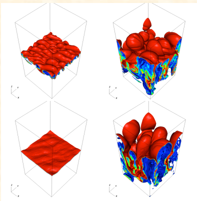
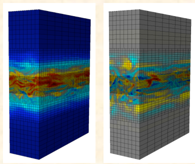
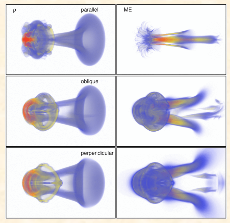
Grid-based methods for hydrodynamics, MHD, and radiation hydrodynamics.



(Four lectures)





Jim Stone
Department of Astrophysical Sciences
Princeton University

http://www.astro.princeton.edu/~jstone/downloads/papers/Lecture2.pdf

Outline of lectures

Lecture 1. Introduction to physics and numerics

Lecture 2. Operator split (ZEUS-like) methods

Lecture 3. Godunov (PPM-like) methods

Lecture 4. Radiation Hydrodynamics

Lecture 2:
Operator split (ZEUS-like) methods

1. An operator split algorithm for MHD.
   - Transport step
   - Source step
   - Artificial viscosity
   - Constrained transport
2. Tests of the method.
3. Implementation in the ZEUS code.
4. ZEUS versus Athena.
5. Other codes.
6. Introduction to Athena: discretization

Numerical Methods

Last lecture we introduced several different methods for the linear advection equation:

- FTCS
- LF
- First-order upwind
- Lax-Wendroff

How do we extend these methods to the nonlinear system of equations that describes MHD?

Start with a simple method: operator splitting

e.g. solve the equations of motion in non-conservative form using operator splitting (e.g. Bowers & Wilson 1991)

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

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v}\mathbf{v} = -\nabla p + \frac{1}{4\pi}(\nabla \times \mathbf{B}) \times \mathbf{B}$$

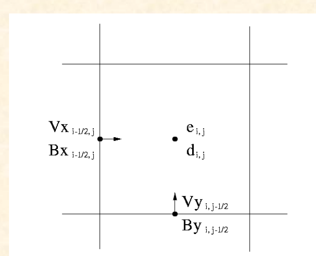
$$\frac{\partial e}{\partial t} + \nabla \cdot e\mathbf{v} = -\frac{p}{\rho}\nabla \cdot \mathbf{v}$$

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

1. LHS=0 evolved using higher-order upwind advection scheme
2. U_t = RHS evolved using finite-differencing, U=(ρv, e)

Obviously something different must be done to evolve B

Such schemes usually based on a staggered mesh



Scalars located at cell centers. Components of vectors located at appropriate cell faces.

Transport step.

Use conservative update of advection terms in "transport step"

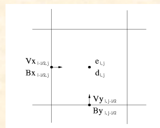
$$\frac{d}{dt}\int_V \rho dV = -\int_{dV} \rho(\mathbf{v}-\mathbf{v}_g)\cdot \mathbf{dS}$$
$$\frac{d}{dt}\int_V \rho\mathbf{v} dV = -\int_{dV} \rho\mathbf{v}(\mathbf{v}-\mathbf{v}_g)\cdot \mathbf{dS}$$
$$\frac{d}{dt}\int_V e dV = -\int_{dV} e(\mathbf{v}-\mathbf{v}_g)\cdot \mathbf{dS}$$

v_g is an arbitrary "grid velocity"

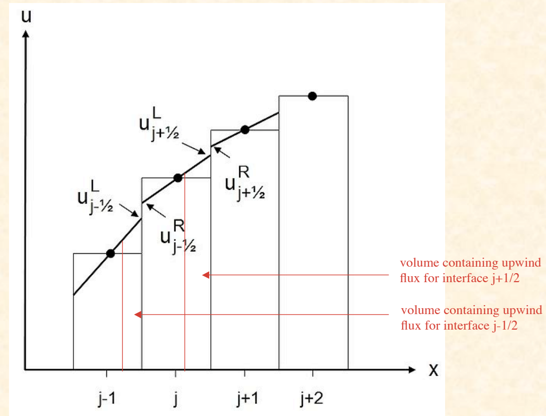
In difference form (conservative, finite volume update):

$$\frac{q_j^{n+1}-q_j^n}{\Delta t} = -\frac{[q^*_{j+1/2}(v-v_g)_{j+1/2} - q^*_{j-1/2}(v-v_g)_{j-1/2}]}{\Delta x}$$

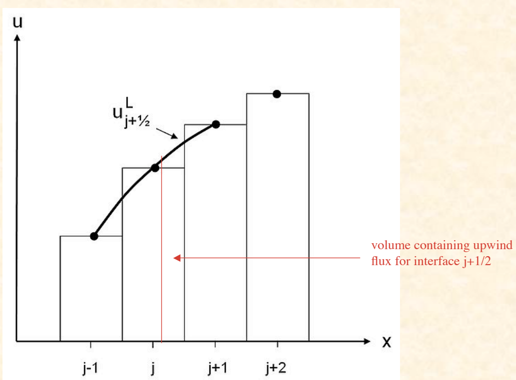
Here, q* is an upwind, interpolated value at cell edges. Note v conveniently located at cell edges as well.



Piecewise linear reconstruction

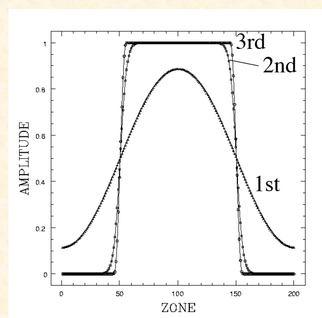


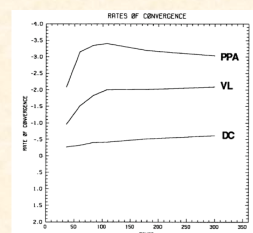
Piecewise parabolic reconstruction



Higher-order reconstruction much less diffusive

Profile of square pulse after advection once around a periodic grid for different order reconstruction.





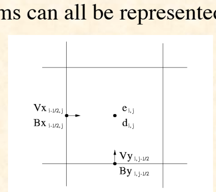
Convergence rates for smooth profiles.

Source step.

Add remaining terms using finite differencing in "source step"

$$\rho\frac{\partial \mathbf{v}}{\partial t} = -\nabla p - \nabla\cdot\mathbf{Q} + \frac{1}{4\pi}(\nabla\times\mathbf{B})\times\mathbf{B}$$
$$\frac{\partial e}{\partial t} = -p\nabla\cdot\mathbf{v} - \mathbf{Q}:\nabla\mathbf{v}$$

- Due to staggered grid, these terms can all be represented using centered differences (2nd order).



- Must add an artificial viscous pressure Q to capture shocks.
- Differencing of Lorentz force is actually quite complex (MoC)

Artificial viscosity

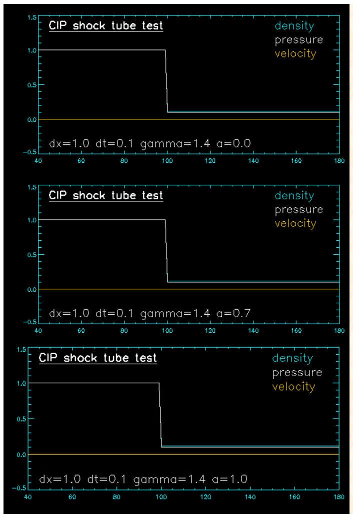
Need **artificial viscosity** to capture shock fronts. Similar to what mediates shocks in real gases (viscosity due to particle collisions). *Artificial* in the sense it should operate only in shocks, and it is much bigger than real viscosity.

von Neumann & Richtmyer proposed adding a scalar viscous pressure of the form

$$q = l^2\rho(\partial v_x/\partial x)^2 \quad \text{if} \quad (\partial v_x/\partial x) < 0$$
$$0 \quad \text{otherwise}$$

Since q is a nonlinear function of $\partial v_x/\partial x$, large only in shocks

Artificial viscosity is ***not*** necessary to make algorithm **stable**.
It is necessary to thermalize kinetic energy in shocks (create entropy).



zero

Too small

Almost right

Effect of artificial viscosity

Extensions to multidimensions

Traditionally, multidimensional methods are constructed using dimensional (directional) splitting:

1. Solve $U_t = F_x$
2. Solve $U_t = G_y$, with G constructed from result of x-update
3. Solve $U_t = H_z$, with H constructed from result of y-update

Sometimes these sweeps are symmetrized to make splitting 2nd order in time (Strang splitting)

This works great for hydrodynamics, and is used for the source and transport steps in ZEUS

BUT: In MHD, this splitting will not preserve div(B)=0.

Must use *directionally unsplit* schemes to update the induction equation, e.g. **constrained transport** (Evans & Hawley 1988)

Keeping div(B) = 0 with CT

Constrained Transport is a conservative scheme for the *magnetic flux.*

Integrate the induction equation over cell face

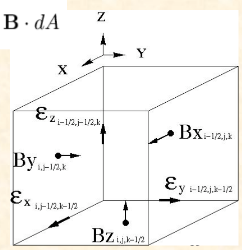
$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}),$$

using Stoke's Law to give

$$\frac{\partial \Phi_B}{\partial t} = \oint_{\partial S} (\mathbf{v} \times \mathbf{B}) \cdot dl \quad \text{where} \quad \Phi_B = \int \mathbf{B} \cdot dA$$

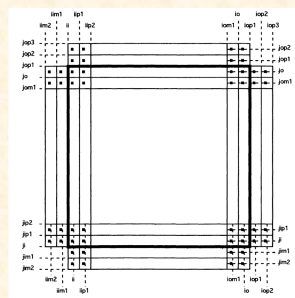
Difference using a staggered B and EMFs located at cell edges.

Appropriately upwinded EMFs must computed from face-centered fields.



Boundary conditions

Most grid codes apply boundary conditions by specifying solution in extra rows of cells ("ghost" or "guard" zones) at boundary of grid. This algorithm requires 2 or 3 rows (for 2nd or 3rd order upwind reconstruction respectively)

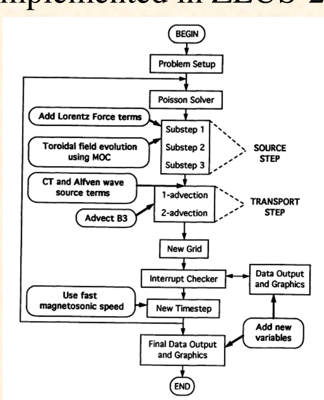


There are different ways of specifying solution in ghost zones for different BCs, e.g.

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

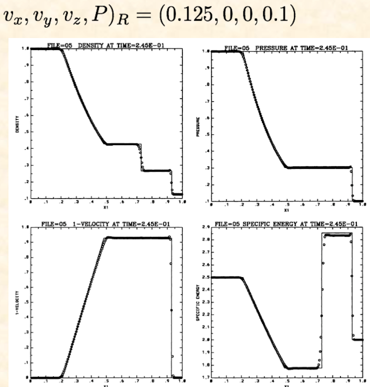
BCs must be applied after every partial update in split methods.

Flow chart of this algorithm, as implemented in ZEUS-2D



Hydro test: Sod shocktube

$$(\rho, v_x, v_y, v_z, P)_L = (1.0, 0, 0, 1.0)$$
$$(\rho, v_x, v_y, v_z, P)_R = (0.125, 0, 0, 0.1)$$



This test is a "low bar" to clear.

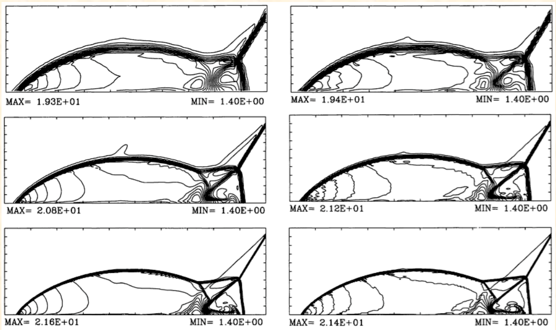
Hydro test: double Mach reflection.

Movie of density

30x120

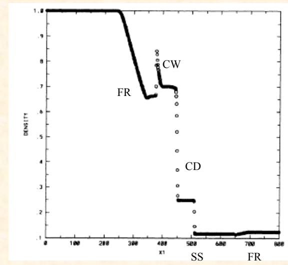
60x240

120x480

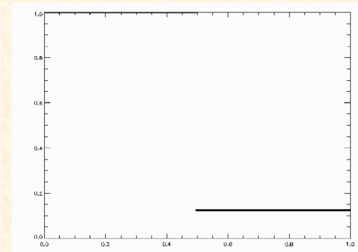


2nd order 3rd order

MHD Riemann problem from Brio & Wu (1988)

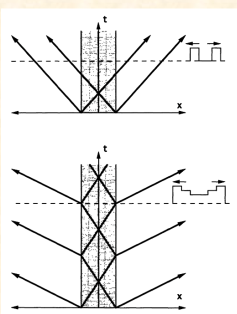


ZEUS

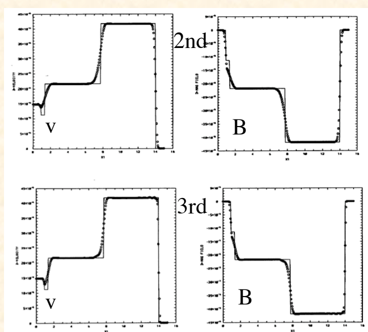


Athena

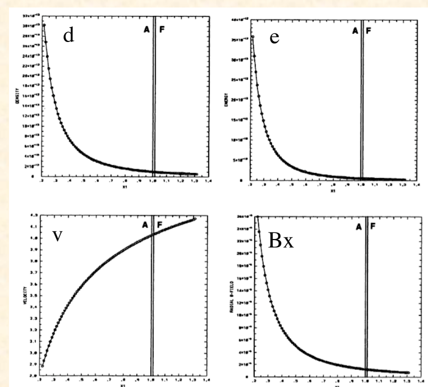
Magnetic braking test.



Space time diagram of solution.

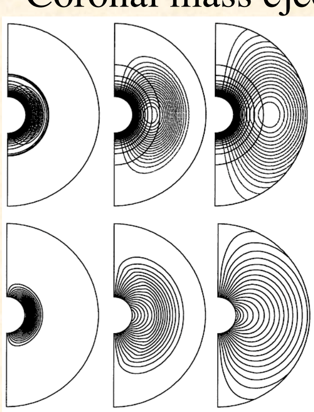


Weber-Davis wind solution.



Rotating, magnetized, pressure-driven spherical wind.

Coronal mass ejection.



Contours of density and representative field lines.

Toroidal component of B

Can be compared to the analytic (!!) solution of Low (1984)

time →

The many flavours of ZEUS.

- **ZEUS05** - first version of ZEUS written by David Clarke from Mike Norman's hydro code.
- **ZEUS-2D** - Stone & Norman (1992). MHD using CT, covariant differencing, self-gravity, full transport radiation hydrodynamics.
- **ZEUS-3D** - David Clarke's ZEUS05 extended to 3D using CT (http://www.ap.smu.ca/~dclarke/zeus3d/)
- **ZEUS3D** - Stone's extension of ZEUS-2D to 3D (http://www.astro.princeton.edu/~jstone/zeus.html)
- **ZEUS-MP/2** - LCA's extension of ZEUS-2D to 3D http://lca.ucsd.edu/portal/software/zeus-mp2
- **ZEUS-MP** (other versions). Various other versions are on web with important bug fixes to ZEUS-MPv1.5, e.g. http://www.netpurgatory.com/zeusmp.html (this site comes up first if you Google 'zeusmp')

Some properties of the implementations.

- All written in FORTRAN
- ZEUS-2D and ZEUS3D use C precompiler (cpp) to allow macros that control physics options. Was necessary because F77 did not have a precompiler. But results in non-standard code (e.g. use of .src instead of .f file extensions).
- ZEUS3D parallelized in OpenMP.
- ZEUS-MP parallelized with MPI using domain decomposition. But have to swap ghost zones several times per time step in operator split algorithm.
- (CMMART parallelized with CM Fortran)

Athena vs. ZEUS

<u>ZEUS (written c.1988)</u>

- Operator split algorithm combining upwind and finite-difference methods
- Still in use (98 citations in 2009)

<u>Athena (written c.2005)</u>

- Fully upwind Godunov scheme

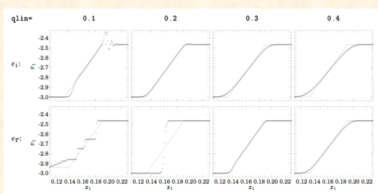
Which code is better?

Well, Athena, of course!

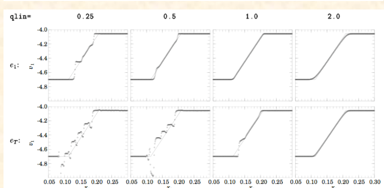
But is ZEUS hopeless for MHD? Especially since Falle (2002) pointed out (quite rightly) that the “off-the-shelf” version of ZEUS-2D failed certain 1-D Riemann problems.

Recently, David Clarke has shown that by adjusting the artificial viscosity coefficients, and adding a total energy equation in some cases, ZEUS-3D v3.5 passes all the tests in Falle (2002)

“rarefaction shocks” fixed by linear viscosity

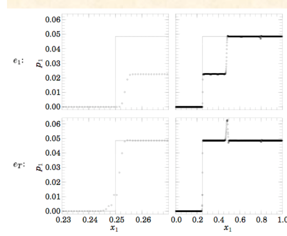


Falle fig. 1

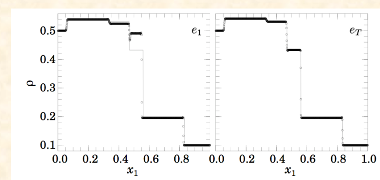


Falle fig. 2

“wrong jump conditions” fixed using total energy option.



Falle fig. 5



Falle fig. 6

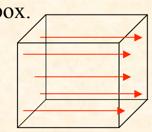
See David Clarke’s ZEUS-3D home page for details: http://www.ap.smu.ca/~dclarke/zeus3d/

Comparison of Athena and ZEUS on real applications: Supersonic MHD turbulence.

Solve equations of isothermal MHD in periodic box.

Uniform density and B initially.

Add arbitrary velocity fluctuations with the following properties:



$$\langle|\delta v_k|^2\rangle \propto k^6 \exp\left[-8k/k_p\right]$$ ← assumed power spectrum

$$\int dV\,(\rho/2)\left[|\mathbf{v}+\delta\mathbf{v}|^2 - |\mathbf{v}|^2\right] = \dot{E}\Delta t$$ ← normalization

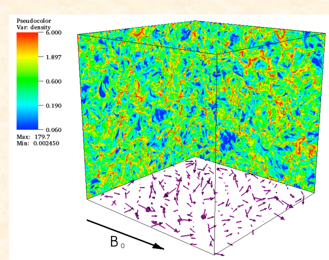
$$\int dV\,\rho\,\delta\mathbf{v} = 0$$ ← no net momentum added

$$\nabla\cdot\delta\mathbf{v} = 0$$ ← perturbations incompressible

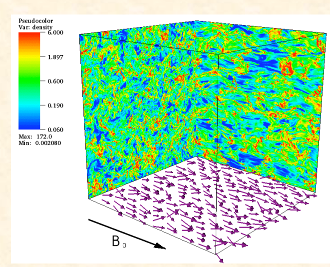
New realization of driving perturbations every few timesteps

Density and magnetic field lines, 512^3 grid.

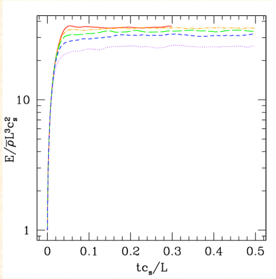
$\beta=1$



$\beta=0.01$



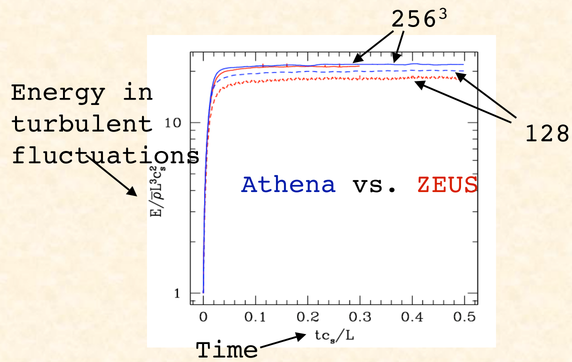
Convergence with resolution is clearly evident, 32^3 to 512^3



$k_p=4$
$\beta=0.01$

Variety of results reported in Lemaster & Stone (2008a; 2008b) using resolution up to 1024^3 for (1) energetics, (2) density PDF, (3) Fourier power spectra, (4) sonic scaling, (5) intermittency.

But how does ZEUS perform compared to Athena on real 3D applications? For example, *supersonic MHD turbulence*

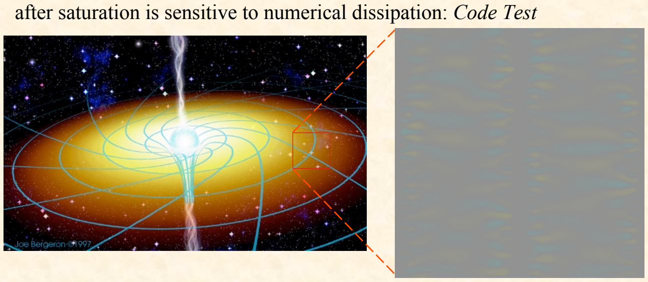


Saturation energy very similar.

2D MRI in shearing sheet; no-net-flux

Start from a vertical field with zero net flux in 2D: $B_z=B_0\sin(2\pi x)$

Sustained turbulence not possible in 2D – dissipation rate after saturation is sensitive to numerical dissipation: *Code Test*

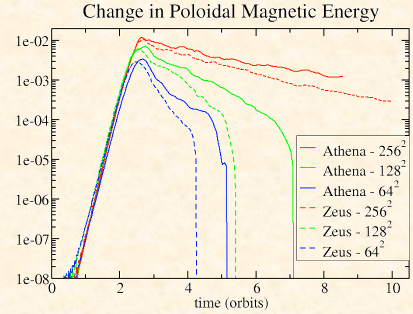


Animation of angular velocity fluctuations: $\delta V_y=V_y+1.5\Omega_0 x$

CTU with 3rd order reconstruction, 256^2 grid, $\beta_{min}=4000$, orbits 2-10

Magnetic Energy Evolution in 2D MRI with no-net-flux.

Change in Poloidal Magnetic Energy



Numerical dissipation is ~1.5 times smaller with CTU & 3rd order reconstruction than ZEUS.

Is ZEUS still useful in the era of AMR Godunov (better?) methods?

Advantages of ZEUS
1. Robust.
2. Very fast.
3. Easy to extend self-consistently with additional physics. (Most physics extensions to higher-order Godunov schemes destroy the formal accuracy of the method).
4. Cartesian, cylindrical, and spherical grids.

Disadvantages of ZEUS
1. Finite-differencing for pressure source terms gives poor dispersion relation for compressive waves.
2. Does not use conservative form (this is easily fixed).
3. No AMR (but in some cases non-uniform grid is better).

Other algorithms for MHD:

1) Finite-differencing with hyper-viscosity.
   e.g. Pencil code
2) Central schemes
   e.g. Lax-Wendroff, WENO
3) Finite-volume schemes
   e.g. Godunov methods
4) Spectral methods
5) Lattice Boltzmann methods
   e.g. Proteus code
6) MHD SPH

Some of these methods are implemented in public codes.

High-order finite-difference with hyperviscosity
- PENCIL

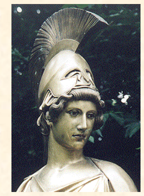
Operator split with artificial viscosity
- ZEUS
- Nirvana-1

Godunov
- Athena
- RAMSES
- PLUTO
- VAC
- BATS-R-US
- FLASH-3

Central schemes
- Nirvana-2

We'll focus on one: Athena

Introduction to Athena: discretization



Takes a completely different approach to solving the MHD equations from operator split methods like ZEUS.

Use a single-step, finite volume discretization of the conservative form of the equations.

Many other codes also take this approach.

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

Motivation for improved methods

Global model of *geometrically thin* (H/R << 1) disk covering 10H in R, 10H in Z, and 2π in azimuth with resolution of shearing box (128 grid points/H) will require nested grids.

Nested (and adaptive) grids work best with single-step Eulerian methods based on the conservative form

Algorithms in ZEUS are 20+ years old - a new code could take advantage of developments in numerical MHD since then.

Our Choice: higher-order Godunov methods combined with CT

Why adopt Godunov methods?

1) Very good for shock capturing and discontinuities (contact waves, slip-surface, current sheet).
2) Can be made strictly conservative, which is necessary for static and adaptive mesh refinement.
3) Best to conserve total energy to study turbulent dissipation and heating.
4) Do not *require* complex and expensive Riemann solvers.

Dai & Woodward 1994; 1998; Zachary, Malagoli & Colella 1994; Ryu, Jones, & Frank 1995; 1998; Balsara 1998; Falle, Komissarov & Joarder 1998; Powell et al. 1999; Balsara & Spicer 1999; Toth 2002; Dedner et al. 2002; Crockett et al 2003; Pen, Arras & Wong 2003; Londrillo & Del Zanna 2004; Ziegler 2005; Fromang, Hennebelle, & Teyssier 2006; Mignone et al. 2007

Athena solve the equations of ideal MHD in conservative form

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

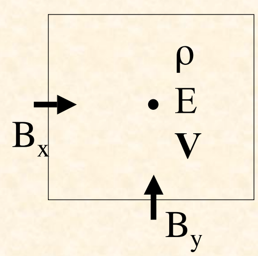
$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot [\rho \mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B} + P^*] = 0$$

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

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

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 "area averaged" fluxes.

Finite-Area discretization of the induction equation.

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

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

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

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

Again, these equations are exact -- no approximation has been made.

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

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

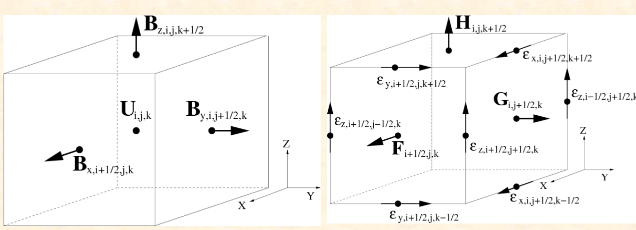
$$\mathcal{E}^{n+1/2}_{y,i-1/2,j,k-1/2} = \frac{1}{\delta y \delta t}\int_{t^n}^{t^{n+1}}\int_{y_{j-1/2}}^{y_{j+1/2}} \mathcal{E}_y(x_{i-1/2},y,z_{k-1/2},t)\ \mathrm{d}y\ \mathrm{d}t$$

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

are "line averaged" electro-motive forces (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!

Summary

- A simple, robust, easy-to-understand algorithm for MHD can be built on operator splitting.
- Various versions of ZEUS code implement this algorithm.
- In comparison to Athena, ZEUS is somewhat more dissipative.
- ZEUS is still useful. But Athena is the future.