CASTRO: A New AMR Radiation-Hydrodynamics Code for Compressible Astrophysics

A. Almgren[1], J. Bell[1], M. Day[1], L. Howell[2], C. Joggerst[3], E. Myra[4], A. Nonaka[1], J. Nordhaus[5], M. Singer[2], M. Zingale[6]

[1]Lawrence Berkeley National Lab; [2]Lawrence Livermore National Lab; [3]UC Santa Cruz; [4]University of Michigan; [5]Princeton University; [6]Stony Brook University

CASTRO is a new, multi-dimensional, Eulerian AMR radiation-hydrodynamics code designed for astrophysical simulations. The code includes routines for various equations of state and nuclear reaction networks, and can be used with Cartesian, cylindrical or spherical coordinates. Time integration of the hydrodynamics equations uses unsplit PPM with new limiters. Self-gravity can be calculated on the adaptive hierarchy using a simple monopole approximation or a full Poisson solve for the gravitational potential. CASTRO includes gray and multigroup radiation diffusion. Multi-species neutrino diffusion for supernovae is nearing completion. The adaptive framework of CASTRO is based on an time-evolving hierarchy of nested rectangular grids with refinement in both space and time; the entire implementation is designed to run on thousands of processors. Our initial applications of CASTRO include Type Ia and Type II supernovae.

What is CASTRO?

We present a new code, CASTRO, that solves the compressible hydrodynamics equations for astrophysical flows. Key features of CASTRO include:

-AMR framework that allows for variable grid size and simultaneous refinement in space and time

-Support for Cartesian, cylindrical, and spherical coordinate systems

-Unsplit PPM for advection

-Modular equation of state and nuclear reaction networks

-Multiple gravity options, including a full Poisson solve

-Massively parallel; tested on 64,000 processors

-Gray and multigroup radiation diffusion; multi-species neutrino diffusion is nearing completion

Equations

We solve the fully compressible equations (given here without radiation), including advection, diffusion, reactions, and gravity.

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\nabla \cdot (\rho \mathbf{u}) \\
\frac{\partial (\rho \mathbf{u})}{\partial t} &= -\nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla p + \rho \mathbf{g} \\
\frac{\partial (\rho E)}{\partial t} &= -\nabla \cdot (\rho \mathbf{u} E + p\mathbf{u}) + \rho H_{\rm nuc} + \rho \mathbf{u}\cdot\mathbf{g} + \nabla \cdot \kappa \nabla T \\
\frac{\partial (\rho X_k)}{\partial t} &= -\nabla \cdot (\rho \mathbf{u} X_k) + \rho \dot{\omega}_k
\end{aligned}$$

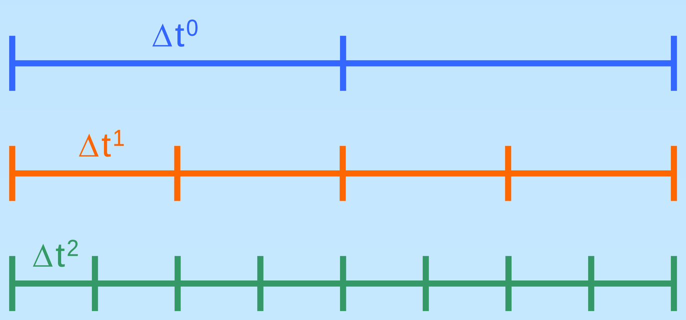
We use Strang splitting for the reaction terms.

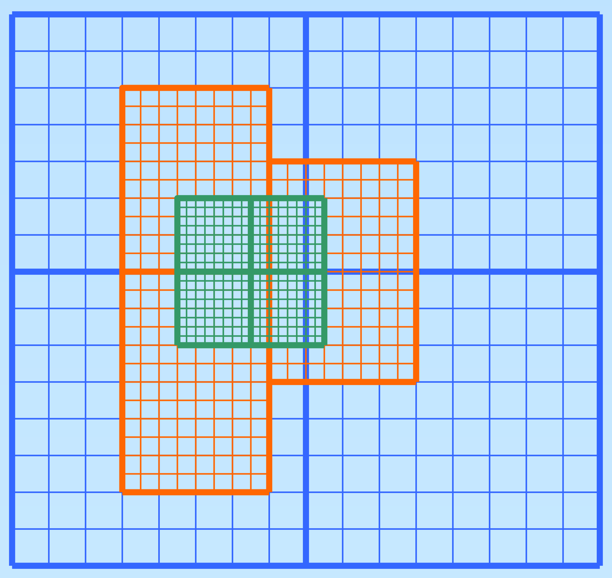
We also provide support for tracers, auxiliary variables, and user-specified external source terms.

Equation of state and reaction networks are modular and are supplied by the user.

AMR

Our AMR approach uses a nested hierarchy of logically rectangular, variable-sized grids with successively finer grids at each level. A user-specified tagging routine indicates where higher resolution is desired.





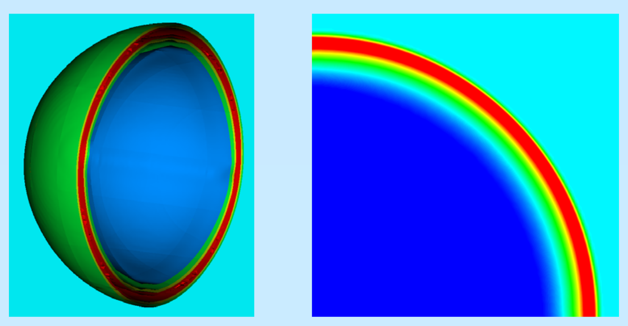
We also subcycle in time, where we advance the finer grids at finer time steps and synchronize the solution between levels to maintain conservation.

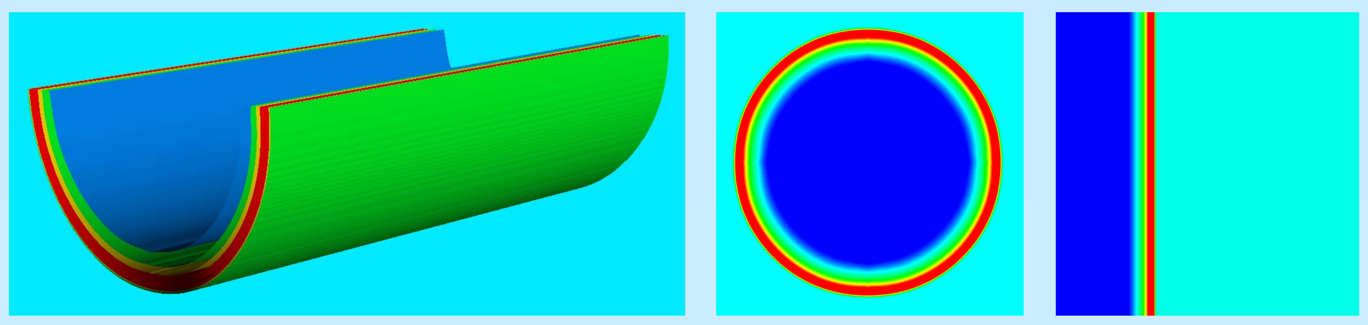
Coordinate Systems

(Above) Spherical Sedov shock in 3D Cartesian and 2D cylindrical coordinates

(Below) Cylindrical Sedov shock in 3D Cartesian, 2D Cartesian, and 2D cylindrical coordinates.

1D Cartesian, cylindrical, and spherical coordinates also available.





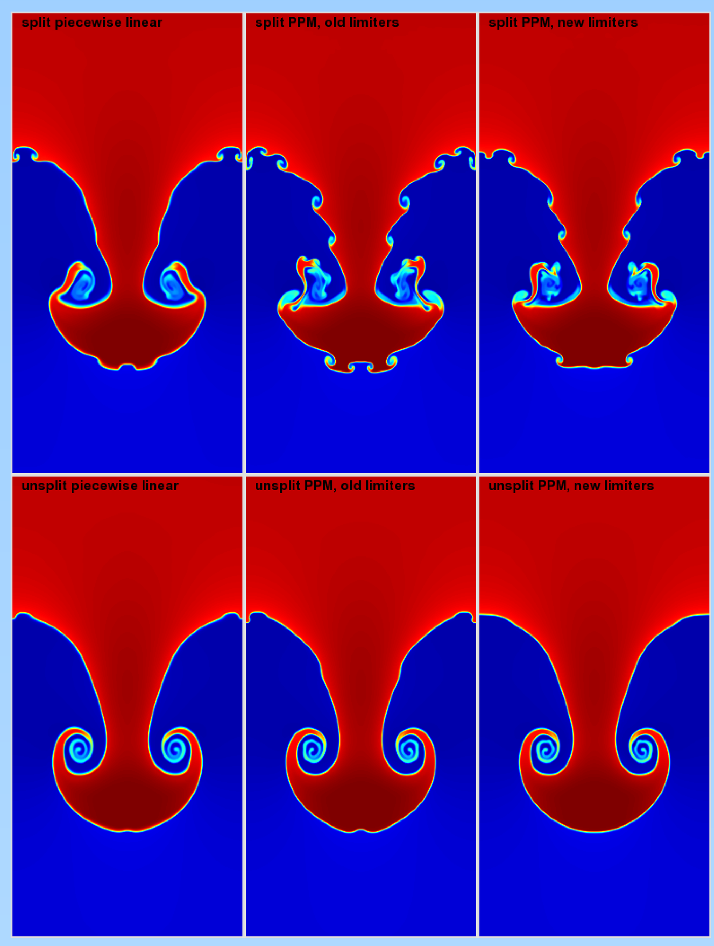
Hydrodynamics

We use an unsplit implementation of a new piecewise-parabolic method (PPM) for the advection terms (Colella 2010, to appear).

Our Rayleigh-Taylor instability simulations demonstrate the effects of various advection schemes.

(Above) We compare a piecewise linear method with PPM in a dimensionally split advection scheme. The "old limiters" option uses the original PPM limiters of Colella and Woodward (1984); the "new limiters" option uses the new limiters mentioned above.

(Below) Here we compare piecewise linear with the two PPM options in the context of our unsplit advection scheme.



Gravity

CASTRO has several different run-time options for how to specify and/or compute the gravitational acceleration. The first option is a constant gravity in space and time; this can be used for small-scale problems in which the variation of gravity throughout the computational domain is negligible.

A second approach uses a monopole approximation to compute a radial gravity consistent with the mass distribution. We first compute the average density profile as a 1D radial array. Then gravity is computed as a direct integral of the mass enclosed. The 1D gravity profile is then interpolated onto the Cartesian grid at each refinement level.

The most general option is the Poisson solve for self-gravity, in which we solve

$$\nabla^2 \phi = 4\pi G \rho; \quad \mathbf{g} = -\nabla \phi.$$

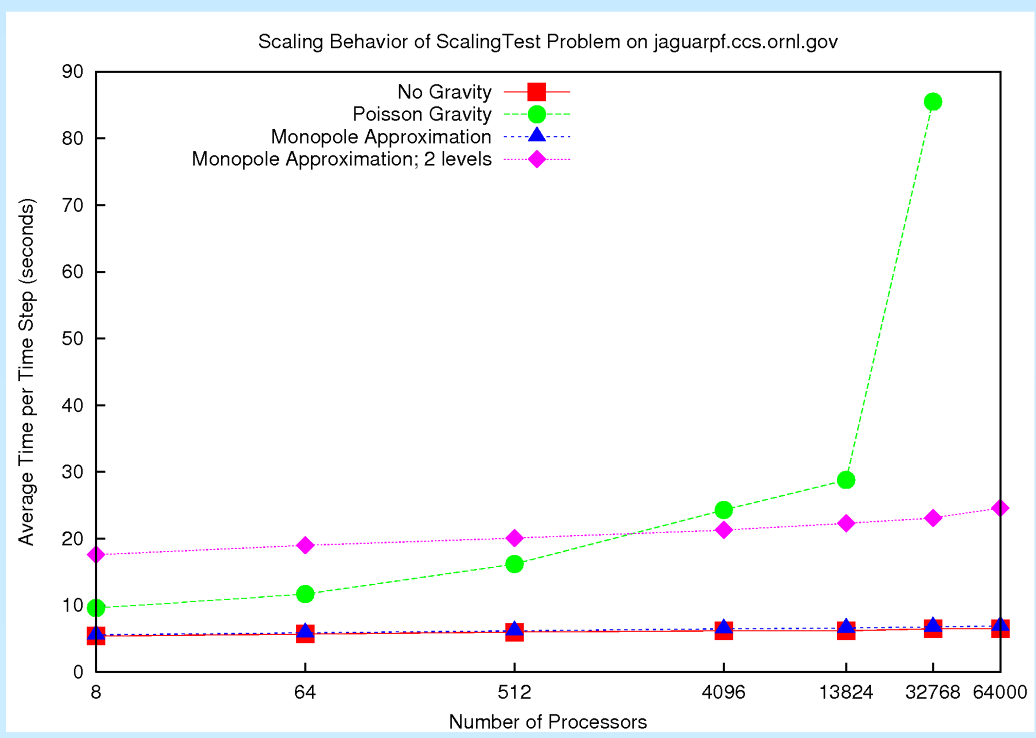
This can be used in one, two, or three spatial dimensions. For multilevel calculations, special attention is paid to the synchronization of the gravitational forcing across levels. However, for simulations in which the gravitational field is sufficiently smooth, it is possible to speed up the code by omitting the multilevel synchronization.

The Poisson equation is discretized using standard finite difference approximations and the resulting linear system is solved using geometric multigrid techniques, specifically V-cycles and red-black Gauss-Seidel relaxation.

At boundaries away from the star we set Dirichlet boundary conditions for φ; these value are determined by computing the monopole approximation for gravity on the coarsest level, integrating this profile radially outward to create φ(r), and interpolating to define the boundary conditions for the solve.

Parallel Performance

Below are scaling numbers for a full white dwarf on a 3D grid with one 64^3 grid per processor. With the monopole approximation to gravity, CASTRO scales well to 64,000 processors. With the full self-gravity using Poisson solves the scalability is limited by the cost of the multigrid solve; this is an area of active research by many groups.



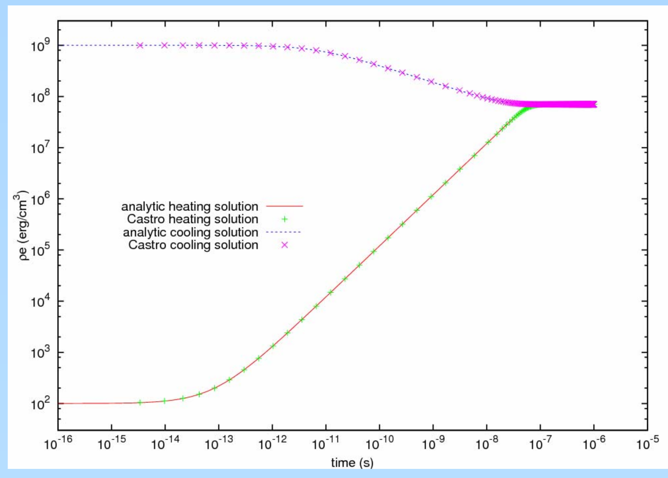
Radiation

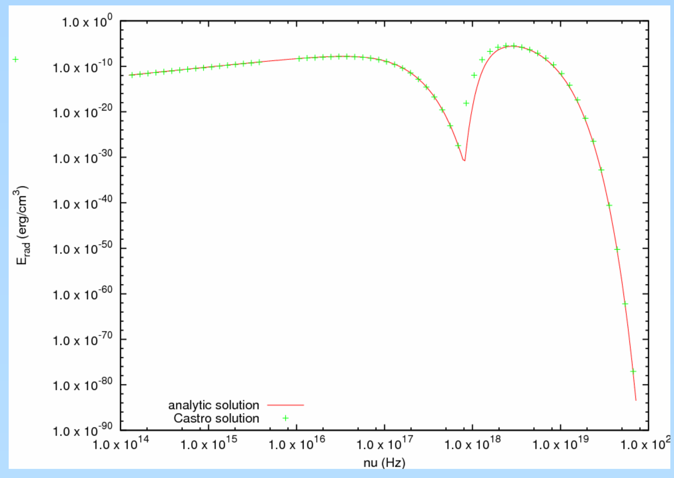
There are two photon radiation solvers in CASTRO - a gray solver and a multigroup solver. The gray solver follows the algorithm outlined in Howell and Greenough (2003). In particular, the radiation energy takes the form of:

$$\frac{\partial E_R}{\partial t} = \nabla \cdot \left(\frac{c\lambda(E_r)}{\kappa_R} \nabla E_R \right) + \kappa_P (4\sigma T^4 - cE_R).$$

CASTRO allows for κ_R and-mean opacity, κ_P is the Planck-mean opacity, and λ is a quantity ≤ 1/3 that is subjected to limiting to keep the radiation field causal. CASTRO allows for κ_R and κ_P to be set independently as power-laws; more generally these will be set based on the local properties of the material. The multigroup solver implements a similar diffusion equation for each radiation energy group.

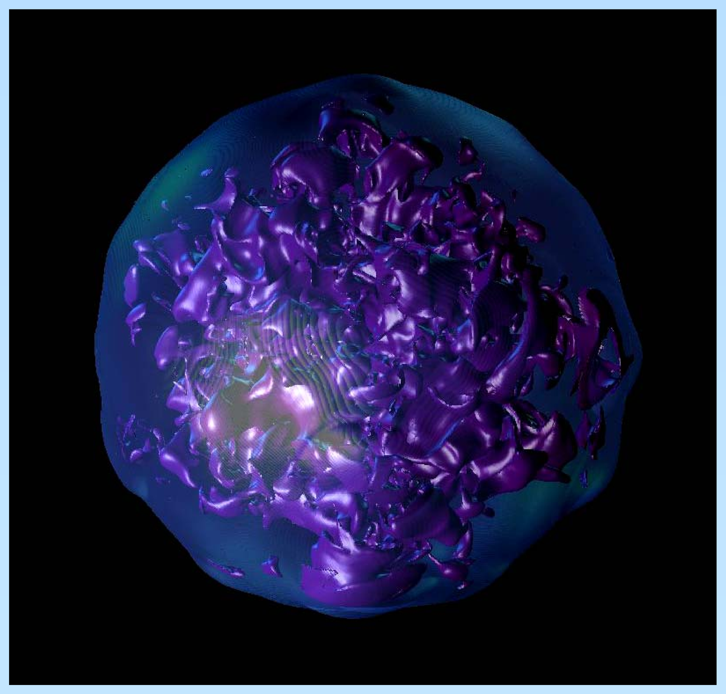
Below are two radiation test problems. On the left is a radiation source problem, which uses the gray photon solver to test the coupling between the radiation field and the gas energy through the radiation source term. Heating and cooling solutions are shown as a function of time, compared to the analytic solution. On the right is a radiating sphere multigroup radiation test problem. A hot sphere is centered at the origin in a spherical geometry. We show the radiation energy density as a function of energy group.





Initial Applications

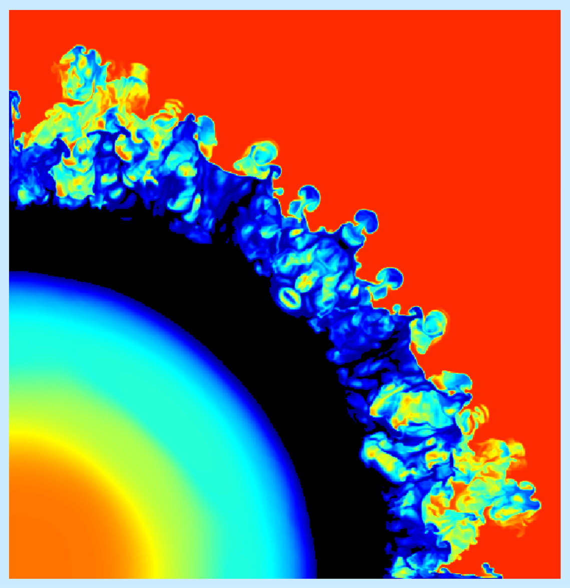
(Right) A 3D simulation of a core-collapse supernova (iso-density and iso- entropy curves). The 15 solar mass progenitor star implodes under its own gravity. During collapse, the equation of state stiffens at nuclear densities and launches a shock which stalls due to neutrino losses and nuclear dissociation. Using a parameterized neutrino heating algorithm (short of full 3D radiation transport), the shock is revived and succeeds in exploding the star. This was run with 4096 cores on Franklin at NERSC in a 4000^3 km^3 domain, divided into 128^3 cells with 3 factors of 4 refinement.

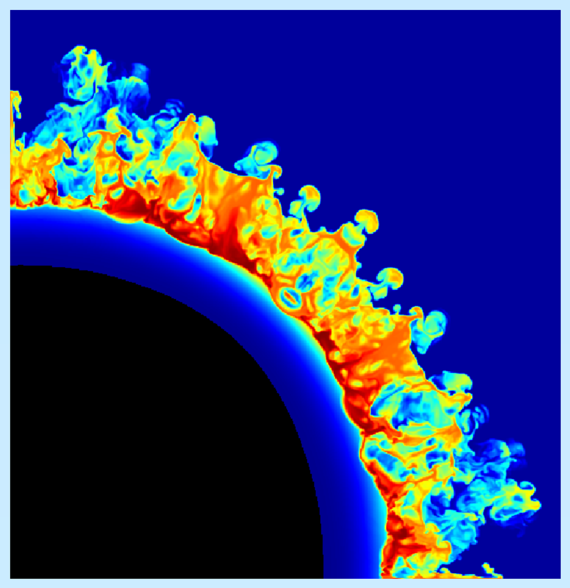


J. Nordhaus

Helium (below, left) and oxygen (below, right) concentration in a three-dimensional simulation of a z=0 solar mass progenitor that was evolved off the main sequence to the point at which its iron core was unstable, and then artificially exploded by a means of a piston with an energy at infinity of 1.2 x 10^{31} erg. A reverse shock has formed and is responsible for the Rayleigh-Taylor instability that's mixing the elemental layers of the star.

This simulation uses self gravity with an infall inner boundary with gravitational contributions from the pointmass at the center and a perfect gas with radiation equation of state. The domain is 2.1 x 10^{13} cm^3, and was run on Franklin at NERSC with 512 processors with 3 levels of refinement.





C. Joggerst

For more information see Almgren et al., to be submitted to ApJ.

This work has been supported by the SciDAC Program of the DOE Office of Mathematics, Information, and Computational Sciences under contracts No. DE-AC02-05CH11231 (LBNL), No. DE-FC02-06ER41438 (UCSC), and No. DE-AC52-07NA27344 (LLNL); and LLNL contracts B582735 and B574691(Stony Brook). Calculations shown were carried out on Franklin at NERSC.