An Adaptive Cartesian Detonation Solver for Fluid-Structure Interaction Simulation on Distributed Memory Computers

R. Deiterding[a]

[a]California Institute of Technology, Mail Code 158-79
1200 East California Blvd., Pasadena, CA 91125, USA

Time-accurate fluid-structure interaction simulations of strong shock and detonation waves impinging on deforming solid structures benefit significantly from the application of dynamic mesh adaptation in the fluid. A patch-based parallel fluid solver with adaptive mesh refinement in space and time tailored for this problem class is presented; special attention is given to the robustness of the finite volume scheme with embedded boundary capability and a scalable implementation of the hierarchical mesh refinement method.

1. Introduction

The Center for Simulating the Dynamic Response of Materials at the California Institute of Technology has constructed a virtual test facility (VTF) for studying the three-dimensional dynamic response of solid materials subject to strong shock and detonation waves propagating in fluids. While the fluid flow is simulated with a high-resolution Cartesian finite volume upwind method that considers the solid as an embedded moving body represented implicitly with a level set function, Lagrangian finite element schemes are employed to describe the time-accurate material response subject to the current hydrostatic pressure loading. A loosely coupling temporal splitting method is applied to update the boundary's positions and velocities between time steps. The Cartesian finite volume scheme is incorporated into a parallel structured dynamic mesh adaptation algorithm that allows very fine local resolutions to capture the near-body fluid-structure interaction (FSI) and incoming waves in the fluid at minimal computational costs.

In this paper, we describe the dynamically adaptive solver for compressible flows, including shock and detonation waves with one-step reaction model, that enables highly efficient FSI simulations in the VTF on distributed memory machines. After introducing the governing fluid equations, we explain our specific implementation of the ghost fluid approach [11]. In Sec. 4, we outline the structured adaptive mesh refinement (SAMR) algorithm of Berger and Collela [3], and in particular the locality-preserving rigorous domain decomposition paradigma under which the method has been parallelized [7]. Section 5 details the extension of the SAMR implementation to loosely coupled FSI problems. The final section, Sec. 6, gives a numerical example in which a detonation wave propagating through a high explosive interacts with a surrounding solid cylinder. The enormous savings in compute time from mesh adaptation and parallelization demonstrate the efficiency of the approach.

2. Governing Equations

In order to model detonation waves we utilize the single-phase model proposed by Fickett and Davis [13], which has also been used by Clarke et al. [4] to evaluate numerical methods for detonation simulation. We assume a single chemical reaction $A \longrightarrow B$ that is modelled by a progress variable λ corresponding to the mass fraction ratio between the density of the product B and the total density ρ, i.e. $\lambda = \rho_B/\rho$. The governing equations of the model read

$$\begin{array}{rcl} \partial_t \rho + & \nabla \cdot (\rho \mathbf{u}) & = 0 \,, \\ \partial_t (\rho E) + & \nabla \cdot ((\rho E + p)\mathbf{u}) & = 0 \,, \end{array} \qquad \begin{array}{rcl} \partial_t (\rho \mathbf{u}) + & \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p & = 0 \,, \\ \partial_t \lambda + & \mathbf{u} \cdot \nabla \lambda & = \psi \,. \end{array} \tag{1}$$

Herein, $\mathbf{u}$ is the velocity vector and E the specific total energy. The hydrostatic pressure p is given by $p = (\gamma - 1)(\rho E - \frac{1}{2}\rho \mathbf{u}^T \mathbf{u} + \rho \lambda q)$ with γ denoting the ratio of specific heats and q the heat release due to the chemical reaction per unit mass. System (1) together with above pressure equation is a valid model both for detonations in combustible gases and high energetic solid materials. As our focus in this paper is on the latter, we use the simple rate function $\psi = \frac{2}{T_R}(1-\lambda)^{1/2}$ proposed by Fickett for detonations in solids in the following. Herein, T_R denotes a typical time associated with the reaction in which the depletion from A to B is complete.

3. Cartesian Finite Volume Scheme with Embedded Boundaries

Following Clarke et al. [4], we apply the method of fractional steps to decouple the chemical reaction and hydrodynamic transport numerically. The *homogeneous* system of (1) and the scalar ordinary differential equation $\partial_t \lambda = \psi(\lambda)$ are solved successively with the data of the preceding step as initial conditions. As the homogeneous system (1) is a hyperbolic conservation law that admits discontinuous solutions, cf. [4], we use a time-explicit finite volume discretization that achieves a proper upwinding in all characteristic fields. The scheme is based on a straightforward generalization of the Roe scheme for the purely hydrodynamic Euler equations and is extended to a multi-dimensional Cartesian scheme via the method of fractional steps, cf. [22]. To circumvent the intrinsic problem of unphysical total densities and internal energies near vacuum due to the Roe linearization, cf. [10], the scheme has the possibility to switch to the simple, but extremely robust Harten-Lax-Van Leer (HLL) Riemann solver. The occurrence of the disastrous carbuncle phenomena [20], a multi-dimensional numerical crossflow instability that affects every simulation of strong grid-aligned shocks or detonation waves, is prevented by introducing a small amount of additional numerical viscosity in a multi-dimensional way [21]. This hybrid Riemann solver is supplemented with the MUSCL-Hancock variable extrapolation technique of Van Leer [22] to achieve second-order accuracy in regions where the solution is smooth.

Geometrically complex moving boundaries are considered by utilizing some of the finite volume cells as ghost cells to enforce immersed moving wall boundary conditions [11]. The boundary geometry is mapped onto the Cartesian mesh by employing a scalar level set function ϕ that stores the signed distance to the boundary surface and allows the efficient evaluation of the boundary outer normal in every mesh point as $\vec{n} = -\nabla\phi/|\nabla\phi|$.

In coupled Eulerian-Lagrangian simulations, ϕ is updated after every boundary synchronization step by calling the closest-point-transform algorithm developed by Mauch [18]. A cell is considered to be a valid fluid cell within the interior, if the distance ϕ in the cell *midpoint* is positive and is treated as exterior otherwise.

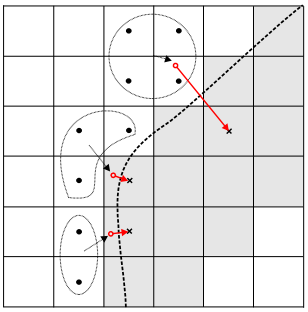

Figure 1: Construction of mirrored values to be used in interior ghost cells (gray).

For system (1), the boundary condition at a rigid wall moving with velocity $\vec{w}$ is $\vec{u}\cdot\vec{n} = \vec{w}\cdot\vec{n}$. Enforcing the latter with ghost cells, in which the discrete values are located in the cell centers, requires the mirroring of the primitive values ρ, $\vec{u}$, p, λ across the embedded boundary. The normal velocity in the ghost cells is set to $(2\vec{w}\cdot\vec{n} - \vec{u}\cdot\vec{n})\vec{n}$, while the mirrored tangential velocity remains unmodified. Mirrored values are constructed by calculating spatially interpolated values in the point $\vec{\tilde{x}} = \vec{x} + 2\phi\vec{n}$ from neighboring interior cells. For instance, in two space dimensions, we employ a bilinear interpolation between (usually) four adjacent cell values, but directly near the boundary the number of interpolants needs to be decreased, cf. Fig. 1. It has to be emphasized that for hyperbolic problems with discontinuities like detonation waves, special care must be taken to ensure the monotonicity preservation of the numerical solution. Figure 1 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 complex boundary (dotted).

After the application of the numerical scheme, cells that have been used to impose internal boundary conditions are set to the entire state vector of the nearest cell in the interior. This operation ensures proper values in case such a cell becomes a regular interior cell in the next step due to boundary movement. The consideration of $\vec{w}$ in the interior ghost cells ensures that the embedded boundary propagates at most one cell further in every time step.

4. Structured Adaptive Mesh Refinement

Numerical simulations of detonation waves require computational meshes that are able to represent the strong local flow changes due to the reaction correctly. The shock of a self-sustained detonation is very sensitive to changes in the energy release from the reaction behind and the inability to resolve all reaction details usually causes a considerable error in approximating the correct speed of propagation. In order to supply the required temporal and spatial resolution efficiently, we employ the structured adaptive mesh refinement (SAMR) method of Berger and Colella [3]. Instead of replacing single cells by finer ones, as it is done in cell-oriented refinement techniques, the Berger-Collela SAMR method follows a patch-oriented approach. Cells being flagged by various error indicators (shaded in Fig. 2) are clustered with a special algorithm [2] into non-overlapping rectangular grids. Refinement grids are derived recursively from coarser ones and a hierarchy of successively embedded levels is thereby constructed (cf. Fig. 2). All mesh widths on level 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.

The numerical scheme is applied on level l by calling a single-grid routine in a loop over all subgrids. The subgrids get computationally decoupled by employing additional ghost cells around each computational grid. Three different types of ghost cells have to be considered: Cells outside of the root domain are used to implement physical boundary conditions; ghost cells overlaid by a grid on level l have a unique interior cell analogue and are set by copying the data value from the grid, where the interior cell is contained (synchronization). On the root level no further boundary conditions need to be considered, but for $l > 0$ internal boundaries can also occur. They are set by a conservative time-space interpolation from two previously calculated time steps of level $l-1$.

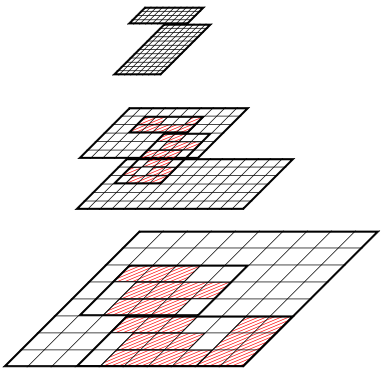
Figure 2: SAMR hierarchy.

The regularity of the SAMR data allows high performance on vector and super-scalar processors that allow cache optimizations. Small data arrays are effectively avoided by leaving coarse level data structures untouched when higher level grids are created. Values of cells covered by finer subgrids are overwritten by averaged fine grid values subsequently. This operation leads to a modification of the numerical stencil on the coarse mesh and requires a special flux correction in cells abutting a fine grid. The correction replaces the coarse grid flux along the fine grid boundary by a *sum* of fine fluxes and ensures the discrete conservation property of the hierarchical method (at least for purely Cartesian problems without embedded boundaries). See [3] or [6] for details.

In our SAMR solver framework AMROC (Adaptive Mesh Refinement in Object-oriented C++) [8], 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 ghost cell synchronization, redistribution of the data hierarchy and the application of the previously mentioned flux correction terms. Interpolation and averaging, but in particular the calculation of the flux corrections remain strictly local [6]. Currently, we employ a generalization of Hilbert's space-filling curve [19] to derive load-balanced root level distributions at runtime. The entire SAMR hierarchy is considered by projecting the accumulated work from higher levels onto the root level cells. Figure 3 shows a representative scalability test for a three-dimensional spherical shock wave problem for the computationally inexpensive Euler equations for a single polytropic gas without chemical reaction. The test was run on a Linux Beowulf cluster of Pentium-4-2.4 GHz dual processor nodes with Quadrics Interconnect. The base grid had 32^3 cells and two additional levels with refinement factors 2 and 4. The adaptive calculation used approx. 7.0 M cells in each time step instead of 16.8 M cells in the uniform case. Displayed are the average costs for each root level time step. Although we utilize a single-grid update routine in Fortran 77 in a C++ framework with full compiler optimization, the fraction of the time spent in this Fortran routine are 90.5 % on four and still 74.9 % on 16 CPUs. Hence, Fig. 3 shows a satisfying scale-up for at least up to 64 CPUs.

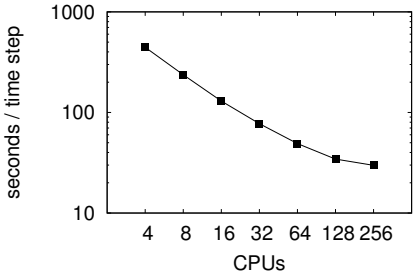

Figure 3: SAMR scalability test.

```
advance_level( l )
   repeat r_l times
      if time to regrid
         regrid( l )
      level_set_generation( φ^l, I )
      update_fluid_level( Q^l, φ^l, w|_I, Δt_l )
      if level l+1 exists
         advance_level(l+1)
         Correct Q^l(t+Δt_l) with Q^{l+1}(t+Δt_l)
      if l = l_c
         send_interface_data( p(t+Δt_l)|_I )
         receive_interface_data( I, w|_I )
      t := t + Δt_l
return
```

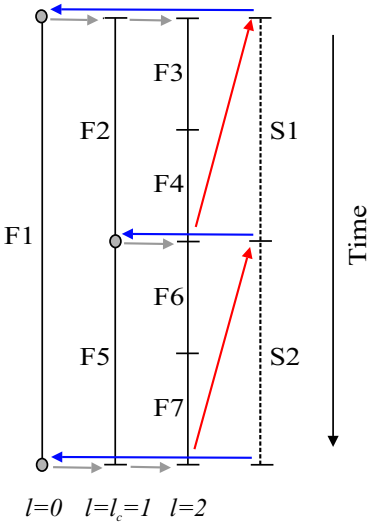

Figure 4. Left: SAMR algorithm for fluid-structure coupling. Right: data exchange between advance_level() and a conventional solid solver throughout one level 0 time step.

5. 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-problem. 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 Eulerian fluid solver with embedded boundary capability (cf. Sec. 3) receives the velocities and the discrete geometry of the solid surface, while only the hydrostatic pressure is communicated back to the Lagrangian solid solver as a force acting on the solid's exterior [1,16]. 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 been implemented to support the boundary data exchange between (dedicated) fluid and solid processes on distributed memory machines, see [9] for details on this.

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 will often not be the highest level of refinement, cf. Sec. 6. We formulate the corresponding extension of the recursive SAMR algorithm of Berger and Collela [3] in the routine advance_level() outlined in pseudo-code on the left side of Fig. 4. 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 $\vec{w}|_{\mathcal{I}}$, the discrete vector of state in the fluid $\vec{Q}$ is updated for the entire level with the scheme detailed in Sec. 3. 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 $\mathcal{I}$ to be 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 dynamically before the current surface mesh, i.e. the level set information derived from it, is actually used to again advance level l_c.

The data exchange between the solid and advance_level(), is visualized in the right graphic of Fig. 4 for an exemplary SAMR hierarchy with two additional levels with $r_{1,2} = 2$. Figure 4 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 $\vec{w}|_{\mathcal{I}}$ after each solid update. The modification of refinement meshes is indicated in Fig. 4 by the gray arrows; the initiating base level that remains fixed throughout the regridding operation is indicated by the gray circles.

6. HMX Detonation in a Tantalum Cylinder

As computational example we present the three-dimensional dynamic interaction of a detonation wave in the high explosive HMX ($C_4H_8N_8O_8$) with the walls (thickness 0.01 m) and the closed end of a cylinder made of Tantalum. The cylinder has the length 0.10 m and an outer radius of 0.0185 m. An inner combustion chamber of depth 0.055 m opens at its left end. A non-adaptive tetrahedral structure mechanics finite element solver with special artificial viscosity formulation for capturing dilatation and shear waves [15] is employed for the solid update. The Tantalum is assumed to obey J2-flow theory of plasticity and Vinet's thermal equation equation of state with parameters derived from first-principle calculations [14]. The shown computation used a solid mesh of 56,080 elements.

For the fluid initial conditions, we assume a fully developed one-dimensional steady Chapman-Jouguet detonation with its front initially located at $x = 0.01\,\mathrm{m}$ that we prescribe according to the theory of Zeldovich, Neumann, and Döring (ZND) (see [13] or [7] for detailed derivations). The detonation is propagating into the positive direction, which allows the prescription of constant inflow boundary conditions at the open left end (cf. Fig. 5). No deformations are allowed in the entire solid for $x < 0.01\,\mathrm{m}$ to model a fully rigid material downstream of the initial wave. Further, no deformations are possible on the outer hull of the cylinder for $0.01\,\mathrm{m} \leq \mathrm{x} \leq 0.03\,\mathrm{m}$.

Unreacted HMX has a density of $\rho_0 = 1900\,\mathrm{kg/m^3}$ and gets depleted by a Chapman-Jouguet detonation with propagation speed $\approx 9100\,\mathrm{m/s}$ resulting in an energy release of $q \approx 5176\,\mathrm{kJ/kg}$ [17]. The hydrodynamic flow can be described with reasonable accuracy with a constant adiabatic exponent of $\gamma = 3$ [17]. We assume atmospheric pressure $p_0 = 100\,\mathrm{kPa}$ in the unreacted material and set the unknown rate factor to $T_R = 1\,\mu\mathrm{s}$. Fig. 5 displays the steadily propagating

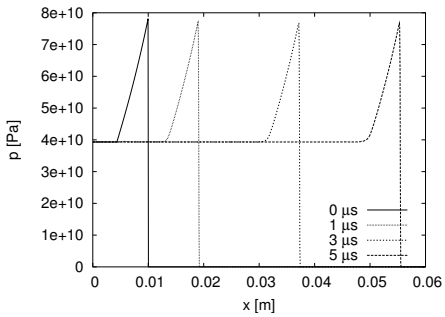

Figure 5: One-dimensional simulation of the detonation in HMX.

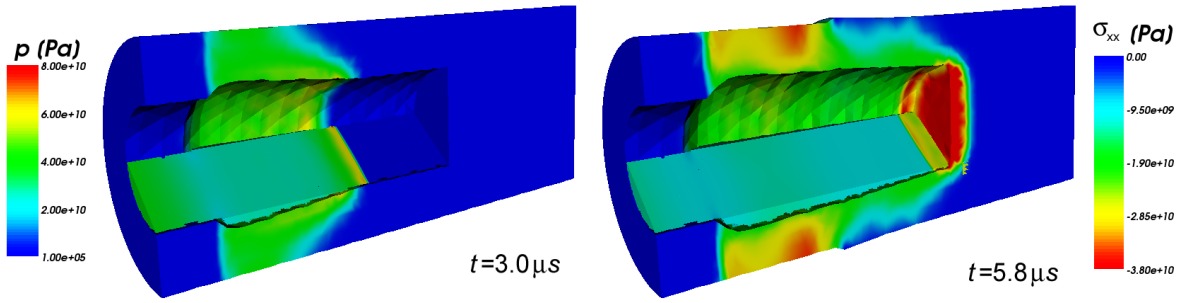

Figure 6. Left: compression of the wall material next to the combustion chamber due to the detonation passage. Right: outward movement of the unconstrained walls and the strong compression in axial direction due to the impact event.

pressure distribution of the ZND wave in a one-dimensional computation on a uniform mesh with 960 finite volume cells. At considerably coarser resolutions, the head of the detonation wave is not approximated to sufficient accuracy leading to an incorrect speed of propagation and a significant reduction of the maximal pressure value. Hence, this FSI problem benefits considerably from the application of dynamic mesh adaptation in the fluid.

Two snapshots of the simulation displaying a cut through the hydrodynamic pressure distribution and the normal stress in the axial direction are shown in Fig. 6. The left graphic shows the initiation of stress waves at the head of the detonation (the slight delay in the solid is due to its coarser mesh resolution). The right graphic at later time exhibits the influence of the boundary conditions: While the material gets strongly compressed initially, no normal stresses arise at the outer surface in the unconstrained section with $x \geq 0.03\,\mathrm{m}$. At $t = 5.8\,\mu\mathrm{s}$, the HMX is fully depleted and the impact of the detonation wave at the closed end has caused a very strong compression wave in the solid in the axial direction. The reflected hydrodynamic shock wave is visible.

The fluid sub-problem has been run on a Cartesian domain of $0.03\,\mathrm{m} \times 0.03\,\mathrm{m} \times 0.06\,\mathrm{m}$ and was discretized with $60 \times 60 \times 120$ cells at the SAMR root level. While the solid boundary is fully refined at the coupling level $l_c = 1$ with $r_1 = 2$, level 2 is only used to capture the head of detonation wave accurately ($r_2 = 4$). The SAMR mesh increases from initially approx. 706 k cells on level 1 and 6.5 M on level 2 to about 930 k and 10.0 M cells at later times. The number of grids on both levels varies between 400 and 1000. Compared with a uniform fluid mesh of $480 \times 480 \times 960 \simeq 221$ M cells, the enormous saving from mesh adaptation is apparent. Figure 7 displays the adapted fluid mesh in the mid plane for $t = 3.0\,\mu\mathrm{s}$ by overlaying a schlieren plot of the fluid density onto regions covered by level 1 (blue) and 2 (red). The simulation ran on 4 nodes of a Pentium-4-2.4 GHz dual processor system connected with Quadrics interconnect for about 63 h real time. Six processes were dedicated to the adaptive fluid simulation, while two were used for the significantly smaller solid problem.

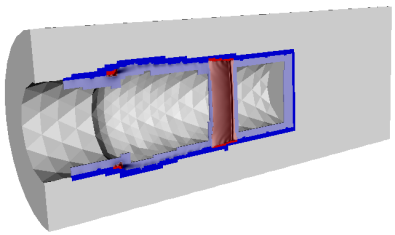

Figure 7: Adaptation at $t = 3.0\,\mu\mathrm{s}$.

7. Conclusions

An efficient Cartesian finite volume method for the simulation of detonation waves in FSI problems has been presented. The method is patch-based and considers embedded complex boundaries with the ghost-fluid approach [11]. Distributed memory parallelization is provided by a parallel variant of the SAMR method of Berger and Collela [3] that follows a rigorous domain-decomposition approach [7]. An algorithmic extension of the recursive SAMR method to loosely coupled FSI simulation with time-explicit solvers has been described. The approach allows the accurate capturing of near-body fluid-structure interaction, while resolution in space *and* time can be reduced effectively in the fluid far field. As computational example, a detonation wave in a solid high energetic material impinging on a Tantalum cylinder has been discussed. The example demonstrates the enormous savings in the computational costs that can be obtained through structured dynamic mesh adaptation in the fluid for this problem class: While the parallel calculation required only 504 h CPU (63 h real time), a simulation with an equivalent fluid unigrid mesh can be expected to be in the range of 10^5 h CPU.

REFERENCES

1. M. Aivazis, W.A. Goddard, D.I. Meiron et al., Comput. Science & Eng. 2(2) 2000 42.
2. J. Bell, M. Berger, J. Saltzman, M. Welcome, SIAM J. Sci. Comp. 15(1) (1994) 127.
3. M. Berger and P. Colella, J. Comput. Phys. 82 (1988) 64.
4. J. F. Clarke, S. Karni, J. J. Quirk et al., J. Comput. Phys. 106 (1993) 215.
5. J. C. Cummings, M. Aivazis, R. Samtaney et al., J. Supercomput. 23 (2002) 39.
6. R. Deiterding, Parallel adaptive simulation of multi-dimensional detonation structures, PhD thesis, Brandenburgische Technische Universität Cottbus, 2003.
7. R. Deiterding, in Notes Comput. Science & Eng. 41, Springer, New York, (2005) 361.
8. R. Deiterding, AMROC, available at http://amroc.sourceforge.net (2005).
9. R. Deiterding, R. Radovitzky, S. P. Mauch et al., Engineering with Computers, Springer, (2005) submitted.
10. B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjögreen, J. Comput. Phys. 92 (1991) 273.
11. R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, J. Comput. Phys. 152 (1999) 457.
12. R. P. Fedkiw, J. Comput. Phys. 175 (2002) 200.
13. W. Fickett, W. .C. Davis, Detonation, Univ. Cal. Press, Berkeley, 1979.
14. D. E. Johnson, E. Cohen, in Proc. IEEE Conf. Robotics & Automation (1998) 3678.
15. A. Lew, R. Radovitzky, and M. Ortiz, J. Comput-Aided Mater. Des. 8 (2002) 213.
16. R. Löhner, J.D.Baum et al. in Notes Comput. Science 2565, Springer,Berlin (2003) 3.
17. C. L. Mader, Numerical modeling of detonations, Univ. Cal. Press, Berkeley, 1979.
18. S. P. Mauch, Efficient Algorithms for Solving Static Hamilton-Jacobi Equations, PhD thesis, California Institute of Technology, 2003.
19. M. Parashar and J. C. Browne, in Proc. 29th Hawaii Int. Conf. System Sciences, 1996.
20. J. J. Quirk. Int. J. Numer. Meth. Fluids 18 (1994) 555.
21. R. Sanders, E. Morano, M.-C. Druguett, J. Comput. Phys. 145 (1998) 511.
22. E. F. Toro, Riemann solvers and numerical methods for fluid dynamics, Springer, Berlin, Heidelberg, 1999.