

Parallel Adaptive Cartesian Upwind Methods for Shock-driven Multiphysics Simulation

Ralf Deiterding

Computer Science and Mathematics Division, Oak Ridge National Laboratory
P.O. Box 2008, MS-6367, Oak Ridge, TN 37831, USA
E-mail: deiterdingr@ornl.gov

Abstract: *The multiphysics fluid-structure interaction simulation of shock-loaded thin-walled structures requires the dynamic coupling of a shock-capturing flow solver to a solid mechanics solver for large deformations. By combining a Cartesian embedded boundary approach with dynamic mesh adaptation a generic software framework for such flow solvers has been constructed that allows easy exchange of the specific hydrodynamic finite volume upwind scheme and coupling to various explicit finite element solid dynamics solvers. The paper gives an overview of the computational approach and presents first simulations that couple the software to the general purpose solid dynamics code DYNA3D.*

Key words: *Shock-capturing Cartesian finite volume method, fluid-structure interaction, adaptive mesh refinement, parallelization*

1 Introduction

The construction of efficient and scalable algorithms for simulating fluid-structure interaction (FSI) problems is an area of active research. This is particularly true for shock-driven problems, for which the discretizations both in fluid and solid are usually time-explicit and therefore computationally comparably inexpensive. On the other hand, major geometric complexities, such as large structural deformations, fracture and even fragmentation, might have to be considered. An approach to this problem is to employ an immersed or embedded boundary method in the fluid solver, in which moving solid structures slide through a fixed Eulerian fluid background mesh. In most cases, structured Cartesian schemes are used for embedded boundary techniques, cf. [19].

In here, some of the inner workings of the generic Cartesian fluid solver framework AMROC will be described. AMROC implements the ghost fluid approach [12, 1] and relies on a scalar level set function, storing the distance to the nearest element of the solid's triangulation, to represent the embedded geometry on the fluid grid. To mitigate boundary approximation inaccuracies the fluid mesh in the vicinity of the immersed boundary is refined on the fly. For coupling, a temporal splitting technique, in which solvers exchange data only at the interface between disjoint computational domains after consecutive time steps, is adopted, cf. [10]. Distributed memory parallelization both of the fluid and the solid mechanics solver is fully supported permitting large-scale computations of technical relevance. The solver suite integrating AMROC with several solid mechanics solvers is named *Virtual Test Facility* (VTF) [9] and was released as public domain software in fall 2007. Successful FSI applications of the VTF software include, for instance, blast waves impinging on deforming viscoplastic materials modeled with a volumetric finite element solver [10], detonation waves in combustible gases causing the fracture of piping using a thin-shell finite element method [3], or strong pressure waves in water inducing the rupture of metallic plates [8]. In this paper, we give an overview of the computational methodologies used in AMROC and, for the first time, present fluid-structure interaction simulations in which

the software is used in combination with the serial version of the general purpose explicit solid mechanics solver DYNA3D [15].

2 Governing equations and used upwind discretizations

The simulation of trans- or supersonic wave phenomena in fluids requires the consideration of the compressibility while viscosity can typically be neglected. The basic system of governing equations are the Euler equations:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \quad \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = 0, \quad \partial_t (\rho E) + \nabla \cdot ((\rho E + p) \mathbf{u}) = 0 \quad (1)$$

Herein, ρ is the density, \mathbf{u} the velocity vector and E the specific total energy. In order to close (1), an equation of state $p = p(\rho, e)$ is required for modeling the dependency of the hydrostatic pressure p on density ρ and specific internal energy $e := E - \frac{1}{2} \mathbf{u}^T \mathbf{u}$. For a single polytropic gas, the equation of state (EOS) reads

$$p = (\gamma - 1) \rho e \quad (2)$$

with γ denoting the constant adiabatic exponent. For Eq. (2), the speed of sound in the fluid is $c = (\gamma p / \rho)^{1/2}$. For very high pressures, as they appear for instance in underwater explosions, a simple extension of Eq. (2) to the *stiffened* gas EOS of the form

$$p = (\gamma - 1) \rho e - \gamma p_\infty \quad (3)$$

is sufficient to model pressure waves in liquids with speed of sound $c = (\gamma (p + p_\infty) / \rho)^{1/2}$. In Section 6, the elastic vibration of a steel panel hit by a shock wave in air, which is modeled as a polytropic gas with $\gamma = 1.4$, is considered. In Section 7, the viscoplastic deformation of a copper plate impacted by a strong pressure wave in water, modeled with EOS (3) using $\gamma = 7.415$, $p_\infty = 296.2$ MPa, is simulated.

To solve Eqs. (1) and (2) numerically in Section 6, we apply a time-explicit shock-capturing finite volume scheme based on Roe's approximate Riemann solver (cf. [22]) specially hybridized with the Harten-Lax-vanLeer (HLL) scheme to ensure strict positivity preservation (see [5] for details). Second-order accuracy in smooth solution regions is achieved with the MUSCL-Hancock variable extrapolation technique. In Section 7, Eqs. (1) and (3) are approximated with a positivity-preserving Harten-Lax-vanLeer scheme with restored contact discontinuity (HLLC) [23] incorporated into the truly multi-dimensional second-order accurate Wave Propagation method of LeVeque [16]. Mathematical details about this particular scheme are given in [8].

3 Embedding of moving boundaries into Cartesian schemes

Geometrically complex moving boundaries are incorporated into the finite volume methods by using some of the cells as ghost cells for enforcing immersed moving wall boundary conditions, cf. [1, 13]. The boundary geometry is mapped onto the Cartesian mesh by employing a scalar level set function ϕ that stores the distance to the boundary surface and allows the efficient evaluation of the boundary outer normal in every mesh point as $\mathbf{n} = -\nabla \phi / |\nabla \phi|$. Since we employ only thin-shell elements in this paper, that have a mesh only within the element midplane and implicitly assumed constant thickness h , we use *unsigned* distance as level set information. Note that for structures discretized with volumetric elements typically a *signed* distance would be employed, cf. [10]. A fluid cell is then treated as an exterior ghost cell if its midpoint satisfies $\phi < h/2$. This condition also considers the mandatory thickening of the surface mesh by the element thickness h . The contour line $\phi = h/2$ effectively represents the embedded boundary for the fluid solver (depicted as dotted line around shell elements in Fig. 1). The hydrodynamic load on each thin

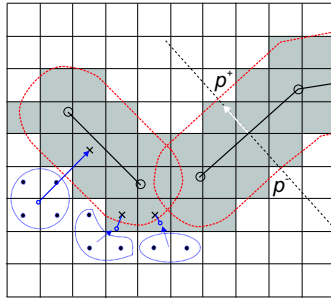


Figure 1: Ghost cells (shaded gray) around shell elements (dark segments) and construction of mirrored values.

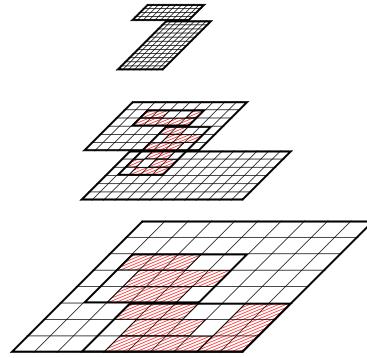


Figure 2: SAMR hierarchy.

element is then evaluated as the difference between the approximated pressure values at $\phi = h/2$ in the positive and negative direction of each element's normal, i.e. $p^F := p^+ - p^-$.

For the governing equations (1), the boundary condition at a rigid wall moving with velocity \mathbf{v} is $\mathbf{u} \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n}$. Enforcing the latter with ghost cells, in which the discrete values are located at the cell centers, requires the mirroring of the values ρ , \mathbf{u} , p across the embedded boundary. The normal velocity in the ghost cells is set to $(2\mathbf{v} \cdot \mathbf{n} - \mathbf{u} \cdot \mathbf{n})\mathbf{n}$, while the mirrored tangential velocity remains unmodified. Mirrored values are constructed by calculating spatially interpolated values in the point $\tilde{\mathbf{x}} = \mathbf{x} + 2\phi\mathbf{n}$ from neighboring interior cells. We employ a dimension-wise linear interpolation for this operation, but it has to be emphasized that directly near the boundary the number of interpolants needs to be decreased to ensure the monotonicity of the numerical solution [7]. This property is essential in simulating hyperbolic problems with discontinuities. Figure 1 also highlights the necessary reduction of the interpolation stencil for some exemplary cases. The interpolation locations are indicated by the origins of the arrows normal to the contour line that defines the embedded boundary.

Crucial for the performance of the overall method is the fast evaluation of the distance information, which is computationally equivalent to determining for every fluid cell the closest solid mesh element. For this purpose, we employ a specially developed algorithm based on characteristic reconstruction and scan conversion [18] that computes the accurate distance information only in a small band around the embedded structure.

4 Parallel block-structured adaptive mesh refinement (SAMR)

As it is characteristic for immersed Cartesian techniques, the boundary treatment described in the previous section results in some geometric approximation inaccuracies. We mitigate this problem by refining the embedded boundary dynamically during the computation, in most cases up to the highest available resolution. A refinement criterion based on $\phi \equiv 0$ has been implemented for this purpose.

In order to dynamically adapt the local resolution we employ the block-structured adaptive mesh refinement (SAMR) method after Berger and Colella [2] which is tailored especially for hyperbolic conservation laws on logically rectangular grids. Cells being flagged by various refinement indicators, e.g., scaled gradients and/or heuristic error estimation by Richardson extrapolation, are clustered with a special algorithm into non-overlapping rectangular grids. The regularity of the grid data enables high computational performance on vector and super-scalar processors and cache optimizations. Values of cells covered by finer subgrids are overwritten by averaged fine grid values subsequently. Refinement grids are derived recursively from coarser ones and a hierarchy of successively embedded levels is thereby constructed, see Fig. 2. Note also that the levels are integrated recursively using hierarchical time step refinement, which also contributes further to the efficiency of the approach. Spatial and temporal mesh widths on level

```

advance_level( l )
  repeat r_l times
    if time to regrid
      regrid( l )
    level_set_generation(  $\phi^l, \mathcal{I}$  )
    update_fluid_level(  $\mathbf{Q}^l, \phi^l, \mathbf{v}|_{\mathcal{I}}, \Delta t_l$  )
    if level l+1 exists
      advance_level(l+1)
      Correct  $\mathbf{Q}^l(t + \Delta t_l)$  with  $\mathbf{Q}^{l+1}(t + \Delta t_l)$ 
    if l = l_c
      send_interface_data(  $p(t + \Delta t_l)|_{\mathcal{I}}$  )
      receive_interface_data(  $\mathcal{I}, \mathbf{v}|_{\mathcal{I}}$  )
    t := t +  $\Delta t_l$ 
  return

```

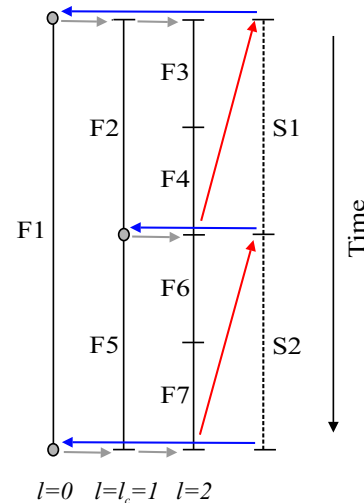


Figure 3: Left: SAMR algorithm for fluid-structure coupling. Right: data exchange between `advance_level()` and an explicit solid solver throughout one level 0 time step.

l are r_l -times finer than on level $l - 1$, i.e. $\Delta t_l := \Delta t_{l-1}/r_l$ and $\Delta x_{k,l} := \Delta x_{k,l-1}/r_l$ with $r_l \geq 2$ for $l > 0$ and with $r_0 = 1$, and a time-explicit finite volume scheme will (in principle) remain stable on all levels of the hierarchy.

Parallelization of the SAMR method is relatively straightforward as already in the serial algorithm subgrids are computationally decoupled by utilizing layers of halo cells. The halos on level l are set either to implement physical boundary conditions, for $l > 0$ by time-space interpolation from data on level $l - 1$, or by copying the data value from an overlying subgrid on l (synchronization). In our SAMR solver framework AMROC we follow a rigorous domain decomposition approach and partition the SAMR hierarchy from the root level on. A careful analysis of the SAMR algorithm uncovers that the only parallel operations under this paradigm are halo cell synchronization, redistribution of the data hierarchy and the application of flux correction terms along internal refinement boundaries that impose the sum of abutting fine cell numerical fluxes on coarse grid cells, cf. [6]. In AMROC, partitions with similar workload are found at runtime as the hierarchy evolves by a hierarchical partitioning algorithm based on a generalization of Hilbert's space-filling curve [20]. The space-filling curve defines an ordered sequence on the cells of the root level that can easily be split in load-balanced portions. As such curves are constructed recursively, they are locality-preserving and therefore avoid an excessive data redistribution overhead. Further on, the surface area is small, which reduces synchronization costs. Recent AMROC benchmark (not shown here) exhibit good scalability for typical SAMR fluid-only cases on several thousand processors.

5 Multiphysics fluid-structure coupling with SAMR

In the VTF, we apply a loosely coupled, partitioned approach and use separated solvers to simulate the fluid and solid sub-problems. Fluid-structure interaction is assumed to take place only at the evolving interface between fluid and solid and is implemented numerically by exchanging boundary data after consecutive time steps. The adaptive fluid solver with embedded boundary capability receives the velocities and the discrete geometry of the solid surface, while only the hydrostatic pressure is communicated back to the solid solver as a force acting on the solid's exterior [17]. As the inviscid Euler equations can not impose any shear on the solid structure, cf. [12], the fluid pressure is sufficient to prescribe the entire stress tensor on the solid boundary. An efficient parallel communication library has also been implemented to support the boundary data exchange between multiple dedicated fluid and processes and the single DYNA3D process, cf. [10].

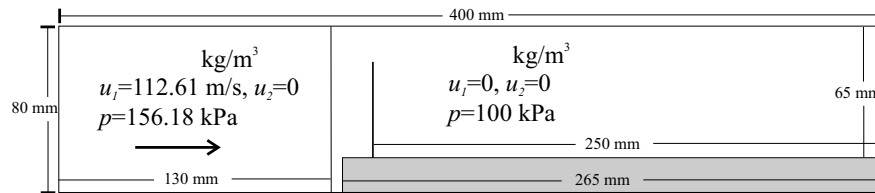


Figure 4: Geometry setup and fluid initial conditions for the deflecting panel case.

While the implementation of a loosely coupled FSI method is straightforward with conventional solvers with consecutive time update, the utilization of the recursive SAMR method is non-apparent. In the VTF, we treat the fluid-solid interface \mathcal{I} as a discontinuity that is a-priori refined at least up to a coupling level l_c . The resolution at level l_c has to be sufficiently fine to ensure an accurate wave transmission between fluid and structure, but might not necessarily be the highest level of refinement. We formulate the corresponding extension of the recursive SAMR algorithm in the routine `advance_level()` outlined in pseudo-code on the left side of Fig. 3. The algorithm calls the routine `level_set_generation()` to evaluate the signed distance ϕ for the actual level l based on the currently available interface \mathcal{I} . Together with the recent solid velocity on the interface $\mathbf{v}|_{\mathcal{I}}$, the discrete vector of state in the fluid \mathbf{Q} is updated for the entire level. The method then proceeds recursively to higher levels and utilizes the (more accurate) data from the next higher level to correct cells overlaid by refinement. If level l is the coupling level l_c , we use the updated fluid data to evaluate the pressure on the nodes of \mathcal{I} , which is sent to the solid and to receive updated mesh positions and nodal velocities. The recursive order of the SAMR algorithm automatically ensures that updated interface mesh information is available at later time steps on coarser levels and to adjust the grids on level l_c .

The data exchange between the solid and `advance_level()`, is visualized in the right graphic of Fig. 3 for an exemplary SAMR hierarchy with two additional levels with $r_{1,2} = 2$. Figure 3 pictures the recursion in the SAMR method by numbering the fluid update steps (F) according to the order determined by `advance_level()`. The order of the solid update steps (S) on the other hand is strictly linear. The red arrows correspond to the sending of the interface pressures $p|_{\mathcal{I}}$ from fluid to solid at the end of each time step on level l_c . The blue arrows visualize the sending of the interface mesh \mathcal{I} and its nodal velocities $\mathbf{v}|_{\mathcal{I}}$ after each solid update. The modification of refinement meshes is indicated in Fig. 3 by the gray arrows; the initiating base level that remains fixed throughout the regridding operation is indicated by the gray circles.

6 Simulation of shock-induced elastic panel deflection

As first test for the coupling between AMROC and DYNA3D, we simulate the quasi two-dimensional verification configuration of a thin-walled steel panel impacted by a planar shock wave in air ($\gamma = 1.4$) proposed by Giordano et al. [14]. The panel has the thickness $h = 1$ mm and extends 50 mm from a mounting with forward-facing step geometry into which it is firmly clamped. Figure 4 depicts the computational domain and initial conditions. Inflow boundary conditions are applied on the left side, rigid wall boundary conditions anywhere else.

We assume that the fluid domain and the panel extend 5 mm in the x_3 -direction. The panel is modeled in DYNA3D simply with ten four-node shell elements. The material is assumed to be linearly elastic with density $\rho_s = 7600$ kg/m³, elasticity modulus $E = 220$ GPa and Poisson ratio $\nu = 0.33$. The panel is embedded into a three-dimensional fluid base mesh of $320 \times 64 \times 2$ cells that allows up to two additional levels of dynamic isotropic refinement (based on ϕ and scaled gradients of ρ and p) with refinement factors $r_{1,2} = 2$. Beside $u_3 = 0$, all fluid initial conditions are shown in Fig. 4. Calculating 19,864 coupled time steps at $l_c = 2$ to $t_e = 5.0$ ms required ~ 450 h CPU (~ 28.2 h wall time) on sixteen 3.4 GHz processors connected with a GB-Ethernet network. To accommodate for mandatorily smaller time steps in the solid, four

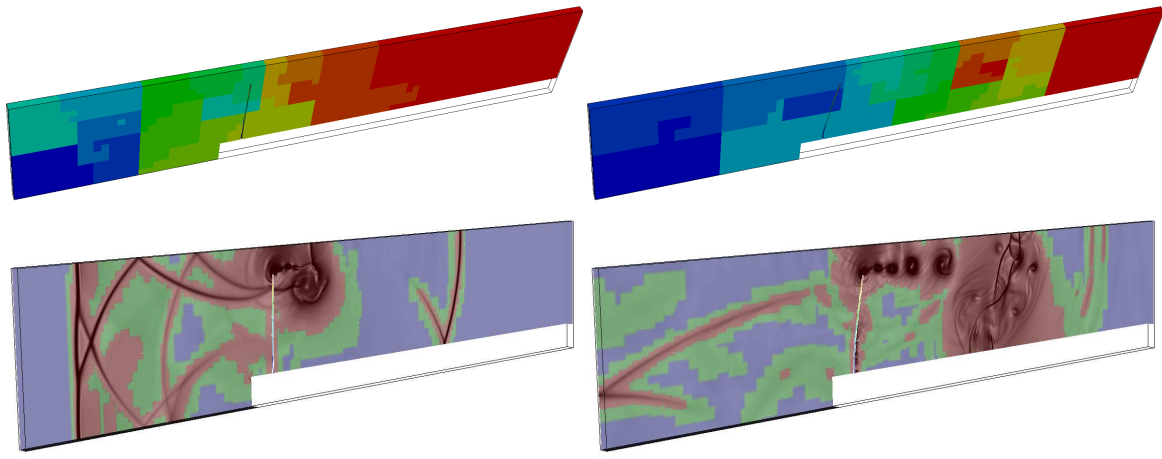


Figure 5: Quasi two-dimensional computation of a thin-shell panel hit by a shock wave at $t \approx 0.43$ ms (left) and $t \approx 1.17$ ms (right) after impact. Upper row: fluid domains for fifteen processors (indicated by color), bottom row: gray-scale schlieren of fluid density on domains of refinement levels (indicated by color).

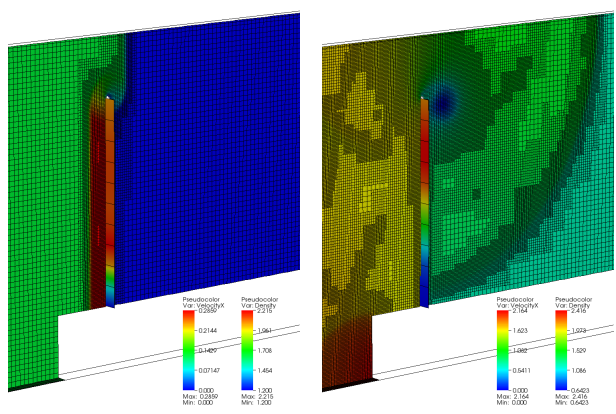


Figure 6: Dynamic mesh adaptation in fluid and deflection of the panel after $t \approx 0.06$ ms (left) and $t \approx 0.24$ ms (right).

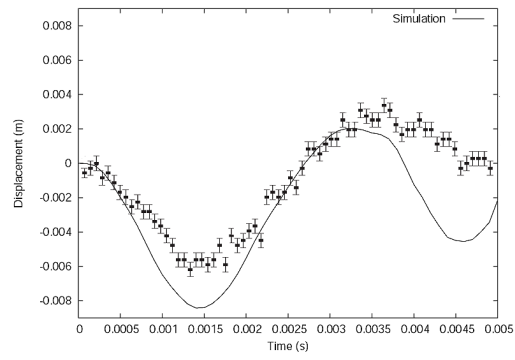


Figure 7: Panel tip displacement vs. time in FSI simulation and experiment [14].

time steps (sub-iterations) were taken in DYNA3D within one coupled FSI time step. Fifteen processors were dedicated to AMROC, one to the serial DYNA3D code.

The lower row of Fig. 5 visualizes the dynamic bending of the plate strip and the evolving fluid mesh adaptation with two additional levels (indicated by color on the back plane) as the initial shock is partially reflected and increased vortex shedding occurs at the panel tip at later times. The upper row of Fig. 5 displays the domain partitions of the fifteen fluid processors (indicated by color) for the two snapshots as the hierarchical mesh changes dynamically and the space-filling curve algorithm (cf. Section 4) leads to re-distribution of the fluid sub-domains to ensure balanced workload. In the left snapshot of Fig. 5 the adaptive computation uses 635,264 cells in 269 subgrids on the finest level, and 1,295,584 cells in 305 subgrids in the right, which would compare to 1,970,176 cells in the uniform case. In order to show the spatial extension of the used elements two enlarged snapshots of the panel and the neighboring fluid domain are depicted in Fig. 6. Figure 6 displays the fluid density ρ and the solid velocity v_1 by color.

A comparison of the simulated panel tip displacement over time versus the experimental measurements from Giordano et al. [14], shown with error bars, is given in Fig. 7. The agreement, especially at later times, is actually better than the computational results in [14], Fig. 10, which

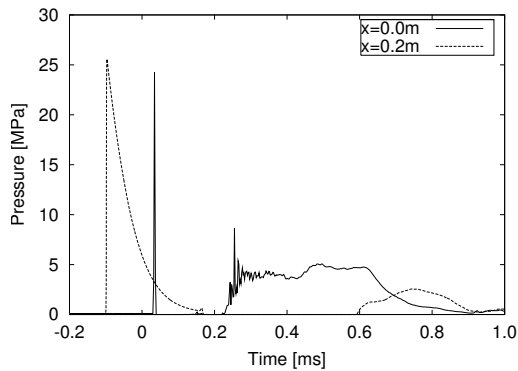


Figure 8: Fluid pressure traces along tube middle axis at $x_1 = 0$ and $x_1 = 0.2$ m.

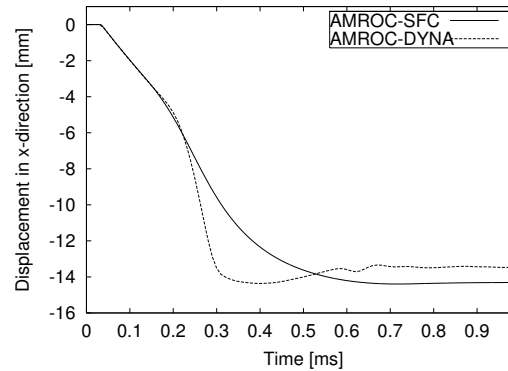


Figure 9: Plate center displacement vs time comparing current and previous results [8].

is likely due to a significantly finer effective resolution, thanks to the availability of SAMR, in the AMROC fluid solver.

7 Simulation of water-hammer-induced plate deformation

The second configuration considered is a computational setup that was described in great detail in [8]. We simulate the experiment, developed by Deshpande et al. [11], that reproduces loading conditions of large-scale underwater explosions in the laboratory. By firing a steel projectile onto a piston inserted into the end of a water shock tube, a strong pressure wave is created that propagates through the water column and impinges onto a circular copper plate sealing the other end. While in [8] the thin-shell research finite element solver SFC [4] was used with a special viscoplastic material law formulation, we employ in here DYNA3D with its standard isotropic elastic-plastic material model. The finally deformed plate configuration using AMROC-SFC had been found to be in excellent agreement with experimental results by Deshpande, cf. [8], and in here we therefore only compare with our previous computational results. As in [8], we model cavitation in the liquid with a simple pressure cutoff model that is implemented by applying the non-conservative energy correction

$$E := \frac{p_c + \gamma p_\infty}{\rho(\gamma - 1)} + \frac{1}{2} \mathbf{u}^T \mathbf{u}, \quad \text{for } p < p_c \quad (4)$$

after every fluid time step and its purpose is to limit all hydrodynamic pressures to a cutoff value p_c , which in here is set to $p_c \equiv 0$.

The shocktube has a length of 1.3 m and a radius of 32 mm, which is considered in the computational setup with an additional signed distance level set function for a cylinder. The tube is filled with water of density $\rho = 1000 \text{ kg/m}^3$ at atmospheric pressure $p_0 = 101.3 \text{ kPa}$ ($\gamma = 7.415$, $p_\infty = 296.2 \text{ MPa}$, $c = 1482 \text{ m/s}$). The piston motion creates a quasi one-dimensional pressure spike with exponential decay rate that propagates through the water column. The realistic piston movement is considered in the computation by employing a third signed distance level set function that corresponds to the piston boundary in contact with the fluid. The level set is initially positioned at $x_1 = 1.3 \text{ m}$ and assumed to move initially with constant velocity b_0 . During a simulation, we integrate the law of motion for the piston

$$\bar{m} \dot{b} = -(\bar{p} - p_0) \quad (5)$$

with the forward Euler method and update level set position and velocity b in direction of the tube middle axis in every time step. The hydrodynamic pressure averaged across the piston boundary \bar{p} is evaluated on the fly. In this computation, the initial velocity is $b_0 = -22.94 \text{ m/s}$

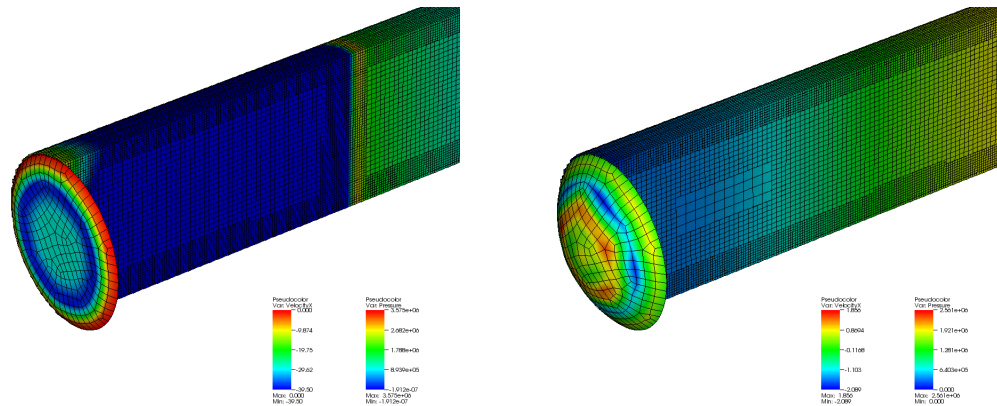


Figure 10: Mesh adaptation in the fluid domain and plastic deformation of the plate after simulating $t \approx 0.11$ ms (left) and $t \approx 0.80$ ms (right) physical time in the FSI simulation. Solid velocity v_1 and fluid pressure p are displayed by color.

and the averaged mass per unit area is $\bar{m} = 74.1 \text{ kg/m}^2$, which gives rise to a decaying pressure profile with initial maximum $\sim 34 \text{ MPa}$. Detailed depictions of the loading profile and comparison to theoretical estimates are given in [8]. For efficiency, we have computed the initial loading profile only once and re-use it as file initial conditions for the fluid domain in the FSI computations. In all subsequent visualizations the time is thereby shifted by -0.82 ms setting the start of the FSI simulations to $t = 0$.

The plate to be simulated is a copper plate of thickness of $h = 0.25 \text{ mm}$ that is initially located at the coordinate origin. The shocktube middle axis is aligned with the x_1 -axis. We use $\rho_s = 8920 \text{ kg/m}^3$, elasticity modulus $E = 130 \text{ GPa}$ and Poisson ratio $\nu = 0.31$. The initial yield stress is set to $\sigma_o = 38.5 \text{ MPa}$ and we use tangent modulus $E_T = 3 \text{ GPa}$ and hardening parameter $\beta = 0.5$ for DYNA3D material model #3. The plate is discretized with 327 four-node shell elements and the nodes along the plate boundary have zero displacements in all three spatial directions. The fluid base mesh consists of $350 \times 20 \times 20$ cells and uses two additional levels of refinement with $r_{1,2} = 2$, where mesh adaptation is based on ϕ and the scaled gradient of p . The coupling level is $l_c = 2$ and 10 sub-iterations in each FSI time step are employed in the DYNA3D solver. The simulation was run on the same computer system as the one in the previous section and again fifteen processors were used to update the fluid and one for the DYNA3D solver. 2192 coupled time steps were calculated, which required ~ 97 h CPU (~ 6.1 h wall time).

Figure 8 shows pressure traces along the tube middle axis at the locations $x_1 = 0.2 \text{ m}$ and $x_1 = 0$. The plot for $x_1 = 0.2 \text{ m}$ exhibits the characteristic exponential profile. Note, however, that the maximum shortly before the target plate is smaller than at the piston because of the compressibility of the liquid. In the plot for $x_1 = 0$ the importance of cavitation consideration can be inferred. The impact of the pressure wave onto the plate at $t \approx 0.03 \text{ ms}$ leads to a rapid structural expansion that lets the hydrodynamic pressure drop to p_c . During the cavitation phase basically no loads are transferred to the plate and the center displacement, displayed in Fig. 9, is linear. After the fluid expansion wave leaves the plate boundary, a hydrodynamic pressure of $\sim 5 \text{ MPa}$ builds up again directly at the plate until it declines finally from $t \approx 0.65 \text{ ms}$ on, cf. Fig. 8. During this phase, the plate deformation is nonlinear and viscoplastic. The difference in material models used in the AMROC-SFC and AMROC-DYNA3D computations is clearly visible in this phase, however, the obtained maximum deflection is basically identical and close to the analytic estimate of 16.1 mm by Qui and Deshpande [21]. Figure 10 shows two snapshots of the computational mesh used for the plate and in the neighboring part of the shocktube. The dynamic resolution adaptation of the fluid cells and the deformation of the plate can be inferred. In both snapshots roughly 735,000 cells are used in ~ 230 subgrids on level 2, $\sim 360,000$ cells

in ~ 95 subgrids on level 1, and 140,000 in 85 subgrids on the base level. A uniformly refined computation would use 8,960,000 cells making the benefit of dynamic adaptation apparent.

8 Conclusions

We have described a generic approach that enables the utilization of Cartesian finite volume upwind schemes in explicit multiphysics fluid-structure interaction simulations. Level-set-based boundary embedding and dynamic mesh adaptation are combined to effectively accommodate for large structural deformations in hydrodynamic shock-capturing solvers. Two different fluid-structure interaction configurations are studied, using DYNA3D as computational solid dynamics solver, to validate the implementation and to demonstrate the computational performance. The obtained results are found to be in good agreement with experimental observations and theoretical predictions, which confirms the suitability of the approach and its practical relevance. The Virtual Test Facility software, including the latest version of AMROC, is freely available from <http://www.cacr.caltech.edu/asc>.

Acknowledgments

R. D. is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. All computations were performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725. The DYNA3D meshes and input files used were created by Stephen Wood during a summer undergraduate internship sponsored by the Department of Energy/Florida International University Science & Technology Workforce Development Initiative. First prototypical data transfer codes coupling DYNA3D to AMROC were developed by Patrick Hung and Julian C. Cummings at the Center for Advanced Computing Research at the California Institute of Technology.

References

- [1] M. Arienti, P. Hung, E. Morano, and J. E. Shepherd. A level set approach to Eulerian-Lagrangian coupling. *J. Comput. Phys.*, 185:213–251, 2003.
- [2] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1988.
- [3] F. Cirak, R. Deiterding, and S. P. Mauch. Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin shells subjected to shocks and detonations. *Computers & Structures*, 85(11-14):1049–1065, 2007.
- [4] F. Cirak and M. Ortiz. Fully c^1 -conforming subdivision elements for finite deformation thin-shell analysis. *Int. J. Numer. Meth. Engineering*, 51:813–833, 2001.
- [5] R. Deiterding. *Parallel adaptive simulation of multi-dimensional detonation structures*. PhD thesis, Brandenburgische Technische Universität Cottbus, Sep 2003.
- [6] R. Deiterding. Construction and application of an AMR algorithm for distributed memory computers. In T. Plewa, T. Linde, and V. G. Weirs, editors, *Adaptive Mesh Refinement - Theory and Applications*, volume 41 of *Lecture Notes in Computational Science and Engineering*, pages 361–372. Springer, 2005.
- [7] R. Deiterding. A parallel adaptive method for simulating shock-induced combustion with detailed chemical kinetics in complex domains. *Computers & Structures*, 87:769–783, 2009.

- [8] R. Deiterding, F. Cirak, and S. P. Mauch. Efficient fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to underwater shock loading. In S. Hartmann, A. Meister, M. Schäfer, and S. Turek, editors, *Int. Workshop on Fluid-Structure Interaction. Theory, Numerics and Applications, Herrsching am Ammersee 2008*, pages 65–80. kassel university press GmbH, 2009.
- [9] R. Deiterding, R. Radovitzki, S. Mauch, F. Cirak, D. J. Hill, C. Pantano, J. C. Cummings, and D. I. Meiron. Virtual Test Facility: A virtual shock physics facility for simulating the dynamic response of materials. Available at <http://www.cacr.caltech.edu/asc>.
- [10] R. Deiterding, R. Radovitzky, S. P. Mauch, L. Noels, J. C. Cummings, and D. I. Meiron. A virtual test facility for the efficient simulation of solid materials under high energy shock-wave loading. *Engineering with Computers*, 22(3-4):325–347, 2006.
- [11] V. S. Deshpande, A. Heaver, and N. A. Fleck. An underwater shock simulator. *Royal Society of London Proceedings Series A*, 462(2067):1021–1041, 2006.
- [12] R. P. Fedkiw. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *J. Comput. Phys.*, 175:200–224, 2002.
- [13] R. P. 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.
- [14] J. Giordano, G. Jourdan, Y. Burtschell, M. Medale, D. E. Zeitoun, and L. Houas. Shock wave impacts on deforming panel, an application of fluid-structure interaction. *Shock Waves*, 14(1-2):103–110, 2005.
- [15] J. Hallquist and J. I. Lin. A nonlinear explicit three-dimensional finite element code for solid and structural mechanics. Technical Report UCRL-MA-107254, Lawrence Livermore National Laboratory, 2005. Source code (U.S. export controlled) available for licensing fee from <http://www.osti.gov/estsc>.
- [16] R. J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge University Press, Cambridge, New York, 2002.
- [17] R. Löhner, J. Baum, C. Charman, and D. Pelessone. Fluid-structure interaction simulations using parallel computers. volume 2565 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2003.
- [18] S. P. Mauch. *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, 2003.
- [19] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.
- [20] M. Parashar and J. C. Browne. On partitioning dynamic adaptive grid hierarchies. In *Proc. of the 29th Annual Hawaii Int. Conf. on System Sciences*, Jan 1996.
- [21] X. Qui, V. S. Deshpande, and N. A. Fleck. Dynamic response of a clamped circular sandwich plate subject to shock loading. *Int. J. Numer. Meth. Engineering*, 71:637–645, 2004.
- [22] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 1999.
- [23] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4:25–34, 1994.