Numerical Methods for MHD:

Relaxation Methods – Time Evolution: Upwind scheme and Flux Limiters

Konstantinos Gourgouliatos

5th HelAS Summer School, Ioannina, 16-20 September 2024

Equilibrium solution

- **Solution of Partial Differential Equations:**

Laplacian: $\nabla^2 U = \partial_{xx} U + \partial_{yy} U$

Poisson Equation: $\nabla^2 U = S(x, y)$

Diffusion Equation: $\partial_t U = \partial_{xx} U + \partial_{yy} U - S$

Iterative Method

- Convert a time-independent equation to a time-dependent one for and solve it as a diffusion equation. The solution diffuses from the boundaries to the center:

$$\partial_t U = \partial_{xx} U + \partial_{yy} U - S$$

$$\frac{U^{n+1}_{i,j} - U^{n}_{i,j}}{\Delta t} = \frac{U^{n}_{i+1,j} - 2U^{n}_{i,j} + U^{n}_{i-1,j}}{(\Delta x)^2} + \frac{U^{n}_{i,j+1} - 2U^{n}_{i,j} + U^{n}_{i,j-1}}{(\Delta y)^2} - S_{i,j}$$

i, j -> spatial discretization, n temporal discretization.

1st Activity

- Solution of the Laplacian $\nabla^2 U = 0$.

  - Go to the following link: Laplacian Solution.
  - Experiment with different boundary conditions by changing lines 43, 44, 50, 51.
  - Change the iterations (line 10) and the resolution (12, 13).

The codes are not written in the optimal, and most efficient, python style, but it is attempted to make them as understandable as possible.

Force-free Equilibrium I

- Force-free problem:

$$\vec{j} \times \vec{B} = \vec{0} = (\nabla \times \vec{B}) \times \vec{B}$$

- Express the magnetic field in cylindrical coordinates (i.e. useful for an axisymmetric jet)

$$\vec{B} = \vec{\nabla}\Psi(R,z) \times \vec{\nabla}\phi + I(R,z)\vec{\nabla}\phi$$

$$\vec{B} = -\frac{1}{R}\frac{\partial \Psi}{\partial z}\hat{R} + \frac{I(R,z)}{R}\hat{\phi} + \frac{1}{R}\frac{\partial \Psi}{\partial R}\hat{z}$$

$$\vec{j} = \frac{c}{4\pi}\left[-\frac{1}{R}\frac{\partial I}{\partial z}\hat{R} - \left(\frac{1}{R}\frac{\partial^2 \Psi}{\partial z^2} - \frac{1}{R^2}\frac{\partial \Psi}{\partial R} + \frac{1}{R}\frac{\partial^2 \Psi}{\partial R^2}\right)\hat{\phi} + \frac{1}{R}\frac{\partial I}{\partial R}\hat{z}\right]$$

Force-free Equilibrium II

$$\boxed{\frac{\partial^2\Psi}{\partial R^2}+\frac{\partial^2\Psi}{\partial z^2}-\frac{1}{R}\frac{\partial\Psi}{\partial R}=-II'(\Psi)}$$

Lower boundary condition

$$B_R=0 \qquad B_z=B_0$$

$$B_\phi=\frac{B_1R_0}{R}\sin\left(\frac{\pi R^2}{R_{max}^2}\right)$$

$$I=B_1R_0\sin\left(\frac{2\pi\Psi}{B_0R_{max}^2}\right)$$

$$I'(\Psi)=\frac{2\pi}{B_0R_{max}^2}B_1R_0\cos\left(\frac{2\pi\Psi}{B_0R_{max}^2}\right)$$

Central difference derivatives:

$$\frac{\partial\Psi}{\partial R}=\frac{\Psi(i+1,j)-\Psi(i-1,j)}{2\Delta R}$$

$$\frac{\partial\Psi}{\partial z}=\frac{\Psi(i,j+1)-\Psi(i,j-1)}{2\Delta z}$$

$$\frac{\partial^2\Psi}{\partial R^2}=\frac{\Psi(i+1,j)-2\Psi(i,j)+\Psi(i-1,j)}{\Delta R^2}$$

$$\frac{\partial^2\Psi}{\partial z^2}=\frac{\Psi(i,j+1)-2\Psi(i,j)+\Psi(i,j-1)}{\Delta z^2}$$

Left, right and top boundary condition

$B_R=0$

2nd Activity

- Solution of a force-free equilibrium
  - Go to the following link: Force-free.
  - Discuss the code and see how the derivatives and the boundary conditions are implemented.
  - Experiment with different strengths of the azimuthal field by changing B_1 in line 31 by setting it equal to 0, 5, 10.

Advection Equation I

- In many physical system the advection equation appears:

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) = 0$$

$$\boxed{\partial_t \boldsymbol{v} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\, \boldsymbol{v}} = -\frac{\boldsymbol{\nabla} p}{\rho} + \boldsymbol{g} + \boxed{\eta \boldsymbol{\nabla}^2 \boldsymbol{v}} \qquad \boxed{\partial_t v_x = -v_x \partial_x v_x + \eta \partial_{xx} v_x}$$

$$\partial_t e + \boldsymbol{\nabla} \cdot (\boldsymbol{v}\,[e + p]) = 0$$

- This example includes a non-linear term where the velocity multiplies its spatial derivative, so a steep maximum will move faster, than a shallow one (Burger’s equation).

Advection Equation II

- Hall evolution (see the neutron star crust configuration):

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left[\frac{c}{4\pi n_e e} (\nabla \times \mathbf{B}) \times \mathbf{B} + \frac{c^2}{4\pi\sigma} \nabla \times \mathbf{B} \right]$$

- Consider a plane-parallel model:

$$\frac{\partial B_y}{\partial t} = \frac{-c}{4\pi e} B_y \left(\nabla n_e^{-1} \times \hat{\mathbf{y}} \right) \cdot \nabla B_y + \frac{c^2}{4\pi\sigma} \nabla^2 B_y$$

- Simplifying it in 1-D (B_y -> b) it corresponds to a field being transported in x direction, and normalize appropriately:

$$\partial_t b + b\partial_x b = \eta \partial_{xx} b$$

- Does the central difference scheme we used before gives acceptable results for this equation?
- Information travels from left to right for positive b, the central difference scheme uses information from points where the disturbance has not arrived yet!
- Forward difference: $\partial_{x+} b = \frac{b_{i+1} - b_i}{\delta x}$, Backward difference $\partial_{x-} b = \frac{b_i - b_{i-1}}{\delta x}$

3rd Activity

- Solution of the advection equation using an upwind scheme: central – forward backward derivative.
  - Go to the following link: Upwind.
  - Discuss the code and see how the derivatives and the boundary conditions are implemented.
  - Experiment running the code using different schemes.
  - Experiment by initialisating the field using a negative value.

Flux limiter I

- Solution of PDEs generate spurious oscillations, i.e. the central difference scheme and even in upwind schemes for appropriate profiles i.e. what happens if a discontinuity appears?
- A more efficient way to tackle this is through flux limiters, that approximate the derivatives to realistic values.
- We can write PDEs in conservative form: $\partial_t u + \partial_x F = 0$, i.e. the linear case is $F = au$, and the case we solved in activity 3 corresponds to $F = \frac{u^2}{2}$.
- The general solution through Flux limiters is the following: $u_j^{n+1} = u_j^n + \frac{\Delta t}{\Delta x}\left(F_{j-1/2} - F_{j+1/2}\right)$

$$F_{j-1/2} = F_l(j-1/2) + \phi(r_{j-1/2})\left[F_h(j-1/2) - F_l(j-1/2)\right]$$
$$F_{j+1/2} = F_l(j+1/2) + \phi(r_{j+1/2})\left[F_h(j+1/2) - F_l(j+1/2)\right]$$

$\phi(r)$ is the flux limiter function and *r* is a measure of the smoothness of the function:

$$r_{j-1/2} = \frac{u_{j-1} - u_{j-2}}{u_j - u_{j-1}}, \qquad r_{j+1/2} = \frac{u_j - u_{j-1}}{u_{j+1} - u_j}$$

Flux limiter II

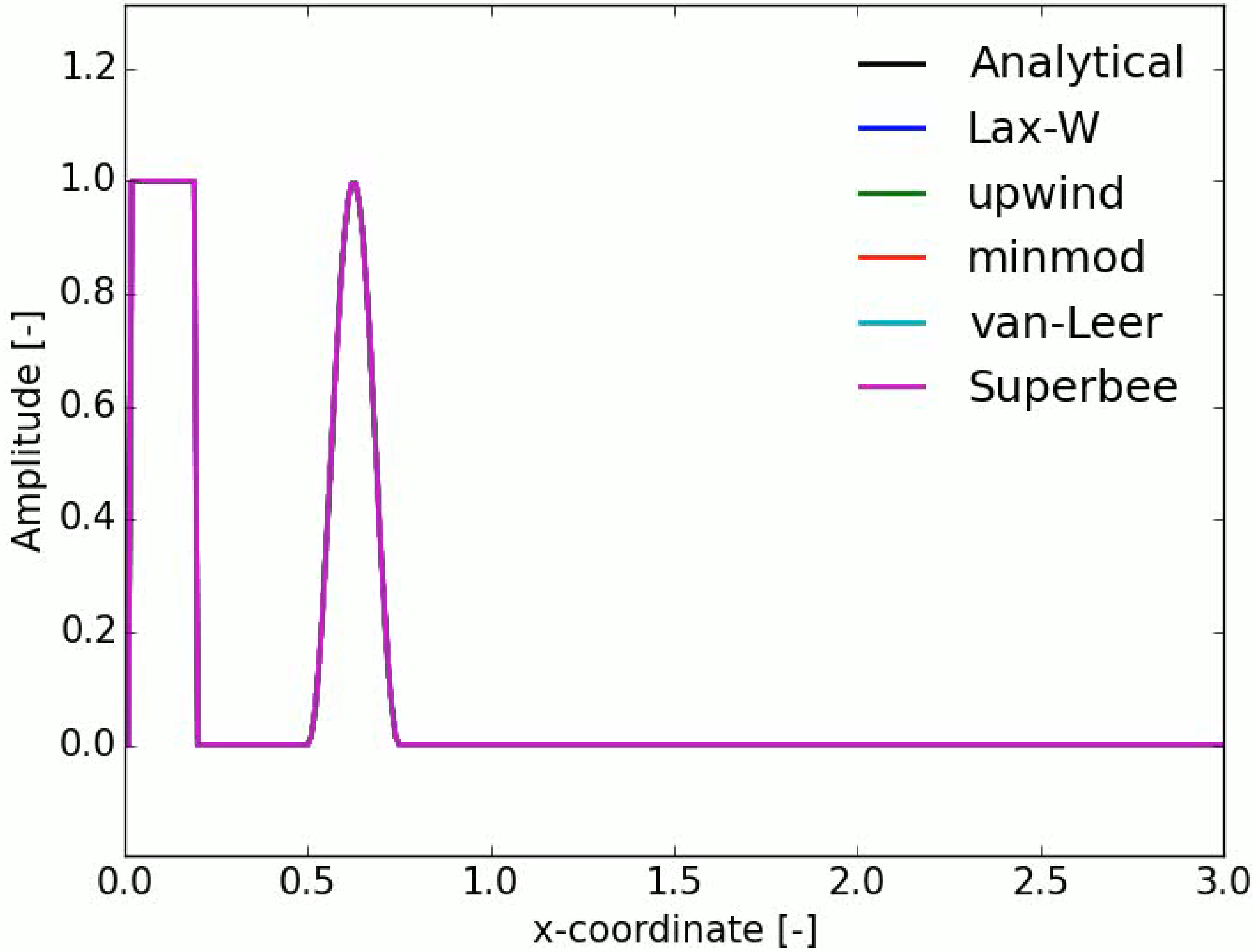
- We choose a functional form for $\phi(r)$ and implement the solution.
- Example: Lax-Wendroff:

$$u^{n+1/2}_{j+1/2} = \frac{1}{2}\left(u^n_{j+1} + u^n_j\right) - \frac{\Delta t}{2\Delta x}\left(F(u^n_{j+1}) - F(u^n_j)\right)$$
$$u^{n+1/2}_{j-1/2} = \frac{1}{2}\left(u^n_{j} + u^n_{j-1}\right) - \frac{\Delta t}{2\Delta x}\left(F(u^n_{j}) - F(u^n_{j-1})\right)$$

$$u^{n+1}_j = u^n_j - \frac{\Delta t}{\Delta x}\left(F(u^{n+1/2}_{j+1/2}) - F(u^{n+1/2}_{j-1/2})\right)$$

- Minmod: $\phi_{mm}(r) = \max\left[0, \min\left(1, r\right)\right]; \quad \lim_{r\to\infty} \phi_{mm}(r) = 1.$

- Koren: $\phi_{kn}(r) = \max\left[0, \min\left(2r, \min\left(\frac{(1+2r)}{3}, 2\right)\right)\right]; \quad \lim_{r\to\infty} \phi_{kn}(r) = 2$


Analytical
Lax-W
upwind
minmod
van-Leer
Superbee
Amplitude [-]
x-coordinate [-]
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

AMR – VAC

- The features we discussed before are implemented in various codes.
- One can choose what type of integration method, flux limiter and various other parameters can use.
- Examples: PLUTO, AMR-VAC and several others.

Proposed Project

- Modify appropriately the Upwind code to implement a flux limiter of your choice (you may start from minmod) and integrate Burger's equation – experiment with $\eta=0$.
- Discuss the advantages and disadvantages of each choice.

- Use of Chat GPT is acceptable provided you understand what is happening in your code ☺ !