AN ADAPTIVE MULTIFLUID INTERFACE-CAPTURING METHOD FOR COMPRESSIBLE FLOW IN COMPLEX GEOMETRIES*

Jeffrey A. Greenough, Vincent Beckner, Richard B. Pember, William Y. Crutchfield, John B. Bell, and Phillip Colella
Lawrence Livermore National Laboratory,
P.O. Box 808, L-316,
Livermore, CA 94550

Abstract

We present a numerical method for solving the multifluid equations of gas dynamics using an operator-split second-order Godunov method for flow in complex geometries in two and three dimensions. The multifluid system treats the fluid components as thermodynamically distinct entities and correctly models fluids with different compressibilities. This treatment allows a general equation-of-state (EOS) specification and the method is implemented so that the EOS references are minimized. The current method is complementary to volume-of-fluid (VOF) methods in the sense that a VOF representation is used, but no interface reconstruction is performed. The Godunov integrator captures the interface during the solution process. The basic multifluid integrator is coupled to a Cartesian grid algorithm that also uses a VOF representation of the fluid-body interface. This representation of the fluid-body interface allows the algorithm to easily accommodate arbitrarily complex geometries. The resulting single grid multifluid-Cartesian grid integration scheme is coupled to a local adaptive mesh refinement algorithm that dynamically refines selected regions of the computational grid to achieve a desired level of accuracy. The overall method is fully conservative with respect to the total mixture. The method will be used for a simple nozzle problem in two-dimensional axisymmetric coordinates.

*This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract W-7405-Eng-48. Support under contract W-7405-Eng-48 was provided by the Applied Mathematical Sciences Program and the HPCC Grand Challenge Program of the Office of Scientific Computing at DOE and by the Defense Nuclear Agency under IACRO 95-2045. Prof. Colella was supported at UC Berkeley by DARPA and the National Science Foundation under grant DMS-8919074; and by a National Science Foundation Presidential Young Investigator award under grant ACS-8958522; and by the Department of Energy High Performance Computing and Communications Program under grant DE-FGO3-92ER25140.

Introduction and Overview

Compressible flows in which the fluid is made up of a number of thermodynamically distinct species, an extreme system being liquid-gas, arise in a wide variety of engineering applications requiring realistic geometries. In this paper we describe an algorithm for modeling inviscid compressible multifluid flows containing complex geometries in two and three space dimensions. The basic algorithm is an operator split second-order Godunov method used to solve the Euler equations for multifluid flow. The algorithm captures rather than tracks the interfaces between distinct materials while maintaining a volume-of-fluid (VOF) representation of the constituent materials. That is, the interface is obtained during the coarse of the Godunov solution without recourse to an interface reconstruction. As such, the present multifluid method provides a complementary approach to VOF interface tracking algorithms. While there are numerous approaches to tracking interfaces, we shall only mention those in the class of VOF techniques. The simplest VOF interface reconstruction algo-

rithms are those based on the Simple Line Interface Calculation (SLIC) method [12]. There are numerous other first-order variations on this such as the center of mass method [16], central differences [11], and Youngs' method [17]. To obtain a second-order reconstruction, there is an algorithm based on a least squares fit to the local volume fractions profile [15]. In all of the interface tracking methods. a sub-grid scale method of reconstructing the interface must be used to compute the current location of the material interface and as a result the interface remains sharp. The primary disadvantage to using these methods is the expense. Having to reconstruct the interface adds extra computation over simply advancing the flow in time. In addition, the reconstruction process is performed on a cell by cell basis hence requiring some coding sophistication so that vectorization can be achieved on modern supercomputers. However, if the number of cells occupied by the interface is small, then this cost may be minimized. When complex geometry is included, there are added difficulties in coupling a reconstruction algorithm more complex than SLIC.

Another point of consideration is that interface tracking techniques may not be appropriate for all problems. If the interface is initially sharp and retains its integrity over time then tracking the interface is appropriate. However, if the fluids become mixed either by diffusion, by large-scale motions or are initially mixed, then treating the interface as a discontinuity gives a representation that is inconsistent and probably meaningless. This leads to consideration of the current method since it does not require tracking the interface, yet has the ability to distinguish thermodynamically distinct fluid components and compute mixture properties using the VOF formulation. Furthermore, the ability to describe such flows in arbitrarily complex geometry provides a computational capability important for real world engineering applications. We refer to the methodology for treating complex geometry as a Cartesian grid method [5].

The basic multifluid method is coupled to a Cartesian grid algorithm which also uses a VOF representation of the fluid-body interface. This representation of the fluid-body interface allows the algorithm to easily accommodate arbitrarily complex geometries. The resulting single grid multifluid-Cartesian grid integration scheme is coupled to a local Adaptive Mesh Refinement (AMR) This is a code based on the original ideas found in [4] and later in [3]. The current version [1], [10] is an object-oriented (C++) code framework for managing a hierarchy of logically rectangular refined grids that is hybridized with Fortran routines that provide low level support and integrator instantiation. In regions where errors are deemed unacceptable, a grid is locally refined. This has the two-fold result of increasing accuracy locally where it is required as well as concentrating the computational effort where it is needed.

What follows is a description of the multifluid VOF representation and the predictor-corrector Godunov solution in one dimension. Then there is an overview of the previously documented Cartesian grid method, followed by a discussion of the modifications necessary to couple the multifluid-Cartesian grid method into AMR. A simple nozzle problem in axisymmetric coordinates illustrates the adaptive code results.

Multifluid Algorithm

VOF Representation

The basic assumptions of the multifluid formulation are that there is pressure equilibrium among all fluid components within a cell and there is a single velocity vector for each cell, independent of the mixture. A rigorous derivation of this system is given elsewhere [8] and is not repeated here. The first assumption says that $p^\alpha(\mathbf{x},t) = p(\mathbf{x},t)$, or that the value of the pressure is independent of the fluid component. This is physically reasonable since across a contact discontinuity (material interface) there is no pressure jump and the partial pressures within a mixed cell must be equal. Under these assumptions, the Euler equations for a multifluid system are

$$\frac{\partial f^\alpha}{\partial t} + \nabla \cdot (\boldsymbol{u} f^\alpha) = f^\alpha \frac{\hat{\Gamma}}{\Gamma^\alpha} \nabla \cdot \boldsymbol{u} \qquad (0.1)$$

$$\frac{\partial}{\partial t}(f^\alpha \rho^\alpha) + \nabla \cdot (\boldsymbol{u} f^\alpha \rho^\alpha) = 0 \qquad (0.2)$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}\rho) + \nabla p = 0 \qquad (0.3)$$

$$\frac{\partial}{\partial t}(f^\alpha \rho^\alpha E^\alpha) + \nabla \cdot (\boldsymbol{u} f^\alpha \rho^\alpha E^\alpha) \qquad (0.4)$$

$$+ p f^\alpha \frac{\hat{\Gamma}}{\Gamma^\alpha} \nabla \cdot \boldsymbol{u} + f^\alpha \frac{\rho^\alpha}{\rho} \boldsymbol{u} \cdot \nabla p = 0$$

where f^α, ρ^α, and E^α are the volume fraction, density, and total energy density of fluid component α. The volume fraction is defined as $f^\alpha = \Lambda_\alpha / \Lambda$ where Λ is the volume of the cell and Λ_α is the volume of the cell occupied by fluid α. Γ^α is the sound speed γ for fluid α, and $\hat{\Gamma} = 1/\sum_\alpha (f^\alpha/\Gamma^\alpha)$ which represents the fraction weighted sound speed γ for the mixture.

The pressure that appears in the above system is defined to be a thermodynamically consistent pressure given as $p = \sum_\alpha (f^\alpha p^\alpha)$, where p^α is the partial pressure of component α. Note that the formulation is sufficiently general to allow real gas EOS systems described by pressure given as a function of density and internal energy.

Godunov Implementation

First rewrite the above system of equations in vector form, in one spatial dimension, in anticipation of an operator split implementation, and for two fluids α and β.

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial F(\mathbf{Q})}{\partial x} = \mathbf{S}(\mathbf{Q}, x) \qquad (0.5)$$

Here, the state vector is given as

$$\mathbf{Q}(x,t) = (\rho^\alpha f^\alpha, \rho u, \rho^\alpha f^\alpha E^\alpha, f^\alpha, \qquad (0.6)$$

$$\rho^\beta f^\beta, \rho^\beta f^\beta E^\beta, f^\beta)^t$$

$F(\mathbf{Q})$ is the flux vector given by

$$F(\mathbf{Q}) = (\rho^\alpha f^\alpha u, (\rho u)u, \rho^\alpha f^\alpha E^\alpha u, f^\alpha u, \qquad (0.7)$$

$$\rho^\beta f^\beta u, \rho^\beta f^\beta E^\beta u, f^\beta u)^t$$

and the source term on the right hand side is given by

$$\mathbf{S}(\mathbf{Q}, x) = (0, \nabla p, \qquad (0.8)$$

$$-(p f^\alpha \frac{\hat{\Gamma}}{\Gamma^\alpha} \nabla \cdot \boldsymbol{u} + f^\alpha \frac{\rho^\alpha}{\rho} \boldsymbol{u} \cdot \nabla p), 0, 0,$$

$$p f^\beta \frac{\hat{\Gamma}}{\Gamma^\beta} \nabla \cdot \boldsymbol{u} + f^\beta \frac{\rho^\beta}{\rho} \boldsymbol{u} \cdot \nabla p, 0)^t$$

Notice that the fluid components are treated in a symmetric fashion. In most tracking implementations, the total densities and energies are solved for along with partial densities, energies and volume fractions for only one of the fluids. The state for the second fluid is obtained by subtraction of the partial values from the totals. In our formulation each component is treated separately, but the method is designed to have the multifluid results reduce upon summing over α, for the case of equal sound speed γ, to the single fluid algorithm. With respect to equations 0.5 through 0.8 given above, summing the partial energy equations, over both fluids, gives conservation of total energy for the mixture.

The operator splitting has the following form in two-dimensions

$$\mathbf{Q}^{n+2} = L_x L_y L_y L_x (\mathbf{Q}^n) \qquad (0.9)$$

and in three-dimensions as

$$\mathbf{Q}^{n+2} = L_x L_y L_z L_z L_y L_x (\mathbf{Q}^n) \qquad (0.10)$$

The operator L_ζ is the sweep in the ζ coordinate direction and at the end of the cycle, the solution is formally second-order accurate.

The scheme used to integrate the above system of equations is a second-order Godunov method. The algorithm is based on the general higher-order Godunov methodology described in [7] and [2]. In general terms it can be thought of as a predictor-corrector scheme where cell centered primitive values are traced along characteristics to the half-time level at cell edges using a higher order slope approximation to the local state. The tracing procedure takes data defined at x_j^n to $x_{j+1/2}^{n+1/2}$ to define the left (L) state. The right (R) state is obtained by tracing data at x_j^n to $x_{j-1/2}^{n+1/2}$. Then a local Riemann problem is approximated at each cell edge at the half-time level given the states L and R. The solution to the Riemann problem is used to compute fluxes at the

half-time level that are finally used to update the solution as written in conservative form. In general, the conservative update can be written as

$$\mathbf{Q}_j^{n+1} = \mathbf{Q}_j^n - \frac{\Delta t}{\Delta x}[F(\mathbf{Q}^*{}_{j+1/2})]_j \qquad (0.11)$$

$$+\frac{\Delta t}{\Delta x}\mathbf{S}(\mathbf{Q}^*{}_{j+1/2}, x_{j+1/2})$$

Note that the source term is time centered and depends on $\mathbf{Q}^*$, the approximation to the Riemann problem, or Godunov state, which exists at $t_{n+1/2}$. The notation, $[\zeta]_j$, is the flux difference of ζ over cell j, i.e. $[\zeta]_j = \zeta_{j+1/2} - \zeta_{j-1/2}$

The volume fraction equation and source terms due to differences in compressibility are discretized in a special way following the general form given in [8] and are specified below. Also, the source terms due to geometric sources are discretized in a straightforward way as above.

To summarize, the solution procedure for a single grid implementation has the following steps: (1) construct limited central difference approximations to traced state slope, (2) trace along characteristics to half-time level at cell edges to obtain L and R states, (3) solve the local Riemann problem approximately, at the cell edges (4) perform a conservative update of the solution using the results generated in the Riemann solution.

Characteristic Analysis

For the characteristic tracing step, we need to perform a characteristic analysis of the system of equations. To accomplish this, we rewrite the system in quasilinear form in terms of the primitive variables, $\mathbf{q}(x,t) = \left(\rho^\alpha f^\alpha, u, p, \rho^\alpha f^\alpha e^\alpha, f^\alpha, \rho^\beta f^\beta, \rho^\beta f^\beta e^\beta, f^\beta\right)^t$ where e^α is the internal energy per unit mass of fluid α and for an ideal gas is given by $e^\alpha = \frac{p^\alpha}{\rho^\alpha f^\alpha(\gamma_\alpha - 1)}$ and likewise for fluid β. It is written as

$$\frac{\partial \mathbf{q}}{\partial t} + \mathcal{A}\frac{\partial \mathbf{q}}{\partial x} = \mathbf{s}(q,x) \qquad (0.12)$$

where $\mathcal{A} = \partial F/\partial \mathbf{q}$

Note that the differential compressibility source terms are absorbed into the quasi-linear form. The remaining sources contained in, $\mathbf{s}(q,x)$, are due to geometric factors if one is using general curvilinear coordinates.

The first step in the analysis requires determining the eigenvalues and corresponding left and right eigenvectors for the matrix $\mathcal{A}$. The eigenvalues are given as

$$\lambda_+ = u + c, \lambda_- = u - c \qquad (0.13)$$

and

$$\lambda_{0i} = u, i = 1,2,3,4,5,6. \qquad (0.14)$$

where mixture sound speed is defined as $c^2 = \hat{\Gamma}p/\rho$. The former are the right and left propagating acoustic waves moving with the local flow. The latter eigenvalue has multiplicity 6 instead of 2 for the single fluid Euler equation. All the eigenvalues are real so that the system above is classified as hyperbolic.

The corresponding left eigenvectors, $\mathbf{l}_i$, and right eigenvectors, $\mathbf{r}_i$ are computed for $i = u + c, u - c, u_{01}, \ldots, u_{06}$ and orthonormalized so that $\mathbf{l}_i^t \cdot \mathbf{r}_j = \delta_{ij}$, where δ is the Kronecker delta function.

Characteristic Tracing

To compute the left state (L) for cell edge $j + \frac{1}{2}$ by characteristic tracing, we begin by Taylor series expanding the solution about the jth cell center. That is,

$$\mathbf{q}_{j+\frac{1}{2},l}^{n+\frac{1}{2}} = \mathbf{q}_j^n + \frac{\Delta x}{2}(\frac{\partial \mathbf{q}}{\partial x})_j^n + \frac{\Delta t}{2}(\frac{\partial \mathbf{q}}{\partial t})_j^n \qquad (0.15)$$

Using the partial differential equation for $\mathbf{q}$ gives

$$\mathbf{q}_{j+\frac{1}{2},l}^{n+\frac{1}{2}} = \mathbf{q}_j^n + \frac{1}{2}(I - \frac{\Delta t}{\Delta x}\mathcal{A}(\mathbf{q}_j^n))(\frac{\partial \mathbf{q}}{\partial x})_j^n \Delta x \qquad (0.16)$$

$$+\frac{\Delta t}{2}\mathbf{s}(q_j^n, x_j)$$

where $(\cdot)_j^n$ denotes evaluation at (t^n, x_j).

The vector, $(\frac{\partial \mathbf{q}}{\partial x})_j^n \Delta x$, is the slope of the local primitive variables. A fourth order approximation to the slope at cell centers is constructed and then limited in a monotone fashion as given by [7]. Denote the limited slope as $\delta^{lim}\mathbf{q}_j^n$ such

that $\delta^{lim}\mathbf{q}_j^n \approx (\frac{\partial \mathbf{q}}{\partial x})_j^n \Delta x$. Now we represent these slopes in an expansion in terms of the right-eigenvectors of the linearized system as

$$\delta^{lim}\mathbf{q}_j^n = \sum_{i=(u+c,u-c;u_{01},\ldots,u_{06})} \alpha_i \mathbf{r}_i \qquad (0.17)$$

The α_i are the expansion coefficients and are set according to $\alpha_i = \mathbf{l}_i \cdot (\delta^{lim}\mathbf{q}_j^n)$.

The current procedure of limiting the raw slopes and then defining the expansion coefficients is in contrast to the method used by [13] where the expansion coefficients are computed and then limited by the above procedure.

Since the system is hyperbolic and has a complete set of eigenvectors, one can construct a similarity transformation so that, $\mathcal{A} = R\Lambda R^{-1}$, where $\Lambda = [\lambda_i]$ is a diagonal eigenvalue matrix and R is the matrix with columns the right eigenvectors. Also, in order to limit the characteristic tracing to directions that contribute to the left (L) state at edge $j + 1/2$ from cell center j, we introduce a projection operator as

$$P_L(w_j) = \sum_{k:\lambda_{k,j}>0} (\mathbf{l}_{k,j} \cdot w_j)\mathbf{r}_{k,j} \qquad (0.18)$$

where the notation $\zeta_{k,j}$ means the kth element of ζ evaluated at cell j.

Now substituting the limited slope approximation, $\delta^{lim}\mathbf{q}_j \approx (\frac{\partial \mathbf{q}}{\partial x})_j^n \Delta x$ and its expansion in terms of right eigenvectors into equation (2.26) and applying the projection operator for left states gives

$$\mathbf{q}_{j+\frac{1}{2},L}^{n+\frac{1}{2}} = \mathbf{q}_j^n + \frac{1}{2}\sum_{k:\lambda_{k,j}>0} (I - \frac{\Delta t}{\Delta x}\lambda_{k,j})\alpha_{k,j}\mathbf{r}_{k,j} + \frac{\Delta t}{2}\mathbf{s}(q_j^n, x_j) \qquad (0.19)$$

The projection operator for the right state is given as

$$P_R(w_j) = \sum_{k:\lambda_{k,j}<0} (\mathbf{l}_{k,j} \cdot w_j)\mathbf{r}_{k,j} \qquad (0.20)$$

and we obtain an expression for the right state (R) as traced from cell j as

$$\mathbf{q}_{j+\frac{1}{2},l}^{n+\frac{1}{2}} = \mathbf{q}_j^n - \frac{1}{2}\sum_{k:\lambda_{k,j}<0} (I + \frac{\Delta t}{\Delta x}\lambda_{k,j})\alpha_{k,j}\mathbf{r}_{k,j} + \frac{\Delta t}{2}\mathbf{s}(q_j^n, x_j) \qquad (0.21)$$

Riemann Solution

Given the traced states, the predictor step is completed by solving the local Riemann problem to obtain the Godunov states. Instead of a full Riemann solver, we adopt an approximate solution that meets the design goal of avoiding EOS evaluations when building the fluxes from the Godunov states necessary for the conservative update. The choice of primitive variables, $\mathbf{q}$, facilitates this design point. The current approximate solver is a simplified version of [9], in the spirit of [2] and given in [8]. The resulting algorithm requires no EOS calls and is nearly twice as fast as the one given in [9].

Conservative Update

Having obtained the Godunov states from the Riemann solution, we can construct the updated solution. We begin with the update for the volume fractions. This algorithm is based on the formulation presented by [8]. The volume fraction update is performed in two steps. The first satisfies the linear advection equation (neglecting the source term). The second takes into account these source terms. The first step is given as

$$\tilde{f}_j^\alpha = f_j^{\alpha,n} - \frac{\Delta t}{\Delta x}[F_4(\mathbf{Q}^*)]_j \qquad (0.22)$$

$$\tilde{f}_j^\beta = f_j^{\beta,n} - \frac{\Delta t}{\Delta x}[F_7(\mathbf{Q}^*)]_j \qquad (0.23)$$

The subscript on the flux vectors denotes that component of it and $\mathbf{Q}^*$ is the Godunov state.

The next step is to calculate the effects of the differential compressibility source on $\tilde{f}^\alpha$ and $\tilde{f}^\beta$ in order to obtain $f^{\alpha,n+1}$ and $f^{\beta,n+1}$. To this end, write the full update (advection plus sources) as

$$f^{\alpha,n+1} = \tilde{f}^\alpha + \Delta t f^\alpha \frac{\hat{\Gamma}}{\Gamma^\alpha}\nabla \cdot (u) \qquad (0.24)$$

Summing this equation over the α and β fluids gives an approximation for $\nabla \cdot (u)$, which we denote as Du. Explicitly this is

$$Du = \frac{1}{\Delta t}\left(1 - \sum_{i=(\alpha,\beta)} \tilde{f}^i\right) \tag{0.25}$$

Therefore, we arrive at the final update as

$$f_j^{\alpha,n+1} = \tilde{f}_j^{\alpha,n}\left(1 + \frac{\hat{\Gamma}}{\Gamma^\alpha}\Delta t(Du)_j\right) \tag{0.26}$$

and likewise for $f_j^{\beta,n+1}$. Note that this approximation has the property that it enforces the constraints that $\sum_{i=(\alpha,\beta)} f^{i,n+1} = 1$ and $0 \leq f^{\alpha,n+1} \leq 1$.

The remainder of the fluid components are first updated using standard conservative differencing for the advective portion. The partial energy equations have source terms and the discrete form of these, taken from [8], will be given below. The conservative update for the partial density is

$$(f^\alpha \rho^\alpha)_j^{n+1} = (f^\alpha \rho^\alpha)_j^n - \frac{\Delta t}{\Delta x}[F_1(\mathbf{Q}^*)]_j \tag{0.27}$$

and likewise for $(f^\beta \rho^\beta)_j^{n+1}$. The momentum equation update is given as

$$(\rho u)_j^{n+1} = (\rho u)_j^n - \frac{\Delta t}{\Delta x}[F_2(\mathbf{Q}^*)]_j + \frac{\Delta t}{\Delta x}[p^*]_j \tag{0.28}$$

Finally, the partial energy equation update, with a similar expression for the second fluid, is

$$(f^\alpha \rho^\alpha E^\alpha)_j^{n+1} = (f^\alpha \rho^\alpha E^\alpha)_j^n - \frac{\Delta t}{\Delta x}[F_3(\mathbf{Q}^*)]_j \frac{\Delta t}{\Delta x}\mathbf{S}_3(\tilde{f}_j^\alpha, \mathbf{Q}_j^*) \tag{0.29}$$

The source term $\mathbf{S}_3$ is given as

$$\mathbf{S}_3(\tilde{f}^\alpha, \mathbf{Q}^*) = \tilde{f}^\alpha \bar{p}\frac{\tilde{\Gamma}}{\Gamma^\alpha}[u] + \frac{(f^\alpha \rho^\alpha)_j^{n+1}}{\rho^{n+1}}\bar{u}[p] \tag{0.30}$$

The term, $\tilde{\Gamma}$, is given by $\tilde{\Gamma} = 1/\sum_{i=\alpha,\beta}(f^i/\Gamma^i)$ The overbar denotes averaging of the Godunov state to obtain a cell centered value at the half-time level. The update for $(f^\beta \rho^\beta E^\beta)_j^{n+1}$ follows in a similar fashion from the above equation. It is worth noting that the above discretization upon summing over the two fluids reduces to the single fluid total energy for the case of equal sound speed Γ.

Cartesian Grid Overview

The Cartesian grid method used is based on a VOF representation of the boundary. The present discussion will serve as an overview; the details are presented in [14]. With respect to the Cartesian grid method, there are volume fractions (not to be confused with the above multifluid volume fractions) that denote the volume of fluid that is outside of a body, or equivalently, inside the flowfield region. These together with aperatures, or area fractions of cell faces that lie inside the flow domain, complete the description of the geometry. In [14], the underlying Godunov integration scheme was an unsplit version. In this work, the Cartesian grid algorithm was converted to an operator split one, which significantly increased performance. Also, this facilitated incorporation of the multifluid integration algorithm.

As a setup procedure for the method, extended states, $\mathbf{Q}^{ext}$, must be defined in the body. These values, $\mathbf{Q}^{ext}$, define sensible values for cells in the body near the body surface. This ensures that the finite difference stencil will compute reasonable fluxes for the cells near the body surface.

Now the above multifluid integrator is used to return the fluxes and the discretized form of the sources terms necessary to update the solution one timestep. Away from the boundaries, these fluxes are sufficient to determine the new solution, but at this point no distinction is made between cells inside or outside of the body. These fluxes are used to update the extended states to yield, $\mathbf{Q}^{ext,n+1}$.

For mixed cells, cells containing both body and fluid, a local modified Riemann solution is computed. A local approximation to the body normal is determined and left and right states are specified by $\mathbf{Q}^n$. The Riemann solution yields a frontal flux across the body. The fluxes returned from the multifluid integration together with the

frontal flux determine a preliminary solution denoted as $\tilde{\mathbf{Q}}$.

Now we define the effect of the body as an incremental change from $\mathbf{Q}^{ext,n+1}$ by defining

$$\delta\mathbf{M}^{ijk} = \Lambda_{ijk}\mathbf{Q}^{ext,n+1}_{ijk} - \Lambda_{ijk}\tilde{\mathbf{Q}}_{ijk} \qquad (0.31)$$

Sufficiently far from the body, it is seen that $\delta\mathbf{M}$ vanishes. So this leads to the update

$$\mathbf{Q}^{n+1}_{ijk} = \mathbf{Q}^{ext,n+1}_{ijk} + \frac{\delta\mathbf{M}^{ijk}}{\Lambda_{ijk}} \qquad (0.32)$$

For Cartesian grid mixed cells, Λ_{ijk} can be arbitrarily small which would require an excessive timestep restriction for the method to remain stable. We use the algebraic redistribution ideas of [6] to modify its discretization in mixed cell to be both stable and conservative. In particular, we perform a preliminary update of the form

$$\tilde{\mathbf{Q}}_{ijk} = \mathbf{Q}^{ext,n+1}_{ijk} + \delta\mathbf{M}^{ijk} \qquad (0.33)$$

which does not have a CFL restriction but violates discrete conservation. Then we redistribute $(1-\Lambda_{ijk})\delta\mathbf{M}^{ijk}/\Lambda_{ijk}$ onto the grid in neighboring cells inside the flow domain and regain conservation. Note that this simplified procedure is only correct for reflecting wall boundary conditions.

AMR Considerations

When coupling any integration algorithm to AMR, we must be concerned with retaining global conservation [3]. This is an important issue because as the problem domain is covered by a hierarchy of refined grid patches, there will be fluxes across coarse/fine grid boundaries. In addition, the redistribution procedure outlined above provides an additional mechanism for moving state quantities across the boundaries. Also, the multifluid integrator has differential compressibility source terms which communicate across coarse/fine grid boundaries. All three of these sources of inter-grid-level communication across grid boundaries must be treated correctly to maintain conservation.

The basic single fluid AMR implementation maintains global conservation by using a procedure known as refluxing. Basically, the fluxes generated in the Godunov solution are saved and accumulated for fine grid faces that border coarse grid cells that are not themselves refined. The difference between these accumulated fluxes and the coarse grid flux for the face is used to update the coarse grid solution. In this way, the fluxes into a coarse grid cell, bordering a fine grid, are consistently approximated, using fluxes taken from the underlying fine grid rather than the coarse grid flux.

The refluxing procedure for the multifluid AMR version is modified to account for the differential compressibility sources terms. Additional differential quantities are accumulated including the average Godunov velocity as represented on the fine grid. Note that this velocity is normal to the cell face undergoing refluxing. In addition, we need access to the current state on the coarse grid cell so that $\hat{\Gamma}$ can be computed. In general, this is only accessible through the EOS. This implies an added computational expense, but the result is global conservation with respect to the mixture.

The additional terms needed to account for the source terms in the volume fraction, and partial energy equations are given as

$$(\rho^\alpha f^\alpha)^C = (\rho^\alpha f^\alpha)^C + iface\frac{\Delta t^C}{\Delta x^C}(\delta F_1) \qquad (0.34)$$

$$(\rho u)^C = (\rho u)^C + iface\frac{\Delta t^C}{\Delta x^C}(\delta F_2 + \delta p) \qquad (0.35)$$

$$(\rho^\alpha f^\alpha E^\alpha)^C = (\rho^\alpha f^\alpha E^\alpha)^C + iface\frac{\Delta t^C}{\Delta x^C} \qquad (0.36)$$

$$\left((\delta F_3) + (\delta(up) - \bar{U}\delta p)f^\alpha\frac{\hat{\Gamma}^C}{\Gamma^{\alpha,C}} + \bar{U}(\delta p)\frac{\rho^\alpha f^\alpha}{\rho}\right)$$

$$f^{\alpha,C} = iface\frac{\Delta t^C}{\Delta x^C}\left((\delta F_4) - (\delta u)f^{\alpha,C}\frac{\hat{\Gamma}}{\Gamma^\alpha}\right) \qquad (0.37)$$

The second partial energy equations follows from above. The factor, $\delta(\zeta)$, is the increment of quantity ζ as computed by taking the difference between the value computed for the coarse grid and

the sum over the underlying fine grid faces. $\bar{U}$ is the average Godunov normal velocity described above. The unit function $iface$ is positive or negative depending on whether or not the flux is oriented in the positive or negative coordinate direction, respectively.

The procedure to retain global conservation for the Adaptive Cartesian grid algorithm is called re-redistribution. In addition to the usual re-fluxing that occurs at coarse/fine grid boundaries away from bodies, there is an additional step necessary to account for state movement induced by the redistribution procedure. This procedure is documented in [14] and not repeated here.

Test Problems

We consider a simple nozzle problem as a test of the algorithm. It can be considered as a simple model for an orifice issuing into free space, similar in design to a rocket exhaust nozzle. This problem is run in two-dimensions with axisymmetric geometry. There is a straight tube section followed by the nozzle section that opens into the ambient medium. The inlet boundary conditions are straight flow down the tube with inlet Mach number, $M_{in} = 2.04$, inlet density to ambient density ratio, $\frac{\rho_{in}}{\rho_\infty} = 0.66$, and inlet pressure to ambient pressure ratio, $\frac{p_{in}}{p_\infty} = 2$. The introduced fluid is pure fluid α, issuing into ambient fluid β. The nozzle is approximated by a parabola with exponent 3, i.e. inlet is in the z direction and the nozzle is given as $z \approx r^3$. In the figure, the quantity $\rho^\alpha f^\alpha$ is shown. The calculation is adaptive with two levels of refinement. Each level of refinement is a factor of two finer over the previous level. The refinement is set to tag the multifluid cells (multifluid mixed cells) as well as flow discontinuities such as shocks and contacts. Note the leading bow shock that is refined by not visualized as it exists in the fluid β.

References

[1] J.B Bell, M.J. Berger, J.S. Saltzman, and M. Welcome. Three dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comp.*, to appear. UCRL JC-108794, Lawrence Livermore National Laboratory, December, 1991.

[2] J.B. Bell, P. Colella, and J. Trangenstein. Higher order Godunov methods for general systems of hyperbolic conservation laws. *J. Comput. Phys.*, 82:362–397, 1989.

[3] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[4] M.J. Berger and Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[5] M.J. Berger and R.J. LeVeque. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. In *Proceedings, AIAA 9th Computational Fluid Dynamics Conference*. Buffalo, New York, June 14-16, 1989, p.1-7.

[6] I. Chern and P. Colella. A conservative front tracking method for hyperbolic conservation laws. UCRL JC-97200, Lawrence Livermore National Laboratory, July, 1987, submitted to J. Comput. Phys.

[7] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM J. Sci. Stat. Comput.*, 6:104–117, 1985.

[8] P. Colella, R.E. Ferguson, and H.M. Glaz. Multifluid algorithms for Eulerian finite difference methods, in preparation.

[9] P. Colella and H.M. Glaz. Efficient solution algorithms for the Riemann problem for real gases. *J. Comput. Phys.*, 59:264–289, 1985.

[10] W.Y. Crutchfield and M.L. Welcome. Object-oriented implementation of adaptive mesh refinement algorithms. *Sci. Prog.*, 2:145–156, 1993.

[11] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics for free boundaries. *J. Comput. Phys.*, 39(201), 1981.

[12] W.F. Noh and P. Woodward. The slic (simple line interface calculation) method. Technical Report UCRL-52111, Lawrence Livermore National Laboratory, 1976.

[13] R. Pember. Numerical methods for hyperbolic conservation laws with stiff relaxation. II. Higher order Godunov methods. *SIAM J. Sci. Stat. Comput.*, to appear July, 1993. UCRL JC-109097, Pt. II, Lawrence Livermore National Laboratory, January, 1992.

[14] R. Pember, J. Bell, P. Colella, W. Crutchfield, and M. Welcome. An adaptive Cartesian grid method for unsteady compressible flow in complex geometries. *J. Comput. Phys.*, accepted, 1995. UCRL JC-115650, Lawrence Livermore National Laboratory, November, 1993.

[15] J.E. Pilliod and E.G. Puckett. Second-order volume-of-fluid algorithms for tracking material interfaces. (in preparation).

[16] E.G. Puckett and J.S. Saltzman. A 3-d adaptive mesh refinement algorithm for multimaterial gas dynamics. Physica D, (to appear).

[17] D.L. Youngs. Time-dependent multimaterial flow with large fluid distortion. In K.W. Morton and M.J. Baines, editors, *Numerical Methods for Fluid Dynamics*. Academic Press, New York, 1982.

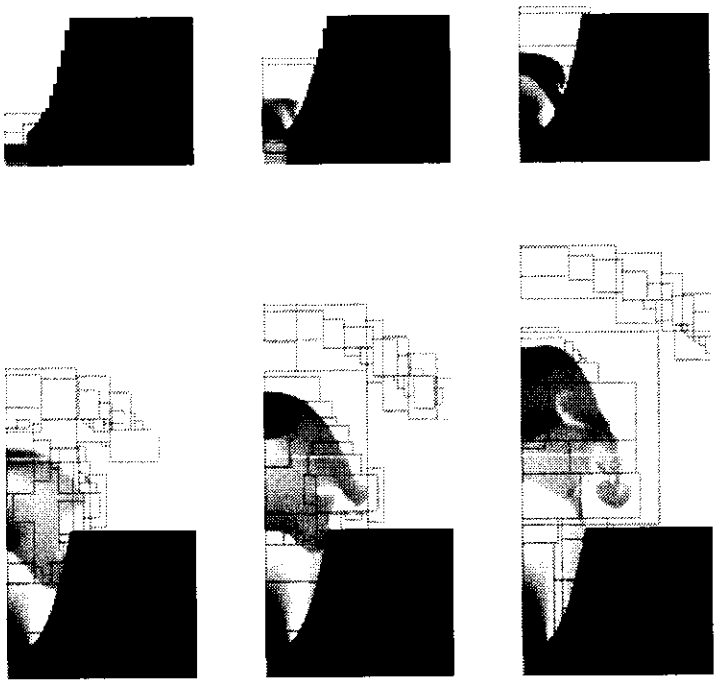

Figure 1: The evolution of $\rho^{\alpha} f^{\alpha}$ in a model nozzle problem is shown at six times. The Cartesian grid body is shown in black. The overlayed grids represent the AMR refined patches with two levels of refinement over the base grid. Each level is a factor of two finer than the previous level The inlet flow is supersonic with Mach number 2.04. The flow expands to accomodate the increased cross-sectional area of the nozzle. Shocks form in the jet core (Mach discs) as the fluid exits the nozzle.