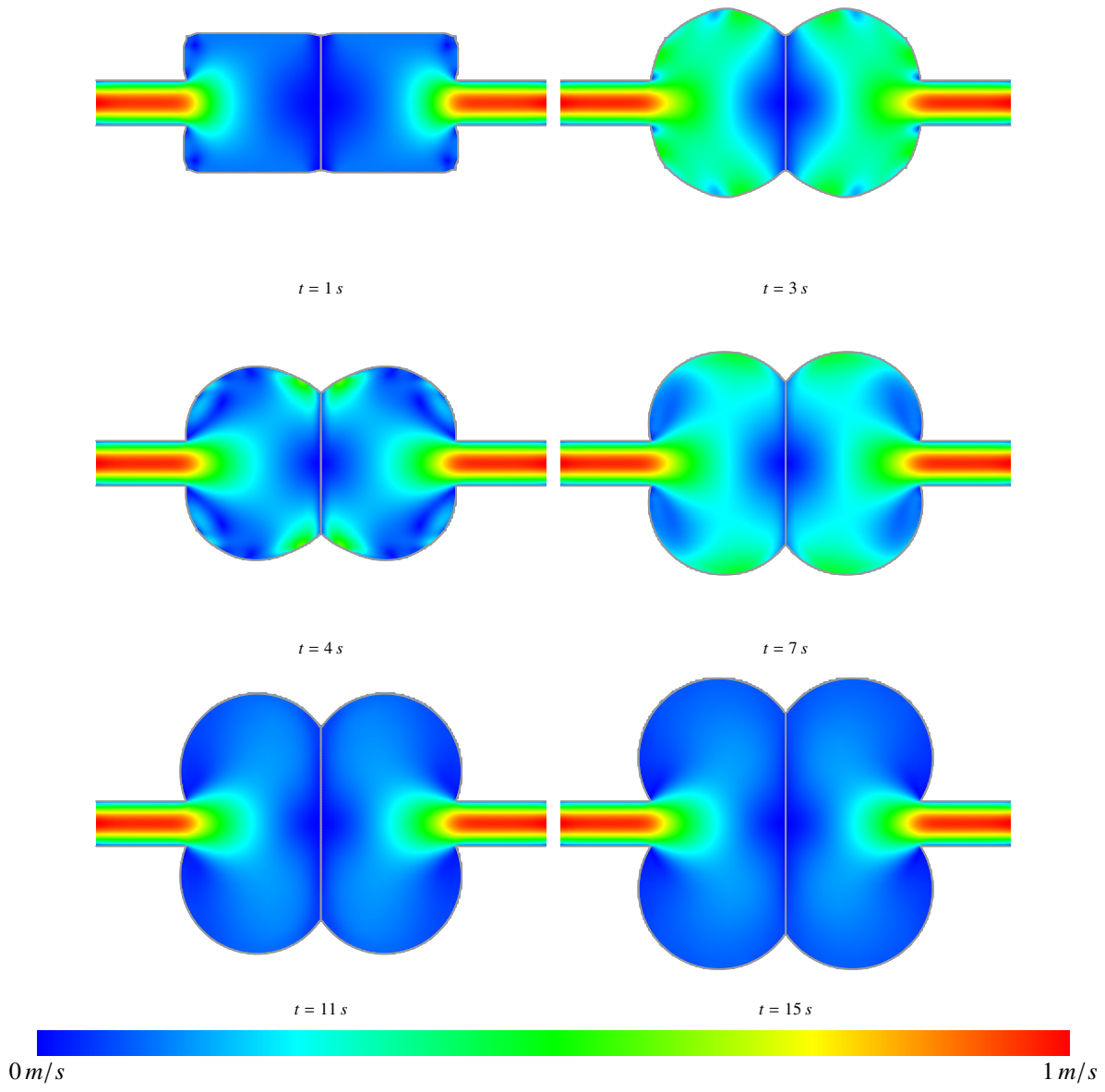


Fig. 30: Snapshots of the solid position and fluid velocity magnitude in the simulation of the inflation of two conjoined, thin-shell balloons. The speeds of the left and right sources are identical, and the two balloons inflate symmetrically with the membrane between them remaining centered.

We carried out two different tests with the same layout. In Test #1, $u^R_{\max}$ is set to the same magnitude as $u^L_{\max}$ in the opposite direction for a symmetric setup. In Test #2, we use a nonsymmetric setup where the right source maximum velocity is halved, $u^R_{\max} = 0.5\,m/s$, while it is still in the opposite direction of $u^L_{\max}$. For both tests, the fluid grid cell size is $dx = 0.02\,m$, and the solid springs have matching resolution.

We use $\tau_0 = 10^{-2}$ for this test. In this test, we found the time step heuristics used by the solid and fluid solvers to be too large to produce good results, and we additionally impose $\triangle t \leq 10^{-3}\,s$. The tests are run until time $t = 15\,s$. The fluid solver pressure field $\tilde{p}$ at the end of the simulation as well as the state of the balloons are shown in Fig. 29. The pressure field extrema are located around the edges of the balloons since they still bounce slightly at the end of

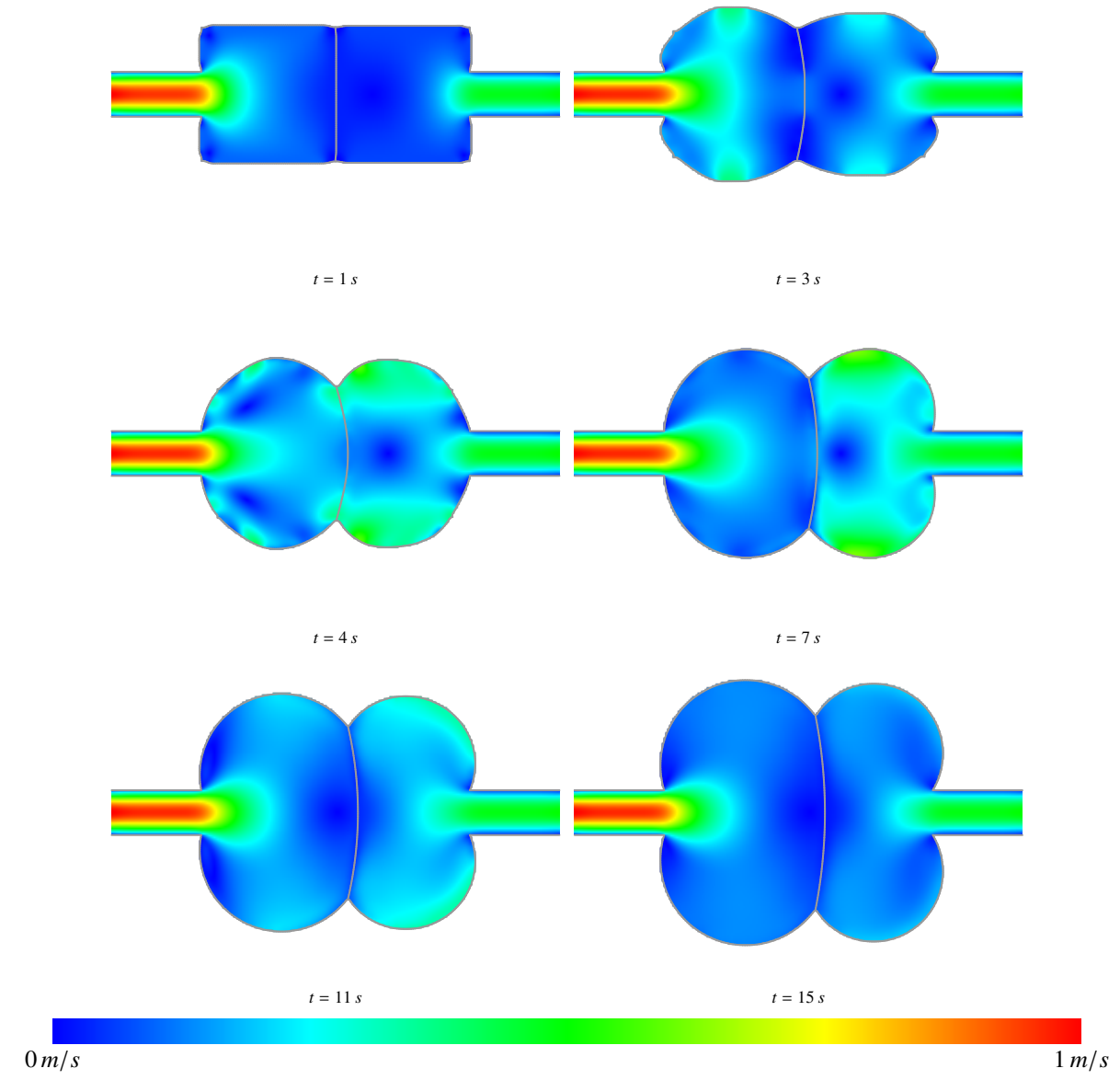
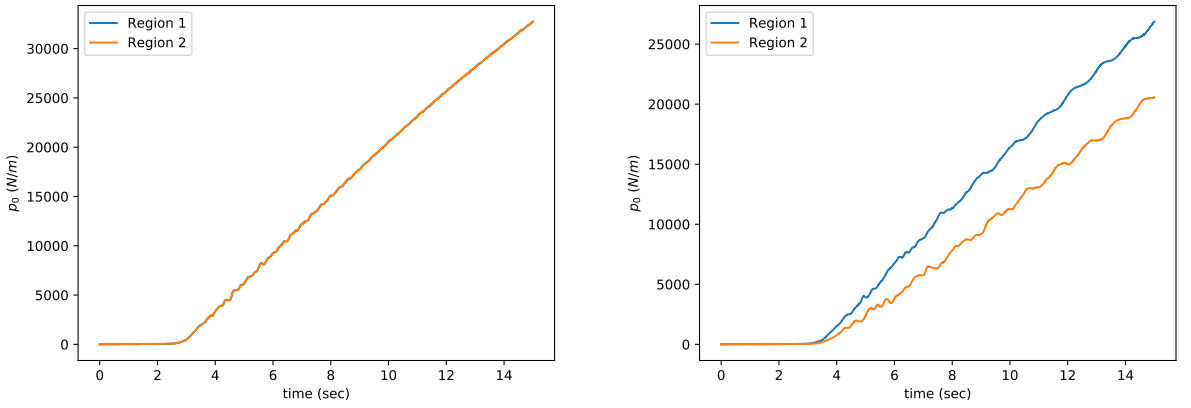


Fig. 31: Snapshots of the solid position and fluid velocity magnitude in the simulation of the inflation of two conjoined, thin-shell balloons. The source speed on the right is half that on the left, leading to an asymmetric configuration where the left balloon is more inflated than the right, and the membrane between the regions bows to the right.

the simulation. Evolution of the balloon shapes and the fluid velocity fields are depicted in snapshots in Fig. 30 and Fig. 31 for the symmetric and nonsymmetric cases, respectively.

In the symmetric case, both the balloons and the velocity fields evolve symmetrically until the end, and the opposing pressures force the balloons to expand vertically creating a butterfly-like shape. The center line between the balloons does not move horizontally during the simulation and the fluid velocity around it is close to zero. The symmetry can be further observed on the p_0 plot in Fig. 32 (left), where the constant modes of the two regions overlap for the duration of the simulation. Similar to the single balloon case, a sudden jump occurs around $t = 3\,s$, as the balloons lose their original shape and bounce back and forth slightly to adopt their new shape.



Symmetric case Nonsymmetric case

Fig. 32: Constant pressure mode, p_0, throughout the simulation of the symmetric (left) and nonsymmetric (right) inflation of two neighboring thin-shell balloons. Region #1 is the left region, and Region #2 is the right region.

In the nonsymmetric case, the right region is inflated at half the rate of left region, which has the same source flow as the balloons in the symmetric case. However, in this case, the left region grows horizontally into a wider shape as the centerline moves towards the right. The movement of the center line is mainly driven by the differences of the constant pressure modes of the left and right regions, p_0^L and p_0^R, respectively. These modes are illustrated in Fig. 32 (right), where the difference in values as well as the jumps after shape deformations can be observed. When compared with the symmetric case, both p_0^L and p_0^R values are smaller. This result is expected for the right region as its input source velocity is smaller. Furthermore, the smaller source velocity on the right combined with the coupling of the two regions across their shared membrane results in a lower pressure in the left region as well.

4.9. Inflating Nested Thin-shell Balloons

In this example, three circular balloon-like thin-shell solids with different radii are centered around a circular fluid source forming a nested layout as depicted in Fig. 33. Three distinct Dirichlet fluid regions are formed by the balloons, and the constant pressures modes inside these regions are determined in a coupled manner using the BPP. In contrast

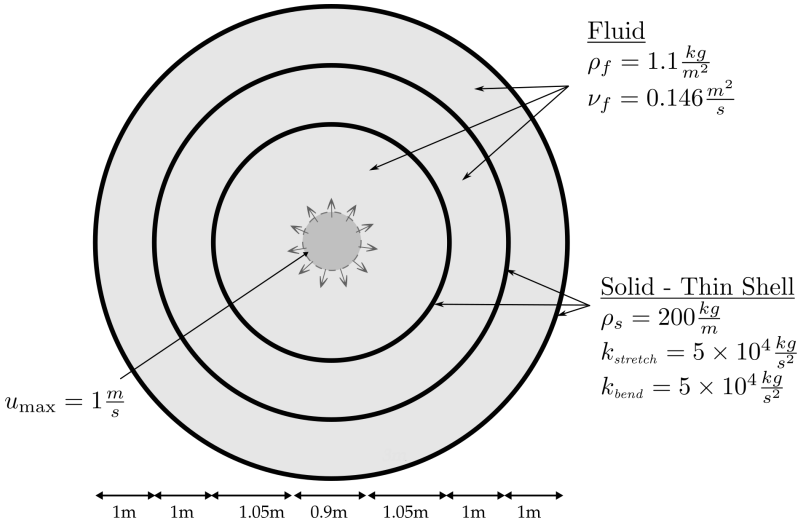


Fig. 33: Setup of the nested balloon rings example. Three distinct Dirichlet fluid regions are formed by the balloons, and the constant pressures modes inside these regions are determined in a coupled manner using the BPP.

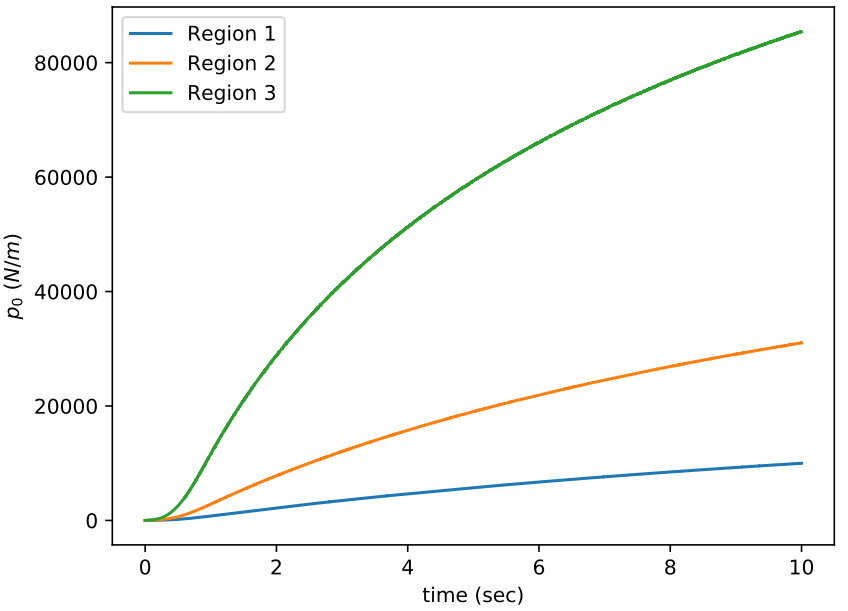


Fig. 34: Plot of constant pressure modes over time for all three Dirichlet regions in the nested thin-shell balloons example. Region #1: outermost region, Region #2: middle region, Region #3: innermost region.

to the two-balloon example, all solid nodes of the two interior balloons are affected by the pressures of two different closed fluid regions.

All three balloons have a one-dimensional density of $200\,kg/m$, and are modeled with the following mass-spring system parameters: $k_{stretch}$=$5\times10^4\,kg/s^2$, k_{bend}=$5\times10^4\,kg/s^2$. The damping is chosen so that each spring in isolation is critically damped. The radii of the solid balloons, from innermost to outermost, are $1.5\,m$, $2.5\,m$, and $3.5\,m$. The circular fluid source at the center of the balloons has a radius of $r_s = 0.45\,m$. The source velocities are in the direction normal to the circle, and have a uniform profile along the surface. They increase gradually from $0\,m/s$ at time $t = 0\,s$ to $u_{\max} = 1\,m/s$ at time $t = 1\,s$ according to the equation $u_s(t) = \frac{1}{2}(1 + \sin(\pi\, t + \frac{3}{2}\pi))u_{\max}$. The fluid density is $\rho_f = 1.1\,kg/m^2$ and the kinematic viscosity is $\nu_f = 0.146\,m^2/s$.

The fluid grid has cell width $dx = 0.02\,m$ and all balloons have matching spring edge lengths initially. We use $\tau_0 = 10^{-2}$ for this example. We enforced a maximum time step size $\triangle t \leq 10^{-3}\,s$ for this test.

Fig. 34 shows the evolution of the constant pressure modes in each region, where the regions are numbered in increasing order from outside to inside. For a given change in volume, the strain on a balloon of radius r is proportional to $\frac{1}{r^2}$. Thus the constant pressure mode needed to balance the balloon stress should increase from outermost to innermost balloon, consistent with the result depicted in Fig. 34.

Evolution of the balloons and fluid velocity field is depicted in snapshots from the simulation in Fig. 35. The flow coming from the source increases the volume of the innermost region, while the volumes of the outer regions are unchanged as they are pushed outwards. The constant pressure mode of each region is coupled to its neighboring regions, and pressure differences move the balloons.

4.9.1. Convergence Study

We study the convergence for the nested rings by comparing against the analytic solution. The velocity field is incompressible and radially symmetric and given by

$$\mathbf{u}(\mathbf{x}) = c\|\mathbf{x}\|^{-2}\mathbf{x}, \tag{43}$$

where $c = 2\pi r_s u_s$. From this we find that the fluid pressure gradient is

$$\nabla p = \rho c^2 \|\mathbf{x}\|^{-4}\mathbf{x}, \tag{44}$$

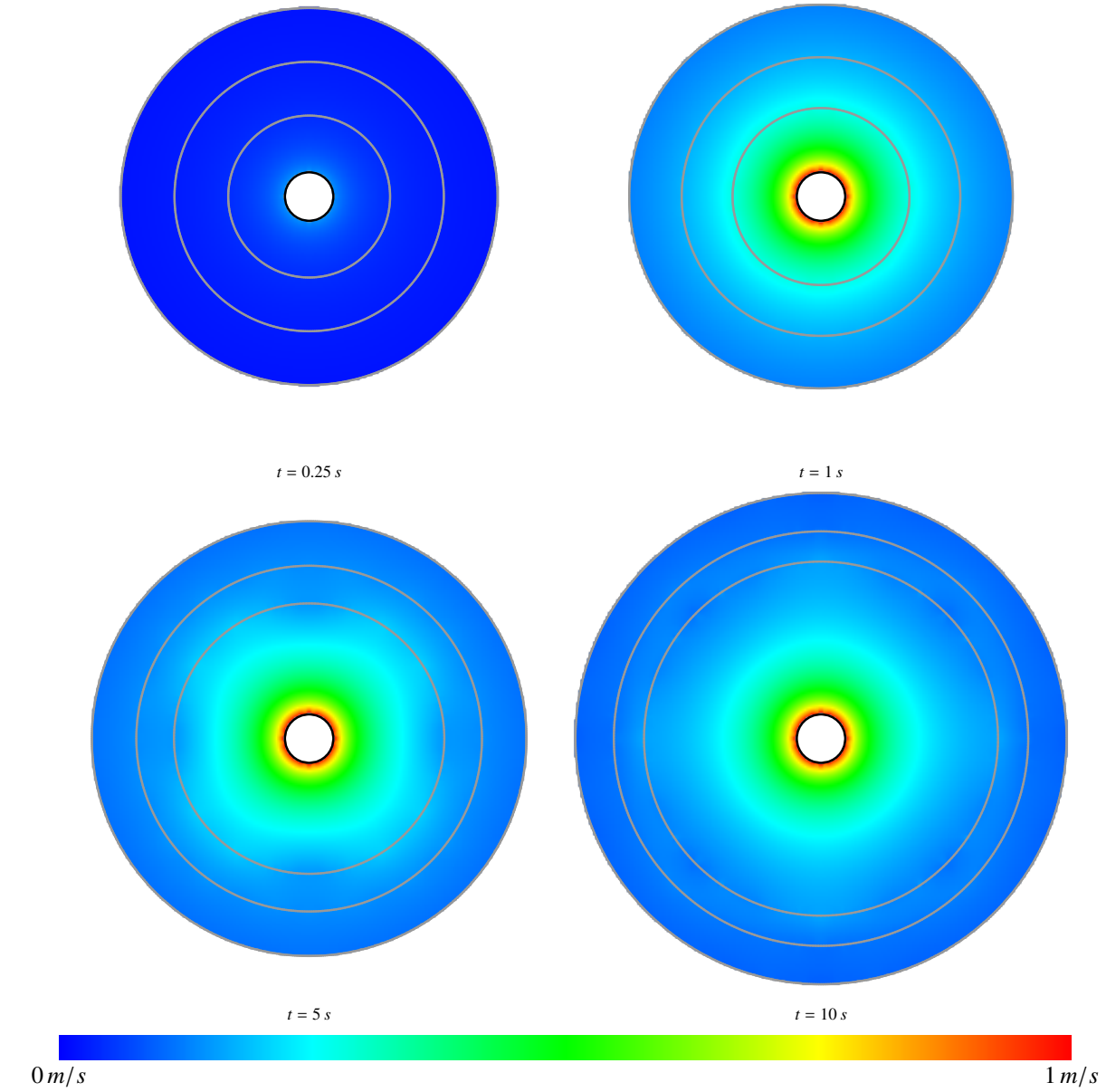


Fig. 35: Snapshots from the nested balloons example at various states of the simulation. The color map illustrates the fluid velocity field magnitude.

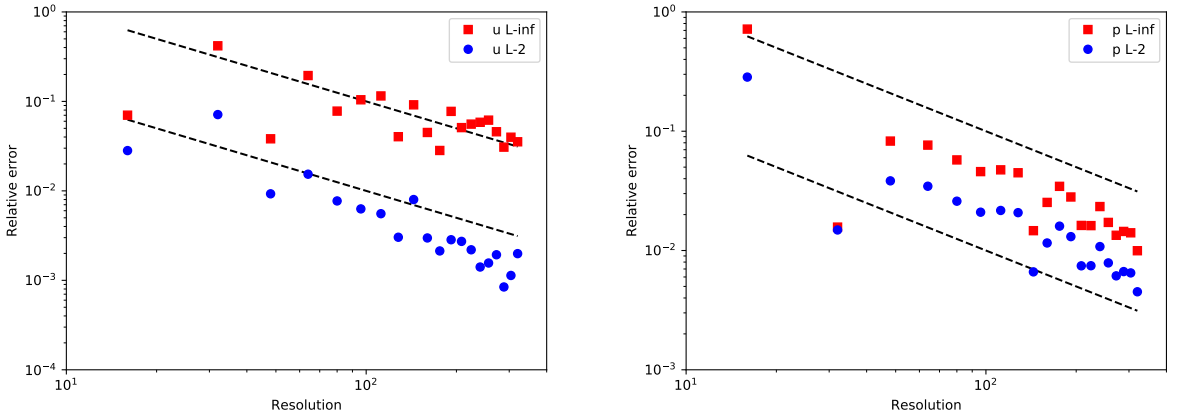


Fig. 36: The pressure and velocity for the nested rings example exhibit first-order convergence in both L_2 and L_∞. The dashed lines represent first order convergence.

so that within each fluid region R, the pressure is

$$p_R(\mathbf{x}) = -\frac{1}{2}\rho c^2 \|\mathbf{x}\|^{-2} + k_R. \tag{45}$$

where k_R is a per-region constant determined from the ambient pressure $p_\infty = 0$ and pressure jump conditions derived next.

The solid rings are modeled with stretching and bending springs with Young's moduli E_s and E_b, respectively. For a solid ring with initial radius r_0 and current radius r, the stretching potential energy is given by $\frac{E_s \pi}{r_0}(r - r_0)^2$. The bending springs connect pairs of nodes two apart (every other node), resulting in a mesh connecting all odd nodes and a mesh connecting all even nodes. The energy associated with bending is thus twice what would be predicted for the stretching case, and the total elastic potential energy of a solid ring is

$$\Phi(r) = (E_s + 2E_b)\pi \frac{(r - r_0)^2}{r_0}. \tag{46}$$

A uniform pressure jump $[p]$ acts on the solid to shift it from r to $r + \Delta r$, by exerting a force of $2\pi r[p]$ over a distance of Δr, which takes work $2\pi r[p]\Delta r$. The potential energy change is $\Phi(r + \Delta r) - \Phi(r)$, so that $2\pi r[p] = \Phi'(r)$. The pressure jump across the solid at r is therefore

$$[p] = (E_s + 2E_b)\left(\frac{1}{r_0} - \frac{1}{r}\right). \tag{47}$$

If the solids are at r_1, r_2, and r_3 from outside to inside, with corresponding initial radii r_{01}, r_{02}, and r_{03}, then the pressure jumps are given by

$$[p]_1 = p_1(r_1) - p_\infty = (E_s + 2E_b)\left(\frac{1}{r_{01}} - \frac{1}{r_1}\right) \tag{48}$$

$$[p]_2 = p_2(r_2) - p_1(r_2) = (E_s + 2E_b)\left(\frac{1}{r_{02}} - \frac{1}{r_2}\right) \tag{49}$$

$$[p]_3 = p_3(r_3) - p_2(r_3) = (E_s + 2E_b)\left(\frac{1}{r_{03}} - \frac{1}{r_3}\right), \tag{50}$$

from which k_1, k_2, and k_3 can be determined.

For the analytic test, we use $\tau_0 = 10^{-2}$ and enforce $\triangle t/dx = 8 \times 10^{-3}$ during refinement. The results of the convergence study are shown in Fig. 36 and indicate first-order convergence of the velocity and pressure in both L_2 and L_∞.

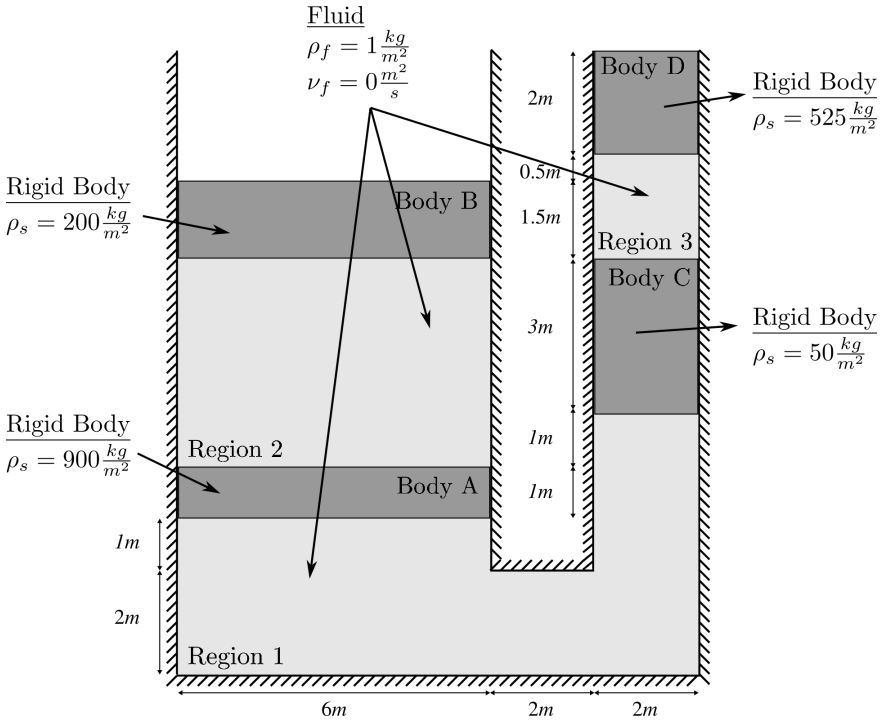


Fig. 37: Setup of the hydraulic press with multiple sections at equilibrium.

4.10. Multi-section Press

In this test, we study a hydraulic press with multiple Dirichlet regions coupled through rigid bodies as shown in Fig. 37. In this example, two rigid bodies are placed in various locations of each cylinder of a $10\,m \times 12\,m$ press, while the regions between the rigid bodies are filled with fluid. The left cylinder has a diameter of $6\,m$, and the right cylinder has a diameter of $2\,m$. The height and width of the bottom tube are both $2\,m$. Different densities and sizes are chosen for each rigid body as illustrated in Fig. 37. The system is designed to be in equilibrium at the initial state and expected to remain stable throughout the simulation since no external forces, other than the gravitational acceleration of $-9.8\,m/s^2$ that acts on both the fluid and solid, are applied. Since the solids are more dense than the fluid in this example, we eliminate the horizontal and rotational degrees of freedom for the rigid bodies to stabilize the dynamics and the equilibrium.

The fluid medium between the solids has a density of $1\,kg/m^2$ and is inviscid, while the cell size of the fluid grid is $dx = 0.02\,m$ in all regions. We use $\tau_0 = 1$ for this test. As with the other static tests, we omit relaxation.

The example is run until time $t \approx 16\,s$. The fluid solver pressure field without the three constant pressure modes is shown in Fig. 38. Upon examination of the figure, one can observe that all pressure fields are determined up to different arbitrary constant pressure modes by the fluid solver. Our method not only computes the pressures required for the rigid bodies to remain in equilibrium, but also compensates for the arbitrary constant modes generated by the fluid solver.

The constant pressure modes calculated by our method throughout the simulation are plotted in Fig. 39, where Region 1 is the bottom region, Region 2 is the left region, and Region 3 is the right region

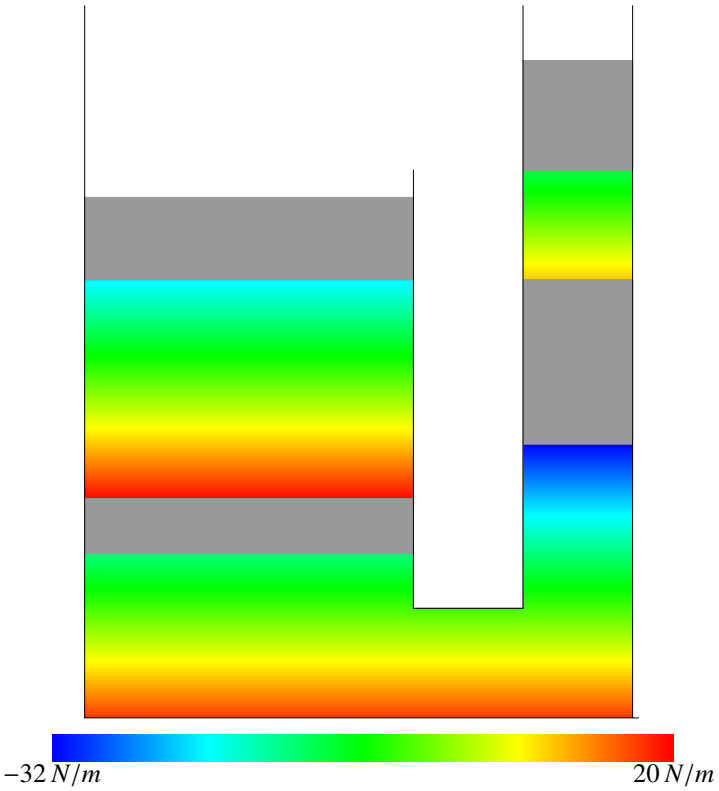


Fig. 38: The fluid solver pressure fields at the end of the simulation for the multi-section hydraulic press example. Note that the fluid solver does not determine the constant pressure modes, and only the fluid solver pressure field $\tilde{p}$ in each region is shown.

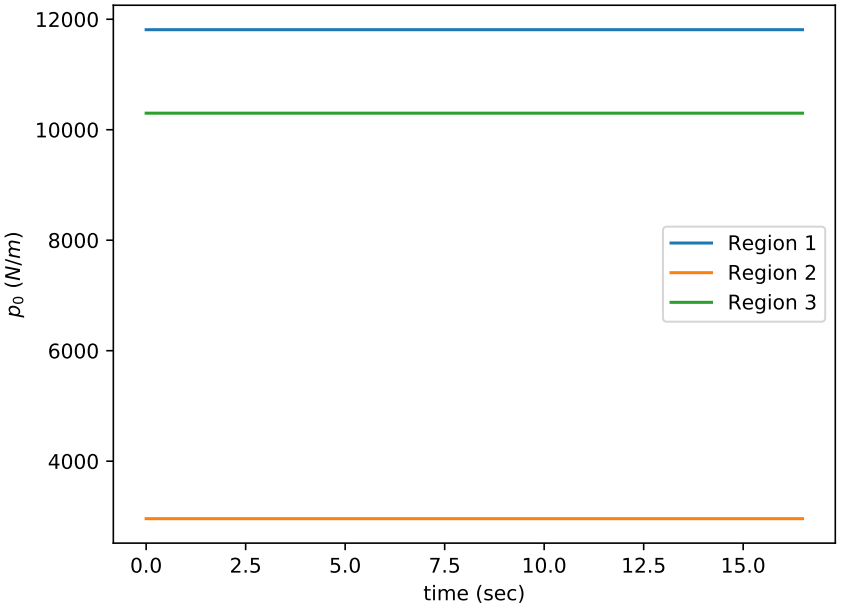


Fig. 39: Constant pressure mode, p_0, for each region plotted against time during the simulation of the multi-section press example as determined by the BPP. Region 1: bottom region (11810.862), Region 2: left region (2959.404), Region 3: right region (10299.604).

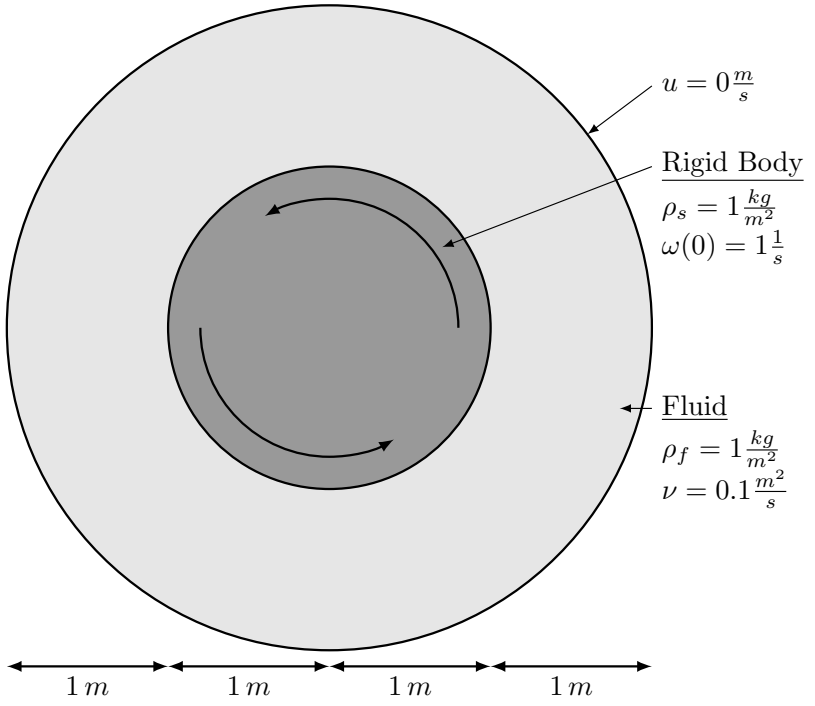


Fig. 40: Setup of the rotating disk example.

4.11. Rotating Disk

We study the convergence of our partitioned solver on an example of a rotating rigid disk inside a ring of incompressible fluid from [4, 5], as depicted in Fig. 40. For completeness, we include the analytic solution here. The fluid velocity and pressure are given in polar coordinates by

$$u_\theta(r,t) = \alpha_b e^{-\lambda^2 \nu t} \frac{J_1(\lambda r)Y_1(\lambda r_o) - J_1(\lambda r_o)Y_1(\lambda r)}{J_1(\lambda r_i)Y_1(\lambda r_o) - J_1(\lambda r_o)Y_1(\lambda r_i)}, \quad u_r(r,t) = 0, \tag{51}$$

$$p(r,t) = \int_{r_i}^{r} \frac{u_\theta(r,t)^2}{s}\, ds + p_{r_i}, \tag{52}$$

where J_1 and Y_1 are the Bessel functions of order 1, α_b is the initial angular velocity of the rigid body at r_i, p_{r_i} is an arbitrary constant, and λ satisfies the eigenvalue problem given in [4, 5]. The angular velocity of the rigid body is

$$\omega(t) = \frac{\alpha_b}{r_i} e^{-\lambda^2 \nu t}. \tag{53}$$

We use $\omega(0) = 1\ s^{-1}$, $r_i = 1\,m$, $r_o = 2\,m$, $\nu = 0.1\,m^2/s$, and $\lambda = 1.97045369767466$. The solid and fluid have densities $\rho_s = \rho_f = 1\,kg/m^2$. We remove the translational degrees of freedom from the rigid body since we are only interested in the rotational dynamics. We use a tighter pressure tolerance of $\tau_p = 10^{-5}$ to help maintain convergence at higher resolutions. We enforce $\triangle t/dx = 0.032$ during refinement.

As shown in Fig. 41, our partitioned solver exhibits first-order convergence of velocity in L_2 and L_∞. The pressure converges to first-order in L_2 and half-order in L_∞. The reduced order in convergence of pressure in L_∞ is due to the voxelized pressure discretization of our fluid solver. Note that this example has an arbitrary pressure constant which does not affect the dynamics of the simulation. Equation (13) is always satisfied, and the matrix in Eq. (21) is thus singular. As such, we do not do a boundary pressure projection for this example and do not require a τ_0 tolerance. Nevertheless, this test is a useful comparison against published work and is a generally good test of our handling of pressure and viscosity coupling with solids.

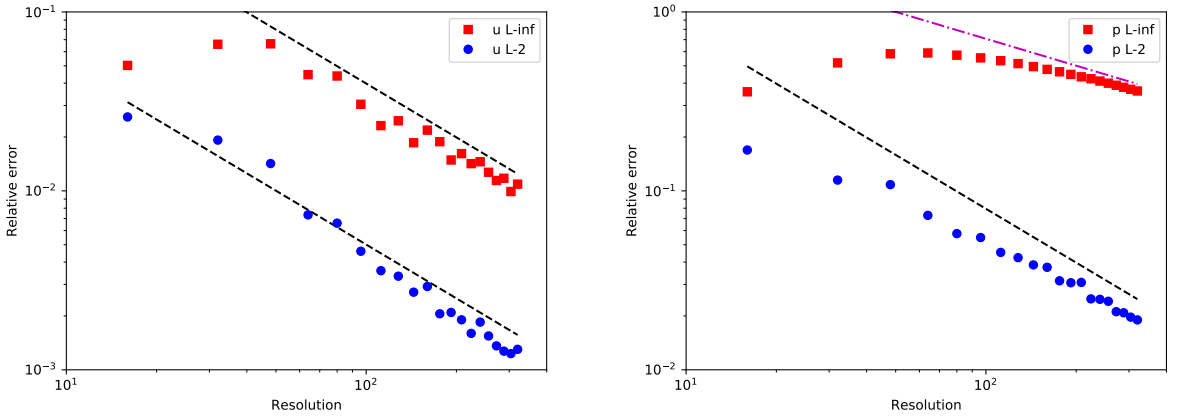


Fig. 41: The velocity for the rotating disk example exhibits first-order convergence in both L_2 and L_∞. The pressure exhibits first-order convergence in L_2 and half-order convergence in L_∞. The dashed black lines represent first order convergence, and the dash-dot magenta line represents half-order convergence.

5. Conclusion

We have a presented the BPP method for computing the constant pressure modes for coupled Dirichlet regions in a partitioned scheme for fluid-structure interaction problems. The method addresses the so-called incompressibility dilemma without requiring modifications to the solid solver, fluid solver, or Dirichlet-Neumann decomposition. We have demonstrated the method on several examples with multiple coupled Dirichlet regions. Though not demonstrated in this paper, the BPP could also be combined with other partitioned solver acceleration techniques [1] and could be applied in weak coupling schemes where a fixed number of iterations are used.

6. Acknowledgements

This work was supported in part by National Science Foundation award IIS-1314813.

References

[1] Akbay, M., Nobles, N., Zordan, V., and Shinar, T. An extended partitioned method for conservative solid-fluid coupling. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 86.

[2] Badia, S., Nobile, F., and Vergara, C. Fluid–structure partitioned procedures based on robin transmission conditions. *Journal of Computational Physics 227*, 14 (2008), 7027–7051.

[3] Badia, S., Nobile, F., and Vergara, C. Robin–robin preconditioned krylov methods for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering 198*, 33-36 (2009), 2768–2784.

[4] Banks, J. W., Henshaw, W. D., and Schwendeman, D. W. An analysis of a new stable partitioned algorithm for fsi problems. part i: Incompressible flow and elastic solids. *Journal of Computational Physics 269* (2014), 108–137.

[5] Banks, J. W., Henshaw, W. D., and Schwendeman, D. W. An analysis of a new stable partitioned algorithm for fsi problems. part ii: Incompressible flow and structural shells. *Journal of Computational Physics 268* (2014), 399–416.

[6] Batty, C., Bertails, F., and Bridson, R. A fast variational framework for accurate solid-fluid coupling. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 100.

[7] Bazilevs, Y., Calo, V. M., Hughes, T. J., and Zhang, Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational mechanics 43*, 1 (2008), 3–37.

[8] Bazilevs, Y., Hsu, M.-C., and Scott, M. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering 249* (2012), 28–41.

[9] Bogaers, A. E., Kok, S., Reddy, B. D., and Franz, T. Extending the robustness and efficiency of artificial compressibility for partitioned fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering 283* (2015), 1278–1295.

[10] Cetin, A., and Sahin, M. A monolithic fluid-structure interaction framework applied to red blood cells. *International journal for numerical methods in biomedical engineering 35*, 2 (2019), e3171.

[11] Chorin, A. J. A numerical method for solving incompressible viscous flow problems. *Journal of computational physics 2*, 1 (1967), 12–26.

[12] Degroote, J. Partitioned simulation of fluid-structure interaction. *Archives of computational methods in engineering 20*, 3 (2013), 185–238.

[13] Deparis, S., Discacciati, M., Fourestey, G., and Quarteroni, A. Fluid–structure algorithms based on steklov–poincaré operators. *Computer Methods in Applied Mechanics and Engineering 195*, 41-43 (2006), 5797–5812.

[14] Eken, A., and Sahin, M. A parallel monolithic algorithm for the numerical simulation of large-scale fluid structure interaction problems. *International Journal for Numerical Methods in Fluids 80*, 12 (2016), 687–714.

[15] Farhat, C., Lesoinne, M., and Le Tallec, P. 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 (1998), 95–114.

[16] Fernández, M. A., Landajuela, M., and Vidrascu, M. Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction. *Journal of Computational Physics 297* (2015), 156–181.

[17] Fernández, M. A., Mullaert, J., and Vidrascu, M. Explicit robin–neumann schemes for the coupling of incompressible fluids with thin-walled structures. *Computer Methods in Applied Mechanics and Engineering 267* (2013), 566–593.

[18] Fernández, M. A., Mullaert, J., and Vidrascu, M. Generalized robin–neumann explicit coupling schemes for incompressible fluid-structure interaction: Stability analysis and numerics. *International Journal for Numerical Methods in Engineering 101*, 3 (2015), 199–229.

[19] Gerbeau, J.-F., and Vidrascu, M. A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis 37*, 4 (2003), 631–647.

[20] Gibou, F., and Min, C. Efficient symmetric positive definite second-order accurate monolithic solver for fluid/solid interactions. *Journal of Computational Physics 231*, 8 (2012), 3246–3263.

[21] González, J., and Park, K. Treatment of non-matching interfaces in partitioned fluid–structure interaction problems. *Advances In Computational Coupling And Contact Mechanics 11* (2018), 145.

[22] Harlow, F. H., and Welch, J. E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids 8*, 12 (1965), 2182–2189.

[23] He, T., Zhang, K., and Wang, T. Ac-cbs-based partitioned semi-implicit coupling algorithm for fluid-structure interaction using stabilized second-order pressure scheme. *Communications in Computational Physics 21*, 5 (2017), 1449–1474.

[24] Hirschvogel, M., Bassilious, M., Jagschies, L., Wildhirt, S. M., and Gee, M. W. A monolithic 3d-0d coupled closed-loop model of the heart and the vascular system: Experiment-based parameter estimation for patient-specific cardiac mechanics. *International journal for numerical methods in biomedical engineering 33*, 8 (2017), e2842.

[25] Joosten, M., Dettmer, W., and Perić, D. Analysis of the block gauss–seidel solution procedure for a strongly coupled model problem with reference to fluid–structure interaction. *International Journal for Numerical Methods in Engineering 78*, 7 (2009), 757–778.

[26] Kassiotis, C., Ibrahimbegovic, A., Niekamp, R., and Matthies, H. G. Nonlinear fluid–structure interaction problem. part i: implicit partitioned algorithm, nonlinear stability proof and validation examples. *Computational Mechanics 47*, 3 (2011), 305–323.

[27] Klöppel, T., Popp, A., Küttler, U., and Wall, W. A. Fluid–structure interaction for non-conforming interfaces based on a dual mortar formulation. *Computer Methods in Applied Mechanics and Engineering 200*, 45-46 (2011), 3111–3126.

[28] Klöppel, T., and Wall, W. A. A novel two-layer, coupled finite element approach for modeling the nonlinear elastic and viscoelastic behavior of human erythrocytes. *Biomechanics and modeling in mechanobiology 10*, 4 (2011), 445–459.

[29] Küttler, U., Förster, C., and Wall, W. A. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure dirichlet fluid domains. *Computational Mechanics 38*, 4-5 (2006), 417–429.

[30] Küttler, U., and Wall, W. A. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational mechanics 43*, 1 (2008), 61–72.

[31] Le Tallec, P., and Mouro, J. Fluid structure interaction with large structural displacements. *Computer methods in applied mechanics and engineering 190*, 24-25 (2001), 3039–3067.

[32] Long, C., Marsden, A., and Bazilevs, Y. Fluid–structure interaction simulation of pulsatile ventricular assist devices. *Computational Mechanics 52*, 5 (2013), 971–981.

[33] Martínez-Ferrer, P. J., Qian, L., Ma, Z., Causon, D. M., and Mingham, C. G. An efficient finite-volume method to study the interaction of two-phase fluid flows with elastic structures. *Journal of Fluids and Structures 83* (2018), 54–71.

[34] Matthies, H. G., and Steindorf, J. Partitioned strong coupling algorithms for fluid–structure interaction. *Computers & structures 81*, 8-11 (2003), 805–812.

[35] Ng, Y. T., Min, C., and Gibou, F. An efficient fluid–solid coupling algorithm for single-phase flows. *Journal of Computational Physics 228*, 23 (2009), 8807–8829.

[36] Raback, P., Ruokolainen, J., Lyly, M., and Järvinen, E. Fluid-structure interaction boundary conditions by artificial compressibility. In *ECCOMAS CFD* (2001), vol. 2001.

[37] Robinson-Mosher, A., Schroeder, C., and Fedkiw, R. A symmetric positive definite formulation for monolithic fluid structure interaction. *Journal of Computational Physics 230*, 4 (2011), 1547–1566.

[38] Robinson-Mosher, A., Shinar, T., Gretarsson, J., Su, J., and Fedkiw, R. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 46:1–46:9.

[39] Shinar, T., Schroeder, C., and Fedkiw, R. Two-way coupling of rigid and deformable bodies. In *SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2008), pp. 95–103.

[40] Tezduyar, T. E., and Sathe, S. Modelling of fluid–structure interactions with the space–time finite elements: solution techniques. *International Journal for Numerical Methods in Fluids 54*, 6-8 (2007), 855–900.

[41] van Opstal, T., and van Brummelen, E. A finite-element/boundary-element method for large-displacement fluid–structure interaction with potential flow. *Computer Methods in Applied Mechanics and Engineering 266* (2013), 57–69.

[42] van Opstal, T., van Brummelen, E., and van Zwieten, G. A finite-element/boundary-element method for three-dimensional, large-displacement fluid–structure-interaction. *Computer Methods in Applied Mechanics and Engineering 284* (2015), 637–663.

[43] Wall, W. A., Mok, D. P., and Ramm, E. Partitioned analysis approach of the transient coupled response of viscous fluids and flexible structures. In *Solids, structures and coupled problems in engineering, proceedings of the European conference on computational mechanics ECCM* (1999), vol. 99, p. 182.