

Mathematical Institute
University of Cologne

August 14, 2017

Numerical techniques in MHD simulations Pre-conference workshop

Session 1: Monday, August 14, 14:00-18:00

PC Pools 1 & 2 (Mathematical Institute, souterrain, rooms -117 & -118)

coffee break: 15:30-16:15 (Mathematical Institute, ground floor, room 005)

Session 2: Tuesday, August 15, 09:00-12:30

PC Pools 1 & 2 (Mathematical Institute, souterrain, rooms -117 & -118)

coffee break: 10:30-11:00 (Mathematical Institute, ground floor, room 005)

Session 3: Tuesday, August 15, 14:00-18:00

PC Pools 1 & 2 (Mathematical Institute, souterrain, rooms -117 & -118)

coffee break: 15:30-16:15 (Mathematical Institute, ground floor, room 005)

The goal of this project is the implementation of a *finite volume scheme* to solve the two-dimensional ideal MHD system. We validate the implementation by convergence tests and extend it by a *GLM approach* to maintain the divergence-free condition of the magnetic field. Finally, we apply the method to simulate a magnetic blast wave. Furthermore, for experts, the scheme can be extended by a *reconstruction technique* to achieve a second order approximation and applied to the *Orszag-Tang vortex* or the *magnetic rotor* test case.

In order to guide the implementation process, the workshop is divided into single projects with suggested time limits. Additionally, the support team is always available regarding any questions. The necessary information concerning the numerical method as well as the MHD system dependent definitions can be found online in the presentation slides:

<http://www.mi.uni-koeln.de/NumSim/pre-conference-workshop>

user name: gassner

password: students

Project 1: Finite volume method for one-dimensional MHD

Estimated time limit: End of Session 1

Objective: Implementation of a first order finite volume method to solve the one-dimensional ideal MHD equations on a periodic domain, validation by convergence test, application to a shock-tube problem

Key elements:

- periodic domain $\Omega = [a, b] \subset \mathbb{R}$ divided into an uniform Cartesian grid with equidistant cells of size $\Delta x = \frac{b-a}{N}$
- data structure for solution array, e.g. $u(i,k)$ for i -th cell and k -th conserved variable
- first order finite volume method with local Lax-Friedrich solver at cell boundaries
- integration in time by explicit Euler method with stable CFL time step computation
- validation by convergence test
- application to a shock-tube problem

Possible code structure:

- `f = getAdvectiveFluxX` (computes the advective flux from the conserved solution `u`)
- `lambda = getMaxEigenvalueX` (computes the maximal eigenvalue of the flux-Jacobian at a particular solution state `u`)
- `F_LLX = RiemannfluxX` (computes the numerical flux at the cell interfaces from the left `u_L` and right `u_R` states)
- `deltat = computeTimestep` (returns the time step determined by the CFL condition)
- `integrateInTime` (updates the solution in time by the finite volume formula)
- `finiteVolumeMain` (contains mesh and parameter definitions, initialization, time loop and plotting routines)

Convergence test:

To verify the correctness of the implementation we consider the domain $\Omega = [0, 1]$ with periodic boundaries and the initial data

$$\rho_0 = 1, \quad \vec{v}_0 = \begin{pmatrix} 0 \\ 0.1 \sin(2\pi x) \\ 0.1 \cos(2\pi x) \end{pmatrix}, \quad p_0 = 0.1, \quad \vec{B}_0 = \begin{pmatrix} 1 \\ 0.1 \sin(2\pi x) \\ 0.1 \cos(2\pi x) \end{pmatrix}$$

The main advantages of this test case are, that the solution is smooth as well as periodic and, moreover, the velocity of the Alfvén-wave remains constant equal to one. Thus, the solution adopts the initial state for each time $t \in \mathbb{N}$. Run your code with this initial data and $\gamma = 5/3$. Set $CFL = 0.9$ and a final time $T = 1$ to compute the maximal errors

$$\epsilon_\nu = \|U - \tilde{U}_{N_\nu}\|_{L^\infty} = \max_{i=1, \dots, N_\nu} |U(x_i) - \tilde{U}_{N_\nu}(x_i)|$$

with respect to your approximations \tilde{U}_{N_ν} for $N_\nu = 10 \cdot 2^\nu, \nu = 0, \dots, 6$. Finally, we can compute the experimental order of convergence

$$EOC_\nu = \frac{\ln\left(\frac{\epsilon_\nu}{\epsilon_{\nu-1}}\right)}{\ln\left(\frac{N_{\nu-1}}{N_\nu}\right)} \text{ for } \nu = 1, \dots, 6$$

and the non-constant components $\rho v_2, \rho v_3, B_2$ and B_3 . Illustrate the final result for these variables in tables of the form:

| ν | N_ν | Δx_ν | ϵ_ν | EOC_ν |
|-------|---------|----------------|----------------|-----------|
|-------|---------|----------------|----------------|-----------|

Compound shock test

Consider the one-dimensional shock-tube in the domain $\Omega = [-0.5, 0.5]$ initialized as

$$\rho(x, 0) = \begin{cases} 1, & x < 0 \\ 0.125, & x \geq 0 \end{cases}, \quad p(x, 0) = \begin{cases} 1, & x < 0 \\ 0.1, & x \geq 0 \end{cases},$$

$$B_1(x, 0) = 0.75, \quad B_2(x, 0) = \begin{cases} 1, & x < 0 \\ -1, & x \geq 0 \end{cases}$$

with $\gamma = 5/3$. All other initial values are set to zero and the boundary conditions are constant. Run this test case to the final time of $T = 0.1$ and plot the density as well as the B_2 -component of the magnetic field. Take care of choosing a stable time step and a sufficiently fine mesh resolution.

Project 2: Finite volume method for two-dimensional MHD

Estimated time limit: End of Session 2

Objective: Implementation of a first order finite volume method to solve the two-dimensional ideal MHD equations on a periodic domain, validation by convergence test

Key elements:

- periodic domain $\Omega = [a, b]^2 \subset \mathbb{R}^2$ divided into an uniform Cartesian grid with squared cells of size $\Delta x = \Delta y = \frac{b-a}{N}$
- data structure for solution array, e.g. $u(i, j, k)$ for cell (i, j) and k -th conserved variable
- extension of first order finite volume method to 2D with local Lax-Friedrich solver at cell boundaries
 - adding second flux routine `g = getAdvectiveFluxY`
 - adapting eigenvalue and time step computation
 - adding second Riemann solver `G_LLF = RiemannFluxY`
- integration in time by explicit Euler method with stable CFL time step computation
- validation by convergence test

Convergence test:

As an extension of the one-dimensional Alfvén wave test we now consider the function $\xi(x, y) = x \cos(\alpha) + y \sin(\alpha)$ in the domain $\Omega = [0, \sqrt{2}] \times [0, \sqrt{2}]$ and initialize the primitive variables of our solution as:

$$\rho_0 = 1, \quad \vec{v}_0 = \begin{pmatrix} -0.1 \sin(2\pi\xi) \sin(\alpha) \\ 0.1 \sin(2\pi\xi) \cos(\alpha) \\ 0.1 \cos(2\pi\xi) \end{pmatrix}, \quad p_0 = 0.1, \quad \vec{B}_0 = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \\ 0 \end{pmatrix} + \vec{v}_0$$

Again, the solution of this test case is smooth as well as periodic in Ω and the velocity of the Alfvén-wave remains constant equal to one. Thus, the solution adopts the initial values for each time $t \in \mathbb{N}$.

Run your code with this initial data, $\alpha = \pi/4$ and $\gamma = 5/3$. Set $CFL = 0.9$ the final time $T = 1$ to compute the maximal errors

$$\epsilon_\nu = \|U - \tilde{U}_{N_\nu}\|_{L^\infty} = \max_{i,j=1,\dots,N_\nu} |U(x_i, y_j) - \tilde{U}_{N_\nu}(x_i, y_j)|$$

with respect to your approximations \tilde{U}_{N_ν} for $N_\nu = 10 \cdot 2^\nu, \nu = 0, \dots, 4$. Finally, we can again verify the correctness of the implementation with the same EOC formula and tables as in the first project.

Bonus: Additional convergence test

So far, we used a generic Alfvén wave solution in the error analysis, which adopts the initial values for each time $t \in \mathbb{N}$. But in general, we do not know the analytical solution for systems of non-linear conservation laws. Thus, we consult the method of *manufactured solutions*, in which we compute the residual for a given solution and solve the inhomogeneous problem:

$$\frac{\partial}{\partial t} U(x, y, t) + \frac{\partial}{\partial x} F(U(x, y, t)) + \frac{\partial}{\partial y} G(U(x, y, t)) = R(x, y, t)$$

Consider $\omega(x, y, t) = \sin(2\pi(x + y) - 4t) + 4$ in $\Omega \times \mathbb{R}^+$ with $\Omega = [0, 1]^2$ and the solution vector of the MHD system defined by

$$\rho = \omega, \quad \vec{v} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad E = 2\omega^2, \quad \vec{B} = \begin{pmatrix} \omega \\ -\omega \\ 0 \end{pmatrix}$$

with $\gamma = 2$. Derive the residual $R(x, y, t)$ for the corresponding $U(x, y, t)$ and solve the inhomogeneous problem with your finite volume program by adding $\Delta t R(x, y, t)$ to your approximation after each time step. Then, the numerical solution $\tilde{U}(x, y, t)$ should converge against the analytical solution of the inhomogeneous problem $U(x, y, t)$ and you can compute the errors and experimental order of convergence as usual. Visualize your convergence results in a table.

Project 3: Divergence cleaning

Estimated time limit: Coffee break of Session 3

Objective: Extension of the two-dimensional finite volume scheme to maintain the divergence-free constraint of the magnetic field, verification by a test example

Key elements:

- implementation of the Generalized Lagrange Multiplier (GLM) divergence correction method
 - introduction of additional divergence correction variable ψ
 - modification of the magnetic field fluxes
 - solving additional (hyperbolic) equation

$$\psi_t + c_h^2 \left(\frac{\partial}{\partial x} B_1 + \frac{\partial}{\partial y} B_2 \right) = - \frac{c_h^2}{c_p^2} \psi.$$

with $c_p = 0.18$ and $c_h = \lambda_{\max}$, the globally maximal eigenvalue

- verification

Verification test case:

In order to observe the damping and propagating of the divergence error, we use a non-divergence-free initialization in $\Omega = [-1, 1]^2$ defined by a Gaussian pulse in the x -component of the magnetic field:

$$\rho(x, y, 0) = 1, \quad E(x, y, 0) = 6, \quad B_1(x, y, 0) = \exp \left(-0.5 \frac{x^2 + y^2}{0.11^2} \right)$$

The other initial values are set to zero and the boundaries are periodic. Again we set $\gamma = 5/3$ and $CFL = 0.5$. Run this testcase for different mesh resolutions and plot B_1 at various times as well as the time evolution of the discrete divergence error $\|\nabla \cdot B\|_{L^\infty}$.

Bonus: Outflow boundary conditions

Implement more appropriate Riemann fluxes at the boundaries to avoid re-entering of the divergence error at opposite domain boundaries.

Project 4: MHD blast wave

Estimated time limit: End of Session 3

Objective: Application of the two-dimensional finite volume solver to a 2D MHD blast wave problem

MHD blast wave

This test problem leads to the onset of strong MHD discontinuities, relevant to astrophysical phenomena where magnetic fields can have strong dynamical effects. It describes an initially circular pressure pulse in the domain $\Omega = [-0.5, 0.5]^2$ defined by the radius $r = \sqrt{x^2 + y^2}$. Accordingly, the primitive variables for the MHD blast wave are initialized as:

$$\begin{aligned}\rho(x, y, 0) &= 1, & B_1(x, y, 0) &= \frac{100}{\sqrt{4\pi}} \\ p(x, y, 0) &= \begin{cases} 100, & r \leq 0.1 \\ 0.1, & r > 0.1 \end{cases}\end{aligned}$$

All other initial values are set to zero, $\gamma = 1.4$ and the boundary conditions are periodic.

Run the simulation for the Euler case ($B_1(x, y, 0) = 0$) and the MHD initialization until $T = 0.01$. For either configuration plot the density ρ and the pressure p at various times. Take care of choosing a stable time step and a sufficiently fine mesh resolution. How do the simulation results differ?

Bonus Project: Reconstruction to second order

Objective: Extension of the two-dimensional finite volume solver to second order, validation by convergence test, comparison of simulation results, application to more test examples

Key elements:

- implementation of the linear reconstruction
 - writing new routine `reconstructSolution`
 - determination of slope in each direction by MINMOD limiter
 - calling routine after each time step to perform linear re-interpolation
- implementation of Heun's method to update solution in time
- validation by convergence test
- comparison of simulation results for magnetic blast wave
- application to Orszag-Tang vortex and magnetic rotor

Orszag-Tang vortex

Concluding, we are now able to simulate a really cool testcase, the *Orszag-Tang Vortex*, which describes a turbulent plasma cloud. We consider the domain $\Omega = [0, 1]^2$ with periodic boundary conditions and the initial data:

$$\begin{aligned} \rho &= \gamma \frac{5}{12\pi} & v_1 &= -\sin(2\pi y) & v_2 &= \sin(2\pi x) \\ p &= \frac{5}{12\pi} & B_1 &= -\frac{1}{\sqrt{4\pi}} \sin(2\pi y) & B_2 &= \frac{1}{\sqrt{4\pi}} \sin(4\pi x) \end{aligned}$$

The other initial values are set to zero and $\gamma = 5/3$. Take care of choosing a stable time step and a sufficiently fine mesh resolution. Run your simulation until a final time of $T = 0.5$ and plot the density together with contour lines of the magnetic field at different times.

Note: This test case needs a really high resolution (and accuracy), which is difficult to achieve with Matlab.

Magnetic rotor

As a last test case we consider the magnetic rotor, which is defined in the domain $\Omega = [0, 1]^2$ as a rotating dense circle surrounded by a static fluid. Let $r = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}$ be the radius and $m = \frac{r_1 - r}{r_1 - r_0}$ with r_0, r_1 and μ_0 fixed parameters. Then, we can initialize the primitive variables for the magnetic rotor as:

$$p(x, y, 0) = 1, \quad B_1(x, y, 0) = \frac{5}{\sqrt{4\pi}}, \quad \rho(x, y, 0) = \begin{cases} 10, & r < r_0 \\ 1 + 9m, & r \in [r_0, r_1] \\ 1, & r > r_1 \end{cases}$$

$$v_1(x, y, 0) = \begin{cases} \frac{\mu_0}{r_0} \left(\frac{1}{2} - y \right), & r < r_0 \\ \frac{m\mu_0}{r_0} \left(\frac{1}{2} - y \right), & r \in [r_0, r_1] \\ 0, & r > r_1 \end{cases} \quad v_2(x, y, 0) = \begin{cases} \frac{\mu_0}{r_0} \left(x - \frac{1}{2} \right), & r < r_0 \\ \frac{m\mu_0}{r_0} \left(x - \frac{1}{2} \right), & r \in [r_0, r_1] \\ 0, & r > r_1 \end{cases}$$

The other initial values are set to zero and $\gamma = 1.4$. Again, we use periodic boundary conditions and set $r_0 = 0.1, r_1 = 0.115, \mu_0 = 2$. Take care of choosing a stable time step and a sufficiently fine mesh resolution. Run the simulation until $T = 0.15$ and plot the density ρ and the pressure p at various times.

Note: This test case needs a really high resolution (and accuracy), which is difficult to achieve with Matlab.