Journal of Computational Physics **183,** 26–82 (2002)
doi:10.1006/jcph.2002.7158

A Conservative Three-Dimensional Eulerian Method for Coupled Solid–Fluid Shock Capturing[1]

G. H. Miller*,† and P. Colella†

*University of California, Department of Applied Science, 1 Shields Avenue, Davis, California 95616;
and †Lawrence Berkeley National Laboratory, Applied Numerical Algorithms Group,
1 Cyclotron Road, Berkeley, California 94720
E-mail: grgmiller@ucdavis.edu and pcolella@lbl.gov

Received March 13, 2002; revised June 26, 2002

A new method is presented for the explicit Eulerian finite difference computation of shock capturing problems involving multiple resolved material phases in three dimensions. We solve separately for each phase the equations of fluid dynamics or solid mechanics, using as interface boundary conditions artificially extended representations of the individual phases. For fluids we use a new 3D spatially unsplit implementation of the piecewise parabolic (PPM) method of Colella and Woodward. For solids we use the 3D spatially unsplit Eulerian solid mechanics method of Miller and Colella. Vacuum and perfectly incompressible obstacles may also be employed as phases.

A separate problem is the time evolution of material interfaces, which are represented by planar segments constructed with a volume-of-fluid method. The volume fractions are advanced in time using a second-order 3D spatially unsplit advection routine with a velocity field determined by solution of interface-normal two-phase Riemann problems. From the Riemann problem solutions we also determine cross-interface momentum and energy fluxes.

The volume fractions in mixed cells may be arbitrarily small, which would ordinarily make the Courant–Friedrichs–Lewy time step stability limit arbitrarily small as well. We overcome this limitation using the mass-redistribution formalism to conservatively redistribute generalized mass in the neighborhood of the split cells.

[1] Work at the Lawrence Berkeley National Laboratory was sponsored by the US Department of Energy (DOE) Mathematical, Information, and Computing Sciences Division Contract DE-AC03-76SF00098, and a DOE LDRD award to GM. Other work was supported by a subcontract from the California Institute of Technology Center for the Simulation of Dynamic Response in Materials, which in turn is supported by the Academic Strategic Alliances Program of the Accelerated Strategic Computing Initiative (ASCI/ASAP) under DOE Contract B341492. Development of the PPM implementation presented here was supported in part by the ASCI/ASAP Center for Astrophysical Thermonuclear Flashes at the University of Chicago under DOE Contract B341495.

0021-9991/02 $35.00

We present an application of this method to an explosion contained in a metal can: a reactive fluid (approximating PBX 9404) is encased within an elastic-plastic solid (approximating copper) surrounded by vacuum. Our implementation is in parallel, and with adaptive mesh refinement.

Key Words: solid mechanics; fluid dynamics; shock waves; Godunov method; elasticity; plasticity; solid–fluid coupling; adaptive mesh refinement; AMR.

1. INTRODUCTION

We are concerned with computing large-deformation flows in problems consisting of multiple resolved phases. The algorithm described here combines two approaches in the treatment of free boundaries. One is the treatment of the propagation of surfaces in space in terms of an equivalent evolution of volume fractions [20, 30], first introduced in the 1970s for representing fluid interfaces. The second is the use of strongly conservative finite-volume discretizations of the time-dependent irregular control volumes on either side of the moving interface. This latter approach was first used by Noh in the CEL code [29]. The combination of these two techniques has been used very successfully in a variety of free-boundary problems in fluid dynamics, such as shock dynamics [5, 9], and combustion fronts [19, 24, 34]. However, the combination has not been applied to problems involving material interfaces. More typically, volume-of-fluid representations have been used in material interface problems in a hybrid mode, i.e., in which the material properties are multiply valued in a cell, but the velocity is single valued [13, 26, 30]. In the present approach, all of the state variables are represented as having distinct values in each phase, with appropriate jump conditions applied at the phase boundaries.

Our method embraces fluids as well as elastic-plastic solids in a single Eulerian framework. This approach is motivated by the excellent performance of Eulerian high-order Godunov methods for single material phases. Furthermore, in the Eulerian framework adaptive mesh refinement (AMR) [3, 6] is a relatively mature technique for dynamically applying high numerical resolution to those parts of a problem domain that require it, while solving less sensitive regions on less expensive, coarser computational grids. In combination, Eulerian high-order Godunov methods with AMR have been proven to obtain highly accurate and efficient solutions to shock capturing problems. Our implementation includes AMR capability in 3D and in parallel.

2. OVERVIEW OF METHOD

We are concerned here with N spatially distinct (resolvable) material phases $\alpha, \alpha = 1, 2, \ldots, N$. For each phase we have a system of conservation laws

$$\frac{\partial U^\alpha}{\partial t} + \nabla \cdot F^\alpha(U) = G^\alpha \quad \text{on } \Omega^\alpha(t). \tag{1}$$

The number of components in the vector U^α of conserved densities, and the vector G^α of source terms, will in general be different for different phases, and the flux functions $F^\alpha(U)$ will differ in number and content: different material phases are governed by different physics and kinematics.

On domain boundaries, $\partial\Omega^{\alpha,\beta}(t) = \partial\Omega^{\alpha}(t) \cap \partial\Omega^{\beta}(t)$, compatibility conditions

$$H(U^{\alpha}, U^{\beta}) = 0 \quad \text{on}\, \partial\Omega^{\alpha,\beta}(t) \tag{2}$$

exist which determine the evolution of the domains $\Omega^{\alpha}(t)$.

We are interested in developing a numerical method for (1) and (2) on a structured Eulerian grid. We denote with integer subscripts i, j, k cell-centered quantities, and with half-integer subscripts, e.g., $i + \frac{1}{2}, j, k$, face-centered quantities. Increments of time are indicated with the superscript n or $n + 1$.

Let us denote by Υ_{ijk} a Cartesian grid control volume,

$$\Upsilon_{ijk} := \left\{ x, y, z \,\middle|\, \begin{array}{ll} x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}} & \text{and} \\ y_{j-\frac{1}{2}} \le y \le y_{j+\frac{1}{2}} & \text{and} \\ z_{k-\frac{1}{2}} \le z \le z_{k+\frac{1}{2}} & \end{array} \right\}, \tag{3}$$

and by $\partial\Upsilon_{i-\frac{1}{2},j,k}$ (for example) the "left" x face of cell i, j, k,

$$\partial\Upsilon_{i-\frac{1}{2},j,k} := \left\{ x, y, z \,\middle|\, \begin{array}{ll} x = x_{i-\frac{1}{2}} & \text{and} \\ y_{j-\frac{1}{2}} \le y \le y_{j+\frac{1}{2}} & \text{and} \\ z_{k-\frac{1}{2}} \le z \le z_{k+\frac{1}{2}} & \end{array} \right\}. \tag{4}$$

We define volume fractions Λ

$$V\Lambda_{ijk}^{\alpha,n} = \int\limits_{\Upsilon_{ijk}\cap\Omega^{\alpha}(t^{n})} dx\, dy\, dz, \tag{5}$$

where $V = \Delta x_i \Delta y_j \Delta z_k$ is the volume of cell ijk. We also define the phase "apertures"

$$A_{i-\frac{1}{2},j,k}^{\alpha} = \int\limits_{t^{n}}^{t^{n+1}} dt \int\limits_{\partial\Upsilon_{i-\frac{1}{2},j,k}\cap\Omega^{\alpha}(t)} dy\, dz. \tag{6}$$

Finally we define the interface boundary aperture

$$A_{ijk}^{B,\alpha} = \int\limits_{t^{n}}^{t^{n+1}} dt \int\limits_{\Upsilon_{ijk}\cap\partial\Omega^{\alpha}(t)} dx\, dy\, dz. \tag{7}$$

A finite volume discretization of (1) is obtained by integrating (1) over the space–time volume of the computational cell ijk occupied by phase α; i.e.,

$$\int\limits_{t^{n}}^{t^{n+1}} dt \int\limits_{\Upsilon_{ijk}\cap\Omega^{\alpha}(t)} dx\, dy\, dz \left(\frac{\partial}{\partial t}\frac{\partial}{\partial x}\frac{\partial}{\partial y}\frac{\partial}{\partial z}\right) \cdot \begin{pmatrix} U^{\alpha} \\ F^{x,\alpha}(U) \\ F^{y,\alpha}(U) \\ F^{z,\alpha}(U) \end{pmatrix} = \int\limits_{t^{n}}^{t^{n+1}} dt \int\limits_{\Upsilon_{ijk}\cap\Omega^{\alpha}(t)} dx\, dy\, dz\, G^{\alpha}. \tag{8}$$

Then, by application of the divergence theorem, we have

$$\begin{aligned}
&\Lambda^{\alpha,n+1} V_{ijk}\bar{U}^{\alpha,n+1}_{ijk} - \Lambda^{\alpha,n} V_{ijk}\bar{U}^{\alpha,n}_{ijk} + A^{\alpha}_{i+\frac{1}{2},j,k}\bar{F}^{x,\alpha}_{i+\frac{1}{2},j,k} - A^{\alpha}_{i-\frac{1}{2},j,k}\bar{F}^{x,\alpha}_{i-\frac{1}{2},j,k}\\
&\quad + A^{\alpha}_{i,j+\frac{1}{2},k}\bar{F}^{y,\alpha}_{i,j+\frac{1}{2},k} - A^{\alpha}_{i,j-\frac{1}{2},k}\bar{F}^{y,\alpha}_{i,j-\frac{1}{2},k} + A^{\alpha}_{i,j,k+\frac{1}{2}}\bar{F}^{z,\alpha}_{i,j,k+\frac{1}{2}} - A^{\alpha}_{i,j,k-\frac{1}{2}}\bar{F}^{z,\alpha}_{i,j,k-\frac{1}{2}}\\
&\quad + A^{B,\alpha}\bar{F}^{B,\alpha}_{ijk} = \Delta t\bar{\Lambda}^{\alpha} V\bar{G}^{\alpha}_{ijk},
\end{aligned} \tag{9}$$

with the notation $\bar{q}$ denoting the average over the appropriate regions in space–time: $\bar{\Lambda}_{ijk}$ is the average over $[t^n, t^{n+1}]$, $\bar{U}^{\alpha}_{ijk}$ the average over $\Upsilon_{ijk} \cap \Omega^{\alpha}(t)$, and $\bar{F}^{x,\alpha}_{i+\frac{1}{2},j,k}$ the average over $\partial\Upsilon_{i+\frac{1}{2},j,k} \cap \Omega^{\alpha}(t)$. Rearranging the terms in (9) leads to a finite-volume discretization of $\nabla \cdot F^{\alpha}$; i.e.,

$$\bar{U}^{\alpha,n+1}_{ijk} = \bar{U}^{\alpha,n}_{ijk} - \Delta t(\nabla \cdot F^{\alpha})^{C} + \Delta t\frac{\bar{\Lambda}^{\alpha}}{\Lambda^{\alpha,n+1}}\bar{G}^{\alpha}_{ijk}, \tag{10}$$

with $(\nabla \cdot F^{\alpha})^{C}$ the conservative flux difference

$$\begin{aligned}
\Delta t(\nabla \cdot F^{\alpha})^{C} = {}& \frac{\Lambda^{\alpha,n+1} - \Lambda^{\alpha,n}}{\Lambda^{\alpha,n+1}}\bar{U}^{\alpha,n}_{ijk} + \frac{1}{\Lambda^{\alpha,n+1}V_{ijk}}\Big(A^{\alpha}_{i+\frac{1}{2},j,k}\bar{F}^{x,\alpha}_{i+\frac{1}{2},j,k} - A^{\alpha}_{i-\frac{1}{2},j,k}\bar{F}^{x,\alpha}_{i-\frac{1}{2},j,k}\\
& + A^{\alpha}_{i,j+\frac{1}{2},k}\bar{F}^{y,\alpha}_{i,j+\frac{1}{2},k} - A^{\alpha}_{i,j-\frac{1}{2},k}\bar{F}^{y,\alpha}_{i,j-\frac{1}{2},k} + A^{\alpha}_{i,j,k+\frac{1}{2}}\bar{F}^{z,\alpha}_{i,j,k+\frac{1}{2}}\\
& - A^{\alpha}_{i,j,k-\frac{1}{2}}\bar{F}^{z,\alpha}_{i,j,k-\frac{1}{2}} + A^{B,\alpha}\bar{F}^{B,\alpha}_{ijk}\Big).
\end{aligned} \tag{11}$$

For the case $G^{\alpha}_{ijk} = 0$, the discretization (10) satisfies a discrete conservation law of the form

$$\sum_{ijk}\bar{\Lambda}^{\alpha,n+1}_{ijk}\bar{U}^{\alpha,n+1}_{ijk} = \sum_{ijk}\bar{\Lambda}^{\alpha,n}_{ijk}\bar{U}^{\alpha,n} + (\text{boundary terms}), \tag{12}$$

where the sum is over an arbitrary subset of the control volumes, and where the boundary terms are a sum of fluxes evaluated at the faces bounding the region corresponding to those control volumes.

There are two issues that must be resolved in order to specify a numerical method based on (10), (11). The first is the choice of quadrature rules with which to replace the averages $\bar{F}, \bar{U}$ with functions of equally spaced grid values. The second issue is that a straightforward explicit discretization typically has a maximum stable time step given by a Courant–Friedrichs–Lewy condition that is no better than $v_{max}\Delta t = O(h\Lambda^{1/3})$ (where v_{max} is the maximum signal speed in that cell) and could be as small as $v_{max}\Delta t = O(h\Lambda)$ for a boundary whose normal is aligned with one of the coordinate directions. To deal with these issues, we take an approach first introduced in [4], following the ideas in [10].

Our primary dependent variables are discretized on the appropriate subsets of the rectangular grid on either side of the front: for all (i, j, k) such that $\Lambda^{\alpha,n}_{ijk} > 0$. $U^{\alpha,n}_{ijk} \approx U(i\Delta x, j\Delta y, k\Delta z, t^n)$, the update of $U^{\alpha,n}_{ijk}$ to time $t^{n+1} = t^n + \Delta t$, is given in the following steps.

1. For all cells for which $\Lambda^{\alpha,n}_{ijk} = 0$, $\Lambda^{\alpha,n+1}_{ijk} > 0$, initialize $U^{\alpha,n}_{ijk}$.
2. For all faces adjacent to cells with $\Lambda^{\alpha,n}_{ijk} \neq 0$ or $\Lambda^{\alpha,n+1}_{ijk} \neq 0$, we compute estimates to the fluxes $F^{x,\alpha}_{i+\frac{1}{2},j,k} \approx F^{x,\alpha}((i + \frac{1}{2})\Delta x, j\Delta y, k\Delta z)$. We use these fluxes to compute $(\nabla \cdot F^{\alpha})^{C}$ by making the substitution $\bar{U}^{n,\alpha}_{ijk} \leftarrow U^{n,\alpha}_{ijk}$, $\bar{F}^{x,\alpha}_{i+\frac{1}{2},j,k} \leftarrow F^{x,\alpha}_{i+\frac{1}{2},j,k}$, in (11). We

also use these fluxes to compute a nonconservative but stable estimate to $\nabla \cdot F^\alpha$:

$$\Delta t(\nabla \cdot F^\alpha)^{NC} = \frac{\Delta t}{\Delta x_i}\left(F^{x,\alpha}_{i+\frac{1}{2},j,k} - F^{x,\alpha}_{i-\frac{1}{2},j,k}\right) + \frac{\Delta t}{\Delta y_j}\left(F^{y,\alpha}_{i,j+\frac{1}{2},k} - F^{y,\alpha}_{i,j-\frac{1}{2},k}\right) + \frac{\Delta t}{\Delta z_k}\left(F^{z,\alpha}_{i,j,k+\frac{1}{2}} - F^{z,\alpha}_{i,j,k-\frac{1}{2}}\right). \tag{13}$$

3. Perform a preliminary stable update to $\bar{U}^\alpha$ using a linear hybridization of $(\nabla \cdot F^\alpha)^C$ and $(\nabla \cdot F^\alpha)^{NC}$:

$$U^{\alpha,n+1,ref}_{ijk} = U^{\alpha,n}_{ijk} - \Delta t\left[\Lambda^{\alpha,n+1}_{ijk}(\nabla \cdot F^\alpha)^C + \left(1 - \Lambda^{\alpha,n+1}_{ijk}\right)(\nabla \cdot F^\alpha)^{NC}\right]. \tag{14}$$

Because the weight multiplying $(\nabla \cdot F^\alpha)^C$ is $\Lambda^{\alpha,n+1}_{ijk}$, the small denominator in (11) is canceled, and we expect that the resulting update formula is stable. However, $U^{\alpha,n+1,ref}_{ijk}$ does not satisfy a conservation relation of the form (12). We can compute δM^α_{ijk}, the difference between the total mass in the control volume given by the preliminary update and that given by a conservative update based on $(\nabla \cdot F^\alpha)^C$ alone:

$$\delta M^\alpha_{ijk} = \left(1 - \Lambda^{\alpha,n+1}_{ijk}\right)\Lambda^{\alpha,n+1}_{ijk}[(\nabla \cdot F^\alpha)^C - (\nabla \cdot F^\alpha)^{NC}]. \tag{15}$$

4. After the preliminary updates for all of the cells have been performed, distribute the mass deficit δM^α_{ijk} onto the grid to maintain conservation:

$$U^{\alpha,n+1}_{ijk} := U^{\alpha,n+1}_{ijk} + \sum_{l=i-1}^{i+1}\sum_{m=j-1}^{j+1}\sum_{p=k-1}^{k+1} \frac{w_{ijk}}{W_{lmp}}\delta M^\alpha_{lmp}. \tag{16}$$

The weights w_{ijk} are assumed to be nonnegative, bounded from above uniformly as $\Lambda \to 0$. The normalization factor W_{ijk} is given by

$$W_{ijk} = \sum_{l=i-1}^{i+1}\sum_{m=j-1}^{j+1}\sum_{p=k-1}^{k+1} \Lambda^{\alpha,n+1}_{lmp} V_{lmp} w_{lmp}. \tag{17}$$

In the present work, we will use mass-weighted distribution, i.e., $w_{ijk} = \rho^{\alpha,n+1,ref}_{ijk}$, the value of the density after the preliminary update. Mass weighting was found to provide more-robust solutions than volume weighting ($w_{ijk} = 1$) in problems involving strong shocks [32].

In the calculation of $(\nabla \cdot F)^C$, we compute approximations to the averages along surfaces and volumes by replacing those averages with estimates of the values evaluated at the centers of the Cartesian cells and faces Υ_{ijk}, $\partial\Upsilon_{i-\frac{1}{2},j,k}$, etc. Since the location of the centroids of the control volumes and apertures differ from the centers of the Cartesian cells and faces by $O(h)$, this leads to a discretization that is formally inconsistent, i.e., that has a truncation error that is $O(1)$ relative to h, at the moving boundary. It is possible to construct a formally consistent method, i.e., one whose truncation error is $O(h)$ [11, 21, 27]. However, the present approach has been used with great success for tracking shocks [4, 15] and fixed solid-wall boundaries [32]. It is also a substantial improvement from a formal standpoint over capturing methods, which have a formal truncation error that is $O(h^{-1})$ [47].

In the remainder of this paper we expand on the ideas introduced here and demonstrate their use with some examples.

In Section 3 we elaborate on geometry issues. Our approach is based on discretizing a transport equation for a volume fraction density for each phase, the averages of which over cells are the volume fractions Λ. The specific form of the transport equation is motivated by previous work in multifluid representations of interfaces [13, 26]; the connection to the mathematics and physics being modeled here is discussed in Section 3.2. Our discretization of the volume fraction transport equation uses local tangent-plane approximations to the interface constructed from the volume fractions Λ^α, as described in Section 3.1. In Section 3.3 we describe how our geometric description of $\delta\Omega^\alpha(t)$ is used to compute apertures A^α.

In Section 4 we describe the computation of fluxes $F(U)$. We describe the computation of extended states $U^{\alpha,ext}$, and high-order Godunov methods for the computation of second-order-accurate fluxes F. Our implementation uses an Eulerian Godunov method for solid mechanics described in [25], and a new non-spatially split implementation of the piecewise parabolic method (PPM) for fluids, described here.

In Section 5 we describe an implementation of the overall method to shock capturing computations with fluids and solids and vacuum. First we specify the variables U^α, flux functions $F^\alpha(U)$, and source terms G^α appropriate to each material phase. We describe the constitutive models (equations of state, chemical reaction modeling, and plastic flow laws) that provide closure conditions for the evolution equations. An example is given of the interface compatibility equation (2): the interface-normal solid–fluid Riemann problem used to construct an interface velocity field to evolve the phase domains $\Omega(t)$. Finally, we present example calculations. One-dimensional solid–fluid and solid–vacuum examples demonstrate the accuracy of the method. A three-dimensional example is also described which includes adaptive mesh refinement (AMR).

In Section 6 we summarize our findings and describe steps one might take to improve the order of accuracy of the method.

3. INTERFACE MOTION AND GEOMETRIC DESCRIPTION

3.1. Volume-of-Fluid Interface Reconstruction

We use the volume fraction Λ^α as the fundamental variable that determines the spatial distribution Ω^α of material phase α. From the volume fractions, a piecewise planar representation of the material interface may be constructed. The reconstructed interface is chosen to satisfy two constraints. First, the calculated planar interface segment in cell ijk breaks the cell into parts with a calculated volume fraction $\Lambda_{ijk}^{\alpha,calc}$ that is exactly equal to the actual value; $\Lambda_{ijk}^{\alpha,calc} = \Lambda_{ijk}^{\alpha}$. Second, the calculated piecewise planar interface is an approximation to the least-squares best fit in the neighborhood of cell ijk. Specifically, we desire an interface $(\vec{n}, d)$,

$$n_x x + n_y y + n_z z = d, \tag{18}$$

with outward-directed normal unit vector $\vec{n}$, $\vec{n} \cdot \vec{n} = 1$, that minimizes the sum of squares error SOS between calculated and input volume fractions on a $5 \times 5 \times 5$ stencil:

$$SOS_{ijk}(\vec{n}_{ijk}, d_{ijk}) = \sum_{l=i-2}^{i+2} \sum_{m=j-2}^{j+2} \sum_{p=k-2}^{k+2} \left(\Lambda_{lmp}^{\alpha,calc}(\vec{n}_{ijk}, d_{ijk}) - \Lambda_{lmp}^{\alpha}\right)^2. \tag{19}$$

In two dimensions, piecewise linear interfaces obeying analogous constraints (on a 3×3 stencil of cells) have been shown to provide second-order accuracy for test problems involving material advection [35].

Instead of solving the constrained least-squares problem directly, we approximate its solution using the ELVIRA algorithm of Pilliod and Puckett [35]. This algorithm determines a finite list of N candidate interface normals $\vec{n}^k$, $k = 1, 2, \ldots, N$. For each normal vector, an intercept d^k is calculated which gives the exact volume fractions for cell ijk. This enforces the first constraint. The sum-of-squares error is then computed for each candidate plane $(\vec{n}^k, d^k)$, and the single plane with the least sum-of-squares error is chosen to represent the interface segment. The algorithm, described below, has the property of calculating exactly the planar interface when the input volume fractions on a 5 × 5 × 5 stencil are consistent with the existence of a single planar interface passing through the center cell.

The 3D ELVIRA algorithm begins by summing the volume fractions in a 3 × 3 × 5 block of cells in the direction z (Fig. 1):

$$v_{lm} = \sum_{p=k-2,k+2} V_{lmp}\Lambda^{\alpha}_{lmp}, \qquad \begin{array}{l} l = i-1, i+1, \\ m = j-1, j+1. \end{array} \tag{20}$$

In this 3 × 3 array, the summed volume fraction v_{lm} is assumed proportional to the height, in the z direction, of the interface at coordinate x_l, y_m; i.e.,

$$\frac{v_{lm}}{\Delta x_l \Delta y_m} = c - \frac{n_x}{n_z}x_l + \frac{n_y}{n_z}y_m, \tag{21}$$

where c is a constant (the indices lm for which this assumption holds best are selected by the ELVIRA algorithm). Thus, for each triplet of index pairs, ratios n_x/n_z and n_y/n_z (and the

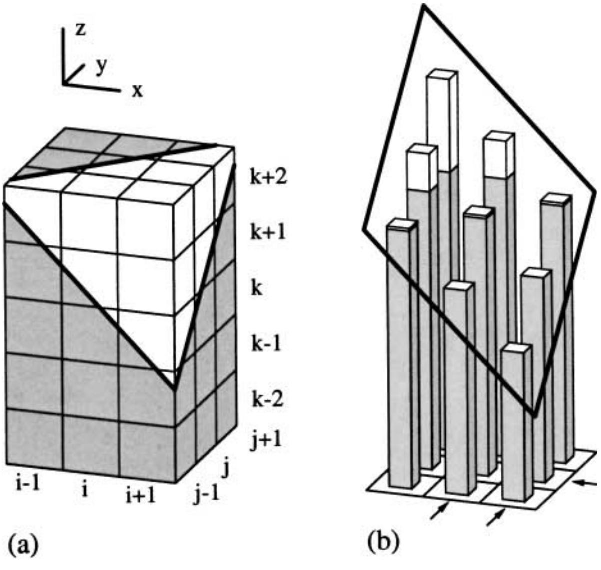

FIG. 1. (a) A 3 × 3 × 5 block of volume fractions (schematically illustrated by shading using a yet-to-be-determined planar interface) are summed columnwise to give a 3 × 3 set of sums v (b) from which noncolinear triplets of points may be used to estimate a reconstructed interface-normal vector. The sums $v_{i,j-1}$, $v_{i+1,j-1}$, and $v_{i+1,j}$ (indicated by arrows) are compatible with the exact plane from which the fractions Λ in this example were derived. The ELVIRA algorithm will select the plane determined from these sums since the corresponding sum-of-squares error will be zero. For each of the remaining sums, the plane crosses the face $k + \frac{5}{2}$, leading to an underestimate of the column volume beneath the plane. Planes determined from these other sums will therefore lead to nonzero sum-of-squares errors.

constant c) may be calculated. Given these ratios, the normal vector $\vec{n} = (n_x, n_y, n_z)$ may be calculated to within a sign factor by applying the normalization condition $\vec{n} \cdot \vec{n} = 1$. The sign may be resolved by choosing the sign of n_z to be consistent with the calculated center of mass [37] (see also [35]). When the z moment of the volume distribution is small, the sign may not be accurately determined by the center-of-mass constraint. In that case, both sign possibilities are tested. The constraint for normal component n_z is considered weak if the absolute value of the z component of the center of mass of the central $3 \times 3 \times 3$ block is less than 0.2 times the width Δz of the central cell.

In projection z, the 24 noncolinear triplets that include the center of the projection (i, j) are tested. This gives rise to between 24 and 48 candidate normal vectors. This same procedure is applied in the x and y directions, to yield a total of 72 to 144 candidate normal vectors.

It may happen that the $5 \times 5 \times 5$ array of volume fractions has a symmetry which does not appear in the vector that provides the lowest sum-of-squares error. We explicitly enforce symmetry by testing the symmetry of the fraction array, and by appropriately symmetrizing the trial normal vectors prior to calculating their sum-of-squares error. For example, on a uniform grid when $\Lambda^{\alpha}_{i+l,j+m,k+p} = \Lambda^{\alpha}_{i+l,j+p,k+m}$, $l, m, p = -2, \ldots, 2$, symmetry dictates $n_y = n_z$. All test vectors $\tilde{n}$ are then modified to obey this symmetry:

$$\begin{pmatrix} n_x \\ n_y, n_z \end{pmatrix} \leftarrow \begin{pmatrix} \tilde{n}_x \\ \operatorname{sign}(\tilde{n}_y + \tilde{n}_z)\sqrt{\frac{1}{2}\left(1 - \tilde{n}_x^2\right)} \end{pmatrix}, \quad \text{if } n_y = n_z \text{ symmetry indicated.} \tag{22}$$

3.2. Interface Advection

Following [13, 26] we evolve material phase fluid fractions by solving the fluid fraction advection equation

$$\frac{\partial}{\partial t}\Lambda^{\alpha} + \frac{\partial}{\partial x_i} \cdot (\Lambda^{\alpha} v_i) = \frac{\Lambda^{\alpha}\bar{M}}{M^{\alpha}} \nabla \cdot v \tag{23a}$$

with

$$\bar{M}_{ijk} = \left(\sum_{\beta} \frac{\Lambda^{\beta}_{ijk}}{M^{\beta}_{ijk}}\right)^{-1}. \tag{23b}$$

v is the velocity field associated with the interface of phase α; its computation is described in Section 5.3.4. M has the meaning of an isentropic modulus of incompressibility; for solids undergoing homogeneous deformation, or for fluids more generally, M is the isentropic bulk modulus. These moduli are discussed further in Section 5.3.5. Because of the term $\bar{M}$, this equation is a collective part of our method (see also Section 5.3).

The above equations are formally equivalent to an advection equation for Λ, since the terms proportional to $\nabla \cdot v$ vanish, and v is continuous at the interface. However, we use the form (23) because discretization errors arising from the volume-of-fluid representation may cause them to be nonzero. In the area of multifluid calculations, the use of (23) to compute the effective dynamics is controversial, since the underlying physical assumptions of pressure matching at the interface and changes of state being isentropic are violated when a shock intersects the interface. In the present case, we use a formulation based on those physical assumptions to be consistent with the rest of the algorithm, which is based on the same assumptions. In practice, the region where the assumptions are violated is a set of codimension 2 or smaller, and it has been observed in tracking other discontinuities using this approach to be correctly treated by virtue of conservation.

We discretize Eq. (23a) by first solving the left hand side in conservation form,

$$\tilde{\Lambda}^{\alpha}_{ijk} = \frac{V_{ijk}\Lambda^{\alpha,n}_{ijk} + \mathcal{V}^{\alpha}_{i-\frac{1}{2},j,k} - \mathcal{V}^{\alpha}_{i+\frac{1}{2},j,k} + \mathcal{V}^{\alpha}_{i,j-\frac{1}{2},k} - \mathcal{V}^{\alpha}_{i,j+\frac{1}{2},k} + \mathcal{V}^{\alpha}_{i,j,k-\frac{1}{2}} - \mathcal{V}^{\alpha}_{i,j,k+\frac{1}{2}}}{V_{ijk} + \mathcal{V}^{tot}_{i-\frac{1}{2},j,k} - \mathcal{V}^{tot}_{i+\frac{1}{2},j,k} + \mathcal{V}^{tot}_{i,j-\frac{1}{2},k} - \mathcal{V}^{tot}_{i,j+\frac{1}{2},k} + \mathcal{V}^{tot}_{i,j,k-\frac{1}{2}} - \mathcal{V}^{tot}_{i,j,k+\frac{1}{2}}}. \tag{24}$$

Here, $\mathcal{V}^{\alpha}_{i-\frac{1}{2},j,k}$ is the volume of phase α advected across cell face $i-\frac{1}{2}$, j, k in the time interval Δt, and $\mathcal{V}^{tot}_{i-\frac{1}{2},j,k}$ is the total material volume advected across that face. The calculation of these advected volumes is described below.

The volume fractions $\tilde{\Lambda}^{\alpha}$ computed from the left hand side of Eq. (23a) do not in general sum to one: $\sum_{\alpha}\tilde{\Lambda}^{\alpha} \neq 1$. In fact,

$$\Delta t(\nabla\cdot v) = 1 - \sum_{\alpha}\tilde{\Lambda}^{\alpha}. \tag{25}$$

The final solution to Eq. (23a) is then given by

$$\Lambda^{\alpha,n+1}_{ijk} = \tilde{\Lambda}^{\alpha}_{ijk} + \frac{\tilde{\Lambda}^{\alpha}_{ijk}\bar{M}_{ijk}}{M^{\alpha}_{ijk}}\left(1 - \sum_{\beta}\tilde{\Lambda}^{\beta}_{ijk}\right), \tag{26}$$

with $\bar{M}$ given by (23b).

We floor these calculated fractions to avoid smearing the interface

$$\Lambda^{\alpha,n+1}_{ijk} \leftarrow \begin{cases} 0 & \text{if } \Lambda^{\alpha,n+1}_{i,j,k} \leq \epsilon, \\ 1 & \text{if } \Lambda^{\alpha,n+1}_{i,j,k} \geq 1-\epsilon, \\ \Lambda^{1,n+1}_{ijk} & \text{otherwise,} \end{cases} \tag{27}$$

with $\epsilon = 10^{-7}$ in our test problems.

The volume fluxes $\mathcal{V}$ used in Eq. (24) are calculated using a non-spatially split advection algorithm developed by Pilliod and Puckett [35] based on the characteristic tracing approach of Bell *et al.* [5]. This method uses a face-centered velocity field, which we construct as follows.

We describe a method in Section 5.3 for the calculation of a cell-centered interface velocity field. From this cell-centered interface velocity field we construct face-centered values by averaging,

$$\begin{aligned} v_{x,i-\frac{1}{2},j,k} &= \frac{1}{2}(v_{x,i-1,j,k} + v_{x,i,j,k}), \\ v_{y,i,j-\frac{1}{2},k} &= \frac{1}{2}(v_{y,i,j-1,k} + v_{y,i,j,k}), \\ v_{z,i,j,k-\frac{1}{2}} &= \frac{1}{2}(v_{z,i,j,k-1} + v_{z,i,j,k}). \end{aligned} \tag{28}$$

Following Bell *et al.* we write the conservation law for the left hand side of Eq. (23a) as

$$\frac{\partial\Lambda^{\alpha}}{\partial t} + v_x\frac{\partial\Lambda^{\alpha}}{\partial x} + \frac{\partial(\Lambda^{\alpha}v_y)}{\partial y} + \frac{\partial(\Lambda^{\alpha}v_z)}{\partial z} + \Lambda^{\alpha}\frac{\partial v_x}{\partial x} = 0. \tag{29}$$

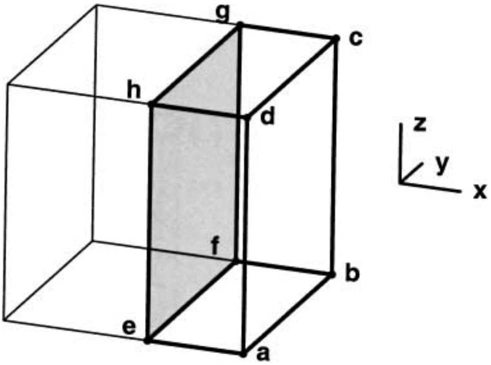

FIG. 2. The volume fluxes through the $i+\frac{1}{2}$ face (rectangle abcd) of cell ijk are given by the sum of (i) the volume abcdefgh, where plane efgh is displaced from abcd by $-(v_x)_{i+\frac{1}{2},j,k}\Delta t$; (ii) a correction to (i) due to v_x having a nonzero gradient dv_x/dx; and (iii) the transverse volume fluxes through rectangles abfe, bcgf, cdhg, and adhe.

Taking the velocity component v_x to be constant, and integrating by parts over the space–time volume traced by characteristics passing through face $i+\frac{1}{2}, j, k$, we have, assuming $v_{x,i+\frac{1}{2},j,k} \geq 0$ for this example (see Fig. 2),

$$\mathcal{V}^{\alpha}_{i+\frac{1}{2},j,k} = \int_{\mathtt{abcdefgh}} \Lambda^{\alpha} dV + \mathcal{V}^{\alpha}_{\mathtt{adeh}} + \mathcal{V}^{\alpha}_{\mathtt{abef}} - \mathcal{V}^{\alpha}_{\mathtt{bcfg}} - \mathcal{V}^{\alpha}_{\mathtt{cdgh}} - \frac{\Delta t}{2}\frac{\partial v_x}{\partial x}\int_{\mathtt{abcdefgh}} \Lambda^{\alpha} dV, \tag{30a}$$

$$\mathcal{V}^{tot}_{i+\frac{1}{2},j,k} = \int_{\mathtt{abcdefgh}} dV + \mathcal{V}^{tot}_{\mathtt{adeh}} + \mathcal{V}^{tot}_{\mathtt{abef}} - \mathcal{V}^{tot}_{\mathtt{bcfg}} - \mathcal{V}^{tot}_{\mathtt{cdgh}} - \frac{\Delta t}{2}\frac{\partial v_x}{\partial x}\int_{\mathtt{abcdefgh}} dV, \tag{30b}$$

where

$$\frac{\partial v_x}{\partial x} = \frac{v_{x,i+\frac{1}{2},j,k} - v_{x,i-\frac{1}{2},j,k}}{\Delta x_i}, \tag{31}$$

and where

$$\mathtt{a} = \left(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}\right), \tag{32a}$$

$$\mathtt{e} = \left(x_{i+\frac{1}{2}}, -v_{x,i+\frac{1}{2},j,k}\Delta t, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}\right), \tag{32b}$$

etc. In Eq. (30), the integral $\int \Lambda^{\alpha} dV$ is the part of volume abcdefgh occupied by phase α. This is calculated with elementary geometry using the coordinates abcdefgh and the time t^n planar interface $(\vec{n}, d)$, if one exists.

To evaluate the transverse flux $\mathcal{V}_{\mathtt{adeh}}$ appearing in (30) (for example), we write Eq. (23a) as

$$\frac{\partial \Lambda^{\alpha}}{\partial t} + v_x\frac{\partial \Lambda^{\alpha}}{\partial x} + v_y\frac{\partial \Lambda^{\alpha}}{\partial y} + \frac{\partial(\Lambda^{\alpha} v_z)}{\partial z} + \Lambda^{\alpha}\frac{\partial v_x}{\partial x} + \Lambda^{\alpha}\frac{\partial v_y}{\partial y} = 0. \tag{33}$$

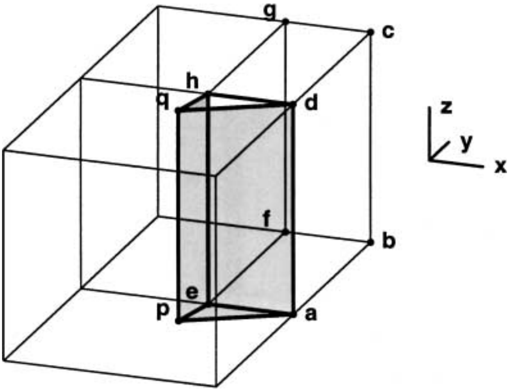

FIG. 3. The volume fluxes through rectangle $\mathtt{adhe}$ contributing to the volume fluxes through face $i+\frac{1}{2}$ of cell ijk are given by the sum of (i) the volume of the prism $\mathtt{adhepq}$; (ii) a correction due to dv_x/dx and dv_y/dy being nonzero; and (iii) the transverse volume fluxes through the triangles $\mathtt{dhq}$ and $\mathtt{aep}$.

Now, holding velocities v_x and v_y constant, and integrating by parts (assuming $v_{y,i,j-\frac{1}{2},k} > 0$ in this example; see Fig. 3),

$$\mathcal{V}^{\alpha}_{\mathtt{adeh}} = \int_{\mathtt{adehpq}} \Lambda^{\alpha} dV + \mathcal{V}^{\alpha}_{\mathtt{aep}} - \mathcal{V}^{\alpha}_{\mathtt{dhq}} - \frac{\Delta t}{3}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \int_{\mathtt{adehpq}} \Lambda^{\alpha} dV, \tag{34a}$$

$$\mathcal{V}^{tot}_{\mathtt{adeh}} = \int_{\mathtt{adehpq}} dV + \mathcal{V}^{tot}_{\mathtt{aep}} - \mathcal{V}^{tot}_{\mathtt{dhq}} - \frac{\Delta t}{3}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \int_{\mathtt{adehpq}} dV, \tag{34b}$$

where

$$\frac{\partial v_x}{\partial x} = \frac{v_{x,i+\frac{1}{2},j-1,k} - v_{x,i-\frac{1}{2},j-1,k}}{\Delta x_i}, \tag{35a}$$

$$\frac{\partial v_y}{\partial y} = \frac{v_{y,i,j-\frac{1}{2},k} - v_{y,i,j-\frac{3}{2},k}}{\Delta y_{j-i}}, \tag{35b}$$

and where

$$\mathtt{p} = \begin{cases} \left(x_{i+\frac{1}{2}} - \Delta t v_{x,i+\frac{1}{2},j-1,k},\, y_{j-\frac{1}{2}} - \Delta t v_{y,i,j-\frac{1}{2},k},\, z_{k-\frac{1}{2}}\right) & \text{if } v_{x,i+\frac{1}{2},j-1,k} v_{x,i+\frac{1}{2},j,k} > 0, \\ \left(x_{i+\frac{1}{2}},\, y_{j-\frac{1}{2}} - \Delta t v_{y,i,j-\frac{1}{2},k},\, z_{k-\frac{1}{2}}\right) & \text{otherwise,} \end{cases} \tag{36}$$

and similarly for point $\mathtt{q}$. Note that in (36) we limit the velocity v_x at face $i+\frac{1}{2}, j-1, k-1$ to ensure that its sign is the same as at face $i+\frac{1}{2}, j, k$.

To determine the transverse volume flux $\mathcal{V}_{\mathtt{aep}}$, appearing in (34), write Eq. (23a) as

$$\frac{\partial \Lambda^{\alpha}}{\partial t} + v_x \frac{\partial \Lambda^{\alpha}}{\partial x} + v_y \frac{\partial \Lambda^{\alpha}}{\partial y} + v_z \frac{\partial \Lambda^{\alpha}}{\partial z} + \Lambda^{\alpha} \frac{\partial v_x}{\partial x} + \Lambda^{\alpha} \frac{\partial v_y}{\partial y} + \Lambda^{\alpha} \frac{\partial v_z}{\partial z} = 0. \tag{37}$$

Holding v_x, v_y, and v_z constant, and integrating by parts (assuming $v_{z,i,j-1,k-\frac{1}{2}} > 0$ for this example; see Fig. 4),

$$\mathcal{V}^{\alpha}_{\mathtt{aep}} = \int_{\mathtt{aepr}} \Lambda^{\alpha} dV - \frac{\Delta t}{4}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right) \int_{\mathtt{aepr}} \Lambda^{\alpha} dV, \tag{38a}$$

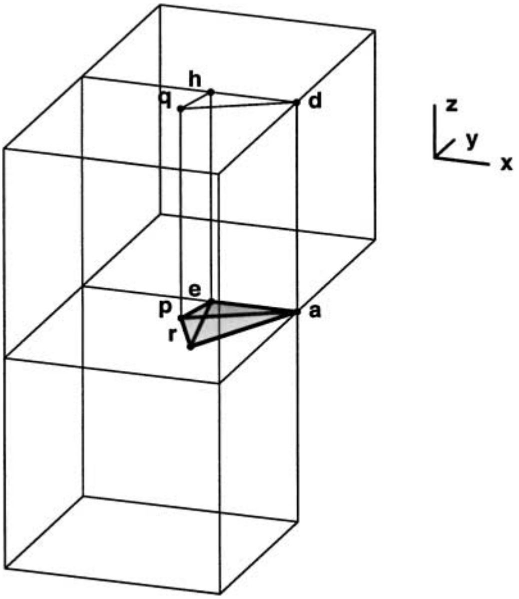

FIG. 4. The volume fluxes through triangle aep contributing to the volume fluxes through face $i + \frac{1}{2}$ of cell ijk are given by the volume of the tetrahedron aepr and a correction due to dv_x/dx, dv_y/dy, and dv_z/dz being nonzero.

$$\mathcal{V}^{tot}_{\mathtt{aep}} = \int_{\mathtt{aepr}} dV - \frac{\Delta t}{4}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right)\int_{\mathtt{aepr}} dV, \tag{38b}$$

where

$$\frac{\partial v_x}{\partial x} = \frac{v_{x,i+\frac{1}{2},j-1,k-1} - v_{x,i-\frac{1}{2},j-1,k-1}}{\Delta x_i}, \tag{39a}$$

$$\frac{\partial v_y}{\partial y} = \frac{v_{y,i,j-\frac{1}{2},k-1} - v_{y,i,j-\frac{3}{2},k-1}}{\Delta y_{j-i}}, \tag{39b}$$

$$\frac{\partial v_z}{\partial z} = \frac{v_{z,i,j-1,k-\frac{1}{2}} - v_{z,i,j-1,k-\frac{3}{2}}}{\Delta z_{k-i}}, \tag{39c}$$

and where

$$\mathtt{r}_x = \begin{cases} x_{i+\frac{1}{2}} - \Delta t v_{x,i+\frac{1}{2},j-1,k-1} & \text{if } v_{x,i+\frac{1}{2},j-1,k-1} v_{x,i+\frac{1}{2},j,k} > 0, \\ x_{i+\frac{1}{2}} & \text{otherwise,} \end{cases} \tag{40a}$$

$$\mathtt{r}_y = \begin{cases} y_{j-\frac{1}{2}} - \Delta t v_{y,i,j-\frac{1}{2},k-1} & \text{if } v_{y,i,j-\frac{1}{2},k-1} v_{y,i,j-\frac{1}{2},k} > 0, \\ y_{j-\frac{1}{2}} & \text{otherwise,} \end{cases} \tag{40b}$$

$$\mathtt{r}_z = z_{k-\frac{1}{2}} - \Delta t v_{z,i,j-1,k-\frac{1}{2}}. \tag{40c}$$

Figure 5 demonstrates the fidelity of this volume advection scheme using the test problem of [37]. A notched block is rotated with a constant prescribed rotational velocity field. The prescribed velocity is calculated explicitly at each cell face, circumventing step (28). The computational domain is 100 × 100 × 100, with 10 computational cells per unit distance.

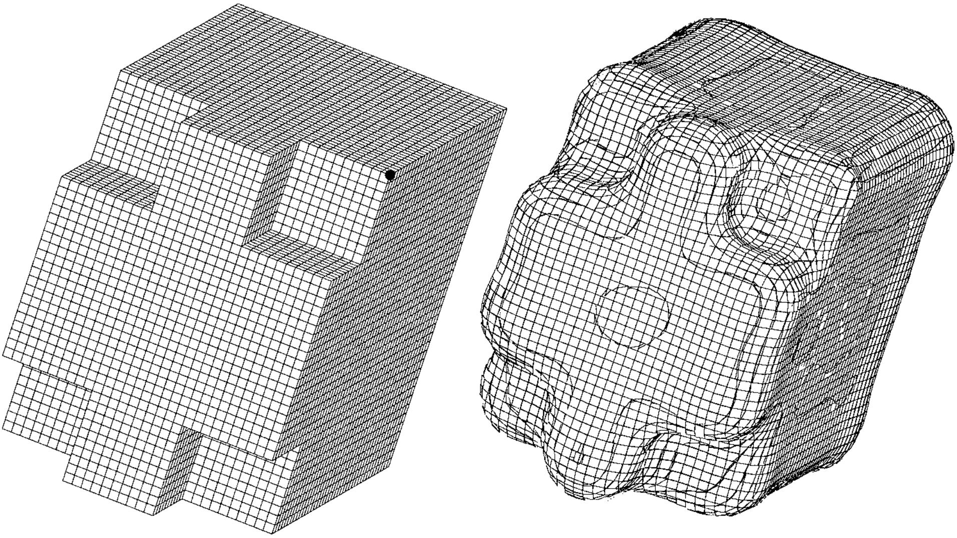

FIG. 5. A notched block of size 3 × 4 × 5 with four corner notches, 1 × 1 × 1 each, taken off the $+z$ side, before and after rotation through 2π radians in 400 time steps. The figures display the outline of each computational cell, truncated when interfaces are present by the piecewise planar reconstructed interface—not a constant volume fraction isosurface. The axis of rotation passes through the dot and the diagonally opposite corner; rotation is clockwise in this orientation.

Figure 5 should be directly compared with Fig. 3 of [37] (see also Figs. 16 and 18 of [22]). Comparisons in two dimensions, with convergence results demonstrating second-order accuracy, appear in [35].

In two-material problems, only one material phase volume fraction (say Λ^0) need be advanced: at each step of the algorithm where we compute volume fluxes $\mathcal{V}^0$ and $\mathcal{V}^{tot}$, we have trivially $\mathcal{V}^1 = \mathcal{V}^{tot} - \mathcal{V}^0$. For two-material problems, if one then used instead of (23a) the simpler equation

$$\frac{\partial}{\partial t}\Lambda^\alpha + \frac{\partial}{\partial x_i} \cdot (\Lambda^\alpha v_i) = \Lambda^\alpha \nabla \cdot v, \tag{41}$$

equivalent to M^α being equal for all phases, for example, then one might thereby circumvent the collective nature of this algorithm. However, for three-material or more-general N-material problems, the condition $\sum_\alpha \Lambda^\alpha = 1$ is not automatically satisfied. This is because our interface model assumes a single piecewise planar interface in each computational cell. When more than two phases meet in a single cell, this assumption cannot be valid. Also, different interfaces will have different velocity fields. The term $\nabla \cdot v$ appearing in (23a) cannot therefore be deduced from the computed $\mathcal{V}^{tot}$ obtained in advecting any single phase α, but must instead be obtained by separately advecting each of the N material phase volume fractions and renormalizing them collectively (via the RHS of (23a) or some other expedient).

This incompatibility between three-material junctions and piecewise planar interface representations leads to other ambiguities within our overall method. In particular, it necessitates a renormalization of apertures (Section 3.3) that is not necessary where only two phases meet. We note that two-material interfaces form a manifold of codimension 1, and three-material interfaces form one of codimension 2. Therefore, the overwhelming majority of our problem domain is free of these complications. According to the truncation error analysis in [11, p. 165], truncation errors as large as $\mathcal{O}(h^{-1})$ associated with the *ad hoc* treatment of codimension 2 states still lead to solutions that converge in L_1, provided the method remains stable.

3.3. Space–Time Apertures

A space–time aperture A^α is the time-integrated area though which fluxes of material phase α may pass unobstructed through the cell face. Interior to a single-phase domain, fluxes are unobstructed and, for example, $A^\alpha_{i+\frac{1}{2},j,k} = \Delta t \Delta y_j \Delta z_k$. In general, the sum over phases of the apertures on a given face is Δt times the area of the face: e.g., in a two-material system, $A^0_{i+\frac{1}{2},j,k} + A^1_{i+\frac{1}{2},j,k} = \Delta t \Delta y_j \Delta z_k$. An exception to this rule occurs when the interface is stationary and coincident with the cell face. Then, $A^0_{i+\frac{1}{2},j,k} = A^1_{i+\frac{1}{2},j,k} = 0$: neither phase's flux crosses the interface.

We calculate the apertures using cell-centered time t^n and time t^{n+1} reconstructed interfaces. Since the apertures are face centered, we average the apertures calculated from left- and right-cell data, e.g.,

$$A_{i+\frac{1}{2},j,k} = \frac{1}{2}\left(A_{L,i+\frac{1}{2},j,k} + A_{R,i+\frac{1}{2},j,k}\right). \tag{42}$$

Figure 6 shows a mixed cell containing a time t^n and t^{n+1} reconstructed interface intersecting the face $i + \frac{1}{2}, j, k$. The volume beneath these interfaces contains phase α (the interface normal is outward directed by convention). In our implementation, there is no constraint placed on the interface normals; thus the interfaces differ by translation as well as rotation. We assume, given

$$n^n_x x + n^n_y y + n^n_z z = d^n \qquad \text{at } t^n \quad \text{and} \tag{43a}$$

$$n^{n+1}_x x + n^{n+1}_y y + n^{n+1}_z z = d^{n+1} \quad \text{at } t^{n+1}, \tag{43b}$$

that

$$n_x(t)x + n_y(t)y + n_z(t)z = d(t), \tag{44}$$

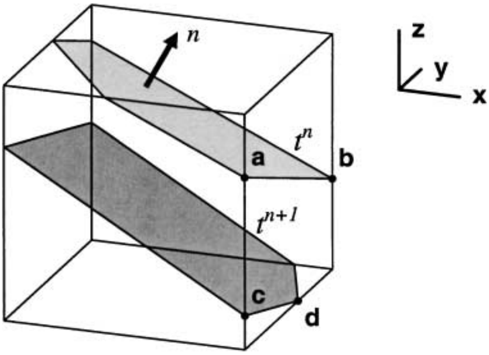

FIG. 6. Time t^n and time t^{n+1} piecewise planar interfaces are shown in fractional cell i, j, k. The aperture $A_{i+\frac{1}{2},j,k}$ is given by the integral over Δt of area on face $i + \frac{1}{2}, j, k$ accessible to the fluid.

with

$$n_x(t) = n_x^n + \left(n_x^{n+1} - n_x^n\right)\tau(t), \tag{45a}$$

$$n_y(t) = n_y^n + \left(n_y^{n+1} - n_y^n\right)\tau(t), \tag{45b}$$

$$n_z(t) = n_z^n + \left(n_z^{n+1} - n_z^n\right)\tau(t), \tag{45c}$$

$$d(t) = d^n + (d^{n+1} - d^n)\tau(t), \tag{45d}$$

$$\tau(t) = \frac{t - t^n}{t^{n+1} - t^n}. \tag{45e}$$

With this linear interpolation algorithm the vector $\vec{n}(t)$ is not, in general, normalized.

In the time interval $[t^n, t^{n+1}]$ the shape of the interpolated interface (44) contained within cell ijk may change topology. These changes occur when the interface crosses one of the eight corners of the cell. We calculate these intersections, sort them in time, and estimate the time integral of the area beneath the interfaces using the trapezoidal sum rule.

For example, the t^n and t^{n+1} interfaces of Fig. 6 give an interpolated interface whose intersection with face $i + \frac{1}{2}, j, k$ defines the (curved) surface $\mathtt{abfdce}$ (Fig. 7) in the space yzt. The volume $\mathtt{aa'\,bb'\,cc'\,dee'\,f}$ is the time-integrated area beneath the interfaces on the $+x$ side of cell ijk. At time $t_{\mathtt{f}}$ the interpolated interface crosses the corner $i + \frac{1}{2}, j + \frac{1}{2}, k - \frac{1}{2}$, and the topology of the interface changes. We approximate volume $\mathtt{aa'\,bb'\,cc'\,dee'\,f}$ as

$$\begin{aligned} A_{L,i+\frac{1}{2},j,k} &\approx (t_{\mathtt{f}} - t^n)\frac{(\text{area } \mathtt{aa'bb'} + \text{area } \mathtt{ee'f})}{2} \\ &\quad + (t^{n+1} - t_{\mathtt{f}})\frac{(\text{area } \mathtt{ee'f} + \text{area } \mathtt{cc'd})}{2}. \end{aligned} \tag{46}$$

The circumstance occurs that a t^n or a t^{n+1} interface exists in cell ijk, but not both. In this case, the missing interface is estimated. We take the existing interface and translate it along its normal vector in time using the time t^n extended interface velocity field.

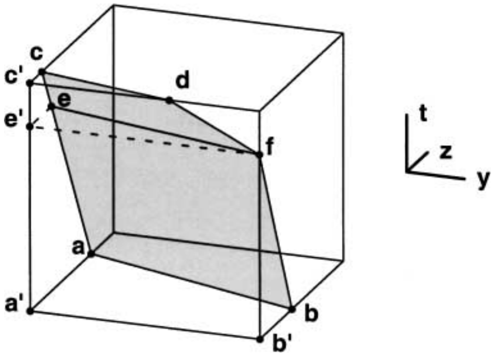

FIG. 7. Aperture calculation shown in *yzt* space. A surface is constructed by interpolation between the time t^n interface $\mathtt{ab}$ and the t^{n+1} interface $\mathtt{cd}$. The interpolation function assumes that all coefficients of the plane (n_x, n_y, n_z, d) are linear in time. The surface may therefore be twisted. The interpolated function changes topology at $\mathtt{ef}$.

Pember *et al.* [32] note that the interface aperture A^B may be computed from the cell-face apertures through a divergence theorem trick; i.e.,

$$\begin{aligned} \tilde{n}_x A^{B,\alpha} &= A^{\alpha}_{i-\frac{1}{2},j,k} - A^{\alpha}_{i+\frac{1}{2},j,k}, \\ \tilde{n}_y A^{B,\alpha} &= A^{\alpha}_{i,j-\frac{1}{2},k} - A^{\alpha}_{i,j+\frac{1}{2},k}, \\ \tilde{n}_z A^{B,\alpha} &= A^{\alpha}_{i,j,k-\frac{1}{2}} - A^{\alpha}_{i,j,k+\frac{1}{2}}, \end{aligned} \tag{47}$$

from which, with $\tilde{n} \cdot \tilde{n} = 1$, one may compute $\tilde{n}$ and A^B.

4. FLUX CALCULATION

4.1. Single-Phase Extended States

It is necessary to extend data for phase α into the two-cell border of cells $\Lambda^{\alpha} = 0$ surrounding the region where phase α exists (where $\Lambda^{\alpha} \neq 0$). This is because the first border of cells $\Lambda^{\alpha,n} = 0$ may be occupied at the next time step, $\Lambda^{\alpha,n+1} \neq 0$, and fluxes must be calculated on all faces of this cell. At a minimum, we assume that flux calculation requires some estimate of cell-centered densities on both sides of the face.

We perform this extension by some averaging scheme. We then modify the momentum and energy values of the extended cells to incorporate some information about the interface velocity field. Our approach gives extended state descriptions which are accurate to $\mathcal{O}(h)$. To achieve higher order accuracy, extrapolation is required [27].

Extension by averaging was suggested by [4]. This extension algorithm, implemented in two passes, is as follows. We use a volume-weighted average in the first pass,

$$U^{\alpha,n,ext}_{ijk} = \frac{\sum_{l=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{p=k-1}^{k+1} V_{lmp} \Lambda^{\alpha,n}_{lmp} U^{\alpha,n}_{lmp}}{\sum_{l=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{p=k-1}^{k+1} V_{lmp} \Lambda^{\alpha,n}_{lmp}}, \tag{48}$$

to extend data into the first border of boundary cells. In the second pass we use simple averaging,

$$U^{\alpha,n,ext}_{ijk} = \frac{\sum_{l=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{p=k-1}^{k+1} V_{lmp} U^{\alpha,n,ext}_{lmp}}{\sum_{l=i-1}^{i+1} \sum_{m=j-1}^{j+1} \sum_{p=k-1}^{k+1} V_{lmp}}. \tag{49}$$

Pilliod [33] introduced a different extension scheme, which he advocated for interface velocity extension (Fig. 8). We optionally use this scheme for vector quantities and the components of the tensors (Fig. 9). Pilliod's scheme for vectors $\vec{q}$ extends separately the components q_x, q_y, and q_z and also extends the magnitude $|\vec{q}|$. After extension of these four quantities by method (48), one then renormalizes the extended vector $(q_x^{ext}, q_y^{ext}, q_z^{ext})$ so that its length is given by the value $(|\vec{q}|)^{ext}$ already obtained by extrapolation. This renormalization procedure is conducted twice: once after executing method (48), and again after method (49). To apply Pilliod's extension method to tensors, we treat each row as an independent vector.

Another algorithmic choice is whether to extend in primitive or conserved variables (as indicated in (48) and (49)). In the examples described later, we use conserved variables and Pilliod's method.

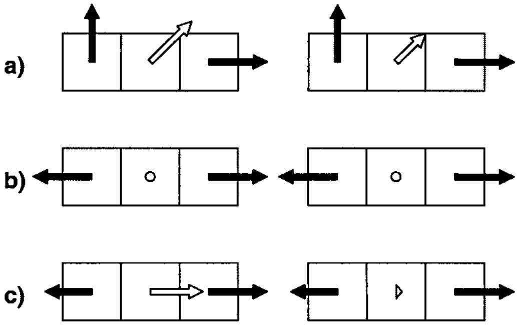

FIG. 8. Vector averages by Pilliod's scheme (first column of figures), and straightforward componentwise averaging (second column) for three cases.

We found in certain stiff modeling experiments (solid and gas) that the robustness of our method was improved by modifying the extended states to incorporate some information regarding interface velocity. The difference between the interface-normal velocity and the material velocity is given by the vector

$$\Delta v = [n \cdot (v_{interface} - v_{material})]\vec{n}. \tag{50}$$

We use this vector difference to modify the extended state velocity according to the prescription

$$v_{material}^{extended} \leftarrow v_{material}^{extended} + \overline{\Delta v}, \tag{51}$$

where the $\overline{\Delta v}$ is the Pilliod average of Δv over neighboring cells that contain an interface:

$$\widetilde{\Delta v} = \frac{1}{|nbh|} \sum_{nbh} \Delta v, \tag{52a}$$

$$|\widetilde{\Delta v}| = \frac{1}{|nbh|} \sum_{nbh} (\Delta v \cdot \Delta v)^{1/2}, \tag{52b}$$

$$\overline{\Delta v} = \frac{|\widetilde{\Delta v}|}{(\widetilde{\Delta v} \cdot \widetilde{\Delta v})^{1/2}} \widetilde{\Delta v}. \tag{52c}$$

For the set of cells immediately adjacent to cells containing valid data, *nbh* is the set of 27 nearest-neighbor cells. For the cells one removed, *nbh* is the set of 125 nearest and second-nearest-neighbor cells.

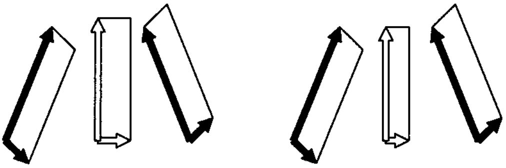

FIG. 9. Tensor averages by Pilliod's scheme (first set), and straightforward componentwise averaging (second set).

4.2. Single-Phase Integration

Each single-phase region is given a boundary of "ghost cell" data, determined from a boundary condition, or obtained by extension (Section 4.1). The single-phase regions are then advanced in time with high-order Godunov methods. The approach we use for solid mechanics is basically identical with the approach described in [25]. For fluid mechanics, we use a variation of the piecewise parabolic method (PPM) [16]. Our PPM implementation differs from the original in two essential ways. First, our implementation is 3D without use of spatial operator splitting. Instead, we use the high-order non-split method of [10, 40]. Second, the limiters we use are different than in [16]. We use simpler methods and avoid altogether the "discontinuity detection" algorithm.

We present below our unsplit-PPM implementation. Emphasis is placed on elucidating the differences between our PPM algorithm and the corresponding solid mechanics solver [25], which is based on piecewise linear reconstructions. We describe this integrator without reference to material boundaries. It should be understood that the update is performed wherever $\Lambda^{\alpha,n} \neq 0$, and also in the surrounding cells where $\Lambda^{\alpha,n} = 0$ but possibly $\Lambda^{\alpha,n+1} \neq 0$. Thus the data U^{α} must be extended, $U^{\alpha,ext}$. For clarity we will omit both the "α" phase designation and the "*ext*" extended-state designation.

In the interior of a single-phase region, the fluid update is constructed using the unsplit-PPM approach, as described below. When extended-state information enters the computation, we limit the method locally to be first-order accurate (thus reference to data three or more cells removed from the interface is circumvented). The solid mechanics solver is similarly modified for this multiphase application.

4.2.1. *Unsplit-PPM Conservative Update*

We use the method of [10, 40] to solve (1) without the use of spatial operator splitting. Use of this technique is motivated by the observation that with operator splitting, one tends to observe grid-oriented artifacts. For instance, an expanding circle in 2D will develop an octagonal profile with split spatial operators. Full corner coupling introduced via the approaches of [10, 40] significantly mitigates that effect.

Discretized on a rectangular structured grid, the equations of fluid dynamics (1) are solved using

$$U_{ijk}^{n+1} = U_{ijk}^{n} - \frac{\Delta t}{\Delta x}\left[F_{i+\frac{1}{2},j,k}^{x,n+\frac{1}{2}} - F_{i-\frac{1}{2},j,k}^{x,n+\frac{1}{2}}\right] - \frac{\Delta t}{\Delta y}\left[F_{i,j+\frac{1}{2},k}^{y,n+\frac{1}{2}} - F_{i,j-\frac{1}{2},k}^{y,n+\frac{1}{2}}\right] - \frac{\Delta t}{\Delta z}\left[F_{i,j,k+\frac{1}{2}}^{z,n+\frac{1}{2}} - F_{i,j,k-\frac{1}{2}}^{z,n+\frac{1}{2}}\right] + \Delta t G^{n+\frac{1}{2}}, \tag{53}$$

where (for example) $F^{x,n+\frac{1}{2}}$ is a flux centered in on the space–time centroid of the x side of a computational cell. These fluxes are obtained using a second-order corner-coupled advection scheme [10, 40]. Schematically, we obtain these fluxes by solution of Riemann problems $\mathcal{R}(q_L, q_R)$ using particular left and right states q_L and q_R. Here, q denotes a vector of primitive variables. A mapping between primitive and conserved variables is understood: $q \leftrightarrow U$. We approximate the solution to fluid-phase Riemann problems using the method of [12].

The first step in the non-operator-split approach is a predictor flux using left and right states obtained (as described in Section 4.2.2) using the PPM reconstruction with upwind characteristic tracing:

$$
\begin{aligned}
\tilde{F}^{x}_{i+\frac{1}{2},j,k} &= F^{x}\left(\mathcal{R}\left(q^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k}, q^{n+\frac{1}{2}}_{xR,i+\frac{1}{2},j,k}\right)\right),\\
\tilde{F}^{y}_{i,j+\frac{1}{2},k} &= F^{y}\left(\mathcal{R}\left(q^{n+\frac{1}{2}}_{yL,i,j+\frac{1}{2},k}, q^{n+\frac{1}{2}}_{yR,i,j+\frac{1}{2},k}\right)\right),\\
\tilde{F}^{z}_{i,j,k+\frac{1}{2}} &= F^{z}\left(\mathcal{R}\left(q^{n+\frac{1}{2}}_{zL,i,j,k+\frac{1}{2}}, q^{n+\frac{1}{2}}_{zR,i,j,k+\frac{1}{2}}\right)\right).
\end{aligned}
\tag{54}
$$

The second step in 3D is the construction of six secondary predictors. Here, for example, $\tilde{F}^{x|y}_{i+\frac{1}{2},j,k}$ has the meaning of “an x-directed flux with L/R states affected by the y-direction predictor flux”; i.e.,

$$
\begin{aligned}
\tilde{F}^{x|y}_{i+\frac{1}{2},j,k} &= F^{x}\left(\mathcal{R}\left(q^{\prime(y)^{n+\frac{1}{2}}}_{xL,i+\frac{1}{2},j,k}, q^{\prime(y)^{n+\frac{1}{2}}}_{xR,i+\frac{1}{2},j,k}\right)\right),\\
\tilde{F}^{x|z}_{i+\frac{1}{2},j,k} &= F^{x}\left(\mathcal{R}\left(q^{\prime(z)^{n+\frac{1}{2}}}_{xL,i+\frac{1}{2},j,k}, q^{\prime(z)^{n+\frac{1}{2}}}_{xR,i+\frac{1}{2},j,k}\right)\right),\\
\tilde{F}^{y|x}_{i,j+\frac{1}{2},k} &= F^{y}\left(\mathcal{R}\left(q^{\prime(x)^{n+\frac{1}{2}}}_{yL,i,j+\frac{1}{2},k}, q^{\prime(x)^{n+\frac{1}{2}}}_{yR,i,j+\frac{1}{2},k}\right)\right),\\
\tilde{F}^{y|z}_{i,j+\frac{1}{2},k} &= F^{y}\left(\mathcal{R}\left(q^{\prime(z)^{n+\frac{1}{2}}}_{yL,i,j+\frac{1}{2},k}, q^{\prime(z)^{n+\frac{1}{2}}}_{yR,i,j+\frac{1}{2},k}\right)\right),\\
\tilde{F}^{z|x}_{i,j,k+\frac{1}{2}} &= F^{z}\left(\mathcal{R}\left(q^{\prime(x)^{n+\frac{1}{2}}}_{zL,i,j,k+\frac{1}{2}}, q^{\prime(x)^{n+\frac{1}{2}}}_{zR,i,j,k+\frac{1}{2}}\right)\right),\\
\tilde{F}^{z|y}_{i,j,k+\frac{1}{2}} &= F^{z}\left(\mathcal{R}\left(q^{\prime(y)^{n+\frac{1}{2}}}_{zL,i,j,k+\frac{1}{2}}, q^{\prime(y)^{n+\frac{1}{2}}}_{zR,i,j,k+\frac{1}{2}}\right)\right),
\end{aligned}
\tag{55}
$$

with, e.g.,

$$
U^{\prime(y)^{n+\frac{1}{2}}}_{xL,i+\frac{1}{2},j,k} = U^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k} - \frac{\Delta t}{3\Delta y_j}\left(\tilde{F}^{y}_{i,j+\frac{1}{2},k} - \tilde{F}^{y}_{i,j-\frac{1}{2},k}\right),
\tag{56}
$$

and with the primitive variables q' in (55) derived from the corresponding conserved variables U' in (56) by mapping.[2]

Finally, the fluxes entering (53) are fully corner coupled. For example, the flux $F^{x}_{i+\frac{1}{2},j,k}$ has the meaning of “an x-directed flux constructed from L/R states that are fully coupled with flow in the y, z, and combined $\pm y \pm z$ directions”; i.e.,

$$
\begin{aligned}
F^{x,n+\frac{1}{2}}_{i+\frac{1}{2},j,k} &= F^{x}\left(\mathcal{R}\left(q''^{\,n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k}, q''^{\,n+\frac{1}{2}}_{xR,i+\frac{1}{2},j,k}\right)\right),\\
F^{y,n+\frac{1}{2}}_{i,j+\frac{1}{2},k} &= F^{y}\left(\mathcal{R}\left(q''^{\,n+\frac{1}{2}}_{yL,i,j+\frac{1}{2},k}, q''^{\,n+\frac{1}{2}}_{yR,i,j+\frac{1}{2},k}\right)\right),\\
F^{z,n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} &= F^{z}\left(\mathcal{R}\left(q''^{\,n+\frac{1}{2}}_{zL,i,j,k+\frac{1}{2}}, q''^{\,n+\frac{1}{2}}_{zR,i,j,k+\frac{1}{2}}\right)\right),
\end{aligned}
\tag{57}
$$

with, e.g.,

$$
U''^{\,n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k} = U^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k} - \frac{\Delta t}{2\Delta y_j}\left(\tilde{F}^{y|z}_{i,j+\frac{1}{2},k} - \tilde{F}^{y|z}_{i,j-\frac{1}{2},k}\right) - \frac{\Delta t}{2\Delta z_k}\left(\tilde{F}^{z|y}_{i,j,k+\frac{1}{2}} - \tilde{F}^{z|y}_{i,j,k-\frac{1}{2}}\right).
\tag{58}
$$

[2] Flux differencing as in (56) updates conserved variables. This was not presented correctly in [25].

The Riemann solver ([12]; see also Section 5.3) takes as input the pressures P_L, P_R of the left and right states in addition to the primitive variables q_L, q_R. Pressure is not a conserved variable, and so the updates (56) and (58) do not directly give corrected pressures. The simplest approach to solving this problem is a recalculation of the equation of state. For the present ideal gas model, this would be just fine. For some applications, however, the equation of state evaluation is one of the most computationally expensive parts of the solution. With this in mind, we employ a thermodynamic update based on partial differentiation of $\mathcal{E}(\rho, S)$.

In each computational cell, we determine the quantities

$$P_{\mathcal{E}} \equiv \left.\frac{\partial P}{\partial \mathcal{E}}\right|_{\rho\mathcal{E}} \tag{59}$$

and

$$P_{\rho\mathcal{E}} \equiv \left.\frac{\partial P}{\partial(\rho\mathcal{E})}\right|_{\mathcal{E}} \tag{60}$$

at the same time that we evaluate the equation of state. For an ideal gas, $P_{\mathcal{E}} = 0$ and $P_{\rho\mathcal{E}} = (\gamma_p - 1)$. Then, in conjunction with the conservative variable update (56), we additionally solve a pressure update. For example,

$$P'^{(y)^{n+\frac{1}{2}}}_{xL,i+\frac{1}{2},j,k} = P^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k} + (P_{\mathcal{E}})^n_{i,j,k}\Delta\mathcal{E} + (P_{\rho\mathcal{E}})^n_{i,j,k}\Delta(\rho\mathcal{E}), \tag{61}$$

with

$$\Delta\mathcal{E} = \mathcal{E}'^{(y)^{n+\frac{1}{2}}}_{xL,i+\frac{1}{2},j,k} - \mathcal{E}^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k}, \tag{62a}$$

$$\Delta(\rho\mathcal{E}) = \rho'^{(y)^{n+\frac{1}{2}}}_{xL,i+\frac{1}{2},j,k}\mathcal{E}'^{(y)n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k} - \rho^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k}\mathcal{E}^{n+\frac{1}{2}}_{xL,i+\frac{1}{2},j,k}. \tag{62b}$$

A similar correction takes place after conservative update (58).

After determination of fully corner-coupled fluxes (57) we construct a preliminary update $\tilde{U}^{n+1}$ using (53), but with omission of the source terms G. The source terms G are then determined using averaged values of the time t^n and preliminary time (t^{n+1}) primitive variables:

$$\begin{aligned}
\tilde{U}^{n+1}_{ijk} &= U^n_{ijk} - \frac{\Delta t}{\Delta x}\left[F^{x,n+\frac{1}{2}}_{i+\frac{1}{2},j,k} - F^{x,n+\frac{1}{2}}_{i-\frac{1}{2},j,k}\right] \\
&\quad - \frac{\Delta t}{\Delta y}\left[F^{y,n+\frac{1}{2}}_{i,j+\frac{1}{2},k} - F^{y,n+\frac{1}{2}}_{i,j-\frac{1}{2},k}\right] - \frac{\Delta t}{\Delta z}\left[F^{z,n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} - F^{z,n+\frac{1}{2}}_{i,j,k-\frac{1}{2}}\right],
\end{aligned} \tag{63a}$$

$$\tilde{q}^{n+1}_{ijk} \leftarrow \tilde{U}^{n+1}_{ijk}, \tag{63b}$$

$$\bar{q}_{ijk} = \frac{1}{2}\left[\tilde{q}^{n+1}_{ijk} + q^n_{ijk}\right], \tag{63c}$$

$$U^{n+1}_{ijk} = \tilde{U}^{n+1}_{ijk} + \Delta t G_{ijk}(\bar{q}_{ijk}). \tag{63d}$$

4.2.2. *PPM Reconstruction*

High-order Godunov methods rely on calculation of second-order face- and time-centered fluxes. The first step toward calculating these is the reconstruction of the spatial distribution

$q(x)$ beginning with the cell-centered discretization q_i. This reconstruction step is one dimensional and computed separately in each of the grid directions. The notation will be one dimensional as well, except where higher dimensionality is introduced via a 3D limiter.

The method described here was first derived by [16], and substantial additional detail and justification can be found therein. Our implementation of this reconstruction differs from the original mainly in the design of limiters. Colella and Woodward used two monotonicity-related limiters, both of which may be found here. They also used a third “flattening” limiter. Our flattening limiter is similar to theirs and differs principally in being multidimensional. They also had a slope-steepening “contact detection” feature, which we omit altogether. That feature was useful for tracking material interfaces (contact discontinuities), and the point of this report is to introduce a different strategy for that problem. We caution the reader that while we try to adopt as much of the notation of [16] as possible, we deviate in several places. The most confusing of these changes regards labeling of face-centered values. In their notation, $q_{L,i}$ ($q_{R,i}$) denotes the left (right) face-centered value associated with cell i. Instead we use throughout $q_{R,i-\frac{1}{2}}$ ($q_{L,i+\frac{1}{2}}$) to denote these positions: in Colella and Woodward, subscripts L/R refer to the sides of cells; here they refer to the sides of faces.

First we compute the average cross-cell difference in the primitive variables, assuming a quadratic interpolation:

$$\delta q_i = \frac{1}{2}(q_{i+1} - q_{i-1}). \tag{64}$$

This equation, and (66) below, assumes that all Δx_i are equal. More general expressions appear in [16].

A linear reconstruction based on these slopes may introduce new extrema. To enforce monotonicity we limit δq_i with the method of van Leer [43]:

$$\delta q_i^{vL} = \min(|\delta q_i|, 2|q_{i+1} - q_i|,\ 2|q_i - q_{i-1}|)\operatorname{sgn}(\delta q_i). \tag{65}$$

Using these limited cross-cell differences, together with the cell-centered values q_{i-1}, q_i, q_{i+1}, q_{i+2}, we interpolate the data to obtain an estimate of the cell face value $q_{i+\frac{1}{2}}$:

$$q_{i+\frac{1}{2}} = q_i + \frac{1}{2}(q_{i+1} - q_i) - \frac{1}{6}\left(\delta q_{i+1}^{vL} - \delta q_i^{vL}\right). \tag{66}$$

When not limited ($\delta q^{vL} := \delta q$) (66) is the fourth-order estimate [16]:

$$q_{i+\frac{1}{2}} = \frac{7}{12}(q_{i+1} + q_i) - \frac{1}{12}(q_{i+2} + q_{i-1}). \tag{67}$$

At this point, we have single-valued estimates of the cell face values of the primitive variables at time t^n. These values incorporate some sensitivity to linear monotonicity but do not necessarily provide a monotonic reconstruction, particularly in conjunction with a quadratic model. In anticipation of further limiting, which will give rise to double-valued face-centered values, we introduce the notation $q_{L,i+\frac{1}{2}}$ and $q_{R,i+\frac{1}{2}}$:

$$q_{L,i+\frac{1}{2}} = q_{i+\frac{1}{2}}, \tag{68a}$$

$$q_{R,i+\frac{1}{2}} = q_{i+\frac{1}{2}}. \tag{68b}$$

If one were to construct a piecewise parabolic (quadratic) interpolation $q(x)$ in the range $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ using the values $q_{R,i-\frac{1}{2}}, q_i$, and $q_{L,i+\frac{1}{2}}$, then the function $q(x), x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, could take on values outside the range $q_{R,i-\frac{1}{2}}$, q_i, $q_{L,i+\frac{1}{2}}$. To prevent this from happening, we reset the face values as necessary using the quadratic limiter:

$$\left.\begin{aligned} q_{L,i+\frac{1}{2}} &:= q_i \\ q_{R,i-\frac{1}{2}} &:= q_i \end{aligned}\right\} \quad \text{if } \left(q_{i+\frac{1}{2}} - q_i\right)\left(q_i - q_{i-\frac{1}{2}}\right) \leq 0, \tag{69a}$$

$$q_{R,i-\frac{1}{2}} := 3q_i - 2q_{i+\frac{1}{2}} \quad \text{if } \left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}\right)\left(q_i - \frac{q_{i-\frac{1}{2}} + q_{i+\frac{1}{2}}}{2}\right) > \frac{\left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}\right)^2}{6}, \tag{69b}$$

$$q_{L,i+\frac{1}{2}} := 3q_i - 2q_{i+\frac{1}{2}} \quad \text{if } -\frac{\left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}\right)^2}{6} > \left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}\right)\left(q_i - \frac{q_{i+\frac{1}{2}} + q_{i-\frac{1}{2}}}{2}\right). \tag{69c}$$

The above limiters (65) and (69) are one dimensional and designed to prevent artificial extrema in the reconstructed values. We introduce in (78) a third limiter χ, $0 \leq \chi \leq 1$, a "flattening parameter,"

$$q_{R,i-\frac{1}{2}} \leftarrow \chi_i q_{R,i-\frac{1}{2}} + (1 - \chi_i)q_i, \tag{70a}$$

$$q_{L,i+\frac{1}{2}} \leftarrow \chi_i q_{L,i+\frac{1}{2}} + (1 - \chi_i)q_i. \tag{70b}$$

When $\chi_i \to 0$, $q_{L,i+\frac{1}{2}} = q_{R,i-\frac{1}{2}} = q_i$, and our PPM method reverts to a first-order Godunov scheme locally. When $\chi = 1$, no additional limiting takes place.

Now, with properly limited face values one has the limited piecewise parabolic reconstruction in each cell; i.e.,

$$q_i(x) = q_{R,i-\frac{1}{2}} + \xi(x)[\Delta q_i + q_{6i}(1 - \xi(x))], \tag{71}$$

with

$$\xi(x) = \frac{x - x_{i-\frac{1}{2}}}{\Delta x_i}, \quad x_{i-\frac{1}{2}} \leq x \leq x_{i+\frac{1}{2}}, \tag{72a}$$

$$\Delta q_i = q_{L,i+\frac{1}{2}} - q_{R,i-\frac{1}{2}}, \tag{72b}$$

$$\Delta q_{6i} = 6\left[q_i - \frac{1}{2}\left(q_{L,i+\frac{1}{2}} + q_{R,i-\frac{1}{2}}\right)\right]. \tag{72c}$$

This reconstruction (71) has the property that

$$\frac{1}{\Delta x_i} \int_{x_i - \frac{1}{2}\Delta x_i}^{x_i + \frac{1}{2}\Delta x_i} q(x) = q_i. \tag{73}$$

The three-dimensional limiter χ used in (70) is computed as follows (see [25]). First, we construct a dimensionless measure of shock resolution across cell ijk,

$$\beta_{x,ijk} = \frac{|P_{i+1,j,k} - P_{i-1,j,k}|}{|P_{i+2,j,k} - P_{i-2,j,k}|} \tag{74}$$

(and similarly β_y, β_z). If β is $\frac{1}{2}$, then pressure is linear across four computational cells. If β is small enough, then we assume that any discontinuity is already sufficiently well resolved that additional dissipation (flattening) is not required. Introducing two parameters, a_0 and a_1, we estimate the minimum value of the flattening parameter χ to be

$$\tilde{\chi}^{\min}_{x,ijk} = \max\left(0, \min\left(1, \frac{a_1 - \beta}{a_1 - a_0}\right)\right), \tag{75}$$

with numerical values $a_0 = 0.75$ and $a_1 = 0.85$ in our examples.

Next, we construct a dimensionless measure of shock strength,

$$Z_{x,ijk} = \frac{|P_{i+1,j,k} - P_{i-1,j,k}|}{K_{S,ijk}}, \tag{76}$$

where $K_S = \rho c^2$, with c the speed of sound. For an ideal gas, $K_S = \gamma_p P$. Introducing two additional parameters, Z_0 and Z_1, we derive a one-dimensional flattening parameter

$$\tilde{\chi}_{x,ijk} = \begin{cases} \max\left(\tilde{\chi}^{\min}_{x,ijk}, \min\left(1, \frac{Z_1 - Z_{x,ijk}}{Z_1 - Z_0}\right)\right) & \text{if } u_{x,i+1,j,k} < u_{x,i-1,j,k}, \\ 1 & \text{otherwise.} \end{cases} \tag{77}$$

Note that flattening is applied only in convergent flow. Our applications use numerical values $Z_0 = 0.25$ and $Z_1 = 0.75$.

Finally, we combine these one-dimensional parameters to generate a three-dimensional limiter:

$$\begin{aligned} \chi_{ijk} = \min(&\tilde{\chi}_{x,i-1,j,k}, \tilde{\chi}_{x,i,j,k}, \tilde{\chi}_{x,i+1,j,k}, \\ &\tilde{\chi}_{y,i,j-1,k}, \tilde{\chi}_{y,i,j,k}, \tilde{\chi}_{y,i,j+1,k}, \tilde{\chi}_{z,i,j,k-1}, \tilde{\chi}_{z,i,j,k}, \tilde{\chi}_{z,i,j,k+1}). \end{aligned} \tag{78}$$

4.2.3. *PPM Characteristic Tracing*

Let q be a vector of variables; for fluid dynamics we use $q^T = (\rho, v_x, v_y, v_z, \rho\mathcal{E}, P, Q)$, with ρ the mass density, v the velocity, $\mathcal{E}$ the internal energy per unit mass, P the pressure, and Q an advected scalar (e.g., a chemical energy). Note that here the components of q are not strictly primitive variables, as $\rho\mathcal{E}$ is derived by multiplication and $P = P(\rho, \mathcal{E})$ is obtained from an equation of state. Linearized, the Euler equations ((1) specialized to fluids, or (91)) are

$$\frac{\partial q}{\partial t} + A\frac{\partial q}{\partial x} = s, \tag{79}$$

with

$$A = \begin{pmatrix} v_x & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 0 & 1/\rho & 0 \\ 0 & 0 & v_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & v_x & 0 & 0 & 0 \\ 0 & \rho\mathcal{E} + P & 0 & 0 & v_x & 0 & 0 \\ 0 & \rho c^2 & 0 & 0 & 0 & v_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_x \end{pmatrix}. \tag{80}$$

Here, c is the speed of sound. A may be decomposed as $A = R\Lambda L$, with R the matrix of right eigenvectors (with columns r_ν), L the matrix of left eigenvectors (with rows l_ν), and Λ the diagonal matrix of eigenvalues (with diagonal elements λ_ν). Without loss of generality, choose the eigenvectors to be orthonormal so that $RL = I$:

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & c^{-2} & c^{-2} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\rho c} & \frac{1}{\rho c} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \frac{\rho\mathcal{E}+P}{\rho c^2} & \frac{\rho\mathcal{E}+P}{\rho c^2} \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{81}$$

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -\frac{\rho\mathcal{E}+P}{\rho c^2} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -c^{-2} & 0 \\ 0 & -\frac{1}{2}\rho c & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2}\rho c & 0 & 0 & 0 & \frac{1}{2} & 0 \end{pmatrix}, \tag{82}$$

$$\Lambda = \begin{pmatrix} v_x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & v_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & v_x - c & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_x + c \end{pmatrix}. \tag{83}$$

A Taylor series expansion using (79) to determine time derivatives from spatial ones gives the time- and face-centered estimates

$$q_{R,i-\frac{1}{2}}^{n+\frac{1}{2}} = q_i^n - \frac{\Delta x}{2}\frac{dq}{dx} - \frac{\Delta t}{2}A\frac{dq}{dx} + \frac{\Delta t}{2}s_i^n, \tag{84a}$$

$$q_{L,i+\frac{1}{2}}^{n+\frac{1}{2}} = q_i^n + \frac{\Delta x}{2}\frac{dq}{dx} - \frac{\Delta t}{2}A\frac{dq}{dx} + \frac{\Delta t}{2}s_i^n \tag{84b}$$

to second order. Expanding A in eigenvalues λ_ν, and eigenvectors r_ν and l_ν, we have equivalently

$$q_{R,i-\frac{1}{2}}^{n+\frac{1}{2}} = q_i^n - \sum_\nu l_\nu \cdot \left[\frac{dq}{dx}\left(\lambda_\nu\frac{\Delta t}{2} + \frac{\Delta x}{2}\right)\right] r_\nu + \frac{\Delta t}{2}s_i^n, \tag{85a}$$

$$q_{L,i+\frac{1}{2}}^{n+\frac{1}{2}} = q_i^n - \sum_\nu l_\nu \cdot \left[\frac{dq}{dx}\left(\lambda_\nu\frac{\Delta t}{2} - \frac{\Delta x}{2}\right)\right] r_\nu + \frac{\Delta t}{2}s_i^n, \tag{85b}$$

from which it is evident that characteristics of any sign are included. The extrapolation is made upwind by limiting the characteristics in (85) to select those emanating from the cell center:

$$q_{R,i-\frac{1}{2}}^{n+\frac{1}{2}} = q_i^n - \sum_{\nu \text{ such that } \lambda_\nu \le 0} l_\nu \cdot \left[\frac{dq}{dx}\left(\lambda_\nu\frac{\Delta t}{2} + \frac{\Delta x}{2}\right)\right] r_\nu + \frac{\Delta t}{2}s_i^n, \tag{86a}$$

$$q_{L,i+\frac{1}{2}}^{n+\frac{1}{2}} = q_i^n - \sum_{\nu \text{ such that } \lambda_\nu \ge 0} l_\nu \cdot \left[\frac{dq}{dx}\left(\lambda_\nu\frac{\Delta t}{2} - \frac{\Delta x}{2}\right)\right] r_\nu + \frac{\Delta t}{2}s_i^n. \tag{86b}$$

These expressions (86) are the characteristic tracing equations used in piecewise linear schemes, including [25]. To develop the piecewise parabolic tracing equations, note that one can modify (86) by introduction of reference states $q^-_{R,i-\frac{1}{2}}$ and $q^+_{L,i+\frac{1}{2}}$,

$$q^{n+\frac{1}{2}}_{R,i-\frac{1}{2}} = q^-_{R,i-\frac{1}{2}} - \sum_{\nu\ \mathbf{s.t.}\ \lambda_\nu \leq 0} l_\nu \cdot \left[q^-_{R,i-\frac{1}{2}} - q^n_i + \frac{dq}{dx}\left(\lambda_\nu \frac{\Delta t}{2} + \frac{\Delta x}{2}\right)\right] r_\nu + \frac{\Delta t}{2} s^n_i, \quad (87a)$$

$$q^{n+\frac{1}{2}}_{L,i+\frac{1}{2}} = q^+_{L,i+\frac{1}{2}} - \sum_{\nu\ \mathbf{s.t.}\ \lambda_\nu \geq 0} l_\nu \cdot \left[q^+_{L,i+\frac{1}{2}} - q^n_i + \frac{dq}{dx}\left(\lambda_\nu \frac{\Delta t}{2} - \frac{\Delta x}{2}\right)\right] r_\nu + \frac{\Delta t}{2} s^n_i, \quad (87b)$$

without loss of generality provided the vector $q^-_{R,i-\frac{1}{2}} - q^n_i$ lies in the vector space $\{r_\nu\}$ such that $\lambda_\nu \leq 0$ (and $q^+_{L,i+\frac{1}{2}} - q^n_i$ lies in the vector space $\{r_\nu\}$ such that $\lambda_\nu \geq 0$).

Now note that appearing in (87a) is the term

$$q^n_i - \frac{dq}{dx}\left(\lambda_\nu \frac{\Delta t}{2} + \frac{\Delta x}{2}\right) = q\left(x_{i-\frac{1}{2}} - \frac{1}{2}\lambda_\nu \Delta t, t^n\right) \quad (88a)$$

$$\approx \frac{1}{-\lambda_\nu \Delta t} \int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}} - \lambda_\nu \Delta t} q(x)\, dx. \quad (88b)$$

Expressions (88a) and (88b) are interchangeable if the function $q(x)$ appearing in the integrand is linear, but with the quadratic reconstruction (71) these expressions differ. Now introducing the approximation (88) into (87) we have the reconstruction formula used in the PPM method; i.e.,

$$q^{n+\frac{1}{2}}_{R,i-\frac{1}{2}} = q^-_{R,i-\frac{1}{2}} - \sum_{\nu\ \text{s.t.}\ \lambda_\nu \leq 0} l_\nu \cdot \left[q^-_{R,i-\frac{1}{2}} - q^\nu_{R,i-\frac{1}{2}} \right] r_\nu + \frac{\Delta t}{2} s^n_i, \quad (89a)$$

$$q^{n+\frac{1}{2}}_{L,i+\frac{1}{2}} = q^+_{L,i+\frac{1}{2}} - \sum_{\nu\ \text{s.t.}\ \lambda_\nu \geq 0} l_\nu \cdot \left[q^+_{L,i+\frac{1}{2}} - q^\nu_{L,i+\frac{1}{2}} \right] r_\nu + \frac{\Delta t}{2} s^n_i, \quad (89b)$$

with

$$q^\nu_{R,i-\frac{1}{2}} = \begin{cases} \frac{1}{-\lambda_\nu \Delta t} \int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}} - \lambda_\nu \Delta t} q(x)\, dx & \text{if } \lambda_\nu \leq 0, \\ q^n_i & \text{otherwise}, \end{cases} \quad (90a)$$

$$q^\nu_{L,i+\frac{1}{2}} = \begin{cases} \frac{1}{\lambda_\nu \Delta t} \int_{x_{i+\frac{1}{2}} - \lambda_\nu \Delta t}^{x_{i+\frac{1}{2}}} q(x)\, dx & \text{if } \lambda_\nu \geq 0, \\ q^n_i & \text{otherwise}. \end{cases} \quad (90b)$$

The notation $q^+_{L,i+\frac{1}{2}}$ means $q^\nu_{L,i+\frac{1}{2}}$ with ν such that $\lambda_\nu = v + c$: the wave of the "+" family. Likewise, $q^-_{R,i-\frac{1}{2}}$ corresponds to the wave of the "−" family, $\lambda\_ = v - c$.

Note that according to the definitions (90), we do not necessarily satisfy the condition that $q^-_{R,i-\frac{1}{2}} - q^n_i$ be expressible in right eigenvalues $\{r_\nu\}$ s.t. $\lambda_\nu \leq 0$, etc. Therefore, the choice of reference state does matter: (89) with a linear reconstruction $q(x)$ is not equivalent to

(86) because of the reference states. The rationale for choosing the fastest wave of the given direction as a reference state is that this choice minimizes (approximately) the size of the terms subject to characteristic limiting [10, p. 181].

In uniform flow with $\Delta t = \Delta x/\lambda$, and assuming a quadratic reconstruction $q(x)$, the PPM discretization gives the correct result that all mass in the cell will be swept out in time Δt. For this example the truncation error of discretization (86) is $\mathcal{O}(\Delta x^2)$, while that for (89) is $\mathcal{O}(\Delta x^2 - (\lambda\Delta t)^2) + \mathcal{O}(\Delta x^3, \Delta t^3)$ (which is superior to $\mathcal{O}(\Delta x^2)$, notably in the limit $\Delta t = \Delta x/\lambda$). Employing a piecewise linear reconstruction $q(x)$ in conjunction with discretization (86) also gives second-order results: $\mathcal{O}(3(\Delta x - \lambda\Delta t)^2) + \mathcal{O}(\Delta x^3, \Delta t^3)$ for the example described above.

5. SOLID–FLUID COUPLING

5.1. Governing Equations

Specialized to compressible inviscid fluid dynamics the system of PDEs (1) is the Euler equations; i.e.,

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho v \\ \rho E \\ \rho Q \end{pmatrix} + \frac{\partial}{\partial x_\mu}\begin{pmatrix} \rho v_\mu \\ \rho v v_\mu + P e_\mu \\ \rho E v_\mu + P v_\mu \\ \rho Q v_\mu \end{pmatrix} = \begin{pmatrix} 0 \\ \rho f \\ \rho(\Phi + v \cdot f) \\ \rho \dot{Q} \end{pmatrix}, \tag{91}$$

where ρ is the mass density, v the velocity vector, E the total energy, P the pressure, and Q an advected scalar. In the present application, Q is the chemical potential energy, and E is then

$$\begin{aligned} E &= E^{mechanical} + E^{kinetic} + E^{chemical} \\ &= \mathcal{E}(\rho, S) + \frac{1}{2} v \cdot v + Q, \end{aligned} \tag{92}$$

where $\mathcal{E}(\rho, S)$ is specified by an equation of state model (S is the specific entropy). We include for generality a heat source Φ, and body force term f (e.g., gravitational acceleration). e_μ refers to column μ of the identity matrix.

For solid mechanics, the governing equations (1) become [25]

$$\frac{\partial}{\partial t}\begin{pmatrix} \rho \\ \rho v \\ \rho E \\ g e_x \\ g e_y \\ g e_z \\ \rho \mathcal{F}^p e_x \\ \rho \mathcal{F}^p e_y \\ \rho \mathcal{F}^p e_z \\ \rho \kappa \end{pmatrix} + \frac{\partial}{\partial x_\mu}\begin{pmatrix} \rho v_\mu \\ \rho v v_\mu - \sigma e_\mu \\ \rho E v_\mu - v_\beta \sigma_{\beta\mu} \\ g v \delta_{x\mu} \\ g v \delta_{y\mu} \\ g v \delta_{z\mu} \\ \rho \mathcal{F}^p e_x v_\mu \\ \rho \mathcal{F}^p e_y v_\mu \\ \rho \mathcal{F}^p e_z v_\mu \\ \rho \kappa v_\mu \end{pmatrix} = \begin{pmatrix} 0 \\ \rho f \\ \rho(\Phi + v \cdot f) \\ (v \times (\nabla \times g^T))^T e_x \\ (v \times (\nabla \times g^T))^T e_y \\ (v \times (\nabla \times g^T))^T e_z \\ \rho h e_x \\ \rho h e_y \\ \rho h e_z \\ \rho K \end{pmatrix}. \tag{93}$$

Here, g is the inverse total deformation tensor relating a reference material coordinate frame

$\{a\}$ to the current spatial frame $\{x\}$:

$$g_{\mu\beta} = \frac{\partial a_\mu}{\partial x_\beta}. \tag{94}$$

g is subject to two constraints. First, as a gradient it is subject to the equality of mixed partial derivatives, which we write as

$$\nabla \times g^T = 0. \tag{95}$$

Second, the determinant of g relates the material density ρ to the material density ρ_0 in the reference frame $\{a\}$:

$$\tilde{\rho} \equiv \rho_0 \det(g), \tag{96a}$$

$$\rho = \tilde{\rho}. \tag{96b}$$

These constraints are obeyed by the PDE (93) in the sense that if true in the initial condition, they will remain true for all time. Numerically, however, differences arise that must be addressed in the solution of (93).

$\mathcal{F}^p$ is the plastic deformation tensor, which relates a hypothetical stress-free reference frame $\{b\}$ to the material reference frame $\{a\}$,

$$\mathcal{F}^p_{\mu\beta} = \frac{\partial b_\mu}{\partial a_\beta}. \tag{97}$$

Associated with $\mathcal{F}^p$ is the plastic deformation source tensor h (93). We assume the multiplicative decomposition [23]

$$\mathcal{F} = \mathcal{F}^e \mathcal{F}^p, \tag{98}$$

where $\mathcal{F} = g^{-1}$ is the total deformation and $\mathcal{F}^e$ is the elastic deformation tensor. We assume that $\det(\mathcal{F}^p) = 1$: plastic deformation is volume preserving. As with the fluid we define

$$\begin{aligned} E &= E^{mechanical} + E^{kinetic} \\ &= \mathcal{E}(C^e, \kappa, S) + \frac{1}{2} v \cdot v, \end{aligned} \tag{99}$$

with the internal energy a function of the elastic Green tensor

$$C^e = \mathcal{F}^{eT} \mathcal{F}^e = (\mathcal{F}^p g)^{-T} (\mathcal{F}^p g)^{-1}, \tag{100}$$

entropy S, and a scalar constitutive parameter κ, the work hardening parameter, with source K (93).

In (93) σ is the stress tensor, derived as a thermodynamic derivative of the internal energy function $\mathcal{E}$; i.e.,

$$\sigma_{\mu\beta} = -\rho \frac{\partial \mathcal{E}}{\partial g_{\gamma\beta}} g_{\gamma\mu}, \tag{101}$$

where $\mathcal{E}$ (99) is a function of the inverse deformation $g = \mathcal{F}^{-1}$ through C^e (100) by way of the decomposition (98). Stress σ and fluid pressure P are related by $P = -\frac{1}{3}\mathrm{tr}(\sigma)$, and the stress tensor of a fluid is given by $\sigma = -PI$, with I the identity matrix.

Note that Eq. (93) is not strictly in conservation form since the terms $v \times (\nabla \times g^T)^T$ contain gradients. However, to the extent that the constraint $\nabla \times g^T = 0$ is obeyed, these terms are negligible and we will treat (93) as though it were a system of conservation laws.

5.2. Constitutive Models

To ensure thermodynamic self-consistency, we formulate our material models using an entropy-dependent internal energy function (a so-called "fundamental equation" [8]). Temperature, stress (pressure), and other intensive state variables are determined from the energy function by taking appropriate thermodynamic derivatives. Two such material models will be described here. The first is an ideal gas—a familiar model recast as a fundamental equation. The second is an elastically isotropic compressible solid. This model's complexity is largely inherent: there are a large number of internal variables and a correspondingly large number of experimental observations that must be adequately reproduced. The resulting material model is in essence little different from the commonly used Mooney–Rivlin model (a two-parameter model for rubber elasticity [28, 38]) but is more realistic for crystalline solids under large compression.

5.2.1. *Reacting Gas*

In energy function form $\mathcal{E}(\rho, S)$, the fundamental equation for an ideal gas is given by

$$\mathcal{E}(\rho, S) = \mathcal{E}_0 \left[\frac{\rho}{\rho_0} \exp\left(\frac{S - S_0}{R} \right) \right]^{(\gamma_p - 1)}, \tag{102}$$

where R is the gas constant and γ_p is the polytropic gas index, equal to the ratio of specific heats $\gamma_p = C_P/C_V > 1$. $\mathcal{E}_0$, S_0, and ρ_0 are state points in some thermodynamic reference state; $\mathcal{E}_0 = \mathcal{E}(\rho_0, S_0)$.

The pressure P is given by

$$P = \rho^2 \frac{\partial \mathcal{E}}{\partial \rho}\bigg|_s = (\gamma_p - 1)\mathcal{E}\rho, \tag{103}$$

and the temperature T is given by

$$T = \frac{\partial \mathcal{E}}{\partial S}\bigg|_\rho = \frac{\mathcal{E}}{C_V}, \tag{104}$$

where C_V is the constant volume specific heat of an ideal gas:

$$C_V = \frac{R}{\gamma_p - 1}. \tag{105}$$

We augment this thermodynamic model with a very simple model for chemical reaction kinetics. We assume that no reaction occurs when the temperature is below some threshold

T_R, and that the rate is constant otherwise:

$$\dot{Q} = \begin{cases} \dot{Q}_0 & \text{if } T \geq T_R, \\ 0 & \text{if } T < T_R. \end{cases} \tag{106}$$

Note that this reaction model only has the effect of transferring chemical potential energy Q to internal energy $\mathcal{E}$: there is no change in mole number.

5.2.2. *Elastically Isotropic Solid*

We base our fundamental equation of an elastically isotropic solid on a so-called "universal equation of state" model for hydrostatic compression. This model was originally proposed as a scaling relation for the binding energy of metals [39] and has since been found to apply to a wide variety of different chemical interactions. Applications to the hydrostatic equation of state of solids are described in [44, 45].

We assume a separation of the total internal energy as

$$\mathcal{E}(C^e, \kappa, S) = \mathcal{E}_h(I_3) + \mathcal{E}_t(I_3, S) + \mathcal{E}_s(I_1, I_2, I_3) + \mathcal{E}_w(\kappa), \tag{107}$$

where $\mathcal{E}_h$ describes the isentropic, hydrostatic compressional energy, $\mathcal{E}_t$ is the thermal energy associated with changing entropy at constant volume, $\mathcal{E}_s$ is the energy associated with isochoric shearing, and $\mathcal{E}_w$ is the energy term associated with work hardening. In (107) I_1, I_2, and I_3 are the isotropic invariants of the elastic Green tensor C^e:

$$I_1(C^e) = \mathrm{tr}(C^e), \tag{108a}$$

$$I_2(C^e) = \frac{1}{2}\left[(\mathrm{tr}C^e)^2 - \mathrm{tr}(C^e)^2\right], \tag{108b}$$

$$I_3(C^e) = \det(C^e) = \left(\frac{\rho_0}{\rho}\right)^2. \tag{108c}$$

The hydrostatic energy is given by the universal equation of state, fit to the zero pressure isentropic bulk modulus K_{0S} and to the isentropic pressure derivative of the isentropic bulk modulus at zero pressure, K'_{0s}; i.e.,

$$\mathcal{E}_h(I_3) = -\frac{4K_{0s}}{\rho_0(K'_{0s} - 1)^2}(1 + r_K)e^{-r_K}, \tag{109}$$

with

$$r_K = \frac{3(K'_{0s} - 1)}{2}\left[\left(\frac{\rho_0}{\rho}\right)^{1/3} - 1\right], \tag{110}$$

and where the density ρ is understood to depend on I_3 through (108c). ρ_0 is the density at zero pressure.

$\mathcal{E}_t(I_3, S)$ is the thermal part, modeled on a Mie–Grüneisen form,

$$\mathcal{E}_t(I_3, S) = C_V T_0\left(\exp\left[\frac{S - S_0}{C_V}\right] - 1\right)\exp\left[\frac{\gamma_0 - \gamma(I_3)}{q}\right], \tag{111}$$

where C_V is a constant heat capacity, S_0 and T_0 are the entropy and temperature in the reference configuration (at zero pressure and density ρ_0), and where $\gamma(I_3)$ is the thermodynamic Grüneisen parameter given by the model equation

$$\gamma = \gamma_0 \left(\frac{\rho_0}{\rho} \right)^q, \tag{112}$$

where γ_0 and $q \neq 0$ are constants. Again, in (112) ρ is determined from I_3 via (108c).

The energy change due to shearing motion at constant volume is given by

$$\mathcal{E}_s(I_1, I_2, I_3) = \frac{G(\rho)}{2\rho} \left[\beta I_1 I_3^{-1/3} + (1-\beta) I_2 I_3^{-2/3} - 3 \right]. \tag{113}$$

The function $G(\rho)$ is the shear modulus, also constructed to follow the universal equation of state formalism, and fit to the zero pressure shear modulus G_0 and the pressure derivative of the shear modulus also evaluated at zero pressure, G_0':

$$G(\rho) = G_0 \left[(1 - r_G) \left(\frac{\rho}{\rho_0} \right)^{1/3} - \frac{4}{3} r_G \left(\frac{K_{0S}}{G_0} G_0' - 1 \right)^{-1} \left(\frac{\rho}{\rho_0} \right)^{2/3} \right] e^{-r_G} \tag{114}$$

$$r_G = \frac{3}{2} \left[\frac{K_{0S}}{G_0} G_0' - 1 \right] \left[\left(\frac{\rho_0}{\rho} \right)^{1/3} - 1 \right]. \tag{115}$$

Note that on the hydrostat of an elastically isotropic solid $I_1 = 3 I_3^{1/3}$ and $I_2 = 3 I_3^{2/3}$, and so $\mathcal{E}_s \to 0$. In (113) the parameter β, $0 \le \beta \le 1$, is an adjustable parameter chosen to control the symmetry of the shear potential away from the hydrostat (see [28]).

The work hardening potential $\mathcal{E}_w(I_3, \kappa)$ is given by

$$\mathcal{E}_w(\kappa) = \frac{\vartheta_0}{\rho_0} \left(\kappa + \frac{1}{\vartheta_1} [e^{-\vartheta_1 \kappa} - 1] \right). \tag{116}$$

This equation gives a work hardening modulus with units of pressure,

$$\vartheta(\kappa) = \rho \frac{\partial \mathcal{E}}{\partial \kappa} = \vartheta_0 \frac{\rho}{\rho_0} (1 - e^{-\vartheta_1 \kappa}), \tag{117}$$

in terms of two parameters: $\vartheta_0(\rho/\rho_0)$ is the ultimate, asymptotic value of the work hardening modulus, and ϑ_1 dictates the rate of approach of the asymptotic limit.

Our plastic flow model is a simple isotropic rate-independent-associated model based on the yield surface

$$f(\sigma, \vartheta) = \sqrt{\frac{3}{2}} \|\text{dev}\sigma\| - \left(\sigma_Y + \frac{\rho_0}{\rho} \vartheta \right), \tag{118}$$

where $\text{dev}(\sigma) = \sigma - \frac{1}{3}\,(\text{tr}\sigma) I$ is the stress deviator and $\|A\|$ is the Schur norm of A,

$\|A\|^2 = A_{\mu\beta}A_{\mu\beta} = \mathrm{tr}(A^T A)$. $\sqrt{\frac{3}{2}}\|\mathrm{dev}\sigma\|$ is sometimes called J_2, or the second invariant of the stress deviator.

Associated with this yield surface are the rate laws [25, 36]; i.e.,

$$\dot{\mathcal{F}}^p = \zeta \mathcal{F}^p g \frac{\mathrm{dev}(\sigma)}{\|\mathrm{dev}(\sigma)\|}\mathcal{F}, \tag{119a}$$

$$\dot{\kappa} = \zeta\sqrt{\frac{2}{3}}\frac{\rho_0}{\rho}, \tag{119b}$$

with ζ a (Lagrange undetermined) parameter chosen to satisfy the Kuhn–Tucker conditions and the "consistency condition" [41]:

$$\left.\begin{array}{l} f = 0 \\ \zeta \geq 0 \\ \dot{f} = 0 \end{array}\right\} \text{plastic flow.} \tag{120}$$

The work hardening energy (116), yield surface description (118), and resulting flow laws (119) differ from the model used in [25] by better separating plastic work from elastic strain energy. In the current formulation, the elastic response of the solid is not affected by its plastic strain history.

Although $\mathcal{E}$ is presented formally as a fundamental equation depending on specific entropy S, S and its reference value S_0 need never be specified. Instead, one could use (102) to determine $S - S_0 = \Delta S(\rho, \mathcal{E})$, etc. In the solid case one could determine

$$\mathcal{E}_t = \mathcal{E} - \mathcal{E}_h(I_3) - \mathcal{E}_s(I_1, I_2, I_3) - \mathcal{E}_w(\kappa) \tag{121}$$

and thereby obtain the temperature

$$T = \left.\frac{\partial\mathcal{E}}{\partial S}\right|_{C^e,\kappa} = \frac{\mathcal{E}_t}{C_V} + T_0 \exp\left[\frac{\gamma_0 - \gamma}{q}\right]. \tag{122}$$

The reference temperature T_0, and the heat capacity C_V are not required to solve the dynamical equations except in so far as the temperature itself is required by constitutive models, e.g., because of a temperature-dependent yield strength. The Grüneisen parameter γ_0 and its constitutive parameter q are necessary, however, to determine the thermal stress arising from the volume dependence of $\mathcal{E}_t$.

5.3. Interface Riemann Problems

At a point on the two-material interface, we may have "left" and "right" states with different velocities and stresses. Let us approximate the left and right states as being constant. This, then, is the canonical Riemann problem: an initial value problem consisting of two piecewise constant states in juxtaposition. The solution to this problem is a set of waves (three in each solid, one in each fluid), with amplitudes chosen to satisfy appropriate interface compatibility conditions. In this solution, we denote by superscript "*" the state on the material interface.

The methods described here are the main part of our overall method that bring together the different physics of the fluid (91) and solids (93). The other part concerns volume fraction normalization.

5.3.1. *Posing an Interface-Normal Problem*

We wish to pose a one-dimensional Riemann problem, with one material on the left and another on the right. In general, the interface is not oriented with the computational grid. Therefore, we need to rotate our state vectors, and transform state tensors, from the current spatial system $(\vec{x}, \vec{y}, \vec{z})$ into a new right-handed orthogonal coordinate system $(\vec{n}, \vec{\tau}, \vec{\omega})$, where $\vec{n}$ is the interface normal and $\vec{\tau}$ and $\vec{\omega}$ are interface tangential directions. $\vec{n}$ is computed as part of our interface reconstruction method, described above. $\vec{\tau}$ and $\vec{\omega}$ are orthogonal to $\vec{n}$ but are otherwise arbitrary. Here we describe our particular choices for $\vec{\tau}$ and $\vec{\omega}$, and their use in transforming the state variables.

One choice of tangents is

$$\tau = \frac{1}{\sqrt{2(1 - n_x n_y - n_x n_z - n_y n_z)}} \begin{pmatrix} (n_y - n_z) \\ (n_z - n_x) \\ (n_x - n_y) \end{pmatrix} \tag{123a}$$

and

$$\omega = \frac{1}{\sqrt{2(1 - n_x n_y - n_x n_z - n_y n_z)}} \begin{pmatrix} \left(n_x(n_y + n_z) - n_y^2 - n_z^2\right) \\ \left(n_y(n_x + n_z) - n_x^2 - n_z^2\right) \\ \left(n_z(n_x + n_y) - n_x^2 - n_y^2\right) \end{pmatrix}. \tag{123b}$$

This choice fails in particular when $n_x = n_y = n_z = \pm 1/\sqrt{3}$, and so we use it only when $|n_y + n_z| \le |n_y - n_z|$. When $|n_y + n_z| > |n_y - n_z|$ we use instead

$$\tau = \frac{1}{\sqrt{2(1 + n_z(n_y - n_x) + n_x n_y)}} \begin{pmatrix} (n_y + n_z) \\ (n_z - n_x) \\ -(n_x + n_y) \end{pmatrix} \tag{124a}$$

and

$$\omega = \frac{1}{\sqrt{2(1 + n_z(n_y - n_x) + n_x n_y)}} \begin{pmatrix} \left(n_x(n_z - n_y) - n_y^2 - n_z^2\right) \\ \left(n_y(n_x + n_z) + n_x^2 + n_z^2\right) \\ \left(n_z(n_x - n_y) - n_x^2 - n_y^2\right) \end{pmatrix}. \tag{124b}$$

(which fails in particular when $n_x = -n_y = n_z = \pm 1/\sqrt{3}$).

Given a new orthonormal basis $(\vec{n}, \vec{\tau}, \vec{\omega})$ we construct the rotation matrix R,

$$R = \begin{pmatrix} n^T \\ \tau^T \\ \omega^T \end{pmatrix}, \tag{125}$$

with which rotated material properties are derived by straightforward vector and tensor manipulations [31]. For example, the primitive variables characterizing a solid are transformed as follows:

$$\begin{pmatrix} \rho \\ \mathcal{E} \\ v \\ g \\ \mathcal{F}^p \\ \kappa \end{pmatrix}^{rotated} = \begin{pmatrix} \rho \\ \mathcal{E} \\ R \cdot v \\ R \cdot g \cdot R^T \\ R \cdot \mathcal{F}^p \cdot R^T \\ \kappa \end{pmatrix}. \tag{126}$$

The transformation of fluid-phase primitive variables are contained as a subset of (126) (scalar Q transforms like scalar κ).

5.3.2. *Conditions of Compatibility*

The compatibility conditions are constraints linking the state variables on one side of a material interface to those on the opposite side. Physically, these conditions describe continuity of normal velocity, normal stress, and shear traction. Mathematically, these conditions describe the Riemann invariants of the contact discontinuities (genuinely nonlinear waves propagating at the material velocity). Here we will describe the compatibility conditions for a number of two-material interfaces. In Section 5.3.3 we describe in more detail how these conditions are employed to solve a two-material Riemann problem. We distinguish several special cases characterized by different material pairs, or different idealized assumptions regarding shear tractions.

- Solid–vacuum: A solid in contact with vacuum has no normal stress and no shear stresses on the contact surface. Thus, the solution of the Riemann problem is characterized by the compatibility conditions

$$\begin{aligned} \sigma_{nn}^{*S} &= 0, \\ \sigma_{n\tau}^{*S} &= 0, \\ \sigma_{n\omega}^{*S} &= 0. \end{aligned} \tag{127}$$

- Fluid–vacuum: A fluid in contact with vacuum has no pressure, therefore

$$P^{*F} = 0. \tag{128}$$

- Solid–fluid: The normal stress is continuous across the contact, as is the normal velocity. For an ideal inviscid fluid there are no shear stresses on the contact surface; i.e.,

$$\begin{aligned} v_n^{*S} &= v_n^{*F}, \\ \sigma_{nn}^{*S} &= -P^{*F}, \\ \sigma_{n\tau}^{*S} &= 0, \\ \sigma_{n\omega}^{*S} &= 0. \end{aligned} \tag{129}$$

- Fluid–fluid: The two-fluid condition is equality of normal stress and of normal velocity; i.e.,

$$\begin{aligned} v_n^{*F_1} &= v_n^{*F_2}, \\ P^{*F_1} &= P^{*F_2}. \end{aligned} \tag{130}$$

- "Slip" solid–solid: When two solids are in contact, the normal velocity and normal stress are continuous. If the contact is frictionless, then there are no shear stresses and tangential velocities may jump discontinuously; i.e.,

$$\begin{aligned} v_n^{*S_1} &= v_n^{*S_2}, \\ \sigma_{nn}^{*S_1} &= \sigma_{nn}^{*S_2}, \\ \sigma_{n\tau}^{*S_1} &= 0, \\ \sigma_{n\tau}^{*S_2} &= 0, \\ \sigma_{n\omega}^{*S_1} &= 0, \\ \sigma_{n\omega}^{*S_2} &= 0. \end{aligned} \tag{131}$$

This is the "slip" solid–solid boundary condition.

- "Stick" solid–solid: Another idealized limit is the "stick" solid–solid boundary condition. This case is analogous to the solid–solid Riemann problem posed within any single-material region. In this case, all components of velocity are continuous, as are normal stress and in-plane shear stresses; i.e.,

$$\begin{aligned} v_n^{*S_1} &= v_n^{*S_2}, \\ v_\tau^{*S_1} &= v_\tau^{*S_2}, \\ v_\omega^{*S_1} &= v_\omega^{*S_2}, \\ \sigma_{nn}^{*S_1} &= \sigma_{nn}^{*S_2}, \\ \sigma_{n\tau}^{*S_1} &= \sigma_{n\tau}^{*S_2}, \\ \sigma_{n\omega}^{*S_1} &= \sigma_{n\omega}^{*S_2}. \end{aligned} \tag{132}$$

Other model compatibility conditions may be used. For example, when modeling gas flow with irregular internal boundaries it is convenient to assume incompressible rigid walls. As these boundaries are stationary, the compatibility condition becomes $v_n = 0$ [32].

For each solid phase, the Riemann solution will involve three waves. For each fluid phase, the Riemann solution will involve a single wave. Each interface compatibility condition provides sufficient constraints to determine the requisite number of waves. For example, a solid–solid interface Riemann solution will consist of six total waves and six constraints are provided by each of (131) and (132). Solid–fluid interface solutions will involve four waves: three in the solid and one in the fluid; (129) gives the four constraints needed.

5.3.3. *Solution of the Solid–Fluid Riemann Problem*

In detail, the solution of each interface special case is constructed differently. Here we illustrate the general approach by describing the particular case of a solid in contact with a fluid.

The one-dimensional two-material Riemann problems are solved approximately, using rarefaction shock approximations [12, 26]. Use of this approximation is justified by the observation that the rarefaction wave curves and the corresponding shock Hugoniot are C^2, and so the error in treating a rarefaction as a shock is third order, thus negligible in a numerical scheme whose overall order of accuracy is 2 or less. As rarefaction wave fans expand, the magnitude of the "rarefaction jumps" will diminish, and the solution will therefore converge to the proper result as the waves becomes resolved across several computational cells.

As in [25] our approach to the solid Riemann problem begins with the linearized one-dimensional equation of motion

$$\frac{\partial q}{\partial t} + A\frac{\partial q}{\partial x_n} = s \tag{133}$$

in terms of the primitive variables $q^T = (\rho, v, \mathcal{E}, ge_n, ge_\tau, ge_\omega, \mathcal{F}^p e_n, \mathcal{F}^p e_\tau, \mathcal{F}^p e_\omega, \kappa, \sigma e_n)$. The matrix A is given by

$$A = \begin{pmatrix} v_n & \rho e_n^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_n I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I/\rho \\ 0 & -(\sigma e_n)^T/\rho & v_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g\delta_{nn} & 0 & v_n I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g\delta_{\tau n} & 0 & 0 & v_n I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g\delta_{\omega n} & 0 & 0 & 0 & v_n I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_n I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_n I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_n I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_n & 0 \\ 0 & -\mathcal{A}_{nn} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_n I \end{pmatrix}, \tag{134}$$

where

$$\mathcal{A}_{\mu\beta} = -\frac{\partial \sigma e_\mu}{\partial g e_\beta} g, \tag{135}$$

and where $\mathcal{A}_{nn}$ is the acoustic tensor for waves propagating in the normal (n) direction.

The vector s contains source terms,

$$s^T = (0, f, \Phi, 0, 0, 0, he_n, he_\tau, he_\omega, K, b_n), \tag{136}$$

with

$$b_n = \frac{\partial \sigma e_n}{\partial \mathcal{F}^p} : h + \frac{\partial \sigma e_n}{\partial \kappa} K + \frac{\partial \sigma e_n}{\partial \mathcal{E}} \Phi. \tag{137}$$

The right eigenvectors of A, X, are

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\rho e_n^T X_{ac} & -\rho e_n^T X_{ac} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_{ac}\Lambda_{ac} & -X_{ac}\Lambda_{ac} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (\sigma e_n)^T X_{ac}/\rho & (\sigma e_n)^T X_{ac}/\rho \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & -gX_{ac} & -gX_{ac} \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X_{ac}\Lambda_{ac}^2\rho & X_{ac}\Lambda_{ac}^2\rho \end{pmatrix}, \tag{138}$$

where X_{ac} are the right eigenvectors of the acoustic wave propagation equation

$$\mathcal{A}_{nn} X_{ac} = \rho X_{ac} \Lambda_{ac}^2. \tag{139}$$

The columns of the 27×27 matrix X are ordered such that the first 21 are associated with waves of the “0” family (with characteristic speed given by eigenvalue $\lambda = v_n$). Columns 22–24 are associated with waves of the “−” family ($\lambda = v_n - \lambda_{ac}$), and columns 25–27 are associated with waves of the “+” family ($\lambda = v_n + \lambda_{ac}$).

Now, following the approach of [42] we determine the state q^* of the solid at the material interface by expanding the waves in the eigenvectors X. Let us assume that in our rotated coordinate frame the solid is to the left of the fluid. We would then solve for q^* using the three waves of the “−” family, using in this case eigenvectors 22, 23, and 24:

$$q^* = q + \frac{\Delta t}{2} s + \sum_{\mu=22}^{24} c_\mu X_\mu. \tag{140}$$

From (140) with compatibility condition (129) we have

$$\begin{pmatrix} c_{22} \\ c_{23} \\ c_{24} \end{pmatrix} = \frac{1}{\rho} \Lambda_{ac}^{-2} X_{ac}^{-1} \begin{pmatrix} -P^* - \sigma_{nn} \\ -\sigma_{n\tau} \\ -\sigma_{n\omega} \end{pmatrix}. \tag{141}$$

Also, the normal velocity is determined by (140) and (129):

$$v_n^* = v_n^S + X_{ac}\Lambda_{ac} \begin{pmatrix} c_{22} \\ c_{23} \\ c_{24} \end{pmatrix} = v_n^S + \frac{1}{\rho} X_{ac} \Lambda_{ac}^{-1} X_{ac}^{-1} \begin{pmatrix} -P^* - \sigma_{nn} \\ -\sigma_{n\tau} \\ -\sigma_{n\omega} \end{pmatrix}. \tag{142}$$

This equation (142) is one equation coupling the interface velocity v_n^* to the interface pressure P^*.

A second equation relating these variables is obtained by consideration of the fluid jump conditions. Following [12], we write the Rankine–Hugoniot equations for the fluid as

$$\begin{aligned} |v_n^* - v_n| &= \pm \frac{|P^* - P|}{W}, \\ \frac{|P^* - P|}{W^2} &= -|\rho^{*-1} - \rho^{-1}|, \\ |\mathcal{E}^* - \mathcal{E}| &= \frac{1}{2}(P^* + P)|\rho^{*-1} - \rho^{-1}|, \end{aligned} \tag{143}$$

where W is the mass flux through the (shock) wave connecting the initial state and the final "*" state. In the present case, where we assume that the fluid lies to the right of the interface, we take the + sign in (143). Recall that the variables $\mathcal{E}^*$, P^*, and ρ^* are connected through the equation of state.

Equations (143) determine a one-parameter (W) monotonic relation between v_n^* and P^*. If the solid–fluid interface Riemann problem has a physical solution that does not require cavitation, then the $P^*(v_n^*)$ curve specified by (143) intersects the linear relation given by (142) exactly once. We compute this intersection using an iterative approach.

- We begin by estimating an interface velocity

$$v_n^{*(0)} = \frac{1}{2}\left(v_n^S + v_n^F\right). \tag{144}$$

- For each step m of our iterative solution, an estimate $P^{*(m)}$ of the interface pressure is obtained from the input velocity $v^{*(m-1)}$ by solving (143). This step itself is iterative, using the secant method approach described in [12]:

$$P^{*(m)} \xleftarrow[W^{(m)}]{} v_n^{*(m-1)}. \tag{145}$$

- (142) provides a function

$$\Delta V = \underbrace{v^*(P^{*(m)})}_{\text{from}\,(142)} - v^{*(m-1)} \tag{146}$$

whose zero $v^{*(m)}$ we estimate with a Newton iteration,

$$v^{*(m)} = v^{*(m-1)} - \frac{\Delta V}{\frac{d\Delta V}{dv}}, \tag{147}$$

with

$$\frac{d\Delta V}{dv} = -\left(\frac{1}{\rho}\sum_{\mu=1}^{3} X_{ac,1\mu}\lambda^{-1}_{ac,\mu}X^{-1}_{ac,\mu 1}\right)_{solid} \left.\frac{dP^*}{dv^*}\right|_{fluid} - 1. \tag{148}$$

- The iteration is stopped when the change $|v_n^{*(m)} - v_n^{*(m-1)}|$ is small enough, say 0.1% of the fluid sound speed.

On convergence, the "*" state of the solid and fluid states are readily determined from (140) and (143).

In our present implementation, we take as the left state the cell-centered properties of the solid at time t^n, augmented by one half time step, $\frac{1}{2}\Delta t^n$, of the source term (136). Similarly, we take as the right state the cell-centered properties of the fluid at time t^n, augmented by one half time step of the fluid source term (including chemical reaction). The centering of this solution is thus only $\mathcal{O}(h)$ with respect to the space–time centroid of the interface during time step Δt^n. This centering, and its overall impact on the solution accuracy, is discussed further in Section 6.

5.3.4. *The Interface Velocity and Flux*

For each material (left and right) we transform the "*" Riemann solution state back to the coordinate frame of our solution grid. This makes use of the same transformation matrix R

(125) used to transform into the interface normal direction (126):

$$\begin{pmatrix} \rho^* \\ \mathcal{E}^* \\ v^* \\ g^* \\ \mathcal{F}^{p*} \\ \kappa^* \end{pmatrix} = \begin{pmatrix} \rho^* \\ \mathcal{E}^* \\ R^T \cdot v^* \\ R^T \cdot g^* \cdot R \\ R^T \cdot \mathcal{F}^{p*} \cdot R \\ \kappa^* \end{pmatrix}^{\textit{rotated}} . \tag{149}$$

Note that in the rotated frame $v^*_{Ln} = v^*_{Rn}$ as a consequence of the compatibility conditions, but the tangential components will differ in general. However, in general the grid-oriented velocities given by (149) will differ entirely.

We additionally wish to rotate the vector of stresses acting on the interface,

$$(\sigma^*_{nx}\sigma^*_{ny}\sigma^*_{nz}) = (\sigma^*_{nn}\sigma^*_{n\tau}\sigma^*_{n\omega}) \cdot R. \tag{150}$$

Unlike the velocities, these stress components are single valued at the interface.

Now, given the Riemann solutions "*" for each material we may compute the flux vector (the generalization of $F^\mu(U^*)$ in (1) to direction $\mu = n$). Written for a solid,

$$F^B = \underbrace{\begin{pmatrix} \rho(v_n - v_s) \\ \rho v_x(v_n - v_s) - \sigma_{nx} \\ \rho v_y(v_n - v_s) - \sigma_{ny} \\ \rho v_z(v_n - v_s) - \sigma_{nz} \\ \rho E(v_n - v_s) - \sigma_{nx}v_x - \sigma_{ny}v_y - \sigma_{nz}v_z \\ n_x(g_{11}v_x + g_{12}v_y + g_{13}v_z) - v_s g_{11} \\ n_x(g_{21}v_x + g_{22}v_y + g_{23}v_z) - v_s g_{21} \\ n_x(g_{31}v_x + g_{32}v_y + g_{33}v_z) - v_s g_{31} \\ n_y(g_{11}v_x + g_{12}v_y + g_{13}v_z) - v_s g_{12} \\ n_y(g_{21}v_x + g_{22}v_y + g_{23}v_z) - v_s g_{22} \\ n_y(g_{31}v_x + g_{32}v_y + g_{33}v_z) - v_s g_{32} \\ n_z(g_{11}v_x + g_{12}v_y + g_{13}v_z) - v_s g_{13} \\ n_z(g_{21}v_x + g_{22}v_y + g_{23}v_z) - v_s g_{23} \\ n_z(g_{31}v_x + g_{32}v_y + g_{33}v_z) - v_s g_{33} \\ \rho\mathcal{F}^p_{11}(v_n - v_s) \\ \rho\mathcal{F}^p_{21}(v_n - v_s) \\ \rho\mathcal{F}^p_{31}(v_n - v_s) \\ \rho\mathcal{F}^p_{12}(v_n - v_s) \\ \rho\mathcal{F}^p_{22}(v_n - v_s) \\ \rho\mathcal{F}^p_{32}(v_n - v_s) \\ \rho\mathcal{F}^p_{13}(v_n - v_s) \\ \rho\mathcal{F}^p_{23}(v_n - v_s) \\ \rho\mathcal{F}^p_{33}(v_n - v_s) \\ \rho\kappa(v_n - v_s) \end{pmatrix}^*}_{\text{general case}} \rightarrow \underbrace{\begin{pmatrix} 0 \\ -\sigma_{nx} \\ -\sigma_{ny} \\ -\sigma_{nz} \\ -\sigma_{nx}v_x - \sigma_{ny}v_y - \sigma_{nz}v_z \\ n_x(g_{11}v_x + g_{12}v_y + g_{13}v_z) - v_n g_{11} \\ n_x(g_{21}v_x + g_{22}v_y + g_{23}v_z) - v_n g_{21} \\ n_x(g_{31}v_x + g_{32}v_y + g_{33}v_z) - v_n g_{31} \\ n_y(g_{11}v_x + g_{12}v_y + g_{13}v_z) - v_n g_{12} \\ n_y(g_{21}v_x + g_{22}v_y + g_{23}v_z) - v_n g_{22} \\ n_y(g_{31}v_x + g_{32}v_y + g_{33}v_z) - v_n g_{32} \\ n_z(g_{11}v_x + g_{12}v_y + g_{13}v_z) - v_n g_{13} \\ n_z(g_{21}v_x + g_{22}v_y + g_{23}v_z) - v_n g_{23} \\ n_z(g_{31}v_x + g_{32}v_y + g_{33}v_z) - v_n g_{33} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^*}_{\text{material interface: } v_s = v^*_n} \tag{151}$$

In writing (151) we distinguished between the material velocity at the interface v_n^* and the velocity of the interface itself v_s. In our present application, where we are concerned with material interfaces (which move with the material velocity), $v_s = v_n^*$, resulting in considerable simplification. The more complete form is important if $v_s \neq v_n^*$, as when the interface in question is a detonation wave or a shock wave.

The evolution of the material domain $\Omega^\alpha(t)$ is governed by a volume fraction advection equation (23a) that requires an interface velocity field v defined in the neighborhood of the interface $\delta\Omega^\alpha$. Away from the interface an estimate of the interface velocity field may be taken from the material velocity field,

$$v^{interface} = v^{material}. \tag{152}$$

At material interfaces, the two-material Riemann problem provides a unique interface-normal velocity. The components of velocity tangential to the interface are not given by the two-material Riemann problem when slip interfaces are allowed (as done here). In that case, the tangential components of the material velocities on either side of the interface are averaged with density weighting. We use the material densities and material velocities obtained as solutions to interface-normal Riemann problems,

$$\begin{pmatrix} v_n \\ v_\tau \\ v_\omega \end{pmatrix}^{interface} = \begin{pmatrix} v_n^* \\ \frac{\rho_S^* v_\tau^{*S} + \rho^{*F} v_\tau^{*F}}{\rho^{*S} + \rho^{*F}} \\ \frac{\rho^{*S} v_\omega^{*S} + \rho^{*F} v_\omega^{*F}}{\rho^{*S} + \rho^{*F}} \end{pmatrix}, \tag{153a}$$

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}^{interface} = R^T \cdot \begin{pmatrix} v_n \\ v_\tau \\ v_\omega \end{pmatrix}^{interface}. \tag{153b}$$

In summary, for solid–fluid interfaces we construct a velocity field in the neighborhood of the material interface using results from the interface Riemann problem (153), augmented away from the material interface by the actual material velocity (152). For other intrfaces this procedure may not be appropriate. For example, in applying these methods to nonmaterial interfaces (e.g., a detonation [33]) the material velocity has no relation to the detonation front velocity. In this case, one should instead extend the velocity determined at the interfaces itself (153) to cells adjacent to the interface [4, 33]. Even when tracking material interfaces, the approach described above may be inappropriate, for example a solid–vacuum or fluid–vacuum interface. Then, the approach described above fails for want of a vacuum material velocity. In this case, too, velocity extension ideas work well. The algorithms we use to extend interface velocity are described in Section 4.1.

5.3.5. *Moduli of Incompressibility*

The volume fraction evolution equations (23a) introduce a modulus of incompressibility. According to the derivation in [26] this modulus acts to normalize Λ^α such that $\sum_\alpha \Lambda^\alpha = 1$, while maintaining equality of pressure at the interface. For fluids, then, M is isentropic bulk modulus K_S. For solids, and solid–fluid mixtures, equality of normal stress, not equality of pressure, is the correct generalization. However, in this implementation we use the bulk modulus. For the fluid phase,

$$M^F := K_S = \left.\frac{\partial P}{\partial \ln \rho}\right|_S . \tag{154}$$

For the solid phase,

$$M^S := K_S = s_{e11} + s_{e22} + s_{e33} + 2(s_{e12} + s_{e23} + s_{e31}), \tag{155}$$

where s_e is the elastic compliance tensor, here indexed in Voigt notation. s_e is the inverse of the Eulerian elastic constant tensor c_e centered at the current state, which may be written in terms of components of the acoustic propagation tensors $\mathcal{A}$ (135):

$$c_e = \begin{pmatrix} (\mathcal{A}_{xx})_{xx} & (\mathcal{A}_{xy})_{xy} & (\mathcal{A}_{xz})_{xz} & (\mathcal{A}_{xy})_{xz} & (\mathcal{A}_{xx})_{xz} & (\mathcal{A}_{xx})_{xy} \\ (\mathcal{A}_{yx})_{yx} & (\mathcal{A}_{yy})_{yy} & (\mathcal{A}_{yz})_{yz} & (\mathcal{A}_{yy})_{yz} & (\mathcal{A}_{yx})_{yz} & (\mathcal{A}_{yx})_{yy} \\ (\mathcal{A}_{zx})_{zx} & (\mathcal{A}_{zy})_{zy} & (\mathcal{A}_{zz})_{zz} & (\mathcal{A}_{zy})_{zz} & (\mathcal{A}_{zx})_{zz} & (\mathcal{A}_{zx})_{zy} \\ (\mathcal{A}_{yx})_{zx} & (\mathcal{A}_{yy})_{zy} & (\mathcal{A}_{yz})_{zz} & (\mathcal{A}_{yy})_{zz} & (\mathcal{A}_{yx})_{zz} & (\mathcal{A}_{yx})_{zy} \\ (\mathcal{A}_{xx})_{zx} & (\mathcal{A}_{xy})_{zy} & (\mathcal{A}_{xz})_{zz} & (\mathcal{A}_{xy})_{zz} & (\mathcal{A}_{xx})_{zz} & (\mathcal{A}_{xx})_{zy} \\ (\mathcal{A}_{xx})_{yx} & (\mathcal{A}_{xy})_{yy} & (\mathcal{A}_{xz})_{yz} & (\mathcal{A}_{xy})_{yz} & (\mathcal{A}_{xx})_{yz} & (\mathcal{A}_{xx})_{yy} \end{pmatrix} . \tag{156}$$

5.4 Examples

We demonstrate our method with one- and three-dimensional examples using three material phases. The parameters for our elastic-plastic solid were chosen to approximate the response of oxygen-free annealed copper. Our fluid is a crude approximation to the explosive PBX 9404 modeled as an ideal gas with realistic heat of reaction and initial unreacted solid density. The equation-of-state parameters for these materials are given in Table I. Equation of state and reaction energy Q for PBX 9404 are taken from [18]. The rate $\dot{Q}$ is estimated to give a reaction zone spread over a few computational cells. Elastic equation-of-state parameters for copper come from [1] except for G_0', which is approximated from the scaling relation $G'/K' \approx (G/K)^2$ [2], and C_V, which is set to $3R$ per mol of atoms. β, q, and the plastic constitutive parameters for copper are estimated.

In single-level computations (those not using adaptive mesh refinement) the global time step is chosen to maintain the maximum Courant–Friedrichs–Lewy number near a prescribed value. The CFL number for a single phase α is calculated as the maximum over occupied cells of $\Delta t(|v_x| + \max(c^\alpha))/\Delta x$, without modification for fractional occupancy ($\Lambda^\alpha \neq 1$). The CFL number for fractionally occupied cells is the maximum over phases of the single-phase CFL numbers. In multilevel (AMR) computations we do employ time subcycling; thus a $2\times$ refined grid is solved with half the time step of its coarser parent. The CFL numbers are computed at all levels of refinement, and the largest is used to reset the time step at the beginning of a coarse cycle.

By this strategy, the material with the fastest wave speeds controls the overall time step. This will lead to low computational efficiency when the wave speeds of different materials differ dramatically. However, in the cylinder test problem (Section 5.4.2) this does not occur. During the initial combustion phase the explosive controls the CFL number; afterward, as the cylinder inflates, the solid liner controls the CFL number. Throughout the computation, the maximum CFL number of the solid and that of the fluid agree to within approximately 25%.

TABLE I
EOS Parameters Used in Examples

Parameter	Value	Units
Reacting gas (Section 5.2.1)		
$\mathcal{E}_0$	2.938×10^{-5}	[kJ/g]
ρ_0	1.84	[g/cc]
γ_p	2.85	[ ]
Q_0	100	[kJ/g · s]
T_R	2000	[K]
MW	25	[g/mol]
Elastically isotropic solid (Section 5.2.2)		
ρ_0	8.93	[g/cc]
K_{0S}	138	[GPa]
K'_{0S}	4.96	[ ]
G_0	46.9	[GPa]
G'_0	0.57	[ ]
β	0	[ ]
T_0	300	[K]
C_V	3.9×10^{-4}	[kJ/g · K]
γ_0	1.96	[ ]
q	1	[ ]
σ_Y	0.23	[GPa]
ϑ_0	0.12	[GPa]
ϑ_1	1	[ ]

5.4.1. *One-Dimensional Riemann Problems*

To demonstrate the ability of this method to correctly model simple wave interactions, we present computational examples of various one-dimensional Riemann problems.

Figures 10–12 display the results of a one-dimensional solid–fluid shock–shock Riemann problem using the initial values given in Table II. The interface was at coordinate $x = 50$ mm

TABLE II
Initial Values for Solid–Fluid Riemann Problem

Parameter	Value	Units
Reacting gas		
Q	5.543	[kJ/g]
ρ	1.84	[g/cc]
$\mathcal{E}$	2.938×10^{-5}	[kJ/g]
v	0	[km/s]
Elastically isotropic solid		
ρ	ρ_0	[g/cc]
$\mathcal{E}$	Such that $\mathcal{E}_t = 0$	[kJ/g]
g	$(\rho/\rho_0)^{\frac{1}{3}} I$	[ ]
$\mathcal{F}^p$	I	[ ]
κ	0	[ ]
v	2	[km/s]

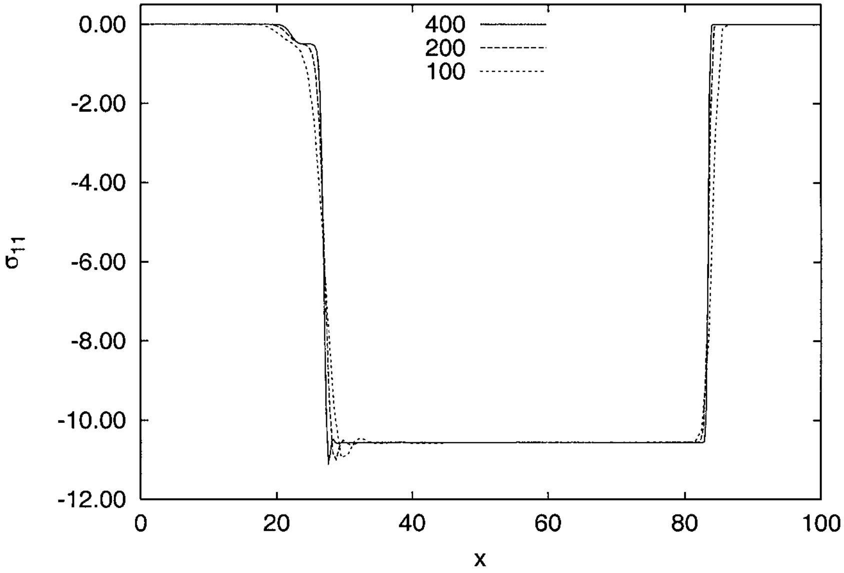

FIG. 10. Stress for one-dimensional solid–fluid Riemann problem.

initially. Results are plotted at $t = 10\,\mu\text{s}$ using 100, 200, and 400 mesh points and CFL = 0.7. In the fluid phase, the shock is captured in about four computational cells without ringing or overshoot. In the solid an elastic precursor shock is seen, followed by a slower plastic shock. These discontinuities are captured in approximately eight cells with modest overshoot. This behavior was also observed in [25]. We suspect that the broader solid shock zone and the overshoot are artifacts of the linearized solid Riemann solver we employ for solid mechanics.

Figures 13–15 display the results of a one-dimensional solid–vacuum rarefaction Riemann problem using the equation of state parameters given in Table I and the initial

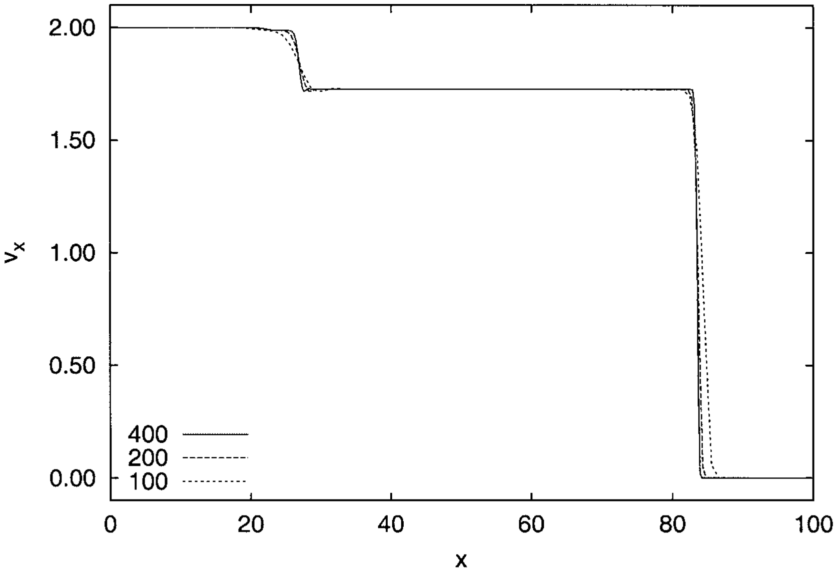

FIG. 11. Velocity for one-dimensional solid–fluid Riemann problem.

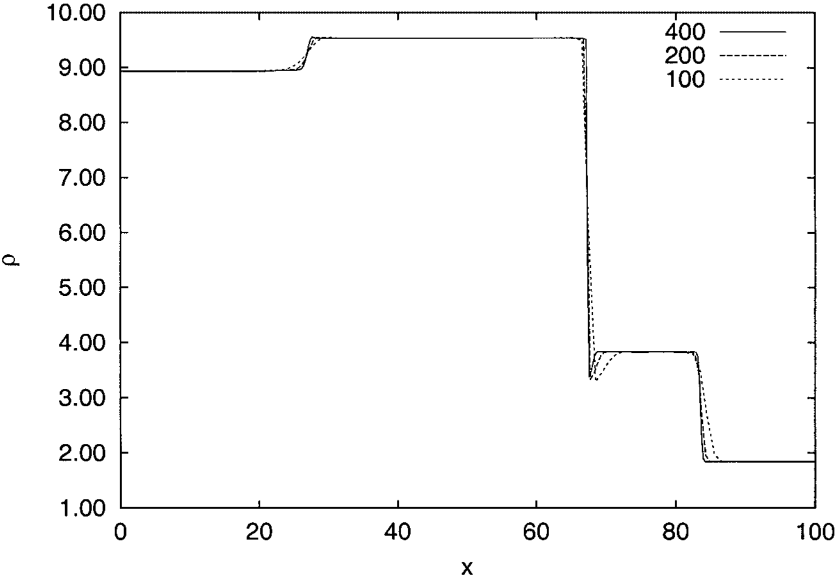

FIG. 12. Density for one-dimensional solid–fluid Riemann problem.

values given in Table III. The interface was at coordinate $x = 80$ mm initially. Results are plotted at $t = 10\,\mu\text{s}$ using 200, 400, and 800 mesh points and CFL $= 0.7$. A weak elastic rarefaction precedes the larger amplitude plastic rarefaction fan.

In smooth flow, our solid and fluid solvers give second-order convergence. For these test problems involving discontinuities, the convergence results are given in Table IV. $\mathcal{O}(1)$ entropy errors occur at the material interfaces: a so-called "wall heating" effect. This entropy error accounts for the apparent density deficit in the gas phase seen in Fig. 12. A discussion of the order of convergence is given in Section 6.

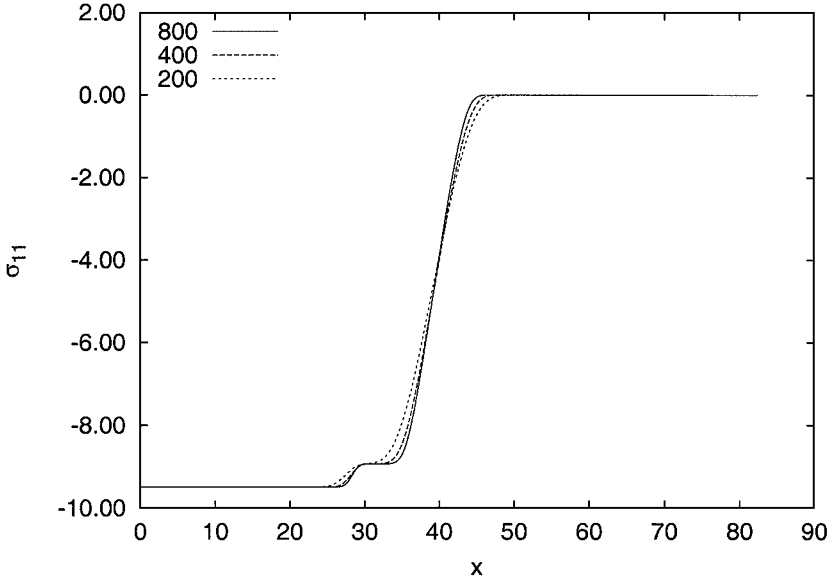

FIG. 13. Stress for one-dimensional solid–vacuum Riemann problem.

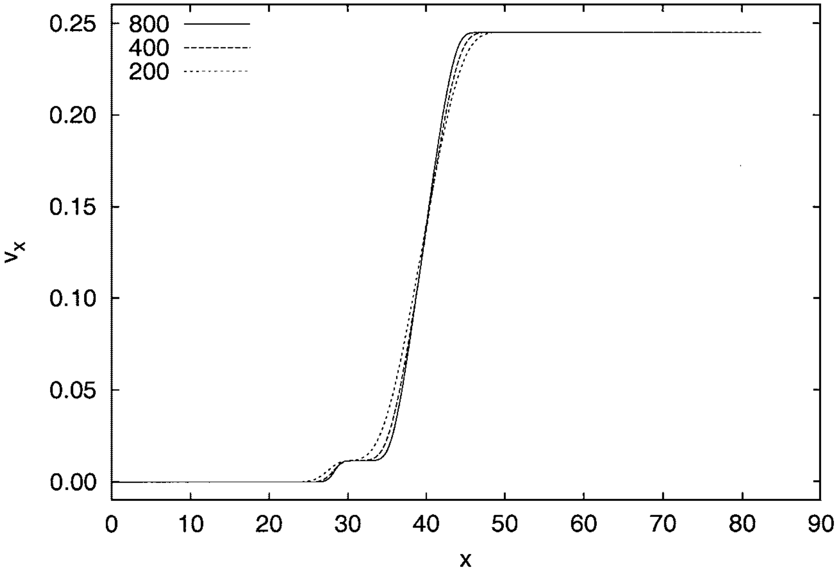

FIG. 14. Velocity for one-dimensional solid–vacuum Riemann problem.

5.4.2. *Cylinder Test with AMR*

This example is a highly idealized model of a cylinder test: an experiment in which an explosive is detonated inside a cylindrical metal tube for the purpose of measuring detonation velocity. Wilkins [46] included this example in his description of a Lagrangian multiphase technique. Other Lagrangian techniques are described in [18]. The nominal initial values are given in Table V. The reaction was initiated by creating a disk-shaped region at one end of the fluid cylinder, where the initial chemical energy Q_0 was completely converted to internal energy: $\mathcal{E}_0 := \mathcal{E}_0 + Q_0$; $Q_0 := 0$. Surrounding the solid is vacuum.

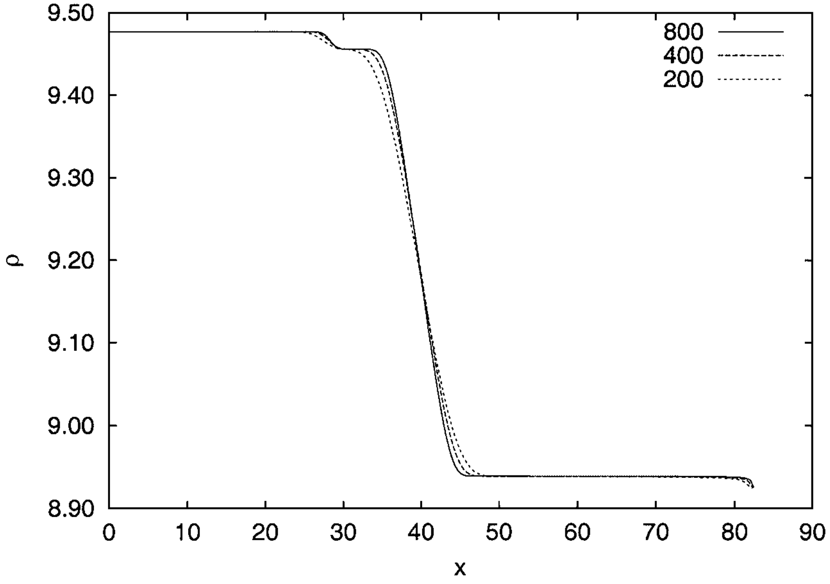

FIG. 15. Density for one-dimensional solid–vacuum Riemann problem.

TABLE III
Initial Values for Solid-Vacuum Riemann Problem

Parameter	Value	Units
Elastically isotropic solid		
ρ	$1.05^3\rho_0$	[g/cc]
$\mathcal{E}$	Such that $\mathcal{E}_t = 0$	[kJ/g]
g	$(\rho/\rho_0)^{\frac{1}{3}} I$	[ ]
$\mathcal{F}^p$	I	[ ]
κ	0	[ ]
v	0	[km/s]

TABLE IV
Order of Convergence

	Solid–fluid		Solid-vacuum	
Field	L_1	L_∞	L_1	L_∞
ρ	1.64	1.00	1.09	1.89
v_x	1.82	0.63	0.76	0.60
σ_{11}	1.33	0.33	0.77	0.68

TABLE V
Initial Values for Contained Explosion Problem

Parameter	Value	Units
Reacting gas		
Q	5.543	[kJ/g]
ρ	1.84	[g/cc]
$\mathcal{E}$	2.938×10^{-5}	[kJ/g]
v	0	[km/s]
Elastically isotropic solid		
ρ	ρ_0	[g/cc]
$\mathcal{E}$	Such that $\mathcal{E}_t = 0$	[kJ/g]
g	I	[ ]
$\mathcal{F}^p$	I	[ ]
κ	0	[ ]
v	0	[km/s]

Initially, the metal can has length 24.6 (say cm) and outside radius of 14.0, with a wall thickness of 2.0. The inside corners are given a chamfer of approximate size 0.6×0.6. The coarse base mesh is $112 \times 64 \times 64$ with 0.3 units of length per coarse cell.

Adaptive mesh refinement was employed to resolve at 2× resolution the material interface, the solid, and those parts of the fluid containing gradients. At small times, the unreacted fluid is represented only on a coarse grid except for those parts adjacent to the interface. As the reaction proceeds, the resolved mesh grows to encompass the entire solid and fluid regime. Coarse regions persist outside the solid where only vacuum is present.

Results are displayed in Figs. 16–18. We show each computational cell containing solid or fluid, with color corresponding to pressure (scales vary). As in Fig. 5, fractionally occupied cells are shown truncated by a piecewise planar interface. The geometry pictured corresponds directly to the internal representation of our method.

In Fig. 16 the detonation wave has traversed approximately half the cylinder length. Ahead of the detonation wave a zone remains that is not adaptively refined, but behind the detonation all fluid and solid regions are refined. Solid and fluid pressure need not be continuous (normal stress is), but because of the low yield strength of "copper" in this model it appears so. The leading shock is refracted backward in the copper and reflects off the free surface as a rarefaction fan. The explosive shock is normal to the cylinder axis at the center but is bent backward near the wall, suggesting something like a Mach reflection. In Fig. 17 the shock has begun to reflect off the far copper wall. Note the jump in pressure scales. In Fig. 18 the pressure had been released significantly by ballooning of the copper can. The reflected shock is about half way back along the axis.

6. DISCUSSION

6.1 Implementation

We have implemented the multiphase tracked interface algorithms within a framework for parallelism and data management called Chombo [14]. Chombo is a collection of C++ classes that includes dimension-independent support for box and set calculus, distributed data management, and support for patch-based adaptive mesh refinement (AMR) based on the approach of Berger and Colella [6] and grid generation models of Berger and Rigoutsos [7]. Chombo derives from an object-oriented approach to AMR developed initially by Crutchfield and Welcome [17].

To facilitate this discussion, we first review some standard AMR nomenclature. A *refinement level L*, or *level L*, refers to a set of data or operations characterized by a given spatial resolution. Level 0 is the coarsest resolution, with grid spacing Δx^0, Δy^0, and Δz^0. Level $L + 1$ is refined relative to *level L* by a factor $R^L : R^0 \Delta x^1 = \Delta x^0 \cdot R^L$, the *level L refinement ratio*, is restricted to be an integer power of 2. A *patch* is a rectangular region of data associated with a particular level. A *grid* is the set of all patches at a given level. At a given level the patches are nonoverlapping.

Level $L + 1$ grids are *properly nested* within their parent level L grid. This means that a boundary of level L cells surrounds the level $L + 1$ grid, except where the level $L + 1$ grid abuts the boundary of the overall problem domain. The width of boundary cells is the *buffer size*: an implementation-dependent adjustable parameter.

To advance level L data in time (say from time t^{n_L} to time t^{n_L+1}) requires some combination of the following steps. First, boundary conditions for the level L grid must be obtained

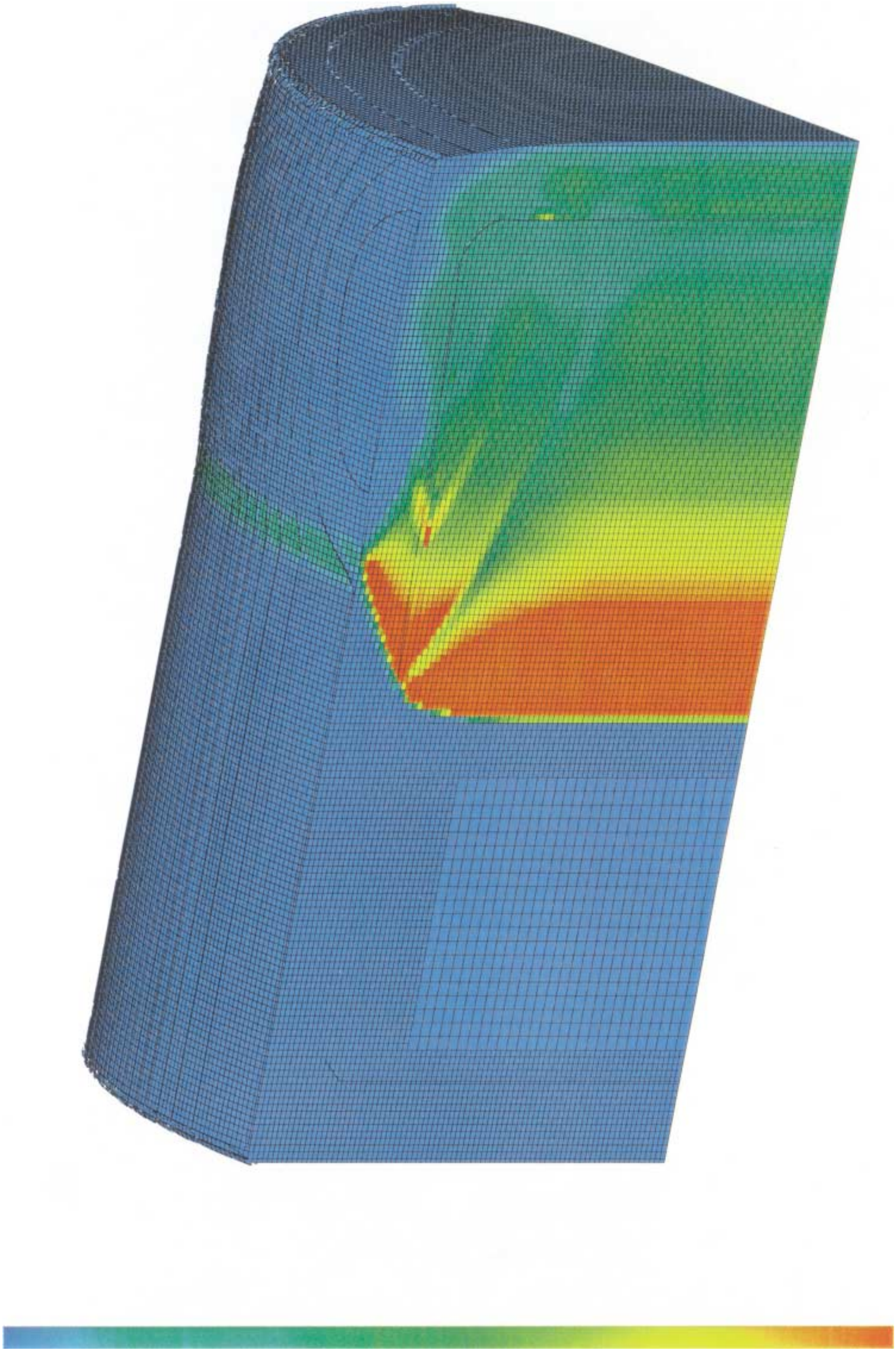

FIG. 16. Pressure (0–20 GPa) at time 1.30.

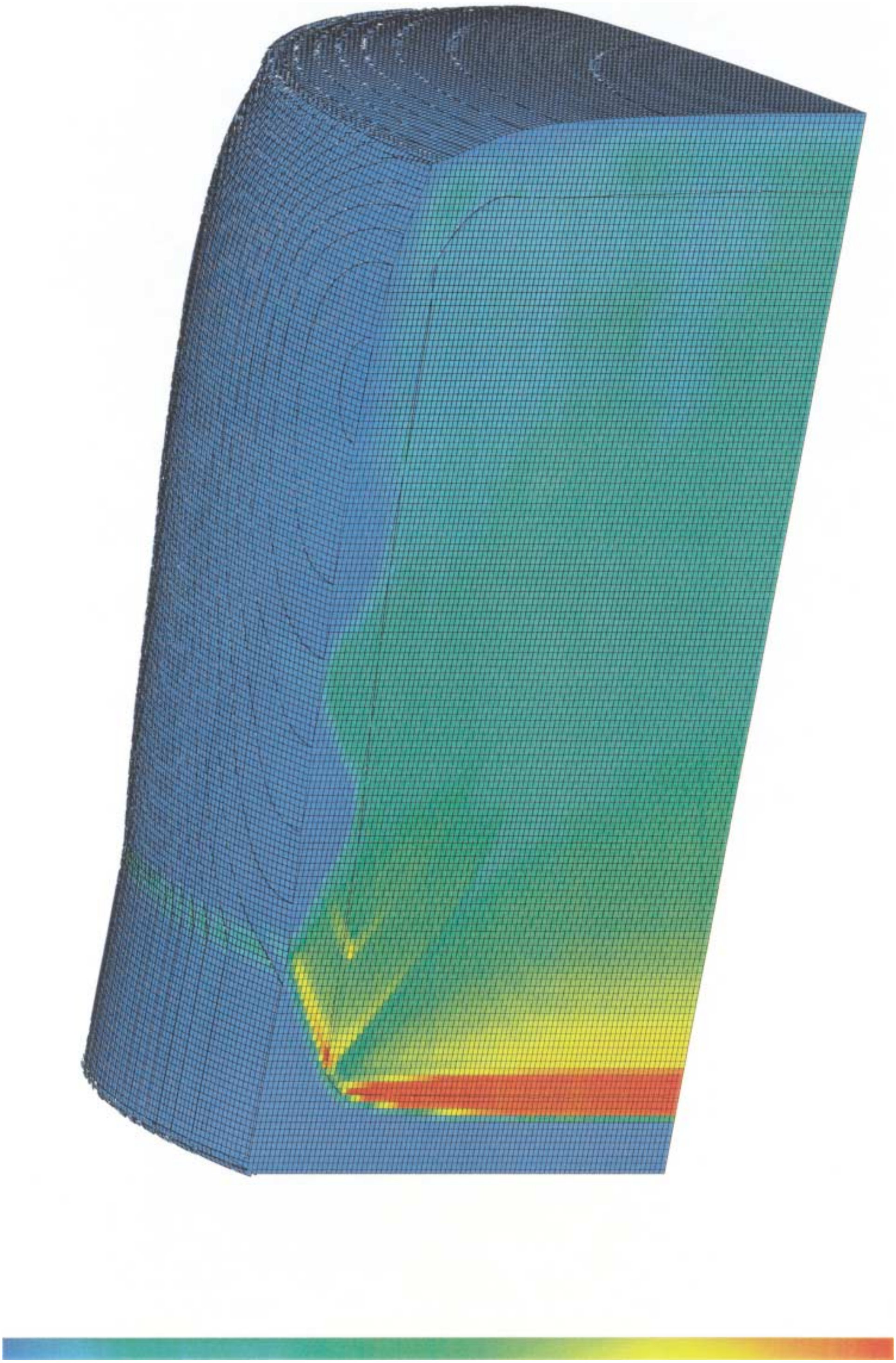

FIG. 17. Pressure (0–40 GPa) at time 2.39.

by space–time interpolation from the level $L-1$ parent grid, or by some prescription for those boundary cells that lie outside the problem domain. Second, the algorithms described above are used to update the solution on the level L grid. Third, those level L cells that are underlain by additionally refined level $L+1$ cells are adjusted to agree with the refined cell time t^{n_L+1} results. Fourth, to maintain conservation, the solution in level L cells that abut the level $L+1$ grid must be adjusted to be compatible with the cumulative fluxes calculated on level $L+1$ across the $L+1$ grid boundary (*flux correction*).

To avoid the problem of space–time interpolation of the interface itself (represented either as volume fractions or piecewise planar segments), we cover all interface cells with the maximum resolution grid. This also solves a potential flux correction issue: mass redistribution effectively creates flux terms associated with 13 spatial directions (e.g., redistribution fluxes may connect cell i, j, k to cell $i+1,\ j-1,\ k+1$), whereas otherwise only the three cardinal directions, x, y, and z, need be considered.

An important issue concerns the overall domain of dependence of the composite method described here. In total the domain of dependence is $29 \times 29 \times 29$ if volume fractions and conserved quantities are chosen as the basic variables (see Table VI): the solution at a given cell may depend on the data in the surrounding 24,389-cell neighborhood. If implemented as a single stencil, then associated with each patch would be a ghost cell boundary with a width of 14 cells. This has obvious and serious impact on computer memory requirements and is clearly impractical.

Instead, the overall domain of dependence may be decomposed into a collection of smaller stencils; however, doing so means that (i) methods must be provided for the space–time interpolation of each new variable, which implies that (ii) the new variables must be stored on two time levels on the parent grid. For example, given the algorithm decomposition indicated in Table VI, the smallest stencil possible for the composite method is $9 \times 9 \times 9$, or a ghost width of 4. An AMR strategy based on this stencil would require storage of interface data, apertures, the interface velocity field, and other "internal" variables at two time levels on each parent grid, and a prescription for space–time interpolation of these quantities.

In the implementation presented here, we designed a strategy based upon a stencil of $13 \times 13 \times 13$ (ghost width 6), with conserved variables, volume fractions, and piecewise planar interface segments chosen as basic variables. This strategy involves two inter-patch-communication operations per time step. When the interface region is always fully refined, only conservative variables need be space–time interpolated. The need for space–time interpolation of the interface $(\vec{n}, d)$ is circumvented.

TABLE VI

Stencils of Algorithm Components

Interface reconstruction	$5 \times 5 \times 5$
Aperture calculation	$5 \times 5 \times 5$
Data extension	$5 \times 5 \times 5$
Interface velocity	$9 \times 9 \times 9$
Volume fraction time advancing	$5 \times 5 \times 5$
Godunov integrators	$9 \times 9 \times 9$
Redistribution	$5 \times 5 \times 5$

6.2. Algorithmic Details

6.2.1. *Redistribution*

In the shock front tracking applications of Chern and Colella [9] and Bell *et al.* [4], the mass deficit δM is decomposed in the sense of characteristics. If any characteristics cross the front, then some fraction of δM is distributed to cells on the opposite side of the front. In the present material interface tracking application, it does not make sense to distribute mass across the front, whatever the characteristics. The correct generalization of this approach would be to redistribute volume between phases, if so indicated by the characteristic structure.

If one were to redistribute material phase volume across the fronts, then two problems would arise. First, after this redistribution the volume fractions would not necessarily sum to 1, and a renormalization would be required (as in Section 3.2). Second, it is possible that after redistributing volumes in the sense of characteristics one would have a volume fraction field that is inconsistent with the existence of a resolved interface: this volume redistribution would smear the interface. For this reason, we do not decompose δM in the sense of characteristics, and we instead redistribute this quantity on one side of the front only.

6.2.2. *Geometric Consistency*

In (47) the normal $\tilde{n}$ appearing on the left hand side is the normal of the surface that closes the volume bounded by cell-face apertures $A_{i-\frac{1}{2},j,k}$, etc. $\tilde{n}$ is not, in general, equal to the interface-normal determined as in Section 3.1, at t^n (where the fluxes F^B are computed), t^{n+1}, or elsewhere. Even if there exists a single planar interface, stationary in time, $\tilde{n}$ would not generally coincide with n because the cell-face apertures are averages of left- and right-cell calculations (42).

We use (47) to compute A^B, which is consistent with interface normal $\tilde{n}$. In (15) we use this aperture with our interface flux F^B, which is consistent with a different normal n.

Note also that because of aperture averaging (42) we may compute a nonzero A^B in cells for which no interface had been detected, and in which F^B is therefore not known. In this case, we construct an estimated flux F^B by averaging the flux over neighbor cells in which it has been determined. This remedy assumes that the normal vector $\tilde{n}$ in the current cell (where F^B had not been computed) is similar to normals n in those neighbor cells where F^B had been computed.

6.2.3. *Source Terms*

One way to discretize (1) is by solving the homogeneous part stably, using redistribution, and then adding the source term contribution afterward:

$$\tilde{U}^{n+1}_{ijk} \underset{\text{redistribution}}{\leftarrow} U^{ext,n+1}_{ijk} \text{ (homogeneous)}\,, \tag{157a}$$

$$U^{n+1}_{ijk} = \tilde{U}^{n+1}_{ijk} + \Delta t \frac{\Lambda^{n+1}_{ijk} + \Lambda^{n}_{ijk}}{2\Lambda^{n+1}_{ijk}} G\left(U^{n+\frac{1}{2}}_{ijk}\right). \tag{157b}$$

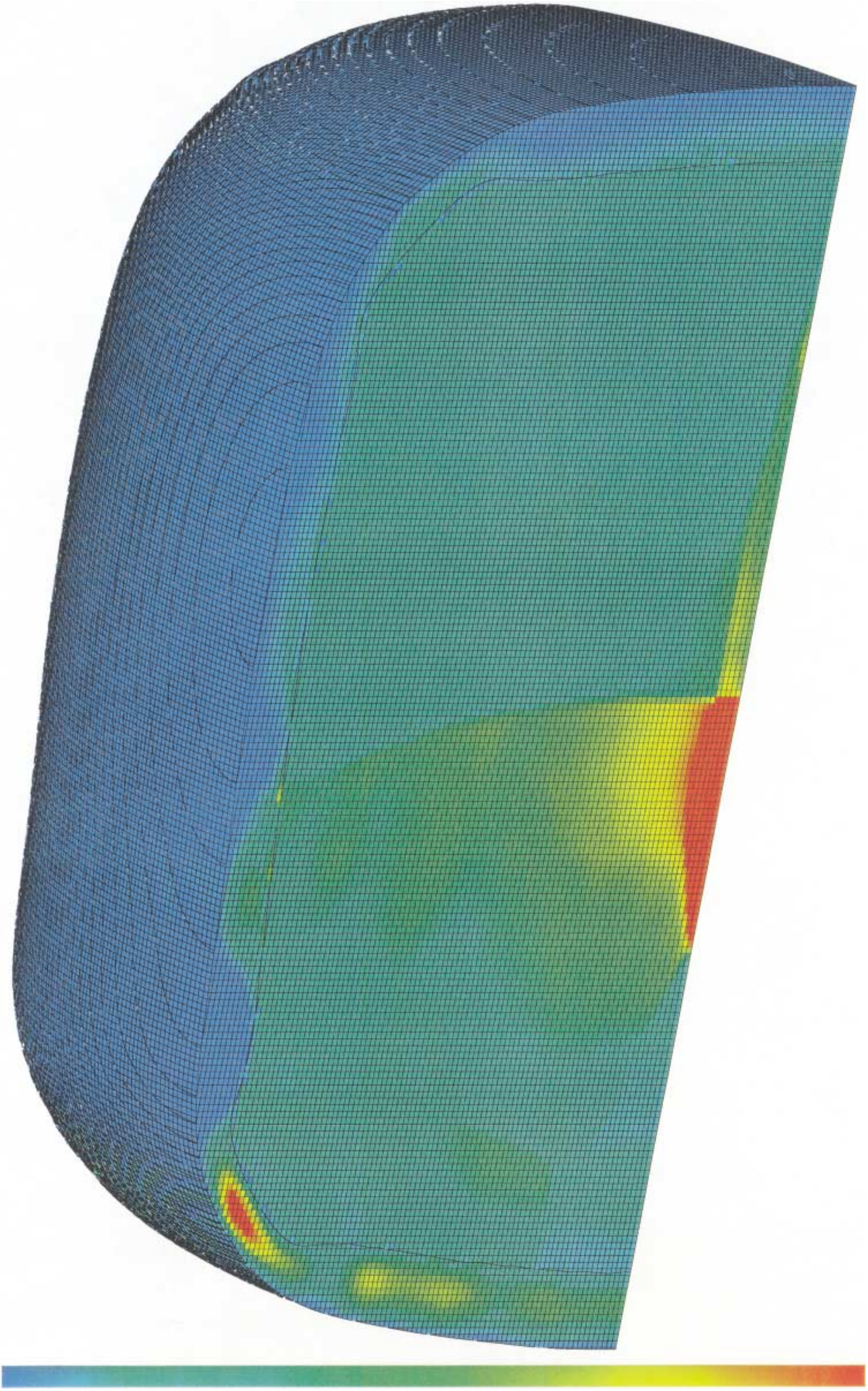

FIG. 18. Pressure (0–10 GPa) at time 4.43.

We use this approach for the plastic source terms h and K occurring in solid mechanics, but with the simplification

$$U_{ijk}^{n+1} = \tilde{U}_{ijk}^{n+1} + \Delta t G\left(\frac{1}{2}\left[q_{ijk}^{ext,n} + \tilde{q}_{ijk}^{n+1}\right]\right) \tag{158}$$

in lieu of (157b). That is, for these source terms we approximate the time-centered state by averaging time t^n (extended) primitive variables, and time t^{n+1} primitive variables obtianed by redistribution of the homogeneous system of equations. We omit the factor $\bar{\Lambda}/\Lambda$ and, in so doing, introduce on $\mathcal{O}(h)$ error.

A diffusion correction [25] is also introduced into the equations of solid mechanics, and this too is approximated with (158).

Another approach to discretizing (1) is to introduce the source terms in the single-phase solvers prior to redistribution:

$$U_{ijk}^{ext,n+1} = \cdots + \Delta t G\left(\frac{1}{2}\left[q_{ijk}^{ext,n} + \tilde{q}_{ijk}^{ext,n+1}\right]\right), \tag{159a}$$

$$U_{ijk}^{n+1} \underset{\text{redistribution}}{\leftarrow} U_{ijk}^{ext,n+1}. \tag{159b}$$

This is the approach we use for all other source terms, including chemical reaction. Note that in this second approach we are also omitting the factor $\bar{\Lambda}/\Lambda$.

In our treatment of solid mechanics [25] we introduced two source terms to enforce constraints. One, a relaxation to enforce $\det(g) = \rho/\rho_0$, is implemented with (159). The other, a diffusive term to enforce $\nabla \times g^T = 0$, follows redistribution (158). Likewise, following redistribution we enforce $\det(\mathcal{F}^p) = 1$ by rescaling.

6.2.4. *Constraints and Extension*

In extended solid cells the constraints $\det(g) = \rho/\rho_0$ and $\det(\mathcal{F}^p) = 1$ may be violated by the extension procedure described in Section 4.1. We therefore rescale g and $\mathcal{F}^p$ in solid extended cells after extension by averaging. The constriant $\nabla \cdot g^T = 0$ may also be violated in the neighborhood of the extended cells, but the extended states are not further modified to enforce this condition.

6.3. Future Directions

Colella [11] describes self-consistent centering schemes by which second-order-accurate truncation errors have been demonstrated for stationary-boundary "Cartesian grid" problems [21, 27]. For moving boundaries one order lower is expected: $\mathcal{O}(h)$ for characteristic variables and $\mathcal{O}(h^2)$ for noncharacteristic ones.

Figure 19 illustrates the self-consistent centering idea in 1D. The quadrature solution of the divergence form PDEs uses fluxes F centered at the centroid of the space–time

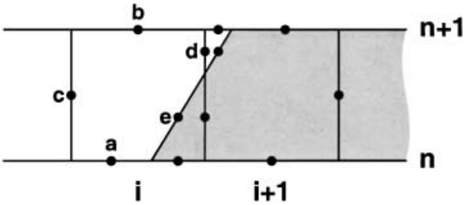

FIG. 19. Centroid-centered second-order fluxes give a first-order-accurate ($\mathcal{O}(h)$) difference scheme.

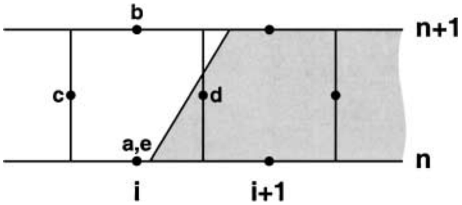

FIG. 20. First-order fluxes, and/or non-centroid-centered fluxes, give a zero-order-accurate ($\mathcal{O}(1)$) difference scheme.

boundaries of the fluid-occupied region of the computational cell, e.g., at points c, d, and e. Data U, points a and b, are likewise centered at the centroid of the occupied fraction, not the cell center. If the centroid-centered fluxes are second-order accurate, then a first-order-accurate difference scheme results.

Material interfaces travel with speed v, the characteristic speed of the "0" family of characteristics. Therefore, those variables associated with the "0" family will have truncation errors of the underlying discretization scheme: $\mathcal{O}(h)$ with the self-consistent quadrature method. Waves of the "+" families and "−" families coincide with the interface cells for a time of order h/λ, where λ is the speed of the "±" characteristic wave. For variables associated with these waves, we therefore expect a higher order of accuracy, i.e., $\mathcal{O}(h^2)$.

The implementation described in this report is illustrated in 1D in Fig. 20. Our data U is cell centered and thus differs from the interpretation of Fig. 19 by terms of order h. Likewise, our cell-face fluxes are time centered (points c, d), not centroid centered, and our interface flux (point e) is cell centered at time t^n (but with time-centered source term contributions). These differences also introduce order h discrepancies, lowering the order of accuracy of the scheme. We therefore expect $\mathcal{O}(1)$ accuracy for "0" family characteristic variables, and $\mathcal{O}(h)$ accuracy for "±" family characteristic variables.

The redistribution of mass δM, an order $\mathcal{O}(h\Delta t)$ quantity (with consistent centering; $\mathcal{O}(\Delta t)$ with inconsistent centering) in the neighborhood of fractional cells expands slightly the neighborhood where local $\mathcal{O}(h)$ ($\mathcal{O}(1)$) truncation errors occur.

To adapt the algorithm described in this report to achieve a higher order of accuracy, several modifications will be necessary. First, a higher order algorithm for the construction of extended states is required in order to calculate second-order-accurate fluxes on mixed-cell face, e.g., at point d in Figs. 19 and 20. Modiano and Colella [27] accomplished this by using one-sided derivatives to fill boundary cells with extrapolated data. Second, it will be necessary to treat conserved data as being centered at the centroid of the fluid-occupied cell fraction, not the cell center. This affects the construction of slopes used in the calculation of time- and face-centered states. Third, fluxes must be centered at the centroids. For face states (e.g., point d), this may be accomplished by interpolation in space using neighboring cell-face fluxes, and interpolation in time making use of the time t^n extended data and time t^{n+1} reference calculation. For interface fluxes, e.g., point e in Figs. 19 and 20 a recalculation of the interface-normal Riemann problem using properly centered left and right states is appropriate.

These same centering ideas might profitably be applied to the volume fraction advection method in Section 3.2. There we used face-centered velocity fields to trace characteristics. Instead, centroid-centered velocity fields might be used. For example, in tracing point p in

Fig. 3 instead of using the face-centered velocity $v_{y,i,j-\frac{1}{2},k}$, use the velocity interpolated to the centroid of `adhe` (cf. (36)):

$$\mathtt{P}_y = y_{j-\frac{1}{2}} - \Delta t \left[v_{y,i,j-\frac{1}{2},k} + \left(\frac{v_{y,i+1,j-\frac{1}{2},k} - v_{y,i,j-\frac{1}{2},k}}{x_{i+1} - x_i} \right) \left(x_{i+\frac{1}{2}} - x_i - v_{x,i+\frac{1}{2},j,k} \Delta t \right) \right]. \tag{160}$$

APPENDIX A: NOMENCLATURE

a	material reference frame coordinate; (94).
A	coefficient matrix in linearized PDE; (133), (79).
A	aperture; Section 3.3.
$\mathcal{A}$	acoustic tensor, (135).
b	plastic reference frame coordinate; (97).
b	stress source term in linearized PDE; (137).
c	expansion coefficient; (140).
c	sound speed; (80).
c_e	elastic constant tensor; (156).
C^e	elastic Green tensor; (100).
C_V	constant volume heat capacity; (104).
d	distance of plane; (18).
e	column of identity matrix; (93).
E	total specific energy, internal plus kinetic; (92), (99).
$\mathcal{E}$	internal energy; Section 5.2.
f	body force vector; (91), (93).
f	yield surface; (118).
F	conserved variable flux; (1).
$\mathcal{F}$	total deformation; (98).
$\mathcal{F}^e$	elastic deformation; (98).
$\mathcal{F}^p$	plastic deformation; (97).
g	inverse total deformation; (94).
G	conserved variable source terms; (1).
G	shear modulus; (114).
h	plastic deformation source term; (93).
h	spatial resolution, Δx; Section 6.
I	identity matrix
I_1, I_2, I_3	invariants of isotropic matrix; (108).
K	work hardening parameter source term; (93).
K_S	bulk modulus; Section 5.2.2, (154), (155).
l	a left eigenvector; Section 4.2.3.
L	matrix of left eigenvectors; (82).
L	level of AMR refinement; Section 5.4.2.
δM	generalized mass difference; (15).
M	modulus of incompressibility; (23a).
n	normal direction; (18).
P	pressure; (91), (103).

q	primitive variable; (133), (79).
q	variable in constitutive model for Grüneisen parameter; (112).
Q	chemical potential energy; (91).
r	a right eigenvector; Section 4.2.3.
r_G, r_K	variable internal to universal equation of state model; Section 5.2.2.
R	ideal gas constant; (102).
R	matrix of right eigenvectors; (81).
R^L	refinement ratio; Section 5.4.2.
$\mathcal{R}$	shorthand for Riemann problem solution; Section 4.2.1.
s	primitive variable source terms; (133), (79).
s	front velocity; (151).
s_e	elastic compliance tensor; (155).
S	specific entropy; Section 5.2.
SOS	sum of squares error; (19).
t	time; (1).
T	temperature; (104).
U	conserved variables; (1).
v	velocity; (91), (93).
v	volume fraction sum; (21).
V	cell volume; Sections 2, 3.2.
$\mathcal{V}$	advected volume; Section 3.2.
w	redistribution weighting factor; (17).
W	mass flux, Rankine-Hugoniot conditions; (143).
x	spatial frame coordinate; (94).
Z	measure of shock strength; (76).
α	superscript denotes material phase.
β	parameter in shear energy equation; (113).
β	measure of shock resolution; (74).
γ	Grüneisen parameter; (112).
γ_p	polytropic gas constant; (102).
ζ	Lagrange undetermined multiplier in plasticity rate models; (119).
ϑ	work hardening modulus; (117).
κ	work hardening parameter; (93).
λ	eigenvalue, wave speed; Sections 4.2.3, 5.3.3.
Λ	volume fraction; Sections 3.1, 3.2, 2.
Λ	diagonal matrix of eigenvectors; (83).
ξ	a distance interpolation function; (72a).
ρ	density; (91), (93).
σ	Cauchy stress; (101).
σ_Y	yield stress; (118).
τ	a tangent direction; (123a), (124a).
τ	a time interpolation function; (45e).
Υ	domain of one computational cell; (3).
Φ	heat source; (91), (93).
χ	flattening parameter; (78).
ω	a tangent direction; (123b), (124b).
Ω	material domain; (1).

ACKNOWLEDGMENTS

We thank E. G. Puckett for an implementation of the 3D ELVIRA algorithm, which includes code contributions from J. Saltzman, J. Helmsen, and R. Pember. D. Trebotich made several contributions to the description and implementation of our solid equation of state model. Computations were performed on Lawrence Livermore National Laboratory's ASCI blue and ASCI white IBM SP supercomputers.

REFERENCES

1. T. J. Ahrens, *Mineral Physics and Crystallography: A Handbook of Physical Constants* (American Geophysical Union, Washington, DC, 1995).
2. D. L. Anderson, *Theory of the Earth* (Blackwell, Boston, 1989).
3. J. Bell, M. Berger, J. Saltzman, and M. Welcome, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, *SIAM J. Sci. Comput.* **15**, 127 (1994).
4. J. B. Bell, P. Colella, and M. L. Welcome, Conservative front-tracking for inviscid compressible flow, In *AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii*, *1991* (AIAA Press, Washington, DC, 1991), p. 814.
5. J. B. Bell, C. N. Dawson, and G. R. Shubin, An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions, *J. Comput. Phys.* **74**, 1 (1988).
6. M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* **82**, 64 (1989).
7. M. J. Berger and I. Rigoutsos, An algorithm for point clustering and grid generation, *IEEE Trans. Systems, Man Cybernet.* **21**, 1278 (1991).
8. H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*, 2nd ed. (Wiley, New York, 1985).
9. I.-L. Chern and P. Colella, *A Conservative Front Tracking Method for Hyperbolic Conservation Laws*, Technical Report UCRL-97200 (Lawrence Livermore National Laboratory, July 1987).
10. P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.* **87**, 171 (1990).
11. P. Colella, Volume-of-fluid methods for partial differential equations, in *Godunov Methods: Theory and Applications*, edited by E. F. Toro (Kluwer, New York, 2001), p. 161.
12. P. Colella and H. M. Glaz, Efficient solution algorithms for the Riemann problem for real gases, *J. Comput. Phys.* **59**, 264 (1985).
13. P. Colella, H. M. Glaz, and R. E. Ferguson, *Multifluid Algorithms for Eulerian Finite Difference Methods* (1996).
14. P. Colella, D. T. Graves, D. Modiano, D. B. Serafini, and B. van Straalen, *Chombo Software Package for AMR Applications*, Technical Report (Lawrence Berkeley National Laboratory, 2000), http://seesar.lbl.gov/anag/chombo/.
15. P. Colella and L. F. Henderson, The von Neumann paradox for the diffraction of weak shock waves, *J. Fluid. Mech.* **213**, 71 (1990).
16. P. Colella and P. R. Woodward, The piecewise parabolic method (PPM) for gas dynamical simulations, *J. Comput. Phys.* **54**, 174 (1984).
17. W. Y. Crutchfield and M. Welcome, Object-oriented implementation of adaptive mesh refinement algorithms, *Sci. Program.* **2**(4), 145 (1993).
18. W. Fickett and L. M. Scherr, *Numerical Calculation of the Cylinder Test*, Technical Report LA-5906 (Los Alamos National Laboratory, 1975).
19. J. Hilditch and P. Colella, A front tracking method for compressible flames in one dimension, *SIAM J. Sci. Comput.* **16**, 755 (1995).
20. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
21. H. Johansen and P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, *J. Comput. Phys.* **147**, 60 (1998).

22. D. B. Kothe, W. J. Rider, S. J. Mosso, J. S. Brock, and J. I. Hochstein, *Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions*, AIAA 34th Science Meeting, Reno, Nevada, 1996, Paper 96-0859 (1996).

23. E. H. Lee and D. T. Liu, Finite-strain elastic-plastic theory with application to plane–wave analysis, *J. Appl. Phys.* **38**, 19 (1967).

24. A. J. Majda and A. Bourlioux, Theoretical and numerical structure of unstable detonations, *Philos. Trans. R. Soc. London* **350**, (1995).

25. G. H. Miller and P. Colella, A high-order Eulerian Godunov method for elastic-plastic flow in solids, *J. Comput. Phys.* **167**, 131 (2001).

26. G. H. Miller and E. G. Puckett, A high-order Godunov method for multiple condensed phases, *J. Comput. Phys.* **128**, 134 (1996).

27. D. Modiano and P. Colella, *A Higher-Order Embedded Boundary Method for Time-Dependent Simulation of Hyperbolic Conservation Laws*, Technical Report, LBNL-45239 (Lawrence Berkeley National Laboratory, 2000).

28. M. Mooney, A theory of large elastic deformation, *J. Appl. Phys.* **11**, 582 (1940).

29. W. F. Noh, CEL: A time-dependent, two-space-dimensional, coupled Eulerian–Lagrange code, *Methods Comput. Phys.* **3**, 117 (1964).

30. W. F. Noh and P. R. Woodward, *SLIC (Simple Line Interface Calculation)*, Springer Lecture Notes in Physics (Springer-Verlag, Berlin, 1976), Vol. 59.

31. J. F. Nye, *Physical Properties of Crystals* (Clarendon, Oxford, 1985).

32. R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* **120**, 278 (1995).

33. J. E. Pilliod, *A Second-Order Unsplit Method for Modeling Flames in Two-Dimensional Compressible Flow*, PhD thesis (University of California, Davis, 1996).

34. J. E. Pilliod and E. G. Puckett, An unsplit, second-order accurate Godunov method for tracking deflagrations and detonations, in *Proceedings of the 21st International Symposium on Shock Waves, edited by Allan Paul Great Keppel Island, Queensland, Australia, July 20–25, 1997*, (Pauther, Fyshwick, Australia, 1997), p. 1053.

35. J. E. Pilliod and G. E. Puckett, *Second-Order Accurate Volume-of-Fluid Algorithms for Tracking Material Interfaces*, Technical Report LBNL-40745 (Lawrence Berkeley National Laboratory, 1997).

36. B. J. Plohr and D. H. Sharp, A conservative formulation for plasticity, *Adv. Appl. Math.* **13**, 462 (1992).

37. E. G. Puckett and J. S. Saltzman, A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics, *Physica D* **60**, 84 (1992).

38. R. S. Rivlin, Large elastic deformations of isotropic materials. IV. further developments of the general theory, *Proc. R. Soc. London Ser. A* **241**, 379 (1948).

39. J. H. Rose, J. Ferrante, and J. R. Smith, Universal binding energy curves for metals and bimetallic interfaces, *Phys. Rev. Lett.* **47**, 675 (1981).

40. J. Saltzman, An unsplit 3D upwind method for hyperbolic conservation laws, *J. Comput. Phys.* **115**, 153 (1994).

41. J. C. Simo and T. J. R. Hughes, *Computational Inelasticity* (Springer-Verlag, New York, 1997).

42. J. A. Trangenstein and P. Colella, A higher-order Godunov method for modeling finite deformation in elastic-plastic solids, *Commun. Pure Appl. Math.* **44**, 41 (1991).

43. B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.* **32**, 101 (1979).

44. P. Vinet, J. Ferrante, J. H. Rose, and J. R. Smith, Compressibility of solids, *J. Geophys. Res. B* **92**, 9319 (1987).

45. P. Vinet, J. Ferrante, J. R. Smith, and J. H. Rose, A universal equation of state for solids, *J. Phys. C* **19**, L467 (1986).

46. M. L. Wilkins, Calculation of elastic-plastic flow, in *Methods in Computational Physics*, edited by B. Alder, S. Fernbach, and M. Rotenberg (Academic Press, New York, 1964), Vol. 3, p. 211.

47. P. R. Woodward, Piecewise parabolic methods for astrophysical fluid dynamics, in *Astrophysical Radiation Hydrodynamics*, edited by K.-H. Winkler and M. Norman (Reidel, Dordrecht, 1986), p. 245.