A Virtual Test Facility for Simulating Detonation-Induced Fracture of Thin Flexible Shells

Ralf Deiterding, Fehmi Cirak, Sean P. Mauch, and Daniel I. Meiron

California Institute of Technology, MC 158-79, Pasadena CA 91125, USA
{ralf, cirak, sean, dim}@cacr.caltech.edu
http://www.cacr.caltech.edu/asc

Abstract. The fluid-structure interaction simulation of detonation- and shock-wave-loaded fracturing thin-walled structures requires numerical methods that can cope with large deformations as well as topology changes. We present a robust level-set-based approach that integrates a Lagrangian thin shell finite element solver with fracture and fragmentation capabilities with an Eulerian Cartesian detonation solver with optional dynamic mesh adaptation. As an application example, the rupture of a thin aluminum tube due to the passage of an ethylene-oxygen detonation wave is presented.

1 Introduction

The Center for Simulation of Dynamic Response of Materials at the California Institute of Technology has developed a virtual test facility (VTF) for studying the three-dimensional dynamic response of solid materials subjected to strong shock and detonation waves propagating in fluids [7]. The fluid flow is simulated with a Cartesian finite volume upwind method in Eulerian coordinates that considers the solid as a moving embedded body by utilizing a ghost-fluid approach (see Sec. 2). In the present paper, the fracturing thin-walled solid is modeled as a Kirchhoff-Love type thin-shell in Lagrangian coordinates. The finite element discretization of the underlying energy functional is accomplished with subdivision finite elements (see Sec. 3).

The evolving shell geometry is implicitly represented on the Cartesian fluid mesh by a scalar level set function, which stores the unsigned distance to the solid surface. As the solid deforms, the level set function is updated on-the-fly by a highly efficient auxiliary algorithm based on geometric characteristic reconstruction and scan conversion (see Sec. 4). In the VTF, we apply a loosely coupled temporal splitting method for updating the boundary's position and velocity between time steps. The fluid-structure coupling approach and its implementation on distributed memory computing platforms are briefly described in Sec. 5. As a large-scale three-dimensional computational example, we present in Sec. 6 the simulation of the rupture of a thin aluminum tube due to an internal detonation wave arising from the combustion of an ethylene-oxygen mixture. Simulations of

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 122–130, 2006.
© Springer-Verlag Berlin Heidelberg 2006

this type will be compared directly to available experimental results [1] and will serve as a validation case for the VTF.

2 Eulerian Detonation Solver

The governing equations of detonation wave propagation in gases are the inviscid Euler equations [9]. Throughout this paper, we consider only the simplified case of a single exothermic chemical reaction $A \longrightarrow B$ with a progress variable Y corresponding to the mass fraction ratio between the partial density of the reactant A and the total density ρ, i.e. $Y = \rho_A/\rho$. The governing equations of the hydrodynamic model are

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

Herein, $\boldsymbol{u}$ is the velocity vector and E the specific total energy. The hydrostatic pressure p is given by $p = (\gamma - 1)(\rho E - \frac{1}{2}\rho \boldsymbol{u}^T \boldsymbol{u} - \rho Y q)$ with γ denoting the ratio of specific heats and q the heat release due to the chemical reaction per unit mass. A one-step reaction would typically be modeled with an Arrhenius law such as $\psi = -kY\rho \exp(-E_A \rho/p)$ [9], but in the specific case considered here, we utilize the constant volume burn model suggested by Mader [10]. This model neglects the detailed chemical depletion, and therefore the internal detonation structure, but ensures the right propagation speed and the correct state in chemical equilibrium at all grid resolutions. The model is intended to be applied together with the fractional step method that numerically decouples chemical reaction and hydrodynamic transport. First, the *homogeneous* system (1) is advanced at a full time step, then the reactant density ρ_A, pressure p, and total energy E are modified locally in each cell; the total density ρ and the velocity vector $\boldsymbol{u}$ remain unaltered. The algorithm for the detonation model reads:

$V := \rho^{-1}$, $V_0 := \rho_0^{-1}$, $V_{CJ} := \rho_{CJ}$, $Y' := 1 - (V - V_0)/(V_{CJ} - V_0)$
if $0 \leq Y' \leq 1$ and $Y > 10^{-8}$
  if $Y < Y'$ and $Y' < 0.9$ then $Y' := 0$
  if $Y' < 0.99$ then $p' := (1 - Y')p_{CJ}$ else $p' := p$
  $\rho_A := Y'\rho$, $E := p'/(\rho(\gamma - 1)) + Y'q_0 + \frac{1}{2}\boldsymbol{u}^T\boldsymbol{u}$

In the latter, the index 0 indicates the unreacted state (assumed to be constant), while CJ refers to the equilibrium values that can be calculated in advance following Chapman-Jouguet theory [9, 6] for a given detonation velocity.

For the shock-capturing finite volume upwind scheme, we utilize a straightforward extension of the flux-vector splitting method by Van Leer (cf. [6]). Second-order accuracy in smooth solution regions is achieved with the MUSCL-Hancock variable extrapolation technique [12]. Geometrically complex moving boundaries are incorporated in the upwind scheme by using some of the finite volume cells as ghost cells for enforcing immersed moving wall boundary conditions [8]. The boundary geometry is mapped onto the Cartesian mesh by employing a scalar level set function ϕ that stores the unsigned distance to the boundary surface

and allows the efficient evaluation of the boundary outer normal in every mesh point as $\boldsymbol{n} = -\nabla\phi/|\nabla\phi|$. Note that for topologically closed boundary surfaces the signed distance may be used instead of the unsigned distance [5]. A cell is considered to be a valid fluid cell, if the distance at the cell *midpoint* satisfies the condition $\phi > h/2$ and as an exterior ghost cell otherwise. The mesh received from the shell solver corresponds to a two-dimensional manifold surface mesh (cf. Sec. 3) and the utilization of condition $\phi > h/2$ is a straightforward, unambiguous solution to achieve the mandatory thickening of this mesh by the shell 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 shell element is 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 $\boldsymbol{w}$ is $\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{w} \cdot \boldsymbol{n}$. Enforcing the latter with ghost cells, in which the discrete values are located at the cell centers, requires the mirroring of the primitive values ρ, $\boldsymbol{u}$, p, ρ_A across the embedded boundary. The normal velocity in the ghost cells is set to $(2\boldsymbol{w} \cdot \boldsymbol{n} - \boldsymbol{u} \cdot \boldsymbol{n})\boldsymbol{n}$, while the mirrored tangential velocity remains unmodified. Mirrored values are constructed by calculating spatially interpolated values in the point $\tilde{\boldsymbol{x}} = \boldsymbol{x} + 2\phi\boldsymbol{n}$ 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. This property is essential in simulating hyperbolic problems with discontinuities, like detonation waves. 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. 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 fluid 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 $\boldsymbol{w}$ in the ghost cells guarantees that the embedded boundary propagates at most one cell in every time step.

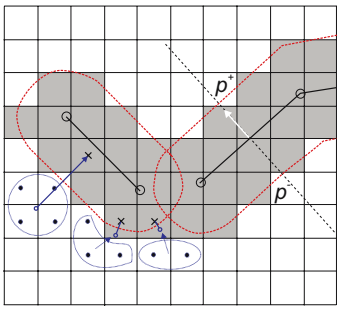

Fig. 1. Ghost cells (shaded gray) around shell elements and construction of mirrored values

In order to supply a fine local temporal and spatial resolution efficiently, the finite volume scheme described above has been incorporated into a block-oriented hierarchical mesh adaptation method. The numerical scheme is then applied as a single-grid routine in a loop over all subgrids (see [6] or [7] for details).

3 Lagrangian Thin-Shell Solver

The Kirchhoff-Love type thin-shell model applied in this study takes the membrane as well as bending response of the surface into account, and has been discretized with subdivision finite elements [2, 3]. The underlying kinematic assumptions allow for arbitrarily large displacements as well as rotations of the shell. Fracture initiation and propagation is considered as a progressive failure phenomenon in which the separation of the crack flanks is resisted by cohesive tractions. The relationship between the crack-opening displacements and the tractions is given by a cohesive model. Cohesive interface elements are inserted at inter-element edges and constrain the opening of the crack flanks to the deformation of the shell middle surface and its normal. This approach allows for fracture in an in-plane or tearing mode, a shearing mode, or a "bending of hinge" mode.

To kinematically describe a fractured thin-shell as sketched in Fig. 2, we consider a shell of uniform thickness $\overline{h}$ occupying an undeformed configuration $\overline{V}$. The position vector $\overline{\boldsymbol{\varphi}}$ of a material point on the undeformed shell body is assumed to be

$$\overline{\boldsymbol{\varphi}} = \overline{\boldsymbol{x}} + \theta^3 \overline{\boldsymbol{n}} \tag{2}$$

with $-h/2 \leq \theta^3 \leq h/2$. The position vector of the shell middle surface is denoted by $\overline{\boldsymbol{x}}$ and its out-of-surface unit normal by $\overline{\boldsymbol{n}}$. In other words, the shell middle surface represents a two-dimensional manifold in $\mathbb{R}^3$. The deformation mapping $\boldsymbol{\varphi}$ maps the shell body into the deformed configuration V and is discontinuous across the crack

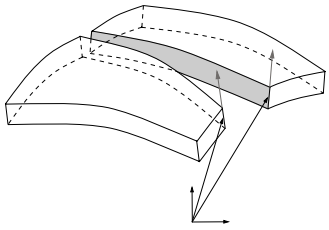

Fig. 2. Fractured shell body: Opposite crack flanks and corresponding normals.

$$[\![\boldsymbol{\varphi}]\!] = \boldsymbol{\varphi}^+ - \boldsymbol{\varphi}^- = [\![\boldsymbol{x}]\!] + \theta^3 [\![\boldsymbol{n}]\!] \tag{3}$$

where the superscripts $+$ and $-$ refer to the opposing crack flanks. Further, the first term describes the discontinuity of the deformation of the middle shell surface, and the second term the discontinuity in the shell out-of-surface normal. The discontinuities in the deformations can also be interpreted as the *opening displacement* of the crack.

A standard semi-inverse approach is followed for obtaining the shell equilibrium equations in weak from. To this end, the assumed reduced kinematic equations for the shell body (Equations (2) and (3)) are introduced into the conventional virtual work expression for the three-dimensional body. As previously mentioned, we consider fracture as a gradual separation phenomenon, resisted by cohesive tractions. Consequently, the internal virtual work expression contains the virtual work of the cohesive interface ($\delta \Pi_{C,\mathrm{int}}$) in addition to the virtual work of the bulk material ($\delta \Pi_{S,\mathrm{int}}$)

$$\delta \Pi_{S,\mathrm{int}} + \delta \Pi_{C,\mathrm{int}} - \delta \Pi_{\mathrm{ext}} = 0 \tag{4}$$

with

$$\delta \Pi_{S,\mathrm{int}} = \int_{\overline{\Omega}} \int_{-\overline{h}/2}^{\overline{h}/2} \boldsymbol{P} : \delta \boldsymbol{F}\, \mu \, d\theta^3 d\overline{\Omega} \;, \quad \delta \Pi_{C,\mathrm{int}} = \int_{\overline{\Gamma}_C} \int_{-\overline{h}/2}^{\overline{h}/2} \boldsymbol{T} \cdot [\![\boldsymbol{\varphi}]\!]\, \mu \, d\theta^3 d\overline{\Gamma}_C$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor, $\boldsymbol{T}$ the related traction vector at the cohesive surface, and $\boldsymbol{F}$ the deformation gradient. The virtual work expression for the bulk material is integrated over the undeformed shell middle surface $\overline{\Omega}$ and for the cohesive interface over the crack path $\overline{\Gamma}_C$. The scalar factor μ accounts for the curvature of the shell in the volume computation [3].

Next, we briefly outline the discretization of the governing equation (4). A detailed presentation of the used subdivision finite element discretization technique can be found in [2] and [3]. In this approach, the reference ($\overline{\boldsymbol{x}}$) and deformed ($\boldsymbol{x}$) shell surfaces are approximated using smooth subdivision surfaces belonging to the Sobolev space H_2 with square-integrable curvatures. The subdivision interpolation within one element is accomplished with shape functions, which have support on the element as well as on the one-ring of neighboring elements. The overlapping local subdivision interpolants, each defined over one patch, together lead to a global interpolant with square-integrable curvatures. In the presence of fracture, the smoothness and/or continuity of the interpolation has to be relaxed and the subdivision interpolant needs to be modified (see [4] for details).

Once fracture nucleates along an element edge, the element patches on the left and right side of the cracked edge interact only through cohesive tractions. The cohesive tractions are self-balanced internal forces derived from a cohesive fracture model [4]. In this model, the opening displacement $[\![\boldsymbol{\varphi}]\!]$ plays the role of a deformation measure while the traction $\boldsymbol{T}$ is the conjugate stress measure.

Finally, the inelastic behavior of the bulk material, i.e. the relation between $\boldsymbol{P}$ and $\boldsymbol{F}$, is described with a conventional J_2 plasticity model with isotropic power-law hardening. The rate-dependent behavior is described with a power viscosity law and constant rate sensitivity.

4 Efficient Level Set Evaluation

In Sec. 2, we have sketched the concept of employing a distance function to represent a complex embedded boundary on a Cartesian mesh. While distance functions are easily prescribed for single elementary geometric objects, their evaluation can be cumbersome for complex shapes. In coupled Eulerian-Lagrangian simulations, this complex shape is defined by the deforming shell surface mesh.

One can efficiently compute the distance on a grid by solving the eikonal equation with the method of characteristics and utilizing polyhedron scan conversion [11]. For a given grid point, the relevant closest point on the triangular mesh lies on one of the primitives (faces, edges and vertices) that comprise the surface. The characteristics emanating from each of these primitives form polyhedral shapes. Such a *characteristic polyhedron* contains all of the points which are possibly closest to its corresponding face, edge or vertex. The closest points to a triangle face must lie within a triangular prism defined by the face and

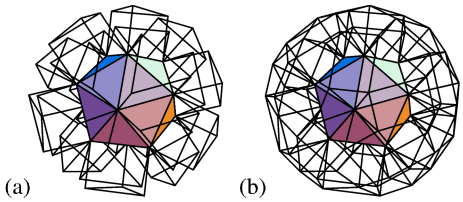

Fig. 3. The characteristic polyhedra for faces and edges of an icosahedron

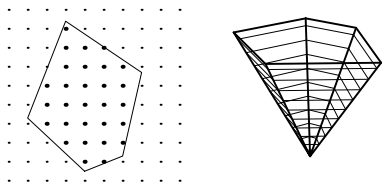

Fig. 4. Scan conversion of a polygon in 2-D and slicing of a polyhedron to form polygons

its normal; the closest points to an edge lie in a cylindrical wedge defined by the line segment and the normals to the two incident faces (see Fig. 3 for face (a) and edge (b) polyhedra as an example). Analogously, polygonal pyramids emanating from the vertices are also possible (not shown). We then determine the grid points that lie inside a characteristic polyhedron with polyhedron scan conversion. The polyhedron is first sliced along each sheet of the grid lattice to produce polygons, cf. Fig. 4. Simple geometric formulas are finally used to calculate the distance.

By utilizing the outlined techniques, and evaluating the distance exactly only within a small distance around the surface, a highly efficient algorithm can be formulated that has linear computational complexity both in the number of Cartesian mesh points and the surface triangles [11, 7].

5 Fluid-Structure Coupling

The explicit fluid and solid solvers are weakly coupled by successively applying appropriate boundary conditions in a time-operator splitting technique. In the case of inviscid flows, the compatibility conditions are simply the continuity of the velocity component normal to the embedded boundary u_n in solid (S) and fluid (F), i.e. $u_n^S = u_n^F$, and the continuity of the component normal of the solid's Cauchy traction vector, $p^S = (\boldsymbol{\sigma}\boldsymbol{n})\boldsymbol{n}$ with $\boldsymbol{\sigma} = 1/det(\boldsymbol{F})\boldsymbol{F}\boldsymbol{P}$, and the hydrodynamic pressure p^F, i.e $p^S = p^F$. We use the update algorithm of Fig. 5 to implement these conditions numerically. After evaluating the distance function ϕ for the currently available shell surface mesh, the embedded wall boundary velocities for the fluid solver are set to the solid velocities in the nearest shell element midplane. The same velocity $\boldsymbol{w}$ is enforced in the fluid on upper ($^+$) and lower ($^-$) side of each element. After setting embedded rigid wall boundary conditions as sketched in Sec. 2 and the fluid update, a new hydrodynamic pressure load $p^F := p^+ - p^-$ on each shell element is derived (compare Fig. 1). With these new boundary conditions, the cycle is completed by advancing the solid by Δt,

update $\phi(t)$
$\boldsymbol{w}_F^{+/-} := \boldsymbol{u}^S(t)$
update_fluid(Δt)
$p^S := p^F(t + \Delta t)$
update_solid(Δt)
$t := t + \Delta t$

Fig. 5. Basic coupling algorithm

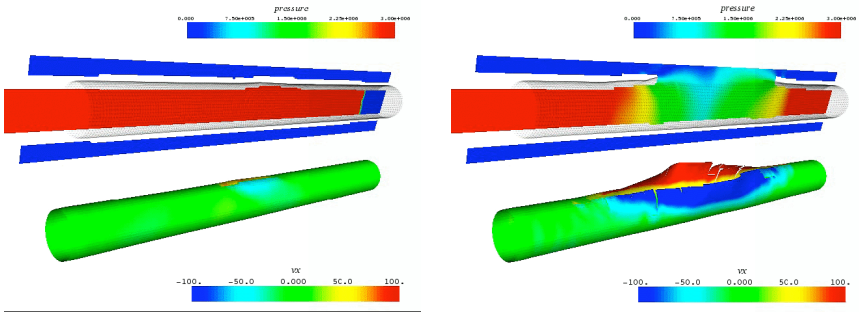

Fig. 6. Coupled simulation of detonation-driven rupture of a thin aluminum tube. Two snapshots of the fracturing tube with velocity iso-contours (lower row); cuts through the fluid domain show the resulting hydrodynamic venting (upper row).

which in practice is typically done by taking multiple, smaller time steps in the solid solver to effectively accommodate the more restrictive stability condition in the solid. An extended version of above coupling scheme compatible with the optional fluid mesh adaptation method with recursive time step refinement has also been developed [7].

In our current implementation, fluid and shell solver are parallelized separately for distributed memory machines using rigorous domain decomposition methods. The two independent solvers run on two disjoint set of processors. To facilitate the efficient exchange of the distributed fluid-shell boundary information (surface mesh and velocities to the fluid; pressure loadings to the solid) we have implemented a non-blocking high-level communication library that determines the necessary point-to-point communication patterns by intersecting Cartesian bounding boxes enclosing the local domains.

6 Detonation-Driven Fracture Example

We present one representative simulation that corresponds to an experimental configuration studied by Chao [1]. The setup consists of a detonation tube of 1.52 m, filled with $C_2H_4 + 3\,O_2$ at $p_0 = 180\,\mathrm{kPa}$ and $\rho_0 = 2.33\,\mathrm{kg/m^3}$, to which a thin-walled aluminum (Al6061-T6) test tube is attached. The test specimen has a length of 45.7 cm, an inner radius of 1.975 cm, and a wall thickness of 0.89 mm. While the lower end of the entire device is closed, a thin diaphragm seals the upper end. To ensure a reproducible fracture pattern Chao's specimen has a central longitudinal notch of 6.32 cm at the middle, which is modeled as an initial crack in the computations.

The mixture is thermally ignited at the closed end. A combustion wave arises that has the characteristics of a freely propagating self-sustained detonation when it enters the test specimen. Utilizing the full GRI 3.0 reaction mechanism Chapman-Jouguet theory predicts its velocity to be $D_{\mathrm{CJ}} = 2404\,\mathrm{m/s}$ and an adiabatic mixture coefficient for the fully reacted state of 1.24 [1]. For

the simplified detonation model of Sec. 2, we choose $\gamma = 1.24$ and evaluate $q_0 \approx 5.20\,\mathrm{MJ/kg}$, $p_{\mathrm{CJ}} \approx 6.10\,\mathrm{MPa}$, and $\rho_{\mathrm{CJ}} \approx 4.16\,\mathrm{kg/m^3}$. A one-dimensional simulation is used to calculate the flow field at the moment the detonation enters the specimen. The one-dimensional data is then taken as initial conditions for the three-dimensional reactive Euler equations. Fig. 7 shows the detonation propagation in the one-dimensional case and the development of a Taylor rarefaction wave due to the closed end is clearly visible. To model this wave correctly, the Cartesian fluid domain also encompasses 0.92 m of the downstream end of the detonation tube as a static embedded boundary. Fig. 6 shows computational results of a typical coupled fluid-structure interaction simulation with a shell mesh of 8665 elements and a uniform Cartesian fluid mesh of $40 \times 40 \times 725$ cells that required $\approx 900\,\mathrm{h}$ CPU on 27 nodes of a Pentium-4-2.4 GHz dual processor system (21 fluid and 33 solid processes). 1300 coupled time steps with fixed step size to a final time of 0.26 ms have been calculated (20 solid solver sub-steps in each fluid time step). The left graphic of Fig. 6 shows the beginning of the crack opening at $t \approx 0.15\,\mathrm{ms}$ (detonation propagating from left to right). The snapshot on the right shows the rupture at the final time $t = 0.26\,\mathrm{ms}$. The venting of high pressure reacted gas out of the tube is clearly visible. During this simulation, level set evaluation and finite volume scheme (excluding parallel synchronization operations) have about the same computational costs on each fluid processor.

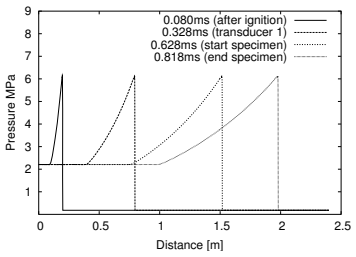

Fig. 7. 1d detonation propagation through tube and specimen

7 Conclusions

A parallel level-set-based fluid-structure coupling method for the time-accurate simulation of thin flexible shells dynamically responding to gaseous detonation waves has been described. The approach has been demonstrated to handle arbitrary topology changes and large deformations and is computationally very efficient. Its implementation in the software framework "Virtual Test Facility" is freely available for research purposes (cf. http://www.cacr.caltech.edu/asc).

References

1. T.-W. Chao. *Gaseous detonation-driven fracture of tubes.* PhD thesis, California Institute of Technology, 2004.
2. F. Cirak, M. Ortiz, P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int. J. Numer. Meth. Engineering*, 47:2039–2072, 2000.
3. 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.

4. F. Cirak, M. Ortiz, and A. Pandolfi. A Cohesive Approach to Thin-Shell Fracture and Fragmentation. *Computer Methods in Appl. Mechanics and Engineering*, 194:2604–2618, 2005.
5. F. Cirak and R. Radovitzky. A Lagrangian-Eulerian Shell-Fluid Coupling Algorithm Based on Level Sets. *Computers & Structures*, 83:491–498, 2005.
6. R. Deiterding. *Parallel adaptive simulation of multi-dimensional detonation structures*. PhD thesis, Brandenburgische Technische Universität Cottbus, Sep 2003. Available at http://www.cacr.caltech.edu/~ralf/html/pub.htm.
7. R. Deiterding, R. Radovitzky, S. P. Mauch et al. A virtual test facility for the efficient simulation of solid materials under high energy shock-wave loading. *Engineering with Computers*, 2005. Invited submission.
8. 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.
9. W. Fickett and W. C. Davis. *Detonation*. University of California Press, Berkeley and Los Angeles, California, 1979.
10. C. L. Mader. *Numerical modeling of detonations*. University of California Press, Berkeley and Los Angeles, California, 1979.
11. S. P. Mauch. *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, 2003.
12. E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, Heidelberg, 2nd edition, 1999.