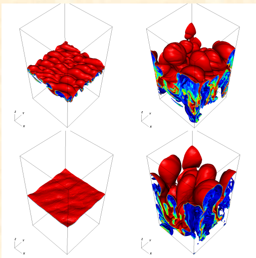
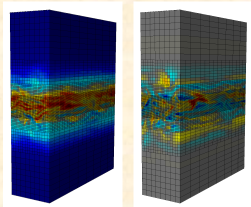
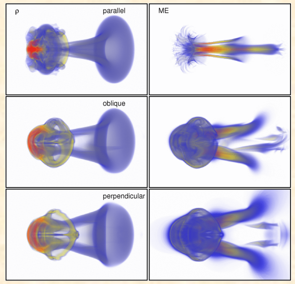
All about Athena

(Five lectures)







Jim Stone
Department of Astrophysical Sciences
Princeton University

Athena: what is it?

- Best to start with the Trac project page
  https://trac.princeton.edu/Athena

Outline of lectures

Lecture 1. Introduction to basic algorithm

Lecture 2. Grids in grid codes

Lecture 3. Extra physics

Lecture 4. Radiation hydrodynamics

Lecture 5. Example applications; future developments

Outline of Lecture 1:

1. The Godunov algorithm
   - Discretization
   - Riemann solvers
   - Reconstruction
   - Unsplit Integrators
2. Implementation issues: the Athena code
3. Tests

Athena solve the equations of ideal MHD in conservative form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

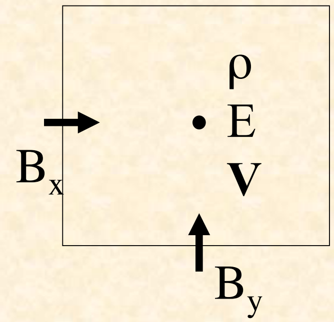
$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot [\rho \mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B} + P^*] = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + P^*)\mathbf{v} - \mathbf{B}(\mathbf{B} \cdot \mathbf{v})] = 0$$

$$\frac{\partial B}{\partial t} - \nabla \times (\mathbf{v} \times \mathbf{B}) = 0$$

The first three equations are differenced using a finite-volume form. The third equation requires something special: finite-area form

Basic Algorithm: Discretization



Scalars *and velocity* at cell centers

Magnetic field at cell faces

Cell-centered quantities *volume-averaged* $\rho^n_{i,j} = \int \rho(t, \mathbf{x}) dV / \int dV$

Face centered quantities *area-averaged* $\mathbf{B}^n_{i,j} = \int \mathbf{B}(t, \mathbf{x}) \cdot d\mathbf{A} / \int dA$

Area averaging is the natural discretization for the magnetic field.

Finite Volume Discretization

Conservations laws for mass, momentum and energy can all be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = 0,$$

Integrate over the volume of a grid cell, and over a timestep dt, apply the divergence theorem to give

$$\begin{aligned}
\mathbf{U}^{n+1}_{i,j,k} = \mathbf{U}^n_{i,j,k} \quad & - \frac{\delta t}{\delta x}\left(\mathbf{F}^{n+1/2}_{i+1/2,j,k} - \mathbf{F}^{n+1/2}_{i-1/2,j,k}\right) \\
& - \frac{\delta t}{\delta y}\left(\mathbf{G}^{n+1/2}_{i,j+1/2,k} - \mathbf{G}^{n+1/2}_{i,j-1/2,k}\right) \\
& - \frac{\delta t}{\delta z}\left(\mathbf{H}^{n+1/2}_{i,j,k+1/2} - \mathbf{H}^{n+1/2}_{i,j,k-1/2}\right)
\end{aligned}$$

(This equation is exact -- no approximations have been made!)

Where, in the previous equations:

$$\mathbf{U}^n_{i,j,k} = \frac{1}{\delta x \delta y \delta z} \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x, y, z, t^n)\ \mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z$$

are "volume averaged" values, while

$$\mathbf{F}^{n+1/2}_{i-1/2,j,k} = \frac{1}{\delta y \delta z \delta t} \int_{t^n}^{t^{n+1}} \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}(x_{i-1/2}, y, z, t)\ \mathrm{d}y\ \mathrm{d}z\ \mathrm{d}t$$

$$\mathbf{G}^{n+1/2}_{i,j-1/2,k} = \frac{1}{\delta x \delta z \delta t} \int_{t^n}^{t^{n+1}} \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{G}(x, y_{j-1/2}, z, t)\ \mathrm{d}x\ \mathrm{d}z\ \mathrm{d}t$$

$$\mathbf{H}^{n+1/2}_{i,j,k-1/2} = \frac{1}{\delta x \delta y \delta t} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{H}(x, y, z_{k-1/2}, t)\ \mathrm{d}x\ \mathrm{d}y\ \mathrm{d}t$$

are "time- and area-averaged" fluxes.

Finite-Area discretization of the induction equation.

Integrate the induction equation over each cell face, apply Stokes Law

$$B^{n+1}_{x,i-1/2,j,k} = B^{n}_{x,i-1/2,j,k} - \frac{\delta t}{\delta y}(\mathcal{E}^{n+1/2}_{z,i-1/2,j+1/2,k} - \mathcal{E}^{n+1/2}_{z,i-1/2,j-1/2,k}) + \frac{\delta t}{\delta z}(\mathcal{E}^{n+1/2}_{y,i-1/2,j,k+1/2} - \mathcal{E}^{n+1/2}_{y,i-1/2,j,k-1/2})$$

$$B^{n+1}_{y,i,j-1/2,k} = B^{n}_{y,i,j-1/2,k} + \frac{\delta t}{\delta x}(\mathcal{E}^{n+1/2}_{z,i+1/2,j-1/2,k} - \mathcal{E}^{n+1/2}_{z,i-1/2,j-1/2,k}) - \frac{\delta t}{\delta z}(\mathcal{E}^{n+1/2}_{x,i,j-1/2,k+1/2} - \mathcal{E}^{n+1/2}_{x,i,j-1/2,k-1/2})$$

$$B^{n+1}_{z,i,j,k-1/2} = B^{n}_{z,i,j,k-1/2} - \frac{\delta t}{\delta x}(\mathcal{E}^{n+1/2}_{y,i+1/2,j,k-1/2} - \mathcal{E}^{n+1/2}_{y,i-1/2,j,k-1/2}) + \frac{\delta t}{\delta y}(\mathcal{E}^{n+1/2}_{x,i,j+1/2,k-1/2} - \mathcal{E}^{n+1/2}_{x,i,j-1/2,k-1/2})$$

Again, these equations are exact -- no approximation has been made.
Moreover, this discretization keeps div(B)=0 to machine precision.

Where, in the induction equation,

$$B^{n}_{x,i-1/2,j,k} = \frac{1}{\delta y \delta z}\int_{z_{k-1/2}}^{z_{k+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}} B_x(x_{i-1/2},y,z,t^n)\ \mathrm{d}y\ \mathrm{d}z$$

$$B^{n}_{y,i,j-1/2,k} = \frac{1}{\delta x \delta z}\int_{z_{k-1/2}}^{z_{k+1/2}}\int_{x_{i-1/2}}^{x_{i+1/2}} B_y(x,y_{j-1/2},z,t^n)\ \mathrm{d}x\ \mathrm{d}z$$

$$B^{n}_{z,i,j,k-1/2} = \frac{1}{\delta x \delta y}\int_{y_{j-1/2}}^{y_{j+1/2}}\int_{x_{i-1/2}}^{x_{i+1/2}} B_z(x,y,z_{k-1/2},t^n)\ \mathrm{d}x\ \mathrm{d}y$$

are "area averaged" components of the magnetic field, and

$$\mathcal{E}^{n+1/2}_{x,i,j-1/2,k-1/2} = \frac{1}{\delta x \delta t}\int_{t^n}^{t^{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}} \mathcal{E}_x(x,y_{j-1/2},z_{k-1/2},t)\ \mathrm{d}x\ \mathrm{d}t$$

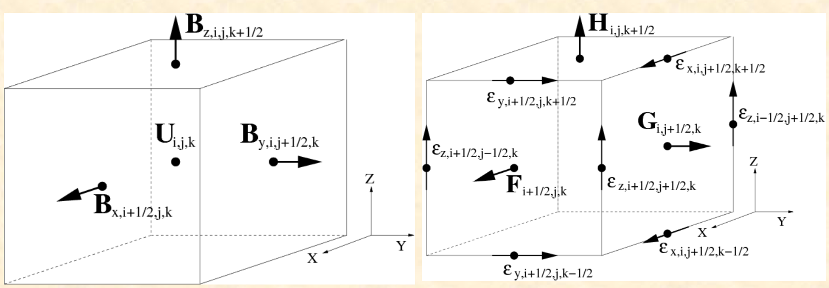
$$\mathcal{E}^{n+1/2}_{y,i-1/2,j,k-1/2} = \frac{1}{\delta y \delta t}\int_{t^n}^{t^{n+1}}\int_{y_{j-1/2}}^{y_{j+1/2}} \mathcal{E}_y(x_{i-1/2},y,z_{k-1/2},t)\ \mathrm{d}y\ \mathrm{d}t$$

$$\mathcal{E}^{n+1/2}_{z,i-1/2,j-1/2,k} = \frac{1}{\delta z \delta t}\int_{t^n}^{t^{n+1}}\int_{z_{k-1/2}}^{z_{k+1/2}} \mathcal{E}_z(x_{i-1/2},y_{j-1/2},z,t)\ \mathrm{d}z\ \mathrm{d}t$$

are "time- and line-averaged" electric field (v x B).

Summary of the discretization.

Uses cell-centered mass, momentum, energy; face-centered field:

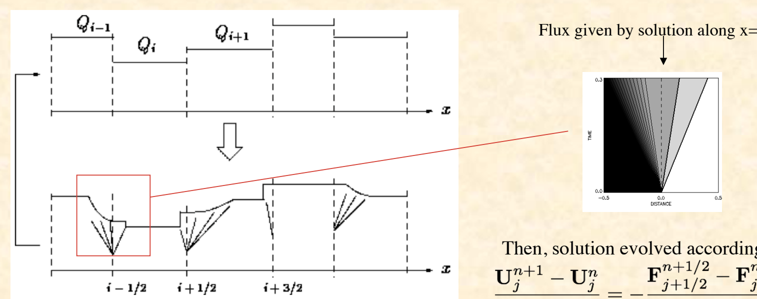


Uses face-centered fluxes, and edge-centered EMFs.

The key is how to compute these fluxes and EMFs all at once!

Godunov's orginal (first-order) method

- Difference in cell-averaged values at each grid interface define set of Riemann problems (evolution of initially discontinuous states).
- Solution of Riemann problems averaged over cell give time-evolution of cell-averaged values, until waves from one interface crosses the grid and interacts with the other, that is for $\Delta t \leq \Delta x/(v+C)$
- Due to conservation, don't actually need to solve Riemann problem exactly. Just need to compute state *at location of interface* to compute fluxes.



Then, solution evolved according to

$$\frac{\mathbf{U}^{n+1}_j - \mathbf{U}^n_j}{\Delta t} = -\frac{\mathbf{F}^{n+1/2}_{j+1/2} - \mathbf{F}^{n+1/2}_{j-1/2}}{\Delta x}$$

Riemann solvers

For pure hydrodynamics of ideal gases, exact/efficient nonlinear Riemann solvers are possible.

In MHD, nonlinear Riemann solvers are complex because:

1. There are 3 wave families in MHD – 7 characteristics
2. In some circumstances, 2 of the 3 waves can be degenerate (e.g. $V_{Alfven} = V_{slow}$)

Equations of MHD are not *strictly hyperbolic*
(Brio & Wu, Zachary & Colella)

Thus, in practice, MHD Godunov schemes use approximate and/or linearized Riemann solvers.

Many different approximations are possible:

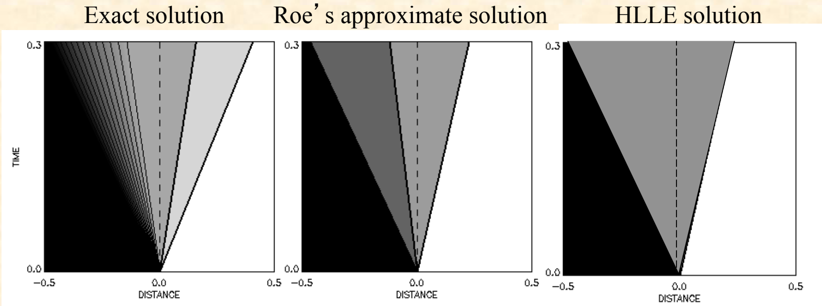
1. Roe's method – keeps all 7 characteristics, but treats each as a simple wave.
   Good resolution of all waves
   Requires characteristic decomposition in conserved variables
   Expensive and difficult to add new physics
   Fails for strong rarefactions

2. Harten-Lax-van Leer-Einfeldt (HLLE) method – keeps only largest and smallest characteristics, averages intermediate states in-between.
   Very simple and efficient
   Guarantees positivity in 1D
   Very diffusive for contact discontinuities

3. HLLC(HLLD) methods – Adds entropy (and Alfven) wave back into HLLE method, giving two (four) intermediate states.
   Reasonably simple and efficient
   Guarantees positivity in 1D
   Better resolution of contact discontinuities

Effect of various approximations on the solution to the Riemann problem in hydrodynamics

Exact solution — Roe's approximate solution — HLLE solution

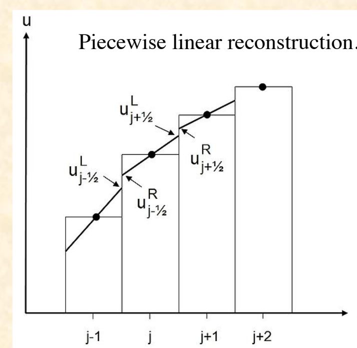


So which approximation is "best"? Must explore the use of each.

Use of a Riemann solvers is a benefit, not a weakness, of a Godunov method: makes shock capturing more accurate.

Higher-order reconstruction

- Using cell-centered values for left- and right-states to define Riemann problems at cell interfaces is first-order and very diffusive.
- Higher-order methods use piecewise linear (MUSCL) or piecewise-parabolic (PPM) reconstruction within cells.
- Difference between L/R states is small for smooth flow, large near shocks. Riemann solver automatically gives correct dissipation for shocks. *No artificial viscosity is needed*.



van Leer unsplit integrator.

Stone & Gardiner, NewA, 2009

- For multidimensional hydrodynamics, directional splitting can be used.
- For MHD, unsplit integrators are necessary if the conservative form is adopted.
- Simplest integrator: modified MUSCL-Hancock ("van Leer") method due to Falle (1991).

Steps in algorithm

1. Compute first-order fluxes at every interface
2. Use these fluxes to advance solution for Δt/2 (predict step)
3. Compute L/R states using time-advanced state, and compute fluxes
4. Advance solution over full time step (correct step) using new fluxes

Since this is a multi-step method, time-advance of L/R states (characteristic tracing) is NOT needed in reconstruction step.

This greatly simplifies algorithm, and makes it much easier to extend to multi-physics, since characteristic decomposition of linearized equations not needed.

Corner transport upwind (CTU) integrator

A more accurate unsplit integrator is due to Colella (1990), extended to MHD by Gardiner & Stone (2005; 2008)

Steps in algorithm:

1. Compute L/R states including time advance using characteristic tracing and source terms for multi-dimensional MHD
2. Compute fluxes from Riemann solver
3. Correct L/R states with transverse flux gradients for Δt/2 including source terms for MHD, e.g. in 2D x-face states corrected via:

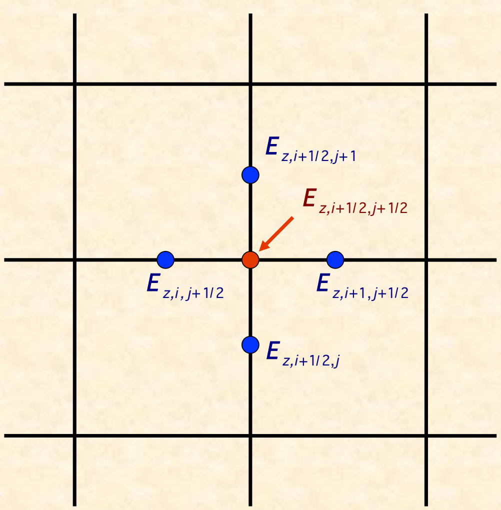
$$\mathbf{q}^{n+1/2}_{L,i-1/2,j} = \mathbf{q}_{L,i-1/2,j} + \frac{\delta t}{2\delta y}\left(\mathbf{g}^{*}_{i-1,j+1/2} - \mathbf{g}^{*}_{i-1,j-1/2}\right) + \frac{\delta t}{2}\mathbf{s}_{x,i-1,j}$$

$$\mathbf{q}^{n+1/2}_{R,i-1/2,j} = \mathbf{q}_{R,i-1/2,j} + \frac{\delta t}{2\delta y}\left(\mathbf{g}^{*}_{i,j+1/2} - \mathbf{g}^{*}_{i,j-1/2}\right) + \frac{\delta t}{2}\mathbf{s}_{x,i,j}$$

4. Compute multi-dimensional fluxes from corrected L/R states
5. Advance solution full time step using multi-dimensional fluxes

see Stone et al, APJS, 2008

Constrained Transport in 2D



- Finite Volume / Godunov algorithm gives E-field at face centers.
- "CT Algorithm" needs E-field at grid cell corners.
- Arithmetic averaging: 2D plane-parallel flow does not reduce to equivalent 1D problem
- Algorithms which reconstruct E-field at corner are superior Gardiner & Stone 2005

Advection of a field loop (2N x N grid)

Field Loop Advection ($\beta = 10^6$): MUSCL - Hancock integrator
Movies of B^2

Arithmetic average (Balsara & Spicer 1999) | Gardiner & Stone 2005

Good test of stability of CT algorithm (obviously trivial for vector potential approaches)

Good test of whether codes preserves div(B) on appropriate stencil: Run in 3D with non-zero V_z. Does method keep B_z zero?

Carbuncle instability

Quirk 1994, Sutherland et al 2003

Small perturbations in upstream flow produce large perturbations in postshock gas.

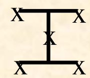
For grid aligned shocks, transverse dissipation is too small to damp perturbations. Transverse pressure gradient produces flow which amplifies perturbations in shocks: *carbuncle instability*

Solution, increase dissipation in transverse direction for grid-aligned shocks, e.g. using **H-correction** in Roe solver:

Replace eigenvalues (wave speeds) in Roe's linearization with $|\bar{\lambda}^{\alpha}| = \max(|\lambda^{\alpha}|, \bar{\eta}_{i-1/2,j})$.

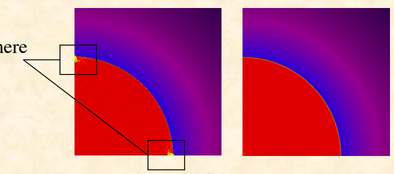
where $\bar{\eta}_{i-1/2,j} = \max(\eta_{i-1,j+1/2}, \eta_{i-1,j-1/2}, \eta_{i-1/2,j}, \eta_{i,j+1/2}, \eta_{i,j-1/2})$

$$\eta_{i-1/2,j} = \frac{1}{2}|(u_{i,j} + C_{f,i,j}) - (u_{i-1,j} - C_{f,i-1,j})|$$



Test with Noh shocktube (M=10^6 converging flow)

Carbuncle in regions where shock aligned with grid

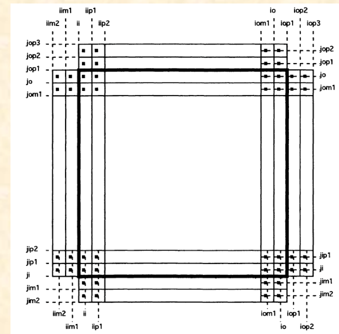


With H-correction, carbuncle is fixed.

Density at t=2 in Noh shock test

Boundary conditions

As in ZEUS, BCs are applied by specifying solution in "ghost zones"



Several default choices for BCs provided, e.g.

1. Reflecting
2. Inflow
3. Outflow
4. Periodic

- Unsplit integrator with PPM requires 4 rows of ghost zones.
- BCs applied only once per time step -- more efficient parallelization.
- New user-defined BCs easily added through use of function pointers.

Athena: one implementation of a MHD Godunov scheme.

- Two versions: C (most capable) and F90 (cleanest)
  - http://www.astro.princeton.edu/~jstone/athena.html for C version
  - http://www.astro.virginia.edu/VITA/athena.php for F90 version
- Modularity: makes extensions to code easier
  - Riemann solvers, reconstruction algorithms, unsplit integrators all separate functions with common interface.
- Ease-of-use:
  - configure in C, modules in F90
  - flexible variety of output files (that don't depend on external libraries!)
  - Input files have intuitive format enabled by special-purpose parser.
- Portability ensured by:
  - Strict adherence to ANSI standards (don't use language extensions!)
  - No reliance on external libraries (except when absolutely necessary, e.g. parallelization with MPI)
- Performance: unsplit integrators require large number (~100) of 3D scratch arrays, however method is so expensive (~10^4 flops per cell) the overall method is cpu, not memory, bound. Requires 10μsec per cell update for 3D MHD.

Configure provides a very useful way to control physics and algorithm options before compiling. Usage: `configure [--with-package=choice] [--enable-feature]`

The -c command-line option enables output of configuration details from executable:

```
ophir> athena -c

Configuration details:

 Problem:                    linear_wave
 Gas properties:             MHD
 Equation of State:          ADIABATIC
 Passive scalars:            0
 Self-gravity:               OFF
 Resistivity:                OFF
 Viscosity:                  OFF
 Thermal conduction:         OFF
 Particles:                  OFF
 Coordinate System:          Cartesian
 Special Relativity:         OFF
 Order of Accuracy:          3 (THIRD_ORDER_CHAR)
 Flux:                       roe
 Unsplit integrator:         ctu
 Precision:                  DOUBLE_PREC
 Ghost cell Output:          OFF
 Parallel Modes: MPI:        OFF
 H-correction:               OFF
 FFT:                        OFF
 Shearing Box:               OFF
 FARGO:                      OFF
 Super timestepping:         OFF
 Static mesh refinement:     OFF
```

Lot's of options!!

See documentation for description of each.

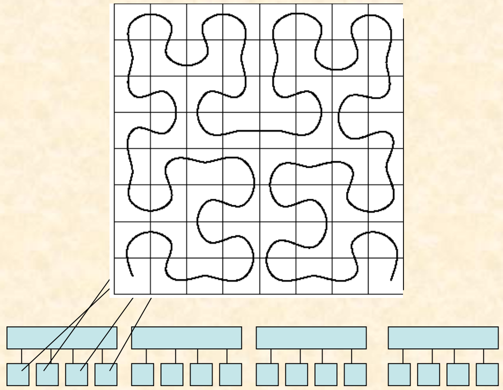
Parallelization

1. Parallelization with MPI via domain decomposition.
   - Any arbitrary decomposition in X, Y, or Z possible (blocks are best for large N_p)
   - Can compute optimum decomposition to minimize data communicated automatically for given N_p
   - No diagonal communication required if data swapped sequentially in each direction.
   - Ideal MPI blocksize seems to be 64^3 on current processors.
2. Balancing workload is easy since flops/zone fixed.
3. Can overlap work and communication by updating outer zones in MPI block first.
4. Tried OpenMP on multi-core, and find it does not perform any better then pure MPI (but saves some memory).
5. FFTs parallelized using block (not just slab) decomposition using Steve Plimpton's interface to FFTW.

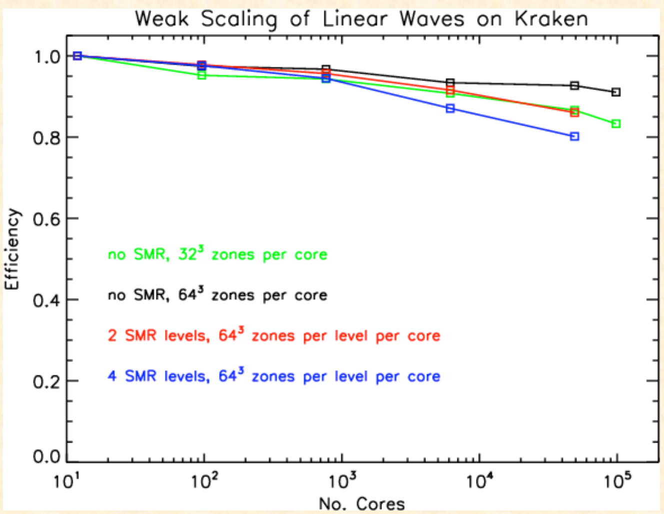
Domain decomposition on multi-core processors to ensure locality

Best to map blocks of MPI domains to cores on processors, rather than using linear ordering along dimensions.

Equivalent to using Peano-Hilbert ordering for space-filling curve.



Weak scaling of Athena is very good, since it is all just explicit MHD (nearest-neighbor communications).



Bug tracking.

Athena uses Trac+SVN to manage software development

See https://trac.princeton.edu/Athena

Site contains documentation, milestones, bug tickets, and ability to browse SVN repository.

Some Tests

Five test problems we have found very useful (all drawn from basic physics of fluids studied in Lecture 1):

1. Linear wave convergence
2. Nonlinear circularly polarized Alfven waves
3. Brio & Wu, and Ryu-Jones shocktubes
4. Field loop advection
5. MHD instabilities (KH, RT, MRI, etc.)

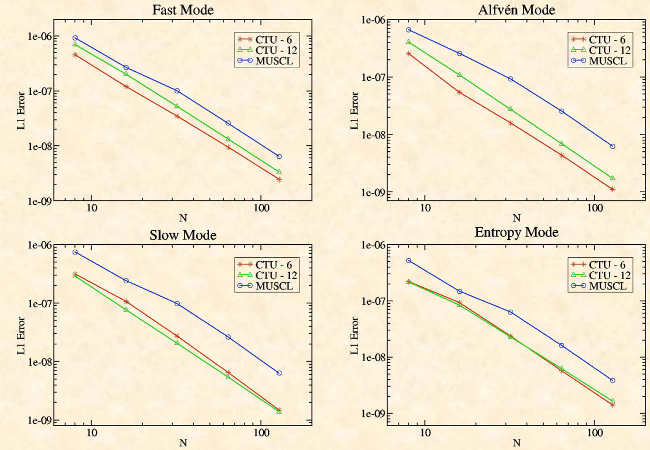
For MHD, ***must*** focus on multidimensional tests.

Convergence rate and ability to capture shocks are equally important.

See http://www.astro.princeton.edu/~jstone/athena.html

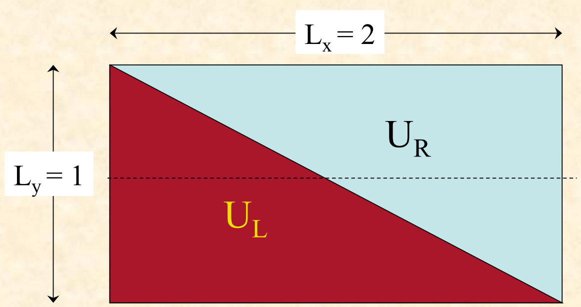
Linear Wave Convergence: 3D (2N x N x N) grid

Initialize pure eigenmode for each wave family
Measure RMS error in U after propagating one wavelength
quantitative test of accuracy of scheme



RJ2a Riemann problem rotated to grid

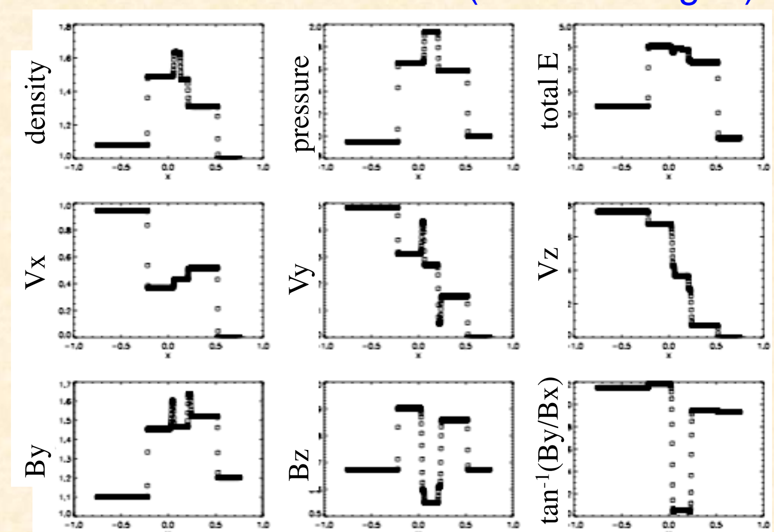
Initial discontinuity inclined to grid at $\tan^{-1}\theta = 1/2$
Magnetic field initialized from vector potential to ensure div(B)=0
$\Delta x = \Delta y$, 512 x 256 grid



Problem is Fig. 2a from Ryu & Jones 1995

Final result plotted along horizontal line at center of grid

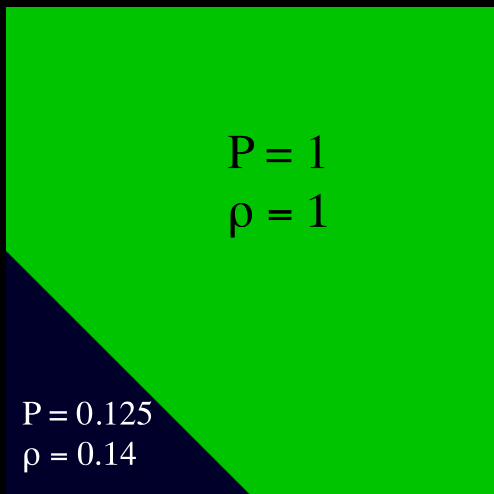
RJ2a shocktube in 3D (2N x N x N grid)



HLLD solver, all 7 MHD waves captured well.

Hydrodynamical Implosion

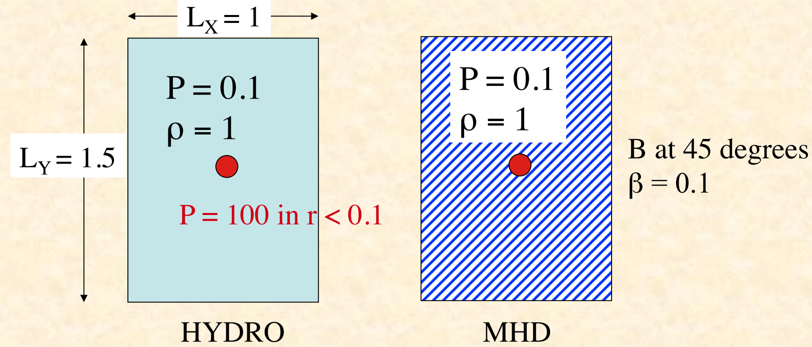
From Liska & Wendroff; 400 x 400 grid,



Additional benefit of using unsplit integration scheme: Code maintains symmetry

RM instability in Spherical Blast Waves

Impulsive acceleration of a dense fluid by a less-dense fluid (e.g. by a shock propagating across a CD) is subject to RT-like instability. Algebraic rather than exponential growth.



Δx = Δy, 400 x 600 grid, *periodic* boundary conditions

Hydrodynamic Blast Wave
400 x 600 grid

Compare to Fig. 23 in Springel (2009)

MHD Blast Wave
400 x 600 grid

Summary

- Godunov methods for MHD are now mature.
- They are an excellent choice for problems involving shocks.
- With CT, divergence-free condition can be enforced to machine precision using Godunov methods.
- Such methods can scale extremely well to $10^{5\text{-}6}$ cores, even with mesh refinement.
- The future: Athena++
  - 10x faster per core (1 μsec per cell for 3D MHD)
  - mixed (OpenMP/MPI) parallelization model for Intel Xeon Phi
  - full GR, including time-dependent metrics
  - relativistic radiation transport
  - AMR