RAHYD: An ICF Target Simulation Code Written in C++ (U)

K.G. Budge, J.S. Peery, M.K.Wong, and T.G. Trucano
Sandia National Laboratories

J.E. Morel and M.L. Hall
Los Alamos National Laboratory

This paper describes RAHYD, a multi-material arbitrary Lagrangian-Eulerian (MMALE) ICF target simulation code. RAHYD is built on RHALE, Sandia's unclassified 3-D MMALE shock physics code, and incorporates the additional physics required for ICF calculations. The Lagrangian capabilities of RAHYD include arbitrary mesh connectivity, advanced artificial viscosity, and sophisticated equations of state. The ALE features of RAHYD provide the accuracy of a Lagrangian code while allowing a calculation to proceed under very large distortions. The radiation transport capabilities built into RAHYD use an SP_n transport approximation based on a second-order diffusion solver. Opacities are supplied by XSN or by a simple analytic model. We present a small set of sample calculations which illustrate the capabilities of RAHYD. (U)

Introduction

Designers of targets for inertial confinement fusion (ICF) experiments make extensive use of computer simulations in order to predict the behavior of the target. The computational kernel of ICF simulation codes must be highly efficient and capable of handling large material deformations, strong shocks, and sophisticated transport physics.

The equations of motion solved by a simulation code can be cast in either Eulerian or Lagrangian form. A pure Eulerian formulation (in which the computational grid is fixed in space) is unsuitable for ICF target simulation codes because the length scales of interest may decrease by more than an order of magnitude in the course of a calculation. Hence, ICF codes have generally been based on a pure Lagrangian formulation, in which the computational grid is embedded in and moves with the material. However, the Lagrangian formulation is unsatisfactory for modelling the turbulent flow that takes place near maximum compression. Thus, neither a pure Eulerian or a pure Lagrangian formulation of the equation of motion is entirely satisfactory for ICF simulations.

The RAHYD 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 large compressions and has less numerical dissipation than a purely Eulerian calculation (since less advection takes place). Quite large shearing 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 RAHYD 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 for reasons explained below.

RAHYD is based on the RHALE kernel (Peery, *et al*, 1992) and adds the additional physics and other special features required to accurately simulate ICF target implosions. In this paper, we describe the RAHYD code, with some emphasis on those features which are not part of the RHALE kernel.

RAHYD Mesh and Lagrangian Hydrodynamics

RHALE (hence RAHYD) supports three types of mesh movement: pure Lagrangian, pure Eulerian, and ALE. All meshes are based on a bilinear 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 RAHYD 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 mass matrix is diagonalized by lumping (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. 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 keystoning. There are currently three artificial viscosity and four hourglass control options available to users. 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 RAHYD. Since RAHYD uses the same uniform-strain quad or hex element that is used by RHALE, 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 the Margolin-Pun hourglass control method.

RAHYD can handle 2-D Cartesian, 2-D cylindrical, and 3-D Cartesian geometries. We have experimented with both a volume-weighted and an area-weighted cylindrical element (Taylor and Flanagan, 1987). The former has the advantage that the nodal lumped masses are not time dependent and the energy and momentum are exactly conserved; however, the element fails to maintain spherical symmetry. The area-weighted formulation has the advantages of a closer correspondence between the 2-D Cartesian and axisymmetric formulations; in addition, symmetry is preserved. Since RAHYD must maintain symmetry for isotropic implosions, we are currently using an area-weighted axisymmetric element.

Material Models and Material Data

There are currently four material model libraries implemented in RHALE, of which the following are important for RAHYD:

- **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, and the SNL-SESAME tabular equation of state.

- **The Conductivity library** computes the thermal conductivity of a material based on a Lee-More conductivity model.

RAHYD also includes the following material library that is not included in RHALE:

- **The Opacity library** computes the opacities of a material for radiation transport modelling using the XSN opacity package. In the future, a SESAME tabular opacity option may also be included.

MMALE ALGORITHMS IN RAHYD

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. Two element-based algorithms are key to RHALE's MMALE hydrodynamics: the half interval shift algorithm, HIS, (Benson, 1992) and a variational

approach for equipotential schemes (Tipton, 1992). RHALE contains a version of the Sandia Modified Youngs' Reconstruction Algorithm, SMYRA (Youngs, 1987; Bell and Hertel, 1992), a high resolution interface tracker for determining material fluxes. All of these algorithms carry over to RAHYD automatically.

TRANSPORT PHYSICS

RAHYD includes algorithms for energy transport by thermal conduction. RAHYD adds radiative transport using an SP_n algorithm based on the same diffusion solver used for thermal conduction (the AUGUSTUS package).

Diffusion Solver

Both thermal conduction and radiative transport simulations require a fast and accurate diffusion solver. We have experimented with three different solvers, which are described below.

Explicit Solver. The first diffusion solver installed in RAHYD was an explicit, low-order method based on the same gradient operator calculation used for the explicit hydrodynamics. This approach permitted rapid prototyping of the conduction code and gave good answers for problems in which the hydrodynamic time scale is shorter than the thermal time scale. Unfortunately, thermal time scales in ICF calculations can be much shorter than hydrodynamic time scales, especially when radiative transport is modelled.

Bilinear Discontinuous Solver. We then installed a bilinear discontinuous solver in RAHYD. This solver is based on a novel finite element technology that showed promise of being highly accurate, easily preconditioned, and of treating material interfaces very accurately. However, numerical experiments showed that this solver fails to preserve spherical symmetry, which makes it unsuitable for ICF calculations.

Cell-Centered Solver. Thermal conduction is now implemented in RAHYD using AUGUSTUS, an implicit cell-centered diffusion solver package (Morel *et al.*, 1992). The AUGUSTUS algorithm correctly handles material interfaces and is second-order accurate so long as elements are not highly distorted.

The diffusion matrix generated by AUGUSTUS is solved using JTPACK, an iterative solver package based on the GMRES algorithm (Turner 1993). The matrix is preconditioned using a simplified diffusion stencil that is symmetric and positive-definite, and convergence is rapid even for nearly steady-state problems.

Radiative Transport

RAHYD adds radiative transport to the RHALE kernel by basing an SP_n transport formulation on the AUGUSTUS diffusion solver. The SP_n approximation assumes that transport is locally one-dimensional; this greatly reduces the number of angular groups, but still captures most of the transport physics that is lost by the diffusion approximation (Larsen *et al.*, 1993). SP5 or SP3 will typically be selected for simulations; SP1 is identical to P1, which is equivalent to the diffusion approximation for steady-state problems.

The set of SP_n equations are solved iteratively, using diffusion-synthetic acceleration (DSA) to accelerate convergence of the Thompson scattering source term and linear multifrequency-grey acceleration (LMFG) to accelerate convergence of the thermal emission source term. This approach permits the equation for each energy and angle group to be solved individually using AUGUSTUS.

OBJECT-ORIENTED PROGRAMMING

Perhaps the most revolutionary feature of the RHALE kernel that we base RAHYD on is that it is written in the C++ programming language using the object-oriented programming paradigm. Our decision to use C++ was based on the following considerations:

- C++ supports features, such as strong type checking, that eliminate entire classes of bugs found in FORTRAN codes.
- C++ provides dynamic memory management as an integral part of the language.
- C++ supports nested types (*classes*) that allow a clean implementation of data structures (*encapsulation*).
- C++ supports operator and function overloading (*data abstraction*).
- C++ supports inheritance and polymorphism (*object-oriented programming*).

Some of these features are provided by FORTRAN-90. However, at the time the RHALE project began, compilers for FORTRAN-90 were virtually nonexistent. High-quality compilers and software tools remain much more widely available for C++ than FORTRAN-90 even today. FORTRAN-90 cannot realistically support object-oriented programming, which is an important part of our approach to software design.

We will not attempt to describe the object-oriented programming paradigm here. An excellent discussion is found in Booch (1990), and particulars of applying this paradigm to numerical programming are found in Wong *et al.* (1993). In our experience, the main benefit of object-oriented programming is that it permits rapid

prototyping of code by enhancing reusability and reducing complexity.

APPLICATIONS

In order to demonstrate the unique capabilities of the RAHYD code, we briefly describe a calculation of interest to the ICF community.

Non-uniform 2-D Implosion

This calculation is an example of the non-uniform implosion of a two layer capsule that is similar to an inertial confinement fusion target capsule. The pressure boundary condition, extracted from an ion beam deposition study performed by George Allshouse (private communication) has a 0.2% P_6 (Legendre polynomial) pole to equator asymmetry and varies with time as shown in Figure 1.

Details of two RAHYD calculations are presented in Figures 2 and 3. Figure 2 shows the grid of a pure Lagrangian RAHYD calculation and a MMALE calculation near peak compression, at t=28.7 ns. We have indicated the gas region in that plot. Figure 3 illustrates density contours for the two calculations at this time, spanning the density range from 10.0 to 270.0 g/cm^3. The RAHYD 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 RAHYD gas region also maintains a strong memory of the asymmetry of 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 energy transport (in particular, thermal conduction) physics is not used. However, it is a good hydrodynamics test when we compare it with an MMALE RAHYD calculation.

Figure 2(b) shows that the grid of a MMALE RAHYD simulation which has geometrically adapted as triggered by the formation of severe zone aspect ratios observed in Figure 2(a). Density contours presented in Figure 3 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 g/cm^3 in the Lagrangian calculation as opposed to 245.8 g/cm^3 in the MMALE calculation.

The contours are a little noisier in Figure 3(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 3(a). the algorithm has not moved as many regular zones into the region of sharp density gradient as we might wish. Adding adaptivity to RAHYD 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 severe jetting may occur, as it does in the test problem at a time slightly greater than the time in Figure 2 and Figure 3.

Conclusion

In this paper, we have described an ongoing code development project of interest to the ICF target design community. We have given a brief description of the current capabilities of our code, RAHYD, and have discussed our design philosophy. RAHYD is constructed to implement some of the most modern numerical algorithms that can be applied for general ALE hydrodynamics in multiple dimensions. We illustrated the intended MMALE capabilities of RAHYD and illuminate our code validation approach with a calculation capturing the necessary implosion behavior for light-ion driven ICF high-gain targets.

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

References

Bell, R. L and Hertel, E.S., *An Improved Material Interface Reconstruction Algorithm for Eulerian Codes,* Sandia National Laboratories, Albuquerque, NM, SAND92-1716 (1992).

Benson, D.J., "Momentum Advection on a Staggered Mesh," *J. Comp. Phys.* **100**, 143-162 (1992).

Booch, G., *Object-Oriented Design with Applications*, (Benjamin/Cummings, Redwood City, CA), 1991.

Dukowicz, J.K. and Meltz, B.J.A., "Vorticity Errors in Multidimensional Lagrangian Codes," *J. Comp. Phys.* **99**, 115-134 (1992).

Hughes, T.J.R., *The Finite Element Method*, (Prentice-Hall, Englewood Cliffs, NJ, 1987).

Larsen, E.W., McGhee, J.M., and Morel, J.E. "Asymptotic Derivation of the Simplified P_n Equations," *Proceedings of the Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications,* 19-23 April 1993, Kongresszentrum, Karlsruhe, Germany, Vol. 1, p. 718.

Margolin, L.G. and Pyun, J.J., "A Method for Treating Hourglass Patterns," *Proceedings of the Fifth International Conference on Numerical Methods in Laminar and Turbulent Flow,* Montreal, Canada, 1987.

Morel, J.E., Dendy, J.E., Hall, M.L., and White, S.W., "A Cell-Centered Lagrangian-Mesh Diffusion Differencing Scheme," *J. Comp. Phys.* **103**, 286-299 (1992).

Peery, J. S., K. G. Budge and M. K. Wong, "RHALE: A 3-D MMALE Shock Physics Code," in Proceedings of the 1992 NECDC, Sunnyvale, California (1992).

Taylor, L.M. and Flanagan, D.P., *PRONTO-2D: A Two-Dimensional Transient Solid Dynamics Program*, Sandia National Laboratories, Albuquerque, NM, SAND86-0594 (1987).

Tipton, R., Lawrence Livermore National Laboratory, Livermore, CA, private communication (1992).

Turner, J., Los Alamos National Laboratory, Los Alamos, NM, private communication (1994).

Wong, M.K., Budge, K.G., Peery, J.S., and Robinson, A.C. "Object-Oriented Numerics: A Paradigm for Numerical Object-Oriented Programming." *Computers in Physics* **7**, 6 (1993)

Youngs, D.L., *An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code*, Atomic Weapons Research Establishment, AWRE/44/92/35 (1987).

Acknowledgments

This work performed at Sandia National Laboratories supported by the U. S. Department of Energy under contract number DE-AC04-94AL85000.

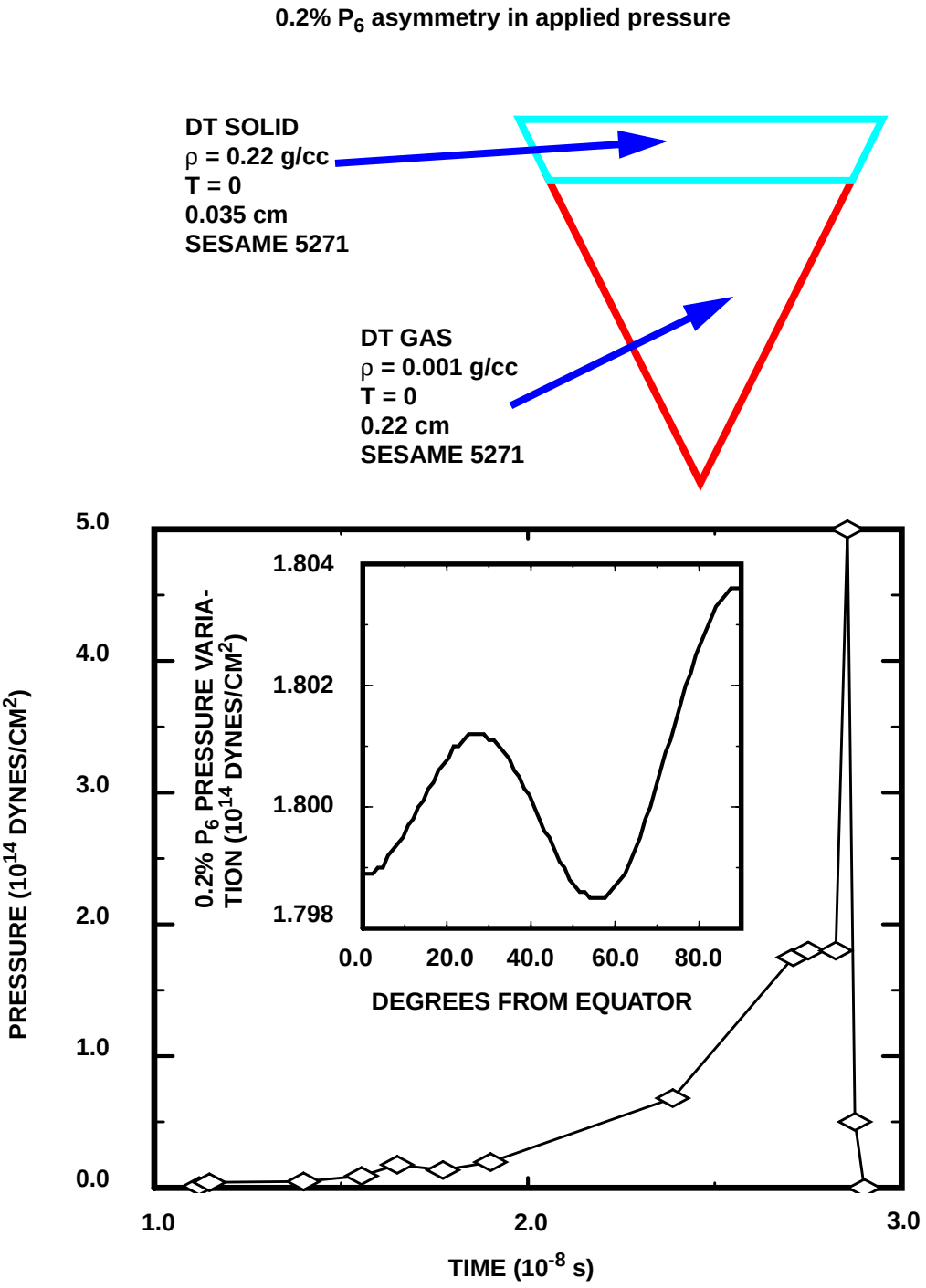

Figure 1.(a) Non-uniform drive initial target configuration. (b) Driving pressure history (with angular asymmetry inset).

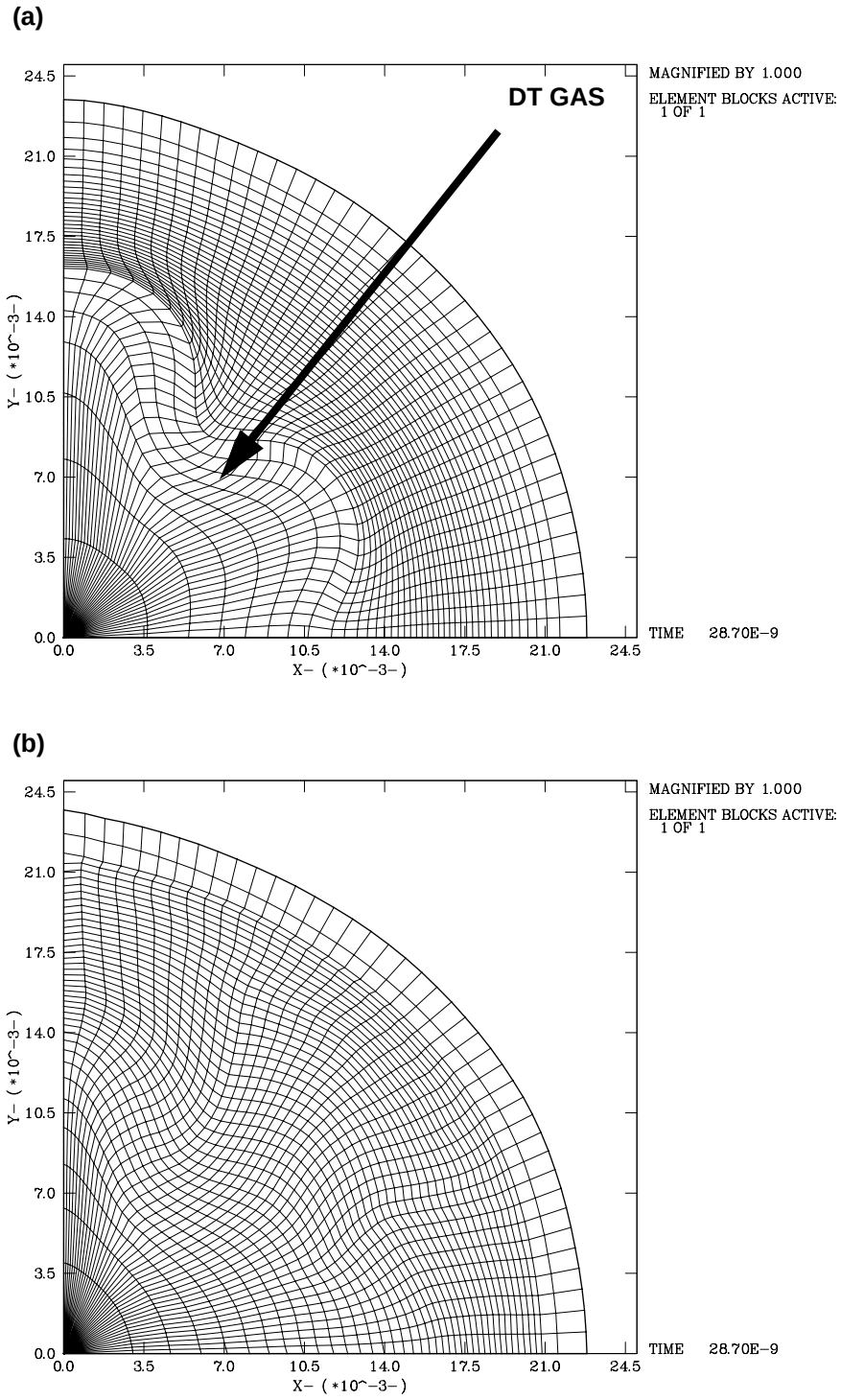

Figure 2.The mesh of the non-uniform implosion at 28.7 ns (near peak compression). (a) Lagrangian calculation; (b) MMALE calculation.

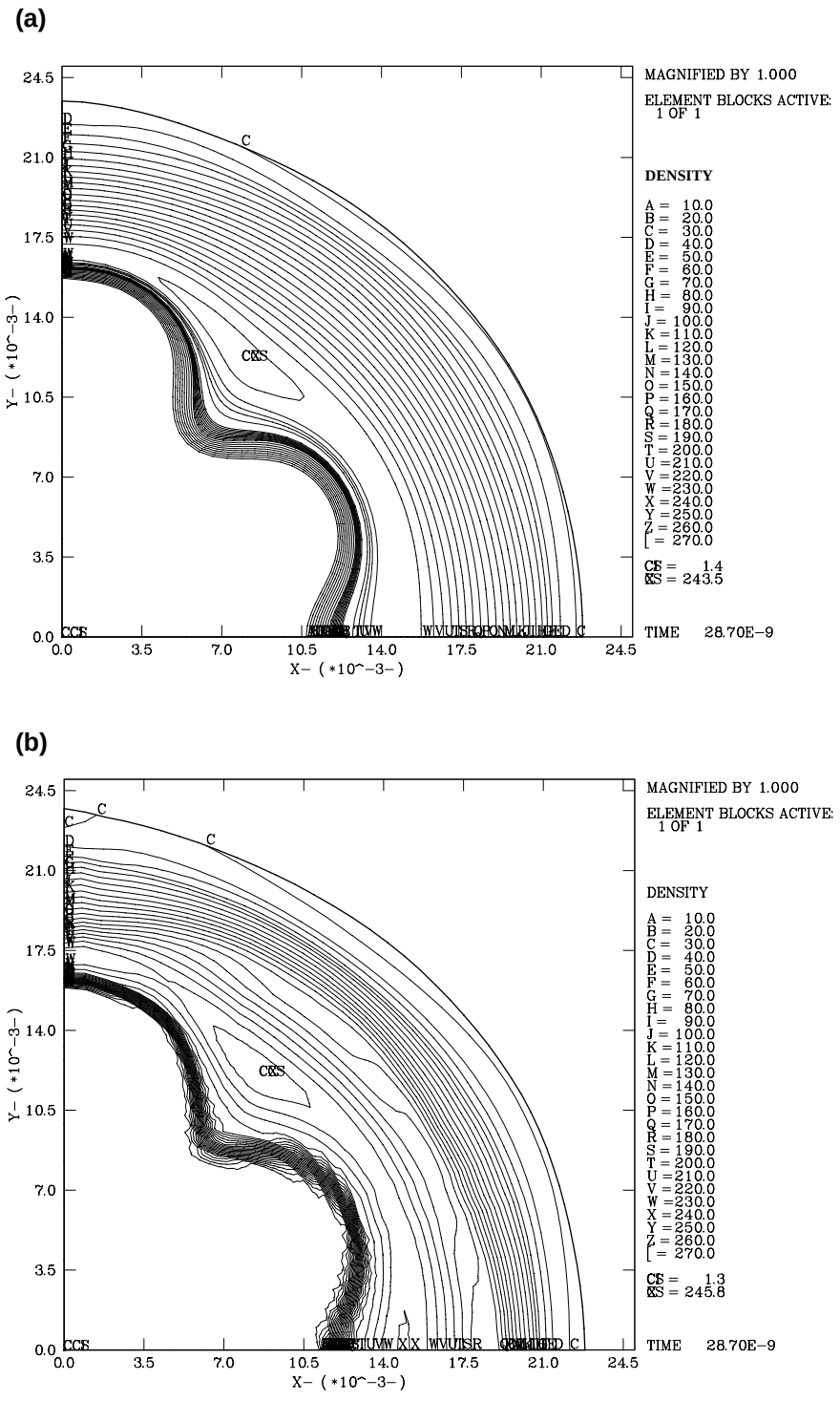

Figure 3.Density contours of the non-uniform implosion at 28.7 ns. (near peak compression). (a) Lagrangian calculation; (b) MMALE calculation.