



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# CHASM Challenge Problem: Lagrangian Hydrodynamics

R. Hornung, J. Keasler, M. Gokhale

September 14, 2010

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

# **CHASM Challenge Problem: Lagrangian Hydrodynamics\***

Rich Hornung, Jeff Keasler, Maya Gokhale  
Lawrence Livermore National Laboratory, Livermore, CA

## **Motivation**

Computer simulations of many science and engineering problems require modeling the equations of hydrodynamics which describe the motion of materials relative to each other induced by various forces. Many important DoD simulation problems involve complex multi-material systems that undergo large deformations. Examples include the analysis of armor defense, penetration mechanics, blast effects, structural integrity, and conventional munitions such as shaped charges and explosively formed projectiles. Indeed, the original motivation for developing codes that solve the equations of hydrodynamics, herein referred to as “hydrocodes”, was to solve problems with defense applications.

The FY2010 Requirements Analysis Report issued by the DoD High Performance Computing Modernization Program (HPCMP) Office shows that a major portion of DoD HPC activities involves hydrocodes [HPCMP2010]. The report surveyed 496 projects across the Services and various Agencies, representing 4,050 HPCMP users at more than 125 locations, including government, contractors, and academia, and grouped each project into one of ten categories. The Computational Fluid Dynamics (CFD) category accounted for the most projects (37% of the total) and the most users (27% of the total). The Computational Structural Mechanics (CSM) category was fourth with about 10% of total users. According to the report, hydrocodes, specifically, are among the most used of all applications with several ranked in the top ten in terms of number of users. In addition, of all non-real time applications, four DOE hydrocodes (CTH, ALE3D, Sierra, and Alegra) are ranked in the top ten in terms of CPU hours.

Numerical algorithms found in hydrocodes present unique computational issues not found in our other challenge problems. A typical hydrocode approximates the hydrodynamics governing equations using a discrete representation in which the spatial problem domain is partitioned into a collection of volumetric elements defined by a mesh. Difference equations that approximate differential operators in the equations couple data quantities on the mesh (e.g, at nodes and elements) via “stencil” operations. Other computations, involving material properties and equation of state, are interleaved with the stencil operations. The operations must be performed in a specific order for numerical and physical accuracy. Due to these computational characteristics, plus the importance of hydrocodes to DoD HPC efforts, a hydrodynamics challenge problem is important inclusion in a code suite used to evaluate UHPC architectures.

---

\* This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL report LLNL-SR-454998.

## Hydrodynamics Methods Overview

There are two alternative specifications of the hydrodynamics governing equations. These formulations can be found in any book on continuum mechanics; e.g., see [Batchelor2000]. In the *Eulerian* frame of reference, physical quantities, such as density, pressure, and velocity, are defined as functions of spatial position and time. Thus, the Eulerian form of the equations can be thought of as describing the spatial distribution of the flow quantities at each instant in time. The *Lagrangian* formulation exploits the fact that some physical quantities refer to identifiable pieces of matter in addition to their positions in space, a view similar to particle mechanics. In the Lagrangian formulation, the flow quantities are defined as functions of time and particular material elements and describe the dynamical history of those elements. Both formulations are used in hydrocodes and have advantages and disadvantages for various applications.

Eulerian hydrocodes, such as CTH, typically employ an orthogonal mesh for accuracy of the numerical approximation. The mesh is fixed in time and space and the materials flow through the mesh as a simulation progresses. Such codes are particularly useful for simulation problems that exhibit strong shearing and vortical motion such as turbulent flows. However, moving material boundaries and interactions among multiple materials are less natural to express in Eulerian methods. For example, each material in a mesh element may be represented as a fraction of element volume. Thus, material interfaces tend to diffuse when not aligned with the mesh. Without additional numerical machinery to resolve material interfaces, Eulerian methods require very fine mesh resolution for good spatial accuracy. See the left-hand image in Figure 1.

In Lagrangian hydrocodes, the initial mesh configuration partitions the problem domain into material elements and element boundaries are constructed to align with material interfaces. As a simulation evolves, the mesh follows the motion of these elements through space and time. Lagrangian methods handle moving boundaries and multiple materials naturally and can provide a highly accurate solution without an excessively refined mesh for many problems. See the right-hand image in Figure 1. However, when the flow involves sufficiently complex structure, Lagrangian methods can perform poorly as mesh elements distort and possibly tangle.

ALE (Arbitrary Lagrangian Eulerian) codes (such as ALE3D [ALE3DWeb]) have been developed to seek a compromise between the Eulerian and Lagrangian formulations. ALE methods can accurately solve problems involving moving boundaries, multiple materials, and strong shearing and vortical flow regions. The general strategy is to evolve the problem using the Lagrangian approach until the mesh reaches a level of distortion such that continuing in this fashion is problematic. At this point, the mesh is “relaxed” to a more desirable configuration and the physical quantities are mapped to the new mesh and the simulation continues.

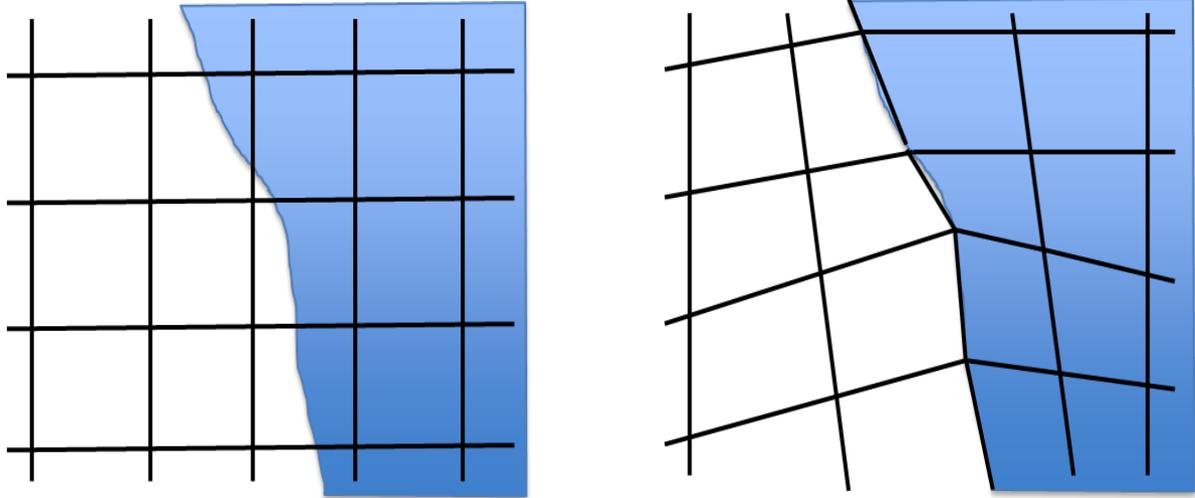


Figure 1: Illustration of how the boundary between two materials (white and blue) may be represented on a mesh. The left image shows a fixed Eulerian mesh and a material interface that does not align with mesh element boundaries. The position of the material interface may be approximated using material volume fractions in the elements. The right image shows how Lagrangian mesh nodes follow the motion of the material interface, thus representing it much more accurately.

## Lagrangian Hydrodynamics Challenge Problem

Regardless of algorithm methodology, operations performed in most hydrocodes are similar in terms their computational characteristics and data access patterns. In the interest of algorithm simplicity and smaller code size, we have chosen Lagrangian hydrodynamics for our challenge problem.

### Governing Equations

The equations of hydrodynamics represent the conservation of mass, momentum, and energy [Batchelor2000]. In the Lagrangian description, the differential equations are:

$$\begin{aligned} \frac{D\rho}{Dt} &= -\rho \vec{\nabla} \cdot \vec{U} \\ \rho \frac{D\vec{U}}{Dt} &= \vec{\nabla} \cdot \Sigma \\ \frac{De}{Dt} &= \frac{1}{\rho} \text{Tr}(\epsilon_{tot} \cdot \Sigma) = -p \frac{DV}{Dt} + V \text{Tr}(\epsilon \cdot S) \end{aligned}$$

Here, the Lagrange time derivative (or material derivative) is a total derivative moving with the flow field:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{U} \cdot \vec{\nabla}$$

The variable quantities on the left hand-side are density  $\rho$ , the velocity vector  $\vec{U}$ , internal energy  $e$ . The total strain rate tensor is:

$$\boldsymbol{\varepsilon}_{tot} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$$

where  $\vec{X} = (x_1, x_2, x_3)$  is the position vector in three-dimensional space. The total stress tensor is

$$\boldsymbol{\Sigma}_{ij} = S_{ij} - p\delta_{ij}$$

where  $p$  is the isotropic pressure

$$p = -\frac{1}{3} Tr(\boldsymbol{\Sigma})$$

and the tensor  $S$  contains the stress deviators  $S_{ij} = \Sigma_{ij} + p\delta_{ij}$ . The quantity  $V$  is the specific volume:  $V = \frac{1}{\rho}$ . The tensor  $\boldsymbol{\varepsilon}$  is the deviatoric strain rate tensor:

$$\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_{tot}) - \frac{1}{3} \vec{\nabla} \cdot \vec{U}$$

The pressure is usually determined by an equation of state that gives the pressure as a function of density and internal energy:  $p = EOS(\rho, e)$ . The stress deviator terms are usually determined by some constitutive relations.

In the interest of simplicity for our challenge problem, we will consider the Euler equations which describe a single material and assume an inviscid approximation of the stress tensor; i.e., no shearing stresses  $S_{ij} = 0$ . The resulting equations are:

$$\begin{aligned} \frac{D\rho}{Dt} &= -\rho \vec{\nabla} \cdot \vec{U} \\ \rho \frac{D\vec{U}}{Dt} &= -\vec{\nabla} \cdot p \\ \frac{De}{Dt} &= -\frac{p}{\rho} \vec{\nabla} \cdot \vec{U} = -p \frac{DV}{Dt} \end{aligned}$$

## Mesh Quantities

These equations will be solved on a “staggered” spatial mesh [Wilkins1964], where each mesh element is a (potentially distorted) hexahedron. Specifically, thermodynamic quantities  $\rho$ ,  $e$ ,  $p$  are approximated as piece-wise constants within each element with their value represented at the element center. This is known as *single-point quadrature*. Kinematic variables  $\vec{X}$ ,  $\vec{U}$  are defined at the element nodes. The spatial relationships among these quantities are illustrated in Figure 2. Spatial gradients are computed using finite element formulations that will be described later.

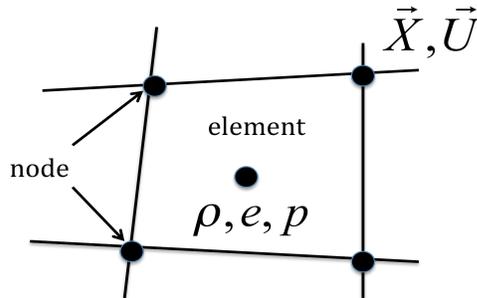


Figure 2: Staggered mesh representation for flow quantities. Figure shows a two-dimensional mesh for simplicity. Representation on a three-dimensional mesh is the obvious extension.

## Numerical Time Integration

After establishing the initial state of the solution variables on the mesh and appropriate boundary conditions, the equations described above are integrated in time to advance the solution. We use an explicit time stepping algorithm which advances the solution through a discrete sequence of time increments; i.e., the solution at time  $t^n$  is advanced to time  $t^{n+1} = t^n + \Delta t^{n+1}$ , where  $n$  is the step number and  $\Delta t^{n+1} = t^{n+1} - t^n$  is the time increment. The Courant-Friedrichs-Lewy (CFL) condition determines the maximum size of the time increment and is based on the shortest distance across any mesh element and the sound speed of the material in the element [CFL1967].

To numerically model the entropy-conserving properties of the governing equations, they must be augmented with a dissipation mechanism. In reality, physical viscosity has a dissipation length scale of a few molecular mean free paths. Such a small length scale is inappropriate for the length scale of mesh elements. Nevertheless, using the exemplar of real viscosity, von Neumann and Richtmyer originally developed the concept of an artificial viscosity and their approach has been extended to many useful variations over the years [Reference for Q model in code].

Single-point quadrature elements (whereby element quantities are defined by a single value as described earlier) possess spurious singular modes, or “hourglass” modes, which can result in physically unrealistic response. To remedy this, we employ the Flanagan-Belytschko kinematic hourglass filter [FlanBely1981].

The use of an artificial viscosity and an hourglass filter imply that we solve modified forms of the momentum and energy equations. For the Euler equations described above, these equations become:

$$\rho \frac{D\vec{U}}{Dt} = -\vec{\nabla} \cdot (p + q) + \rho a_{HG}$$

$$\frac{De}{Dt} = -(p + q) \frac{DV}{Dt} + \dot{e}_{HG}$$

Here,  $q$  (which acts like a pressure) is the artificial viscosity term and  $a_{HG}$  and  $\dot{e}_{HG}$  are the acceleration and heating due to the due to the hourglass filter.

### Summary of the Lagrange Time Step

Given the solution quantities at time  $t^n$  ( $\rho^n, p^n, e^n, \vec{X}^n, \vec{U}^n$ ) a time increment  $\Delta t^{n+1}$ , the procedure for advancing the solution to time  $t^{n+1}$  ( $\rho^{n+1}, p^{n+1}, e^{n+1}, \vec{X}^{n+1}, \vec{U}^{n+1}$ ) consists of the following steps.

1. Construct the force at each mesh node.

Integrate  $-(p + q)$  over a control volume at each node.

Calculate the hourglass filter contribution  $a_{HG}$  at each node.

Sum these to get the force  $\vec{F}$  at each node.

2. Compute the acceleration at each mesh node, including boundary conditions.

The acceleration at each node is the force computed in step 1 divided by the nodal mass,  $\vec{a} = \vec{F} / m$ . Where we have a symmetry boundary condition, we set the appropriate component of the acceleration vector to zero.

3. Compute the velocity at each node.

The new velocity is computed by integrating the acceleration:  $\frac{D\vec{U}}{Dt} = \vec{a}$ .

Then, apply cutoffs to velocity components that are close to zero.

4. Update the position of each mesh node.

The new position of a node is computed by integrating the velocity:  $\frac{D\vec{X}}{Dt} = \vec{U}$ .

5. Calculate various element quantities based on new node positions. These include element volume, characteristic length (function of element volume and area of element faces), velocity gradient, etc.
6. Calculate artificial viscosity  $q$  in each element. This includes linear and quadratic terms. Artificial viscosity is set to zero if element is expanding.
7. Evaluate material model properties based on material in each element. This includes equation of state evaluation, pressure  $p$  update, and internal energy  $e$  update.
8. Calculate next time increment and return to step 1.

## Test Problems

The Lagrange algorithm described above can be applied to a wide range of hydrodynamics simulation problems. To simplify the challenge problem code, we will limit its application to a few well-know test problems that can be used to verify correctness of the implementation and analyze execution performance. Specifically, the problem setup phase will be hardcoded for these test problems. Such test problems will include the Noh infinite strength and Sedov blast wave problems [NohWeb, Sedov1959]. Example calculations of these problems are shown in Figures 3 and 4. Both problems have known analytic solutions and can be scaled to arbitrarily large problem sizes with mesh resolution being the primary scaling factor. We expect that a UHPC system should be able to run a calculation containing up to one billion mesh elements and potentially more than that, depending on available memory.

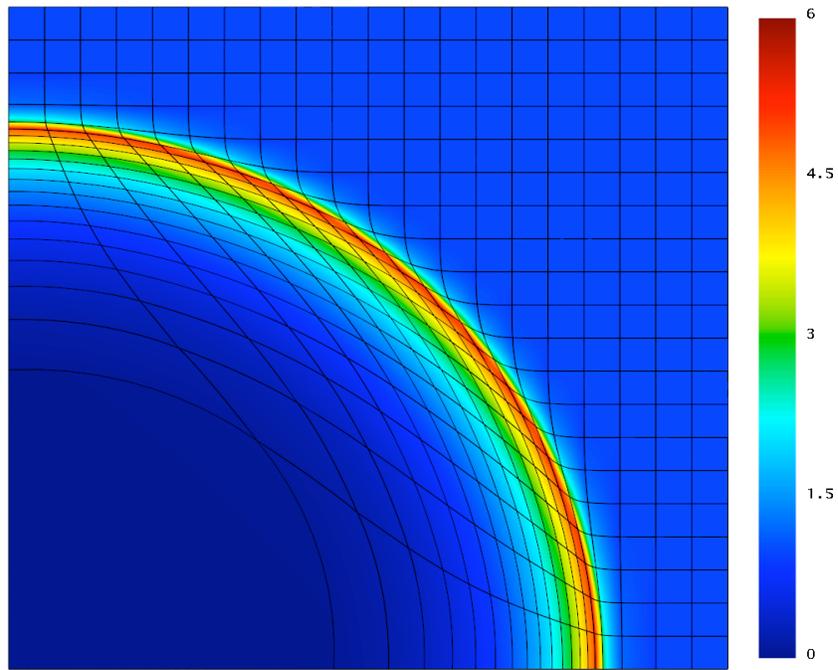


Figure 3: The Sedov blast wave problem models an expanding shock front originating from a point blast.

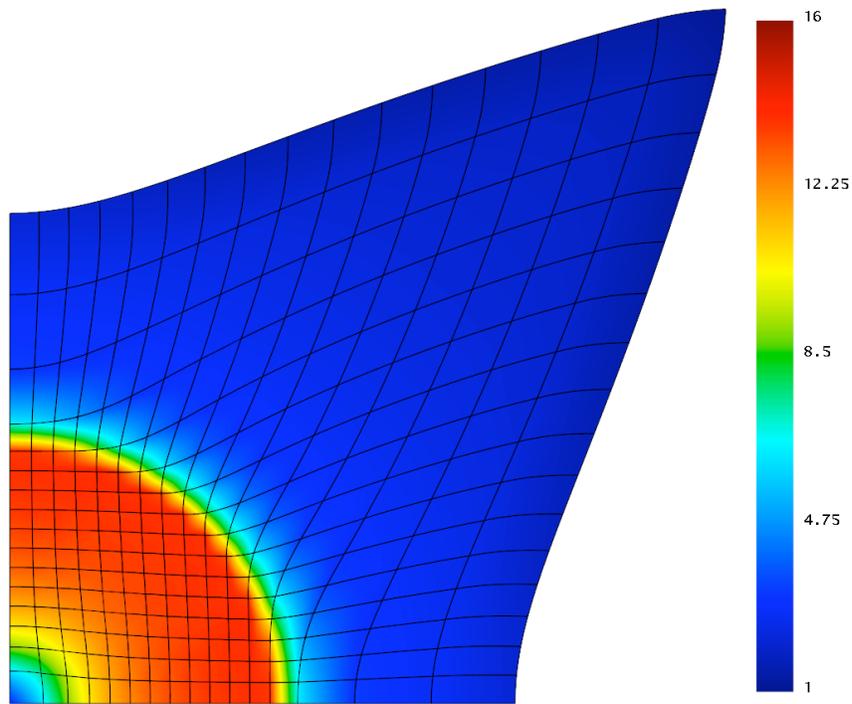


Figure 4: The Noh problem involves a spherically-symmetric shock directed toward the origin and then reflected back with "infinite" strength.

## Benchmarks and Metrics

Several of the operations found in the Lagrange time step will be broken out into smaller benchmark kernels for performance analysis on various architectures. These may include the element volume calculation, artificial viscosity calculation, and computation of the hourglass filter.

We will propose various metrics to evaluate performance along with values of these metrics that we observe on current systems for similar calculations. These may include:

- Grind Time: A common metric used to evaluate performance of finite element codes is “grind time”, which is the time required to update the solution variables in an element through one time increment, typically in microseconds. Current production codes typically execute the Sedov problem on x86-64/Linux systems with a grind between 2 and 3 microseconds on one processor.
- Memory Bandwidth: Our implementation will utilize an unstructured mesh representation which is often employed in production hydro codes for flexibility in defining complex geometries. The use of mesh connectivity arrays, such as those that define nodes associated with each mesh element, results in indirection that can stress system memory bandwidth.
- Scalability and Parallelism: Our implementation will be designed to treat each mesh element as the smallest unit of work. This will allow a very large amount of fine-grained parallelism to be exercised by UHPC systems.
- Programmability: Our challenge problem reference implementation will be derived from a production hydrocode containing hundreds of thousands of lines of source code. However, our reference implementation will consist of only a few thousand lines. This will allow exploration of implementation alternatives suited to novel UHPC architectures, potentially involving significant code rewriting.

## References

- [HPCMP2010] DoD High Performance Computing Modernization Program Office, *Department of Defense High Performance Computing Modernization Program FY 2010 Requirements Analysis Report* (2010).
- [Batchelor2000] G. K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press: New York (2000).
- [ALE3DWeb] ALE3D web page, <https://wci.llnl.gov/codes/ale3d/>.
- [Wilkins1964] M. L. Wilkins, *Methods in Computational Physics*. Academic Press: New York (1964).
- [CFL1967] R. Courant, K. Friedrichs, H. Lewy, “On the Partial Difference Equations of Mathematical Physics”, *IBM Journal*, March 1967, pp. 215-234. English translation of the German original published in *Mathematische Annalen* **100**(1):32-74, 1928.
- [FlanBely1981] D. P. Flanagan, T. Belytschko, “A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control”, *International Journal for Numerical Methods in Engineering* **17**:679-706, 1981.
- [NohWeb] <http://www.mso.anu.edu.au/fyris/3dnoh.html>.
- [Sedov1959] L. I. Sedov, *Similarity and Dimensional Methods in Mechanics*. Academic Press: New York (1959).