Radiation Transport in Castro

May 19, 2010

Louis Howell
Center for Applied Scientific Computing

Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94551

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

LLNL-PRES-432597

The radiation module for Castro is being developed at LLNL, in collaboration with the code team at LBNL

- Radiation diffusion is primarily the responsibility of Louis Howell in the Center for Applied Scientific Computing (CASC) at LLNL.
  - Mike Singer was a CASC postdoc but has now left this project.
- Andy Nonaka at LBNL is starting to work on radiation also.
- EOS and opacity tables, advice, and preliminary core collapse simulations, have come from Burrows and Nordhaus at Princeton.
- We have collaborated with Zingale, Calder, Myra, and Swesty at Stony Brook on hydro and radiation verification problems.
  - *Hypre* is a general parallel linear solver package developed in CASC. It has multiple interfaces and is not limited to AMR.
- There are also tenuous connections with other LLNL code efforts: mainly software and algorithmic similarities.

The target application is multigroup neutrino diffusion in core collapse supernovae

- There are other astrophysical applications for radiation, of course, but we need a particular serious application to focus development efforts. We will attempt to maintain enough flexibility in the code to explore other applications.
- For this problem we need to model neutrinos in multiple energy groups and at least three different neutrino species. Castro as a whole is intended for high-resolution 3D simulations; doing a full six-dimensional transport calculation with dependence on angle on such a mesh would not be practical.
- We therefore need to work with a flux-limited diffusion model or two-moment system. These are closely related, differing mainly in the time-dependence of the flux, so both can be handled in the same implementation.

For neutrino diffusion we use a mixed-frame, multigroup, multi-species, two-moment system

- Hubeny and Burrows (2007); Mihalas and Klein (1982)

$$\frac{1}{c}\frac{\partial J}{\partial t} + \frac{\partial H^j}{\partial x^j} = \eta_0^{\text{th}} - \kappa_0 J + \Xi_j H^j$$

$$\frac{1}{c}\frac{\partial H^j}{\partial t} + \frac{\partial \left(f^{ij} J\right)}{\partial x^i} = \frac{v_j}{3c}\eta_0^{\text{th}}\left(2 - \frac{\partial \ln \eta_0^{\text{th}}}{\partial \ln \nu}\right) - \left(\kappa_0 + \sigma_{\text{tr}}\right)H^j + \xi_j J$$

$$\frac{D(\rho E)}{Dt} + \cdots = 4\pi \sum_i \int_0^\infty \left(\kappa_0 J - \eta_0^{\text{th}} - \Xi_j H^j\right) d\nu$$

$$\frac{D(\rho v_j)}{Dt} + \cdots = \frac{4\pi}{c}\sum_i \int_0^\infty \left[\left(\kappa_0 + \sigma_{\text{tr}}\right)H^j + O\left(\tfrac{v}{c}\right)\right] d\nu$$

$$\frac{D(\rho e)}{Dt} + \cdots = 4\pi \sum_i \int_0^\infty \left[\kappa_0 J - \eta_0^{\text{th}} - \frac{v_j}{c} H^j \left(2\kappa_0 + \frac{\partial \kappa_0}{\partial \ln \nu}\right)\right] d\nu$$

$$N_A \frac{D(\rho Y_e)}{Dt} + \cdots = 4\pi \sum_i s_i \int_0^\infty \left[\kappa_0 J - \eta_0^{\text{th}} - \frac{v_j}{c} H^j \left(\kappa_0 + \frac{\partial \kappa_0}{\partial \ln \nu}\right)\right] \frac{d\nu}{\nu}$$

So how do we solve those equations that you can't see anymore, anyway?

- Ignore or discretize the time derivative of the radiation flux H.
- Solve the H equation for H and substitute this into the J equation, making the latter into a diffusion equation for J. Add flux limiter.
- Implicitly couple to fluid internal energy and electron fraction (Newton iteration for this is the outer loop).
- Solve modified multigroup system iteratively (inner loop).
- For a practical algorithm we require:
  - Multigroup convergence acceleration (multifrequency-gray).
  - Algorithm should conserve energy and lepton number even if neither outer nor inner iteration are converged.
- Explicitly update momentum and fluid total energy once we are done with the implicit update.

Scorecard: What is the current status of radiation development?

We have:

- Hydro and gravity packages (LBNL)
- Gray flux-limited diffusion
- Prototype multigroup implementation
- 2nd prototype multigroup implementation
- Fully-implicit coupling to fluid energy
- 1D, 2D, and 3D Cartesian, 2D RZ, and 1D spherical coordinates
- Parallel AMR
- Several verification test problems, some with analytic solutions
- Interface to scalable linear solvers supporting the form of the systems we need to solve, including nonsymmetric terms associated with the fluid velocity
- Interaction coefficients associated with neutrinos in a Type II supernova

We have:

- Multigroup convergence acceleration (2 ways)
- New AMR sync algorithm that avoids the need for multilevel linear solvers
- Multigroup, multi-species neutrino diffusion, implicit coupling to both fluid energy and electron fraction equations

Work in progress:

- Velocity effects and momentum coupling implemented in neutrino model but still being debugged and tested for robustness
- Testing on small rad / hydro core collapse
- Retrofit developments from neutrino solver back into photon multigroup

Still in future:

- Inelastic scattering
- H as a conserved state quantity
- Tensor Eddington factor

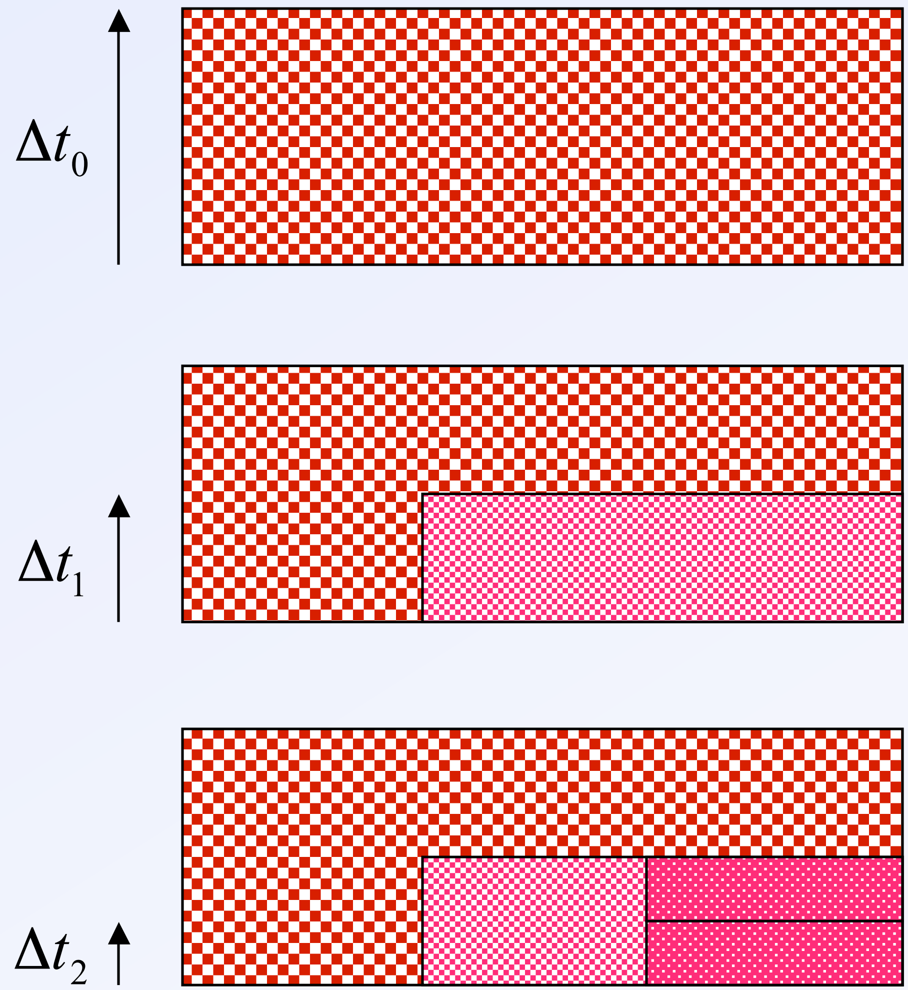
Two of the recent developments are performance improvements in the neutrino algorithm

- Two convergence acceleration schemes have been added to the multigroup iteration. One is an elaboration on the multifrequency-gray algorithm with coupling both to T and Y_e. The other is a simpler local rebalance scheme that performs almost as well in many cases.
  - These make the multigroup iteration many times faster in strongly-coupled regions, and make it more practical to truncate it early.

- The AMR “reflux” algorithm has been changed to eliminate the need for a multilevel sync solve at the end of each coarse timestep.
  - Instead, sync corrections are deferred until the next coarse timestep. This complicates the AMR algorithm (regridding, restart, etc), but eliminates the most expensive calls to the linear solver and may also dodge a potential source of instability.

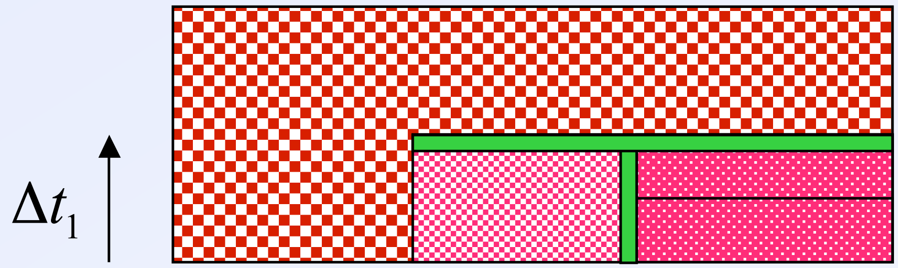
The AMR timestep is a recursive process where coarse grid levels are advanced before finer levels

- Advance Coarse (L0)
  (Level solve at level 0)
- Advance Finer (L1)
  (Level solve at level 1)
- Advance Finest (L2)
  (Level solves at level 2)

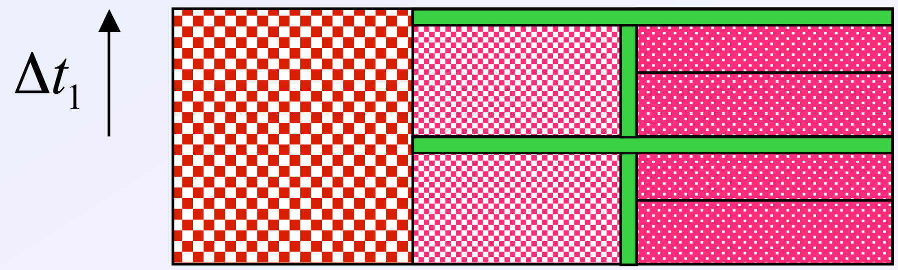


Old: To ensure conservation, the radiation solver requires multilevel synchronization after each coarse step

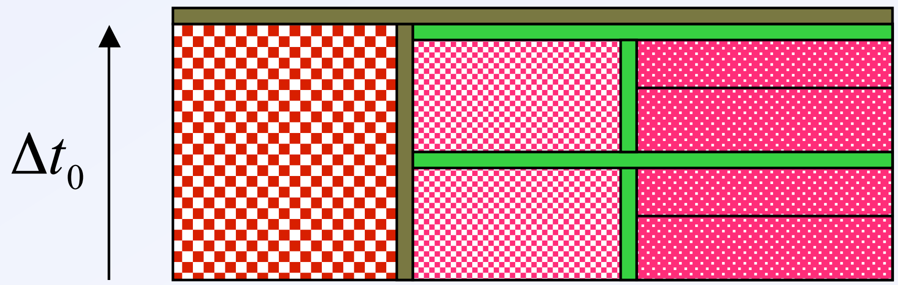
- Synchronize L1 and L2
  (Multilevel solve)



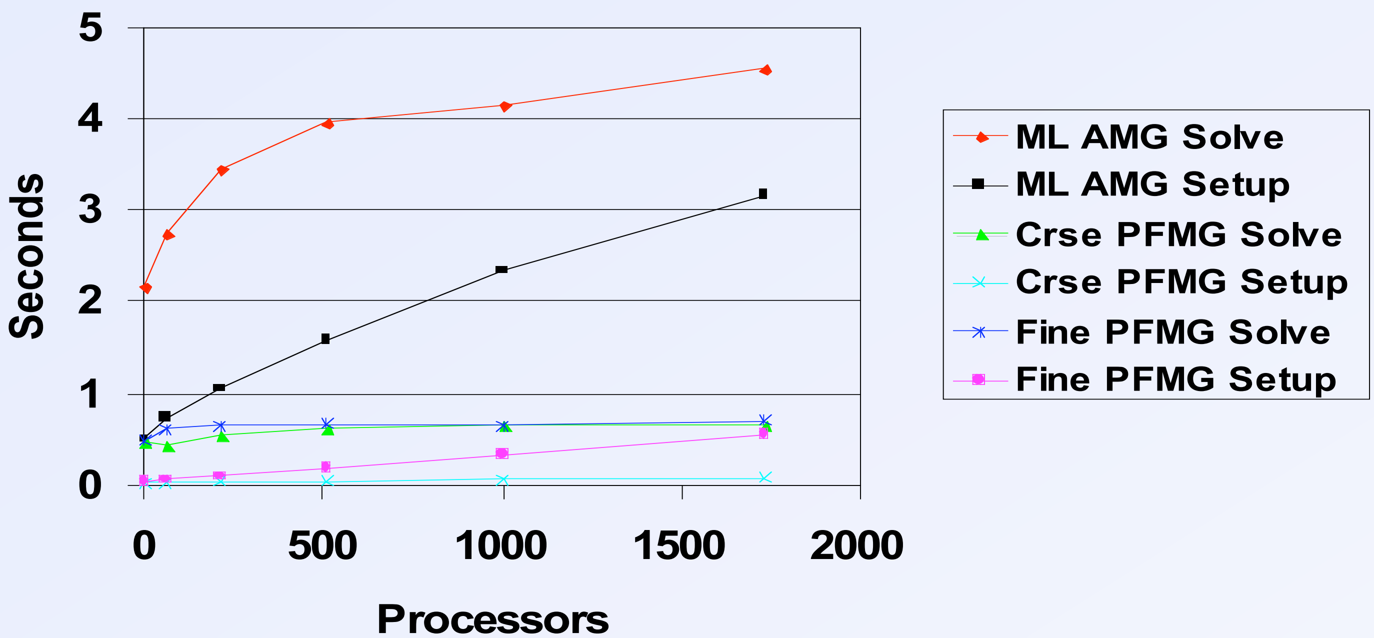
- Repeat L1 and L2
  (advance and synchronize)



- Synchronize L0 and L1
  (Multilevel solve)



AMG is the only solver available for multilevel systems, but for single-level systems PFMG is usually faster



The new AMR sync algorithm eliminates the need for more expensive AMG multilevel solves. It also improves the implicitness of the scheme by avoiding the perturbation from a separate synchronization step.

Testing on a 1D rad-hydro-gravity core collapse without O(*v*/*c*) effects succeeds, but misses important physics

- Electron fraction does not stay within appropriate limits, may reach 0.7 or higher.
- Evidence suggests that missing dynamic diffusion ("advection" of the radiation field) may be the most important factor.
- In the mixed-frame formulation, this means we need the $\xi_j J$ term in the H equation, where ξ_j is

$$\xi_j = \frac{v_j}{3c}\sigma_0\left(2 - \delta - \frac{\partial \ln \sigma_0}{\partial \ln \nu} - \frac{\partial \ln J}{\partial \ln \nu}\right) + \frac{v_i}{c} f^{ij}\left[\tilde{\kappa}_0 + \sigma_0\left(1 + \frac{\partial \ln \sigma_0}{\partial \ln \nu} + \frac{\delta}{3}\frac{\partial \ln(f^{ij} J)}{\partial \ln \nu}\right)\right]$$

We have a preliminary implementation of the O(*v*/*c*) terms, including both Ξ and ξ coefficients

$$\Xi_j = \frac{v_j}{c}\left[\kappa_0\left(1+\frac{\partial\ln\kappa_0}{\partial\ln\nu}\right)+\sigma_{\mathrm{tr}}\left(\frac{\partial\ln\sigma_0}{\partial\ln\nu}+\frac{\partial\ln H^j}{\partial\ln\nu}\right)\right]$$

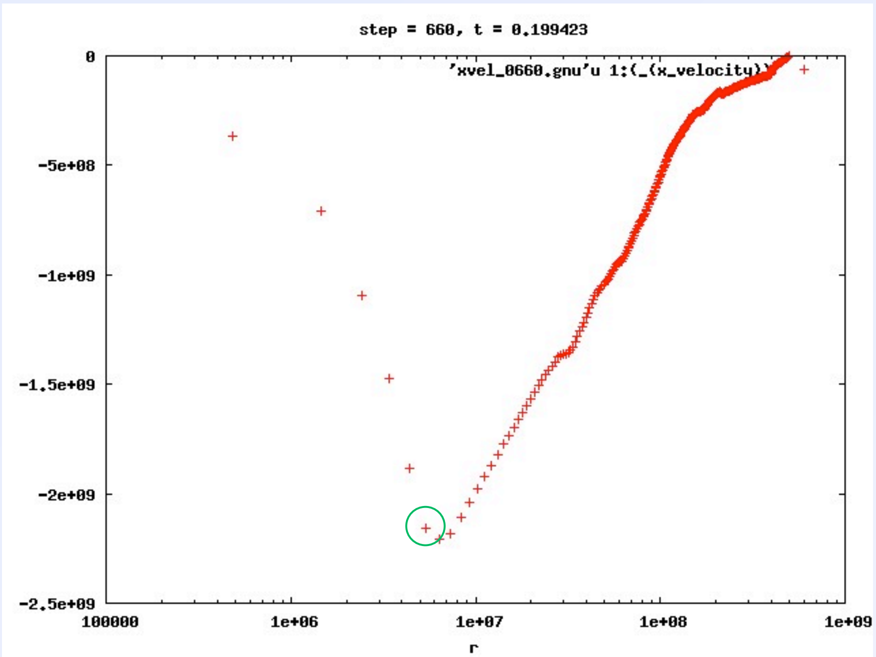
$$\xi_j = \frac{v_j}{3c}\sigma_0\left(2-\delta-\frac{\partial\ln\sigma_0}{\partial\ln\nu}-\frac{\partial\ln J}{\partial\ln\nu}\right)$$
$$+\frac{v_i}{c}f^{ij}\left[\kappa_0\left(1+\frac{\partial\ln\kappa_0}{\partial\ln\nu}\right)+\sigma_0\left(1+\frac{\partial\ln\sigma_0}{\partial\ln\nu}+\frac{\delta}{3}\frac{\partial\ln(f^{ij}J)}{\partial\ln\nu}\right)\right]$$

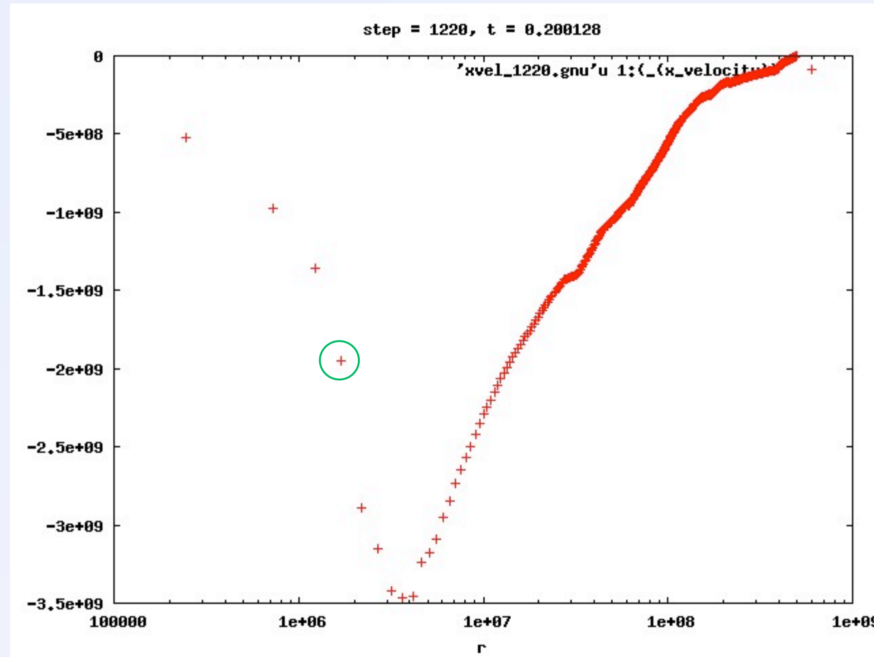
- One complication is the appropriate discretization of the energy derivatives shown here.
- Integration of these terms over energy yields identities for both energy and lepton number. An ideal discretization would satisfy these identities exactly.
- Discretizations with this property are not always robust, particularly when the energy bins are very coarse.

Another issue with the current implementation is convergence of the nonlinear update iteration

- The algorithm is well-behaved at lower velocities.
- When velocities get to around 0.1 c in a core collapse simulation, though, this update loop suddenly becomes unstable.
- The difficulty appears to derive from variation of ξ with respect to T and Y_e, coupled with the steep gradient in ξ as the accretion shock forms. These derivatives are not currently included in the Jacobian because they are formally O(*v*/*c*).
- Several variations and tweaks were tried and did not fix the problem (timestep, iteration parameters, discretizations, lagging ξ, etc).
- It looks like the change in behavior is real, perhaps reflecting the local importance of advection over diffusion. The next thing to try is adding the relevant terms into the Jacobian to test their effect.

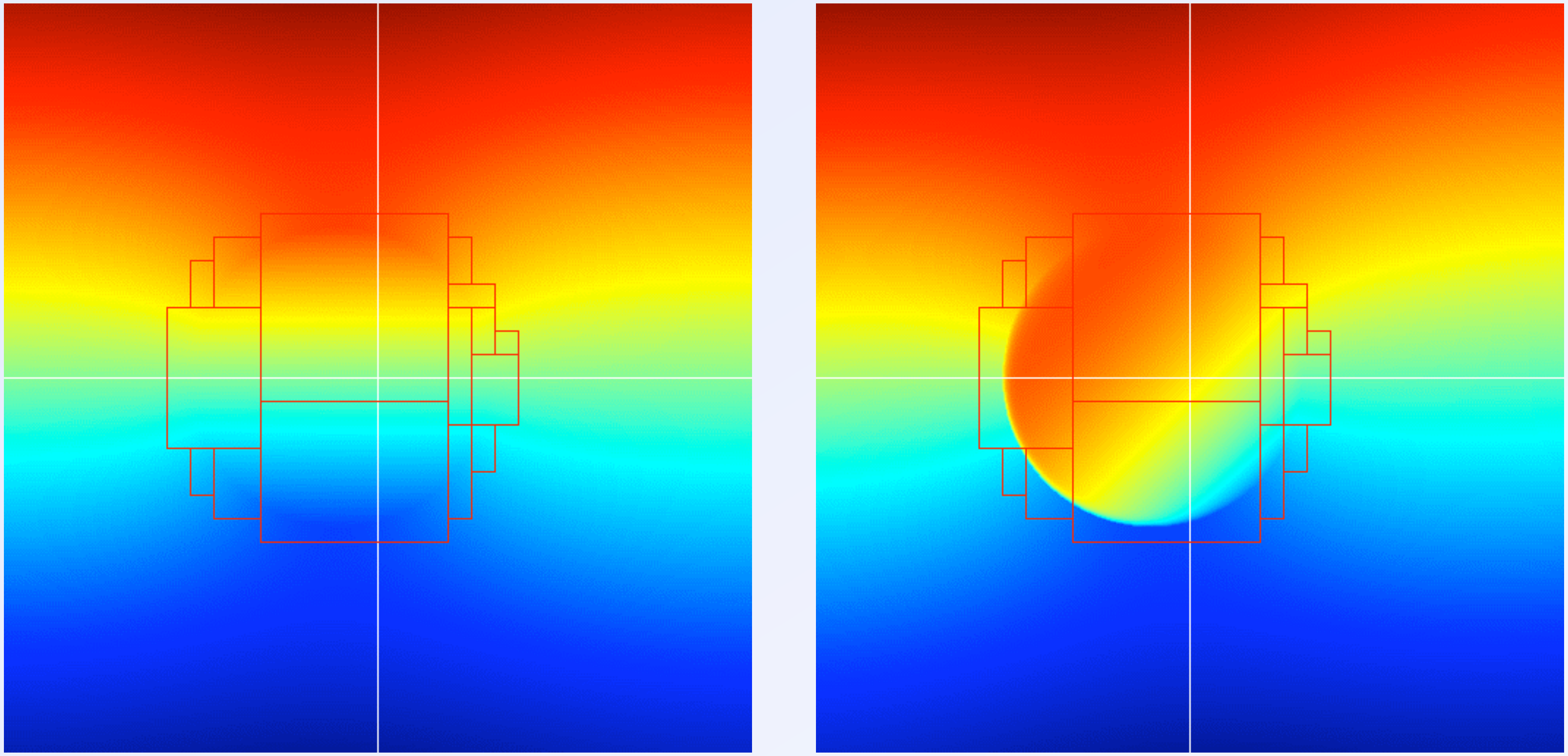
The nonlinear update instability appears at points where advection may dominate over diffusion





Details vary, but the point that first goes unstable always tends to have a combination of high velocity, steep velocity gradient, and material dense enough to trap neutrinos.

Radiation energy density, with and without nonsymmetric matrix perturbation due to fluid velocity



- Scattering coefficient varies by factor of 10^4
- Velocity in second frame is toward lower left at 0.05c

Summary: The neutrino update is implemented and is being tested in coupled simulations

- The code is largely dimension-independent.
- We are using 1D tests to troubleshoot the algorithm itself and for simple code-to-code verification.
- Higher-dimensional tests have been made of the solver infrastructure, including parallel scaling.
- Once we are happy with the algorithm in 1D, the development emphasis in higher dimensions is likely to shift to performance tuning for large-scale coupled simulations.

Algorithmic development in the neutrino solver has leapfrogged past the older multigroup package

- We would like to update the multigroup solver to include the mature form of the implicit update algorithm, multigroup convergence acceleration, and the new AMR sync algorithm.
- This should give us a more practical multigroup (photon) radiation solver for other applications.
- It will also provide a simpler testbed for studying variations on the mixed-frame radiation algorithm.
- Finally, we will be able to directly compare the two AMR sync schemes on the same test problems.

AMR Scaling: 2D grid layout: Isolated groups of fine grids

- To investigate weak scaling in AMR problems, we need to be able to generate "similar" problems of different sizes.
- We tile repetitions of a unit cell with 4 coarse grids in 2D, 8 coarse grids in 3D.
- Each processor gets 1 coarse grid. Fine grid size varies, so different processors get different numbers of fine grids.
- Computational examples shown will all be 3D with nonsymmetric matrices.