Preprint
UCRL-JC-151904

A Dynamically Adaptive Arbitrary Lagrangian-Eulerian Method for Solution of the Euler Equations

R. W. Anderson, N. S. Elliott, and R. B. Pember

This article was submitted to
Nuclear Explosive Code Developers Conference, Monterey, CA,
October 21-24, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

February 14, 2003

Proceedings of the NECDC 2002

A Dynamically Adaptive Arbitrary Lagrangian-Eulerian Method for Solution of the Euler Equations (U)

R.W. Anderson*, N.S. Elliott*, R.B. Pember*

*Lawrence Livermore National Laboratory

A new method that combines staggered grid arbitrary Lagrangian-Eulerian (ALE) techniques with structured local adaptive mesh refinement (AMR) has been developed for solution of the Euler equations. The novel components of the methods are driven by the need to reconcile traditional AMR techniques with the staggered variables and moving, deforming meshes associated with Lagrange based ALE schemes. We develop interlevel solution transfer operators and interlevel boundary conditions first in the case of purely Lagrangian hydrodynamics, and then extend these ideas into an ALE method by developing adaptive extensions of elliptic mesh relaxation techniques. Conservation properties of the method are analyzed, and a series of test problem calculations are presented which demonstrate the utility and efficiency of the method. (U)

1 Introduction

The numerical simulation of compressible flows with time-dependent boundaries as well as broad ranges of spatial and temporal scales that must be resolved is a computational challenge in many important application areas including astrophysics, inertial confinement fusion (ICF), plasma physics, and aeroelasticity. Resolution of localized small scale features such as shocks, mixing layers, and reaction zones requires high resolution which would be prohibitive if used globally throughout the calculation. When solving problems including time-dependent boundaries, Lagrangian and ALE techniques have often been favored in the above application areas [1]. In addition to providing a natural mechanism for handling moving boundaries, an attractive feature of Lagrange based ALE methods is that they are to some extent self-adapting, i.e., cells cluster into high density regions behind shocks, and material discontinuities can be tracked intrinsically.

However, this inherent form of adaptivity present in ALE methods, while an advantage over pure Eulerian codes in some applications, is less general and robust than a dynamically adaptive method in which the number of cells may change with time. It seems quite common in a wide variety of applications to be interested in a relatively simple initial condition which can be sufficiently described by relatively few degrees of freedom, which is then followed in time as it grows in complexity, requiring increasing degrees of freedom in the discrete solution vector to maintain a desired level of accuracy.

This is the fundamental motivation for the introduction of a dynamic spatio-temporal description of the solution into the class of ALE methods described herein. An approach which has proven effective in providing the kind of dynamic solution description that we desire is structured grid local adaptive mesh refinement (AMR) [2, 3, 4, 5].

Extension of the AMR idea to Lagrangian and ALE solution techniques is nontrivial for several reasons. The first is that typical AMR solution techniques are based on cell-centered discretizations, in contrast with the class of staggered mesh discretizations considered here that use a nodal description of mass element position and velocity, but cell-centered descriptions of thermodynamic quantities. The second source of additional complexity is that AMR methods typically utilize (nested) Cartesian meshes, while our Lagrange based method must consider moving, deforming meshes. Finally, our approach to ALE mesh motion will be of the "grid relaxation" variety, in which some smoothing operator will be applied to prevent mesh tangling which would otherwise occur as the mesh undergoes large deformation. Typical smoothing operators are based on elliptic equations, and the application of these smoothing operators must be extended to the case of a multi-level adaptive mesh hierarchy.

2 Single-Level Lagrangian and ALE Methods

The governing equations of inviscid gasdynamics employed in this work are, in conservative form,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1a}$$

$$\nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p = 0 \tag{1b}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \mathbf{u} E + p\mathbf{u}) = 0 \tag{1c}$$

where ρ, E, p, and $\mathbf{u}$ are the fluid density, total energy, pressure, and velocity respectively, and t is time. The internal energy is $e = E - \frac{1}{2}\mathbf{u} \cdot \mathbf{u}$, and an equation of state $p = p(\rho, e)$ closes the equation set. In this work, we use $p = (\gamma - 1)\rho e$.

The ALE method employed for integration of the system (1) is of the explicit, time-marching, Lagrange plus remap type. Schemes of this type involve two distinct phases. In the first phase, a Lagrange step advances the flowfield through a physical timestep. The optional second phase involves a modification of the grid and a remapping of the solution to the new grid. The new grid can be the original grid, which results in a fully Eulerian method, or it may be a "relaxed" grid that has been smoothed in some manner. The grid modification procedure is designed to alleviate the mesh tangling problem inherent in the Lagrangian methods for flows with regions of shear. The mesh relaxation algorithm employed here is essentially a Laplace iteration, and has its origins in the work of Winslow [6]. The subsequent solution interpolation procedure is formulated as an apparent advection problem, and is discussed in Section 2.3.

2.1 Notation

We present simultaneously, where convenient, algorithms for the general case of d dimensions where $d = 1, 2,$ or 3 through the use of the following notations. The physical space coordinates x, y, z shall be referred to as x_1, x_2, x_3, as in the indicial notation, although there is no implied summation. The real-valued logical space coordinates ξ, η, ζ shall be referred to as ξ_1, ξ_2, ξ_3, and will be called i, j, k at integer values to refer to locations on the discrete computational grid.

To exploit periodicities in the indices i, j, k we employ indexing functions mapping integers to index place holders [7]. In 3D, the index functions are:

$$\alpha = \alpha(l) : \alpha(1) = i, \alpha(2) = j, \alpha(3) = k \tag{2a}$$

$$\beta = \beta(l) : \beta(1) = j, \beta(2) = k, \beta(3) = i \tag{2b}$$

$$\gamma = \gamma(l) : \gamma(1) = k, \gamma(2) = i, \gamma(3) = j. \tag{2c}$$

In 2D, the index functions are:

$$\alpha = \alpha(l) : \alpha(1) = i, \alpha(2) = j \tag{3a}$$

$$\beta = \beta(l) : \beta(1) = j, \beta(2) = i. \tag{3b}$$

All equations involving the indexing functions implicitly assume l to range from 1 to d, unless otherwise noted.

Further economy of notation is obtained with implied indexing. In this notation, unspecified indices take on default values. For example, the notation q implies q_i in 1D, $q_{i,j}$ in 2D, and $q_{i,j,k}$ in 3D. The notation q_{j+1} implies $q_{i,j+1}$ in 2D, $q_{i,j+1,k}$ in 3D, and is not meaningful in 1D.

We distinguish cell-, node-centered indices by their positioning. Subscripts from the set i, j, k imply nodal indices and superscripts imply cell indices. Where there may be ambiguity due to implied indexing, on over-tilde will be used for node-centered quantities. Time levels, when required, are notated with superscript n.

Finally, we define two families of shift operators, S_λ and S^λ, which operate on subscript and superscript indices respectively, for $\lambda = 1, 2, .., 2^d$ as

$$\begin{aligned}
S_1\phi &= \phi & S^1\phi &= \phi \\
S_2\phi &= \phi_{i+1} & S^2\phi &= \phi^{i-1} \\
S_3\phi &= \phi_{j+1} & S^3\phi &= \phi^{j-1} \\
S_4\phi &= \phi_{i+1,j+1} & S^4\phi &= \phi^{i-1,j-1} \\
S_5\phi &= \phi_{k+1} & S^5\phi &= \phi^{k-1} \\
S_6\phi &= \phi_{i+1,k+1} & S^6\phi &= \phi^{i-1,k-1} \\
S_7\phi &= \phi_{j+1,k+1} & S^7\phi &= \phi^{j-1,k-1} \\
S_8\phi &= \phi_{i+1,j+1,k+1} & S^8\phi &= \phi^{i-1,j-1,k-1}.
\end{aligned}$$

λ always ranges from 1 to 2^d, either over a summation or to define a family of equations. The connection between the cell- and node-centered indexing systems is then that cell w is surrounded by nodes $S_\lambda x$, and correspondingly node x is surrounded by cells $S^\lambda w$.

2.2 Single Level ALE Method

The Lagrange step discretization follows more directly from integral expressions for the governing equations over a Lagrangian volume V bounded by a moving surface S:

$$\frac{d}{dt}\int_V \rho\, dV = 0 \tag{4a}$$

$$\frac{d}{dt}\int_V \rho\mathbf{u}\, dV = -\int_S p\mathbf{n}\, dS \tag{4b}$$

$$\int_V \frac{de}{dt}\, dV = -\int_V p\frac{dv}{dt}\, dV, \tag{4c}$$

where $\mathbf{n}$ are outward normal vectors on the surface elements dS, and $v = 1/\rho$ is the specific volume.

A structured mesh composed of quadrilaterals in two dimensions and hexahedra in three dimensions is used to discretize (4). As an initial basis for constructing an ALE-AMR scheme, we will employ for the Lagrange step the general approach taken by Tipton [8], et al. We use a predictor-corrector discretization in time, and a staggered spatial discretization [9, 10, 11]. The scheme employs a monotonic artificial viscosity, q, due to Christensen [12], and a kinematic hourglass filter [13]. The two-dimensional scheme has been described extensively previously; algorithmic details as well as comparisons with more widely known Eulerian methods can be found in a recent work by Pember, et al.[14]. Instead, we present the algorithm in a form which extends to three dimensions, and point out the aspects of the method which have particular relevance for the adaptive formulation.

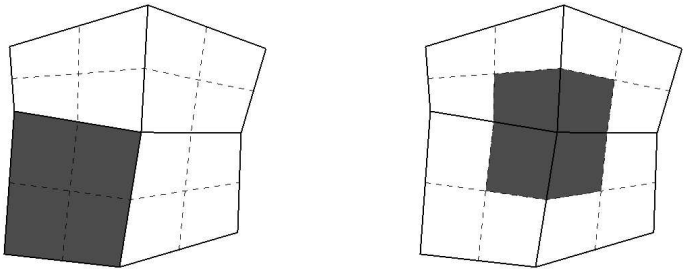
The Lagrange step consists in broad outline of an integration of the momentum equation for an acceleration at each node, a kinematic advance of nodal positions, and an energy update for the resulting $p\, dv$ work done on each cell, in both predictor and corrector steps. The acceleration is computed from a discretized (4b) by constructing a control volume around each node, which we will refer to as a "dual cell." These dual cells can be constructed from a "median mesh," as shown in Fig. 1.

The median mesh is generated by a d-linear interpolant $\mathbf{x}^{(d)}(\xi_l)$, which can be expressed using a shift operator

$$E_l x = x_{\alpha+1}, \tag{5}$$

as the composition of d semi-interpolants

$$\mathbf{x}^{(d)} = u_d\mathbf{x}^{(d-1)} + (1-u_d)E_d\mathbf{x}^{(d-1)}. \tag{6}$$



(a) Primary Cell (b) Dual Cell

Fig. 1. Primary and dual cells are composed of 2^d corners as shown for $d = 2$.

with the recursion truncating with the definition $\mathbf{x}^{(0)} = \mathbf{x}$. The u_l are local parameters

$$u_l = \xi_l - \alpha, \qquad \alpha \leq \xi_l \leq \alpha + 1. \tag{7}$$

The nodes of the median mesh are then located at all integer and half-integer values of ξ_l. While (6) is a convenient notation, note that the interpolant is invariant with respect to the ordering of the calculation of semi-interpolants for $d > 1$.

The extension to 3D of the algorithm described in [14] is then fairly straightforward. It requires a 3D formulation for the artificial viscosity q, a 3D formulation for hourglass filtering, and the ability to compute 3D geometric quantities such as the volume of a hexahedron. The hourglass filter we use is an extension of the method due to Margolin and Pyun [13], and the viscosity is an extension of the method described by Christensen [12].

It turns out that the definition of nodal mass as an average of the neighboring cell masses

$$\tilde{m} = \frac{1}{2^d} \sum_{\lambda} S^{\lambda} m, \tag{8}$$

has important implications for the properties of the adaptive method. In Section 3.2, we discuss this issue in detail. Also, as shown in 3.2, the definition of the hexahedral volume in 3D is central to the mass conservation properties of the adaptive scheme. The best choice for consistency in the adaptive method is an exact integration of the volume of the d-linear solid defined by (6). There are a number of equivalent formulas for this computation; the most efficient appears to be that given by Dukowicz [15]. Additional details regarding the properties of the tri-linear mapping and the computation of its volume may be found in Ushakova [16].

At the end of a Lagrange step, it is often desirable to smooth the grid to prevent excessive mesh distortion which can lead to inaccuracy or even failure of the Lagrangian algorithm. An effective smoothing algorithm can be based upon a Laplace iteration for the

transformed coordinates with respect to the Cartesian coordinates of each node. This type of equipotential method [17, 18] extends straightforwardly to three dimensions, but requires careful treatment in the context of an adaptive mesh hierarchy, to be discussed in Section 3.3.1.

2.3 Solution Remap

Once the relaxed mesh has been defined, it remains to interpolate the solution from the old Lagrange grid to the relaxed grid. A convenient way to formulate this problem is in terms of an apparent advection equation, which puts at our disposal a wide variety of well-proven solution methods, from which we choose to implement a variant of the Corner Transport Upwind (CTU) method. As in the Lagrange step, we generalize the 2D methods presented in [14] such that they extend to 3D.

The governing equation is simply a statement of stationary solution for all scalars ϕ in the physical space

$$\frac{\partial \phi}{\partial \tau} = 0, \tag{9}$$

through a pseudo-time τ. Transforming into time-dependent logical coordinates gives

$$\frac{\partial J\phi}{\partial \tau} = \sum_{l=1}^{d} \frac{\partial}{\partial \xi_l} \left(\mathbf{n}^{\xi_l} \cdot \mathbf{s}\phi \right), \tag{10}$$

where

$$J = \det \frac{\partial x_l}{\partial \xi_m} \tag{11}$$

is the Jacobian, $\mathbf{s}$ is a grid velocity vector

$$s_l = \frac{\partial x_l}{\partial t}, \tag{12}$$

and the $\mathbf{n}^{\xi_l}$ are normal vectors, the components of which are the cofactors of the Jacobian matrix

$$J_{lm} = \frac{\partial x_l}{\partial \xi_m}, \tag{13}$$

i.e.,

$$n_k^{\xi_i} = \frac{1}{2} \sum_{p,q,l,m} \epsilon_{klm} \epsilon_{ipq} J_{pl} J_{qm}. \tag{14}$$

where ϵ_{ijk} is the usual permutation symbol.

We discretize the grid velocity as

$$\mathbf{s} = \left(\mathbf{x}^{new} - \mathbf{x}^{old} \right) / \Delta\tau, \tag{15}$$

and make the arbitrary choice $\Delta\tau = 1$. The Jacobian matrix terms are discretized as

$$\frac{\partial \mathbf{x}}{\partial \xi_l} = \left(E_l - 1 \right) \mathbf{x}. \tag{16}$$

Since we will be applying (10) to both node- and cell-centered quantities, we require approximations for J, $\mathbf{s}$ and $\mathbf{n}^{\xi_i}$ on the dual mesh. These are obtained by averaging

$$\tilde{\phi} = \frac{1}{2^d} \sum_{\lambda=1}^{2^d} S_\lambda \phi, \tag{17}$$

where ϕ is J, $\mathbf{s}$, or $\mathbf{n}^{\xi_i}$.

We wish to construct conservative updates of the form

$$\phi^{n+1} x^{n+1} = \phi^n x^n + \sum_m F_m \tag{18}$$

where ϕ is a solution scalar, e.g., (ρ, u_k, E), x is some basis appropriate for the relevant conservation law, and F is a flux function for each of the m "faces" of a control volume. In particular, we require a volume basis for density, and a mass basis for components of velocity and energy. If we choose the flux function to be products of face states times a transport quantity, $F = \phi\, \delta x$, then a constant field preserving scheme requires that

C1) face states reproduce a constant value

C2) $x^{n+1} = x^n + \sum_m \delta x_m$

The sequence of updates for each solution variable for the average mass discretization, given a general prescription for computing face states, has been described previously [14]. One subtlety of particular note is that conservation of total energy is achieved through the independent remapping of kinetic energy and internal energies. This choice is essentially empirical in that it seems to produce better results than a direct remap of total energy in a number of test problems [14]. We describe here the extensions to 3D, following the general approach of Saltzmann [7].

The 3D method of obtaining face states consists of three phases in which successive 1D, 2D, and 3D approximations are computed. The relations are

$$\phi^{3D}_{\alpha+\frac{1}{2}} = \phi + \frac{1}{2}\left(1 - \Delta\tau\kappa_l\right)\overline{\Delta}_\alpha\phi - \frac{1}{2}\Delta\tau\kappa_{l+1}\Delta^-_\beta\left({}_\alpha\phi^{2D}_{\beta+\frac{1}{2}}\right) - \frac{1}{2}\Delta\tau\kappa_{l+2}\Delta^-_\gamma\left({}_\alpha\phi^{2D}_{\gamma+\frac{1}{2}}\right) \tag{19}$$

$${}_\alpha\phi^{2D}_{\beta+\frac{1}{2}} = \phi + \frac{1}{2}\left(1 - \Delta\tau\kappa_l\right)\overline{\Delta}_\alpha\phi - \frac{1}{d}\Delta\tau\kappa_{l+2}\Delta^-_\gamma\phi^{1D}_{\gamma+\frac{1}{2}} \tag{20}$$

$${}_\alpha\phi^{2D}_{\gamma+\frac{1}{2}} = \phi + \frac{1}{2}\left(1 - \Delta\tau\kappa_l\right)\overline{\Delta}_\alpha\phi - \frac{1}{d}\Delta\tau\kappa_{l+1}\Delta^-_\beta\phi^{1D}_{\beta+\frac{1}{2}} \tag{21}$$

$$\phi^{1D}_{\alpha+\frac{1}{2}} = \phi + \frac{1}{2}\left(1 - \Delta\tau\kappa_l\right)\overline{\Delta}_\alpha\phi \tag{22}$$

where

$$\kappa_l = \frac{\mathbf{n}^{\xi_l} \cdot s}{J}, \tag{23}$$

Δ^- is a backwards difference operator

$$\Delta^-_\alpha \phi_{\alpha+\frac{1}{2}} = \phi_{\alpha+\frac{1}{2}} - \phi_{\alpha-\frac{1}{2}}, \tag{24}$$

and $\overline{\Delta}_\alpha$ is a Van Leer limited central difference operator. The 2D approximations are double-valued corresponding to the two possible plane orientations for 2D expansions to each face. As in the 2D algorithm, the face state is upwinded on the sign of the transport mass or volume at each stage of approximation.

3 ALE-AMR Algorithm

3.1 AMR Overview

The conceptual starting point for the AMR methodology development is the pioneering work of Berger, Oliger, and Colella [3, 19]. In this approach, a hierarchical grid structure is employed which changes dynamically in time, and is composed of logically rectangular, uniform grid "patches" of varying resolution. In the original work, the grid hierarchy is constructed so that a coarse grid cell is covered precisely by r^d fine grid cells, where r is called the refinement ratio. The solution is defined on all cells, including coarse cells which underlay cells of finer resolution. The collection of grid patches at a given resolution is referred to as a level.

An explicit time-marching method of a general hierarchy of l_{max} levels of refinement can be expressed as a recursive procedure beginning with the coarsest level $l = 0$:

repeat
  construct interpolated $l-1$ and/or domain boundary conditions
  advance level l to t = min($t + \Delta t$, t_{l-1})
  if $l < l_{max}$ **then**
    recurse with $l = l + 1$
    synchronize levels $l + 1$ and l
    optionally regrid levels $l + 1$ to l_{max}
  end if
until $t = t_{l-1}$

where Δt is a stable time step for the level, and t_{-1} is the desired end simulation time. Typically $\Delta t_l \approx r\Delta t_{l+1}$, resulting in a nesting or "subcycling" of time steps for the levels in the hierarchy.

We will construct the necessary interpolation, coarsening, and synchronization operators first for a purely Lagrangian method for which the above procedure can remain essentially unchanged, and then extend the ideas into the ALE-AMR method, which requires the introduction of some fundamental modifications to the above algorithm to accomodate mesh relaxation.

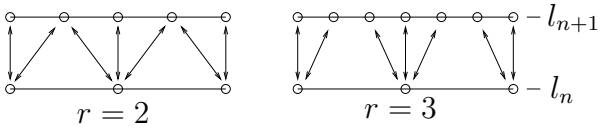


Fig. 2. Odd refinement ratios are required to maintain an r:1 correspondence between fine nodes and their corresponding coarse nodes in any given nodal stencil.

3.2 Lagrangian AMR

Interlevel solution transfer operators are required when new grids are created, for the generation of boundary conditions on finer levels in the hierarchy, for synchronizing coarse and fine data in the hierarchy, and upon the removal of refined grids.

The refinement and coarsening operators presented here are designed with the following properties in mind:

P1) Constant field preservation

P2) 2nd order accuracy (in smooth regions)

P3) Monotonicity

P4) Local conservation

P5) Exact inversion of refinement by coarsening

A simple way to ensure that P5 is easily achieved is to maintain an exclusive r:1 correspondence between fine and coarse node data locations. In this case, any conservative distribution of a quantity from the coarse mesh to its corresponding fine mesh stencil may be inverted exactly with a simple summation. This is only achieved for *both* cell- and node-centered quantities by choosing r odd, as shown in Figure 2.

3.2.1 Refinement

Consider a one-dimensional linear interpolation for N values ϕ_k of a scalar density function ϕ with slope ϕ_0' and average value ϕ_0 defined over an interval Δx_0 on some coordinate basis x:

$$\phi_k = \phi_0 + \phi_0' \left(x_k - \frac{1}{2}\Delta x_0 \right).$$

With the connection between x_k and Δx_k being

$$x_k = \sum_{i=1}^{k-1} \Delta x_i + \frac{1}{2}\Delta x_k, \tag{25}$$

the construction is conservative of $\phi\Delta x$ in the sense that

$$\sum_{k=1}^{N} \phi_k \Delta x_k = \phi_0 \Delta x_0,$$

provided that

$$\sum_{k=1}^{N} \Delta x_k = \Delta x_0, \tag{26}$$

since

$$\sum_{k=1}^{N} \phi_k \Delta x_k = \phi_0 \Delta x_0 + \phi_0' \left(\sum_{k=1}^{N} x_k \Delta x_k - \frac{1}{2} \Delta x_0^2 \right). \tag{27}$$

We will refer to (26) as the refinement consistency condition.

In a constant field, all slopes ϕ_0' are zero (by definition), and constant fields are preserved as long as the consistency condition holds. We now have a general one-dimensional expression for interpolation that satisfies P1, P2, and P5.

In order to satisfy P3, we employ the well-known van Leer limiter for slope determination. If we desire to prevent oscillations in the primitive variables $\phi = (\rho, u, v, E)$, where E is the total energy, the required interpolation basis to obtain property P4, local conservation, is $x = (V, \tilde{m}, \tilde{m}, m)$, where V is cell volume, $\tilde{m}$ is nodal mass, and m is cell mass.

The multi-dimensional case can be handled with d one-dimensional sweeps, or, if a strictly symmetric operator is desired, the average of $d!$ permutations of d sweeps. However, we use a more efficient formula based on a d-linear (in volume or mass coordinates) interpolant can be used. For example, a formula for a (continuous) density interpolant is

$$\rho = \rho_0 + \sum_{l=1}^{d} \delta\rho_l \left(u_l - \frac{1}{2} \right) \tag{28}$$

where the $\delta\rho_l$ are limited differences, and the u_l are normalized volume coordinates, where the discrete values are again taken at average volume coordinates $\overline{u}$ over an interval, as in (25).

We now face a significant problem in reconciling the design goal P4 in the case of momentum, with the relationship between cell and nodal mass as defined in (8). Nodal mass, defined as an average of neighboring cells, does not meet the consistency condition (26) on the fine mesh after a cell-centered interpolation of density.

In 1D, intuitively it can be seen that the nodal mass definition (8) is equivalent to a definition which prescribes a constant density profile in each cell, and integrates those profiles over the control volume bounded by the median mesh surrounding each node. However, to satisfy P2, we interpolate cell densities using non-zero slopes, which is a fundamentally incompatible formulation. One can readily see that non-zero slopes in a cell serve to distribute mass unequally to the fine nodal stencils corresponding to the

neighboring nodes of the coarse cell; whereas on the coarse mesh, that distribution is assumed to be equal by definition.

There are a number of alternative formulations which can provide a resolution to the momentum conservation problem. One such alternative is adopt the concept of "corner mass" as employed by Caramana et. al. [20, 21, 22]. One can then apply the density interpolation on the corner mesh directly, which eliminates the apparent nodal mass flux which leads to non-conservation of the averaged mass formulation. There are two drawbacks to this approach, however. The first is the computational expense associated with storing and computation of sub-grid densities. The second is that there are at present no known advection schemes for this discretization that meet all the criteria set forth in Section 2.3, although there is ongoing work in this area [22]. A second alternative is motivated by the idea that one can define a nodal mass for use in the Lagrange step which is consistent with the interpolation procedure by construction, i.e., to use the density interpolant as employed in the refinement operation to explicitly integrate the mass associated with any node. This requires a new approach to the Lagrange step, however, since the nodal mass will not be constant from one time step to the next, which is a central assumption to the Lagrange formulation. However, the Lagrange step can be generalized to include a nodal momentum flux term which accounts for the implied mass transport between the corners of a cell through a Lagrange step. Such a method is Lagrangian with respect to cell mass, but is not Lagrangian with respect to nodal mass. While the method does result in the desired conservation properties for both the Lagrange step and the interpolation, the expense associated with exactly integrating the nodal mass from a d-linear interpolant of density at each step is not insignificant.

While several such alternatives have been implemented to reconcile this difficulty, we consider none so compelling as to be clearly superior to the relatively straightforward and computationally efficient approach described above. Rather it has been the case in a wide variety of numerical tests that we see no evidence of shock speed accuracy degradation, which is the prime pragmatic concern regarding conservation errors.

3.2.2 Coarsening

By choosing r odd, we have enabled particularly simple coarsening operators which are local weighted sums of conserved quantities, to satisfy both P4 and P5. The consistency condition is a central issue. Unless the condition is satisfied, there are two choices for a coarsening operator

$$\phi_0 = \sum_{k=1}^{N} \phi_k \Delta x_k / \sum_{k=1}^{N} \Delta x_k, \tag{29}$$

or

$$\phi_0 = \sum_{k=1}^{N} \phi_k \Delta x_k / \Delta x_0, \tag{30}$$

One can readily see that (29) is the constant field preserving choice, and (30) is the conservative choice, but not vice-versa.

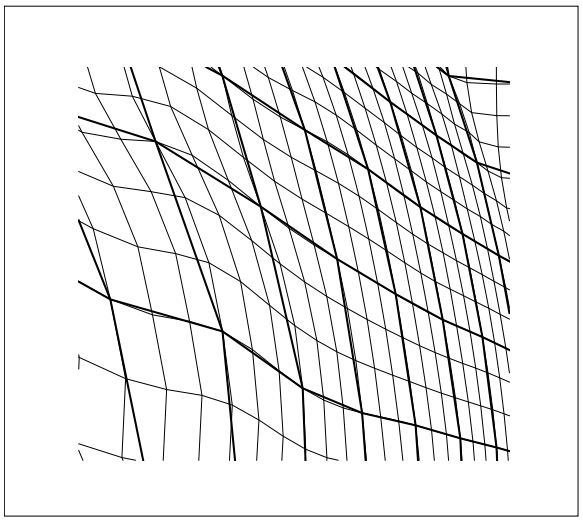


Fig. 3. Lagrange grid fine-coarse misalignment under deformation. Heavy lines are coarse mesh, thin lines are fine mesh.

After one or more Lagrangian advances, the fine and coarse meshes will no longer be spatially aligned at all nodes and/or edges. Even after the initial injection of nodal positions, there is still misalignment between the collections of fine cells correspondingly logically to an underlying coarse cell, as shown in Figure 3. Indeed, if by some methodological device this were not the case, the AMR method would not be providing the additional degrees of freedom in the solution we desire on finer levels. One can easily see that the consistency condition in the case of volume, for example, will not in general be satisfied after one or more Lagrange steps. One way to reconcile (29) and (30) is to remap the fine data as a preprocessing step to satisfy the consistency condition, i.e., from the original grid to a fine grid which is aligned with the underlying coarse mesh. Whether the additional expense is justified is a matter of problem dependent empiricism. Again we find that when the mesh is not grossly distorted, the conservation errors do not lead to significant shock speed accuracy degradation.

3.2.3 Interlevel Boundary Conditions

Boundary conditions on finer grids in the hierarchy require careful treatment. The principle consideration is the spatial synchronization of the coarse and fine boundary nodes which will in general not stay aligned without special treatment. We have chosen to linearly interpolate, first in time, if necessary, and then in space, the positions of boundary nodes from the next coarser level, and employ the refinement operators developed in the previous section for all other quantities in ghost regions. This is motivated by a desire to always have quad or hex elements on a composite mesh. If boundary node positions were integrated according to the numerical scheme rather than imposed, then in general there would form non-quad or non-hex elements, as shown in 2D in Fig. 4.

This procedure is no compromise as long as the adaption criteria are sufficiently well

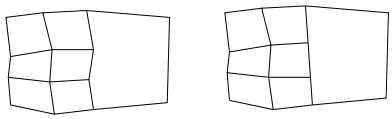


Fig. 4. On the left, "free" fine boundary nodes introduce non-quad elements. On the right, interlevel boundary conditions interpolate positions to preserve quad elements.

designed that the coarse mesh solution alone is of acceptable accuracy in the vicinity of coarse-fine mesh boundaries.

3.3 ALE-AMR

The introduction of a mesh relaxation operator of the elliptic relaxation type necessitates a fundamental change to the AMR hierarchy integration procedure as outlined in Section 3.2 to accomodate the intrinsic global coupling of the solution for the relaxed mesh. The basic requirement is that coarse meshes cannot be relaxed independently of finer meshes in the hierarchy. The key idea is that *we must defer the relaxation of a level's grids until all finer levels have been advanced to the same simulation time.* While it may seem natural to simply define a new operator that combines the Lagrange and remap steps into a composite step, such an algorithm fails, as it decouples the coarse mesh relaxation from finer meshes, leading to ill-behaved mesh motion on the composite grid.

We define a "level grid" g_l as the set of all nodes on level l, and a "composite grid" g_l^c as

$$g_l^c = g_l \setminus I(g_{l+1}) \cup g_{l+1}^c. \tag{31}$$

where $\setminus$ is the set difference operator, I is a restriction operator of simple injection, and the definition $g_{max}^c = g_{max}$ closes the recursion.

An ALE-AMR algorithm is then

repeat
  construct interpolated $l-1$ and/or domain boundary conditions
  advance level l Lagrangian to t = min($t + \Delta t_l$, t_{l-1})
  if $l < l_{max}$ **then**
    recurse with $l = l + 1$
    repeat
      relax g_l^c
    until g_l^c is sufficiently smooth
    remap levels l to l_{max}
    synchronize levels l through l_{max}
    optionally regrid levels $l + 1$ to l_{max}
  end if
until $t = t_{l-1}$

The logical diagram in Fig. 5 visualizes the process for a 3-level hierarchy.

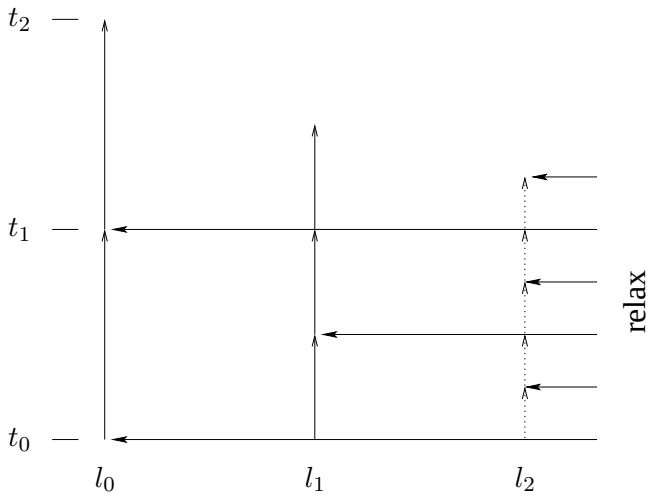


Fig. 5. ALE-AMR hierarchy integration logical diagram, shown for a 3-level hierarchy. Vertical lines are Lagrange steps, horizontal lines are relaxation operations.

3.3.1 Multilevel Mesh Relaxation

The remaining key issue is to construct a relaxation operator for a composite grid g_l^c, using an extension of the elliptic relaxation approach. We define a "finest available" grid

$$g_l^* = g_l \setminus I(g_{l+1}), \tag{32}$$

the union of which for all l is the composite grid.

A design principle of the composite mesh operator R is

$$\text{if } R(g_l^*) = g_l \forall l, \text{ then } R(g_0^c) = g_0^c. \tag{33}$$

This principle requires that a uniform mesh at each level should remain uniform, as opposed to letting the finer meshes expand in response to coarse-fine boundary regions, which would occur if we applied some type of unstructured or semi-structured discretization of the Winslow method to the composite mesh directly. In order to get the desired behavior (33), the discretization must respect the per-level index space of each node in the composite mesh, with the interlevel coupling occuring through injection and interpolation of boundary conditions between the g_l^*. Each iteration of $R(g_l^c)$ consists of:

for $m = l_{max}$ to l **do**
  constructed interpolated boundary conditions around g_m from g_{m-1}
  relax g_m^*
  inject g_m to g_{m-1}
end for

3.3.2 Implementation

The implementation utilizes SAMRAI [23], an object-oriented framework for the development of structured grid adaptive mesh refinement applications. The framework has been extended to accommodate many of the novel or unusual AMR features developed in the current work. The SAMRAI framework is a C++ library, and the application code was developed using both C++ and FORTRAN 90, with FORTRAN 90 being reserved for performance of critical inner loop constructs. We have found this dual language choice to be an effective paradigm for scientific calculation when the algorithms and data structures are of sufficient complexity to warrant the abstraction mechanisms provided by the C++ language.

4 Numerical Results

An ideal AMR extension of a given method is one which provides solutions identical to the single-grid formulation at a reduced computational cost. With this in mind we seek to characterize the performance of the ALE-AMR method with respect to this ideal, as well as demonstrate the potential for improving the state of the art in high resolution 3D ICF calculations.

The criteria used for the selection of refined regions is constructed from normalized second differences:

$$C_\alpha = \left\| \frac{\phi_{\alpha+1} - 2\phi + \phi_{\alpha-1}}{\max(\phi_0, \epsilon)} \right\|_{L2} \tag{34}$$

where ϵ is a small constant to prevent blowup, and ϕ_0 is a global normalization constant. In practice, we define two criteria separated by a tolerance of a few percent. The smaller of the two values is required to untag a refined cell, and the larger to tag for refinement an untagged cell. This helps prevent “chatter” of refining and de-refining when the criteria is near the threshold. The scalar ϕ is usually evaluated both from the pressure and density with an “or” operation on the result of the evaluation. Pressure criteria alone can miss contact surfaces, and density differences alone often miss critical regions of large acceleration. While additional work into robust error indicators is warranted, we find that this simple heuristic indicator seems to work quite well in practice. For all of the following calculations, the refinement ratio is chosen to be 3.

4.1 Relative Accuracy and Efficiency Characterization

We begin with a one-dimensional interacting blast wave problem due to Woodward and Colella [24]. The initial conditions are:

$$p = \begin{cases} 1000 & 0 < x < 1/3, \\ .01 & 1/3 < x < 2/3, \\ 100 & 2/3 < x < 1, \end{cases} \tag{35}$$

with everywhere unity density and zero velocity, and $\gamma = 1.4$.

In Figure 6, the results at $t = 0.38$ are compared between a relatively highly resolved single-level solution and an adaptive solution using an equivalent maximum resolution of 2700 cells and two levels of refinement, using a purely Lagrangian calculation. The solutions are essentially indistinguishable. Figure 7 demonstrates that these essentially equivalent solutions can be obtained with a significant increase in computational efficiency, with the maximal efficiency being reached with three solution levels, i.e., two levels of refinement. One should note that the computational savings associated with advancing coarsened cells in the adaptive algorithm is proportional to r^{d+1}; the efficiency benefits tend to increase with increasing spatial dimension.

To quantify the trade-off between relative accuracy and efficiency further, we can compute L1 norms of difference vectors between the adaptive and fully resolved calculations. By relative accuracy it is meant the accuracy with which the adaptive solution replicates a solution using fine grid everywhere. Table 8 lists norms which represent relative mass defects for a matrix of calculations. By effective resolution we mean the resolution at the finest level of the calculation. Consider the 2-level solution at an effective resolution of 8100 cells. The difference vector norm shown is always relative to the single-level calculation at 8100 cells, and is of order 10^{-3}. To provide a scale for the significance of that "error," we can compare to a calculation at the next coarser resolution on a single level, which produces an error of the order 10^{-2}, an order of magnitude greater. This result should be weighed against the wall clock cost of the solutions, listed in Figure 9.

A similar exercise, this time utilizing relaxation and remap, for the 2D Taylor-Sedov blast wave produces the results shown in Figures 11 and 12. The AMR speedup increases to about 16 for this problem, even though we are covering more of the domain with fine grid, due to the improved efficiency associated with the work scaling factor r^{d+1}. An example 3-level solution is shown in Figure 10.

4.2 3D Implosion Richtmeyer Meshkov Instability

To demonstrate the potential of the ALE-AMR algorithm in more demanding applications, we have run a proof-of-principle calculation of a Richtmeyer-Meshkov instability in an idealized ICF implosion. In this model problem, we follow the unstable

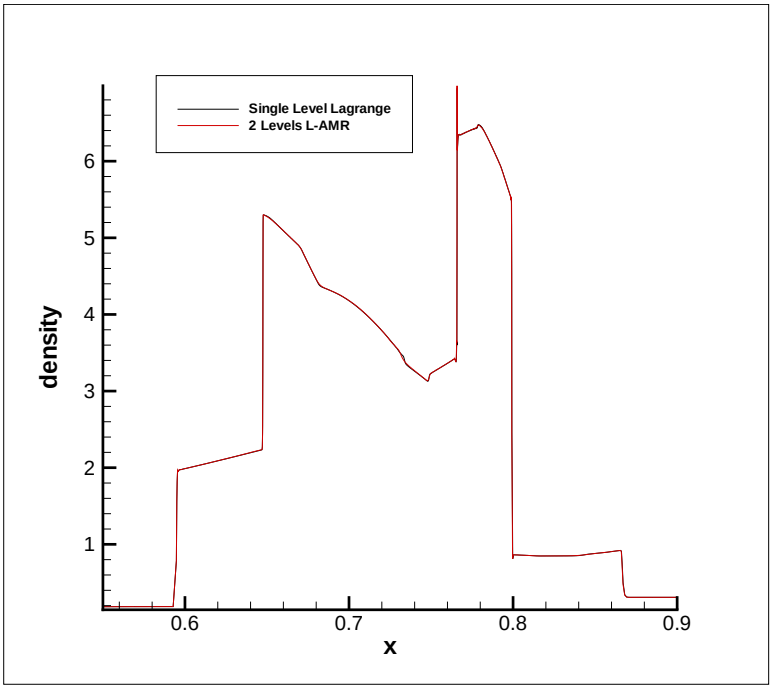


Fig. 6. 1D Colella-Woodward blast wave solutions at t=0.38 s. The adaptive solution has a coarsest resolution of 300 cells and 2 refinement levels, and is essentially indistinguishable from the fully resolved solution with 2700 cells.

evolution of a single mode perturbation on the outer surface of the inner shell of a double-shell ICF target. The initial conditions contain a high pressure region in the outer surface of the outer shell to model the effects of ablation by rapid laser energy deposition. Previous calculations using other models have shown that approximately .67 MJ of energy is transferred to the hohlraum from 2.5 MJ of laser energy. We assume complete thermalization, and choose a pressure at constant density to match this energy deposition at $t = 0$.

Our computational domain is a “wedge” configuration bounded by four planes with a diverergence angle sufficient to capture three wavelengths of the initial perturbation, which was taken to be 10% of the inner shell thickness in amplitude, with a wavenumber of approximately 80. The calculation used four adaptive meshing levels with a base resolution of 800 x 10 x 10, resulting in a finest resolution of 21600 x 270 x 270, or 1.5 billion zones, if fine grid covered the entire domain. The finest mesh size is initially 1/3 micron, although with the ALE mesh motion it becomes significantly smaller in regions of high compression. A closeup of the initial conditions on the outer surface of the inner shell are shown in Figure 13. Since the analytical solution to this problem is symmetric until the incoming shock impinges on the perturbation, we ran a 1D calculation to generate initial conditions for the 3D simulation. After initializing the 3D flowfield using the 1D calculation dataset, the inner shell perturbation was overlayed to generate the full 3D initial conditions. This saved a significant amount of relatively uninteresting calculation time.

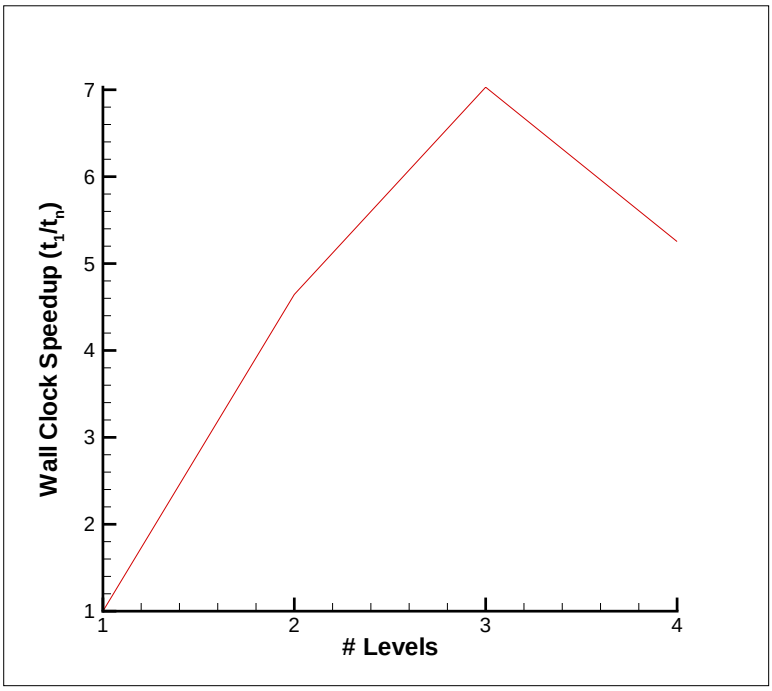


Fig. 7. Wall clock speedups obtained for the 1D Collella-Woodward blast wave problem using a fixed maximum resolution, with increasing numbers of adaptive levels, corresponding to coarser base resolutions. t_{8100} is the finest resolution single level wall-clock time and t_N the N-level adaptive solution time.

$(\|\rho_N - \rho_{8100}\|) / \|\rho_{8100}\|$				
	Effective Resolution			
Levels	8100	2700	900	300
1	0	.010546	.030476	.1035780
2	.001605	.011424	.039504	-
3	.004807	.020680	-	-
4	.015228	-	-	-

Fig. 8. $\frac{\|\rho_N - \rho_{8100}\|}{\|\rho_{8100}\|}$ for Colella-Woodward Blast Wave problem for a matrix of gridding strategies. "Effective resolution" is the finest resolution occuring in the calculation. ρ_{8100} is the solution for the maximum resolution single-level calculation, and ρ_N is the solution for the N-level adaptive calculation.

Speedup (t_{8100}/t_n)				
	Effective Resolution			
Levels	8100	2700	900	300
1	1	10.1	89.5	760
2	4.6	25	72	-
3	7.0	18	-	-
4	5.3	-	-	-

Fig. 9. Speedup for Colella-Woodward Blast Wave problem.

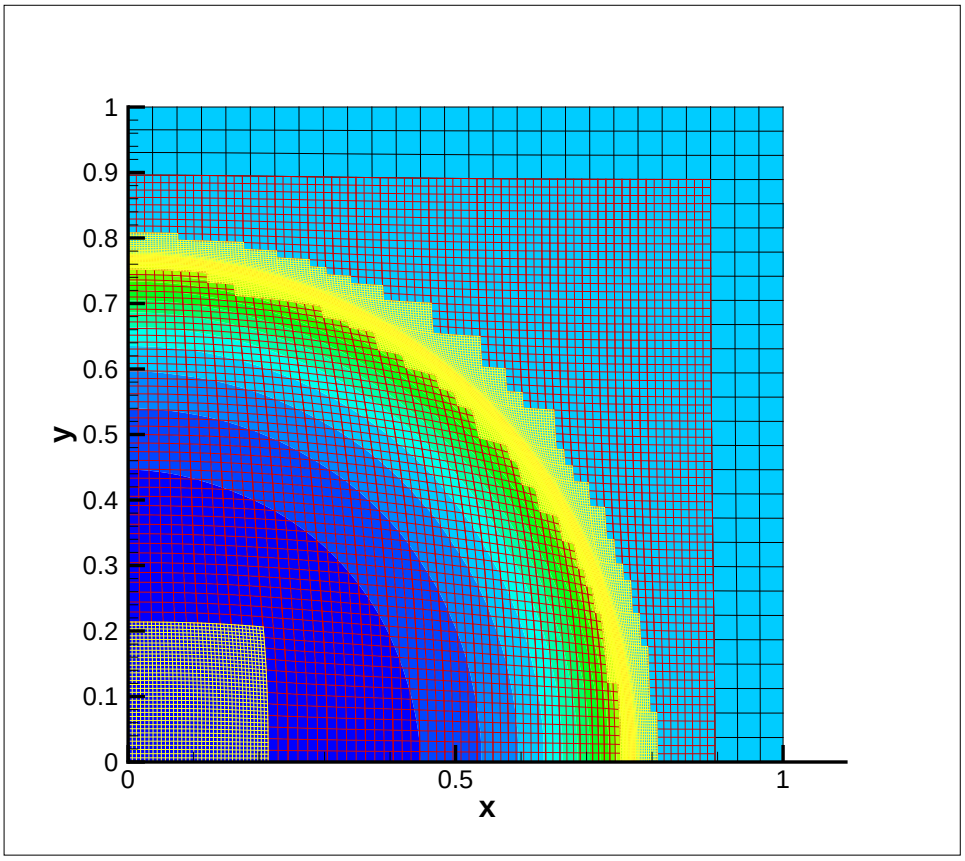


Fig. 10. Adaptive Lagrange+Remap Sedov problem solution using 3 levels of refinement.

$(\|\rho_N - \rho_{729}\|) / \|\rho_{729}\|$				
	Effective Resolution			
Levels	729^2	243^2	81^2	27^2
1	0	.00610	.0254	.0711
2	.00184	.00570	.0246	-
3	.00290	.00617	-	-
4	.00362	-	-	-

Fig. 11. $\frac{\|\rho_N - \rho_{729}\|}{\|\rho_{729}\|}$ for the Taylor-Sedov Blast Wave problem for a matrix of gridding strategies. "Effective resolution" is the finest resolution occuring in the calculation. ρ_{729} is the solution for the maximum resolution (729×729) single-level calculation, and ρ_N is the solution for the N-level adaptive calculation.

Speedup (t_{8100}/t_n)				
	Effective Resolution			
Levels	729^2	243^2	81^2	27^2
1	1	32.3	858	17900
2	11.3	159	2250	-
3	16.4	180	-	-
4	12.1	-	-	-

Fig. 12. Speedup for 2D Taylor-Sedov Blast Wave problem.

This calculation was then run on 404 processors of the TC2K Compaq cluster for 48 hours over 15,900 coarse time steps. The instability was well resolved, and the solution at the end of the calculation is shown in Figure 14 in an isosurface of density and a cutting plane showing contours of density. The black boxes represent refinement block boundaries.

The potential for 3D sub-micron resolution as demonstrated by ALE-AMR in the ICF context raises the possibility of decreasing dependence on mix modeling to determine yield degradation due to fluid mechanical instability growth. While it is certain that direct numerical simulation to the Kolmogorov scale is not attainable, it is an open problem to determine the range of scales which are sufficient to adequately resolve the yield degradation phenomenon. AMR techniques allow researchers to continue to push further into highly resolved solution spaces without inhibitive growth of computational resource requirements.

5 Conclusion and Future Work

We have successfully combined the ALE and AMR methods to create a new, more powerful adaptive method for solution of the equations of inviscid gasdynamics. We have presented both quantitative and qualitative evidence that the method improves

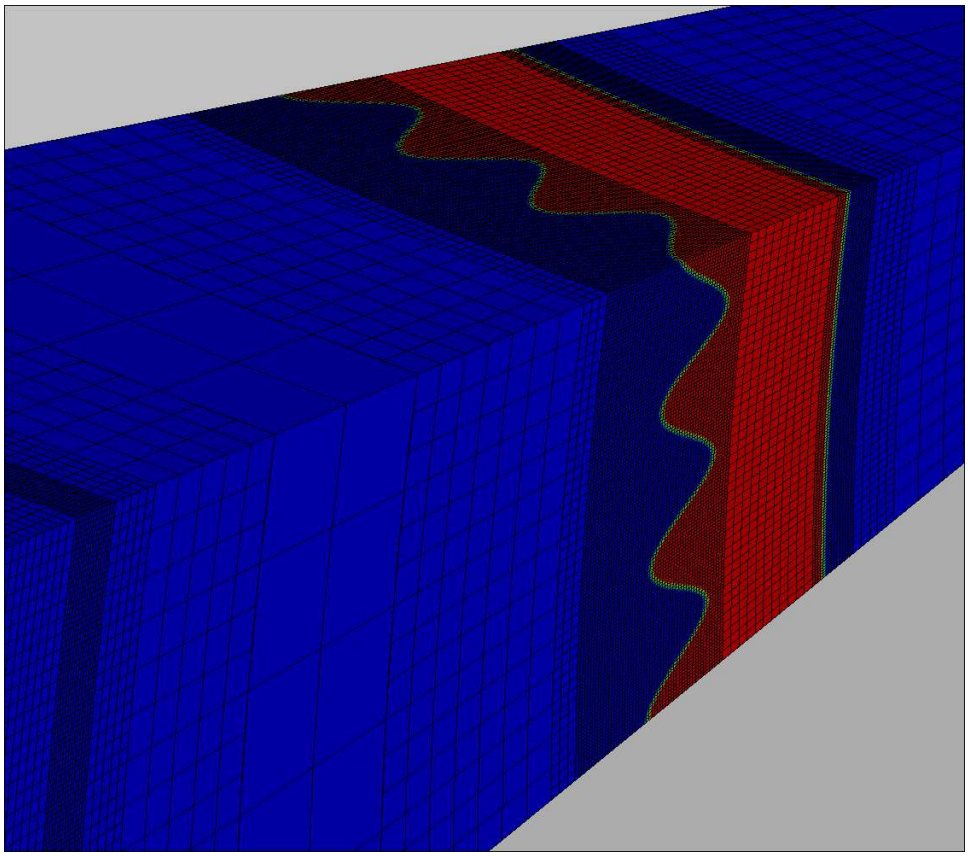


Fig. 13. Refinement on inner shell perturbation before shock impingement. Contours of density. Refined region at left is location of incoming shock.

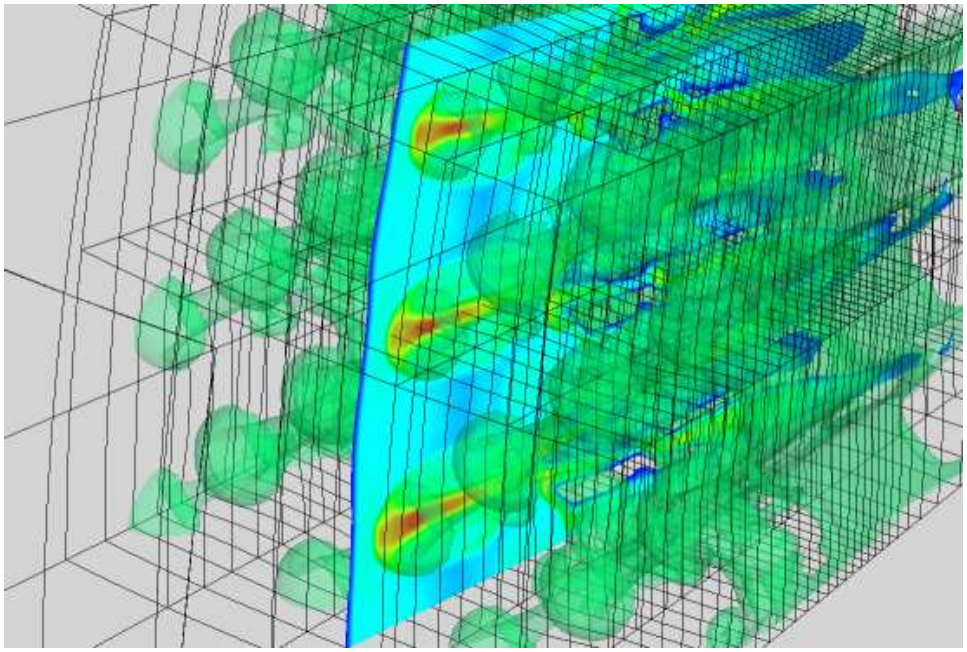


Fig. 14. Richtmeyer-Meshkov instability growth at later time. Isosurfaces of density. Black boxes represent refinement block boundaries. Cutting plane shows contours of density.

computational efficiency without undermining the robustness or accuracy of the underlying ALE method. The adaptive components of the algorithm are applicable to a class of methods that employ staggered variables of the type described, and are not limited to the particular details of the hydrodynamical or mesh relaxation methods we have employed and presented in this work.

Order-of-magnitude efficiency increases have been demonstrated on simple test problems in 1D and 2D, and a highly resolved proof-of-principle ICF model problem was demonstrated in 3D that would not have been feasible without AMR methods for the given computational resources.

In order to further explore the potential of ALE-AMR based methods, we are currently adding a radiation modeling capability, a reactive chemistry capability, as well as interface reconstruction for the modeling of multiple immiscible fluids. All three of these physics capabilities open up new opportunities to exploit the advantages of AMR to enable the solution of problems at higher resolution, or to eliminate ad-hoc or semi-empirical methods and replace them models based on first principles.

Acknowledgment

This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

References

[1] B. J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comp. Meth. Appl. Mech. Eng. 72 (1989) 205–350.

[2] J. Bell, M. Berger, J. Saltzman, M. Welcome, Three dimensional adaptive mesh refinement for hyperbolic conservation laws, SIAM J. Sci. Comp. 15 (1994) 127–138.

[3] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.

[4] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.

[5] R. Pember, J. Bell, P. Colella, W. Crutchfield, M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in complex geometries, J. Comput. Phys. 120 (1995) 278–304.

[6] A. Winslow, Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh, J. Comput. Phys. 1 (1967) 149–172.

[7] J. Saltzman, An unsplit 3-D upwind method for hyperbolic conservation laws, J. Comput. Phys. 115 (1994) 153–168.

[8] R. Tipton, unpublished report, Lawrence Livermore National Laboratory, 1990.

[9] M. L. Wilkins, Calculation of elastic-plastic flow, Meth. Comp. Phys. 3 (1964) 211–263.

[10] R. Sharp, HEMP advection model, Tech. Rep. UCID-17809, Lawrence Livermore National Laboratory (1978).

[11] M. L. Wilkins, Computer Simulation of Dynamic Phenomena, Springer Verlag, 1999.

[12] R. Christensen, Godunov methods on a staggered mesh — An improved artificial viscosity, Tech. Rep. UCRL-JC-105269, Lawrence Livermore National Laboratory (1990).

[13] L. Margolin, J. Pyun, A method for treating hourglass patterns, Tech. Rep. LA-UR-87-439, Los Alamos National Laboratory (1987).

[14] R. Pember, R. Anderson, Comparison of direct eulerian godunov and lagrange plus remap artificial viscosity schemes for compressible flow, Tech. Rep. AIAA Paper 2001-2644 (2001).

[15] J. Dukowicz, Efficient volume computation for three-dimensional hexahedral cells, Journal of Computational Physics 74 (2) (1988) 493–496.

[16] O. V. Ushakova, Conditions of non-degeneracy of three-dimensional cells. a formula of a volume of cells., SIAM Journal of Scientific Computation 23 (4) (2001) 1274–1290.

[17] A. Winslow, Equipotential zoning of two dimensional meshes, Tech. Rep. UCRL-7312, Lawrence Livermore National Laboratory (1963).

[18] R. Tipton, Grid optimization by equipotential relaxation, lawrence Livermore National Laboratory, 1992.

[19] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.

[20] E. J. Caramana, D. E. Burton, M. J. Shashkov, P. P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, Journal of Computational Physics 146 (1) (1998) 227–262.

[21] E. Caramana, P. Whalen, Numerical preservation of symmetry properties of continuum problems, Journal of Computational Physics 141 (2) (1998) 174–198.

[22] E. Caramana, Issues of an eulerian?ale algorithm based on a compatible, staggered-grid, lagrangian step, SIAM, San Diego, CA, 2001.

[23] R. Hornung, S. Kohn, Managing application complexity in the samrai object-oriented framework, submitted to Concurrency: Practice and Experience UCRL-JC-141749, Lawrence Livermore National Laboratory, August, 2001. See also *www.llnl.gov/CASC/SAMRAI*.

[24] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.