Parallel adaptive fluid-structure interaction simulation of explosions impacting on building structures

R. Deiterding[a,*], S. Wood[b]

[a] *Oak Ridge National Laboratory P.O. Box 2008, MS-6367, Oak Ridge, TN 37831, USA*
[b] *University of Tennessee - Knoxville, The Bredesen Center, 821 Volunteer Blvd., Knoxville, TN 37996, USA*

Abstract

We pursue a level set approach to couple an Eulerian shock-capturing fluid solver with space-time refinement to an explicit solid dynamics solver for large deformations and fracture. The coupling algorithms considering recursively finer fluid time steps as well as overlapping solver updates are discussed. Our ideas are implemented in the AMROC adaptive fluid solver framework and are used for effective fluid-structure coupling to the general purpose solid dynamics code DYNA3D. Beside simulations verifying the coupled fluid-structure solver and assessing its parallel scalability, the detailed structural analysis of a reinforced concrete column under blast loading and the simulation of a prototypical blast explosion in a realistic multistory building are presented.

Keywords: Fluid-structure interaction, adaptive mesh refinement, Cartesian embedded boundary method, building structures, 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 (e.g. Banks et al. (2012)), for which the discretizations both in fluid and solid are usually time-explicit and therefore computationally comparably inexpensive. On the other hand,

[*] Corresponding author

Email addresses: `deiterdingr@ornl.gov` (R. Deiterding), `swood@utk.edu` (S. Wood) major geometric complexities, such as large structural deformations (Wang et al., 2012), 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 (Mittal and Iaccarino, 2005), 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 (Zhao et al., 2008).

Here, we employ our verified generic Cartesian fluid solver framework AMROC (Deiterding, 2005b, 2009, 2011; Ziegler et al., 2011) that implements a ghost fluid approach (Fedkiw, 2002) and relies on a scalar level set function, storing the distance to the nearest boundary facet of the solid's triangulation, to represent the embedded geometry on the fluid grid (Arienti et al., 2003). 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 (Löhner et al., 2003; Cirak and Radovitzky, 2005). 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) and was first released as public domain software in fall 2007 (Deiterding et al., 2007). Successful FSI applications of the VTF software include, for instance, blast waves impinging on deforming viscoplastic materials modeled with a volumetric finite element method (Deiterding et al., 2006; Moore et al., 2009), detonation waves in combustible gases causing the fracture of piping using a thin-shell finite element approach (Cirak et al., 2007), strong pressure waves in water inducing the rupture of metallic plates (Deiterding et al., 2009), the response of tubes made of fiber composites (Perotti et al., 2013), or simulation of blunt bodies and parachutes in supersonic flows (Laurence et al., 2011; Karagiozis et al., 2011).

In this paper, we give a brief overview of the computational methodologies used and – for the first time – present FSI simulations in which AMROC is coupled through VTF modules to the serial version of the general purpose explicit solid mechanics solver DYNA3D by Hallquist and Lin (2005) and used to simulate prototypical blast explosions impacting on realistic building structures. The presentation is organized as follows: In Sections 2 and 3 we give an overview of the respective aspects of the AMROC and the DYNA3D solver that are relevant to this paper. Section 4 contains a presentation of the FSI coupling methodology considering adaptive space-time refinement in the fluid solver and parallel computation of fluid and solid update steps. In Section 5, four configurations of increasing complexity are discussed: a simple elastic panel under planar shock impact, a parallel scalability assessment of a blast explosion under a realistic highway bridge, analysis of a blast corresponding to the Oklahoma bombing report (Mlakar Sr. et al., 1998) on a concrete column reinforced with steel, and finally, the simulation of a blast explosion event in the lobby of a complex seven-story building. The conclusions follow in Section 6.

2. AMROC adaptive fluid dynamics solver

2.1. Governing equations

The equations solved in AMROC for the purpose of this paper are the Euler equations

$$\begin{aligned}\partial_t\rho + \nabla\cdot(\rho\mathbf{u}) = 0\ ,\\ \partial_t(\rho\mathbf{u}) + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) + \nabla p = 0\ ,\\ \partial_t(\rho E) + \nabla\cdot((\rho E + p)\mathbf{u}) = 0.\end{aligned} \tag{1}$$

In the latter, ρ is the fluid density, $\mathbf{u}$ the velocity vector, and E the specific total energy. The hydrostatic pressure p is given by the polytropic gas equation $p = (\gamma - 1)(\rho E - \frac{1}{2}\rho\mathbf{u}^T\mathbf{u})$ with $\gamma = 1.4$ denoting the constant adiabatic exponent of air.

2.2. Numerical method

To solve Eq. (1) numerically, we apply a time-explicit shock-capturing finite volume scheme based on Roe's approximate Riemann solver (Toro, 1999) specially hybridized with the Harten-Lax-vanLeer (HLL) scheme to ensure strict positivity preservation (Deiterding, 2003). Second-order accuracy in smooth solution regions is achieved with the MUSCL-Hancock variable extrapolation technique.

The boundary geometry is mapped onto the Cartesian mesh with 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 $\mathbf{n} = -\nabla\varphi/|\nabla\varphi|$ (Deiterding, 2009). Our implementation allows both the use of *signed* distance level set functions for representing volumetric elements (Deiterding et al., 2006) as well as *unsigned* distance level set functions to consider thin-shell elements (Cirak et al., 2007). In the signed distance case, a fluid cell is treated as an embedded ghost cell if its *midpoint* satisfies $\varphi < 0$. For thin-shell elements, which have a mesh only within the element midplane and implicitly assumed constant thickness h, we employ the condition $\varphi < h/2$ and additionally evaluate the hydrodynamic load on each thin element as the difference between the approximated pressure values at $\varphi = h/2$ in the positive and negative direction of each element's normal.

The vector of state in embedded ghost cells is then adjusted to model the boundary conditions of a non-Cartesian reflective wall moving with velocity $\mathbf{v}$ before applying the unaltered Cartesian finite volume discretization. The last step involves interpolation and mirroring of ρ, $\mathbf{u}$, p across the boundary and modification of the normal velocity in the immersed boundary cells to $(2\mathbf{v} \cdot \mathbf{n} - \mathbf{u} \cdot \mathbf{n})\mathbf{n}$, cf. (Deiterding, 2011). Here, we employ a dimension-wise linear interpolation operation that can decrease the number of interpolants directly near the boundary to ensure the monotonicity of the numerical solution (Deiterding, 2009).

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 facet on the solid surface mesh. For this purpose, we employ a specially developed algorithm based on characteristic reconstruction and scan conversion developed by Mauch (2003) that is used to compute the distance exactly only in a small band around the embedded structure. In the following this algorithm is denoted as *Closest Point Transform* (CPT).

2.3. Parallel adaptive mesh refinement

As it is characteristic for immersed Cartesian techniques, the boundary treatment described in the previous section results in some geometric approximation inaccuracies (Mittal and Iaccarino, 2005). We mitigate this problem by refining the embedded boundary dynamically during the computation, typically up to the highest available resolution. A refinement criterion based on $\varphi = 0$ has been implemented for this purpose.

For local dynamic mesh adaptation we have adopted the block-structured adaptive mesh refinement (SAMR) method after Berger and Colella (1988) that is tailored especially for hyperbolic conservation laws on logically rectangular grids. In this approach, finite volume cells are clustered with a special algorithm into non-overlapping computationally effective rectangular grids. Refinement levels are integrated recursively using hierarchical time step refinement. Spatial and temporal mesh widths on level l are r_l-times finer than on level $l-1$, and a time-explicit finite volume scheme will (in principle) remain stable on all levels of the hierarchy. Here, we assume $r_l \geq 2$ for $l > 0$ and $r_0 = 1$.

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 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 (Deiterding, 2005a). Partitions with similar workload are found at runtime as the hierarchy evolves by a domain decomposition algorithm based on a generalization of Hilbert's space-filling curve (Parashar and Browne, 1996). 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. Benchmark results (not shown here) exhibit good scalability for typical SAMR fluid-only benchmarks on several thousand processors.

Finally, it is worth mentioning that the described computational techniques are equally applicable to viscous high-speed flows. For instance, Ziegler et al. (2011) utilize AMROC to simulate the chemically reactive Navier-Stokes equations. In addition to an upwind scheme for convection, time-explicit conservative central difference stencils are used to approximate the viscous fluxes. The previously described embedded boundary method can easily be adjusted to impose no-slip boundary conditions but note that the resolution along the embedded boundary typically needs to be significantly finer than for Euler equations, cf. (Mittal and Iaccarino, 2005). The adaptive mesh refinement methodology of AMROC provides an effective means for coping with these increased resolution requirements in space as well as in time.

3. DYNA3D solid mechanics solver

3.1. Governing equations

DYNA3D is an explicit, nonlinear, finite element code for problems where high rate dynamics or stress wave propagation effects are important. It uses a lumped mass formulation for efficiency. This yields a diagonal mass matrix $\mathbf{M}$, which enables a direct solution of the momentum equation

$$\mathbf{M}\mathbf{a}_{n+1} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}, \tag{2}$$

where $\mathbf{f}^{\text{ext}}$ are the externally applied forces, $\mathbf{f}^{\text{int}}$ the internal element forces and $\mathbf{a}_{n+1}$ the accelerations of the new time step indexed $n+1$ from which updated nodal positions and velocities are computed, cf. (Hallquist and Lin, 2005).

The basic continuum finite element in DYNA3D is the eight-node hexahedral solid element. This element is valid for large displacements and large strains. DYNA3D also supports several quadrilateral four-node thin-shell elements. Hexahedral and quadrilateral elements may be degenerated to wedge and tetrahedral elements, or triangular elements respectively. Degenerate elements are less accurate and may lock due to excessive transverse shear (Hallquist and Lin, 2005).

3.2. Material models

DYNA3D supports numerous models suitable for a variety of materials and loading regimes. In here, we have employed the linearly elastic (Model #1) and the elastic-plastic with failure (Model #13) material models. Beside the rigid ground model, we have used only Model #1 in the panel and bridge setups of Sections 5.1 and 5.2. The more sophisticated configurations of the fracturing reinforced column (Section 5.3) and collapsing building used primarily the elastoplastic Model #13 that considers linear strain hardening before element failure. The required material model parameters are listed in Tables 1 and 2, respectively. In Model #13, the hardening parameter, β, specifies a combination of kinematic and isotropic hardening; $\beta = 0.0$ yields purely kinematic hardening, while $\beta = 1.0$ gives purely isotropic hardening (the same behavior as Model #1). Plastic deformation requires the perfectly plastic yield function

$$\phi = \bar{\sigma} - \sigma_y(\bar{\epsilon}^p) \tag{3}$$

to be positive. Otherwise, elastic material behavior is assumed. In Eq. (3), $\bar{\sigma}$ is the effective stress and σ_y is the current yield stress which may be a function of the effective plastic strain $\bar{\epsilon}^p$ if strain hardening is included. The effective stress is dependent on the deviatoric stress tensor for isotropic hardening and on the translated stress tensor for kinematic hardening. The linear isotropic hardening law has the form

$$\sigma_y = \sigma_0 + \beta E_p \bar{\epsilon}^p \quad , \tag{4}$$

where σ_0 is the initial yield stress, E_p is the plastic hardening modulus

$$E_p = \frac{E E_T}{E - E_T} \quad , \tag{5}$$

and the effective plastic strain is the integral in time of the incremental plastic strain tensor. The numerical algorithms used in this model are adapted from Krieg and Key (1976).

Element failure occurs when either the effective plastic strain $\bar{\epsilon}^p$ is reached or when the pressure in the element becomes more tensile (negative) than the failure pressure p_f. These criteria are checked simultaneously in every DYNA3D time integration step. In here, elements are removed from the calculation when either failure criteria is satisfied. Deleted elements support no stresses – not even hydrostatic compression.

4. Fluid-structure interaction coupling

4.1. Coupling conditions

Fluid-structure interaction is assumed to take place only at the evolving interface $\mathcal{I}$ between fluid and solid and is implemented numerically by exchanging boundary data after consecutive time steps. In the case of inviscid flows, the boundary conditions along $\mathcal{I}$ are

$$v_n = u_n \ , \quad \sigma_{nn} = p \ , \tag{6}$$

with v_n and u_n denoting the velocity in the normal direction in solid and fluid, respectively. The solid stress tensor is denoted by σ; σ_{nn} are the normal stresses, and p is the fluid pressure. For compressible fluids, a stable coupling is achieved if the fluid solver with embedded boundary capability receives the velocities and the geometry of the solid surface, while only the hydrostatic pressure is communicated as a force acting on the solid's exterior (Löhner et al., 2003; Specht, 2000). A straightforward serial fractional step algorithm for this type of FSI coupling reads:

ALGORITHM 1: Fractional step fluid-structure coupling algorithm.

```
$u_n := v_n(t)|_{\mathcal{I}}$
update_fluid( $\Delta t$ )
$\sigma_{nn} := p(t + \Delta t)|_{\mathcal{I}}$
update_solid( $\Delta t$ )
$t := t + \Delta t$
```

4.2. Fluid-structure coupling of AMROC and DYNA3D

While the implementation of Algorithm 1 is straightforward with solvers with consecutive time update, the hierarchical time step refinement of the SAMR method introduces an additional complexity. Our core assumption in incorporating FSI coupling into the recursive SAMR algorithm is that the fluid-solid interface $\mathcal{I}$ is a-priori refined at least up to a predefined coupling level l_{fsi}. Note that the resolution at level l_{fsi} should be sufficiently fine to ensure an accurate wave transmission between fluid and structure, but might not necessarily be the highest level of refinement. From the point of view of the SAMR fluid solver, an updated interface geometry $\mathcal{I}$ and boundary velocities $\mathbf{v}|_{\mathcal{I}}$ should be received before the mesh is adapted on levels with $l \geq l_{\mathrm{fsi}}$. On the other hand, the interface pressure $p|_{\mathcal{I}}$ should be evaluated and communicated back to the solid solver only when the highest available refinement level has been computed. The algorithms that we have implemented in AMROC and DYNA3D following these principles are described in detail in Appendix A. Noteworthy in these pseudo-codes is also the parameter *update_type* that can be used to alter the FSI coupling data exchange from a sequential fractional step method, in the spirit of Algorithm 1, to a lagged communication approach that allows the parallel execution of the computationally intensive numerical update steps in both solvers. The benefit of this parameter will be demonstrated in Section 5.1.

In our actual implementation, the serial DYNA3D solid mechanics solver is run on a single processor and the AMROC fluid dynamics solver is assigned a fixed number of processors that is commensurate with the usually more compute intensive fluid problem. As sketched in Section 2.3, the SAMR mesh is redistributed among these nodes as the hierarchy is adapted. In order to facilitate an effective data transfer between the two coupled solvers nevertheless, a specially developed non-blocking, parallel Eulerian-Lagrangian coupling (ELC) module has been incorporated into the VTF software that determines the evolving communication patterns between the solid and fluid nodes. Details are given by Deiterding et al. (2006).

A sketch visualizing the data flow between AMROC and DYNA3D is shown in Figure 1. The problem-specific setup parameters are given within the hexagons at the top of the figure. Note that our specific interface codes are able to regenerate the solid surface mesh $\mathcal{I}$ from DYNA3D-internal data structures in every coupling time step, which allows for topology changes and dynamic element removal. All found exterior element facets are automatically set to DYNA3D's pressure boundary condition type. Querying DYNA3D-internal data structures provides the updated coordinates and velocities of the nodes associated to these facets. In this paper, the level set function φ is always computed from the triangulated solid surface mesh $\mathcal{I}$ by applying the CPT algorithm of Mauch (2003). But note that for simple bodies, e.g. rigid spheres, AMROC also provides the possibility to define level sets with analytic functions as used by Laurence et al. (2011).

5. Computational results

5.1. Simulation of shock-induced elastic panel deflection

As a first, easily reproducible test for the coupling, we simulate the quasi-two-dimensional verification configuration of a thin-walled steel panel impacted by a planar shock wave in air proposed by Giordano et al. (2005). The panel has the thickness $h = 1\,\mathrm{mm}$ and extends 50 mm from a mounting with forward-facing step geometry into which it is firmly clamped. The fluid domain has the extensions 400 mm and 80 mm in the x- and y-direction, respectively. The step has a height of 15 mm and starts 135 mm from the left boundary. The fluid domain and the panel extend 5 mm into the z-direction. Inflow boundary conditions are applied on the left side, rigid wall boundary conditions anywhere else. The panel is modeled in DYNA3D with only ten 4-node thin-shell elements. The material is assumed to be linearly elastic with material parameters given in Table 1. 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 refinement (based on φ and scaled gradients of ρ and p) with refinement factors $r_{1,2} = 2$.

The discontinuous shock wave is placed at $x = 130\,\mathrm{mm}$ and travels into air at rest of density $\rho_0 = 1.2\,\mathrm{kg/m^3}$ and pressure $p_0 = 100\,\mathrm{kPa}$. The conditions behind the shock are $\rho_s = 1.6458\,\mathrm{kg/m^3}$, $u_{x,s} = 112.61\,\mathrm{m/s}$, $u_{y,s} = 0$, $p_s = 156.18\,\mathrm{kPa}$. We carry out two computations with $l_{\mathrm{fsi}} = 2$ to a final physical time $t_e = 5\,\mathrm{ms}$ in which the algorithmic parameter *update_type* is varied. Four sub-iterations of the DYNA3D solver are used and the computations were run on eight 3.4 GHz dual-processor nodes connected with a 1 Gb Ethernet network. The computations use dynamic time step adjustment that is primarily based on aiming at a constant CFL number of 0.8 in the fluid solver. The computation with the setting *update_type=sequential* computed $19,768$ coupled time steps to reach t_e after ~ 23.9 h wall time; the computation with *update_type=parallel* reached t_e after $19,876$ coupled time steps and required nevertheless only ~ 23.4 h wall time. Although the used DYNA3D model has exceptionally few elements, the latter result provides first evidence for the computational advantage of overlapping both solver updates in parallel.

Figure 2 visualizes the dynamic bending of the plate strip and the evolving fluid mesh adaptation as the initial shock is partially reflected (a) and vortex shedding occurs at the tip as the panel visibly deforms (b). Only the midplane of the thin-shell elements is shown and fluid cells with $\varphi < h/2$, that are used to impose embedded boundary conditions, are removed from the visualization.

A comparison of the simulated panel tip displacements over time versus the experimental measurements by Giordano et al. (2005) is given in Figure 3. Both simulations predict an almost identical panel oscillation, which exhibits a general discrepancy with the experimental results towards the end of the simulation. Nevertheless, the agreement of our FSI simulations, especially at later times, is actually better than the computational results given by Giordano et al. (2005), Figure 10, which is likely due to a significantly finer effective resolution – thanks to the availability of SAMR – in the AMROC fluid solver. Note also the marginal influence of using the more efficient data exchange strategy (*update_type=parallel*) on the computational results. Here, as well as in (Giordano et al., 2005), heat transfer effects between fluid and solid are neglected in the numerical simulations, which seems justified as the air in the shock tube is at room temperature and the temperature behind the shock is only ~ 40 K higher.

5.2. Simulation of blast explosion under a highway bride

The next configuration is more complex and follows a study by Agrawal and Yi (2009) in which the explosion of 150 kg TNT in front of the high middle column of a highway bridge is investigated. Figure 4 depicts the overall setup. The blast loading is positioned 0.5 m in front of the central column on the high side of the bridge, 2 m above the ground plane and along the column's centerline. The fluid is initialized as air at rest at ambient pressure $p_0 = 100\,\text{kPa}$ and temperature $T_0 = 293\,\text{K}$, where the ideal gas equation

$$\rho = \frac{pW}{\mathcal{R}T} \tag{7}$$

with $W = 29.08\,\text{g/mol}$ and $\mathcal{R} = 8.31441\,\text{J/(mol K)}$ is used to compute the ambient density. The blast explosion is initialized by setting the internal energy, e_i, of the fluid in a small sphere to the equivalent energy release by TNT, which we assume to be $4,520,000\,\text{J/kg}$. The uniform fluid pressure in this sphere is specified as

$$p = (\gamma - 1)e_i \left(\frac{4}{3}\pi r^3\right)^{-1}$$

and the density is computed from Eq. (7) assuming a temperature of 1465 K. For the present case an energy increase equivalent to 750 kg TNT in a sphere of radius $r = 0.4\,\text{m}$ is used. Figure 4 shows the air pressure on a slice plane along the bridge's centerline.

For this prototype simulation, only the elastic material model (cf. Table 1) and the rigid ground model of DYNA3D are employed. The solid mesh has $3,365$ solid hexahedral elements; an SAMR base mesh of $240 \times 40 \times 80$ cells refined on the fly with two additional levels with refinement factors $r_1 = 2$ and $r_2 = 4$ is applied. The coupling level is $l_{\text{fsi}} = 2$ and no solid sub-iterations are used. Enlarged snapshots of the critical column and the explosion are shown in Figure 5 and the dynamic mesh adaptation and solid deformation are visible. To reach a final time of $t_{\text{end}} = 20\,\text{ms}$, $1,504$ coupled time steps were computed (target fluid CFL ~ 0.6). Using 63 processors to simulate the fluid and one processor for the solid model, this required 487 h CPU or 7.6 h wall clock time on the Intel-Xeon-3.4-GHz dual-processor cluster (*update_type=sequential*).

In order to assess the parallel performance of the AMROC-DYNA3D code, a scalability study has been conducted for the first 24 coupled time steps. In this study, a single processor is always dedicated to execute the serial DYNA3D solver while the number of fluid processors is doubled consecutively, yielding a strong scalability test for the parallel adaptive fluid solver in combination with the FSI coupling routines. During the benchmarked time steps, the adaptive CFD solver used $\sim 56,500,000$ finite volume cells instead of $393,216,000$ (uniform mesh). Figure 6 visualizes the computational expense of the most important task groups in the AMROC solver as the total processor count is increased. *Integration* denotes the expense of the single-block Cartesian finite volume update routine, *Fluid Sync.* combines all communication costs in order to synchronize the ghost cells of neighboring blocks across processor borders. *Interpolation* accumulates the cost for inter- and extrapolation routines inherent to the SAMR as well as the ghost fluid method (GFM). All remaining operations of the GFM and the level set computation via the CPT routines are combined in *GFM + Level Set*, and *Regridding* accumulates all costs for parallel mesh adaptation, that is, modification of the hierarchy and parallel data redistribution. Finally, *Coupling Data* accumulates the costs for identifying finite volume cells near the embedded solid mesh and constructing pressure values for coupling; *Coupl. Comm.* measures the costs for communicating the FSI coupling data, which primarily consists of the waiting time required to receive an updated surface mesh and velocities from the DYNA3D solid solver.

From Figure 6 is can be inferred that all strictly local operations, that is, *Integration*, *Interpolation*, *GFM + Level Set*, and *Coupling Data* show ideal linear scaling. *Fluid Sync.* and *Regridding* involve parallel communication within the AMROC fluid solver and it is to be expected that in a strong scalability test the measured speed-up is less than ideal. A performance improvement is still nevertheless achieved for 256 cores. However, *Coupl. Comm.* identifies a serious scalability bottleneck. Since DYNA3D is only available as a serial program, its execution times within this study are fixed. When running with setting *update_type=sequential*, all fluid processors are idle during the solid solver update and *Coupl. Comm.* corresponds to the combined expense of the constant solid solver costs plus the communication costs, which rise because a single solid processor needs to communicate with an increasing number of fluid processes.

In the right plots of Figure 6, we demonstrate how this bottleneck, which is intrinsic to coupling a serial to a parallel solver, can at least be mitigated. Firstly, we set the coupling level to $l_{\mathrm{fsi}} = 1$ and employ 4 sub-iterations in the solid solver, which reduces the number of FSI data exchanges by 75%. Secondly, we use *update_type=parallel*, which overlaps the fluid and solid updates. The first measure reduces the costs of *Coupling Data* to roughly 1/3, the combination of both shrinks the costs of *Coupl. Comm.* by $\sim 77\%$ in case of the 256 processor benchmark. Using sub-iterations in the solid solver as well as the lagged coupling data communication strategy can reduce the stability of the overall FSI simulation and careful verification is required. Yet, already for this small solid mesh example the overall runtime in the

256 processor case was reduced by $\sim 12\%$ and using *update_type=parallel* has proved vital for us for considering realistic solid models. This setting is hence applied in both subsequent configurations.

5.3. Single reinforced concrete column

This realistic case study follows the setup described by Ngo et al. (2007), Section 4. We examine a reinforced high strength concrete (HSC) column which is 6.4 m tall with a cross-section of 500 × 900 mm. The steel rebar reinforcement configuration is 20–N32 vertical rebars and a ligature of N12@400 mm. Both the HSC and the steel are modeled with the elastic-plastic model with failure (Model #13) and the parameter values given in Table 2. Flexure and failure of concrete and rebar is considered but friction between the concrete and rebar is not. The blast loading used is based on the Oklahoma bombing report by Mlakar Sr. et al. (1998) which describes the detonation as $\equiv 1,814$ kg TNT at a stand-off distance of 11.2 m and 2 m above ground. The fluid initial conditions are specified as explained in the previous section where here a sphere of radius $r = 1$ m is employed. The surface mesh for FSI coupling is regenerated in every coupled time step to capture the evolving external surface as elements of HSC and rebar fail and are deleted. The fluid used a base resolution of $20 \times 64 \times 140$ cells and two levels of adaptive mesh refinement with $r_{1,2} = 2$. The reinforced column was modeled with $56,056$ hexahedral elements.

Ngo et al. (2007) apply a simplified triangular shape blast profile with a peak load of 10 MPa and a positive phase duration of 1.3 ms uniformly to one of the 900 mm column faces. They utilize a strain-rate dependent material model that allows for elastoplastic behavior and determine column failure using the Modified Compression Field theory of Veccio and Collins (1986).

To compare our results with the simplified loading, elastoplastic material model, and holistic failure criteria employed by Ngo et al. (2007) we first place the column in a fluid domain with all outflow boundary conditions and position the detonation at a height of 3.2 m (half the column's total height) along the column centerline. The pressure history and deflection results from this setup are denoted with "all outflow" in Figures 8 and 9. The spherical blast is reflected and diffracted as it impinges on the column producing the second peak in the mid-point pressure history. Also, we directly simulate the blast loading from (Mlakar Sr. et al., 1998) by placing the explosion center at a height of 2 m and consider the ground by using reflective boundary conditions on the lower boundary of the fluid domain. This computation required 377.5 h CPU on 7+1 cores of the Intel-Xeon cluster ($\sim$ 24 h wall time). 14, 152 coupled time steps were computed. As can be seen in Figure 7, the blast front now reflects off the rigid ground and has lost the initial spherical shape when reaching the column. Shock focusing in the corner formed by the column and ground increases the peak pressure by 67 %. The significant asymmetries in the blast loading are shown by the "ground" pressure histories in Figure 8.

A comparison of lateral deflection histories for the mid-point of the blast facing side of the column (column mid-point) with the results by Ngo et al. (2007) that used a linear loading profile and no coupled FSI simulation are given in Figure 9. There is excellent agreement between the deflection histories for the "all outflow" simulation configuration and the results of Ngo et al. (2007), which predict shear failure of the column at $\sim$ 6.0 cm deflection, providing another verification result for the coupled solver. The importance and benefit of carrying out fully coupled FSI simulations with realistic boundary conditions is illustrated by the "ground" simulation result, in which this critical deflection value is reached more than 1 ms earlier.

Half of the HSC in the fracturing column is not shown in Figure 10 to reveal internal failure of concrete and rebar elements at 11.5 ms and $\sim$ 3.3 cm of deflection. The detail view of the column shows that significant failure has already begun due to the realistic blast loading in the"ground" simulation configuration. Study of the element failure propagation pattern can provide valuable hints for the design of a blast mitigating column.

5.4. Blast explosion in a multistory building

The last computational example is a blast explosion in a seven-story building similar to a case described by Luccioni et al. (2004). The building has roughly the dimensions 20 m × 40 m × 25 m and we have modeled it with 69, 709 hexahedral elements. Three different materials are used and their locations are depicted in Figure 11a. The parameters for the elastic-plastic material model with failure are listed in Table 2; the basement level walls are modeled with the elastic material model and use the same values as in Section 5.2. The fluid domain engulfs the building and uses a base grid of 80 × 120 × 90 cells and two levels of SAMR with refinement factors $r_{1,2} = 2$. The highest level resolution is around 8.4 cm. The embedded structures are refined up to level 1, the blast wave is refined up to the highest level available based on the scaled gradient of p. Figure 11 also visualizes the rigorous domain decomposition of the AMROC fluid solver to 31 processors by showing planes (originating in the explosion center) that are colored by processor number. Figure 11b shows this domain decomposition again but with solid mesh removed and additionally displays the adaptive grid at simulated time $t = 6.1$ms. The blast wave corresponds to the explosion of 400 kg TNT (fluid initial conditions as in Section 5.2, $r = 0.5$ m) and is positioned in the lobby of the building.

FSI coupling level is $l_{\text{fsi}} = 1$ and $1,070$ coupled time steps were computed, which required ~ 830 h CPU (~ 25.9 h wall time) on the Intel-Xeon cluster. Like in the previous configuration, the surface mesh for FSI coupling is regenerated in every coupled time step to allow for element removal. No solid sub-iterations are permitted. Figure 12 shows close-ups of the explosion event in the lobby. Figure 12a depicts the initial blast expansion and a cut through the fluid shows the mesh adapting to the pressure front. At $t = 29.2$ ms, the initial blast has reflected from the building walls and cracking of the floor plus massive deformation of some of the free-standing columns are visible. In the last snapshot, major element failure is occurring indicating massive structural failure. Note the ease of the level-set-based coupling approach to handle even this situation without any problems and without increase in computational costs. In the three snapshots shown the size of the adaptive fluid mesh decreases from $9,757,232$ (6.1ms) over $8,541,080$ (29.2 ms) to a total of $7,247,000$ cells (48.7 ms). A uniform mesh with the same highest level resolution would have $55,296,000$ cells.

6. Conclusions

A highly capable software system for FSI problems driven by shock and blast waves has been constructed. Verification results and simulation of blast explosion events in and around realistic three-dimensional building and civil engineering structures, including fracture and element failure and removal, demonstrate that the software can be used for sophisticated analysis and technically relevant design. While the utilization of an Eulerian fluid mesh with embedded boundary capability is nowadays frequently employed for FSI simulations with large structural motions, cf. (Wang et al., 2012; Zhao et al., 2008; Puso et al., 2012), we enhance its accuracy independent of a particular scheme by combining it with dynamic adaptation of the fluid mesh. Unique to our approach, we additionally employ recursive fluid time step refinement and the specially extended FSI coupling algorithms have been described.

While the AMROC adaptive fluid solver is fully parallelized, the adopted general purpose solid dynamics code DYNA3D is available to the majority of users only in a serial version, thereby creating a natural bottleneck for parallel scalability. To mitigate this problem, a lagged communication strategy is suggested that permits the execution of the numerical update steps in both solvers in parallel. The overlapped update is of crucial importance for considering solid models of realistic size. The discussed simulations indicate that its influence on typical results is small and a slight reduction in coupling stability – albeit unavoidable – of minor practical concern.

Acknowledgments

This material is based in part on work performed by S. Wood while being a RAMS intern and a TN-SCORE Energy Scholar. All computations were carried out at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

We appreciate having had access to first prototypical data transfer codes for solid models without topology evolution coupling DYNA3D to AMROC, which were developed by Patrick Hung and Julian C. Cummings from the Center for Advanced Computing Research at the California Institute of Technology.

Appendix A FSI coupling with SAMR and parallel update

Below, we describe the specific extension of the recursive SAMR algorithm for FSI coupling that we have implemented in AMROC. The new recursive main SAMR routine *advance_level()* is outlined in pseudo-code in Algorithm 2. This algorithm calls the routine *level_set_generation()* to evaluate the distance function φ^l for the actual level l based on the currently available interface $\mathcal{I}$ using the CPT algorithm of Mauch (2003). If level l is the coupling level $l_{\rm fsi}$ and *update_type=parallel*, the existing fluid data field is used to evaluate the pressure on the discrete vertices of $\mathcal{I}$. This data is then sent to the solid before the update of the discrete vector of state in the fluid, $\mathbf{Q}$, utilizing φ^l and present interface velocity, $\mathbf{v}|_{\mathcal{I}}$. The algorithm continues to proceed recursively to higher levels. When available, updated data from the next higher level is used to correct cells overlaid by refinement. When all higher level operations have been completed, updated solid mesh positions and nodal velocities are received. Note that the recursive order of the SAMR

Algorithm 2: Fluid-side FSI coupling considering the recursive SAMR method.

```
fluid_step()
  $\Delta\tau_F := \min_l(R_l \cdot$ stable_fluid_step($l$), $\Delta\tau_S$)
  $\Delta t_l := \Delta\tau_F / R_l$ for $l = 0, \cdots, l_{\max}$
  receive_interface_data($\mathcal{I}$, $\mathbf{v}|_{\mathcal{I}}$)
  advance_level(0)

advance_level($l$)
  Repeat $r_l$ times
      If time to regrid
          regrid($l$)
      level_set_generation($\varphi^l$, $\mathcal{I}$)
      If $l = l_{\mathrm{fsi}}$ and update_type=parallel
          send_interface_data($p(t)|_{\mathcal{I}}$)
      update_fluid_level($\mathbf{Q}^l$, $\varphi^l$, $\mathbf{v}|_{\mathcal{I}}$, $\Delta t_l$)
      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 $l = l_{\mathrm{fsi}}$
          if update_type=sequential
              send_interface_data($p(t+\Delta t_l)|_{\mathcal{I}}$)
          receive_interface_data($\mathcal{I}$, $\mathbf{v}|_{\mathcal{I}}$)
      $t := t + \Delta t_l$
```

algorithm automatically ensures that updated interface mesh information is thereby available at later time steps on coarser levels and to adjust the refinement grids on level l_{fsi}. For setting *update_type=sequential*, the computation and communication of interface pressures $p|_{\mathcal{I}}$ is deferred until all levels with $l \geq l_{\mathrm{fsi}}$ have been updated.

In order to achieve a proper matching of FSI communication operations, we start the cycle by posting a receive-message in the routine *fluid_step()* (which does one fluid time step on level 0) before entering into the SAMR recursion. *fluid_step()* also considers the base time step adjustment for the

Algorithm 3: Solid-side coupling considering the recursive SAMR method.

```
solid_step()
  $\Delta\tau_S := \min(K \cdot R_{l_{\mathrm{fsi}}} \cdot$ stable_solid_step(), $\Delta\tau_F$)
  Repeat $R_{l_{\mathrm{fsi}}}$ times
      $t_e := t + \Delta\tau_S / R_{l_{\mathrm{fsi}}}$, $\Delta t := \Delta\tau_S / (K R_{l_{\mathrm{fsi}}})$
      send_interface_data($\mathcal{I}(t)$, $\mathbf{v}|_{\mathcal{I}}(t)$)
      if update_type=sequential
          receive_interface_data($p|_{\mathcal{I}}$)
      While $t < t_e$
          update_solid($p|_{\mathcal{I}}$, $\Delta t$)
          $t := t + \Delta t$
          $\Delta t := \min($stable_solid_step(), $t_e - t)$
      if update_type=parallel
          receive_interface_data($p|_{\mathcal{I}}$)
```

SAMR method coupled to a solid solver. During one root level time step at level 0, the time steps on all levels remain fixed and are calculated in advance by employing the refinement factor with respect to the root level $R_l = \prod_{\iota=0}^{l} r_l$. The root level fluid time step $\Delta\tau_F$ is taken to be the minimum of the stable time step estimations from all levels and a corresponding time step $\Delta\tau_S$ in the solid. We define $\Delta\tau_S$ as a multiple of the stable time step estimation in the solid solver with respect to the communication frequency $R_{l_{\mathrm{fsi}}}$ in one fluid root level step and an additional factor K that allows subiterations in the solid solver in case of considerably smaller solid time steps.

The solid update algorithm used to advance the solid by one fluid root level step is given in Algorithm 3. The loop for integrating the solid $R_{l_{\mathrm{fsi}}}$ times is apparent. Similar to Algorithm 2, the parameter *update_type* is considered. With setting *update_type=sequential*, a simple fractional step algorithm involving consecutive send and receive operations of the coupling data is executed. With setting *update_type=parallel*, send and receive operations take place before and after the numerical update, respectively. Generally speaking, the setting *parallel* improves the scalability of the AMROC-DYNA3D solver (cf. Section 5.2), but reduces the numerical stability slightly. Technically, the coupled solver is programmed as a single SPMD binary that runs as either a solid or a fluid node, depending on processor number.

References

Agrawal, A.K., Yi, Z., 2009. Blast Load Effects on Highway Bridges. Technical Report. University Transportation Research Center, City College of New York.

Arienti, M., Hung, P., Morano, E., Shepherd, J.E., 2003. A level set approach to Eulerian-Lagrangian coupling. J. Comput. Phys. 185, 213–251.

Banks, J.W., Henshaw, W.D., Schwendeman, D.W., 2012. Deforming composite grids for solving fluid structure problems. Journal of Computational Physics 231, 3518 – 3547.

Berger, M., Colella, P., 1988. Local adaptive mesh refinement for shock hydrodynamics. J. Comput. Phys. 82, 64–84.

Cirak, F., Deiterding, R., Mauch, S.P., 2007. Large-scale fluid-structure interaction simulation of viscoplastic and fracturing thin shells subjected to shocks and detonations. Computers & Structures 85, 1049–1065.

Cirak, F., Radovitzky, R., 2005. A Lagrangian-Eulerian shell-fluid coupling algorithm based on level sets. Computers & Structures 83, 491–498.

Deiterding, R., 2003. Parallel adaptive simulation of multi-dimensional detonation structures. Ph.D. thesis. Brandenburgische Technische Universität Cottbus.

Deiterding, R., 2005a. Construction and application of an AMR algorithm for distributed memory computers, in: Plewa, T., Linde, T., Weirs, V.G. (Eds.), Adaptive Mesh Refinement - Theory and Applications, Springer. pp. 361–372.

Deiterding, R., 2005b. Detonation structure simulation with AMROC, in: Yang, L.T. (Ed.), High Performance Computing and Communications 2005, Springer. pp. 916–927.

Deiterding, R., 2009. A parallel adaptive method for simulating shock-induced combustion with detailed chemical kinetics in complex domains. Computers & Structures 87, 769–783.

Deiterding, R., 2011. Block-structured adaptive mesh refinement - theory, implementation and application. European Series in Applied and Industrial Mathematics: Proceedings 34, 97–150.

Deiterding, R., Cirak, F., Mauch, S.P., 2009. Efficient fluid-structure interaction simulation of viscoplastic and fracturing thin-shells subjected to underwater shock loading, in: Hartmann, S., Meister, A., Schäfer, M., Turek, S. (Eds.), Int. Workshop on Fluid-Structure Interaction. Theory, Numerics and Applications, Herrsching am Ammersee 2008, kassel university press GmbH. pp. 65–80.

Deiterding, R., Radovitzki, R., Mauch, S., Cirak, F., Hill, D.J., Pantano, C., Cummings, J.C., Meiron, D.I., 2007. Virtual Test Facility: A virtual shock physics facility for simulating the dynamic response of materials. Available at http://www.cacr.caltech.edu/asc.

Deiterding, R., Radovitzky, R., Mauch, S.P., Noels, L., Cummings, J.C., Meiron, D.I., 2006. A virtual test facility for the efficient simulation of solid materials under high energy shock-wave loading. Engineering with Computers 22, 325–347.

Fedkiw, R.P., 2002. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. J. Comput. Phys. 175, 200–224.

Giordano, J., Jourdan, G., Burtschell, Y., Medale, M., Zeitoun, D.E., Houas, L., 2005. Shock wave impacts on deforming panel, an application of fluid-structure interaction. Shock Waves 14, 103–110.

Hallquist, J., Lin, J.I., 2005. A nonlinear explicit three-dimensional finite element code for solid and structural mechanics. Technical Report UCRL-MA-107254. Lawrence Livermore National Laboratory. Source code (U.S. export controlled) available for licensing fee from http://www.osti.gov/estsc.

Karagiozis, K., Kamakoti, R., Cirak, F., Pantano, C., 2011. A computational study of supersonic disk-gap-band parachutes using Large-Eddy Simulation coupled to structural membrane. J. Fluids Structures 2, 175–192.

Krieg, R.D., Key, S.W., 1976. Implementation of a time dependent plasticity theory into structural computer programs, in: Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects, American Society of Mechanical Engineers, New York. pp. 125–137.

Laurence, S.J., Parziale, N.J., Deiterding, R., 2011. Shock-wave surfing. J. Fluid Mech. 676, 369–431.

Löhner, R., Baum, J., Charman, C., Pelessone, D., 2003. Fluid-structure interaction simulations using parallel computers, Springer. pp. 3–23.

Luccioni, B.M., Ambrosini, R.D., Danesi, R.F., 2004. Analysis of building collapse under blast loads. Engineering & Structures 26, 63–71.

Mauch, S.P., 2003. Efficient Algorithms for Solving Static Hamilton-Jacobi Equations. Ph.D. thesis. California Institute of Technology.

Mittal, R., Iaccarino, G., 2005. Immersed boundary methods. Annu. Rev. Fluid Mech. 37, 239–261.

Mlakar Sr., P.F., Corley, W.G., Sozen, M.A., Thornton, C.H., 1998. The Oklahoma city bombing: Analysis of blast damage to the Murrah building. J. Perform. Constr. Facil. 12, 113–119.

Moore, D.F., Jerusalem, A., Nyein, M., Noels, L., Jaffee, M.S., Radovitzky, R.A., 2009. Computational biology - modeling of primary blast effects on the central nervous system. NeuroImage 47, Supplement 2, T10 – T20.

Ngo, T., Mendis, P., Gupta, A., Ramsay, J., 2007. Blast loading and blast effects on structures - an overview. Electronic Journal of Structual Engineering .

Parashar, M., Browne, J.C., 1996. On partitioning dynamic adaptive grid hierarchies, in: Proc. of the 29th Annual Hawaii Int. Conf. on System Sciences.

Perotti, L.E., Deiterding, R., Inaba, K., Shepherd, J.E., Ortiz, M., 2013. Elastic response of water-filled fiber composite tubes under shock wave loading. Int. J. Solids and Structures 50, 473–486.

Puso, M.A., Sanders, J., Settgast, R., Liu, B., 2012. An embedded mesh method in a multiple material ALE. Computer Methods in Applied Mechanics and Engineering 245246, 273 – 289.

Specht, U., 2000. Numerische Simulation mechanischer Wellen an Fluid-Festkörper-Mediengrenzen. Number 398 in VDI Reihe 7, VDU Verlag, Düsseldorf.

Toro, E.F., 1999. Riemann solvers and numerical methods for fluid dynamics. Springer-Verlag, Berlin, Heidelberg. 2nd edition.

Veccio, F.J., Collins, M.P., 1986. The modified compression-field theory for reinforced concrete elements subjected to shear. American Concrete Institute 83, 219–231.

Wang, K., Grétarsson, J., Main, A., Farhat, C., 2012. Computational algorithms for tracking dynamic fluid-structure interfaces in embedded boundary methods. International Journal for Numerical Methods in Fluids 70, 515–535.

Zhao, H., Freund, J.B., Moser, R.D., 2008. A fixed-mesh method for incompressible flow-structure systems with finite solid deformations. Journal of Computational Physics 227, 3114 – 3140.

Ziegler, J.L., Deiterding, R., Shepherd, J.E., Pullin, D.I., 2011. An adaptive high-order hybrid scheme for compressive, viscous flows with detailed chemistry. J. Comput. Phys. 230, 7598–7630.

Table 1: Material parameters for the two cases using DYNA3D model #1 (elastic material).

	Panel	Bridge
	steel	concrete
ρ, density [kg/m^3]	7600	2010
E, Young's modulus [GPa]	220	21.72
ν, Poisson's ratio	0.33	0.2

Table 2: Material parameter for cases using DYNA3D's elastic-plastic with failure material model #13.

	Column		Building	
	concrete	rebar	columns	walls
ρ, density [kg/m^3]	2010	7800	2010	2010
σ_0, yield stress [MPa]	25	1800	50	25
E_T, tangent modulus [GPa]	5.6	134.4	11.2	11.2
β, hardening parameter	0.09	0.12	1.0	1.0
K, bulk modulus [GPa]	21.72	170	21.72	6.22
G, shear modulus [GPa]	4.67	80	4.67	4.67
$\bar{\epsilon}^p$, plastic strain limit	0.02	0.2	0.02	0.01
p_f, pressure limit [MPa]	-15	-1970	-30	-15

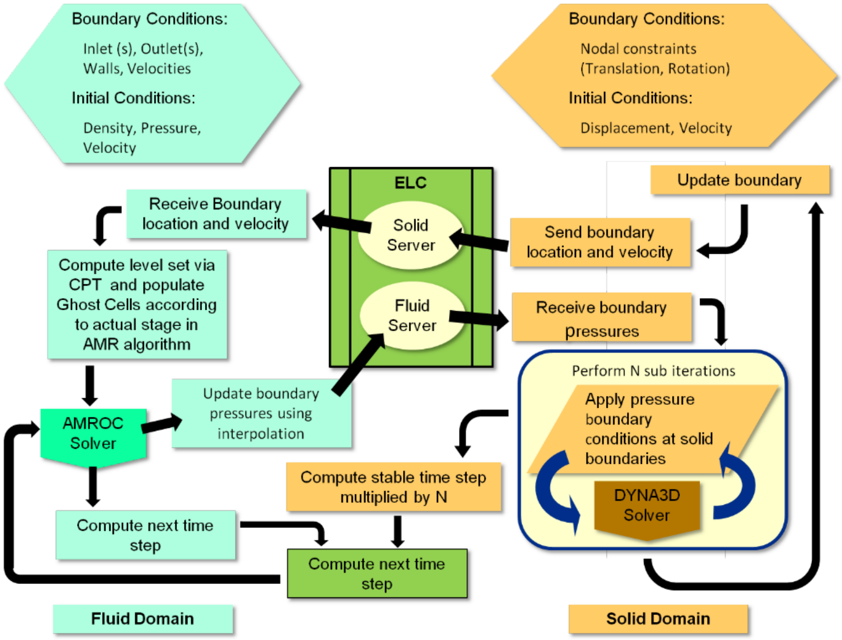
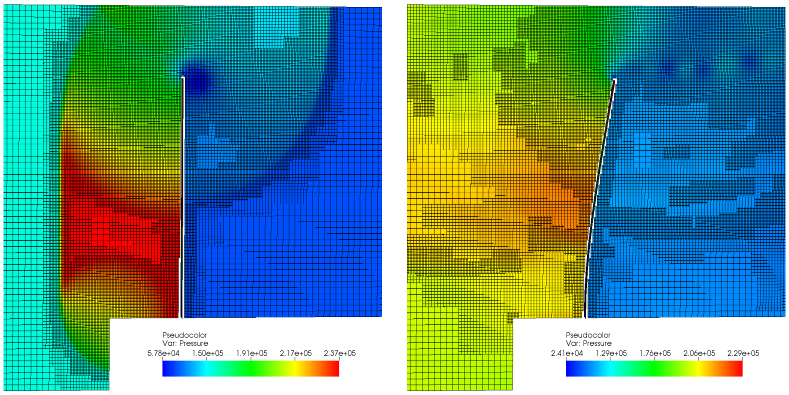


Figure 1: Interaction and data flow between the essential elements of the coupled AMROC-DYNA3D solver.



(a) $t = 0.122\,\mathrm{ms}$

(b) $t = 0.945\,\mathrm{ms}$

Figure 2: Dynamic mesh adaptation in fluid and deflection of the panel.

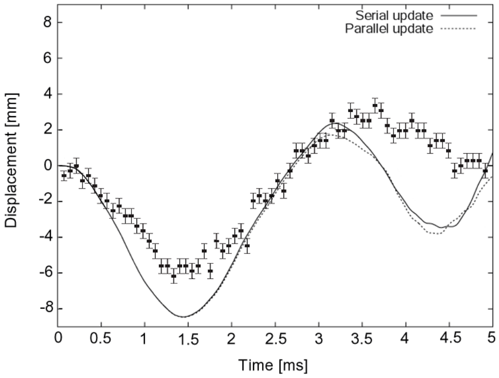


Figure 3: Panel tip displacement vs. time in the two FSI simulations and in the experiment by Giordano et al. (2005).

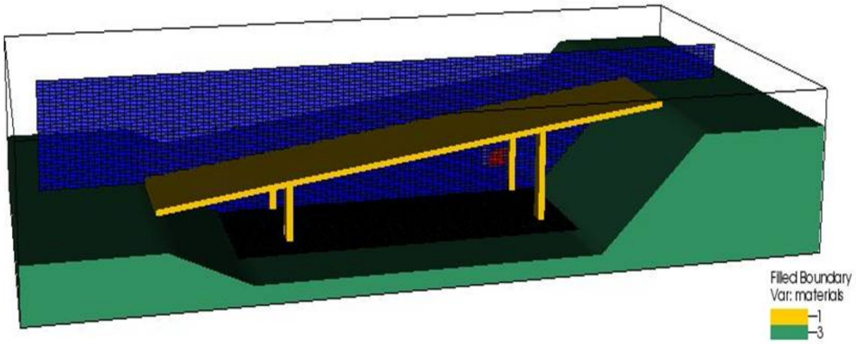
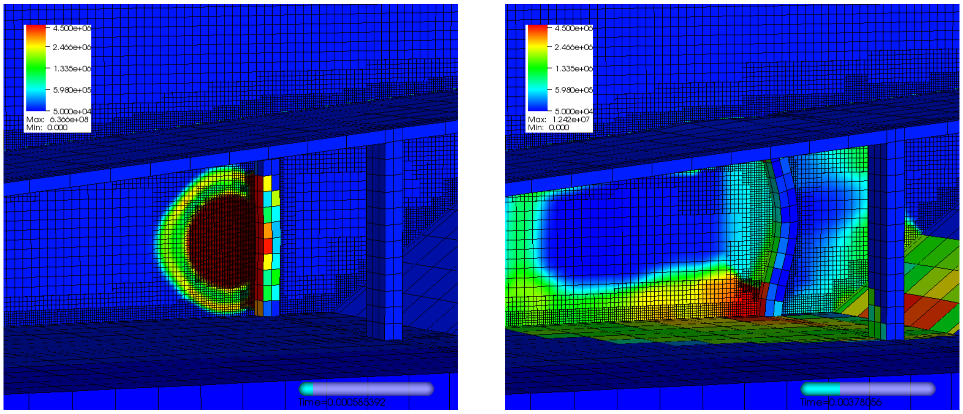


Figure 4: Air pressure on slice plane above rigid ground (green) and around deforming bridge (tan) shown for initial time step.



(a) $t = 0.59\,\mathrm{ms}$ (b) $t = 3.78\,\mathrm{ms}$

Figure 5: Deforming structural model embedded in the adaptive Cartesian mesh.

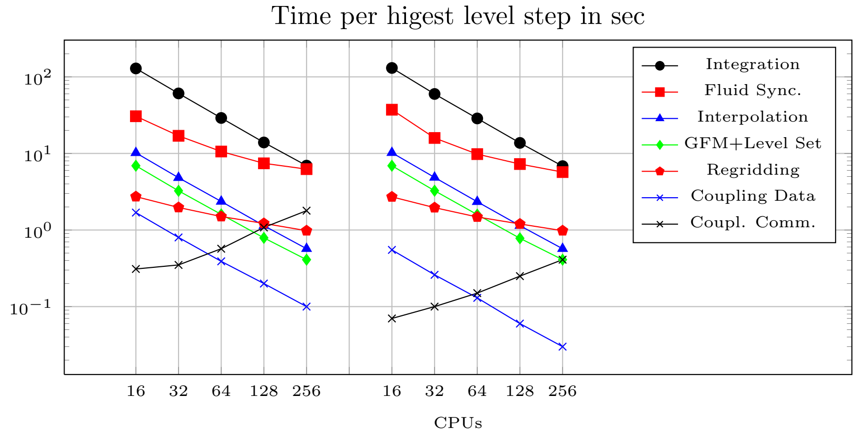


Figure 6: Parallel scalability test for the coupled AMROC-DYNA3D code. Left: *update_type=sequential*, $l_{\mathrm{fsi}} = 2$, no sub-iterations; right: *update_type=parallel*, $l_{\mathrm{fsi}} = 1$, four sub-iterations in solid solver. Note the difference in the location of the curves *Coupling Data* and *Coupl. Comm.* For instance, overall runtime of the 256 core simulation is reduced by $\sim 12\%$ for the right graphic.

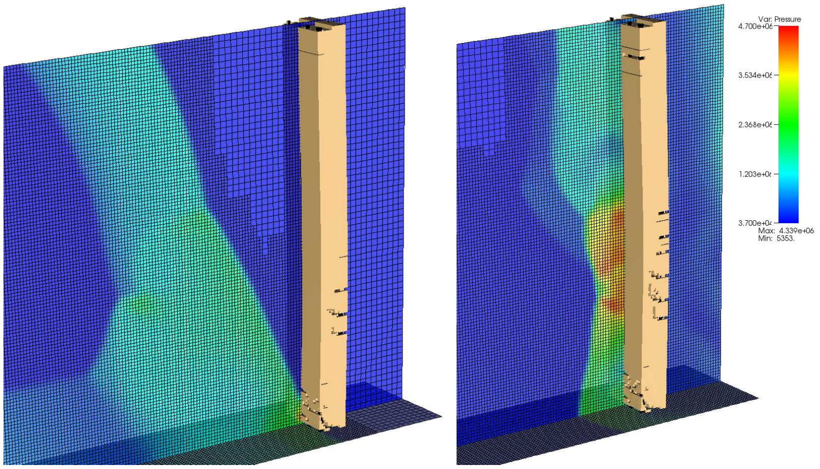
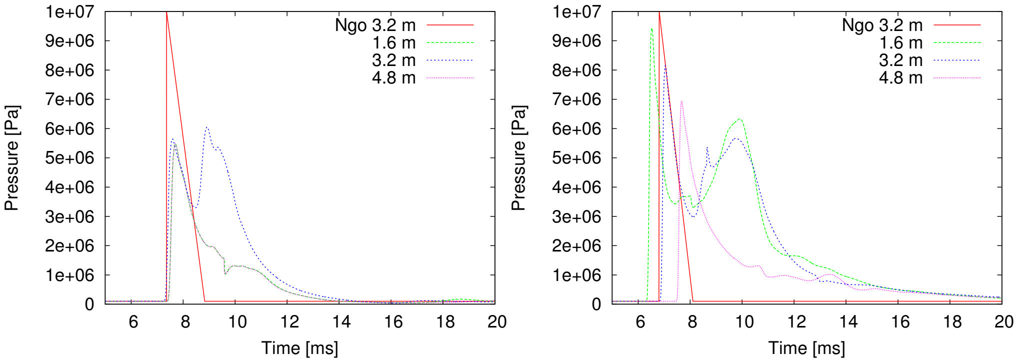


Figure 7: Air pressure and mesh adaption shown on slice planes and ground around fracturing column at $t = 6.05\,\mathrm{ms}$ (left) and $t = 11.5\,\mathrm{ms}$ (right) simulated time.



(a) All fluid boundaries out flow.

(b) Ground as reflective boundary.

Figure 8: Pressure history on column centerline at 1.6 m, 3.2 m, 4.8 m above the column's base.

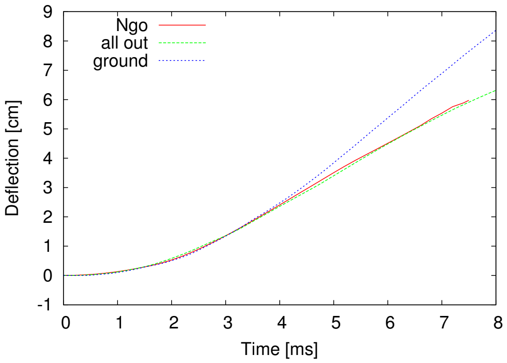


Figure 9: Comparison of lateral deflection history with simulation results by Ngo et al. (2007). Record begins with column mid-point deflection.

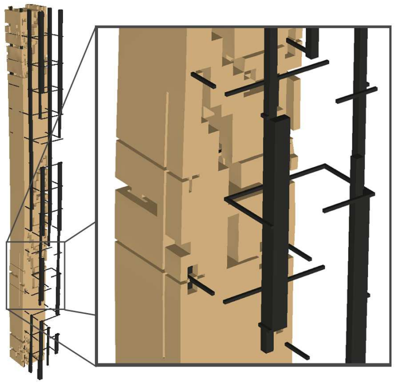
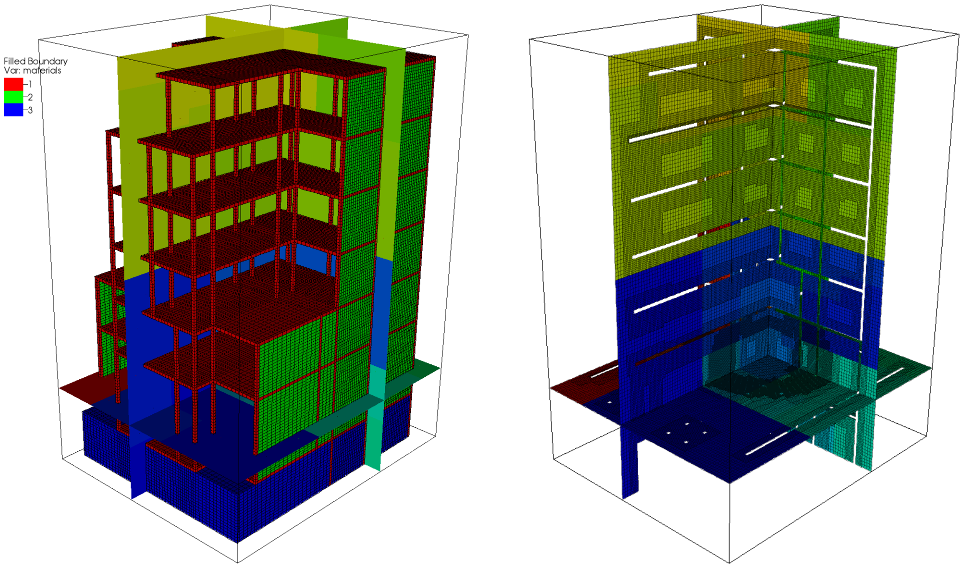


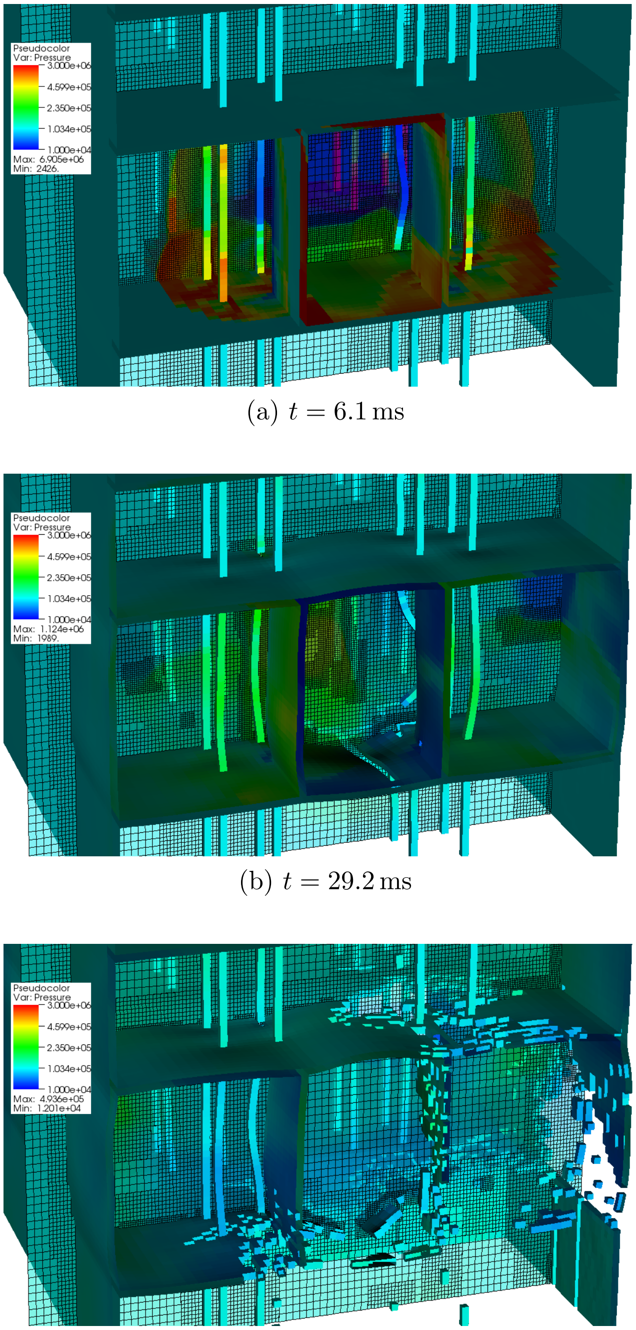
Figure 10: Column with detail view at 11.5 ms (4.55 ms after mid-point deflection begins). Half of concrete is shown to reveal internal failures.



(a)

(b)

Figure 11: Location of three materials used in the building solid model (a). Domain decomposition (shown by color) to 31 processors and adaptive mesh of the fluid model at $t = 5.99\,\mathrm{ms}$ (b).



(a) $t = 6.1\,\mathrm{ms}$

(b) $t = 29.2\,\mathrm{ms}$

(c) $t = 48.7\,\mathrm{ms}$

Figure 12: Pressure in fluid and on solid boundary (indicated by color).