Multiphysics, Multigrid, and More about AMR (M^3)

Ann Almgren

Center for Computational Sciences and Engineering
Lawrence Berkeley National Laboratory
asalmgren@lbl.gov

July 25, 2011

Hydrodynamic Equations

Recall our basic hydrodynamic equations

$$\begin{array}{rl}
\textbf{Mass} & \rho_t \quad\quad + \nabla \cdot \rho U = 0 \\
\textbf{Momentum} & (\rho U)_t \ + \nabla \cdot (\rho U U + p) = \rho \vec{g} \\
\textbf{Energy} & (\rho E)_t \ + \nabla \cdot (\rho U E + p U) = \nabla \cdot \kappa \nabla T \\
\textbf{Species} & (\rho X_m)_t + \nabla \cdot (\rho U X_m) = \dot{\omega}_m
\end{array}$$

Augmented (“closed”) with

- EOS = Thermodynamics, i.e. how to compute p from ρ, T, X
- Network = Reaction kinetics, i.e. how to compute $\dot{\omega}_m$
- Gravity – we’ll come back to this...

Conservation Form

Recall also that we could write this in conservation form with source terms:

$$U_{i,j}^{n+1} = U_{i,j}^{n} - \frac{\Delta t}{\Delta x}(F_{i+1/2,j}^{n+\frac{1}{2}} - F_{i-1/2,j}^{n+\frac{1}{2}}) - \frac{\Delta t}{\Delta y}(G_{i,j+1/2}^{n+\frac{1}{2}} - G_{i,j-1/2}^{n+\frac{1}{2}})$$
$$+ \Delta t \; S_{i,j}^{n+\frac{1}{2}}$$

Because this is a purely hyperbolic system (with source terms),

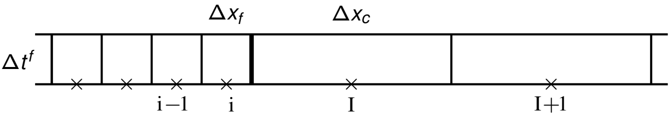
- We can solve it with an explicit method
- Signals move at speeds U, $U + c$, $U - c$,
- If we limit $\Delta t < \Delta x / max(|U| + c)$, then information never travels more than one cell in a time step

Review of AMR for Hyperbolic System–1d

Consider $U_t + F_x = 0$ discretized with an explicit finite difference scheme:

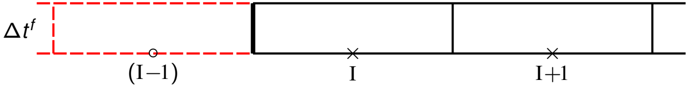
$$U_{i,j}^{n+1} = U_{i,j}^{n} - \frac{\Delta t}{\Delta x}(F_{i+1/2,j}^{n+\frac{1}{2}} - F_{i-1/2,j}^{n+\frac{1}{2}})$$

In order to advance the composite solution we must compute the fluxes consistently:

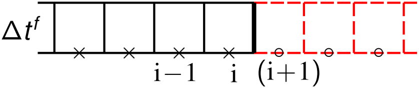


Hyperbolic–composite advance

One can advance the coarse grid



then advance the fine grid



using “ghost cell data” at the fine level interpolated from the coarse grid data.

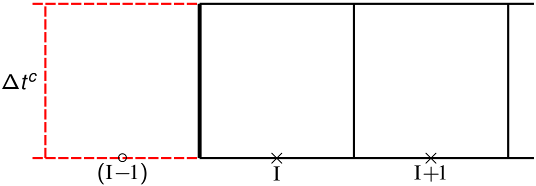
This results in a flux mismatch at the coarse/fine interface, which creates an error in U_I^{n+1}. The error can be corrected by refluxing, i.e. setting

$$\Delta x_c U_{\mathrm{I}}^{n+1} := \Delta x_c U_{\mathrm{I}}^{n+1} - \Delta t^f F^c_{\mathrm{I}-1/2} + \Delta t^f F^f_{i+1/2}$$

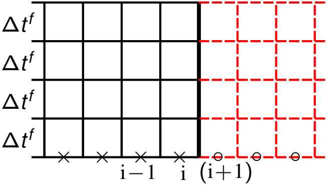
Before the next step average fine grid solution onto coarse grid.

Hyperbolic–subcycling

To subcycle in time we advance the coarse grid with Δt^c



and advance the fine grid multiple times with Δt^f.



The refluxing correction now must be summed over the fine grid time steps:

$$\Delta x_c U_{\mathrm{I}}^{n+1} := \Delta x_c U_{\mathrm{I}}^{n+1} - \Delta t^c F^c_{\mathrm{I}-1/2} + \sum \Delta t^f F^f_{i+1/2}$$

End of story?

No ... let's talk about gravity ...

Gravity (Non-relativistic)

Possible representations – relevance depends on length scale:

- On very small scales, gravity can be viewed as negligible
- On slightly larger scales, gravity can be viewed as constant (in direction and magnitude)
- On even larger scales, we must compute the gravitational field given the mass distribution as a function of space and time – and the mass in one place affects the gravity everywhere else in the domain

Self-Gravity

Can represent mass as

- discrete particles / blobs
- continuous distribution ($\rho(\mathbf{x}, t)$)

Different solution methods include

- particle methods (e.g. $O(N^2)$, multipole methods)
- grid-based methods (i.g., solve $\nabla^2\phi = 4\pi G\rho$ on a grid)
- hybrid methods (e.g., MLC = grid method with particle correction)

Note that we can represent mass as particles but solve for gravity on a grid (e.g. dark matter in cosmological simulations), or we can think of mass as a density distribution on a grid but solve with a particle-type method (e.g. multipole method).

Monopole Approximation

Monopole approximation

- Suitable for mass distribution that is basically spherically symmetric

- Easy to calculate
  - Compute 1-d radial array representing $\rho(r)$
  - Compute enclosed mass $M_{\rm encl}(r) = \int 4\pi(r')^2\rho(r')dr'$
  - Define $g(r) = GM_{\rm encl}/r^2$ and interpolate onto grid

- Note that this is explicit but requires global communication to create $\rho(r)$

- This approach can be extended to much more generality in multipole method

Self-Gravity

For a completely general mass distribution on a grid, we can solve

$$\nabla^2 \phi = -4\pi G \rho(\mathbf{x}, t)$$

Solving this equation

- is inherently non-local
- requires some kind of solver for sparse linear system
  - direct - great for small systems
  - iterative - more efficient for larger systems

Boundary conditions

- periodic
- Neumann, Dirichlet
- free-space / isolated
  - from the monopole approximation
  - using James' method

Iterative Solvers

If you replace

$$\nabla^2 \phi = 0 \qquad (1)$$

by

$$\phi_t = \nabla^2 \phi \qquad (2)$$

and advance (2) forward in pseudo-time until you reach a steady state, the steady state will satisfy (1).

The process of advancing the solution to (2) in pseudo-time is called ”relaxing” or ”smoothing”.

Standard relaxation methods:

Jacobi: $\phi_{i,j}^{k+1} = L(\phi^k)_{i,j}$

Gauss-Seidel: $\phi_{i,j}^{k+1} = L(\phi^k, \phi^{k+1})_{i,j}$

Multigrid Method

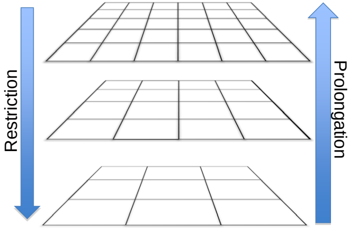
Observation (1) – these smoothers are very efficient at reducing the short-wavelength error, but are very slow to reduce the long-wavelength error.

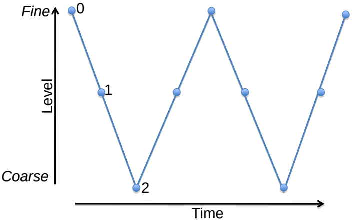
Observation (2) – “wavelength” is relative to the grid spacing – what is long-wavelength at one grid spacing is short-wavelength on a much coarser grid.

Basic multigrid algorithm

- Relax on fine grid
- Coarsen the error onto coarser grid (“restriction”)
- Relax on coarser grid
- Interpolate correction back to fine grid (“prolongation”)

Multigrid Method





(Figures from John Shalf)

Multigrid with AMR

This starts to sound suspiciously like AMR.

So how do we solve $\nabla^2 \phi = -4\pi G \rho(\mathbf{x}, t)$ with AMR?

Need to differentiate between “AMR levels” and “multigrid levels”

- No coarse-fine boundaries between multigrid levels – coarse grids all lie directly ”below” fine grids
- AMR levels contain the solution throughout the simulation
- Multigrid levels are *only* used for solving the linear system, then thrown away

Multigrid with AMR (p2)

But ... the concepts in multigrid are very similar to those in AMR:

- Level operations – relaxation (mg) vs solution advance (AMR)
- Inter-level operations – prolongation and restriction
- However, synchronization procedures due to coarse/fine mismatches occur only between AMR levels

Multigrid with AMR (p3)

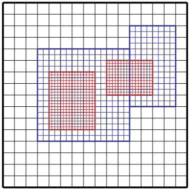
Two types of solves:

- MultiLevel solve: solve for all AMR levels together – e.g., in FLASH, with no subcycling
- Level solve: solve on each AMR level separately, using boundary conditions from coarser AMR level

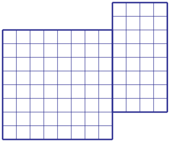
For now we will focus only on the level solves, and look at how we synchronize the solution in 1-d.

Multigrid for Level Solves

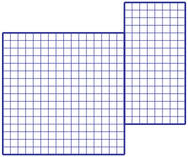
Suppose we want to solve on the level 1 (blue) grids only.



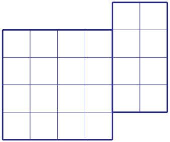
Original Grid Hierarchy



AMR Level 1 / Multigrid Level 1



AMR Level 1 / Multigrid Level 0



AMR Level 1 / Multigrid Level 2

Elliptic AMR

Recall the AMR time advance with subcycling:

- Advance coarse level by Δt^c
    - Advance fine level by Δt^f
    - Advance fine level by Δt^f
- Synchronize levels

Now, suppose we include in each “Advance” the solution of an elliptic equation, i.e. for gravity.

During the synchronization step we must also synchronize the elliptic contribution.

1-D Example: Elliptic AMR

Look at 1d example

$$-\phi_{xx} = \rho$$

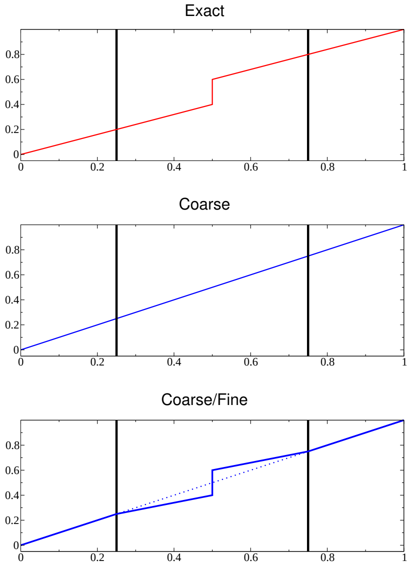
where ρ is a discrete approximation to the derivative of a δ function at the center of the domain

$$\rho^f_J = -\alpha \qquad \rho^f_{J+1} = \alpha$$

but $\rho^c \equiv 0$

Notice that the coarse and fine solutions match but their gradients do not.

In other words, we have enforced a Dirichlet (but not Neumann) condition.



1-D Example: Elliptic AMR (p2)

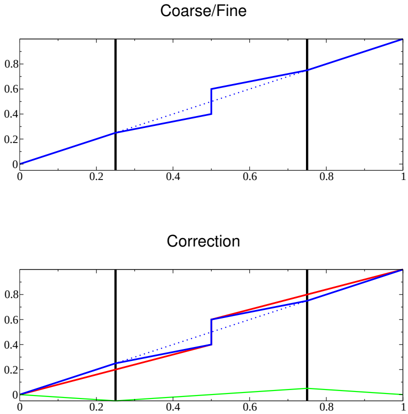
How do we correct the solution during the synchronization?

Similarly to the hyperbolic synchronization, we define the flux mismatch between coarse and fine levels.

Here the flux is in the form of the normal derivative of ϕ at the coarse-fine interface.

Big difference is that the “refluxing” requires the solution of a correction equation, $Le = R$, which is also elliptic.

- Residual is localized to the $c - f$ boundary but correction is global
- The error equation is a discrete layer potential problem
- e is a discrete harmonic function on the fine grid $\rightarrow$ solve only on coarse grid and interpolate


Coarse/Fine

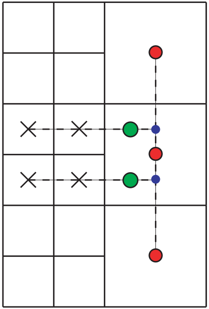
Correction

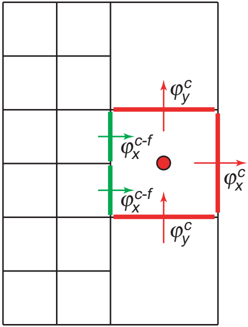
Coarse/Fine (blue) + Correction (green) = Exact (red)

Extension to Multi-D

When we extend to multiple space dimensions, we must now interpolate tangentially as well as in the normal direction.

Simple interpolation formulae that we used for the hyperbolic part are not sufficiently accurate for second-order operators.





Note that we do not pre-compute the green circle as a boundary value – why?

Summary: Multiphysics AMR

Basic integration paradigm works for hyperbolic, elliptic and parabolic PDEs

Synchronization equations match the structure of the process being corrected.

Combine these elements to make a number of different codes

- Add radiation solver to basic hydro code (e.g., MGFLD in CASTRO)
- Cosmology (hydro + dark matter)
- Low Mach number model

Key issue is keeping tracking of different aspects of synchronization and performing them in the right order