

Numerically Stable Fluid-Structure Interactions Between Compressible Flow and Solid Structures

Jón Tómas Grétarsson*, Nipun Kwatra**, Ronald Fedkiw*

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

Abstract

We propose a novel method to implicitly two-way couple Eulerian compressible flow to volumetric Lagrangian solids. The method works for both deformable and rigid solids and for arbitrary equations of state. The method exploits the formulation of [11] which solves compressible fluid in a semi-implicit manner, solving for the advection part explicitly and then correcting the intermediate state to time t^{m+1} using an implicit pressure, obtained by solving a modified Poisson system. Similar to previous fluid-structure interaction methods, we apply pressure forces to the solid and enforce a velocity boundary condition on the fluid in order to satisfy a no-slip constraint. Unlike previous methods, however, we apply these coupled interactions implicitly by adding the constraint to the pressure system and combining it with any implicit solid forces in order to obtain a strongly coupled, symmetric indefinite system (similar to [17], which only handles incompressible flow). We also show that, under a few reasonable assumptions, this system can be made symmetric positive-definite by following the methodology of [16]. Because our method handles the fluid-structure interactions implicitly, we avoid introducing any new time step restrictions and obtain stable results even for high density-to-mass ratios, where explicit methods struggle or fail. We exactly conserve momentum and kinetic energy (thermal fluid-structure interactions are not considered) at the fluid-structure interface, and hence naturally handle highly non-linear phenomenon such as shocks, contacts and rarefactions.

1. Introduction

Direct numerical simulations (DNS) are often used to study the interactions between fluid flows and solid structural models. Under certain assumptions these can be reduced to a one-way coupled system; for example if one wishes to determine the steady-state lift of an airfoil in subsonic flow, it is often reasonable to simulate the airfoil as a kinematic body. With a clever choice of boundary conditions, one can even begin to examine two-way coupled interactions, albeit in a limited fashion. In the more general case, these assumptions miss the interesting two-way coupled interactions between the fluid and the structure. These two-way coupled interactions can be quite important and, if not properly captured in the DNS, can lead to non-physical results. It is therefore important to have a robust numerical method that accurately captures two-way coupled interactions across a fluid-structure interface.

Methods to capture fluid-structure interactions can be broadly separated into two categories. Weakly coupled (partitioned) systems interleave the disparate subsystems by integrating them forward in time separately, using each others' results as boundary conditions in an alternating one-way coupled fashion (see e.g. [21, 15, 6]). This approach is appealing as it permits the use of specialized numerical methods for each of the different materials with only slight modifications to account for the modified time integration and changing boundaries. There are disadvantages to this approach, however, for example new and poorly understood stability restrictions arise independent of the individual subsystems, such as the lumped-mass instability discussed in [4]. The alternative is to employ a strongly coupled (monolithic) system, which are systems

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

**kwatra@stanford.edu, Stanford University

where the fluid and structure are evolved forward in time simultaneously using a solver specially crafted to incorporate phenomena from both fluid and solid phases. Our method is a hybrid of the two; the explicit components of both fluid and solid solvers are evolved forward independently, while the implicit components and interactions are coupled together in a monolithic solve.

State-of-the-art solvers typically use an Eulerian framework to treat fluid flows and a Lagrangian framework to treat solids, and so any coupled system must do one of three things: model the solid in an Eulerian framework, model the fluid in a Lagrangian framework, or find a way to couple Eulerian fluids with Lagrangian solids. The first two options are undesirable as they impose significant limitations on the numerical method, for example Eulerian models only capture material properties (rather than tracking them) which makes it difficult to compute time history variables important to structural simulation, such as loading and damage. Many fluid Lagrangian models have difficulty in obtaining the correct shock speeds due to the lack of discrete flux differencing, and therefore resort to artificial viscosity methods that require a number of zones within a shock in order to obtain the right speed [2, 3]. Lagrangian fluid models also struggle with high-speed and deforming flows, as large deformations can cause significant numerical errors in the flow field and can drive the time step to zero. This can be partially alleviated by applying complex and expensive remeshing, but if the flow field tangles and inverts, the simulation can cease altogether. Arbitrary Lagrange-Eulerian (ALE) methods address the problem of a deforming Lagrangian fluid grid by permitting the fluid grid to move at some velocity other than the velocity of the fluid, but this can still lead to high aspect ratios that necessitate remeshing, especially in the presence of a fluid-structure interface. We address the challenge of coupling Eulerian fluids with Lagrangian solids by introducing an interpolation operator, which conservatively maps quantities from Eulerian boundaries to nearby Lagrangian boundary nodes, and vice versa.

At the fluid-structure interface there is a transfer of information. This information transfer can be handled by weakly coupling each separate subsystem using a one-sided estimate of the transfer, or by strongly coupling subsystems together and introducing new variables to the equations. Weakly coupled approaches have been shown to give high-fidelity results [1, 8, 7], but can struggle when applied to a system with high density-to-mass ratios (and are prone to going unstable, as we discuss in Section 4.3). These problems can be alleviated by using a better estimate of values at the interface, as suggested by [12], but this typically involves solving expensive general Riemann problems at every fluid-structure face. These problems can be avoided entirely by handling the interface in a strongly coupled fashion, but previous work has been limited to incompressible flows [14, 17]. Our method exploits the structure of [11], which treats the pressure flux of compressible flows implicitly. This permits us to treat the fluid pressure as an implicit force on the solid, and use an implicit velocity boundary condition on the Poisson solve, much like previous strongly-coupled work.

Our fluid evolution is comprised of two steps: an advection stage and a pressure solver phase. This permits us to address the complexities arising from the truly non-linear components of the flow separately from the linearly degenerate components. In the pressure phase, we freeze everything to their time t^{n+1} location and perform an implicit solve for the fluid pressure and solid velocity. It is in this phase that we handle the transfer of momentum and kinetic energy across the fluid-structure interface, and as such it is important to be conservative in transferring information between the two sets of degrees of freedom. In the advection stage no information should be transmitted across the interface, but instead we must address the issues which arise by virtue of a moving solid (i.e. the covering and uncovering of fluid cells). There are many examples of how to address these problems in the literature, for example we could track cut cells, re-discretize the fluid in an ALE formulation—all of which significantly complicate the fluid evolution. Instead we make the key observation that since the interface is a contact discontinuity we can afford to be non-conservative, but only in the linearly degenerate components of the flow.

In a traditional explicit method the linearly degenerate and truly non-linear fluxes aren't separated, and as such these methods need to deal with all of the complexities of moving boundaries and information transmission at the same time. That is, they need to be conservative when dealing with information that crosses the interface while at the same time dealing with an interface that moves. Finally, the flux needs to be re-examined carefully in order to determine what forces should be applied to the interface. One could modify traditional methods by separating the conserved quantities into their Riemann invariants, and be conservative

in the truly non-linear invariants while allowing the linearly degenerate invariants to be non-conservative—however this doesn't address the moving boundary, and still leaves us with the (poorly-understood) CFL restriction that arises from explicit fluid-structure interactions. Because of these complications, our method hinges on the existence of [11].

2. Semi-implicit compressible flow

We briefly describe the semi-implicit evolution for compressible flow [11] which forms the basis for our implicit coupling scheme. Consider the multi-dimensional Navier-Stokes equations, given by:

$$\begin{pmatrix} \rho \\ \rho \vec{u} \\ E \end{pmatrix}_t + \begin{pmatrix} \nabla \cdot \rho \vec{u} \\ \nabla \cdot (\rho \vec{u}) \vec{u} \\ \nabla \cdot (E \vec{u}) \end{pmatrix} + \begin{pmatrix} 0 \\ \nabla p \\ \nabla \cdot (p \vec{u}) \end{pmatrix} = \mathbf{f} \quad (1)$$

where we have split the flux terms into an advection and non-advection part and lumped viscous terms into \mathbf{f} . The advection part (as well as any body forces) is integrated explicitly to give intermediate values ρ^* , $(\rho \vec{u})^*$ and E^* . Since pressure does not affect the continuity equation, $\rho^{n+1} = \rho^*$. The momentum update equation can be divided by ρ^{n+1} to obtain

$$\vec{u}^{n+1} = \vec{u}^* - \Delta t \frac{\nabla p}{\rho^{n+1}}, \quad (2)$$

and taking its divergence gives

$$\nabla \cdot \vec{u}^{n+1} = \nabla \cdot \vec{u}^* - \Delta t \nabla \cdot \left(\frac{\nabla p}{\rho^{n+1}} \right). \quad (3)$$

In the case of incompressible flow, we would set $\nabla \cdot \vec{u}^{n+1} = 0$, but for compressible flow we instead use the pressure evolution equation (see e.g. [9]),

$$p_t + \vec{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \vec{u}. \quad (4)$$

If we fix $\nabla \cdot \vec{u}$ to be at time t^{n+1} through the time step (making an $\mathcal{O}(\Delta t)$ error), discretize $p_t + \vec{u} \cdot \nabla p$ explicitly using a forward Euler time step (i.e. $\frac{p^{n+1} - p^n}{\Delta t} + \vec{u}^n \cdot \nabla p^n$), and define the advected pressure as $p^a = p^n - \Delta t (\vec{u}^n \cdot \nabla p^n)$ we obtain

$$p^{n+1} = p^a - \Delta t \rho c^2 \nabla \cdot \vec{u}^{n+1}. \quad (5)$$

Substituting this in Equation (3) and rearranging gives

$$p^{n+1} - \rho^n (c^2)^n \Delta t^2 \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = p^a - \rho^n (c^2)^n \Delta t \nabla \cdot \vec{u}^*, \quad (6)$$

where we have defined ρc^2 at time t^n and the pressure p at time t^{n+1} . Discretizing the gradient and divergence operators yields

$$\left[I + \rho^n (c^2)^n \Delta t^2 G^T \left(\frac{1}{\hat{\rho}^{n+1}} G \right) \right] p^{n+1} = p^a + \rho^n (c^2)^n \Delta t G^T \vec{u}^*, \quad (7)$$

where G is our discretized gradient operator, $-G^T$ is our discretized divergence operator, and $\hat{\rho}$ and \hat{u} represent variables interpolated to cell faces. This is solved to obtain p^{n+1} at cell centers. The time t^{n+1} pressures are then applied in a flux-based manner to the intermediate momentum and energy values to obtain time t^{n+1} quantities in a discretely conservative manner (thereby giving correct shock speeds). This is done by averaging the pressures to cell faces by $p_{i+1/2}^{n+1} = \frac{p_{i+1}^{n+1} \rho_i^{n+1} + p_i^{n+1} \rho_{i+1}^{n+1}}{\rho_i^{n+1} + \rho_{i+1}^{n+1}}$, rewriting Equation (2) using

face-averaged quantities $\hat{u}_{i+1/2} = \hat{u}_{i+1/2}^* - \Delta t \frac{G_{i+1/2} p^{n+1}}{\hat{\rho}_{i+1/2}}$ (where $\hat{\rho}_{i+1/2} = (\rho_i + \rho_{i+1})/2$), and updating the values using

$$(\rho \bar{u})^{n+1} = (\rho \bar{u})^* - \Delta t \left(\frac{p_{i+1/2}^{n+1} - p_{i-1/2}^{n+1}}{\Delta x} \right), \quad E^{n+1} = E^* - \Delta t \left(\frac{(p\hat{u})_{i+1/2}^{n+1} - (p\hat{u})_{i-1/2}^{n+1}}{\Delta x} \right). \quad (8)$$

3. Solid evolution

We give a brief treatment of solid evolution with sufficient detail to properly handle the fluid-structure interactions. A solid state is completely described by its velocity and position. We update the position and velocities in a Newmark scheme in which velocity at time $t^{n+1/2}$ is used to update the position to time t^{n+1} in a second order update. Velocity is then updated from time t^n to time t^{n+1} in a separate step. We describe below the velocity update for deformable and rigid solids. The same procedure is used twice, once with a time step of $\Delta t/2$ to obtain $V^{n+1/2}$ for position update and then with a time step of Δt for the final velocity update.

Deformable body formulation: For deformable body evolution we need to handle both elastic and damping forces. Damping forces can impose strict time step restrictions and are thus treated implicitly. We will describe a method which treats the elastic forces explicitly and damping forces implicitly although one could also incorporate implicit elasticity. The deformable body at a given time t can be described by a vector of positions of its nodes $X_s(t)$ and a vector of velocities of its nodes $V_s(t)$. The evolution of velocities can be described by Newton's second law as

$$M_s(V_s)_t = F(X_s, V_s), \quad (9)$$

where M_s is the mass matrix and F is the vector of all forces acting on the solid nodes. Discretizing and computing the elastic terms explicitly and damping terms implicit in position, but implicit in velocity, i.e. $F(X_s, V_s) = F(X_s^n, V_s^{n+1})$, we obtain

$$M_s V_s^{n+1} = M_s V_s^n + \Delta t F(X_s^n, V_s^{n+1}). \quad (10)$$

Using a Taylor series expansion on F yields

$$M_s V_s^{n+1} = M_s V_s^n + \Delta t (F(X_s^n, V_s^n) + D(V_s^{n+1} - V_s^n)). \quad (11)$$

where $D = \frac{\partial F}{\partial V_s}$. $F(X_s^n, V_s^n) - DV_s^n$ represents the elastic only (and, if present, any non-linear damping terms [19]) component of the force and one can write

$$M_s V_s^{n+1} = M_s V_s^* + \Delta t DV_s^{n+1}, \quad (12)$$

where V_s^* denotes the velocity vector updated explicitly with the elastic terms only.

Rigid body formulation: For a rigid body we define the generalized velocity vector as $V_s = (V_{cm}^T, \omega^T)^T$, where V_{cm} is the velocity of its center of mass and ω is its angular velocity. The velocity evolution can then be described as

$$\begin{pmatrix} M_r & 0 \\ 0 & I_r \end{pmatrix} (V_s)_t = \begin{pmatrix} f \\ \tau \end{pmatrix}, \quad (13)$$

where M_r is a 3×3 diagonal matrix with the rigid body mass in the diagonals, I_r is the inertia tensor and f, τ are the net force and torque acting on it. Writing the mass matrix as M_s and combining f, τ into F , we get a form similar to (9) which can be discretized using forward Euler to obtain

$$M_s V_s^{n+1} = M_s V_s^n + \Delta t F^n = M_s V_s^*. \quad (14)$$

Where V_s^* denotes the velocity vector updated with the explicit forces. Note that this is the same as Equation (12) except without any damping term. We will therefore use Equation (12) as our general solid update equation below, as it covers both the rigid and deformable cases.

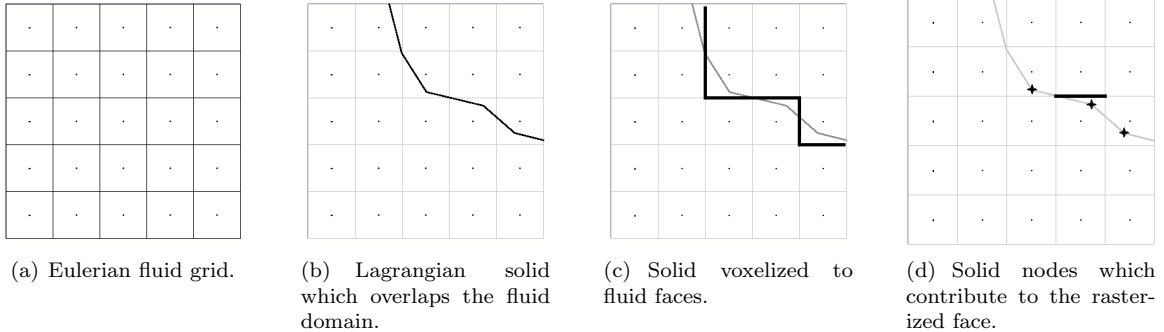


Figure 1: A common challenge with FSI problems is one of overlapping grids. We resolve this issue by voxelizing solid degrees of freedom to the fluid grid using an interpolation operator denoted by the matrix W . The row corresponding to a fluid face gets contributions from nearby solid nodes.

4. Fluid-structure interaction

We solve for the fluid on an Eulerian grid, and the solids on freely deforming Lagrangian meshes. The fluid and structure interact with each other by applying equal and opposite forces at the interface, satisfying physical boundary conditions (we use no-slip, no penetration boundary conditions) in the process. Immersed boundary methods induce extra force variables at the interface and apply a regularization operator to map these forces to fluid faces (see e.g. [20]). They also incorporate an interpolation operator to map fluid velocity to solid nodes for applying boundary conditions. We eliminate the extra interface force variables and conservatively map the fluid pressures directly to solid nodes, and solid velocities to fluid faces using an interpolation operator.

Figure 1 illustrates an example fluid grid which is coupled to a Lagrangian solid which occupies the upper right-hand corner of the grid. In our model, the fluid interacts with a voxelized version of the solid and the solid directly sees forces acting on its nodes. We define an interpolation operator W which maps solid node velocity to the fluid cell faces, where the rows correspond to fluid faces and the columns to solid nodes. W can be constructed in a row-by-row fashion: for each row, we identify the corresponding fluid face and locate the nearby solid nodes. The entry corresponding to each solid node is populated by a weight proportional to its contribution to the fluid face, and then finally the row is normalized to ensure that each row sums to one, making it an interpolation. This is done in a component-by-component manner, e.g. the x -component of solid velocity is voxelized to x -axis fluid faces but not y - or z -axis fluid faces, and so the solid velocity at fluid face $i + 1/2$ is $(WV_s)_{i+1/2}$. Since pressure is defined at cell centers, we also introduce an extrapolation operator B which maps cell-centered pressure to face pressures, as illustrated in Figure 2. These face pressures are then multiplied by the surface area of the cell face to get a force and distributed back to solid nodes using W^T . That is, W maps from solid node degrees of freedom to cell faces, and W^T maps back in the opposite direction. Note that since the rows of W sum to one, the columns of W^T sum to one and therefore the force felt due to the pressure on the face is fully and conservatively distributed to the solid node degrees of freedom.

4.1. The strongly coupled system

The fluid acts on solid degrees of freedom via pressure along the interface. The pressure exerts a force given by $W^T A_f B p$ on the solid degrees of freedom, where A_f is a diagonal matrix whose entries correspond to the areas of fluid-structure faces. We can incorporate these forces into the implicit solid system given by Equation (12):

$$M_s V_s^{n+1} = M_s V_s^* + \Delta t D V_s^{n+1} + \Delta t W^T A_f B p. \quad (15)$$

The fluid sees a velocity boundary condition at the fluid-structure interface. To incorporate this into the fluid equations, we partition the discrete divergence operator $-G^T$ into two components. G_f^T operates

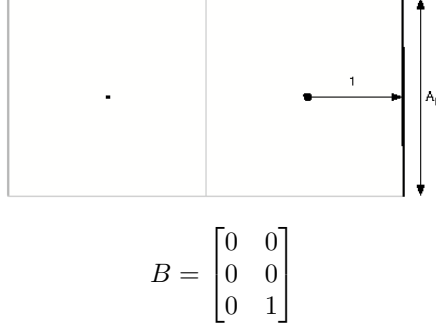


Figure 2: Operator B maps pressure from cell centers to bordering fluid-structure faces. In this example there are x -direction faces, of which the one to the far right represents a rasterized solid face. Therefore B has three rows (one for each vertical face, with the top and the bottom rows corresponding to the far left and far right vertical faces respectively, and the middle row corresponding to the middle vertical face), and two columns (one for each pressure at each cell center). Since the only contribution to the solid is from the second pressure to the third face, B has the form shown above with a single non-zero element. Note that $(1/dx)B^T$ equals $-G_s^T$, as defined in Figure 3(b).

over fluid-fluid faces, while G_s^T is the component of the divergence operator which operates on rasterized fluid-structure faces (as outlined in Figure 3), and $G^T = G_f^T + G_s^T$. We can then set fluid-structure faces to have implicit Neumann boundary conditions; that is,

$$\bar{u}^{n+1} = \begin{cases} \bar{u}^* - \Delta t \frac{G_f p}{\bar{\rho}} & \text{at a fluid-fluid face; and} \\ WV_s^{n+1} & \text{at a fluid-structure face.} \end{cases} \quad (16)$$

Taking the divergence of the velocity field yields

$$G^T \bar{u}^{n+1} = G_f^T \bar{u}^* - \Delta t G_f^T \frac{1}{\bar{\rho}} G p + G_s^T WV_s^{n+1} \quad (17)$$

Using this modified definition for $G^T \bar{u}^{n+1}$ in Equation (5) and substituting into Equation (3) gives

$$\left[\frac{1}{\Delta t \rho^n (c^n)^2} I + \Delta t G_f^T \frac{G}{\bar{\rho}^{n+1}} \right] p^{n+1} - G_s^T WV_s^{n+1} = \frac{p^a}{\Delta t \rho c^2} + G_f^T \bar{u}^*. \quad (18)$$

If we define $V = \Delta x \Delta y \Delta z$ to be the volume of the fluid cell, then $V G_s^T = A_f B^T$. Combining equations (15) and (18), using scaled pressure $\tilde{p} = \Delta t p$ and scaled advected pressure $\tilde{p}^a = \Delta t p^a$, and rescaling the fluid equations by cell volume gives us our symmetric system

$$\begin{pmatrix} \frac{V}{\Delta t^2 \rho c^2} I + V G_f^T \frac{1}{\bar{\rho}} G_f & -A_f B^T W \\ -W^T B A_f & -M_s + \Delta t D \end{pmatrix} \begin{pmatrix} \tilde{p}^{n+1} \\ V_s^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{V}{\Delta t^2 \rho c^2} \tilde{p}^a + V G_f^T \bar{u}^* \\ -M_s V_s^* \end{pmatrix}. \quad (19)$$

It is interesting to note that if we take the incompressibility assumption (i.e. $c \rightarrow \infty$) then this system reduces to one similar to [17].

The system in Equation (19) is symmetric but indefinite, and can be solved using efficient solvers such as Conjugate Residuals [13] to obtain the final time t^{n+1} solid velocity and pressure. The solid part of our update is now complete, but we still need to use the t^{n+1} pressure to update the fluid momentum and energy (noting that $\rho^{n+1} = \rho^*$ is already done).

4.2. Updating fluid momentum and energy

To obtain correct shock speeds we use the flux-based method discussed above, with modifications to account for fluid-structure faces. At a fluid-structure face $i + 1/2$, the fluid applied a force of $(B A_f p)_{i+1/2}$ to

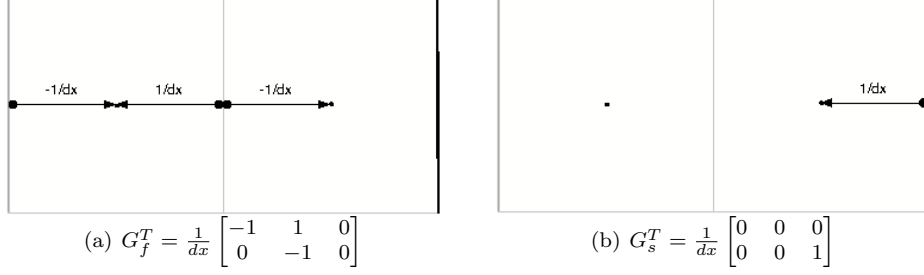


Figure 3: In our derivation, the divergence operator $-G^T$ is split into G_f^T (which operates only on fluid-fluid faces) and G_s^T (which operates only on fluid-structure faces). We show this splitting for a simple two cell example where the right-most face is a fluid-structure interface. The rows in the above matrices correspond to cells and columns to faces. The left most face corresponds to the first column of G_f^T and only has one non-zero element since it only borders one fluid cell. The middle face (which corresponds to the second column of G_f^T) contributes to both fluid cells and hence has two non-zero elements. The third column of G_f^T is zero, as the third face is a fluid-structure face and instead corresponds to G_s^T . Figure (b) depicts G_s^T , which is defined as $-(1/dx)B^T$ in Figure 2.

the solid. To conserve momentum, fluid face $i + 1/2$ should apply an equal and opposite force $-(BA_f p)_{i+1/2}$ on fluid cell i . In our momentum update this is numerically equivalent to setting $p_{i+1/2} = (Bp)_{i+1/2}$ at fluid-structure faces.

Next, we need to consider the work done by the fluid on the solid at a fluid-structure face. We are applying an impulse $\Delta t(BA_f p)_{i+1/2}$ on the solid, which is equivalent to applying a constant force over the interval Δt . In order to compute the work done on the solid system by a single force \vec{f} in the presence of other forces, we lump all forces acting on the solid into a vector \vec{F} and examine

$$\int_0^{\Delta t} \vec{f} \cdot V_s(t) dt = \int_0^{\Delta t} \vec{f} \cdot (V_s^n + M_s^{-1} \vec{F} t) dt = \Delta t \vec{f} \cdot \left[V_s^n + M_s^{-1} \vec{F} \frac{\Delta t}{2} \right] = \Delta t \vec{f} \cdot \left[\frac{V_s^n + V_s^{n+1}}{2} \right], \quad (20)$$

where we take advantage of \vec{F} and \vec{f} being constant over the interval. We are interested in calculating the work done by a single fluid face on the solid, so if we take $W_{i+1/2}^T$ to be the column vector which distributes the pressure from cell face $i + 1/2$ to the solid node degrees of freedom then $\vec{f} = W_{i+1/2}^T (BA_f p)_{i+1/2}$, and the work done on the solid by this face is exactly

$$\Delta t \left[W_{i+1/2}^T (BA_f p)_{i+1/2} \right]^T \left[\frac{V_s^n + V_s^{n+1}}{2} \right] = \Delta t \left[(BA_f p)_{i+1/2} \right] W_{i+1/2} \left[\frac{V_s^n + V_s^{n+1}}{2} \right]. \quad (21)$$

This, if $p_{i+1/2}$ is defined to be $(Bp)_{i+1/2}$ as suggested above in the momentum update, then we merely need to set $\vec{u}_{i+1/2} = (1/2)(W[V_s^n + V_s^{n+1}])_{i+1/2}$ in order to obtain a flux $p\vec{u}$ which exactly conserves the kinetic energy transferred.

4.3. Time step restriction

In our method fluid-structure interactions are handled implicitly and thus we avoid introducing any new time step restrictions. The time step is therefore determined by the minimum of the time steps imposed by the fluid and the structure. For the structure update the time step restriction is determined by the elastic part only, as damping terms are handled implicitly, while our semi-implicit fluid update imposes a time step restriction dependent only on its bulk velocity. The time step restriction imposed by the semi-implicit flow formulation in two spatial dimensions is

$$\frac{\Delta t}{2} \left(\frac{|u|_{max}}{\Delta x} + \frac{|v|_{max}}{\Delta y} + \sqrt{\left(\frac{|u|_{max}}{\Delta x} + \frac{|v|_{max}}{\Delta y} \right)^2 + 4 \frac{|p_x|}{\rho \Delta x} + 4 \frac{|p_y|}{\rho \Delta y}} \right) \leq 1, \quad (22)$$

and we refer the interested reader to [11], which motivates this formulation.

We note that the implicit fluid-structure coupling gives stable results even for very high density-to-mass ratios, where explicit methods struggle even when the CFL restrictions of both solid and fluid systems are obeyed. We explore this in example 6.1.1.

5. Unified time integration

We employ a time integration scheme which incorporates fluid evolution into a Newmark-style solid evolution scheme. The scheme works by computing an intermediate velocity for the solid $V_s^{n+1/2}$, and applying this in a second order update to get solid positions at time t^{n+1} . Velocities are then updated from time t^n to t^{n+1} (discarding intermediate values), and so two linear systems are solved.

In order to compute the intermediate solid velocity $V_s^{n+1/2}$, we begin by applying all explicit solid forces to the system, which gives $V_s^{n+1/2*}$. Explicit body forces such as gravity and viscosity are also applied to the fluid system, yielding $t^{n+1/2*}$ fluid quantities. The coupled system (19) is solved in order to obtain $X_s^{n+1} = X_s^n + \Delta t V_s^{n+1/2}$, and then the entire fluid state and all solid velocities are restored to their time t^n values.

These new positions are then used to compute an *effective* velocity for the solids, i.e. $(X_s^{n+1} - X_s^n)/\Delta t$. Using the effective velocity and then the time t^n position of the solid, we fill ghost cells. These ghost cells are used directly in the stencils of high-order methods, and provide a valid state for which to populate uncovered cells. In order to compute the ghost cell data at location \vec{x}_g , we begin by identifying the closest solid interface point \vec{x}_I , and reflecting across the interface. Density and pressure are interpolated to the reflected point $2\vec{x}_I - \vec{x}_g$ from neighboring cells and then copied to the ghost cell. The surface normal \vec{N} at the interface is used to decompose the velocity at the reflected point \vec{V}_r into its normal component $V_{rN} = \vec{V}_r \cdot \vec{N}$ and its tangential component $\vec{V}_{rT} = \vec{V}_r - V_{rN}\vec{N}$. In order to remain continuous with the effective velocity of the structure at the interface \vec{V}_I , V_{rN} is reflected across the interface, and so we compute $V_{gN} = 2\vec{V}_I \cdot \vec{N} - V_{rN}$. Tangential velocity is decoupled from the interface and thus we can use it directly, giving the final ghost cell velocity $\vec{V}_g = V_{gN}\vec{N} + \vec{V}_{rT}$.

Once ghost cells are filled, explicit body forces such as gravity and viscosity are integrated into the system, and the advection component of flux from Equation (1) is applied using a conservative flux-based method (see [11]). Explicit solid forces are applied in order to compute V_s^{n+1*} , and then the coupled system (19) is solved to obtain V_s^{n+1} and p^{n+1} . This pressure is applied as per Section 4.2 to obtain time t^{n+1} fluid quantities.

We also fill the ghost cells inside the solid using time t^{n+1} data from the fluid and solid velocities, as described above. Although none of our examples use these ghost values, if an explicit body force such as viscosity were to be applied, its stencil would require valid ghost cells to be defined. Note that these are valid as instantaneous ghost cells, whereas the ghost cells above use the *effective* solid velocity, which is the actual motion of the solid through the mesh. Practical experience shows that this can make a meaningful difference.

6. Examples and validation

In order to compare our results with previous methods, we implement an explicit coupling scheme which integrates a fully explicit compressible flow evolution with a Newmark time integration for solids. This explicit method proceeds in a fashion similar to Section 5, except that instead of solving the system (19) we simply fill ghost cells inside the solid once and explicitly evolve the fluid once, while time t^n pressures along the fluid-structure interface are applied to the solid as explicit forces. This gives us an explicitly coupled time evolution scheme, such as the one described in [7].

Although one might assume that the implicit solve would cause efficiency bottlenecks, we observed relatively few Conjugate Residuals iterations per time step. This is likely due to the strongly diagonally dominant nature of Equation (19), and the good initial guess for pressure provided by the equation of state at time

t^n . For all of our one dimensional examples the maximum number of iterations required per time step was 3. For the two dimensional examples, the rigid body coupling example required a maximum of 4 iterations, while the deformable coupling example required a up to 24 iterations per time step.

In all of the examples we consider the fluid is simulated using an ideal gas law, with $\gamma = 1.4$.

6.1. One-dimensional validation

We examine several one dimensional fluid-structure interactions to validate our method. A third order ENO scheme [18] is used along with an advection-based CFL number of .6. All quantities below are in SI units, with density as kg/m^3 , pressure in Pa , lengths in m , spring coefficients in N/m , etc.

6.1.1. Sod shock coupled with a rigid body

Our first example is a Sod shock interacting with a rigid body, with open boundary conditions. The initial condition for the fluid is

$$(\rho(x, 0), u(x, 0), p(x, 0)) = \begin{cases} (1, 0, 1) & \text{if } x \leq .5, \\ (.125, 0, .1) & \text{if } x > .5. \end{cases}$$

A rigid body of mass 1 and width .2 starts at rest with its center of mass a distance of .8 from the left of the domain. The domain is of length 2. The rigid body remains at rest until the shock hits it, at which point it accelerates by virtue of the pressure difference. The solid body continues to accelerate until it converges to a velocity of .927453, which is precisely the interfacial velocity of the Sod Riemann problem. Figure 4 shows snapshots of the pressure profile at various times through the simulation. For comparison, results with the explicit method are shown in Figure 5. We also do a convergence analysis of our method in Figure 6. The error in the position of the rigid body is computed at time .9 from the highest resolution grid simulated, which is 6401 grid cells. The convergence order of the error is estimated as 1.6.

It is interesting to consider this simple problem for a variety of density-to-mass ratios. Figure 7(a) shows the velocity of the rigid body as a function of time for a range of rigid body masses in the semi-implicit case. Figure 7(b) shows this in the explicit case. We note that the explicit simulation struggles with high density-mass ratios. In particular it appears as though the rigid body gains too much momentum in a single time step, causing the fluid on the other side to over-compress, leading to a very stiff oscillatory system, even though the time step obeyed CFL restrictions. We show snapshots of the pressure profile of simulations with a light solid of mass .0001, with semi-implicit and explicit schemes in Figure 8 and Figure 9, respectively.

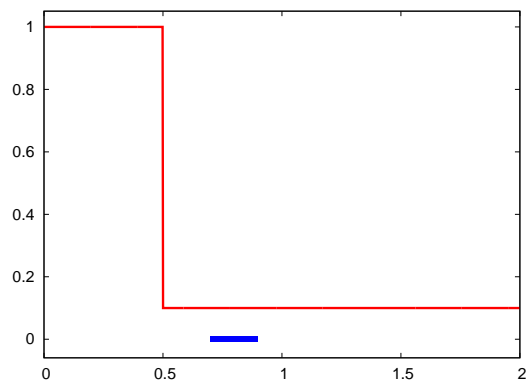
6.1.2. Sod shock interacting with a fluid piston

We consider a similar problem, this time with solid wall boundary conditions and a larger domain, with the initial discontinuity located at distance 1 from the left of the domain. The rigid body has a mass of 1, width .2 and starts at rest with its center of mass at 1.5 from the left of the domain. The domain is of length 3. The shock imparts momentum to the rigid body which in turn compresses the fluid on its right. This compressed fluid creates a high pressure region which pushes back on the solid, in effect creating a “fluid spring.” This causes the rigid body to oscillate as shown in Figure 10, which plots the position of the center of mass of the rigid solid as a function of time. Figure 11 shows snapshots of the pressure profile at various times through the simulation. For comparison, results with the explicit method are shown in Figure 12. We also do a convergence analysis of our method in Figure 13. The error in the position of the rigid body is computed at time 4s from the highest resolution grid simulated, which is 6401 grid cells. The convergence order of the error is estimated as 1.03.

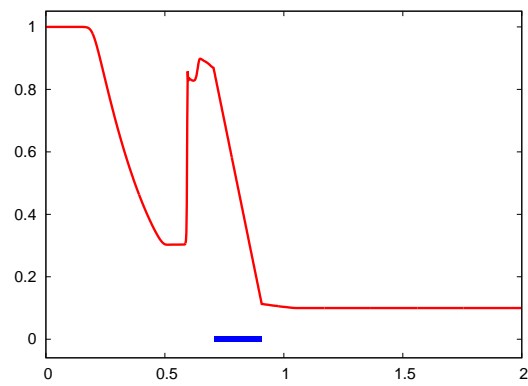
6.1.3. Sod shock coupled with a mass-spring system

To conclude the one-dimensional examples, we consider the mass-spring system interacting with a high pressure gas described in [1] in order to provide validation for our approach against an analytic solution. The domain is of length 20, and a spring is fixed to the right side of the domain which has a rest length of 1, a stiffness of 10^7 , no damping and a mass of 3. The fluid is given by

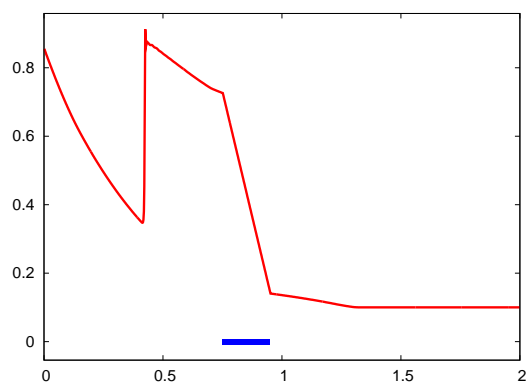
$$(\rho, p, \vec{u}) = (4, 10^6, 0)$$



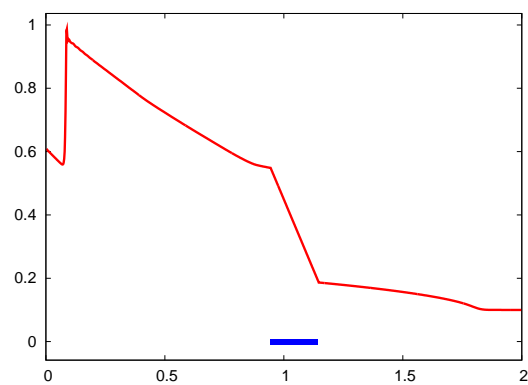
(a) $t = 0$



(b) $t = .25$

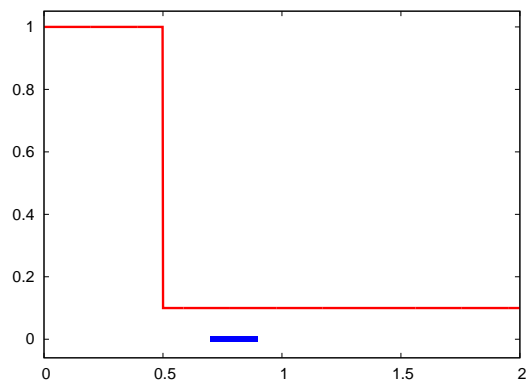


(c) $t = .5$

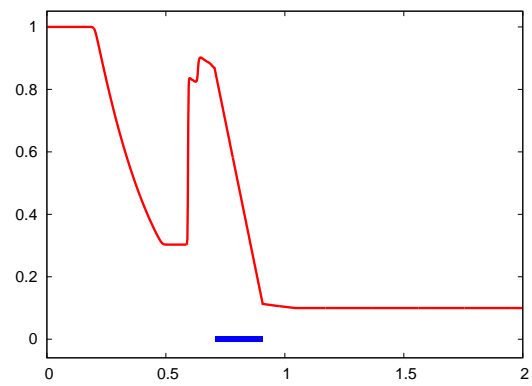


(d) $t = 1$

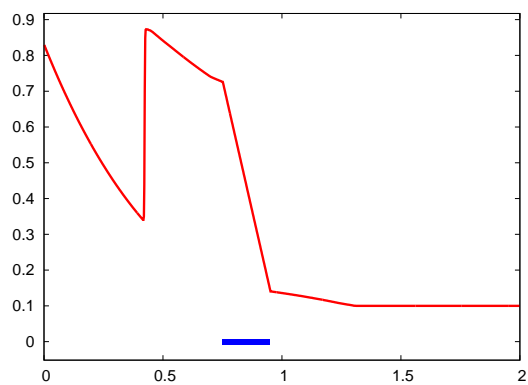
Figure 4: Semi-implicit simulation of a Sod shock hitting a rigid body of mass 1. Pressure profile of the fluid is shown at various times through the simulation. The 1-D rigid body is drawn as a blue line segment at the bottom of the plot, with pressure inside the solid shown as a linear pressure profile. The simulation was done on a grid of resolution 1601.



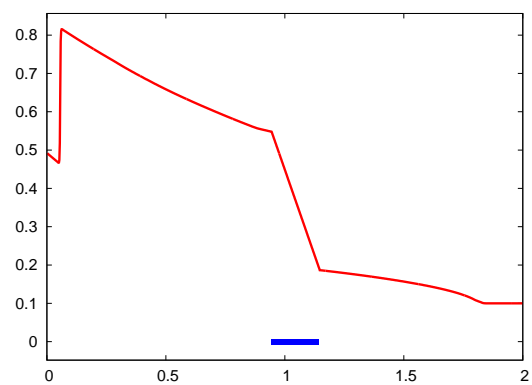
(a) $t = 0$



(b) $t = .25$



(c) $t = .5$



(d) $t = 1$

Figure 5: Explicit simulation of a Sod shock hitting a rigid body of mass 1. Pressure profile of the fluid is shown at various times through the simulation. The 1-D rigid body is drawn as a blue line segment at the bottom of the plot, with pressure inside the solid shown as a linear pressure profile. The simulation was done on a grid of resolution 1601.

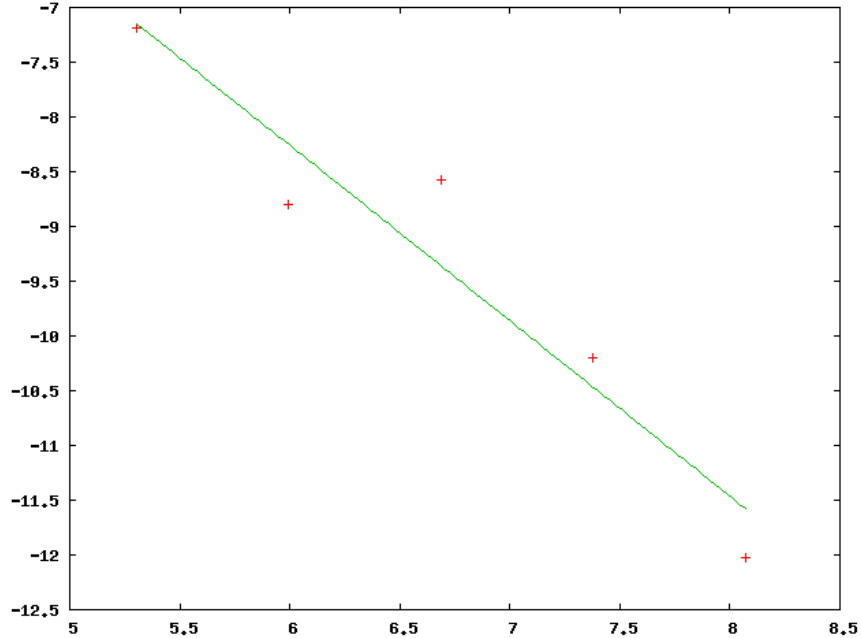


Figure 6: Position error of the center of mass of a rigid body hit by a Sod shock, as compared to a high-resolution simulation, at time .9s. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is 1.6.

An outflow boundary condition is used for the left side of the domain. The spring starts at rest length and is compressed by the gas. Figure 14 shows snapshots of the pressure profile at various times through the simulation. The position of the moving end of the spring as a function of time is shown in Figure 15(a), and a convergence analysis in Figure 15(b). The error in the position of the free end of the spring is computed at time .008, and is compared against the analytic solution provided in [1]. The convergence order of the error is estimated as 1.16.

6.2. Two-dimensional validation

In this section we validate our method for the multidimensional case, and briefly describe a symmetric positive-definite reformulation of the Equation (19). We consider interactions with both rigid and deformable solids. A second order ENO scheme was used along with an advection-based CFL number of .6.

6.2.1. Rigid Cylinder lift-off

This example, which is suggested by [5, 10, 1], examines the interaction of a Mach 3 shock with a rigid cylinder initially at rest on the floor of a rectangular channel. The cylinder is lifted by the shock, due to an asymmetric reflection of the incident wave. The test domain is $1 \times .2$, with the initial shock front positioned at .08 from the left boundary and the remaining domain is filled with the gas at pressure 1 and density 1.4. The top and bottom of the domains are rigid walls, the left boundary is fixed to be the post shock state and an outflow boundary condition is used for the right boundary. The cylinder has a density of 10.77, a radius of .05 and is initially located at (.15, .05). Figure 16 shows the snapshot of the simulation for a selection of times. Our results compare favorably to those shown in [1], and converges at a rate of .93.

6.2.2. Deforming cylinder lift-off

This example is similar to the one described above (in Section 6.2.1), except that the rigid cylinder is replaced by a deformable mass-spring system with 222 triangles, and edge- and altitude-springs with a stiffness of .3. The density of the sphere is 10.77, has a radius of .05 and the center of mass is initially

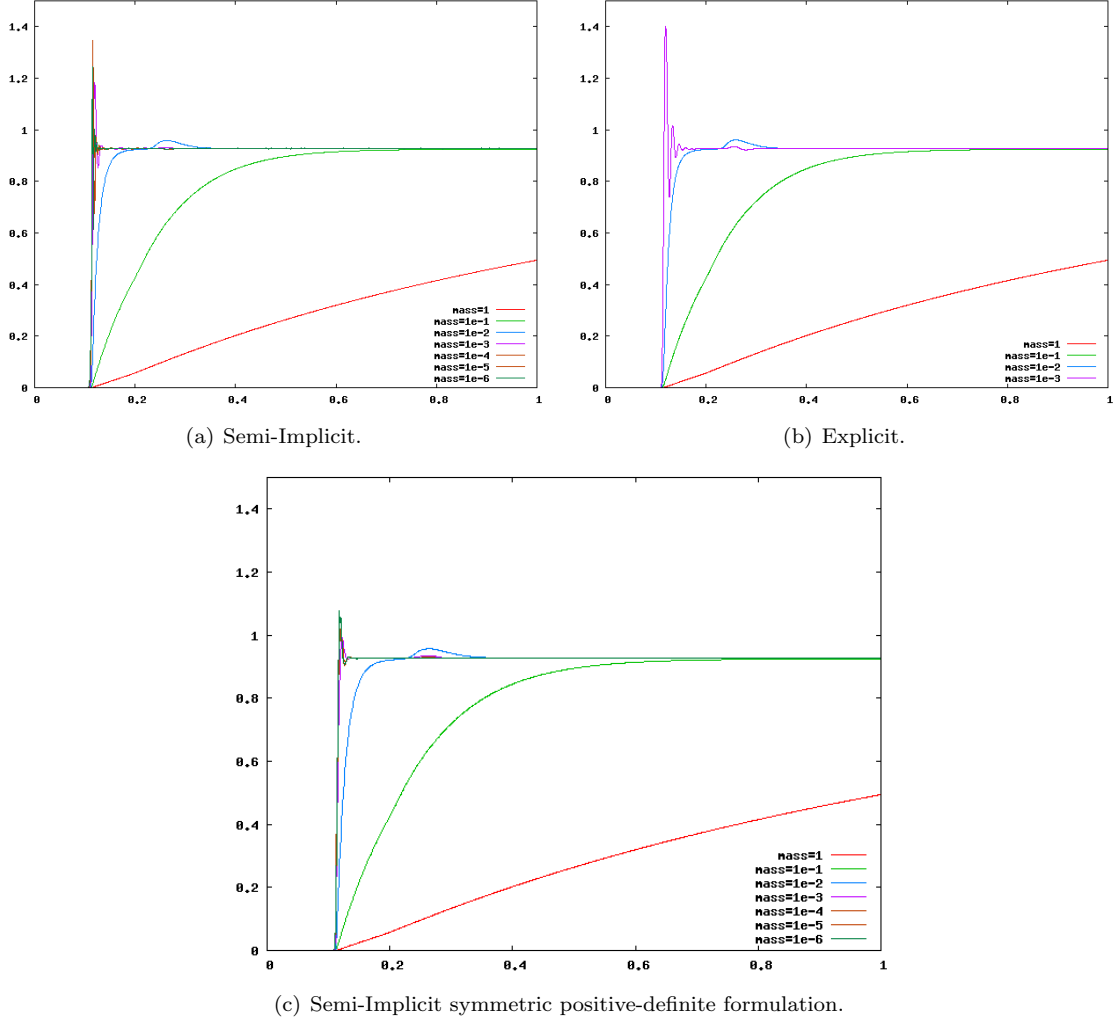
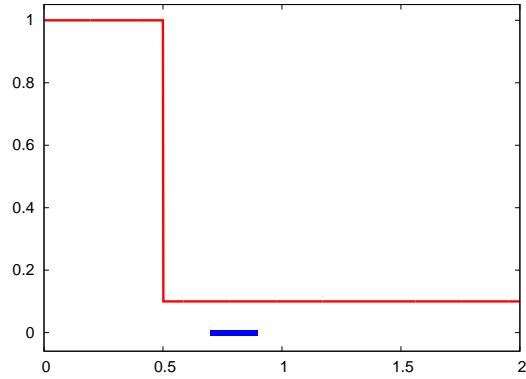
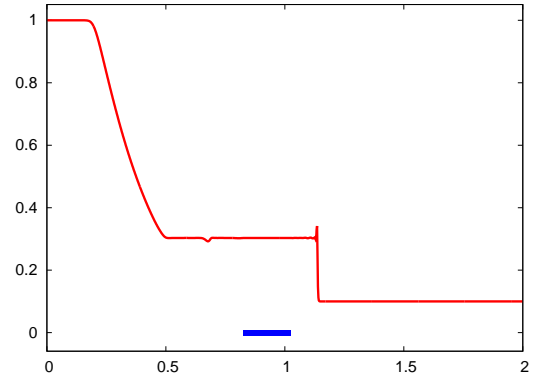


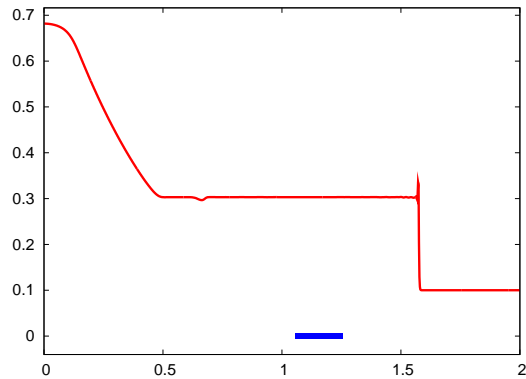
Figure 7: Velocity of a 1-D rigid body hit by a Sod shock, as a function of time. Simulations were done on a grid of resolution 1601. All simulations were run with a CFL number of .6, where the explicit simulation CFL is based on $|u| \pm c$ and the semi-implicit simulation was run with the CFL condition specified in Equation (22). The explicit simulations grow increasingly unstable as mass tends to zero, giving unusable results when mass reaches .0001 (these results are shown in Figure 9), and crashes for lighter masses. As mass tends to zero, the momentum absorbed by the solid tends to zero and the shock passes through the solid relatively unperturbed, and so the flat line to which solid velocities appear to converge is in fact the post-shock velocity of the fluid.



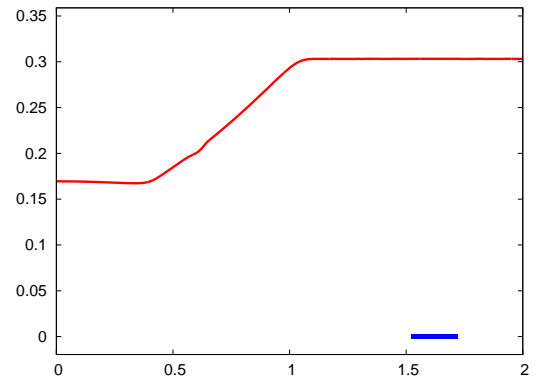
(a) $t = 0$



(b) $t = .25$

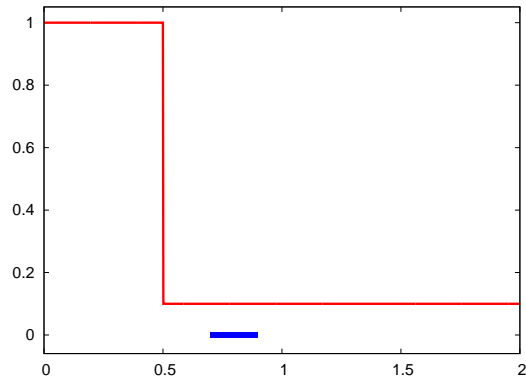


(c) $t = .5$

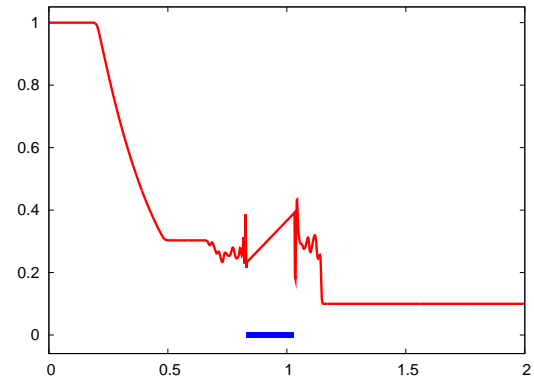


(d) $t = 1$

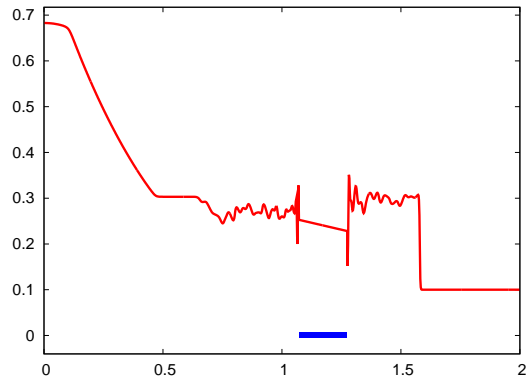
Figure 8: Semi-implicit simulation of a Sod shock hitting a light solid of mass .0001. Pressure profile of the fluid is shown at various times through the simulation. The 1-D rigid body is drawn as a blue line segment at the bottom of the plot. The simulation was done on a grid of resolution 1601. For this light mass, the post-shock state remains practically undisturbed as very little momentum transfers to the solid.



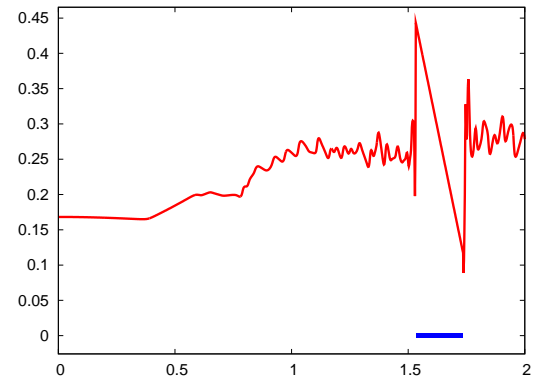
(a) $t = 0$



(b) $t = .25$



(c) $t = .5$



(d) $t = 1$

Figure 9: Explicit simulation of a Sod shock hitting a light solid of mass .0001. Pressure profile of the fluid is shown at various times through the simulation. The 1-D rigid body is drawn as a blue line segment at the bottom of the plot. The simulation was done on a grid of resolution 1601. The CFL number for this simulation is .6, and we use the standard compressible flow CFL, based on $|u| \pm c$. Despite satisfying a reasonable CFL time step restriction, a fully explicit simulation generates unstable results, and even goes unstable and crashes for masses lighter than .0001.

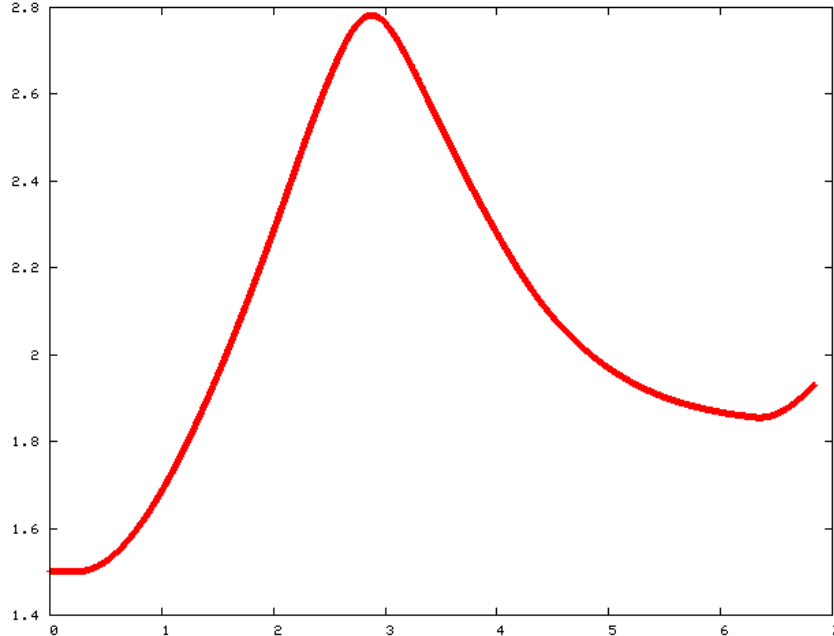


Figure 10: The position of the piston (Section 6.1.2) is plotted as a function of time.

located at $(.15, .05)$. Figure 17 shows snapshots of the simulation for a selection of times. As the shock front passes through the deforming body, it dissipates, scatters and is partially absorbed by the body. The example converges at a rate of .99.

6.2.3. Heavy deforming cylinder lift-off

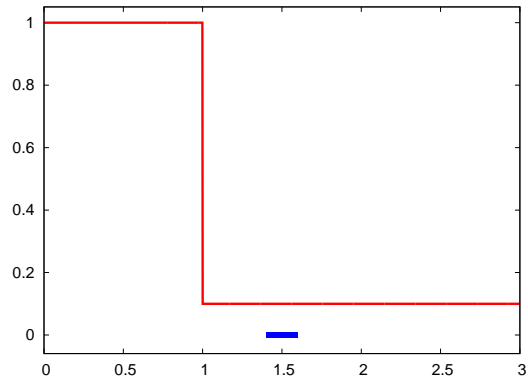
We next consider a heavy deforming cylinder, in the same setup as described in Section 6.2.1 and Section 6.2.2 above. In this case, the cylinder matches the cylinder from Section 6.2.2, except the density is set to 100. As the body absorbs the shock wave, it compresses and delays the shock. Some of the shock is reflected, but most of the shock passes through the cylinder. Figure 18 shows snapshots of the simulation at a selection of times. The example converges at a rate of 1.01.

6.2.4. Shock traveling down a deformable tube

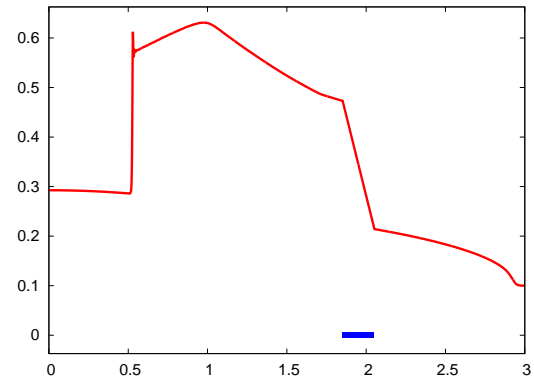
This example is similar to the inflatable bladder examples suggested in [1] and [7] in which a shock wave travels through a deformable tube causing large deformation of the walls. Our results are shown in Figure 19. We also do a convergence analysis of our method in Figure 20. The error in the position of a particle on the deformable tube is computed at time $.00049s$ (which is the approximate time of maximum deformation of that particle in the highest resolution simulation) from the highest resolution grid simulated, which is 800×600 grid cells. The convergence order of the error is estimated as 1.18.

6.2.5. Symmetric positive-definite reformulation

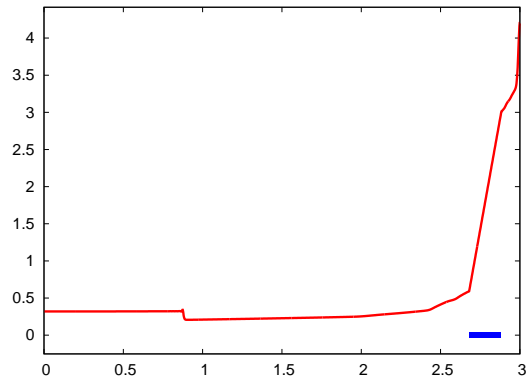
Our numerical method is symmetric, but not positive-definite. Recent developments in [16] discuss a modification of the implicit coupling methodology for incompressible flow by separating out the coupling forces as implicit variables λ (similar to immersed boundary methods), decomposing the symmetric damping force into $D = C^T C$ and solving for $\hat{V}_s = C V_s^{n+1}$. The symmetric positive-definite system they obtain can be modified for compressible flow in a manner similar to Section 4.1 to obtain



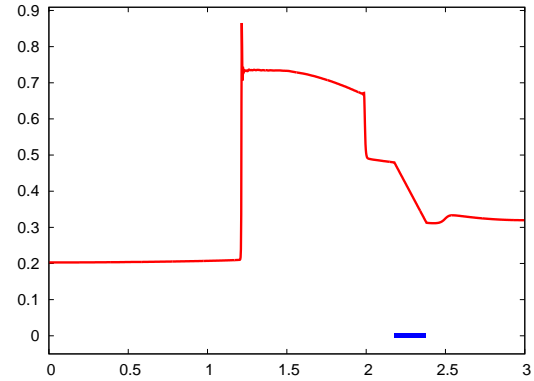
(a) $t = 0$



(b) $t = 1.5$

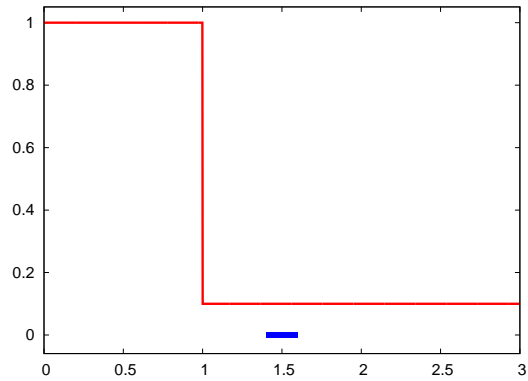


(c) $t = 2.87$

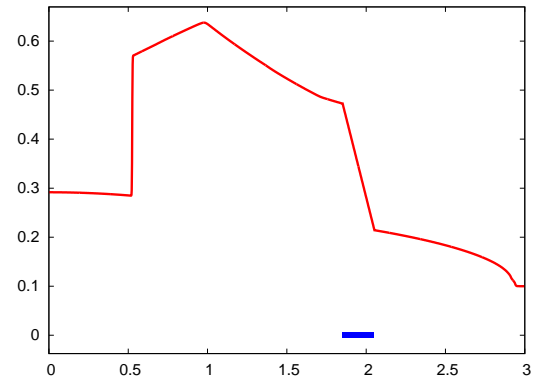


(d) $t = 4$

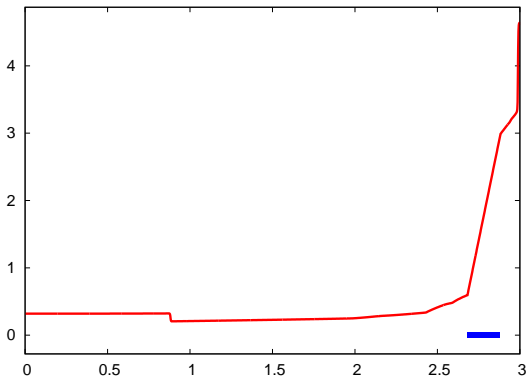
Figure 11: Semi-implicit simulation of a piston hit by a Sod shock, with closed-wall boundary conditions on both sides. Pressure profile of the fluid is shown at various times through the semi-implicit simulation. The 1-D rigid body is drawn as a blue line segment at the bottom of the plot, with pressure inside the solid shown as a linear pressure profile. The simulation was done on a grid of resolution 1601. The shock on the left pushes the rigid body and compresses the fluid on the right into a small high pressure pocket against the wall, which in turn pushes the rigid body back to the left.



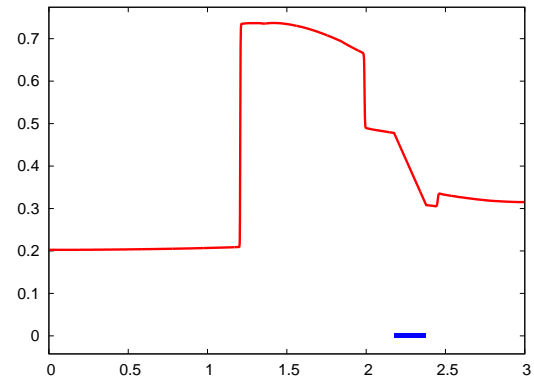
(a) $t = 0$



(b) $t = 1.5$



(c) $t = 2.87$



(d) $t = 4$

Figure 12: Explicit simulation of a piston hit by a Sod shock, with closed-wall boundary conditions on both sides. Pressure profile of the fluid is shown at various times through the explicit simulation. The 1-D rigid body is drawn as a blue line segment at the bottom of the plot, with pressure inside the solid shown as a linear pressure profile. The simulation was done on a grid of resolution 1601. The shock on the left pushes the rigid body and compresses the fluid on the right to a very high pressure against the wall, which in turn pushes the rigid body back to the left.

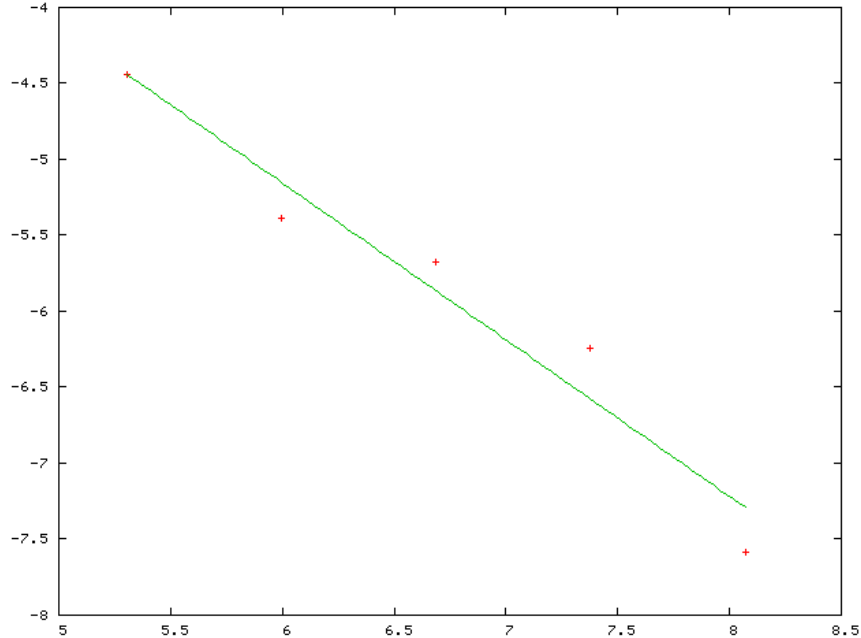


Figure 13: Position error of the center of mass of the piston (Section 6.1.2), as compared to a high-resolution simulation, at time 4s. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is 1.03.

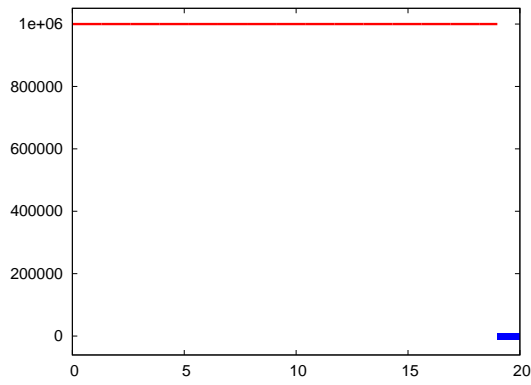
$$\begin{pmatrix} \frac{V}{\Delta t^2 \rho c^2} I + \hat{G}^T \beta^{-1} \hat{G} & -\hat{G}^T \beta^{-1} K^T & 0 \\ -K \beta^{-1} \hat{G} & K(\beta^{-1} + W M^{-1} W^T) K^T & K W M^{-1} C^T \\ 0 & C M^{-1} W^T K^T & I + C M^{-1} C^T \end{pmatrix} \begin{pmatrix} \tilde{p} \\ \lambda \\ \hat{V}_s \end{pmatrix} = \begin{pmatrix} \frac{V}{\Delta t^2 \rho c^2} \tilde{p}^a + \hat{G}^T u^* \\ K W V_s^* - K u^* \\ C V_s^* \end{pmatrix}, \quad (23)$$

where \hat{G} and $-\hat{G}^T$ are the volume weighted gradient and divergence operators respectively, β is the diagonal matrix of fluid dual cell masses, and K^T is the matrix of 1s and 0s mapping λ to the appropriate fluid velocity scalars (see [16] for more details). Note that in order to avoid confusion in notation we renamed a few operators. In particular W and J in [16] correspond to the K and W we use here, respectively. This system is both symmetric and positive-definite. We demonstrate the viability of this modified method in another example, where we've replaced the implicit coupled solve with Equation (23). Our example is similar to the example in Section 6.2.1 except that the sphere is replaced with a diamond whose major axis is of length .1 and minor axis is of length .025. The diamond begins rotated by $\pi/4$, with a center of mass at (.15, .04). Snapshots of the resulting simulation are shown in Figure 21. The convergence analysis for this example is shown in Figure 22 which estimates the convergence order of the error as .84.

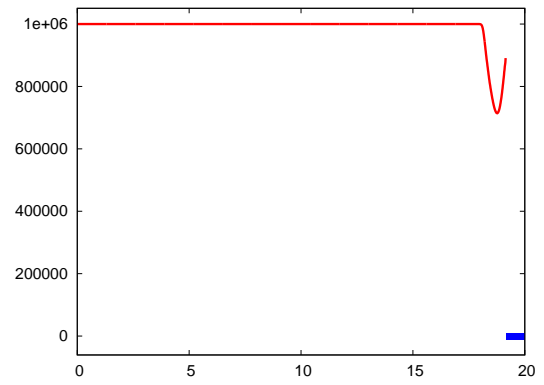
7. Conclusions and future work

We have presented a first order method which implicitly couples compressible flow with solid bodies with arbitrary constitutive models. We show that this method is robust, numerically conservative, and avoids the numerical instabilities which comparable explicit methods suffer from in the presence of high density-to-mass ratios. The same methodology can be applied to reformulate our implicit system into a symmetric positive-definite system.

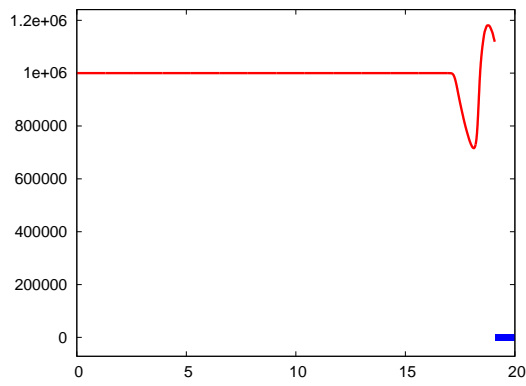
There are several interesting avenues of future work which we wish to explore. Given the promising results which arise from handling fluid-structure interactions implicitly, we believe that an alternative approach would split the fluid flux along Riemann invariants—rather than by pressure—and solve for the Riemann



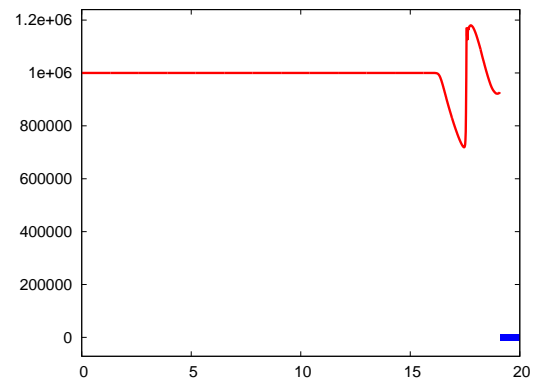
(a) $t = 0$



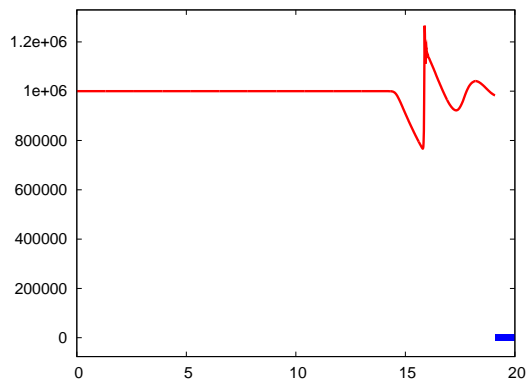
(b) $t = .0015$



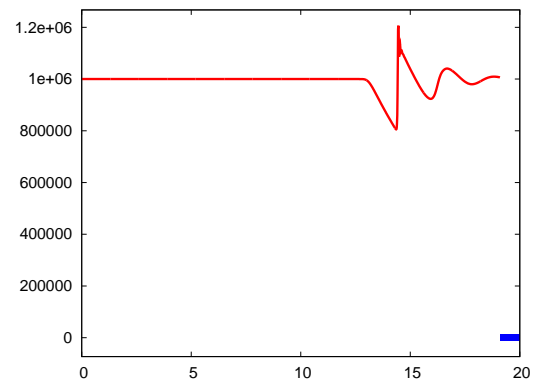
(c) $t = .003$



(d) $t = .0045$



(e) $t = .0075$



(f) $t = .01$

Figure 14: Semi-implicit simulation of a 1-D mass-spring system hit by a Sod shock wave. Pressure profile of the fluid is shown at various times through the semi-implicit simulation. The mass-spring system is drawn as a blue line segment at the bottom of the plot. The simulation was done on a grid of resolution 1601. Note the formation of a spontaneous shock wave.

invariant which interacts with the solid implicitly. Our method also relies on the assumption that the solid has some thickness where ghost cells can be filled, and we believe that the method can be made to work for thin shell structures (such as parachutes). Given the utility of the scheme proposed in [11] in handling fluid-structure interactions, it becomes imperative to address the issues of that original scheme. In particular, the implicit component of the method is overly centrally-differenced, which tends to introduce Gibbs phenomena at shocks. It would be better to add upwind biasing, although it is unclear how to do so.

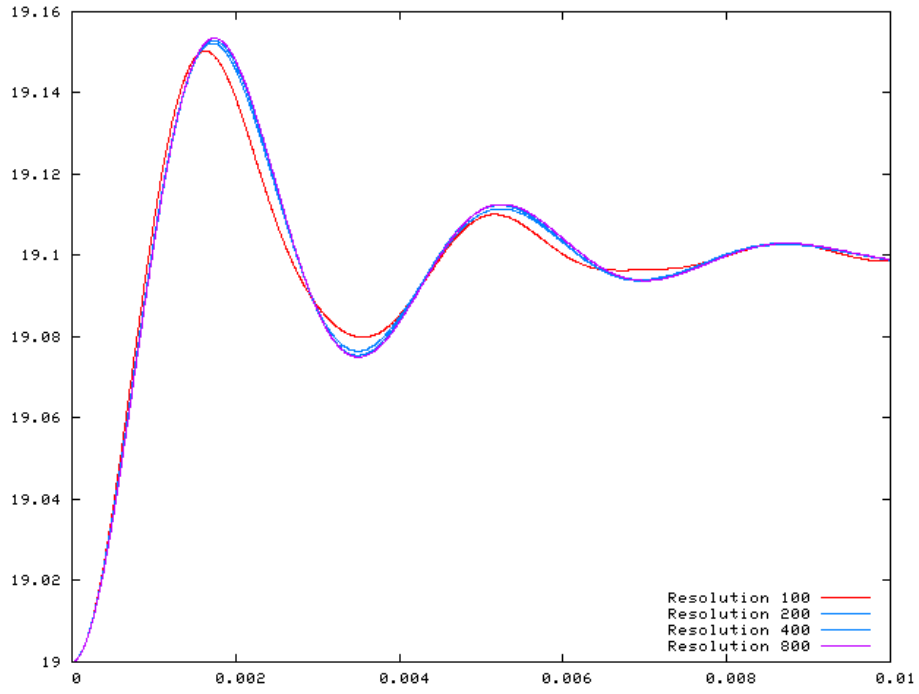
8. Acknowledgements

Research supported in part by a Packard Foundation Fellowship, an Okawa Foundation Research Grant, ONR N00014-06-1-0393, ONR N00014-06-1-0505, ONR N00014-11-1-0027, ONR N00014-05-1-0479 for a computing cluster, NIH U54-GM072970, NSF ACI-0323866, NSF IIS-0326388, and King Abdullah University of Science and Technology (KAUST) 42959. J.G. was supported in part by, and computational resources were provided in part by ONR N00014-06-1-0505 and ONR N00014-09-C-015.

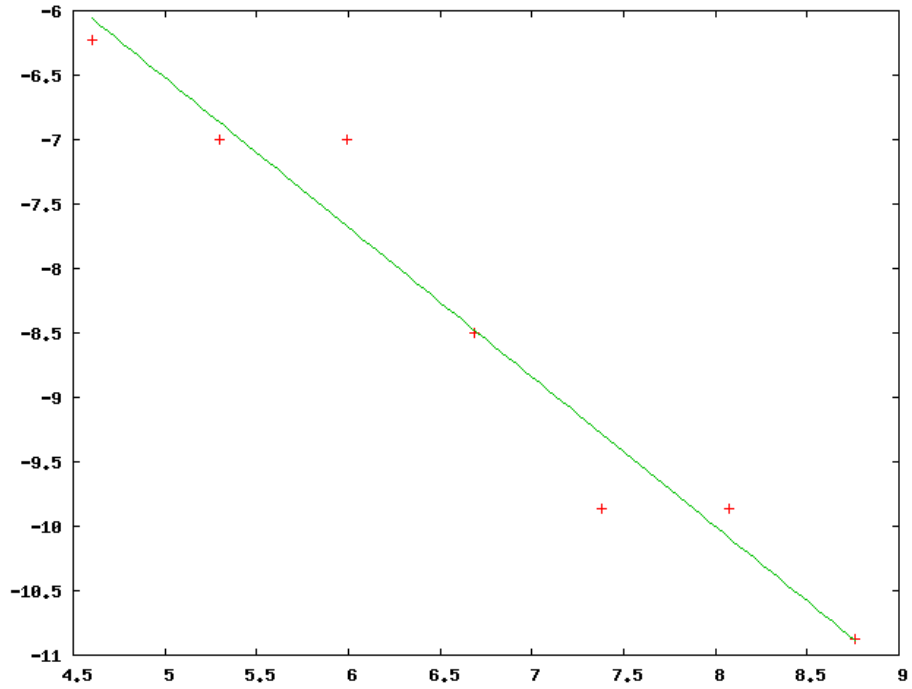
9. Bibliography

- [1] M. Arienti, P. Hung, E. Morano, and J.E. Shepherd. A level set approach to Eulerian–Lagrangian coupling. *J. Comput. Phys.*, 185(1):213–251, 2003.
- [2] D. Benson. A new two-dimensional flux-limited shock viscosity for impact calculations. *Comput. Meth. Appl. Mech. Engng.*, 93(1):39–95, 1991.
- [3] D. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput. Meth. Appl. Mech. Engng.*, 99:235–394, 1992.
- [4] P. Causin, J.-F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comp. Meth. Appl. Mech. Engng.*, 194(42-44), 2005.
- [5] J. Falcovitz, G. Alfandary, and G. Hanoch. A two-dimensional conservation laws scheme for compressible flows with moving boundaries. *J. Comput. Phys.*, 138(1):83–102, 1997.
- [6] C. Farhat, K. G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Comp. Meth. Appl. Mech. Engng.*, 195(17-18):1973 – 2001, 2006.
- [7] R. Fedkiw. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *J. Comput. Phys.*, 175:200–224, 2002.
- [8] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, 152:457–492, 1999.
- [9] R. Fedkiw, X.-D. Liu, and S. Osher. A general technique for eliminating spurious oscillations in conservative schemes for multiphase and multispecies euler equations. *Int. J. Nonlinear Sci. and Numer. Sim.*, 3:99–106, 2002.
- [10] H. Forrer and M. Berger. Flow simulations on Cartesian grids involving complex moving geometries. In *Hyperbolic Problems: Theory, Numerics, Applications; Seventh International Conference in Zürich, February 1998*, page 315. Birkhauser, 1999.
- [11] N. Kwatra, J. Su, J.T. Grétarsson, and R. Fedkiw. A method for avoiding the acoustic time step restriction in compressible flow. *J. Comput. Phys.*, 228(11):4146–4161, 2009.
- [12] T.G. Liu, B.C. Khoo, and W.F. Xie. The modified ghost fluid method as applied to extreme fluid-structure interaction in the presence of cavitation. *Commun. Comput. Phys.*, 1(5):898–919, 2006.

- [13] D.G. Luenberger. The conjugate residual method for constrained minimization problems. *SIAM J. on Numer. Anal.*, pages 390–398, 1970.
- [14] C. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.
- [15] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems part i: Model problem, theory and two-dimensional application. *Comp. Meth. Appl. Mech. Engng.*, 124(1-2):79 – 112, 1995.
- [16] A. Robinson-Mosher, C. Schroeder, and R. Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. *J. Comput. Phys.*, 2010. in review.
- [17] A. Robinson-Mosher, T. Shinar, J.T. Grétarsson, J. Su, and R. Fedkiw. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. on Graphics*, 27(3):46:1–46:9, August 2008.
- [18] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II (two). *J. Comput. Phys.*, 83:32–78, 1989.
- [19] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *J. Sci. Comput.*, 19(1):439–456, 2003.
- [20] K. Taira and T. Colonius. The immersed boundary method: A projection approach. *J. Comput. Phys.*, 225(2):2118–2137, 2007.
- [21] H. Zhang, X. Zhang, S. Ji, Y. Guo, G. Ledezma, N. Elabbasi, and H. deCougny. Recent development of fluid-structure interaction capabilities in the adina system. *Comput. and Struct.*, 81(8-11):1071 – 1085, 2003. K.J Bathe 60th Anniversary Issue.

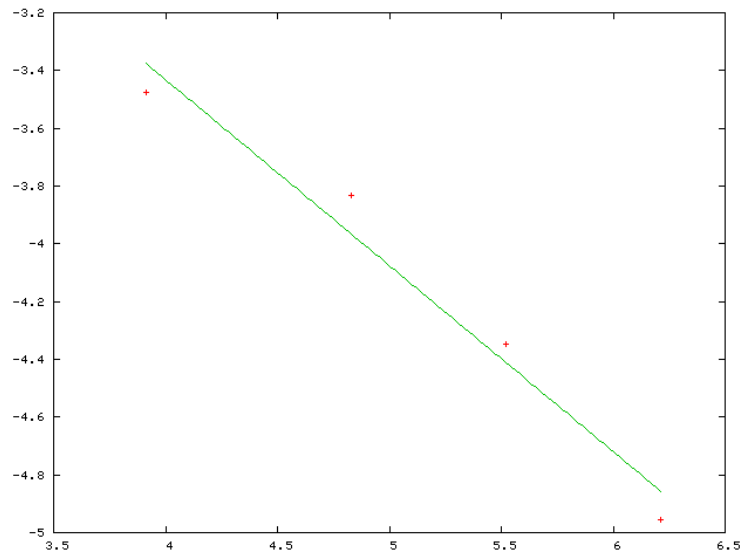
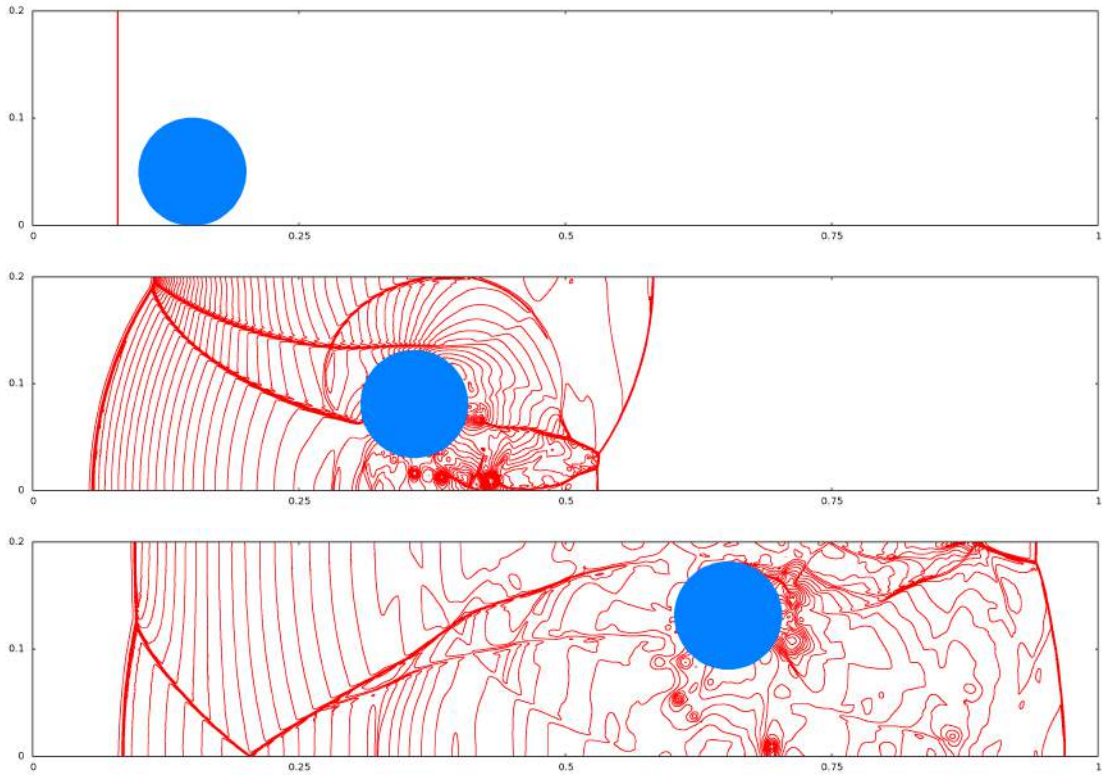


(a) Position of the free end of the spring, as a function of time.



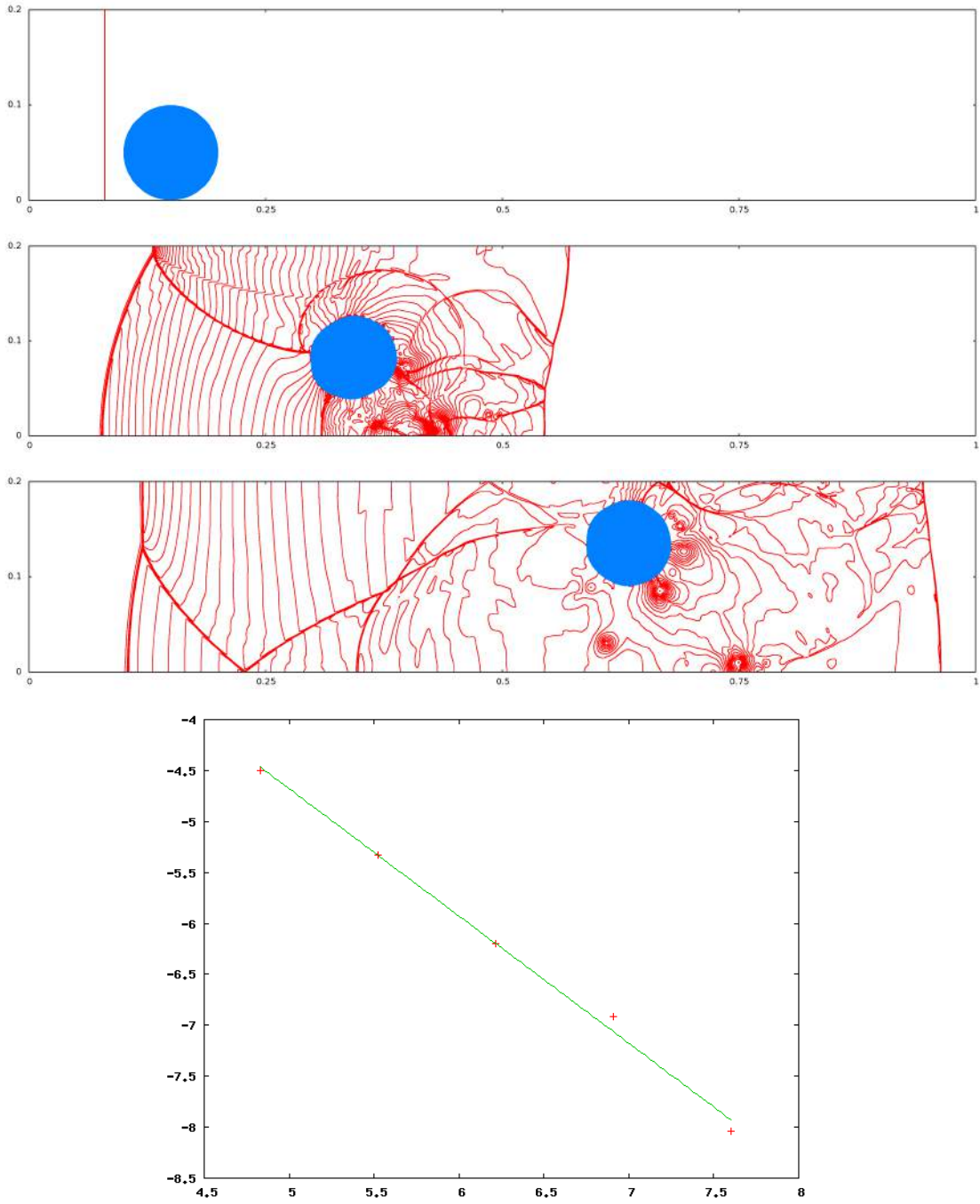
(b) Position error for the left-most side of the mass-spring system, as compared to the analytic solution provided in [1], at time .008s. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is 1.16.

Figure 15: 1-D mass-spring system hit by a Sod shock wave.



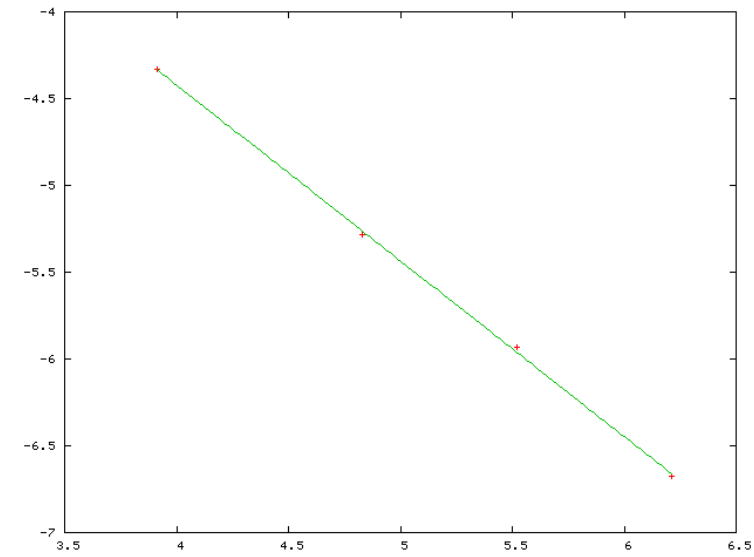
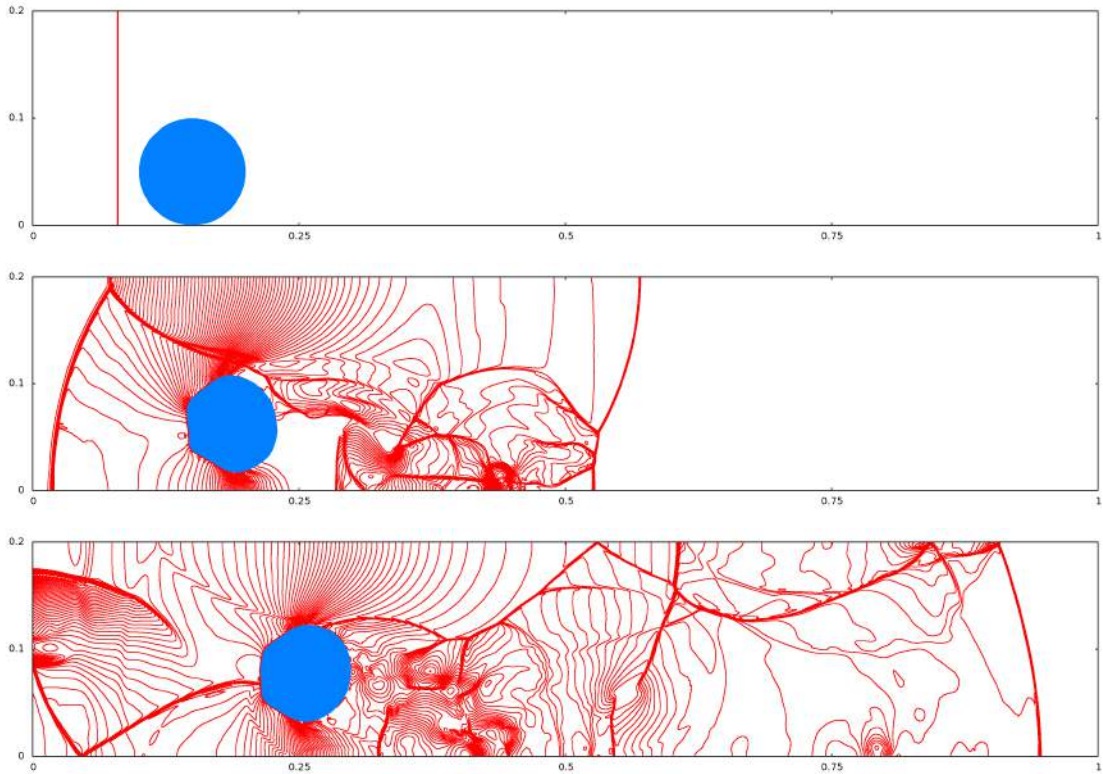
(d) Position error of the center of mass of the cylinder hit by a planar shock, as compared to a high-resolution simulation, at time $t = .15s$, with a convergence of .96.

Figure 16: Pressure contours for semi-implicit simulation of rigid cylinder lift off are shown at $t = 0$, $t = .164$ and $t = .301$. The simulation is run with a CFL number of .6, using the CFL restriction discussed in Equation 22.



(d) Position error of the center of mass of the deformable cylinder hit by a planar shock, as compared to a high-resolution simulation, at time $t = .15s$. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is .99.

Figure 17: Pressure contours for semi-implicit simulation of deformable cylinder lift off are shown at $t = 0$, $t = .164$ and $t = .301$. The simulation is run with a CFL number of .6, using the CFL restriction discussed in Equation 22.



(d) Position error of the center of mass of the heavy deformable cylinder hit by a planar shock, as compared to a high-resolution simulation, at time $t = .15s$. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is 1.01.

Figure 18: Pressure contours for semi-implicit simulation of deformable cylinder lift off are shown at $t = 0$, $t = .164$ and $t = .301$. The simulation is run with a CFL number of .6, using the CFL restriction discussed in Equation 22.

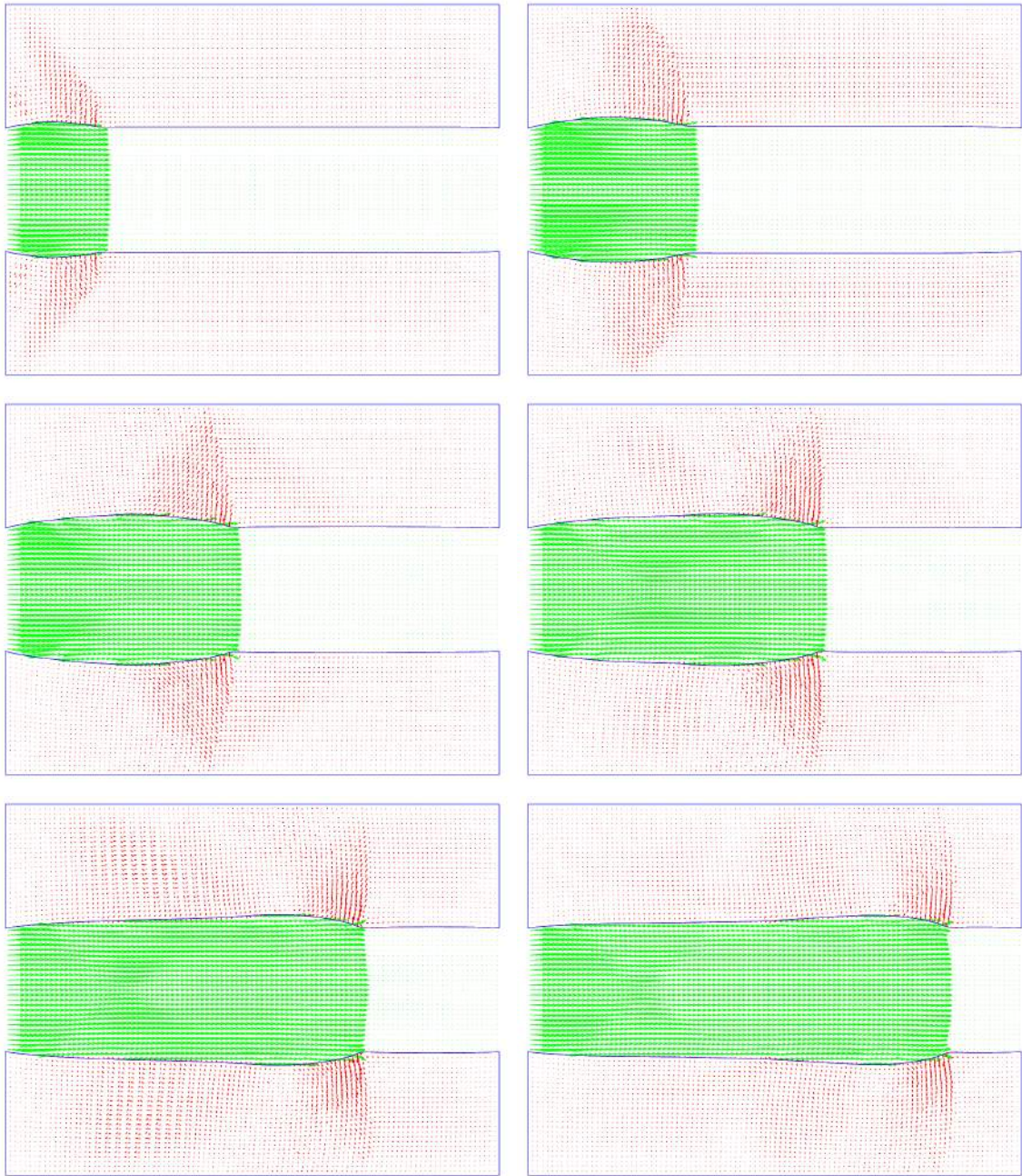


Figure 19: A planar shock travels down a deformable bladder. Shown are the velocity field of the fluid in green and the velocities of the deformable nodes in red at times $t = .0001$, $t = .0002$, $t = .0003$, $t = .0004$, $t = .0005$ and $t = .0006$.

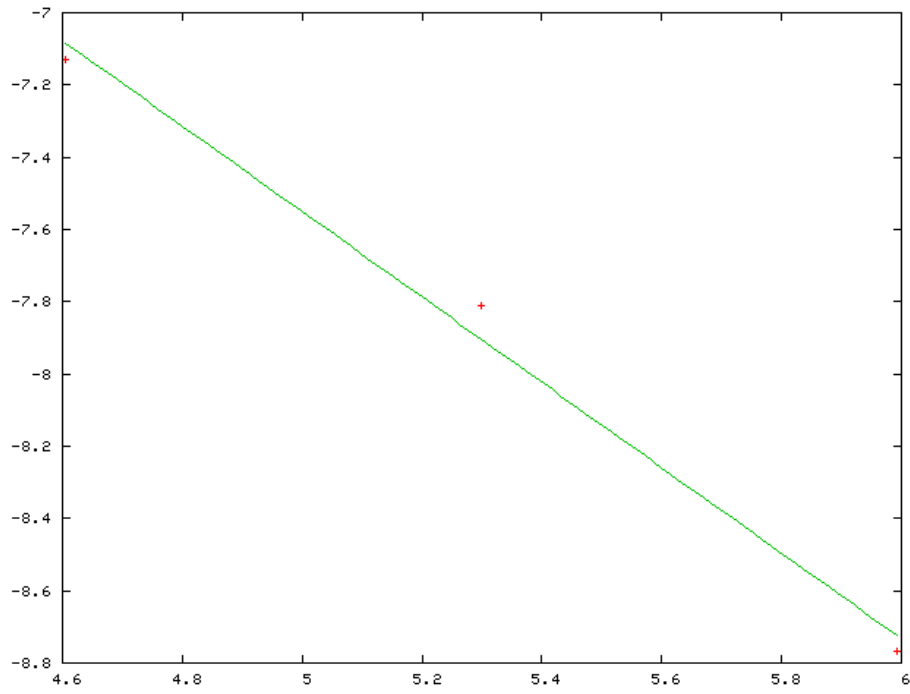


Figure 20: Position error of the position of a particle on the deformable tube hit by a planar shock, as compared to a high-resolution simulation, at time $.00049s$. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is 1.18.

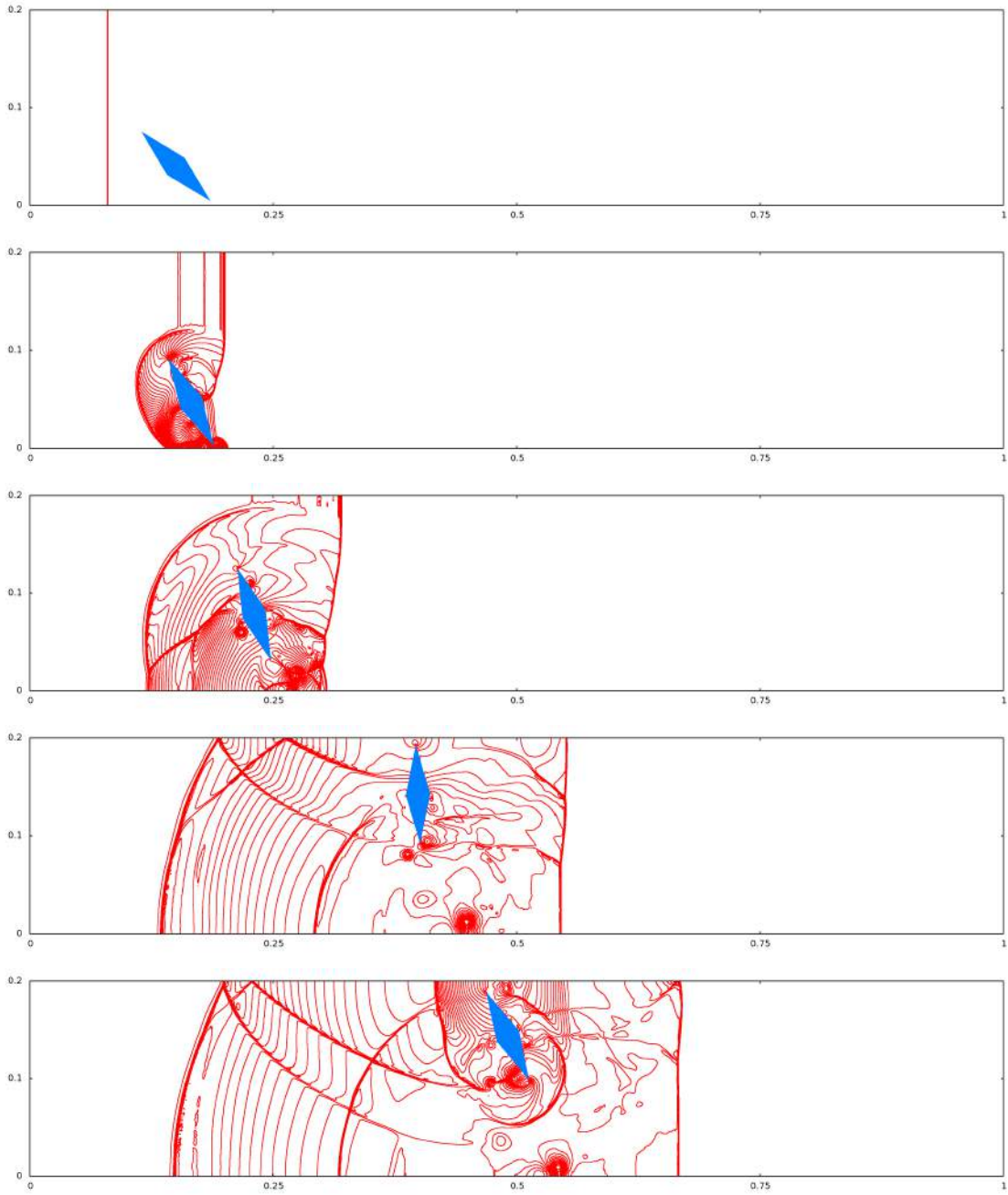


Figure 21: A diamond is hit by a planar shock, and then collides with the top of the channel. Shown are pressure contours at $t = 0$, $t = .04$, $t = .08$, $t = .16$ and $t = .2$.

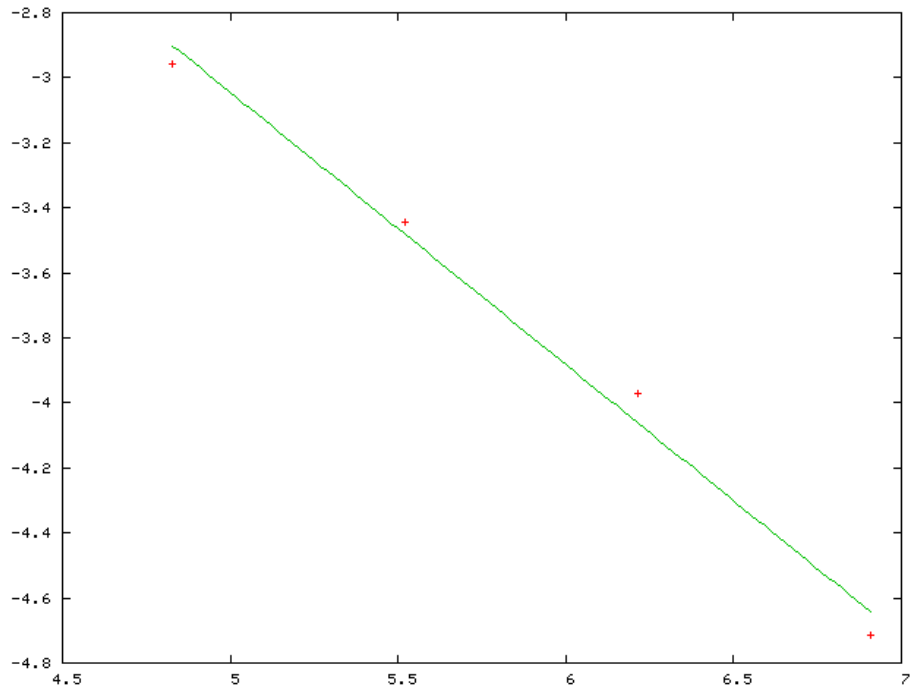


Figure 22: Position error of the center of mass of the diamond hit by a planar shock, as compared to a high-resolution simulation, at time .15s. We plot the log of the relative error, as a function of the log of the resolution of the underlying grid. The convergence rate is .84.