An Eulerian-Lagrangian Approach for Large Deformation Fluid Structure Interaction Problems, Part 2: Multi-Physics Simulations within a Modern Computational Framework

Todd Harman[1], James E. Guilkey[1], Bryan Kashiwa[2], John Schmidt[1] & Patrick McMurtry[1]
[1] *Dept. of Mechanical Engineering, University of Utah, Salt Lake City, UT*
[2] *Theoretical Division, Los Alamos Natl. Laboratory, Los Alamos, NM*

Abstract

A tightly coupled fluid-structure interaction (FSI) solution technique incorporating fluid and solid mechanics, phase change and chemical reactions is presented. The continuum equations are solved with a cell-centered, multi-material ICE solution method. This formulation is integrated with a Lagrangian, particle based, solid mechanics technique, known as the Material Point Method, as described by Kashiwa et al. [1] and Guilkey et al. [2]. The combined method can handle large deformations and phase change within a single grid, without the need of separate domains for fluids and solids, or the passing of boundary conditions. This paper discusses algorithmic issues involved in accounting for chemical reactions and phase transition among material phases (e.g., solid $\rightarrow$ gas). Validation is presented as are simulations showing large deformation with phase change. These simulations were performed within a computational framework that contains tools for parallelization, performance analysis, data management, algorithm integration, and data visualization. Features of this framework are described.

1 Introduction

Advancements in information and computing technology have significantly increased the level of sophistication that can be achieved in fluid structure interaction simulations. These advancements, in both software and hardware, include computational frameworks, scientific visualization, data storage, processing power and commu-

nication bandwidth for massively parallel computations to name a few. These advances have enabled engineers to perform FSI simulations on large parallel platforms. Specifically, the development of a computational framework, with built-in parallelization, data storage and visualization, was key to the implementation of the multi-material Eulerian-Lagrangian algorithm described in Guilkey et al. [2] (which is Part 1 of this paper, and appears in these proceedings).

The algorithm combines a multi-material Eulerian solution for the governing continuum equations for the fluids with a Lagrangian particle based technique for the solid mechanics. Incorporated in the algorithm are terms to account for momentum and energy exchange along with phase change between the different materials or phases. The organization of this paper is as follows, first, the important aspects of the computational framework are described, followed by a discussion directed at the governing equations with an emphasis on mass conversion among the materials. The paper concludes with validation and demonstration simulations.

2 The Computational Framework

Our code development and production environment is called the Uintah Computational Framework (UCF) and was initially developed at the University of Utah in support of the DOE sponsored Center for the Simulation of Accidental Fires and Explosions (C-SAFE). The UCF was specifically designed as a general framework for performing massively parallel simulations as well as a tool for visualizing large datasets. It provides a platform for incorporating a variety of physics algorithms into a computing environment that supports both MPI and thread based parallelism [3, 4]. In the UCF a wide assortment of data structures are available to the application programmer, including Particle, Cell, Node and Face centered data on a structured grid.

Domain decomposition is the foundation of the UCF wherein the computational domain is divided into individual “patches”, upon which the algorithm is performed. To hide the complexities of parallel data management from the researcher, the UCF incorporates three key features; 1) a task graph representation of the algorithm, 2) a component known as the Data Warehouse, 3) and a Scheduler component. The Data Warehouse acts as a global, single assignment memory abstraction with automatic data lifetime management and storage reclamation. This warehouse automatically handles I/O, check-pointing and restarting. The application developer’s algorithm is described by “tasks” where each task corresponds to a single major algorithmic step. Each task contains a list of required input variables and a list of computed variables. The input requirements have information on data dependencies on neighboring patches. The Scheduler component arranges these tasks into a task graph representation of the algorithm. Data flow among the patches and tasks are represented as “edges” on the graph. From the application developers perspective, tasks to do this communication are generated automatically, and are scheduled along with those tasks specified by the researcher.

To use the UCF for any particular application, a researcher defines the algorithm on a single patch with tasks. In those tasks the data requirements must be specified,

including the number of “ghost” cells needed from the neighboring patches. Once the tasks are specified, the Scheduler creates a task graph, and then distributes the tasks to the available computational resources. Since communication tasks are automatically scheduled and executed, retrieval of data from neighboring patches is transparent to the developer.

For example, suppose we have an algorithm that consists of just two steps, interpolating the particle state to the nodes, and then interpolating the nodal data back to the particles. The tasks describing this algorithm are: (see Fig. 1)

1. InterpolateParticleDataToNodes

   Requires: Particle position, mass and velocity, from the patch and one layer of ghost cells.
   Computes: Nodal velocity on the patch

2. InterpolateNodeDataToParticles

   Requires: Nodal velocity, from the patch and one layer of ghost cells.
   Computes: Particle velocity, position on the patch

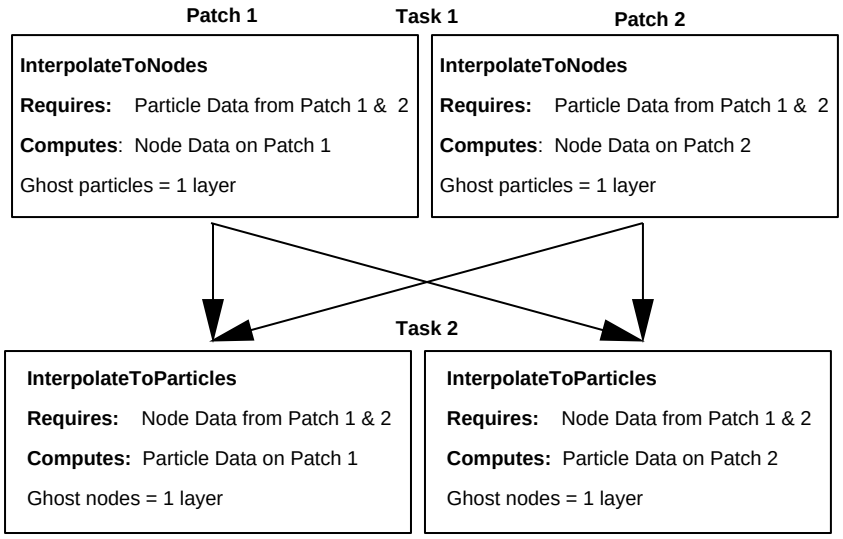

Figure 1: Schematic diagram of a task graph generated by the Scheduler component. Arrows, or “edges”, represent data dependency. Diagonal arrows indicate data communication between patches.

This representation of the algorithm makes it effortless for the researcher to parallelize an algorithm, since no additional code is needed to extend a single patch, single processor simulation onto a large computing platform.

Because an algorithm's individual tasks are insulated from the underlying communication and parallelization infrastructure, it is possible to make significant changes to the framework without disturbing the researcher's work. This gives the computer scientists the freedom to implement new schedulers, integrate performance and analysis tools and optimize existing schedulers without making radical changes to the individual tasks.

3 Incorporation of Phase Change and Heat Release

The coupled Eulerian-Lagrangian approach presented in Part 1 is directed at solving large deformation, full-physics problems. By full-physics, we refer to transient solutions of the governing equations of both solids and fluids, including phase change, without limiting assumptions made on material behavior or eventual state of deformation. An example scenario is an exploding metal container, initially filled with plastic bonded explosives (PBX), subject to an external heat source. The external heating initiates a solid $\rightarrow$ gas reaction in the layer of PBX adjacent to the case. Generation of gas in the gap causes a rapid pressurization and large deformation of the case until it eventually ruptures. Such a scenario is known as a "fast cookoff" test, and is of interest in the solid rocket motor community.

Several different approaches have been developed for solving large deformation, full-physics problems over the years. Most of these fall into a class where distinctly separate domains are assigned to the fluid and solid fields. These techniques often use an ALE mesh for the fluid and a Lagrangian mesh for the solid. Some general classifications and examples have been provided by other authors (e.g., [5]). Other approaches have used a purely Eulerian solution with surface tracking to keep the boundaries among the different materials distinct. In both classes, communication, or boundary condition passing is required between the fluid and the solid. One can imagine the difficulties that either of these approaches would have with our target scenario. For example, the solid $\rightarrow$ gas reaction generates a fluid where solid existed previously or when the container is strained beyond its plastic limit rupturing into several smaller pieces, with interior gases escaping.

As described in detail in Part 1 [2], a different approach is taken here. The governing multi-material equations are solved on a single mesh using a conserving, Eulerian, finite volume method. The solid phase materials are described in a Lagrangian frame using the particle based material point method (MPM) that is described in Sulsky et al. [6, 7]. The MPM solution technique is embedded within the Eulerian method and information is passed back and forth between the two methods in the Eulerian frame of reference via the single grid as described by Kashiwa et al. [1]. Because all materials use the same grid for at least part of the computation, the algorithm is capable of handling any amount of material in any cell (i.e., arbitrarily small). This makes the generation of gas in the midst of a solid a possibility that would be difficult if the solid and fluid were described in separated domains.

The solid phase materials thus have a dual representation in both the Lagrangian and Eulerian reference frames. The reader is encouraged to read Part 1 of this work

for a detailed description of the integrated technique.

3.1 Governing Equations and Reaction Models

We begin by reviewing the governing multi-material equations as described in Guilkey et al.[2] and Kashiwa [8] with our attention focused on the $(r \rightarrow n)$source terms of mass, momentum, energy and specific volume. The multi-material equations solved in the Eulerian framework are

$$\frac{\partial \rho^r}{\partial t} = -\frac{\partial (\rho u_j)^r}{\partial x_j} + \sum_{n=1, n\neq r}^{N} S_{\rho}^{r\leftrightarrow n} \tag{1}$$

$$\frac{\partial (\rho u_i)^r}{\partial t} + \frac{\partial (\rho u_i u_j)^r}{\partial x_j} = \theta^r \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{\partial}{\partial x_j}\theta^r(\sigma_{ij}^r - \sigma_{ij}) + \sum_{s=1}^{N} f_{sr} + \sum_{n=1, n\neq r}^{N} S_{\rho u}^{r\leftrightarrow n} \tag{2}$$

$$\frac{\partial (\rho e)^r}{\partial t} + \frac{\partial (\rho e u_j)^r}{\partial x_j} = -p\left(\frac{\theta \dot{v}}{v}\right)^r + \theta^r(\sigma_{ij}^r - \sigma_{ij})\frac{\partial u_i^r}{\partial x_j} + \theta^r \tau_{ij}^r \frac{\partial u_i^r}{\partial x_j} + \sum_{s=1}^{N} h_{sr} \sum_{n=1, n\neq r}^{N} S_{\rho e}^{r\leftrightarrow n} \tag{3}$$

$$\left(\frac{\theta \dot{v}}{v}\right)^r = f_r^{\theta} \nabla \cdot \sum_{r=1}^{N} (\theta u)^r + \left[(\theta \alpha \dot{T})^r - f_r^{\theta} \sum_{s=1}^{N} (\theta \alpha \dot{T})^r\right] + S_{(\rho v)^r}^{r\leftrightarrow n} \tag{4}$$

where ρ^r, $(\rho u)^r$, and $(\rho e)^r$ are the r-material mass, momentum, and energy per unit volume, θ^r is the volume fraction and v^r is the specific volume. A key feature of the method is that the intrinsic properties (temperature, velocity and specific volume) of the materials are well defined everywhere, even if the mass of that material is zero in a region.

$S_{\rho}^{r\leftrightarrow n}$, $S_{\rho u}^{r\leftrightarrow n}$, $S_{\rho e}^{r\leftrightarrow n}$ and $S_{v^r}^{r\leftrightarrow n}$ represent, for each r-material, the change in mass, momentum, energy and specific volume resulting from the conversion of mass among materials. As an example, we return to the exploding container scenario described above. When the temperature and pressure are above a threshold, the solid $\rightarrow$ gas reaction of PBX occurs on a surface and is given by:

$$S_{\rho^r}^{r\leftrightarrow n} = \text{(A)} * \text{(K)}\left(\frac{P}{P_{\text{reference}}}\right)^n \tag{5}$$

$$S_{(\rho u)^r}^{r\leftrightarrow n} = -u^r S_{\rho^r}^{r\leftrightarrow n} \tag{6}$$

$$S_{(\rho e)^r}^{r\leftrightarrow n} = \Delta h S_{\rho^r}^{r\leftrightarrow n} \tag{7}$$

$$S^{r\leftrightarrow n}_{(\rho v)^r} = S^{r\leftrightarrow n}_{\rho^r} v^r \tag{8}$$

where A is a surface area, K is a reaction coefficent, and Δh is the enthalpy of reaction. K and n in Eq. 5 were experimentally determined [9]. To maintain conservation, careful accounting of not only the mass, but also the transfer of momentum and energy among the materials is important. Accounting for changes to the solid state is straightforward, as decrementing a particle's mass automatically decrements it's momentum and energy. These same quantities are explicitly added to the product material as additional source terms in Step 8 of Part 1[2]. To date, only solid $\rightarrow$ gas reactions have been considered, however, other processes are possible. For example, with appropriate models, a solid material may melt into a liquid that subsequently vaporizes into a gas.

In addition to the source terms in Eqs. 5-8, the estimated time advanced pressure has an additional term to account for mass exchange:

$$\Delta P = -\Delta t \frac{\sum_{r=1}^{N} v^r S^{r\leftrightarrow n}_{\rho^r} - \sum_{r=1}^{N} \nabla\cdot(\theta u^*)^r}{\sum_{r=1}^{N} (\theta\kappa)^r} \tag{9}$$

where κ^r is the r-material bulk compressibility. Equation 9 is derived from the mass equation, Eq. 1 and the statement,

$$\sum_{r=1}^{N} \theta^r = \sum_{r=1}^{N} \rho^r v^r(p) = 1 \tag{10}$$

thus any change in mass must be included. For details see Kashiwa [10].

4 Demonstration Simulations and Validation

In this section simulation results are presented for three different scenarios to validate and demonstrate the capabilities of this approach. The first simulation is a classical thermodynamics problem, a piston compressing a gamma-law gas very slowly. This serves to validate the evolution of the specific volume and the source of internal energy under large volume changes. The next demonstration is of a transient simulation of a flexible structure in a crossflow. This demonstration illustrates algorithm features in large-deformation FSI problems. The final simulation involves large deformations of a sealed metal cylinder resulting from high pressurization from a solid $\rightarrow$ gas reaction. In all cases a single structured mesh was used, with material points representing the solid materials.

4.1 Validation: Adiabatic compression of a confined gas by a piston

A piston adiabatically compressing a confined gamma-law gas was simulated to test the Eulerian evolution equation for the specific volume, Eq. 4 and the source

of internal energy under large volume changes. This simulation also demonstrates that the algorithm accurately represents the interaction between the piston and the gas through the momentum coupling. The piston was described by material points and its motion was specified. As mentioned in Guilkey et al. [2] the specific volume must be computed accurately to obtain the correct equilibration pressure. Figure 2 shows the cell centered gas pressure at $y = 0.0$ versus time, where the analytical solution assuming thermodynamic equilibrium is given by

$$P_2 = P_1 \left(\frac{V_1}{V_2} \right)^{\gamma} \tag{11}$$

A large momentum exchange coefficient, ($K_{rs} = 10^{15}$) was chosen. This drives the velocity of the gas and the piston to the same value in cells where both gas and piston are present (i.e., at the interface between the gas and the solid). This enforces a no-slip, no-interpenetration condition between the piston and the gas. This problem had a compression ratio of 5.9.

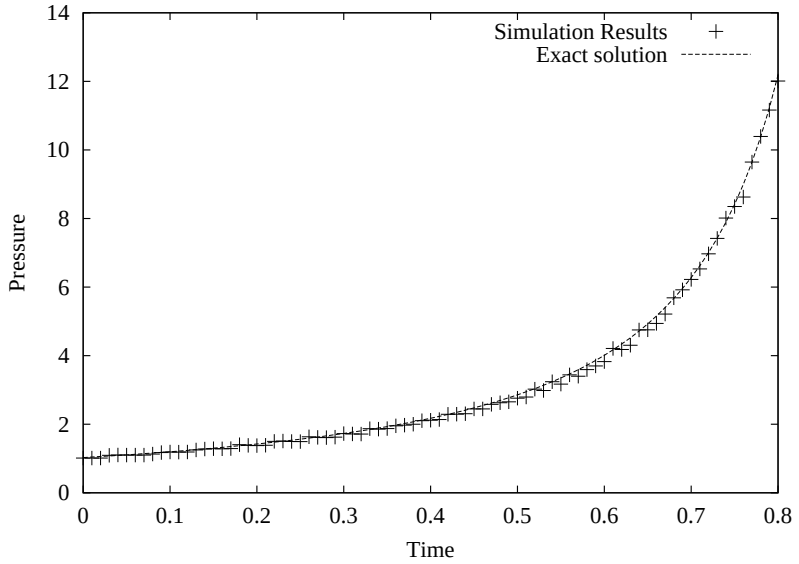

Figure 2: Adiabatic compression of a confined gas by a piston. Pressure versus time at $y = 0.0$.

4.2 Flexible structure in a crossflow

This simulation is intended to demonstrate the dynamic capabilities of the algorithm. Specifically, a light, flexible, rectangular beam with a density of 1.18 kg/m^3 and bulk and shear modulus of .2 MPa and .15 MPa respectively is attached to

the floor of an enclosed channel. The beam is represented by 8 material points per computational cell, has a footprint of .5 m X .5 m, is 1.5 m tall and is initially at a temperature of 300K. Air at 400K is forced through the channel with an inlet velocity of 10 m/s. Upon start-up, the beam deflects and undergoes subsequent oscillations. Figure 3 is a snapshot of the deformed shape of the beam as well as a vertical slice of the surrounding flow field at an instant in time. The particles comprising the beam are shaded by temperature, indicating that heat transfer is occurring in this simulation. The rectangular box shows the computational domain and the dividing line indicates that two patches were used.

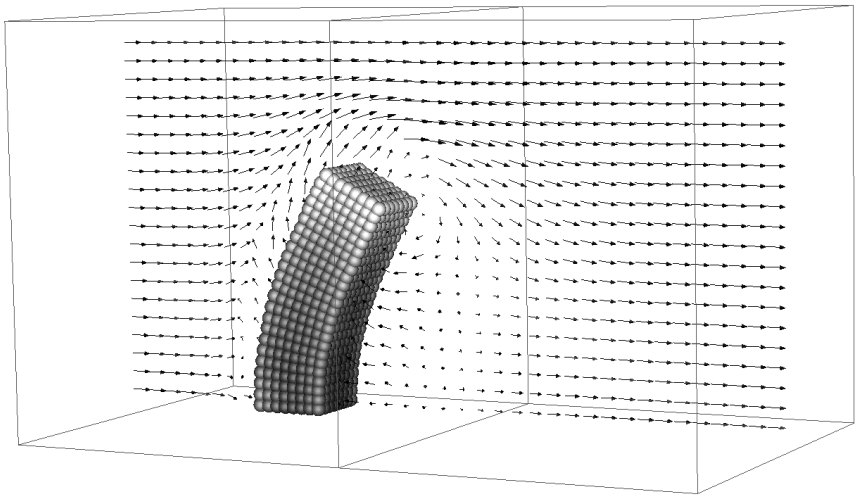

Figure 3: Flexible structure in a crossflow.

4.3 Explosion of Confined Energetic Materials

This final simulation illustrates the algorithm's capabilities with regards to mass exchange among materials, large deformations, and the handling of multiple materials. The simulation involves a cylindrical copper container (gray particles) filled with a PBX (dark particles). The initial temperature of the container was $650K$, well above the reaction threshold temperature for PBX. The reaction model discussed in section 3.1 was employed. Figures 4a-d show a time sequence of the evolution of the container and PBX. The initial configuration is shown in a $1/8^{th}$ symmetry view in Fig. 4a. The reaction initiates at the interface between the copper and PBX, resulting in the generation of gases (not explicitly depicted in the figure). The creation of product gases in the gap generates a high pressure that further accelerates the reaction causing the case to deform as shown in Figs. 4b-c. Notice the uniform circumferential deformation of the case as a result of the high

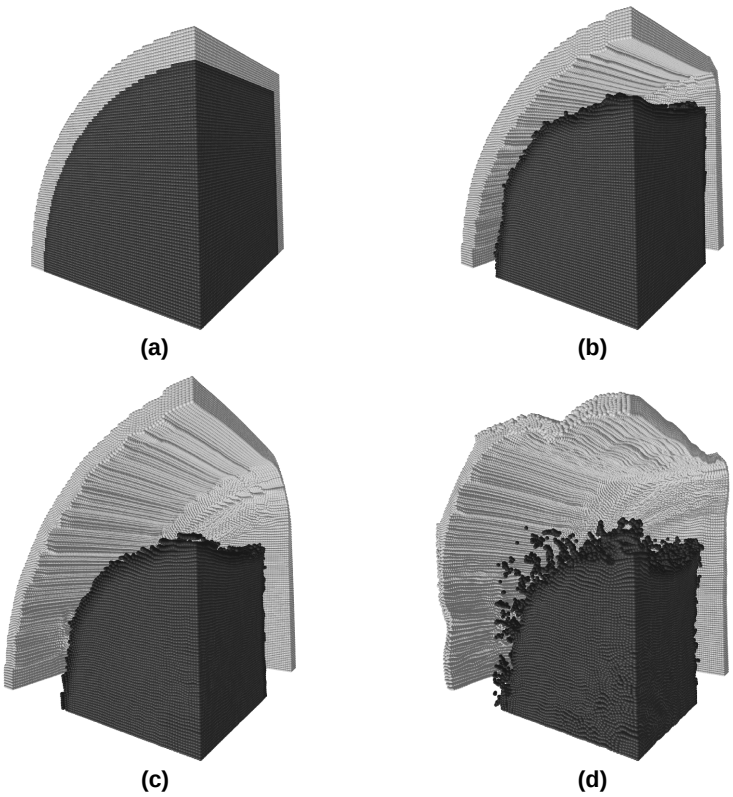

Figure 4: Deformation of a pressurized container from the reaction of explosive contents. The copper container is denoted by gray color particles and the dark particles are PBX. A $1/8$ view of the initial configuration is shown in (a).

pressure forces in Fig. 4c. Eventually, the case impacts on the sides of the computational boundary, resulting in the deformed shape shown in Fig. 4d. An important feature to recognize about this simulation is that initially the volume fraction of the gas was zero everywhere within the container. As the reaction initiated, gas was generated in regions formally occupied by the solid materials. This simulation was performed on 125 processors of an SGI Origin 2000. Such simulations are made possible through recent IT advances, namely the development of the UCF computational framework that is capable of taking full advantage of this hardware.

5 Acknowledgments

This work was supported by the National Science Foundation's Information Technology Research Program under grant CTS0218574, and the U.S. Department of Energy through the Center for the Simulation of Accidental Fires and Explosions, under grant W-7405-ENG-48. Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36; partial support from the Procter & Gamble Company, and from DOE Office of Industrial Technologies, is gratefully acknowledged.

References

[1] Kashiwa, B., Lewis, M. & Wilson, T., Fluid-structure interaction modeling. Technical Report LA-13111-PR, Los Alamos National Laboratory, Los Alamos, 1996.

[2] Guilkey, J., Harman, T., Xia, A., Kashiwa, B. & McMurtry, P., An Eulerian-Lagrangian approach for large deformation fluid structure interaction problems: Algorithm development. *Proceedings of the Second International Conference on Fluid Structure Interaction*, Cadiz, Spain, 2003.

[3] Rawat, R., Parker, S., Smith, P. & Johnson, C., Parallelization and integration of fire simulations in the Uintah PSE. *Tenth SIAM Conference on Parallel Processing for Scientific Computing*, 2001.

[4] Parker, S., A component-based architecture for parallel multi-physics PDE simulation. *International Conference on Computational Science (ICCS2002) Workshop on PDE Software*, 2002.

[5] Rugonyi, S. & Bathe, K., On finite element analysis of fluid flows fully coupled with structural interactions. *CMES*, **2**, pp. 195–212, 2001.

[6] Sulsky, D., Chen, Z. & Schreyer, H., A particle method for history-dependent materials. *Comp Methods Appl Mech Engrg*, **118**, pp. 179–196, 1994.

[7] Sulsky, D., Zhou, S. & Schreyer, H., Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications*, **87**, pp. 236–252, 1995.

[8] Kashiwa, B., A multified model and method for fluid-structure interaction dynamics. Technical Report LA-UR-01-1136, Los Alamos National Laboratory, Los Alamos, 2001.

[9] Atwood, A., Boggs, T., Curran, P., Parr, T., Hanson-Parr, D. & Price, C., Burning rate of solid propellant ingredients, part 1: Pressure and initial temperature effects. *Journal of Propulsion and Power*, **15(6)**, pp. 740–747, 1999.

[10] Kashiwa, B. & Rauenzahn, R., A multimaterial formalism. Technical Report LA-UR-94-771, Los Alamos National Laboratory, Los Alamos, 1994.