

# RHALE: A 3-D MMALE CODE FOR UNSTRUCTURED GRIDS

*James S. Peery, Kent G. Budge, Michael K.W. Wong, and Tim G. Trucano*  
*Sandia National Laboratories*  
*Computational Physics Research and Development Department*  
*Albuquerque, NM 87185*

This paper describes RHALE, a multi-material arbitrary Lagrangian-Eulerian (MMALE) shock physics code. RHALE is the successor to CTH, Sandia's 3-D Eulerian shock physics code, and will be capable of solving problems that CTH cannot adequately address.

We discuss the Lagrangian solid mechanics capabilities of RHALE, which include arbitrary mesh connectivity, superior artificial viscosity, and improved material models. We discuss the MMALE algorithms that have been extended for arbitrary grids in both two- and three-dimensions. The MMALE addition to RHALE provides the accuracy of a Lagrangian code while allowing a calculation to proceed under very large material distortions. Coupling an arbitrary quadrilateral or hexahedral grid to the MMALE solution facilitates modeling of complex shapes with a greatly reduced number of computational cells.

RHALE allows regions of a problem to be modeled with Lagrangian, Eulerian or ALE meshes. In addition, regions can switch from Lagrangian to ALE to Eulerian based on user input or mesh distortion. For ALE meshes, new node locations are determined with a variety of element based equipotential schemes. Element quantities are advected with donor, van Leer, or Super-B algorithms. Nodal quantities are advected with the second order SHALE or HIS algorithms. Material interfaces are determined with a modified Young's high resolution interface tracker or the SLIC algorithm.

RHALE has been used to model many problems of interest to the mechanics, hypervelocity impact, and shock physics communities. Results of a sampling of these problems are presented in this paper.

## INTRODUCTION

Theoretical studies of impact phenomena make extensive use of computer simulations. The computational kernel of these simulation codes (frequently called "shock codes", "hydro codes", or "wave codes") must be highly efficient and capable of handling large material deformations and strong shocks.

The hydrodynamics code CTH (McGlaun *et al.* 1989) is used extensively in the simulation of impact events and other problems involving large deformations. This code uses an Eulerian finite-volume formulation of the equations of motion, includes sophisticated equations of state, and

has strength and fracture models. Its results have been extensively checked against experimental data for regimes where such data are available and have been shown to be remarkably accurate for many classes of problems. It is heavily supported and widely used throughout the DOE, DoD, and NASA communities and its capabilities continue to be enhanced. For example, a version suitable for massively parallel computers, PCTH, is currently in development (Robinson *et al.*, 1992).

However, CTH suffers from the limitations inherent in any hydrodynamics code based on a finite difference, purely Eulerian formulation. Since material flows through a fixed mesh, advection algorithms are required which introduce numerical dispersion and dissipation. Sliding between surfaces is difficult to model. Since finite difference formulations generally use a regular mesh, the size of mesh elements can be varied only in limited ways, which means that element dimensions tend towards the smallest length scale of interest in the calculation. Large regions of empty space must also be included in many grids. The latter two limitations make many interesting calculations prohibitively expensive.

Hydrodynamics codes based on finite element, Lagrangian formulations avoid these difficulties. Since a Lagrangian mesh moves with the material, no advection takes place and the associated numerical dispersion and dissipation is avoided. Sliding is handled much more accurately than in an Eulerian code. Using a finite element formulation permits arbitrary element connectivities and volumes and thus can more easily reflect the very different length scales in different portions of the problem. Finite element, Lagrangian codes are thus preferred for problems requiring high numerical accuracy or where differing length scales are important and in which large deformations do not take place. However, Lagrangian codes fail when large deformations occur, since highly distorted elements lose accuracy or may even invert (thus halting the calculation).

The RHALE code uses a finite element, arbitrary Lagrangian-Eulerian (ALE) formulation in an effort to get the best of both worlds. Although users can specify a purely Lagrangian or purely Eulerian calculation for portions of the problem domain, the normal mode of operation is for a calculation to proceed in Lagrangian fashion until elements become highly distorted (as measured by various criteria specifiable by the user). At this point, material is permitted to flow between elements in the most deformed portion of the mesh so as to reduce the distortion to acceptable levels. This formulation permits accurate treatment of contact surfaces and has less numerical dissipation than a purely Eulerian calculation (since less advection takes place). Quite large deformations can take place without the calculation failing, unlike a purely Lagrangian calculation. We are currently developing ALE algorithms that also preserve variable mesh scaling.

Another innovative feature of the RHALE development project is that the code is being written in C++. We believe that the object-oriented programming paradigm, which C++ supports, is the best approach to the development of such a large and complicated code. We are addressing some of the known efficiency problems with C++ when utilizing operator overloading through a variety of programming techniques, including reference counting, deferred expression evaluation, and hidden calls to assembly language routines.

RHALE is being developed in parallel with PCTH and the two projects are expected to share much of their coding. However, RHALE represents the next generation of strong shock codes and may eventually replace CTH and PCTH. It should be capable of performing any calculation that CTH can perform (though perhaps somewhat less efficiently) and will execute additional calculations that CTH cannot currently treat.

## RHALE MESH AND LAGRANGIAN HYDRODYNAMICS

RHALE supports three types of mesh movement: pure Lagrangian, pure Eulerian, and ALE. All meshes are based on a linear finite element formulation and may have arbitrary connectivity between elements. An arbitrary connectivity mesh allows any number of elements to share a common node. The user may specify different mesh movements for different regions of a problem, so that, for example, the projectile and target region in an impact calculation may be calculated in ALE mode while the far-field target response calculation may take place in pure Lagrangian mode.

The finite element framework of RHALE consists of uniform-strain quadrilaterals (in 2-D) or hexahedrons (in 3-D). Frame invariance for the constitutive models is achieved by using a corotational frame formulation similar to that of the PRONTO finite-element code (Taylor and Flanagan, 1987). The row-summed lumped mass is used to diagonalize the mass matrix (avoiding large matrix inversions) and the time integration is carried out using an explicit central-difference method (Hughes, 1987). Thus, individual time steps are computationally efficient but the maximum time increment is limited by a Courant condition.

We have experimented extensively with a variety of artificial viscosity and anti-hourglassing formulations to control mesh keystoneing. There are currently five artificial viscosity and four hourglass “viscosity” options available to users, plus an hourglass “stiffening” option. None of the tensor artificial viscosity options have proven consistently superior to the scalar bulk viscosity formulation that has been used for over thirty years. However, we are continuing to experiment with the spurious vorticity correction methods discussed by Dukowicz and Meltz (1992) and may eventually incorporate a spurious vorticity control method in RHALE. Since RHALE uses a uniform-strain quadrilateral, spurious zero-energy modes (hourglass modes) exist and must be damped. We provide both the hourglass control method used in PRONTO (Taylor and Flanagan, 1987), which is applicable to materials with strength, and a version of the Margolin-Pyun method (Margolin and Pyun, 1987), which is applicable to fluids. The calculations presented here all use a cell edge projected viscosity due to Barton and an hourglass control similar to that suggested by Margolin and Pyun (1987).

RHALE can handle 2-D Cartesian, 2-D axisymmetric, and 3-D Cartesian geometries. We have experimented with both a volume-weighted and an area-weighted axisymmetric element (Taylor and Flanagan, 1987). The former has the advantages that the nodal lumped masses are only slightly time dependent and that the element passes a restricted patch test; however, the element fails to maintain symmetry. The latter formulation has the advantages of a closer correspondence between the 2-D Cartesian and axisymmetric formulations and of being better conditioned for implicit methods. Since RHALE must maintain symmetry for isotropic implosions, we are currently using an area-weighted axisymmetric element.

## MATERIAL MODELS AND MATERIAL DATA

RHALE uses material libraries being developed for both RHALE and PCTH. There are currently five material model libraries implemented in RHALE:

- The *Equation-of-State* library calculates the thermodynamic state of a material (pressure, temperature, internal energy, and sound speed). This library includes several Mie-Gruneisen models, ideal gas, JWL equations of state for explosives and the SNL-SESAME tabular equation of state.
- The *Constitutive* library calculates the stress tensor associated with the material. This library includes linear elastic, elastic-perfectly plastic, the Johnson-Cook viscoplastic, and the Johnson-Holmquist ceramic models, plus others.
- The *Fracture* library updates the state of the material to limit the tensile response of material in an element. Fracture can occur based on a strain and strain-rate independent stress threshold value. The state of the material is then modified in a consistent manner constrained by “fractured” or “damaged” values.
- The *Burn* library computes the state of high explosive material from information on the unreacted and reacted states and the reaction rate.
- The *Thermal Conductivity* library computes the thermal conductivity of a material for thermal conduction modeling.

## MMALE ALGORITHMS

To the authors knowledge, RHALE is the only unstructured mesh code with second-order accurate MMALE algorithms: second-order monotonic advection algorithms and a high resolution interface tracker. The MMALE addition to RHALE involves remeshing to relieve distortion and remapping element and nodal variables to the new mesh while conserving global quantities. These steps and their substeps will be described in the paragraphs that follow.

For an unstructured mesh it is advantageous to use algorithms that are element based. Developing generalized node based algorithms for an unstructured mesh is difficult since a node can have an arbitrary number of element connections and therefore an arbitrary number of closest node neighbors. Two element based algorithms are the key to RHALE's MMALE hydrodynamics: the half interval shift algorithm, HIS, (Benson, 1992) and a variational approach for equipotential schemes (Tipton, 1992).

### Remesh

The remeshing phase of the MMALE method determines new node locations that will partially alleviate the associated elements' distortion. In RHALE, a node can be of type Lagrangian, single material ALE (SALE), multi-material ALE (MMALE), or Eulerian, reflecting the type of mesh of which it is part. Both SALE and MMALE nodes require algorithms to determine their movement; however, they differ in their movement if they are connected to two or more mesh regions. SALE nodes preserve material interfaces among the mesh regions while MMALE nodes allow materials to cross mesh boundaries and therefore, mix. Nodes located on physical boundaries require special algorithms to determine their movement. Development of boundary node relocation algorithms is an ongoing research effort in RHALE.

Remeshing is a three step process. First, nodes that meet the criteria for movement are tagged. Next, new positions are calculated for candidate nodes. Eulerian nodes are moved to their original location, Lagrangian nodes remain fixed in the remesh/remap phase, and ALE nodes are moved based on distortion criteria. Finally, the actual movement of the selected ALE nodes is limited to some user specified fraction of their calculated movement.

One set of criteria for moving an ALE node is given in the HEMP code (Sharp and Barton, 1981). Node movement criteria amount to two tests: an angle test and a volume test. For these nodes, the angles that are formed by the element edge vectors that originate at the node are calculated. The ideal situation is for the angles to be 90 degrees. Additionally, the volumes of each of the elements that are connected to a node are compared. Ideally, they should all be equal. These calculations must be performed for each of the elements that surround a tagged node. To determine if an ALE node should move, a user defined minimum angle and minimum to maximum volume ratio criteria are used.

Node and Element Ordering. Determining element and node neighbors is trivial for logically regular meshes. However, for arbitrary meshes, determining this information requires sophisticated algorithms and logic.

Many robust algorithms for determining new node locations and advecting nodal variables such as momentum require parameters of node neighbors. The identification of a node's closest neighbors is not a trivial problem for unstructured meshes and should be avoided if possible. Explicitly storing this information or using a searching algorithm would make the ALE formulation unrealistic if not impossible due to the enormous memory and CPU burden. To overcome this problem in two dimensions, the methods of element and node ordering described by Benson (1989) have been used with success, along with node-based algorithms. However, there is no known extension of Benson's methods for three dimensional meshes. Therefore, RHALE currently uses element based algorithms for node relocation and nodal variable advection. These algorithms eliminate the identification of node neighbors.

Nodes that comprise each element are stored in a connectivity array. This array plays a pivotal role in the assembly operation of finite element codes. With this array, element based

algorithms have direct access to nodal parameters. However, element based advection algorithms, must not only know the nodes that comprise each element but also element neighbors. This information is relatively easy to determine through the connectivity array and is stored in RHALE.

Applying Equipotential Solutions to the Mesh. In order to perform the remeshing phase of MMALE, one must determine where an ALE node is to be moved. There are many techniques for deciding where to move a node, but one of the most successful and the one used in our MMALE algorithm for two dimensional problems is a method based on equipotential smoothing (Winslow and Barton, 1982). Tipton (1992) developed a variational approach to equipotential smoothing that is applicable to two- and three-dimensional unstructured meshes. Tipton's method is element based and can be shown to reduce to Winslow's method for structured meshes. Tipton's method has yet to be published and thus will not be described in this paper. However, Winslow's method has been incorporated into RHALE as a node based algorithm for two-dimensional meshes and is described below.

Winslow's method is based on inverting Laplace's equation. This is given as

$$\alpha x_{\phi\phi} - 2\beta x_{\phi\phi} + \gamma x_{\phi\phi} = 0 \quad (1)$$

$$\alpha y_{\phi\phi} - 2\beta y_{\phi\phi} + \gamma y_{\phi\phi} = 0 \quad (2)$$

where

$$\alpha = x_{\phi}^2 + y_{\phi}^2 \quad (3)$$

$$\beta = x_{\phi}x_{\phi} + y_{\phi}y_{\phi} \quad (4)$$

$$\gamma = x_{\phi}^2 + y_{\phi}^2 \quad (5)$$

These equations can be approximated with second order central differencing techniques and are given as

$$x' = \frac{1}{2(\alpha + \gamma)} \left( \alpha(x_4 + x_8) + \gamma(x_2 + x_6) - \frac{1}{2}\beta(x_3 - x_5 + x_7 - x_9) \right) \quad (6)$$

$$y' = \frac{1}{2(\alpha + \gamma)} \left( \alpha(y_4 + y_8) + \gamma(y_2 + y_6) - \frac{1}{2}\beta(y_3 - y_5 + y_7 - y_9) \right) \quad (7)$$

where  $x'$  and  $y'$  are the new node locations,  $x_n$  and  $y_n$  ( $n = \text{even}$  for adjacent neighbors,  $n = \text{odd}$  for diagonal neighbors) are node neighbors and

$$x_{\phi} = \frac{1}{2}(x_4 - x_8) \quad (8)$$

$$x_{\phi} = \frac{1}{2}(x_2 - x_6) \quad (9)$$

$$x_{\phi\phi} = \frac{1}{4}(x_3 - x_5 + x_7 - x_9) \quad (10)$$

Equations 8, 9, and 10 for the  $y$  coordinate are similarly developed.

These equations can be applied for nodes connected to four elements. For the other nodes, a method developed by Budge (1991) is used. In addition, the user can request that Budge's method or an element average method be applied to the mesh.

Finding the  $x'$  and  $y'$  coordinates for each node is best achieved with an iterative procedure since incremental control over the extent of a nodes movement can be maintained. A Jacobi iteration (Press *et al.*, 1986) can be used to solve these equations; however, MMALE algorithms are interested in making small changes to the mesh and therefore these equations are generally iterated only once per remap step. In RHALE, the user is allowed to control the number of iterations performed on these equations.

Limiting Node Movement. Currently, RHALE has no default limiting criteria placed on the coordinates determined from the equipotential solution. In other words, the displacement from the old to new coordinates is multiplied by a fraction of one. The user can specify a fraction of less than one. One should be concerned about a node being moved such that its new position overlaps the old position of one of its neighbors. Advection algorithms can produce nonphysical results if a node is moved too far in a single time step. In other words, although these mesh relocation schemes will produce smooth meshes given enough iterations, there are limits in other algorithms to the extent that a mesh can be smoothed in a single time step. Test cases have not shown this to be a problem for a single iteration of equipotential equations.

### Remap

The explicit relocation of nodes creates an advection problem. Since time is not involved, the advection problem simplifies to a remapping problem. The remap phase consists of determining volumes fluxes, determining material fluxes from volume fluxes through the use of an interface tracker, and advecting material variables and velocities. The various aspects of the remap phase will be discussed in the following sections.

Determining Volume Fluxes. The first step in any advection scheme is to determine the volume fluxes created by the movement of nodes. When a node is moved, volume fluxes are generated through the faces/sides of the elements. The volume flux through an element face is given by the change in volume produced by the movement of nodes on that face. With this definition of volume flux, the new volume of an element is given by

$$V^{new} = V^{old} + \sum_{i=1}^{ns} \Delta V_i \quad (11)$$

where  $i$  refers to an element's face and  $ns$  refers number of element faces (four in two dimensions and six in three dimensions).

Material Fluxes. In MMALE, a simulation generally begins with single material elements. If a problem remained Lagrangian, the elements would remain single material; however, for many problems of interest, elements quickly become distorted and remeshing/remapping is used to relieve distortion. The remeshing/remapping phase can create multi-material elements and thus, the volume of each material within the volume fluxes must be determined.

Currently, RHALE has two options for determining material volume fluxes: the Sandia Modified Youngs' Reconstruction Algorithm, SMYRA (Youngs, 1987; Bell and Hertel, 1992) and the SLIC interface tracking algorithm (Noh and Woodward, 1976). The SLIC algorithm places materials within a cell in an order from left to right by determining the materials contained in neighboring cells and assuming a planer interface between materials that intersects the cell at 90 degrees. The material order determines the precedence of materials to be used in the material makeup of the volume flux. SLIC is considered a first order algorithm because it limits the material planes to have zero slope within the element. On the other hand, SMYRA attempts to construct the geometry of the material interfaces in a cell based on the materials of all cells with a common node to the cell in question. SMYRA constructs plane material interfaces that can intersect the element at any angle. SMYRA is considered a second order accurate algorithm since the material planes have a non-zero slope.

Element Centered Advection. Once the nodes have been moved, the element centered variables must be advected to their new locations. Isotropic advection assumes that the material is advected through all faces of the element simultaneously. This discussion will assume that advection occurs isotropically. RHALE, however, makes one-dimensional remap sweeps through the mesh to aid in corner coupling.

Advection algorithms are expressed in volume and mass coordinates. Using the above definition of volume flux, a newly advected element centered variable is given by

$$f_j^{new} = \frac{f_j^{old} V_j^{old} + \sum_{i=1}^{ns} \tilde{f}_i \Delta V_j}{V_j^{new}} \quad (12)$$

where the  $\tilde{f}_i$ 's are determined by the type and order of the advection algorithm and  $V$  represents either volume or mass. Most intensive quantities are fluxed with mass.

The simplest advection scheme is a first order advection method since it does not involve evaluating derivatives; however, it is also the most diffusive. The advecting material is assumed to carry the average value of an element centered variable from which the material originated. This is analogous to first order upwinding in finite difference methods. For a first order method, the  $\tilde{f}_i$ 's are given by

$$\tilde{f}_i = \begin{cases} f_n & \text{if } \Delta V_{ji} > 0 \text{ (influx)} \\ f_i & \text{if } \Delta V_{ji} < 0 \text{ (outflux)} \end{cases} \quad (13)$$

where  $n$  is the neighbor's value.

Two second order advection methods are available in RHALE, van Leer (van Leer 1984) and Roe's Super-B pointed out to the authors by Christensen (1991). Both are based on slopes among the donor, acceptor and behind elements. For van Leer, this relationship is shown in the figure below where the x-axis represents volume or mass and the y-axis represents a quantity to be fluxed.

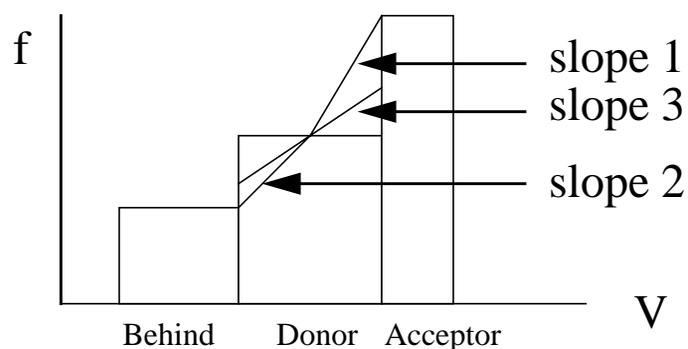


Figure 1. Slopes Used in van Leer Advection Algorithm

The three slopes are given by

$$s1 = (2(f_{i+1} - f_i)) / X_i \quad (14)$$

$$s2 = (2(f_i - f_{i-1})) / X_i \quad (15)$$

$$s3 = (2(f_{i+1} - f_i))/(X_{i+1} + X_i) + (2(f_i - f_{i-1}))/(X_i + X_{i-1}) \quad (16)$$

For the van Leer scheme, the slope used is given by

$$slope' = \begin{cases} 0 & \text{if } \left| \frac{sign(s1) + sign(s2) + sign(s3)}{3} \right| \neq 1 \\ sign(s1)min(|s1|, |s2|, |s3|) & \end{cases} \quad (17)$$

and the value used for  $\tilde{f}_i$  is given as

$$\tilde{f}_i = \begin{cases} f_n + \frac{slope'}{2}(V_n - \Delta V_{ni}) & \text{if } \Delta V_{ni} > 0 \quad (\text{inflow}) \\ f_i + \frac{slope'}{2}(V_i + \Delta V_{ni}) & \text{if } \Delta V_{ni} < 0 \quad (\text{outflow}) \end{cases} \quad (18)$$

All calculations presented in this paper were run with the van Leer advection algorithm.

Vertex Centered Advection. Vertex centered advection is very similar in concept to element centered advection and is required for advecting nodal quantities such as momentum. For vertex centered variables, however, a staggered grid exists with vertices at the center. For a logically connected grid, donor, acceptor and behind nodes are known for each direction and staggered fluxes and advection quantities can be determined. Amsden and Hirt (1973) developed the YAQUI algorithm for this type of mesh. However, determining staggered mesh quantities can be very expensive. In addition, for an arbitrary mesh, the staggered mesh can have a very odd shape and advection quantities are difficult if not impossible to calculate. Attempts to eliminate these problems have resulted in the development of several element centered methods for advecting vertex centered variables.

In RHALE there are three element centered options for nodal vertex centered variables, SALE (Amsden *et al.*, 1980), SHALE (Margolin and Beason, 1988), and HIS (Benson 1992). All of these methods project nodal variables to element centers, advect the new element centered variables with the element centered methods described above, and project these values back to the nodes. SALE simply averages nodal variables to calculate element centered quantities and thus is first order. SHALE averages both the nodal variable and its derivatives and thus is second order. Both SALE and SHALE are monotonic with respect to the element centered advection but not with respect to the nodal variables. This can lead to new maxima or minima for nodal quantities. Benson's HIS algorithm is second order and monotonic. The HIS algorithm places nodal variables at element centers, advects the element quantities and assembles the values back at the nodes. Therefore, a node's value is advected as an element quantity for every element that it is a member of and the results are assembled back at the node. The HIS algorithm is therefore more expensive than either the SALE or SHALE methods but the attribute of being monotonic outweighs this concern. The HIS method has been extended to arbitrary meshes in RHALE and has given excellent results. All calculations shown in this paper were run with the HIS algorithm.

## APPLICATIONS

In order to demonstrate the unique capabilities of the RHALE code, we briefly survey a variety of problems of interest in the mechanics, hypervelocity impact, and shock physics communities.

### Penetration Mechanics

First, we present calculations that simulate experiments performed by Forrestal, Luk, and Brar (1990) of conical tungsten penetrators impacting and penetrating aluminum plates at velocities ranging from 200 to 1200 m/s. These calculations were performed using an axisymmetric finite element grid comprised of Eulerian, Lagrangian, and ALE mesh regions. These regions and the initial and final configurations are shown in Figure 2. This simulation had an impact velocity of 726 m/s.

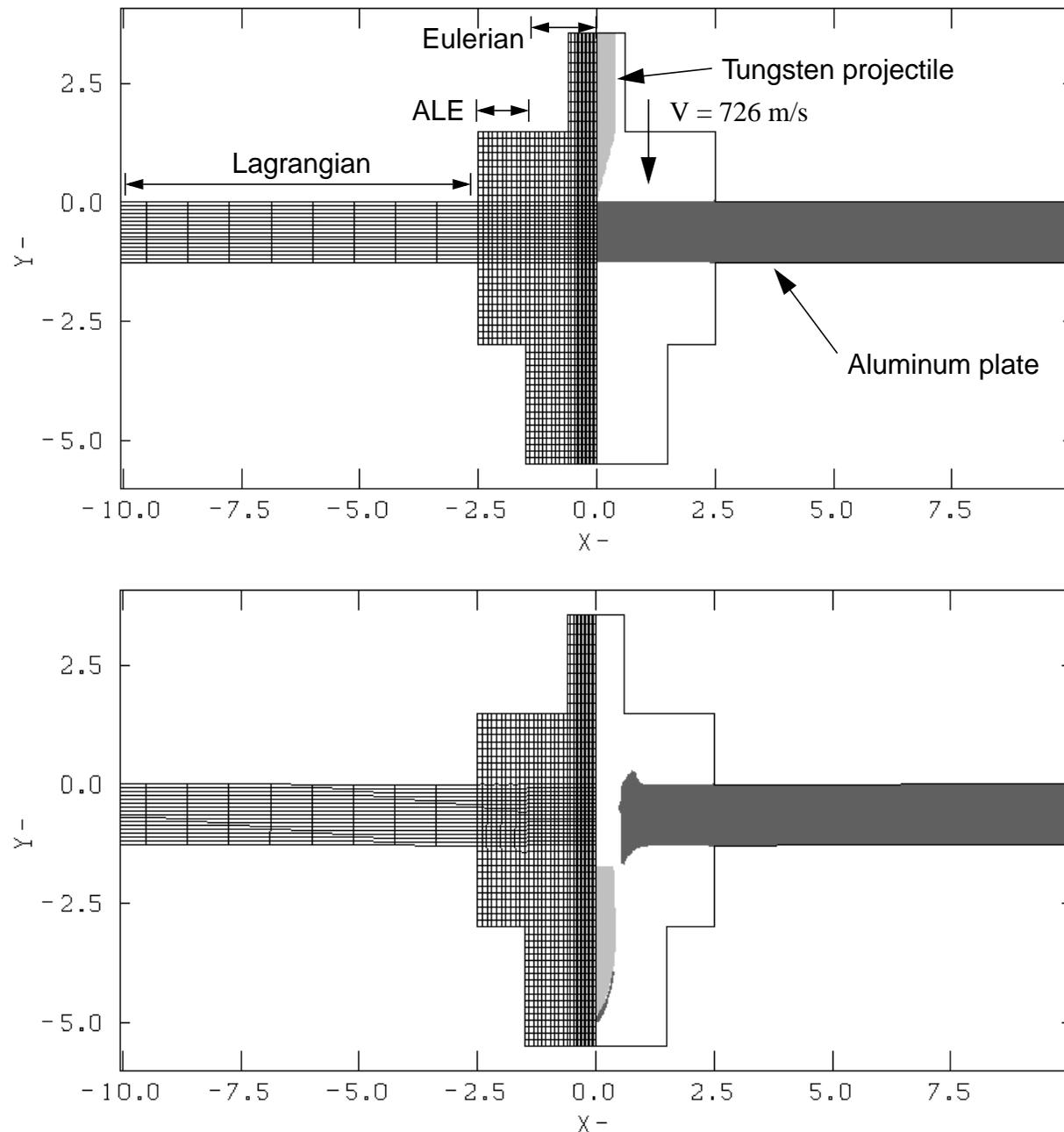


Figure 2. Initial (top) and final (bottom) configurations of mesh and materials for tungsten projectile perforating an aluminum plate at 726 m/s. Mesh types are annotated in top figure.

The tungsten penetrator was modeled as an elastic-plastic metal with a yield stress of  $15 \cdot 10^9$  dynes/cm<sup>2</sup> and a density of 18.5 g/cm<sup>3</sup>. The aluminum target was modeled as an elastic-plastic material with a density of 2.66 g/cm<sup>3</sup> and a yield stress and hardening modulus of  $2.76 \cdot 10^9$  and  $1.24 \cdot 10^9$  dynes/cm<sup>2</sup>, respectively. Both materials had a simple void insertion model to limit the tensile stresses in the material.

A boundary layer algorithm simulated sliding between the penetrator and plate materials in

the Eulerian mesh region. This feature limited excess erosion, distortion, and velocity degradation of the penetrator that is common in multi-material Eulerian calculations. This algorithm treats the interface between the materials as a boundary layer a few elements thick. Stresses in the boundary layer are modified to allowing sliding of the materials relative to each other. An artifact of this algorithm is that the projectile has a thin layer of the target material over the nose after perforating the target as shown in Figure 2. This algorithm is the same algorithm implemented in CTH (Silling, 1992).

A comparison of the impact velocity to residual velocity of the projectile after perforating the plate is shown in Figure 3. The RHALE calculations were able to closely reproduce the residual velocity values measured in the experiments. Also, the lowest velocity simulation did not produce perforation of the target which was consistent with the experiment.

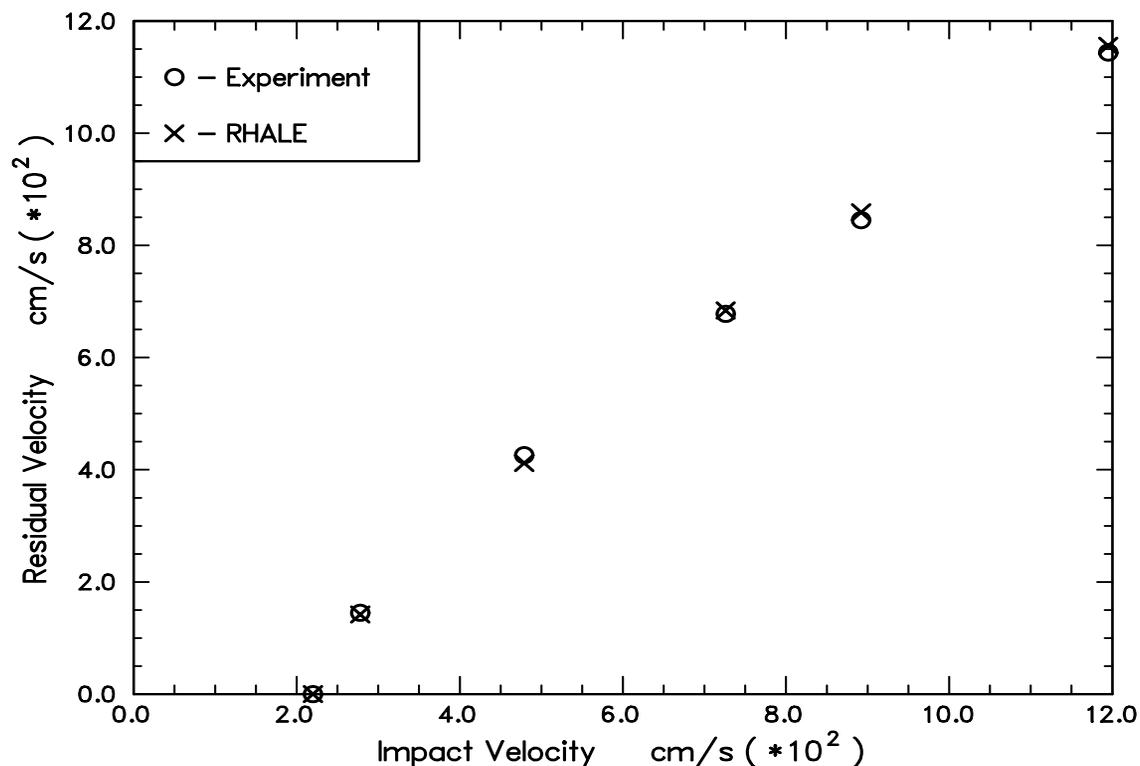


Figure 3. Comparison of RHALE calculations and experiments of impact velocity versus residual velocity after perforation of plate.

#### Structure-Media Interaction

This calculation was performed to demonstrate the capability of RHALE to simulate flow of a solid media around a structure under high pressures. This calculation consists of three regions: a region of high pressure gas, a region of solid media, and the structure. Plots of the mesh in the initial and deformed configurations are shown in Figures 4 and 5. The gas region simulates detonation products and is at a high pressure. A contact surface separates the solid media and the structure. The solid media region uses an ALE mesh. The ALE mesh allows material to flow between elements to reduce mesh distortion and the possibility of element inversions. The structure uses a Lagrangian mesh.

A similar calculation was attempted using a Lagrangian mesh in the solid region. Without nodal relocation and material advection, this calculation failed due to element inversions as material attempted to flow around the structure.

#### Anisotropic 2-D Implosion

This calculation is an example of the anisotropic implosion of a two layer capsule that is similar to an inertial confinement fusion target capsule. The pressure boundary condition has a 0.2%

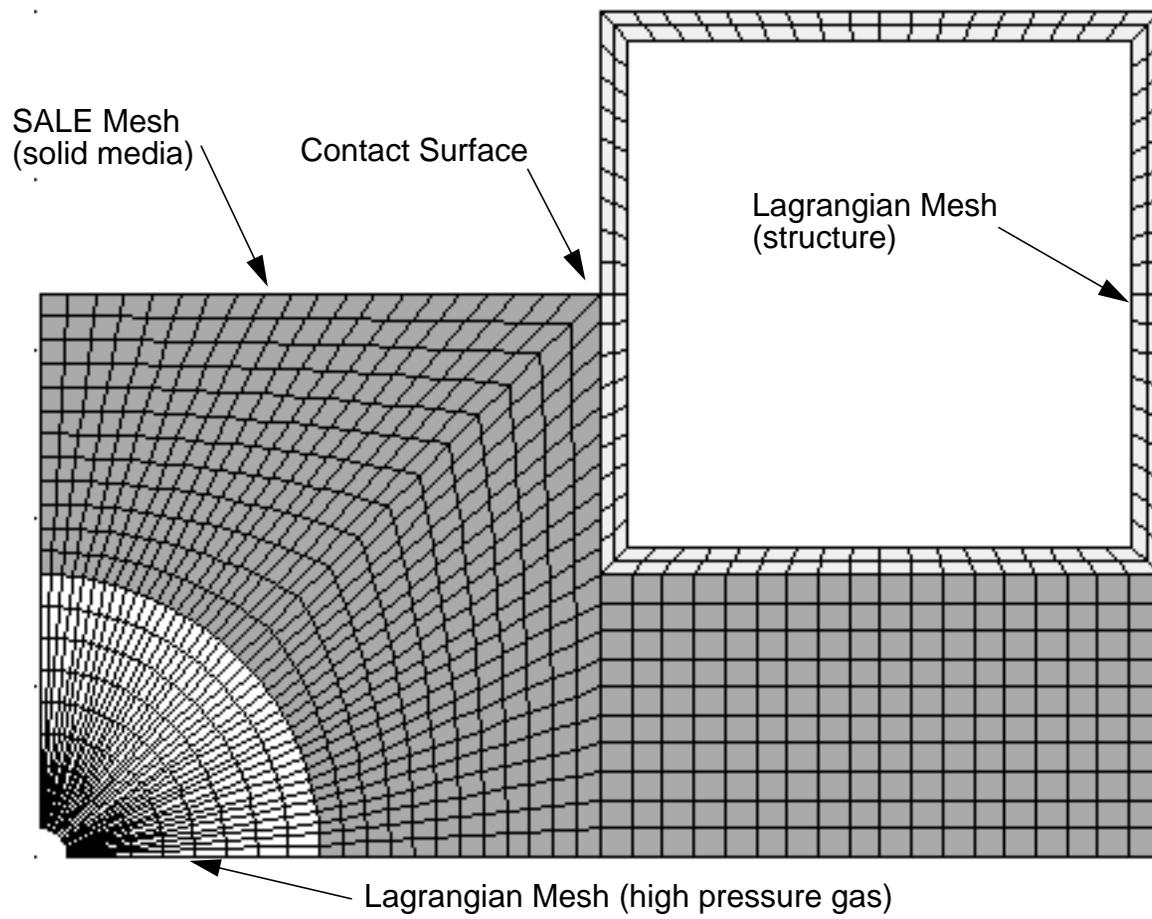


Figure 4. Configuration for structure-media interaction simulation, showing mesh types and contact surface.

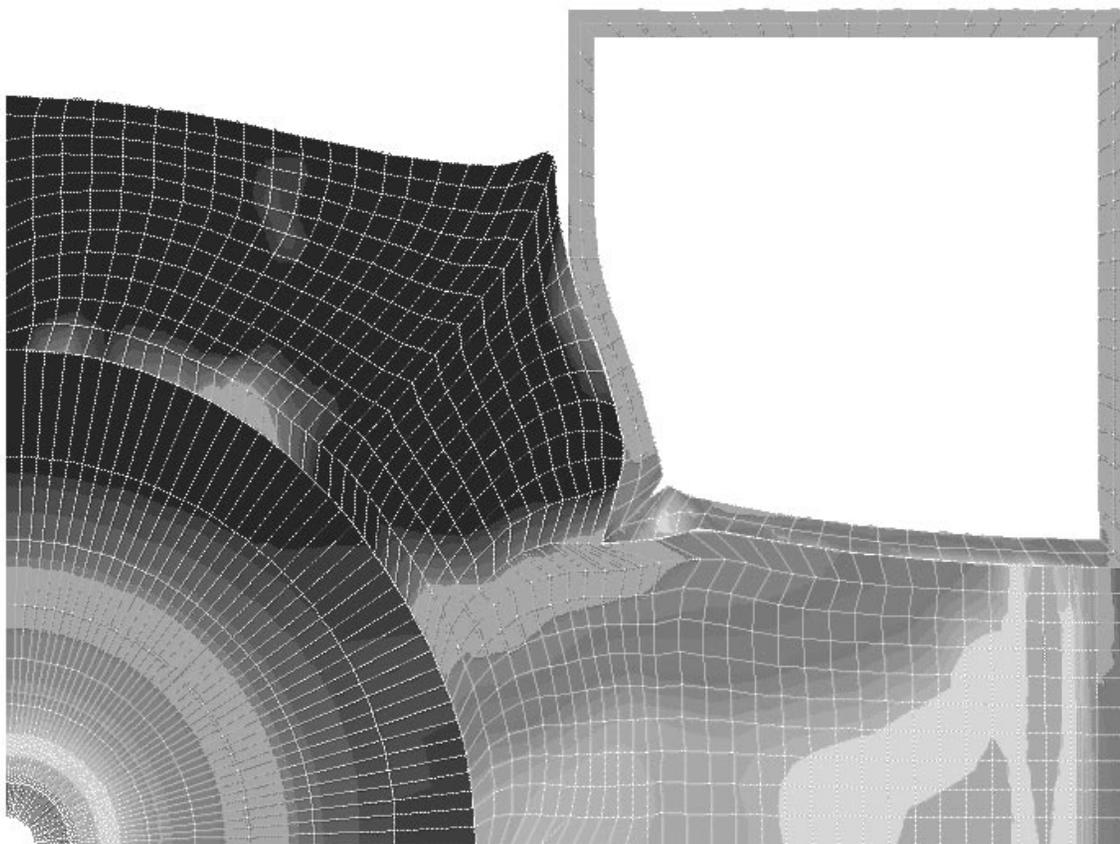


Figure 5. Final configuration of structure-media interaction simulation. Shading indicates pressure contours in each mesh region (lighter shading is higher pressure).

$P_6$  (Legendre polynomial) pole to equator asymmetry and varies with time as shown in Figure 6.

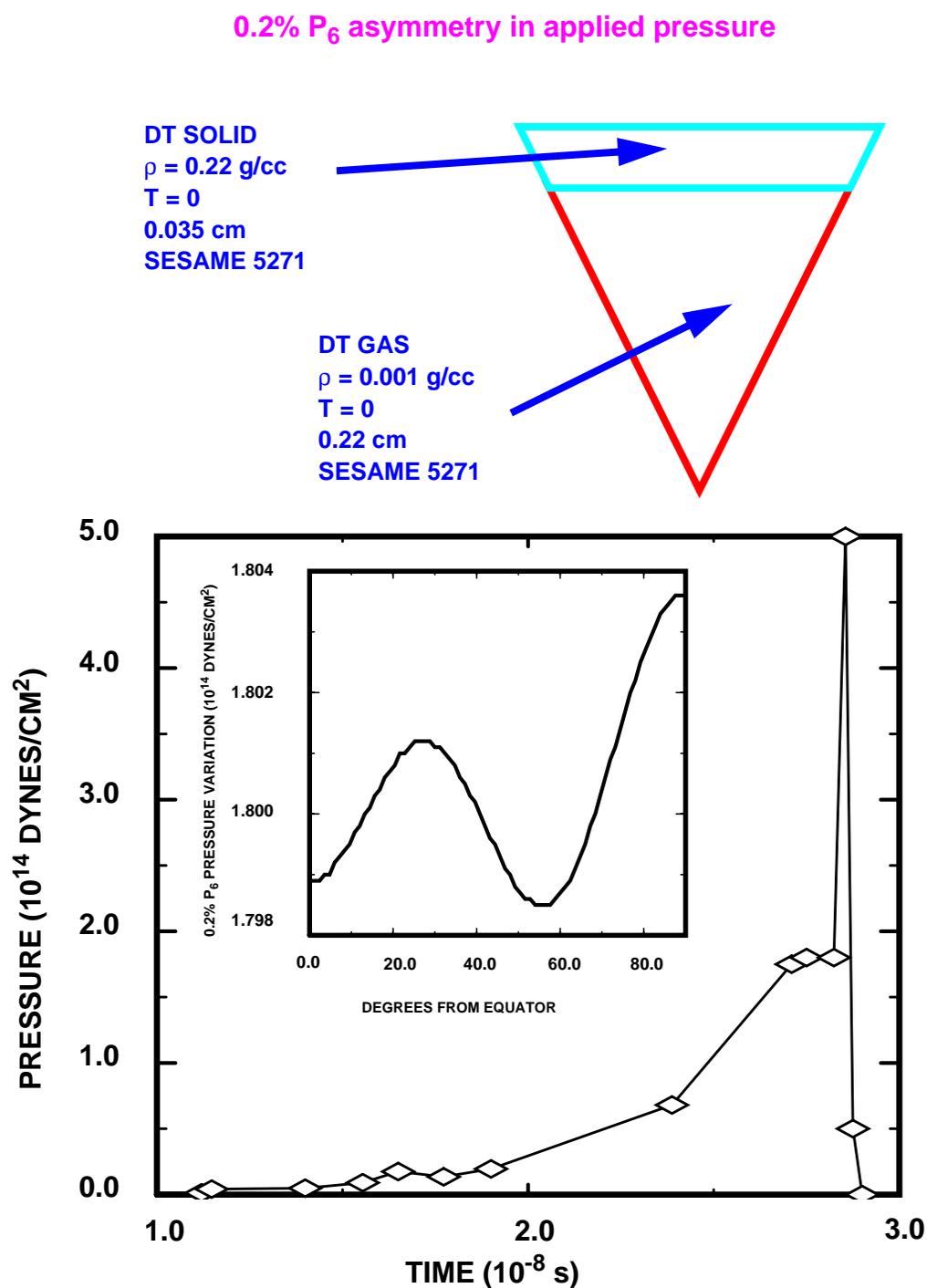


Figure 6. (a) Anisotropic drive initial target configuration. (b) Driving pressure history (with angular anisotropy inset).

Details of two RHALE calculations are presented in Figures 7 and 8. Figure 7 shows the grid of a pure Lagrangian RHALE calculation and a MMALE calculation near peak compression, at  $t=28.7$  ns. We have indicated the gas region in that plot. Figure 8 illustrates density contours for the two calculations at this time, spanning the density range from  $10.0$  to  $270.0 \text{ g/cm}^3$ . The RHALE implosion has almost stagnated at  $28.7$  ns because the gas temperature is so high that the thermal pressure resists the imploding outer layer. The RHALE gas region also maintains a strong memory

of the anisotropy in the boundary pressure, resulting in the observed distortion. Of course, we are not claiming that this is a realistic capsule implosion simulation, because very important physics are missing. However, it is a good hydrodynamics test when we compare it with an MMALE RHALE calculation.

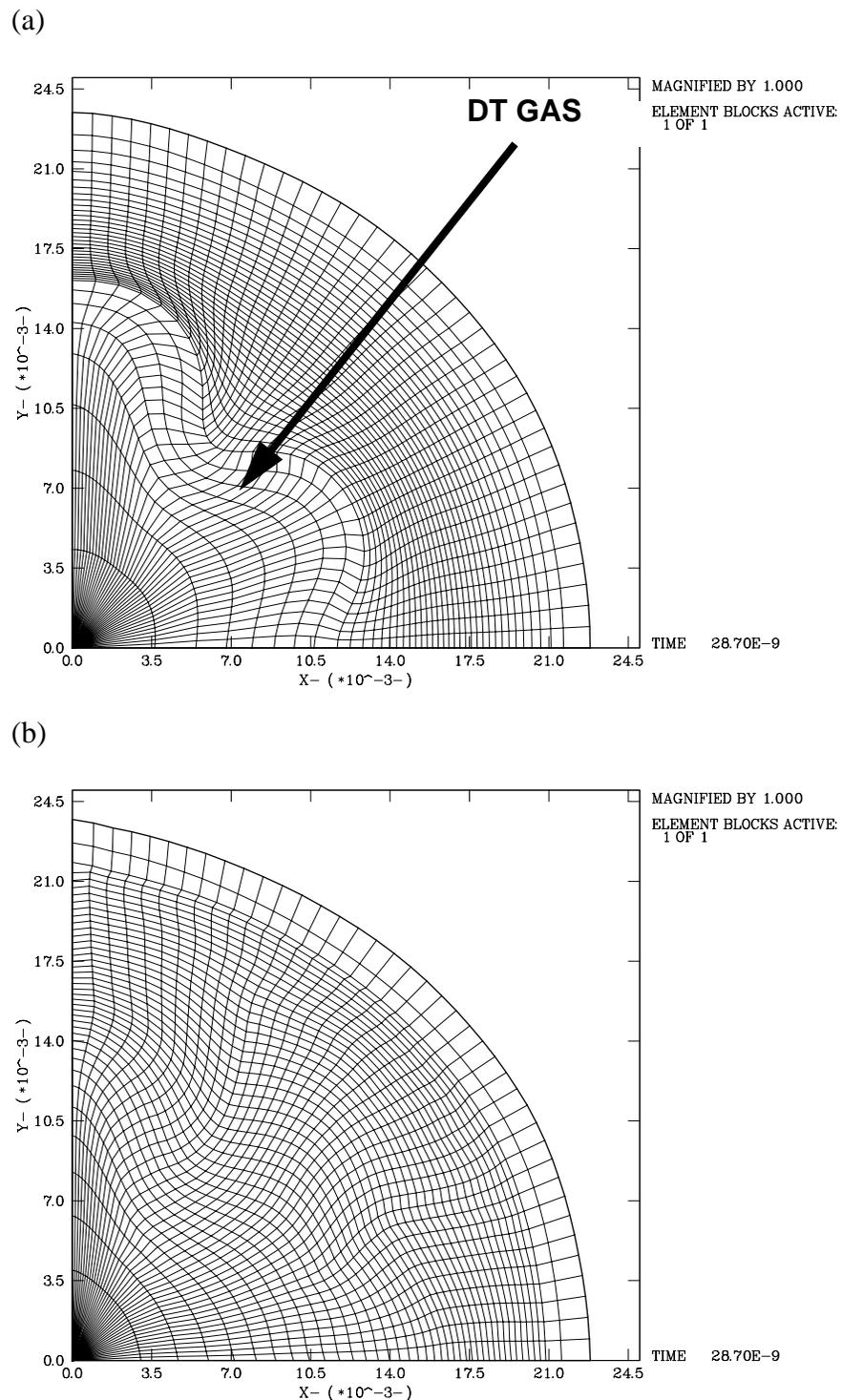


Figure 7. The mesh of the anisotropic implosion at 28.7 ns (near peak compression). (a) Lagrangian calculation; (b) MMALE calculation.

Figure 7(b) shows that the grid of a MMALE RHALE simulation which has geometrically adapted as triggered by the formation of severe zone aspect ratios observed in Figure 7(a). Density contours presented in Figure 8 show that there is relatively little density “diffusion” occurring from the MMALE algorithms. In fact, peak densities compare very well between the two calculations:  $243.5 \text{ g/cm}^3$  in the Lagrangian calculation as opposed to  $245.8 \text{ g/cm}^3$  in the MMALE calculation.

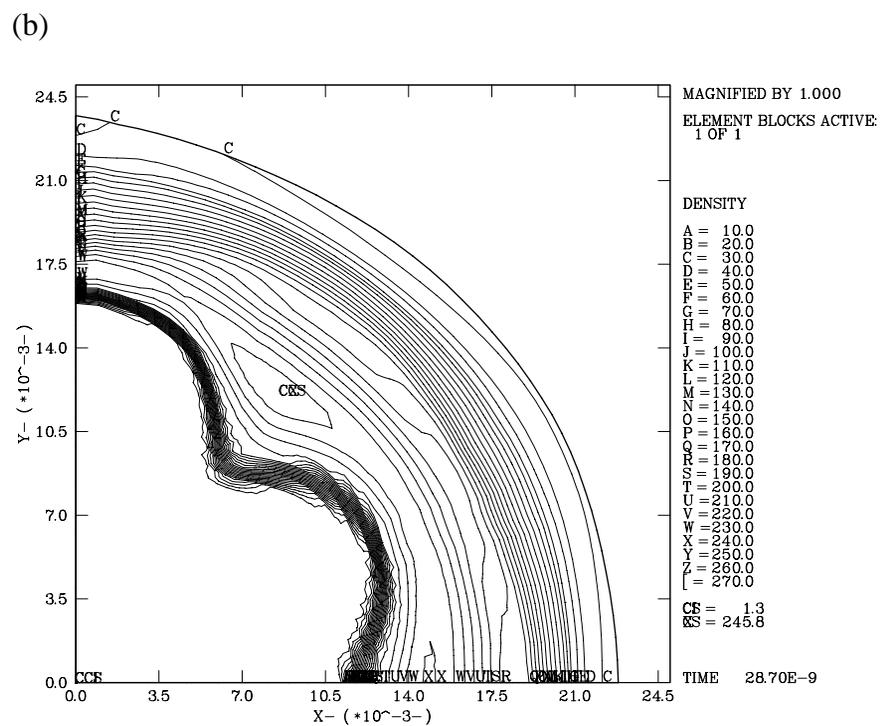
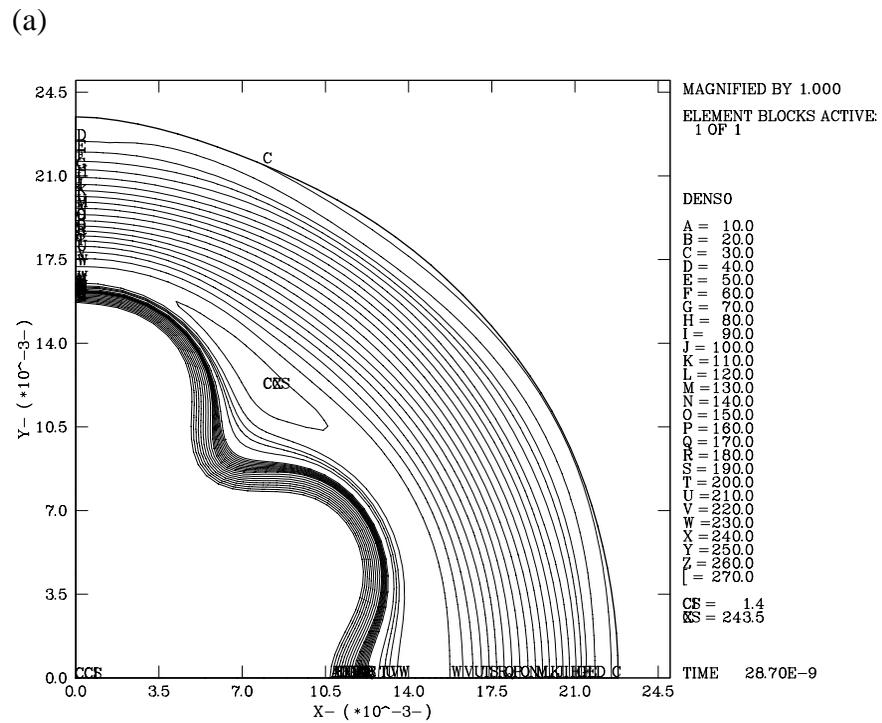


Figure 8. Density contours of the anisotropic implosion at 28.7 ns. (near peak compression). (a) Lagrangian calculation; (b) MMALE calculation.

The contours are a little noisier in Figure 8(b). This is because our mesh movement does not adapt to the density gradients, only to geometric zone behavior. While we have eliminated the “squeezed” zones that can be found in Figure 8(a), the algorithm has not moved as many regular zones into the region of sharp density gradient as we might wish. Adding adaptivity to RHALE would result in precisely this kind of zone movement as the calculation progresses. The zones, of course, would be expected to be multi-material zones. Pure Lagrangian mesh adjustment (Lagrangian rezoning) is insufficient to regularize a grid in which such severe jetting may occur.

## CONCLUSION

In this paper, we have described an ongoing code development project of interest to the mechanics, hypervelocity impact, and shock physics communities. We have given a brief description of the current capabilities of our code, RHALE, and have discussed our design philosophy. RHALE is constructed to implement some of the most modern numerical algorithms that can be applied for general ALE hydrodynamics in multiple dimensions. A series of calculations chosen to demonstrate the intended MMALE capabilities of RHALE and illuminate our code validation approach.

In the near term, we expect RHALE development to grapple with the issues of validation of 3-D MMALE hydrodynamics and adding additional capability for grid adaptivity especially on boundaries. In addition, we are beginning the task of porting RHALE to a massively parallel hardware architectures.

## REFERENCES

1. Amsden, A.A. and C.W. Hirt (1973). Los Alamos Scientific Report LA-5100. Los Alamos National Laboratory, Los Alamos, New Mexico.
2. Amsden, A.A., H.M. Ruppel, and C.W. Hirt (1980). Los Alamos Scientific Report LA-8095. Los Alamos National Laboratory, Los Alamos, New Mexico.
3. Bell, R. L and E. S. Hertel (1992). *An Improved Material Interface Reconstruction Algorithm for Eulerian Codes*. SAND92-1716. Sandia Corporation, Albuquerque, New Mexico
4. Benson, D.J. (1989). An efficient, accurate, simple ALE method for nonlinear finite element programs. *Comp. Meth. Appl. Mech. Eng.* 72, 309-350.
5. Benson, D.J. (1992) Momentum advection on a staggered mesh. *J. Comp. Phys.* 100, 143-162.
6. Budge, K.G. (1991). ALE shock calculations using a stabilized serendipity rezoning scheme. *1991 APS Topical Conference on Shock Compression of Condensed Matter*.
7. Christensen, R. (1991). Private communication.
8. Dukowicz, J.K. and B.J.A. Meltz (1992). Vorticity errors in multidimensional Lagrangian codes. *J. Comp. Phys.* 99, 115-134.
9. Forrestal, M., V. K. Luk, and N. S. Brar, (1990). Perforation of aluminum armor plates with conical-nose projectiles, *Mechanics of Materials*, 10.
10. Hughes, T.J.R., 1987. *The Finite Element Method*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey
11. McGlaun, J.M., S.L. Thompson, and M.G. Elrick (1989). *A Brief Description of the Three-Dimensional Shock Wave Physics Code CTH*. SAND89-0607. Sandia Corporation, Albuquerque, New Mexico
12. Margolin, L.G. and C.W. Beason (1988). Remapping on the staggered mesh. *Fifth Nuclear Code Developers' Conference*, Boulder, CO., October 11-14, 1988.
13. Margolin, L.G. and J.J. Pyun (1987). A method for treating hourglass patterns. *Proceedings of the Fifth International Conference on Numerical Methods in Laminar and Turbulent Flow*, Montreal, Canada.
14. Noh, W.F. and P. Woodward (1976). SLIC (simple line interface calculation), *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics*, Twente University, Enschede, June 28 - July 2, 1976.
15. Press, W.H., B.F. Flannery, S.A. Teukolsky, and W.T. Vetterling (1986). *Numerical Recipes*, Cambridge University Press, Cambridge, pp. 653-655.

16. Robinson, A.C. *et al.* (1992). Massively Parallel Computing, C++, and hydrocode algorithms. ASCE 8th Conference on Computing, June 10-12, 1992.
17. Sharp, R.W. and R.T. Barton (1981). *HEMP Advection Model*, UCID-17609. Lawrence Livermore National Laboratory, Livermore, California.
18. Silling, S. A. (1992). *Eulerian Simulation of the Perforation of Aluminum Plates by Nondeforming Projectiles*, SAND92-0493, Sandia Corporation, Albuquerque, New Mexico.
19. Taylor, L.M. and D.P. Flanagan (1987). *PRONTO-2D: A Two-Dimensional Transient Solid Dynamics Program*. SAND86-0594. Sandia Corporation, Albuquerque, New Mexico.
20. Tipton, R. (1992). Private communication.
21. Van Leer, B. (1984). *Comp. Meth. Appl. Phys. Eng.*, 6.
22. Winslow, A.M. and R.T. Barton (1982). *Rescaling of Equipotential Smoothing*. UCID-19846. Lawrence Livermore National Laboratory, Livermore, California.
23. Youngs, D.L. (1987). *An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code*, AWRE/44/92/35. Atomic Weapons Research Establishment

#### ACKNOWLEDGEMENTS

\*This work performed at Sandia National Laboratories supported by the U. S. Department of Energy under contract number DE-AC04-76DP0078