Modeling ICF Spherical Implosion Instabilities in 3D with Exact Energy Conservation

Milad Fatenejad, Gregory Moses

Fusion Technology Institute, University of Wisconsin – Madison

Summary

Abstract

We will present the results of 3D instability simulations performed on spherically convergent geometries with a new 3D Lagrangian hydrodynamics code, cooper. The code uses a compatible discretization of the conservation equations to ensure that energy is conserved to within machine roundoff error [Caramana *JCP* 146, 227 (1998)]. Modifications are made to the discrete equations to ensure that spherically symmetric implosions can be performed on non-orthogonal Cartesian grids [Caramana *JCP* 157, 89 (2000)]. Subzonal restoring forces counteract anomalous grid distortions [Carmana *JCP* 142, 521 (1998)] and an edge-centered viscosity is used to capture shocks [Caramana *JCP* 215, 385 (2006)]. Cooper is parallelized using domain decomposition. This is necessary due to the large processor and memory requirements associated with simulations in three dimensions. Advanced computational libraries are used to reduce the complexity of the code without sacrificing features. One example is the MOAB library [Tautges *Engr. Comput.* 20, 286 (2004)] which manages the mesh and is responsible for communicating information between processes.

Work Supported By: University of Rochester, Laboratory for Laser Energetics

Cooper currently has several numerical models for performing hydro and rudimentary thermal transport

- Cooper is able to perform three dimensional Lagrangian fluid calculations in parallel on structured hexahedral grids
- Cooper uses the compatible hydrodynamics framework which guarantees global energy conservation to within round-off error levels.
  - The edge-centered artificial viscosity is used to capture shocks
    - The Wilkins viscosity and curl-q methods have also been implemented but have limited effectiveness
  - Subzonal restoring forces are used to damp anomalous grid distortions
  - Symmetry preservation modifications allow cooper to run simulations of spherical implosions with unequal angular zoning
- Ion-electron temperature equilibration has been implemented using both the Spitzer and BPS equilibration times
- Ion and electron thermal conduction has only been implemented for orthogonal grids

Cooper's code can be examined and used without restrictions

Future Work

- Implement an accurate and conservative rezoner to run ALE simulations
- Extend thermal conduction model
  - Implement diffusion solvers that function on non-orthogonal grids
  - Add flux limiters
- Implement flux-limited multi-group radiation diffusion
- Run additional deceleration phase simulations and validate with analytic/experimental results

Description of Numerical Methods and Simulation Results

3D deceleration simulations have been performed with the compatible hydrodynamics code cooper

- Cooper uses advanced hydrodynamics methods which preserve symmetry and ensure "exact" energy conservation globally[1]
  - Small inaccuracies in energy conservation can accumulate in long simulations on large meshes – Cooper conserves energy to roundoff error levels using the compatible hydrodynamic framework
  - Numerical modifications are needed to avoid artificial distortions when spherically convergent, ICF relevant problems are simulated using a Cartesian discretization
- In planar geometry, Raleigh-Taylor (RT) and Richtmyer-Meshkov (RM) simulations have been performed to validate cooper
  - These instabilities manifest themselves at various stages of ICF implosions and must be accurately modeled
- Simulations in spherically convergent geometries have been performed to model deceleration phase instabilities
  - These instabilities grow at the vapor/shell interface of ICF targets as the shell decelerates. Both RT and RM instabilities grow due to the reflection of shocks from the target center

[1] Caramana *JCP* 146, 227 (1998)

Momentum and energy equations are derived to ensure energy conservation in *discrete* form[1]

- First, the momentum equation is discretized:

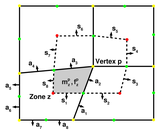
$$m_v \frac{\Delta \mathbf{u}_v}{\Delta t} = \sum_c \mathbf{F}_c^v$$

- Then, a discrete form of the energy conservation equation is derived:

$$\frac{1}{\Delta t}\left[\sum_v m_v \Delta \epsilon_v + \sum_v m_v \mathbf{u}_v^{n+1/2} \cdot \Delta \mathbf{u}_v\right] = \sum_v m_v \frac{\Delta \epsilon_v}{\Delta t} = \sum_v \mathbf{u}_v^{n+1/2} \cdot \sum_c \mathbf{F}_c^v$$

- Finally, specific internal energy is advanced in a manner consistent with energy conservation:

$$\frac{\Delta \epsilon_v}{\Delta t} = -\frac{1}{m_v} \sum_c \mathbf{F}_c^{v} \cdot \mathbf{u}_v^{n+1/2}$$

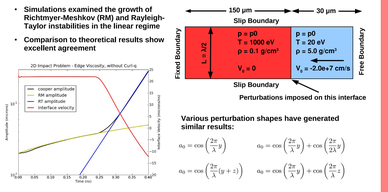


- As long as other numerical methods (such as the edge centered artificial viscosity[1]) are expressed in terms of corner forces, conservation will be preserved
- Energy conservation requires that the corner forces be identical in the energy and momentum equations. To ensure this, a predictor-corrector scheme is used which also leads to second order accuracy in time. This requires twice the work of traditional "leap-frog" approaches

[1] Caramana *JCP* 146, 227 (1998) , [2]Caramana *JCP* 144, 70 (1998)

A model problem involving RT and RM instabilities in slab geometry has been used to validate cooper

- Simulations examined the growth of Richtmyer-Meshkov (RM) and Rayleigh-Taylor instabilities in the linear regime
- Comparison to theoretical results show excellent agreement



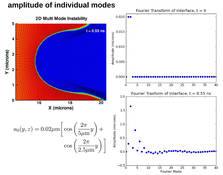
Various perturbation shapes have generated similar results:

$\omega_0 = \cos\left(\frac{2\pi}{\lambda} y\right)$ $\omega_0 = \cos\left(\frac{2\pi}{\lambda} y\right) + \cos\left(\frac{2\pi}{\lambda} x\right)$

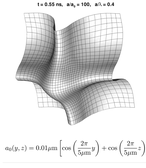
$\omega_0 = \cos\left(\frac{2\pi}{\lambda} (y + z)\right)$ $\omega_0 = \cos\left(\frac{2\pi}{\lambda} y\right) + \cos\left(\frac{2\pi}{\lambda} x\right)$

Numerical simulations have been run into the non-linear regime with 2D and 3D perturbations

Discrete Fourier Transforms can be used to determine the amplitude of individual modes



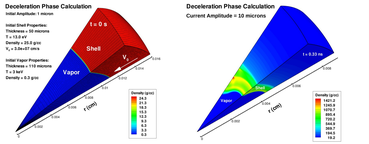
$$a_0(y,z) = 0.02\mu m\left[\cos\left(\frac{2\pi}{5\mu m} y\right) + \cos\left(\frac{2\pi}{2.5\mu m} y\right)\right]$$



$$a_0(y,z) = 0.01\mu m\left[\cos\left(\frac{2\pi}{5\mu m} y\right) + \cos\left(\frac{2\pi}{5\mu m} z\right)\right]$$

Preliminary deceleration phase simulations demonstrate correct qualitative behavior

- The domain is initially divided into a shell region and a vapor region. The shell is given an radially inward initial velocity.
- A perturbation was imposed on the interface with the two regions which grows over time

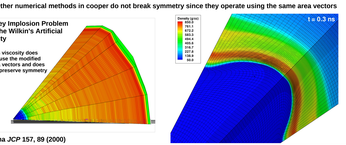


Cooper can run simulations in convergent geometries with non-uniform zoning

- Nonuniform zoning is important for modeling localized structures such as fill tubes
- Since cooper discretizes the hydrodynamic equations in Cartesian geometry it will not preserve symmetry during spherical implosions
  - The cell face areas must be modified to correspond to spherical areas
  - Other numerical methods in cooper do not break symmetry since they operate using the same area vectors

Guderley Implosion Problem using the Wilkin's Artificial Viscosity

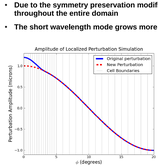
- This viscosity does not use the modified area vectors and does not preserve symmetry

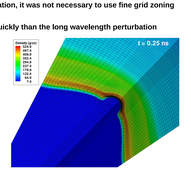


[1] Caramana *JCP* 157, 89 (2000)

Non-uniform zoning can be used to impose a localized perturbation

- A localized, short wavelength perturbation was imposed on a longer wavelength perturbation
- Due to the symmetry preservation modification, it was not necessary to use fine grid zoning throughout the entire domain
- The short wavelength mode grows more quickly than the long wavelength perturbation





Ion-electron temperature equilibration in cooper preserves energy conservation

- Currently, cooper splits the ion-electron temperature equilibration from the conduction calculation:

$$\rho \frac{\partial e_e}{\partial t} = \rho\left(c_{v,e}\frac{\partial T_e}{\partial t} + \frac{\partial e_e}{\partial \rho}\bigg|_{T_e}\frac{\partial \rho}{\partial t}\right) = \frac{\rho c_{v,e}}{\tau_{ei}}(T_i - T_e) + q_e$$

$$\rho \frac{\partial e_i}{\partial t} = \rho\left(c_{v,i}\frac{\partial T_i}{\partial t} + \frac{\partial e_i}{\partial \rho}\bigg|_{T_i}\frac{\partial \rho}{\partial t}\right) = \frac{\rho c_{v,e}}{\tau_{ei}}(T_e - T_i) + q_i$$

- These equations are discretized in time and solved exactly where $\mathbf{m} = \mathbf{c}_{v,e}/\mathbf{c}_{v,i}$:

$$T_e^{n+1} = (A + C) + (B - C)\exp\left[-(1+m)\frac{\Delta t}{\tau_{ei}}\right] + D\Delta t \qquad A = \frac{T_e^n + mT_i^n}{1+m} \quad B = \frac{T_e^n + mT_i^n}{1+m} \quad S_e = \frac{1}{c_{v,e}}\left[\frac{q_e}{\rho} - \frac{\partial e_e}{\partial \rho}\bigg|_{T_e}\frac{\partial \rho}{\partial t}\right]$$

$$T_i^{n+1} = (A - mC) - m(B - C)\exp\left[-(1+m)\frac{\Delta t}{\tau_{ei}}\right] + D\Delta t \quad C = \frac{\tau_{ei}(S_e - S_i)}{(1+m)^2} \quad D = \frac{S_i + mS_e}{1+m} \quad S_i = \frac{1}{c_{v,i}}\left[\frac{q_i}{\rho} - \frac{\partial e_i}{\partial \rho}\bigg|_{T_i}\frac{\partial \rho}{\partial t}\right]$$

- Finally, the expressions for the temperature change are converted *back into changes in specific internal energy* so that the exact energy conservation can be maintained

$$e_e^{n+1} = e_e^n + c_{v,e}\Delta T_e + \frac{\partial e_e}{\partial \rho}\bigg|_{T_e}\Delta\rho \,, \quad e_i^{n+1} = e_i^n + c_{v,i}\Delta T_i + \frac{\partial e_i}{\partial \rho}\bigg|_{T_i}\Delta\rho$$

- It is easy to show that the equations satisfy energy conservation in discrete form regardless of the equation of state since specific internal energy, not temperature, is advanced

Cooper currently supports conduction on orthogonal grids

- Cooper is organized to solve general diffusion equations of the form:

$$a\frac{\partial f}{\partial t} = \nabla \cdot D\nabla f + bf + q$$

- Thus, any physics module can easily use the diffusion solvers, provided that the coefficients f, a, b, q, and D are defined across the domain. This increases the amount of code reuse
  - For ion conduction: a = ρ $c_{v,i}$, D = K_i , q = 0 , b = 0, and f = T_i
  - For electron conduction: a = ρ $c_{v,e}$, D = K_e , q = 0 , b = 0, and f = T_e
  - For multi-group radiation diffusion: a = 1/c , D = 1/3σ_R , q = $\sigma_a$acT⁴ 15/π⁴(P_{g+1} - P_g) , and f = u_g

  (Although at this time only no radiative transfer is supported)
- Initially, diffusion on non-orthogonal grids will be implemented using the 3D Kershaw method[1]. More advanced methods will be implemented subsequently

[1]Fatenejad *JCP* 227, 2187 (2008)

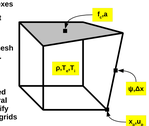
Code Design

Several computational tools are used to simplify cooper without sacrificing functionality

- Cooper is designed to run in parallel with domain decomposition
  - This is necessary due to the large memory and processing requirements associated with running simulations in three dimensions
- Cooper uses a series of libraries which greatly simplify parallel development
  - The MOAB library is used to manage the mesh internally. MOAB also provides the capability to communicate mesh information between processes.
  - The PETSc library performs parallel matrix inversions. PETSc supports multi-grid and Krylov solvers with several preconditioners
  - The HDF5 library stores the initial conditions and all of the output data. HDF5 files are designed to efficiently store, write and read large amounts of scientific data in parallel.
- Cooper's input files are written in the XML format.
  - This offloads input parsing to an XML library. Also, a schema file can be used to automatically validate input outside of cooper.
  - XML is a widely used format and many tools exist which can help the user generate input

The MOAB library[1] manages the mesh for cooper and handles most parallel communication

- MOAB stores the mesh and all data attached to the mesh
  - It provides a convenient interface for setting/accessing data attached to the mesh
  - Data can be "tagged" to cells, faces, edges or vertexes
- MOAB stores mesh data in a novel way which does not sacrifice performance to provide ease of use
- MOAB allows the user to define the boundary region between each processes domain. Data tagged to the mesh can then be "exchanged" between adjacent processes.
- MOAB's interface has been generalized to function on structured and unstructured meshes
  - Since some algorithms are more naturally expressed using a logical i,j,k indexing, cooper contains several wrapper classes around MOAB which further simplify the process of accessing mesh data on structured grids



Cooper is designed to offload several tasks to a pre/post processor

- Cooper only performs the processor/memory intensive physics simulations.
- Additional work, such as preparing the initial grid is performed by the pre/post processor – cooperpp, which is written in the Python programming language
- Cooperpp can perform the following tasks:
  - Generate the mesh in a variety of geometries
  - Place perturbations on the mesh
  - Generate material data for cooper
  - Convert cooper's HDF5 output to other formats, such as VTK files (for Visit) and Tecplot files.
  - Perform Fourier transforms and other analysis on the data. Simple plots can be generated directly within cooperpp
- Performing these tasks in Python (cooperpp) is much easier and less error prone than developing a C++ library (cooper) for both the user and developer