numflux etc...

& geometric source terms

Updating a grid

AMR

- advance
  - b4step – initializes grid if necessary and protects pressure and density etc...
  - **src** – updates domain with source terms
  - **step** – calculates the physical fluxes and updates the conserved quantities except at amr boundaries
- fixup – updates quantities at amr boudaries
- afterfixup_MHD
  - *computes_emf* – calculates the emf's for CT update
  - field_update – updates the cell centered B-fields and energy.

step

- Step cycles through every ray in the grid and updates each ray of cells
  - **numflux** – calculates the numerical fluxes at the cell interfaces for each ray (numFql & numFqr)
  - updateq – applies those fluxes to update the cell centered values along each ray
    - UpdateFixups – stores the numerical fluxes that are transverse to the edge of the domain.
    - CoarseFineFlux – Prevents using fluxes when refined fluxes will later be available.

numflux

- numflux only calls two main routines
  - physflux – does the physics calculations requested by numflux
  - src1D – does a source integration on a ray of cells
- numflux does essentially three things:
  - spatial interpolation of cell edges (method[2])
  - time evolution of cell edges (method[4] *& method[5]*)
  - flux calculation at cell edges (method[6])

method(2)

- Spatial interpolation – All 5 interpolation schemes leave the cell centers in primitive form and all but the Gudonov method leave the cell edges in conservative form. They use the following requests to physflux:
    - RequestPrimitive
    - RequestConserved
    - RequestEigenDecomposition (used only by CASE 2 below)
- Select Case Method(2)
    - CASE 0: Gudonov method – no interpolation
    - CASE 1: Linear interpolation on primitive variables
    - CASE 2: Linear characteristic interpolation – uses the Roe-averaged eigen-decomposition of the wave equation to interpolate each wave mode.
    - CASE 3: Piecewise Parabolic Reconstruction of primitive fields
    - CASE 4: Local Hyberbolic Harmonic Reconstruction of primitive fields

method(4)

- Time evolution – All of the different evolution schemes leave the edge values (q1DL and q1DR) in conservative form. When method(5) /= 0, CASES 1, 3, & 4 call src1D on the cell edges. All of the different cases use the following requests to physflux:
    - RequestPrimitive
    - RequestConserved
    - RequestSidedEigenDecomposition (used by CASES 3 & 4 below)
    - RequestFluxes
    - RequestPredictor (used only by CASE 1, 3, & 4 for MHD)
- Select Case Method(4)
    - CASE 0: Gudonov method – no interpolation
    - CASE 1: Uses the reconstructed edge values to calculate predictor fluxes to update the edge values. (Not a Riemann solve) (MUSCL-Hancock when used with method(2)=1)
    - CASE 2: 2-Step Runge-Kutta (implemented in advance)
    - CASE 3: PPM Characteristic Tracing *(should only be used with method(2)=3)*
    - CASE 4: Linear Characteristic Tracing

method(6)

- Flux Calculation – Solves the Riemann problem at the cell edges. Uses the following requests to physflux:
    - RequestFluxes
    - RequestSpeeds
    - RequestEigenDecomposition
    - RequestHLLDFlux
    - RequestSidedEigenDecomposition
- Select Case Method(6)
    - CASE 0: Roe method
    - CASE 1: Adapted Marquina flux formula
    - CASE 2: Marquina flux formula
    - CASE 3: HLLD solver

method(5)

- Source integration:
    - RequestPrimitive
    - RequestConserved
    - RequestSidedEigenDecomposition (used by CASES 3 & 4 below)
    - RequestFluxes
    - RequestPredictor (used only by CASE 3 below)
- IF Method(4) == 2 and Method(5) != 0, then there is a source step, hydro step, source step, hydro step. Otherwise ...
- Select Case Method(5)
    - CASE 0: No source updating
    - CASE 1: Calls src1D on the reconstructed edges and a full source step in afterfixup
    - CASE 2: Does a half source update, reconstructs, does a src1D update on edge values, updates the grid and calls another half source update. (Strang Splitting)

src

- source routine calculates the various source terms and the jacobian matrix:

$$S_a = \frac{dQ_a}{dt}$$

$$J_{ab} = \frac{dS_a}{dQ_b}$$

Uniform Gravity

$$Q = [\rho, p_x, p_y, E]$$

$$S = \begin{bmatrix} 0 \\ 0 \\ -\rho g \\ -g \rho v_y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \rho \\ -g p_y \end{bmatrix}$$

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -g & 0 & 0 & 0 \\ 0 & 0 & -g & 0 \end{bmatrix}$$

2.5D Geometric Source Terms

$$Q = [\rho, p_r, p_z, p_\phi, E]$$

$$S = \frac{-1}{r} \left(\begin{matrix} \rho u_r \\ \rho u_r^2 - \rho u_\phi^2 \\ \rho u_r u_z \\ 2 \rho u_r u_\phi \\ u_r [E + P(Q)] \end{matrix} \quad = \quad \begin{matrix} p_r \\ \frac{p_r^2}{\rho} \\ \frac{p_r p_z}{\rho} \\ \frac{2 p_r p_\phi}{\rho} \\ \frac{p_r}{\rho} [E + P(Q)] \end{matrix} \right) = \frac{-1}{r} F_r + \begin{bmatrix} 0 \\ \frac{v_\phi^2}{r} \\ 0 \\ -\rho u_r u_\phi \\ 0 \end{bmatrix}$$

$$J = \frac{-1}{r} \left(\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{-p_r^2 + p_\phi^2}{\rho^2} & \frac{2 p_r}{\rho} & 0 & \frac{-2 p_\phi}{\rho} & 0 \\ \frac{-p_r p_z}{\rho^2} & \frac{p_z}{\rho} & \frac{p_r}{\rho} & 0 & 0 \\ \frac{-2 p_r p_\phi}{\rho^2} & \frac{2 p_\phi}{\rho} & 0 & \frac{2 p_r}{\rho} & 0 \\ \frac{-p_r}{\rho^2}[E + P(Q)] & \frac{E + P(Q)}{\rho} & 0 & 0 & \frac{p_r}{\rho} \end{bmatrix} + \frac{p_r}{\rho} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{dP}{d}\rho & \frac{dP}{dp_r} & \frac{dP}{dp_z} & \frac{dP}{dp_\phi} & \frac{dP}{dE} \end{bmatrix} \right)$$

Geometric source terms

- We begin with some coordinates q^i and the conservation equations in vector form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0$$

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \boldsymbol{u}) + \nabla P = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [\boldsymbol{u}(E+P)] = 0$$

$$\nabla \cdot \boldsymbol{V} = V^i_{;i} = \frac{\partial V^i}{\partial q^i} + V^k \Gamma^i_{ki}$$

$$\nabla \boldsymbol{V} = V^i_{;j} = \frac{\partial V^i}{\partial q^j} + V^k \Gamma^i_{kj}$$

$$(\nabla P)^i = P^{ij}_{;j}$$

- Replacing derivatives with covariant derivatives and replacing the pressure with a 2nd rank contravariant tensor introduces geometric source terms:

$$P^{ij} = P g^{ij}$$

$$P^{ij}_{;j} = \frac{\partial g^{ij} P}{\partial q^j} + P g^{im} \Gamma^j_{jm} + P g^{jm} \Gamma^i_{jm}$$

$$\begin{aligned}
S_\rho &= -\rho u^k \Gamma^j_{kj} \\
S_{p_i} &= -\rho u^i u^k \Gamma^j_{kj} - \rho u^j u^k \Gamma^i_{kj} - P g^{ik} \Gamma^j_{jk} - P g^{jk} \Gamma^i_{jk} \\
S_E &= -(E+P) u^k \Gamma^j_{kj}
\end{aligned}$$

Cylindrical Coordinates

- There are only three non-zero coefficients of connection (Christoffel symbols)

$$\Gamma^r_{\phi\phi} = -r \quad \Gamma^\phi_{r\phi} = \frac{1}{r} \quad \Gamma^\phi_{\phi r} = \frac{1}{r} \qquad g_{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad g^{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{-1}{r^2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

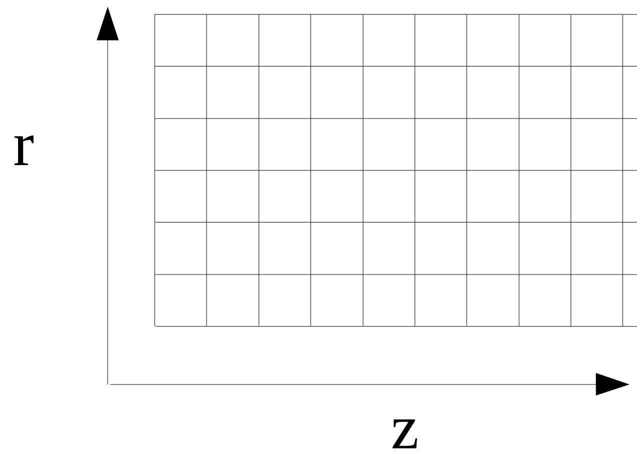
- Plugging these into the source terms gives:

$$\begin{bmatrix} \frac{-\rho u_r}{r} & + & 0 \\ \rho u_r^2 r & + & \rho u_\phi^2 r \\ \frac{-\rho u_\phi u_r}{r} & + & \frac{-2\rho u_\phi u_r}{r} \\ 0 & + & 0 \\ \frac{-u_r (E+P)}{r} & + & 0 \end{bmatrix} = \frac{-F_r}{r} + \begin{bmatrix} 0 \\ \rho u_\phi^2 r \\ \frac{-2\rho u_\phi u_r}{r} \\ 0 \\ 0 \end{bmatrix}$$

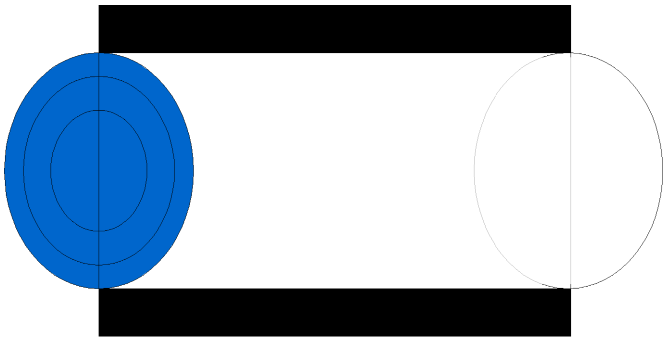
- Note: $P g^{ik} \Gamma^j_{jk} + P g^{jk} \Gamma^i_{jk} = 0$

Alternatively...

- Instead of storing the actual densities, store the projected surface densities for each (r-z annulus)



$$\rho \Rightarrow r\,\rho$$
$$\rho\,\boldsymbol{u} \Rightarrow r\,\rho\,\boldsymbol{u}$$
$$E \Rightarrow r\,E$$
$$P \Rightarrow r\,P$$
$$\frac{P}{\rho} \Rightarrow \frac{P}{\rho}$$



- Then source terms associated with the radial flux go away...

Modified geometric source terms

- We can eliminate many of the source terms associated with the advection of densities if we scale the densities of mass, momentum, and energy by the metric scale factor

$$g^{\frac{1}{2}} = \frac{d\boldsymbol{\tau}}{dq^1 dq^2 dq^3 \ldots}$$

- And use the identity $g^{\frac{1}{2}} \Gamma^j_{kj} = \frac{d\left(g^{\frac{1}{2}}\right)}{dq^k}$

- Then the source terms simplify to:

$$\begin{aligned} S_\rho &= 0 \\ S_{p_i} &= -\rho u^j u^k \Gamma^i_{kj} - P g^{jk} \Gamma^i_{jk} \\ S_E &= 0 \end{aligned}$$

$$\begin{bmatrix} 0 \\ \rho u_\phi^2 r \\ \frac{-2\rho u_\phi u_r}{r} \\ 0 \\ 0 \end{bmatrix}$$

- Note also that AstroBear does this for the angular momentum in order to store the magnitude of the angular momentum density.

$q(\text{iAngMom}) = \rho v_\phi$ instead of $\rho\omega$