Implicit relaxation schemes for compressible fluid models

M. Boileau[2], D. Coulette[2], E. Franck[12], P. Helluy[12]
C. Hillairet [2], M. Mehrenberger[2], L. Mendoza[2], L. Navoret [2]
H. Oberlin[3], A. Ratnani[3], E. Sonnendrücker[3]

Seminar IRMA, March 2017, Strasbourg

[1]Inria Nancy Grand Est, France
[2]IRMA, university of Strasbourg, France
[3]NMPP, IPP, Garching bei München, Germany

Outline

Mathematical and physical problems

Hyperbolic systems and implicit schemes

- We consider the general problem

$$\partial_t \boldsymbol{U} + \partial_x(\boldsymbol{F}(\boldsymbol{U})) = \nu \partial_x(D(\boldsymbol{U})\partial_x \boldsymbol{U})$$

- with $\boldsymbol{U} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ (idem for $\boldsymbol{F}(\boldsymbol{U})$) and D a matrix.
- In the following we consider the limit $\nu << 1$.

Implicit schemes

- **Implicit scheme**: allows to avoid the CFL condition filtering the fast phenomena.
- **Problem**: Direct solvers are not useful in 3D (too large matrices), we need iterative solvers.
- Conditioning of the implicit matrix: given by the ratio of the maximal and minimal eigenvalues.

- Implicit scheme :

$$\boldsymbol{U} + \Delta t \partial_x(\boldsymbol{F}(\boldsymbol{U})) - \Delta t \nu \partial_x(D(\boldsymbol{U})\partial_x \boldsymbol{U}) = \boldsymbol{U}^n$$

- At the limit $\nu << 1$ and $\Delta t >> 1$ (large time step) we solve $\partial_x \boldsymbol{F}(\boldsymbol{U}) = 0$.

Issues of implicit schemes

- **Conclusion**: for $\nu << 1$ and $\Delta t >> 1$ the conditioning number of the full system closed to conditioning number of the steady hyperbolic model (the ratio of the speed waves).

Example of ill-conditioned systems

- Euler equation

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \\ \partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + p I_d) = 0, \\ \partial_t (\rho e) + \nabla \cdot (\rho \boldsymbol{u} e + \boldsymbol{u} p) = 0 \end{cases}$$

- Eigenvalues : $(\boldsymbol{u}, \boldsymbol{nn}) \pm c$ and $(\boldsymbol{u}, \boldsymbol{nn})$ with c the sound speed.
- Mach number : $M = \frac{|\boldsymbol{u}|}{c}$
- Nondimensional eigenvalues :

$$M - 1, M, M + 1$$

- **Conclusion**: ill-conditioned system for

$$M << 1 \text{ and } M = 1$$

- Same type of problem : Shallow - Water with sedimentation transport.

- Ideal MHD

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \\ \rho \partial_t \boldsymbol{u} + \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \boldsymbol{J} \times \boldsymbol{B}, \\ \partial_t p + \boldsymbol{u} \cdot \nabla p + p \nabla \cdot \boldsymbol{u} = 0 \\ \partial_t \boldsymbol{B} = -\nabla \times (-\boldsymbol{u} \times \boldsymbol{B}), \\ \nabla \cdot \boldsymbol{B} = 0, \quad \nabla \times \boldsymbol{B} = \boldsymbol{J}. \end{cases}$$

- Eigenvalues : $(\boldsymbol{u}, \boldsymbol{nn})$, $(\boldsymbol{u}, \boldsymbol{nn}) \pm V_a$, $(\boldsymbol{u}, \boldsymbol{nn}) \pm \phi(c, V_a, \theta)$ with c the sound speed, V_a the Alfven speed and θ the angle between $\boldsymbol{nn}$ and the $\boldsymbol{B}$.
- Mach number : $M = \frac{|\boldsymbol{u}|}{c}$ and β-number : $\beta = \frac{c}{V_a}$
- Approximated Nondimensional eigenvalues for $\beta << 1$ (Tokamak)

$$\beta M, \quad \beta M \pm 1, \quad M\beta \pm (\beta + 1)$$

in the parallel direction of the magnetic field (different in the perpendicular region).

- **Conclusion**: for example we have an ill-conditioned system for

$$M << 1, \quad \beta << 1$$

Idea

Classical method

- Implicit scheme and nonlinear solver as Newton method
- Full linearized problem solving with preconditioned iterative solver or exact solver.

Limit of the classical method

- High memory consumption to store Jacobian and perhaps preconditioning.
- CPU time does not increase linearly with respect to the problem size (effect of the ill-conditioning linked to the physic).

Future of scientific computing

- Machines able to make lots of parallel computing.
- Small memory by node.

Idea: Divide and Conquer

- Propose algorithm which approximates the full problem by a collection of simpler ones.
- Perform the resolution of the simple problems.
- Limit memory consumption using matrix-free method.

Classical Relaxation methods

General principle

- We consider the following nonlinear system

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \nu \partial_x (D(\boldsymbol{U}) \partial_x \boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U})$$

- with $\boldsymbol{U}$ a vector of N functions.
- **Aim**: Find a way to approximate this system with a sequence of simple systems.
- **Idea**: Xin-Jin relaxation method (very popular in the hyperbolic and finite volume community).

$$\begin{cases} \partial_t \boldsymbol{U} + \partial_x \boldsymbol{V} = \boldsymbol{G}(\boldsymbol{U}) \\ \partial_t \boldsymbol{V} + \alpha^2 \partial_x \boldsymbol{U} = \dfrac{1}{\varepsilon}(\boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{V}) \end{cases}$$

Limit of the hyperbolic relaxation scheme

- The limit scheme of the relaxation system is

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{G}(\boldsymbol{U}) + \varepsilon \partial_x ((\alpha^2 - \mid A(\boldsymbol{U}) \mid^2) \partial_x \boldsymbol{U}) + \varepsilon \partial_x \boldsymbol{G}(\boldsymbol{U}) + o(\varepsilon^2)$$

- with $A(\boldsymbol{U})$ the Jacobian of $\boldsymbol{F}(\boldsymbol{U})$.

- **Conclusion**: the relaxation system is an approximation of the hyperbolic original system (error in ε).
- **Stability**: the limit system is dissipative if $(\alpha^2 - \mid A(\boldsymbol{U}) \mid^2) > 0$.

General principle II

Generalization

- The generalized relaxation is given by

$$\begin{cases} \partial_t \boldsymbol{U} + \partial_x \boldsymbol{V} = \boldsymbol{G}(\boldsymbol{U}) \\ \partial_t \boldsymbol{V} + \alpha^2 \partial_x \boldsymbol{U} = \dfrac{R(\boldsymbol{U})}{\varepsilon}(\boldsymbol{F}(\boldsymbol{U}) - \boldsymbol{V}) + \boldsymbol{H}(\boldsymbol{U}) \end{cases}$$

- The limit scheme of the relaxation system is

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{G}(\boldsymbol{U})$$
$$+ \varepsilon \partial_x (R(\boldsymbol{U})^{-1}(\alpha^2 - \mid A(\boldsymbol{U}) \mid^2) \partial_x \boldsymbol{U}) + \varepsilon \partial_x (A(\boldsymbol{U})\boldsymbol{G}(\boldsymbol{U}) - \boldsymbol{H}(\boldsymbol{U})) + o(\varepsilon^2)$$

Treatment of small diffusion

- Taking $R(\boldsymbol{U}) = (\alpha^2 - \mid A(\boldsymbol{U}) \mid^2) D(\boldsymbol{U})^{-1}$, $\varepsilon = \nu$ and $\boldsymbol{H}(\boldsymbol{U}) = A(\boldsymbol{U})\boldsymbol{G}(\boldsymbol{U})$: we obtain the following limit system

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{G}(\boldsymbol{U}) + \nu \partial_x (D(\boldsymbol{U}) \partial_x \boldsymbol{U}) + o(\nu^2)$$

- **Limitation of the method**: the relaxation model cannot approach pde with high diffusion.

Time discretization

Main property

- **Relaxation system**: "the nonlinearity is local and the non locality is linear".
- **Main idea**: splitting scheme between transport and the relaxation.

First order scheme

- We define the three operators for each step :

$$T_{\Delta t} = I_d + \Delta t \begin{pmatrix} \partial_x I_d^v \\ \alpha^2 \partial_x I_d^u \end{pmatrix}, \quad S_{\Delta t} = I_d + \Delta t \begin{pmatrix} \boldsymbol{G}(I_d^u) \\ 0 \end{pmatrix}$$

$$R_{\Delta t} = I_d + \Delta t \begin{pmatrix} 0 \\ -\frac{R(I_d^u)}{\varepsilon}(\boldsymbol{F}(I_d^u) - I_d^v) - \boldsymbol{H}(I_d^u) \end{pmatrix}$$

- The final scheme $T_{\Delta t} \circ S_{\Delta t} \circ R_{\Delta t}$ is consistent with

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{G}(\boldsymbol{U}) + \frac{\Delta t}{2}\left(\partial_t \boldsymbol{G}(\boldsymbol{U}) + \partial_x \boldsymbol{H}(\boldsymbol{U})\right)$$

$$\frac{\Delta t}{2}\left(\partial_x(\alpha^2 \partial_x \boldsymbol{U}) + \partial_x\left(\left(\alpha^2 I_d - A(\boldsymbol{U})^2\right)\partial_x \boldsymbol{U}\right)\right) + \varepsilon \partial_x\left(R^{-1}\left(\alpha^2 I_d - A(\boldsymbol{U})^2\right)\partial_x \boldsymbol{U}\right)$$

$$+ O(\varepsilon \Delta t + \Delta t^2 + \varepsilon^2)$$

- **Remark**: the viscosity induced by the splitting has the same form as the viscosity induced by the relaxation.

Time discretization

Main property

- **Relaxation system**: "the nonlinearity is local and the non locality is linear".
- **Main idea**: splitting scheme between transport and the relaxation.

First order scheme

- We define the three operators for each step :

$$T_{\Delta t} = I_d + \Delta t \begin{pmatrix} \partial_x I_d^v \\ \alpha^2 \partial_x I_d^u \end{pmatrix}, \quad S_{\Delta t} = I_d + \Delta t \begin{pmatrix} \boldsymbol{G}(I_d^u) \\ 0 \end{pmatrix}$$

$$R_{\Delta t} = I_d + \Delta t \begin{pmatrix} 0 \\ -\frac{R(I_d^u)}{\varepsilon}(\boldsymbol{F}(I_d^u) - I_d^v) - \boldsymbol{H}(I_d^u) \end{pmatrix}$$

- The final scheme $T_{\Delta t} \circ S_{\Delta t} \circ R_{\Delta t}$ is consistent with

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \nu \partial_x (D(\boldsymbol{U}) \partial_x \boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U})$$
$$+ \frac{\Delta t}{2} \left(\partial_t \boldsymbol{G}(\boldsymbol{U}) + \partial_x \boldsymbol{H}(\boldsymbol{U}) \right) + \frac{\Delta t}{2} \left(\partial_x (\alpha^2 \partial_x \boldsymbol{U}) + \partial_x \left(\left(\alpha^2 I_d - A(\boldsymbol{U})^2 \right) \partial_x \boldsymbol{U} \right) \right)$$
$$+ O(\nu \Delta t + \Delta t^2 + \nu^2)$$

- **Remark**: the viscosity induced by the splitting has the same form as the viscosity induced by the relaxation.

Discretization of the transport step

Main property

- Transport part:

$$\begin{cases} \partial_t \boldsymbol{U} + \partial_x \boldsymbol{V} = 0 \\ \partial_t \boldsymbol{V} + \alpha^2 \partial_x \boldsymbol{U} = 0 \end{cases}$$

- Can be rewritten as N independent acoustic wave problems.

- We propose an efficient way to solve a single wave equation in the FE/IGA context.

$$\begin{pmatrix} I_d & \theta\Delta t\partial_x \\ \alpha^2\theta\Delta t\partial_x & I_d \end{pmatrix} \begin{pmatrix} u^* \\ v^* \end{pmatrix} = \begin{pmatrix} I_d & -(1-\theta)\Delta t\partial_x \\ -\alpha^2(1-\theta)\Delta t\partial_x & I_d \end{pmatrix} \begin{pmatrix} u^n \\ v^n \end{pmatrix}$$

- Now we propose to apply a Schur decomposition of the implicit matrix.

Final algorithm problem

$$\begin{cases} \text{Predictor}: & v^* = v^n - (1-\theta)\Delta t\partial_x u \\ \text{Update}: & (I_d - \alpha^2\theta^2\Delta t^2\partial_{xx})u^{n+1} = -\theta\Delta t\partial_x v^* + (u^n - (1-\theta)\Delta t\partial_x v^n) \\ \text{Corrector}: & v^{n+1} = v^* - \alpha^2\theta\Delta t\partial_x u^{n+1} \end{cases}$$

- **Systems to solve**: 2 mass matrices and on Laplacian by wave equations.
- **Parallelization** (simple BC): N independent mass matrices, N independent stiffness matrices, N independent mass matrices.
- **Parallelization** (complex BC): N independent mass matrices, one linear matrix of the size N (N Laplacian weakly coupled by the boundary), N independent mass matrices.

2D- Extension

Generalization

- The generalized relaxation is given by

$$\begin{cases} \partial_t \boldsymbol{U} + \partial_x \boldsymbol{V}_x + \partial_y \boldsymbol{V}_y = 0 \\ \partial_t \boldsymbol{V}_x + \alpha^2 B_{xx} \partial_x \boldsymbol{U} + \alpha^2 B_{xy} \partial_y \boldsymbol{U} = \dfrac{\Omega_{xx}}{\varepsilon}(\boldsymbol{F}_x(\boldsymbol{U}) - \boldsymbol{V}_x) + \dfrac{\Omega_{xy}}{\varepsilon}(\boldsymbol{F}_y(\boldsymbol{U}) - \boldsymbol{V}_y) \\ \partial_t \boldsymbol{V}_y + \alpha^2 B_{yx} \partial_x \boldsymbol{U} + \alpha^2 B_{yy} \partial_y \boldsymbol{U} = \dfrac{\Omega_{yx}}{\varepsilon}(\boldsymbol{F}_x(\boldsymbol{U}) - \boldsymbol{V}_x) + \dfrac{\Omega_{yy}}{\varepsilon}(\boldsymbol{F}_y(\boldsymbol{U}) - \boldsymbol{V}_y) \end{cases}$$

- The limit scheme of the relaxation system is

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}_x(\boldsymbol{U}) + \partial_y \boldsymbol{F}_y(\boldsymbol{U}) = \varepsilon \nabla \cdot (\Omega^{-1}(\alpha^2 B - A^q)\nabla \boldsymbol{U}) + o(\varepsilon^2)$$

- **Remark**: classical choice for B is $B_{xx} = B_{yy} = I_d$ and $B_{yx} = B_{xy} = 0$
- B can be a way to reduce the diffusion adding null wave in the linear system.
- **Discretization**: same space, time discretization and algorithm as in 1D.

Parallelization of the models

- **Transport step** (simple BC): $d * N$ independent mass matrices, N independent stiffness matrices, $d * N$ independent mass matrices.
- **Transport step** (complex BC): $d * N$ independent mass matrices, one linear matrix of the size N (structure depends of B), $d * N$ independent.
- **Relaxation step**: $d * N$ independent mass matrices.

High-Order time schemes

Second-order scheme

- Scheme for transport step $T(\Delta t)$: Cranck Nicholson + FE/IGA
- Scheme for relaxation step $R(\Delta t)$: Cranck Nicholson + FE/IGA
- Classical full second order scheme:

$$\Psi(\Delta t) = T\left(\frac{\Delta t}{2}\right) \circ R(\Delta t) \circ T\left(\frac{\Delta t}{2}\right).$$

- AP full second order scheme:

$$\Psi_{ap}(\Delta t) = T\left(\frac{\Delta t}{4}\right) \circ R\left(\frac{\Delta t}{2}\right) \circ T\left(\frac{\Delta t}{2}\right) \circ R\left(\frac{\Delta t}{2}\right) \circ T\left(\frac{\Delta t}{4}\right).$$

- Ψ and Ψ_{ap} symmetric in time. $\Psi_{ap}(0) = I_d$.

High order scheme

- Using composition method

$$M_p(\Delta t) = \Psi_{ap}(\gamma_1 \Delta t) \circ \Psi_{ap}(\gamma_2 \Delta t) ..... \circ \Psi_{ap}(\gamma_s \Delta t)$$

- with $\gamma_i \in [-1, 1]$, we obtain a p-order schemes.
- Susuki scheme : $s = 5$, $p = 4$. Kahan-Li scheme: $s = 9$, $p = 6$.

Kinetic Relaxation methods

Kinetic relaxation scheme

- We consider the classical Xin-Jin relaxation for a scalar system $\partial_t u + \partial_x F(u) = 0$:

$$\begin{cases} \partial_t u + \partial_x v = 0 \\ \partial_t v + \alpha^2 \partial_x u = \dfrac{1}{\varepsilon}(F(u) - v) \end{cases}$$

- We diagonalize the hyperbolic matrix $\begin{pmatrix} 0 & 1 \\ \alpha^2 & 0 \end{pmatrix}$ and note f_+ and f_- the new variables. We obtain

$$\begin{cases} \partial_t f_- - \alpha \partial_x f_- = \dfrac{1}{\varepsilon}(f_e q^- - f_-) \\ \partial_t f_+ + \alpha \partial_x f_+ = \dfrac{1}{\varepsilon}(f_e q^+ - f_+) \end{cases}$$

- with $f_{eq}^{\pm} = \frac{u}{2} \pm \frac{F(u)}{2\alpha}$.

First Generalization

- **Main property**: the transport is diagonal.

First Generalization

- **Vectorial kinetic relaxation**: each wave equation is diagonalized. We obtain N diagonal systems of size 2.

Generic kinetic relaxation scheme

Kinetic relaxation system

- **Lattice**: $W = \{\lambda_1 .... \lambda_{n_v}\}$ a set of velocities.
- **Mapping matrix**: P a matrix $n_c \times n_v$ $(n_c < n_v)$ such that $\boldsymbol{U} = P\boldsymbol{f}$, with $U \in \mathbb{R}^{n_c}$.
- **Kinetic relaxation system**:

$$\partial_t \boldsymbol{f} + \Lambda \partial_x \boldsymbol{f} = \frac{R}{\varepsilon}(\boldsymbol{f}^{eq}(\boldsymbol{U}) - \boldsymbol{f})$$

- Equilibrium vector operator $\boldsymbol{f}^{eq} : \mathbb{R}^{n_c} \to \mathbb{R}^{n_v}$ such that $P\boldsymbol{f}^{eq}(\boldsymbol{U}) = \boldsymbol{U}$.

- Consistance with the initial PDE:

$$\mathcal{C} \left\{ \begin{array}{c} P\boldsymbol{f}^{eq}(\boldsymbol{U}) = \boldsymbol{U} \\ P\Lambda \boldsymbol{f}^{eq}(\boldsymbol{U}) = F(\boldsymbol{U}) \end{array} \right.$$

- For source terms and small diffusion terms, it is the same that the first relaxation method.
- **In 1D** : same property of stability that the classical relaxation method.

Multi-D extension

Multi-D extension

- **Lattice**: $W = \{\lambda_1 .... \lambda_{n_v}\}$ a set of velocities. **Kinetic relaxation system**:

$$\partial_t \boldsymbol{f} + \Lambda \cdot \nabla \boldsymbol{f} = \frac{R}{\varepsilon}(\boldsymbol{f}^{eq}(\boldsymbol{U}) - \boldsymbol{f})$$

- Equilibrium vector operator $\boldsymbol{f}^{eq} : \mathbb{R}^{n_c} \to \mathbb{R}^{n_v}$ such that $P\boldsymbol{f}^{eq}(\boldsymbol{U}) = \boldsymbol{U}$.

Stability condition

- Diffusion limit of kinetic model: $\partial_t \boldsymbol{U} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) = \varepsilon \nabla \cdot (D_f \nabla \boldsymbol{U})$.
- Structure of D_f more complex that the matrix obtained with the classical method.
- **Stability**: first study that the stability condition is more restrictive.

Choice of the Lattice

- **Minimal Lattice**: $Dd - Q(d+1)$ with d the dimension, Q the number of velocities.
- **Minimal Symmetric Lattice**: $Dd - Q(2q)$ (q velocities and the opposite).
- **Bubble Lattice**: $Dd - Q(d+2)$ (additional 0 velocity).
- **Bubble Symmetric Lattice**: $Dd - Q(2q+1)$ (symmetric + 0 velocity).
- **Main question**: Stability and properties of these Lattice. **Best one** ?

Space discretization - transport scheme

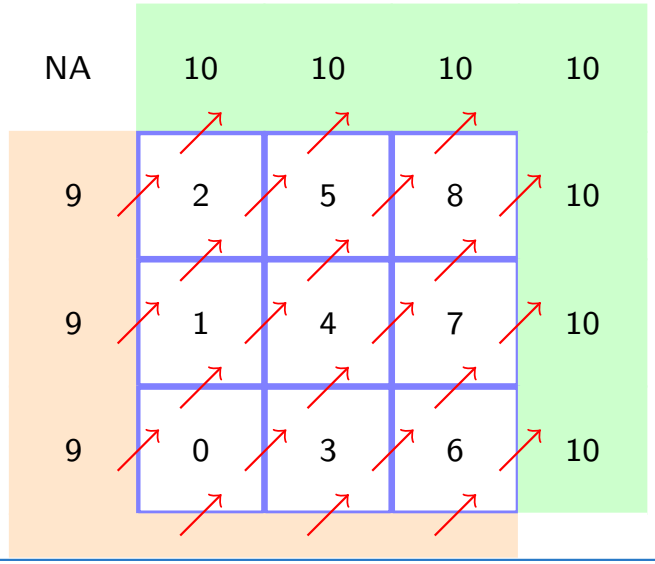
Whishlist

- Complex geometry, curved meshes
- Flexibility $h - p$ refinement
- CFL-free

Candidates for transport discretization

- Finite volumes schemes
- Discontinuous galerkin schemes
- Semi-Lagrangian schemes
- Stochastic schemes (Glimm or particle methods)

- Kirsch code: Curved block mesh. Each block is a Cartesian mesh.
- Kirsch code: Implicit DG schemes:
  - Implicit Cranck-Nicholson scheme
  - Block -Triangular matrix (Upwind scheme) solved avoiding storage of the matrix.



Numerical results

Results Burgers I

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable for $\Delta t < 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$

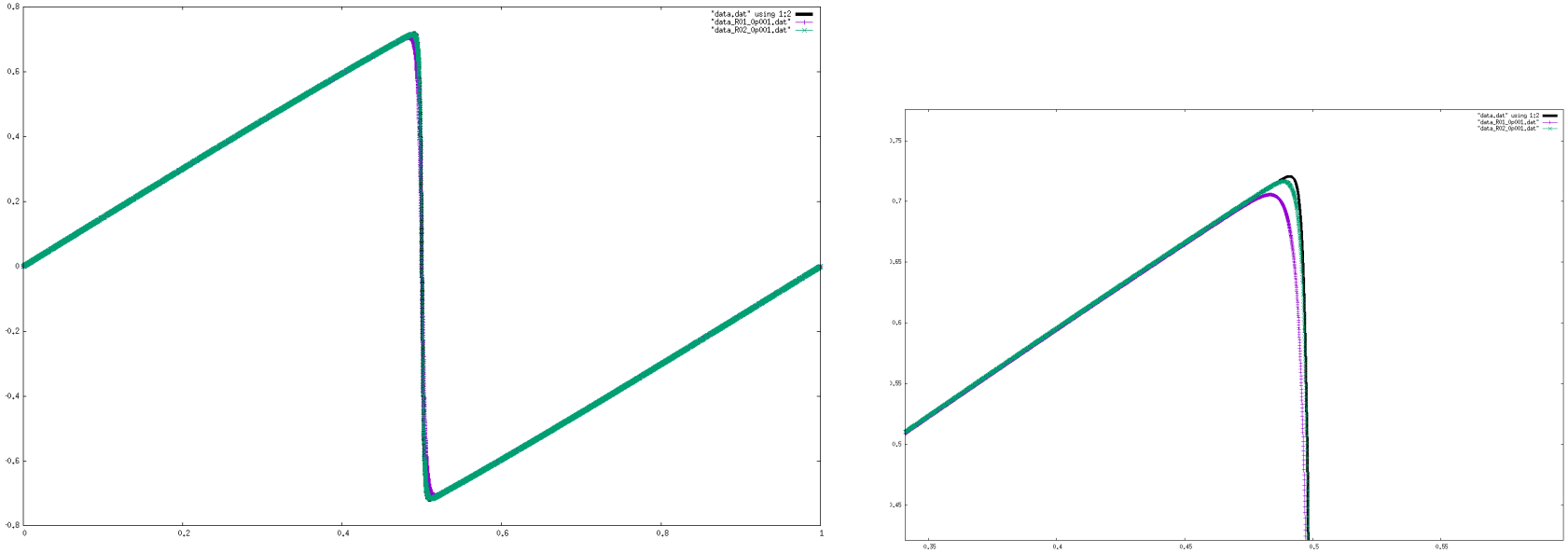


Figure: Left: numerical solution for the first order and the second order schemes for $\Delta t = 0.001$, Right: Zoom

- **Remark**: for discontinuous solutions (or strong gradient solutions) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow \alpha$ increase and α increase $\longrightarrow$ oscillations increase.

Results Burgers I

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable for $\Delta t < 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$, $\Delta t = 5.0E^{-3}$ and $\Delta t = 1.0E^{-2}$ (only for first order and AP second order).

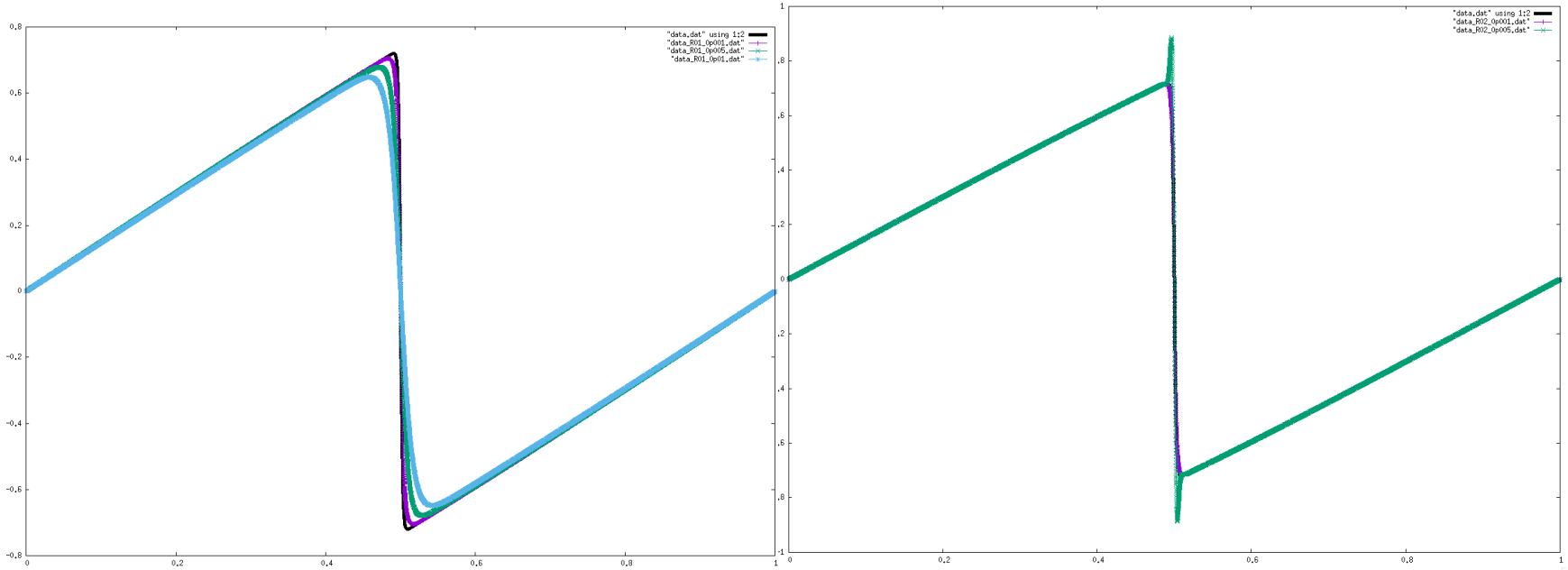


Figure: Left: numerical solution for the first order scheme, Right: numerical solution for the second order scheme. $\nu = 10^{-3}$

- **Remark**: for discontinuous solutions (or strong gradient solutions) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow$ α increase and α increase $\longrightarrow$ oscillations increase.

Results Burgers I

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable for $\Delta t < 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$, $\Delta t = 5.0E^{-3}$ and $\Delta t = 1.0E^{-2}$ (only for first order and AP second order).

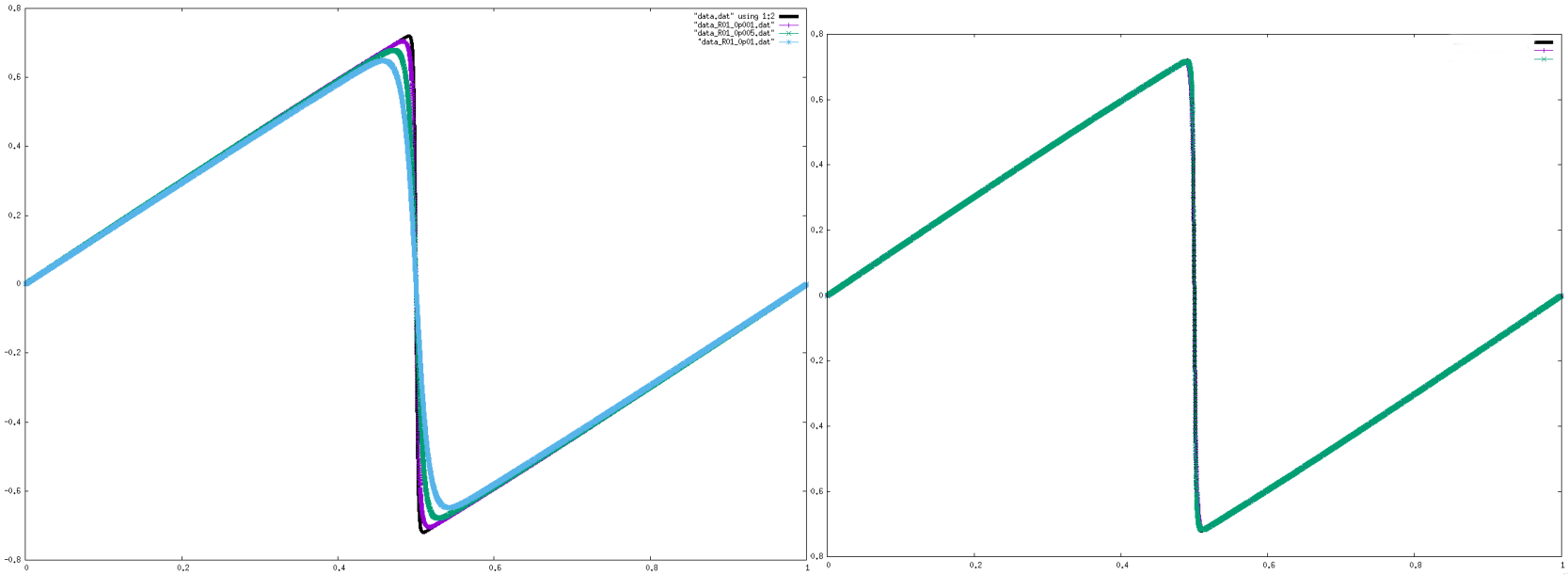


Figure: Left: numerical solution for the first order scheme, Right: numerical solution for the second order scheme. $\nu = 10^{-3}$

- **Remark**: for discontinuous solutions (or strong gradient solutions) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow \alpha$ increase and α increase $\longrightarrow$ oscillations increase.

Results Burgers I

- **Model** : Viscous - Burgers model.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.
- **Explicit time step** : stable for $\Delta t < 1.0E^{-5}$.
- **Implicit time step** : $\Delta t = 1.0E^{-3}$, $\Delta t = 5.0E^{-3}$ and $\Delta t = 1.0E^{-2}$.

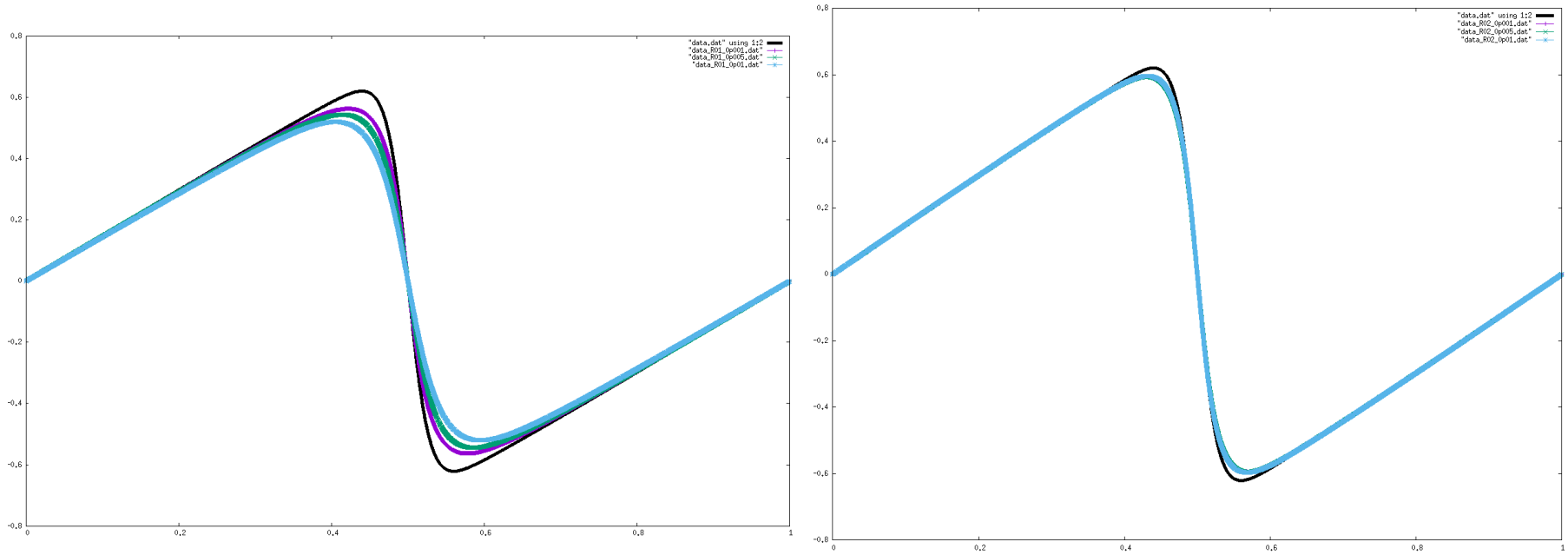


Figure: Left: numerical solution for the first order scheme, Right: numerical solution for the second order scheme. $\nu = 10^{-2}$

- **Remark**: for discontinuous solutions (or strong gradient solutions) the scheme admits high numerical dispersion and instabilities.
- **Instability**: oscillations $\longrightarrow \alpha$ increase and α increase $\longrightarrow$ oscillations increase.

Results Navier-Stokes I

- **Model**: compressible Navier-Stokes equation

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0 \\ \partial_t \rho u + \partial_x(\rho u^2 + p) = \partial_x(\nu(\rho)\partial_x u) - \rho g \\ \partial_t E + \partial_x(Eu + pu) = \partial_x(\nu(\rho)\partial_x \frac{u^2}{2}) + \partial_x(\eta \partial_x T) - \rho v g \end{cases}$$

- **Test**: Propagation of acoustic wave (no viscosity, no gravity).
- **CPU Time** for initial $Mach = 0$:

	CN method			Relaxation method		
Δt / cells	5.10^3	10^4	2.10^4	5.10^3	10^4	2.10^4
$\Delta t = 0.005$	160	540	2350	135	430	1920
$\Delta t = 0.01$	90	315	1550	70	220	1000
$\Delta t = 0.02$	55	175	765	40	125	530
$\Delta t = 0.05$	30	100	420	20	65	270

- **CPU Time** for initial $Mach = 0.5$:

	CN method			Relaxation method		
Δt / cells	5.10^3	10^4	2.10^4	5.10^3	10^4	2.10^4
$\Delta t = 0.01$	145	480	2150	100	320	1470
$\Delta t = 0.02$	80	290	1200	60	200	970

Conclusion:

- In this case the Relaxation method is competitive with the classical scheme without important optimization (no parallelization of the problem, etc).

Results Navier-Stokes II

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.

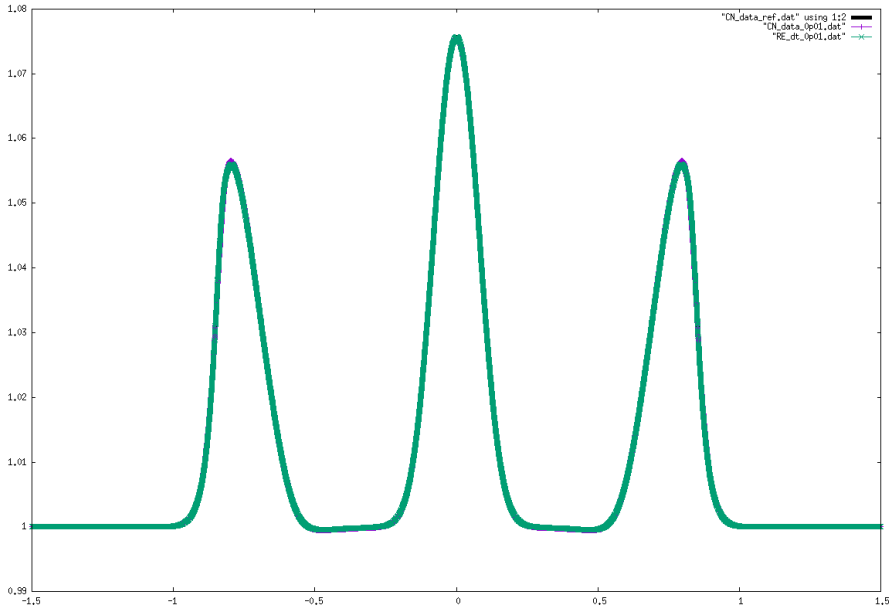


Figure: Fine solution (black). CN solution (violet) and Relaxation solution(green), $M = 0$, $\Delta t = 0.01$

Results Navier-Stokes II

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.

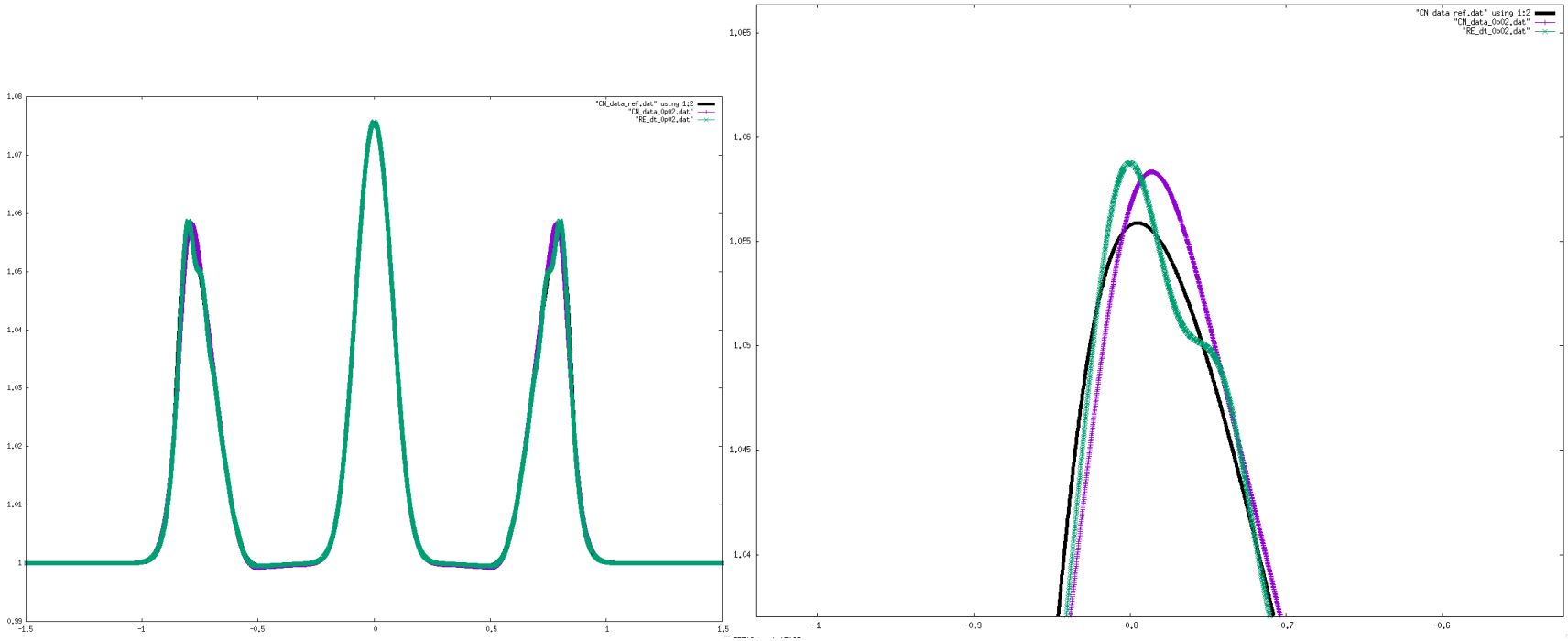


Figure: Fine solution (black). CN solution (violet) and Relaxation solution(green), $M = 0$, $\Delta t = 0.02$

Results Navier-Stokes II

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.

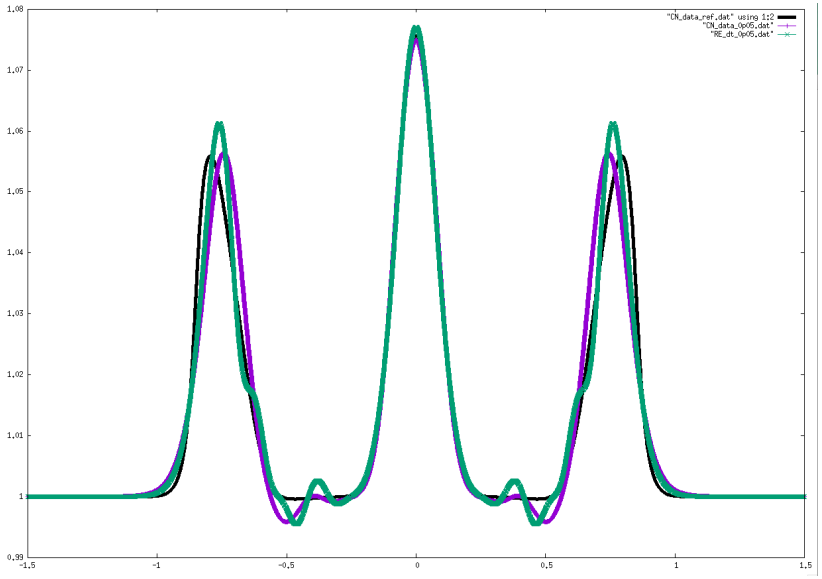


Figure: Fine solution (black). CN solution (violet) and Relaxation solution (green), $M = 0$, $\Delta t = 0.05$

- The two methods (CN and relaxation) capture well the fine solution.

Results Navier-Stokes II

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.

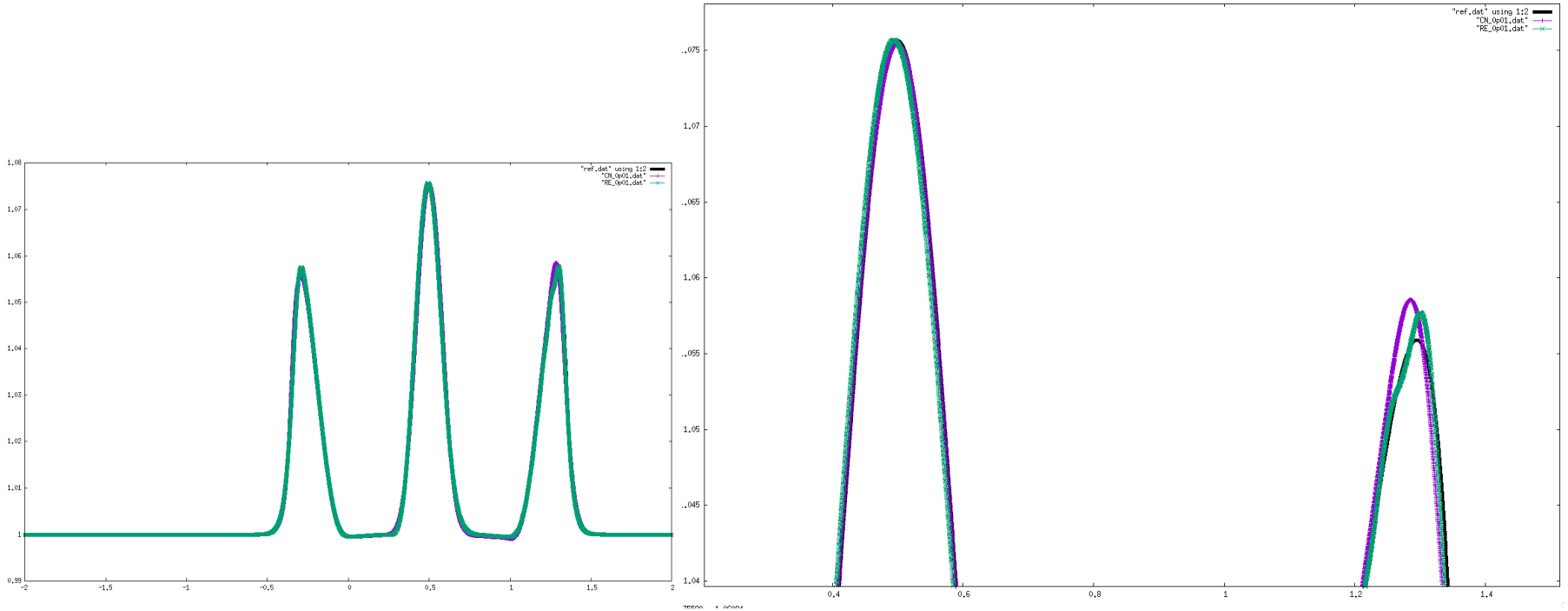


Figure: Fine solution (black). CN solution (violet) and Relaxation solution(green), $M = 0.5$, $\Delta t = 0.01$

Results Navier-Stokes II

- Simple test case: $\rho(t,x) = 1 + G(x - ut)$, $u(t,x) = 2$ and $T(t,x) = 0$.

Scheme Δt	$\Delta t = 1.0E^{-2}$	$\Delta t = 5.0E^{-3}$	$\Delta t = 2.5E^{-3}$	$\Delta t = 1.25E^{-3}$
CN scheme	$8.8E^{-3}$	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$
Relaxation scheme	$2.25E^{-3}$	$5.7E^{-4}$	$1.4E^{-4}$	$3.6E^{-5}$

- **Conclusion**: the relaxation scheme converges with the second order as expected.
- Spatial discretization: $N_{cell} = 10000$, $order = 3$. Initial condition : Gaussian.

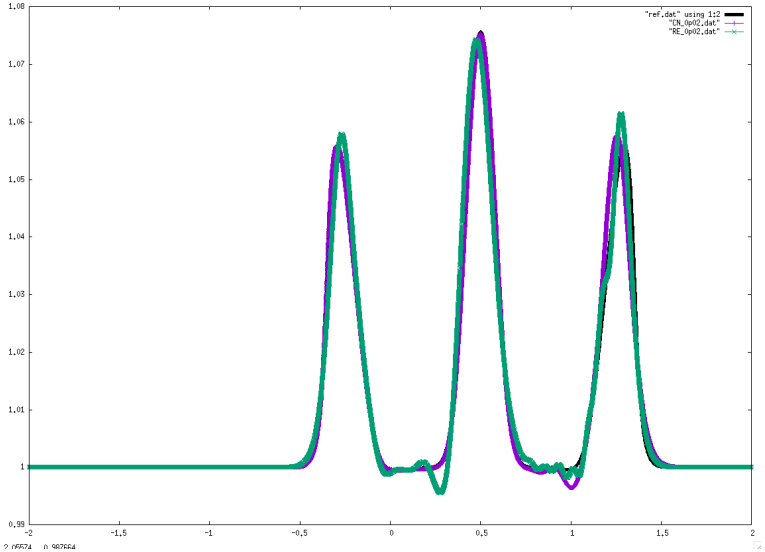


Figure: Fine solution (black). CN solution (violet) and Relaxation solution (green), $M = 0.5$, $\Delta t = 0.02$

- The two methods (CN and relaxation) capture well the fine solution.

Results for Kinetic relaxation

- **Model**: isothermal Euler equation

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0 \\ \partial_t \rho u + \partial_x(\rho u^2 + c^2 \rho) = 0 \end{cases}$$

- **Lattice**: $(D1 - Q2)^n$ Lattice scheme.
- For the transport (and relaxations step) we use 6-order DG scheme in space.
- **Time step**: $\Delta t = \beta \frac{\Delta x}{\lambda}$ with λ the lattice velocity. $\beta = 1$ explicit time step.
- **First test**: acoustic wave with $\beta = 50$ and $T_f = 0.4$, **Second test**: smooth contact wave with $\beta = 100$ and $T_f = 20$.

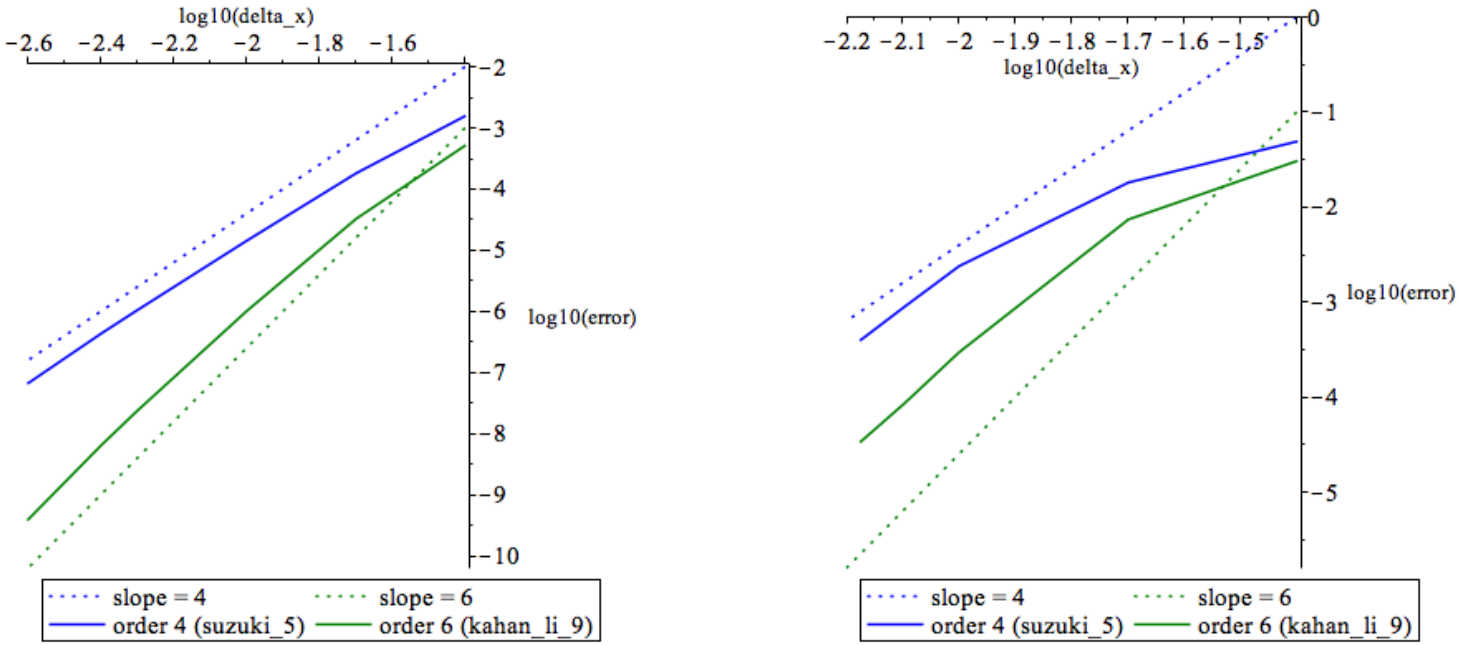


Figure: convergence rates for the first test (left) and for the second test (right).

Results for Kinetic relaxation

- **Test case**: discontinuous initial data (Sod problem). No viscosity, $\beta = 3$. 6 order space-time scheme.

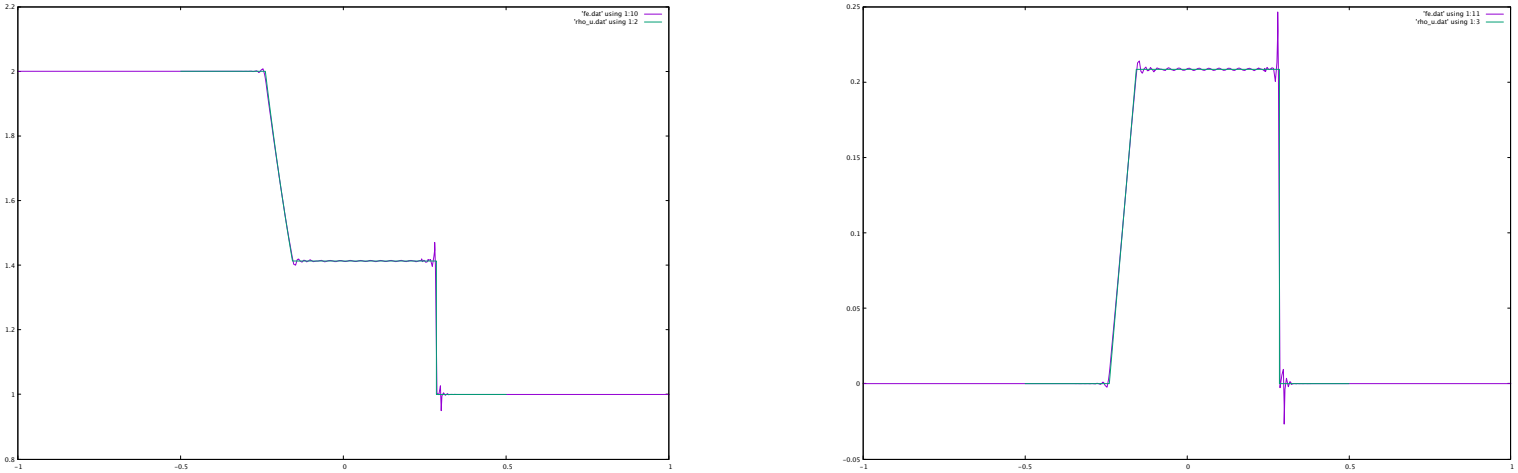
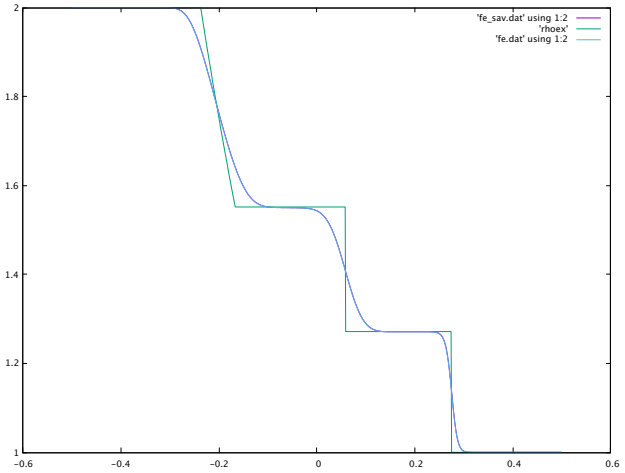


Figure: density (left) and velocity (right).

- With refinement in space we can reduce the oscillations.
- **Test case**: Sod problem. $\nu = 5.10^{-4}$, $\beta = 5$. 6 order space-time scheme.



Results 2D relaxation I

- **Model**: 2D compressible isothermal Navier-Stokes equation

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = S_r \rho \\ \partial_t (\rho \boldsymbol{u}) + \nabla \cdot \left(\rho \boldsymbol{u} \otimes \boldsymbol{u} + c^2 \rho I_d\right) = \mu \Delta \boldsymbol{u} + (\mu + \lambda) \nabla (\nabla \cdot \boldsymbol{u}) + S_r \boldsymbol{u} \end{cases}$$

- **Test I**: Steady state between source and spatial part. **Order of convergence**:

	Error	Order
$\Delta t = 0.025$	$1.6E^{-2}$	-
$\Delta t = 0.0125$	$3.8E^{-3}$	2.05
$\Delta t = 0.00625$	$9.3E^{-4}$	2.03
$\Delta t = 0.003125$	$2.3E^{-4}$	2.02

- **Test II**: Propagation of acoustic wave (no viscosity, no gravity). **CPU Time**:

	CN method			CN Newton			Relaxation method		
Δt / cells	100^2	200^2	400^2	100^2	200^2	400^2	100^2	200^2	400^2
$\Delta t = 0.01$	340	1320	5650	615	2415	9800	330	1260	5040
$\Delta t = 0.02$	170	670	3060	320	1250	6850	165	650	2555
$\Delta t = 0.05$	75	300	1290	140	555	3080	70	275	1115
$\Delta t = 0.1$	45	170	760	100	380	2190	40	155	625

Conclusion:

- The Relaxation method is competitive with the classical schemes (linearized of Newton) without parallelization of the sub-models.

Results 2D relaxation II

- **Test I**: Acoustic wave for isothermal Euler equation.

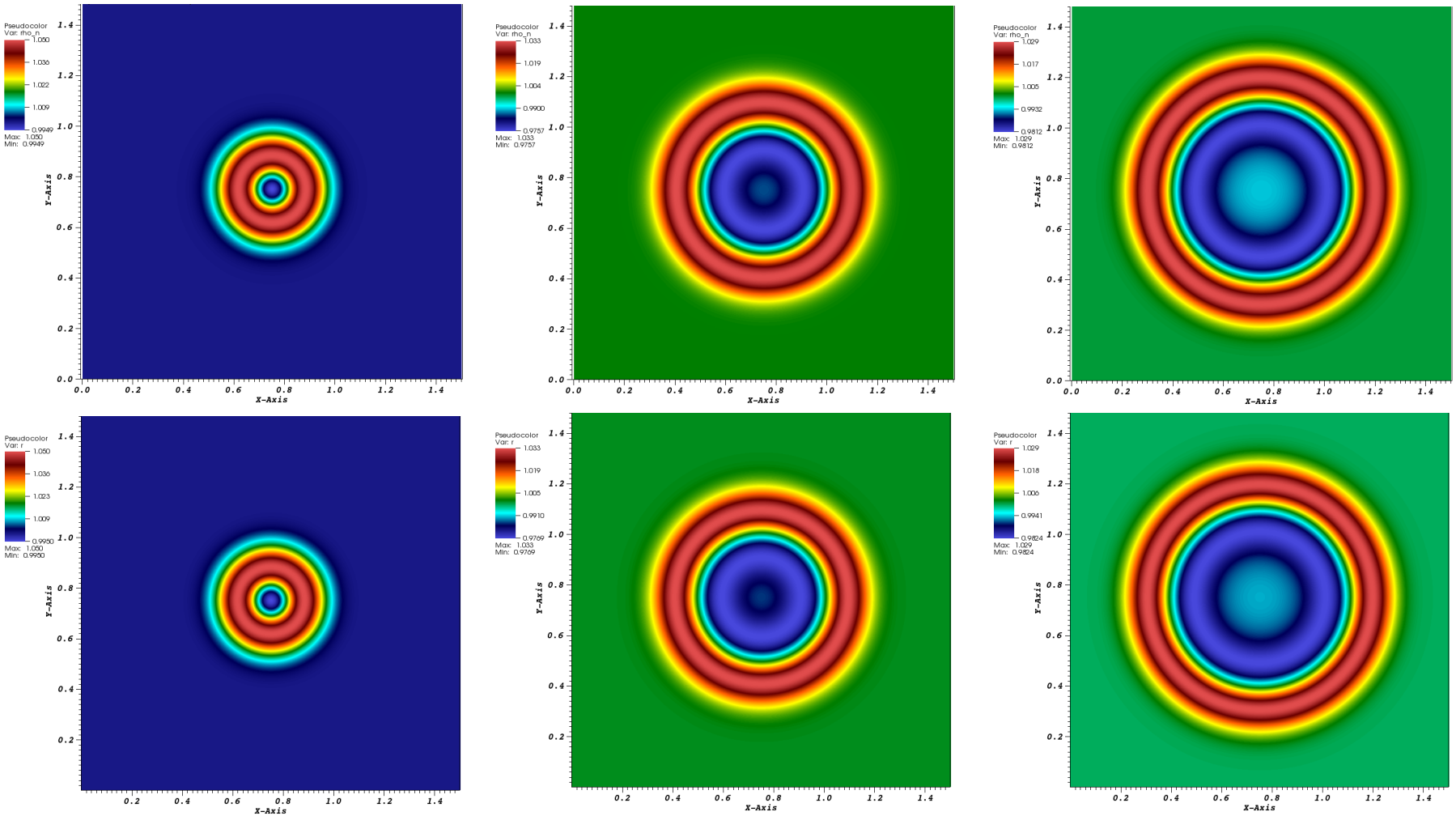


Figure: Comparison between the CN coupled with Newton method (top) and the relaxation (bottom) for a time step $\Delta t = 0.01$.

Results 2D relaxation II

- **Test I**: Acoustic wave for isothermal Euler equation.

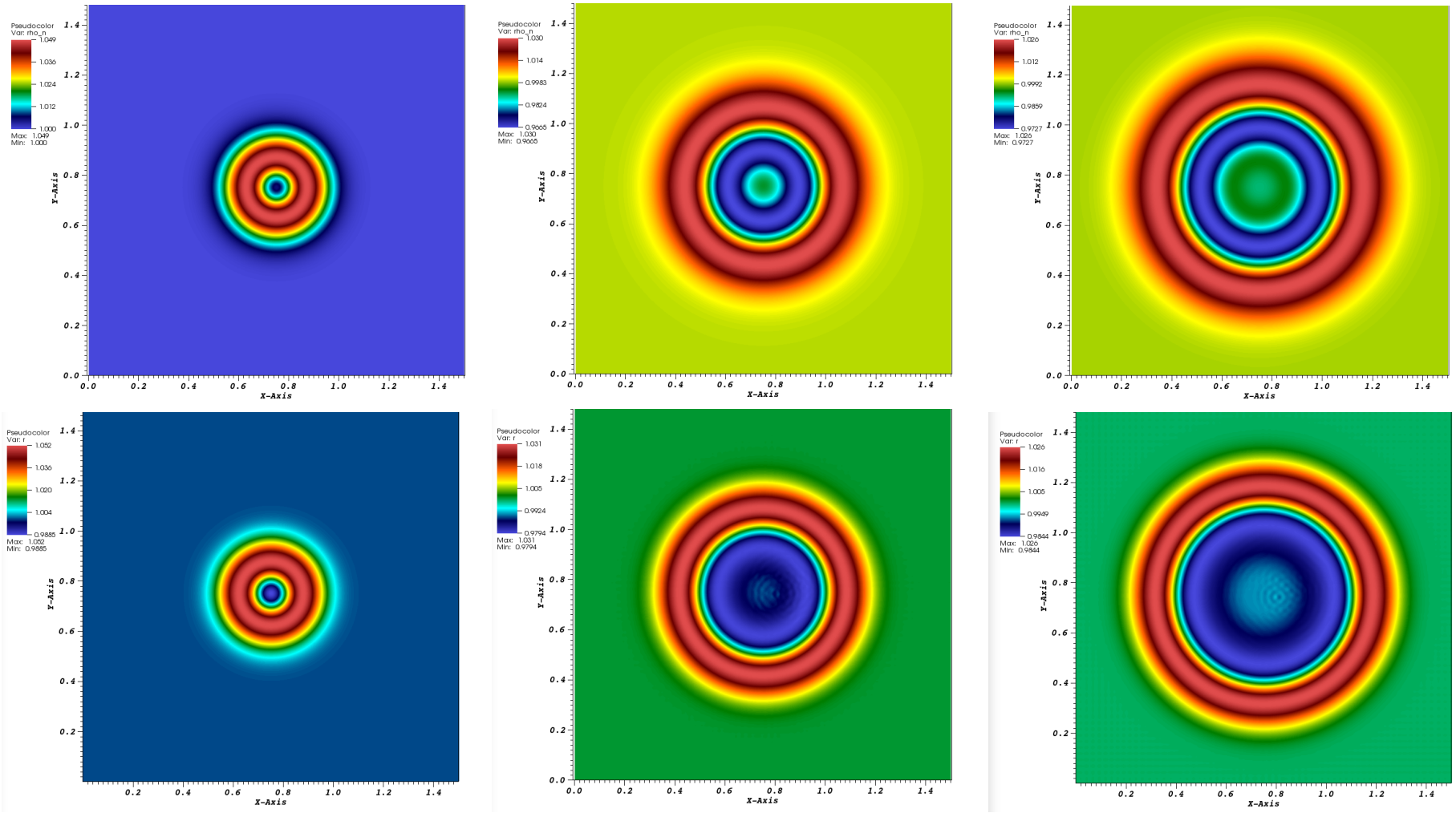


Figure: Comparison between the CN coupled with Newton method (top) and the relaxation (bottom) for a time step $\Delta t = 0.05$.

Results 2D relaxation II

- **Test I**: Acoustic wave for isothermal Euler equation.

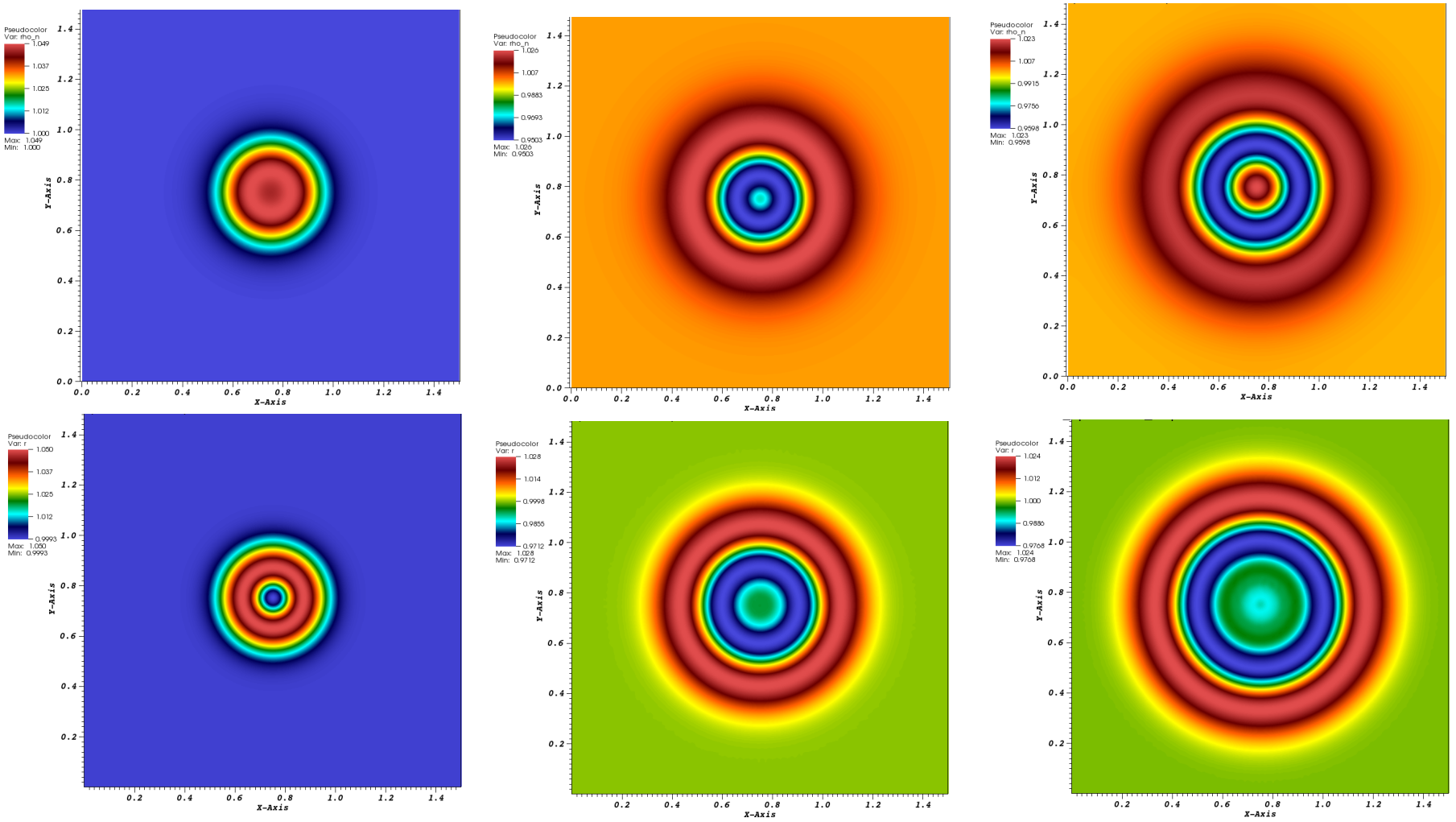


Figure: Comparison between the CN coupled with Newton method (top) and the relaxation (bottom) for a time step $\Delta t = 0.1$.

Results 2D relaxation II

- **Test I**: Acoustic wave for isothermal Euler equation.

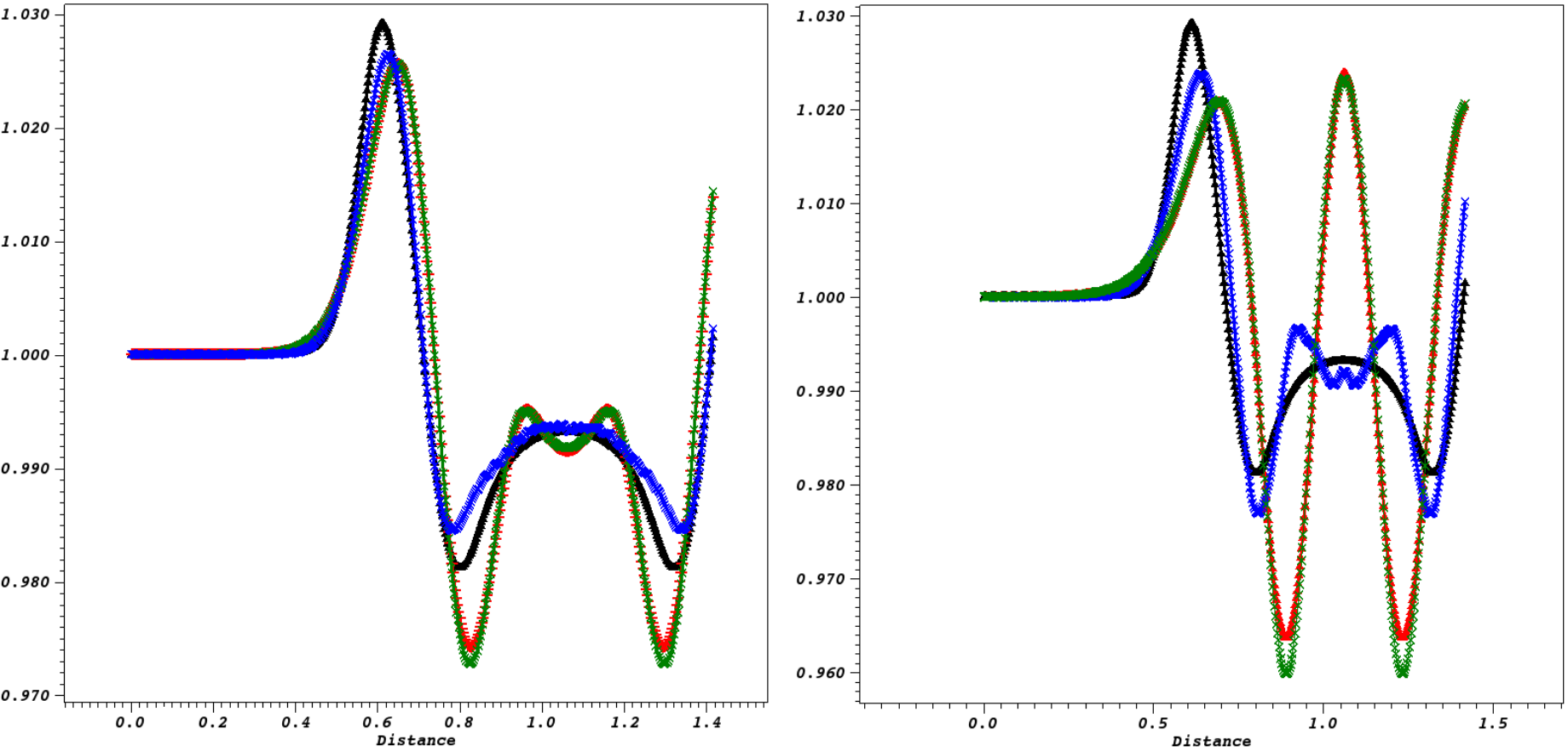
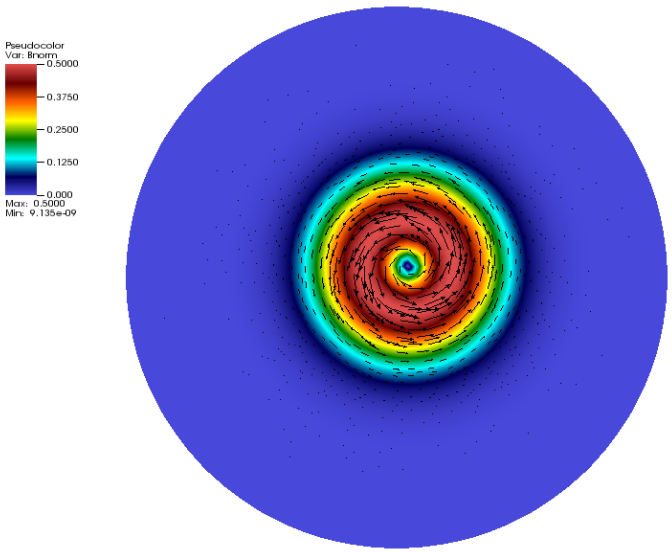


Figure: 1D cut. Fine solution (black), CN method (red), Newton (green) and relaxation (blue). $\Delta t = 0.05$ (left) and $\Delta t = 0.1$ (right)

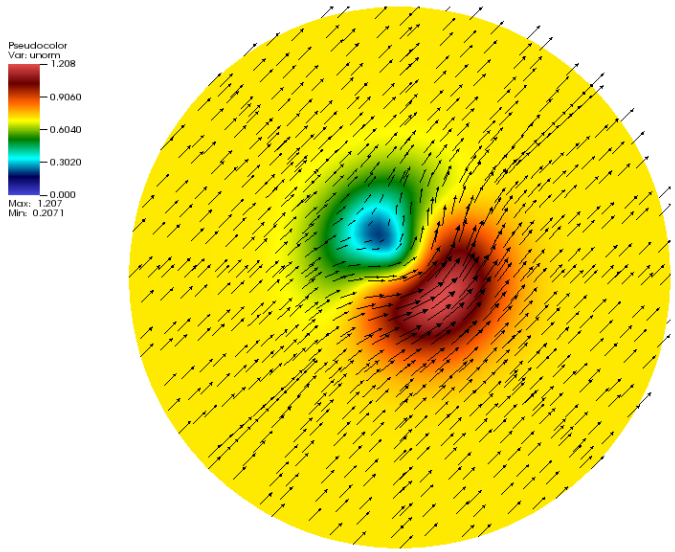
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



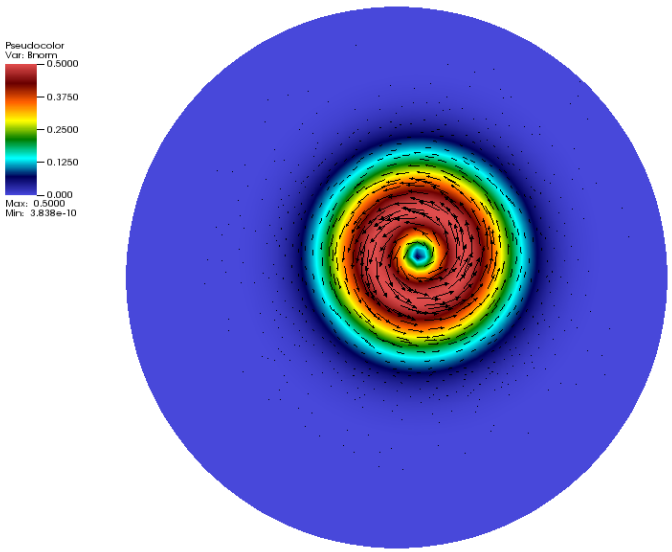
Velocity



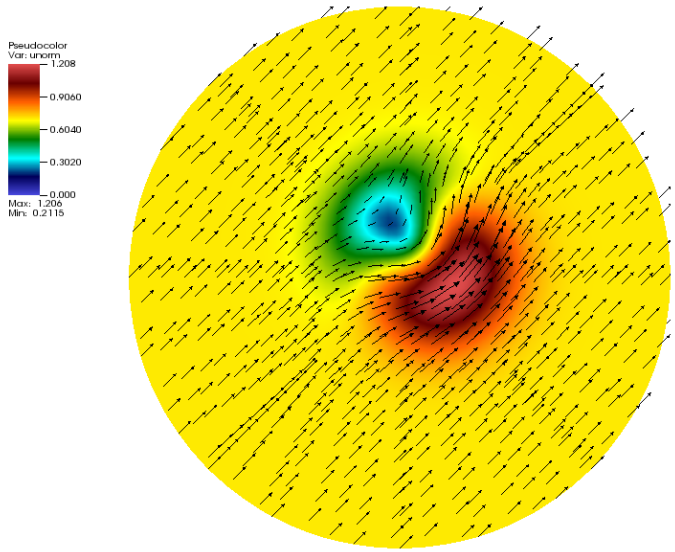
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



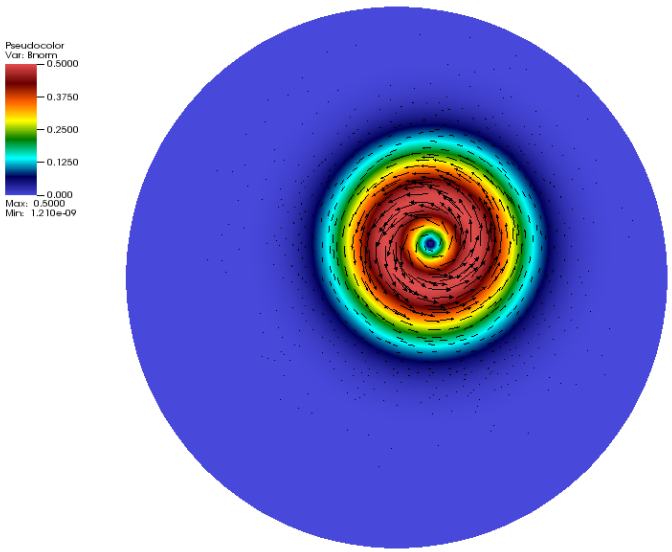
Velocity



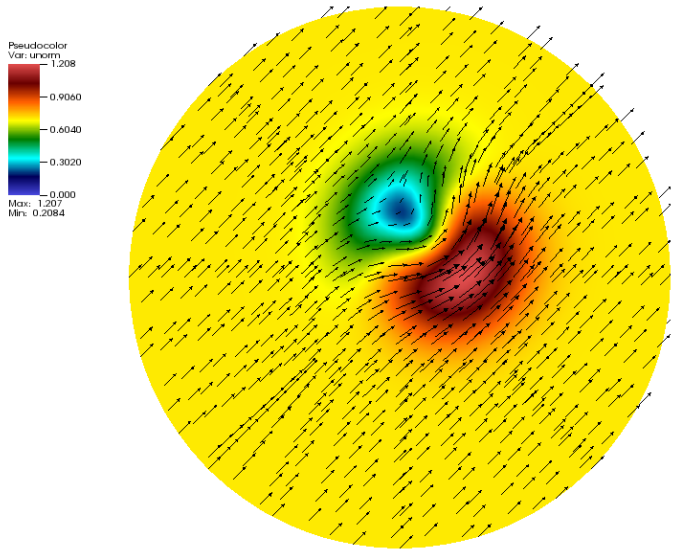
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



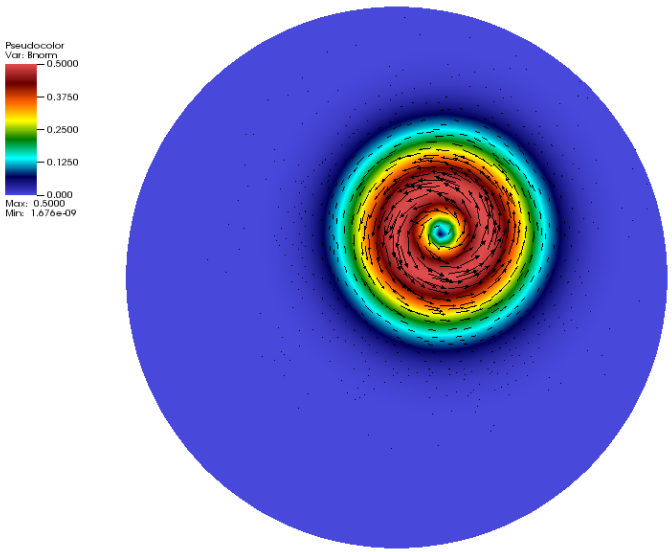
Velocity



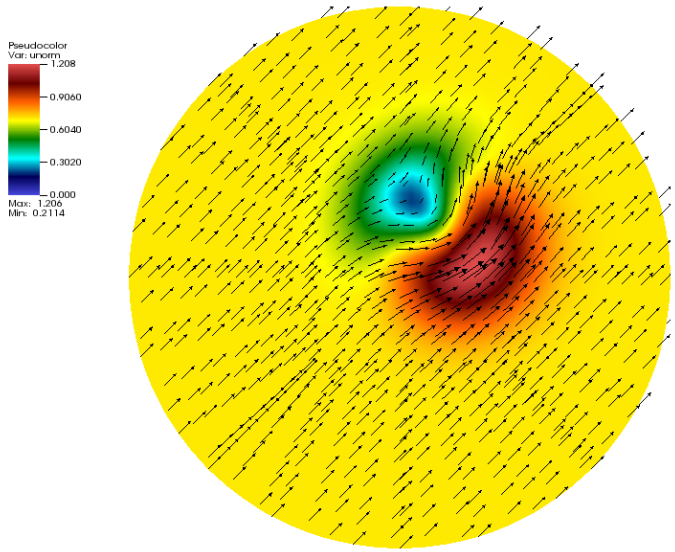
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



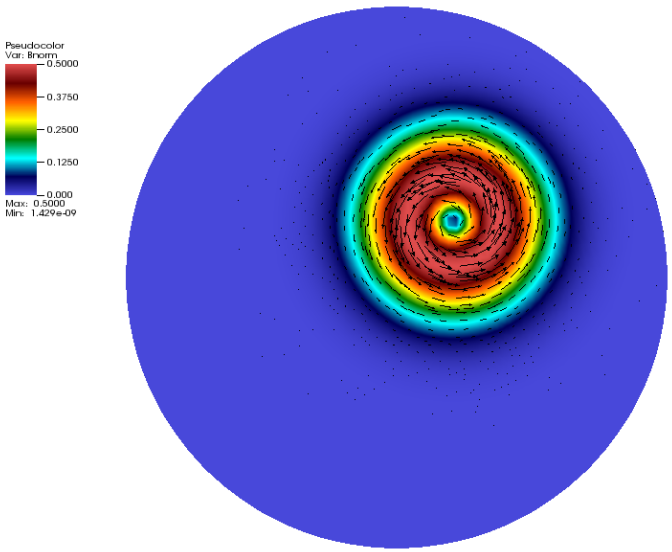
Velocity



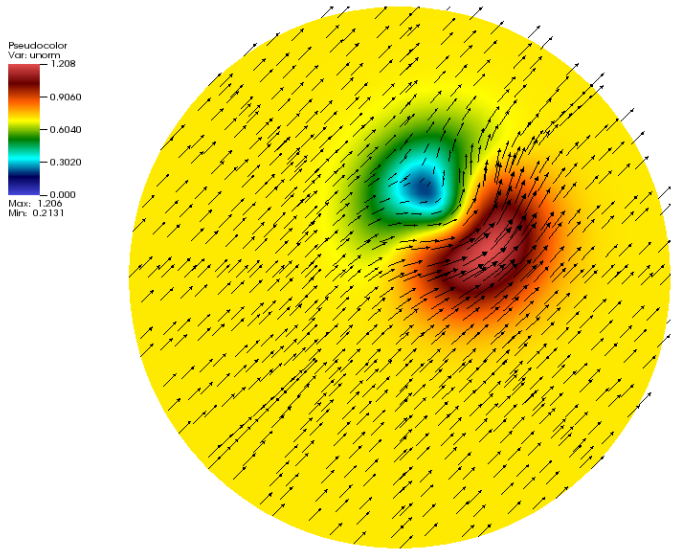
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



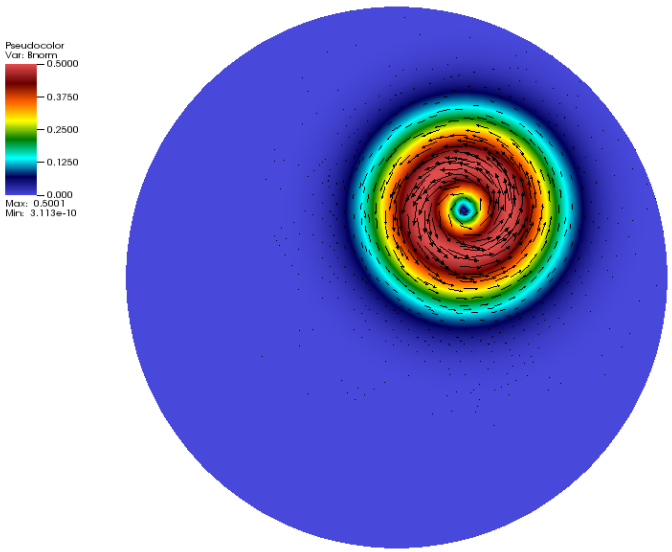
Velocity



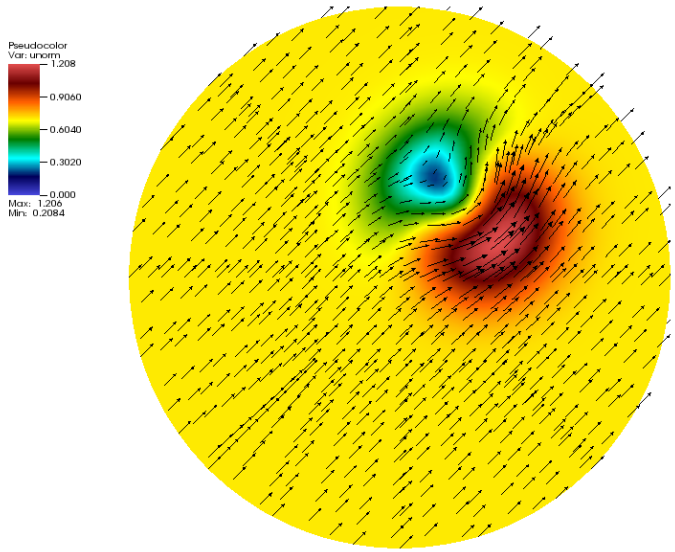
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



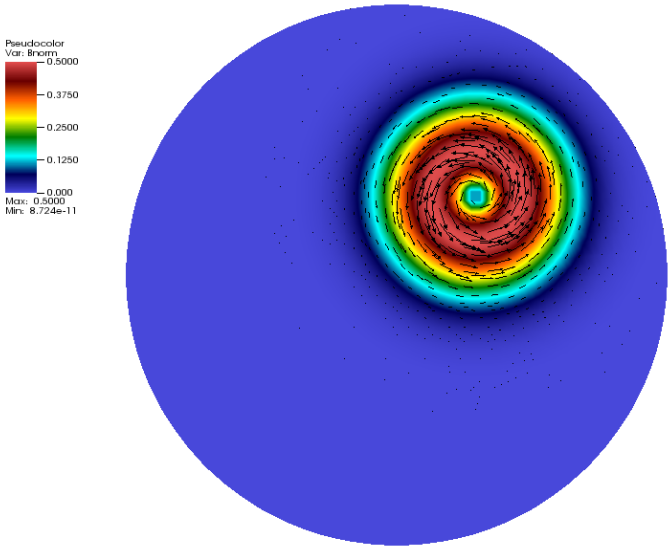
Velocity



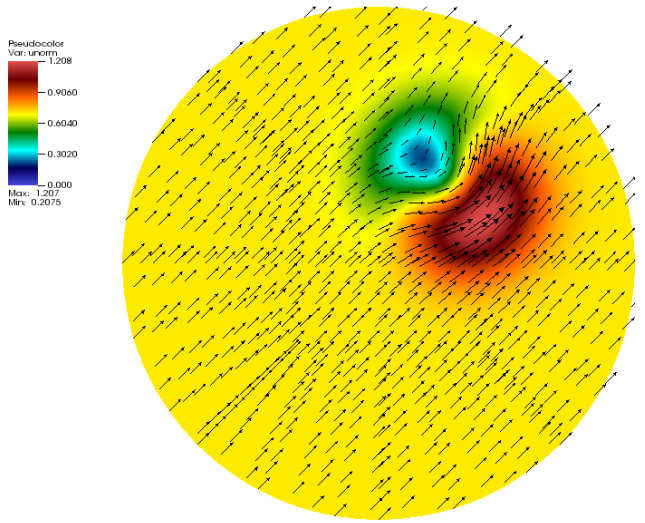
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2-Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1,1]^t$, $h(r) = exp[(1-r^2)/2]$

Magnetic field



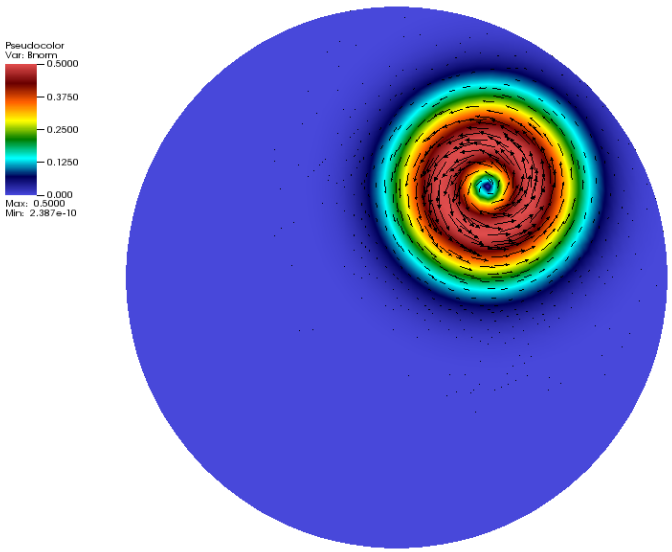
Velocity



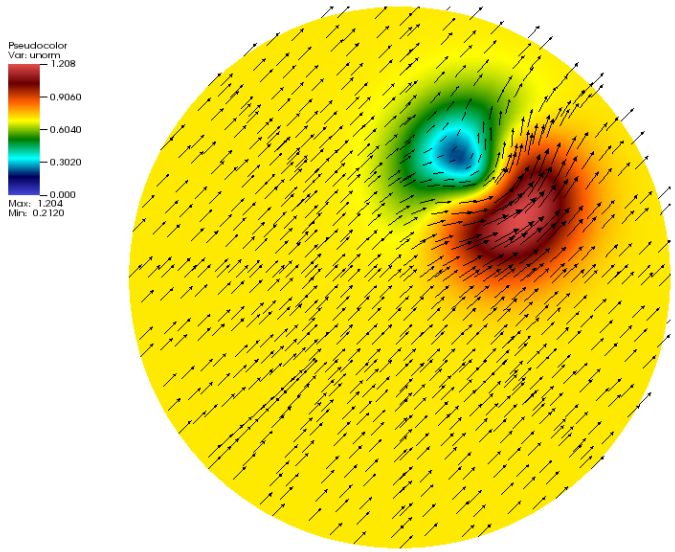
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



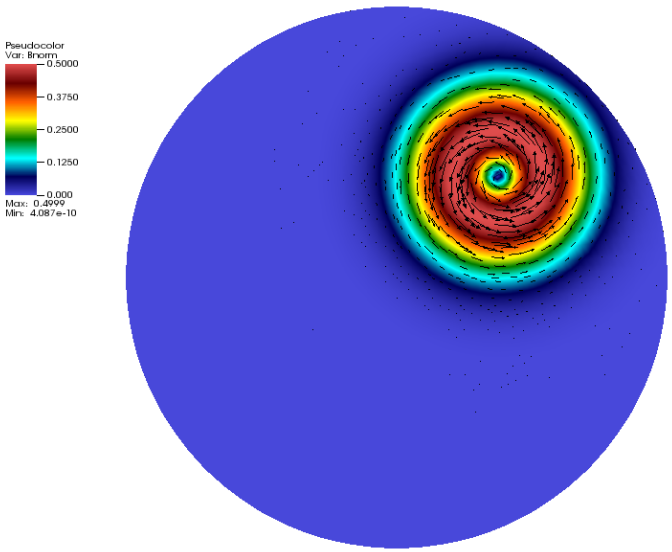
Velocity



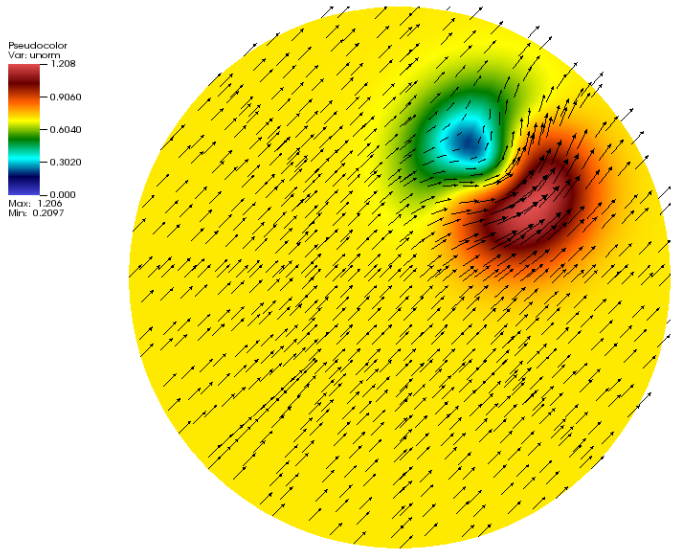
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2-Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1,1]^t$, $h(r) = exp[(1-r^2)/2]$

Magnetic field



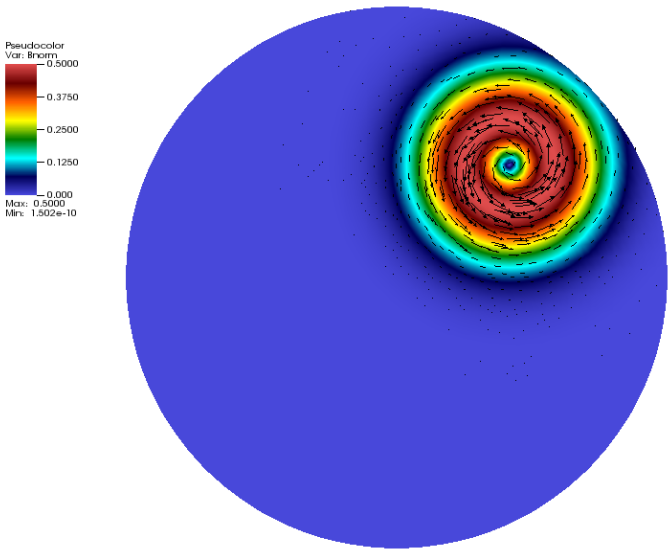
Velocity



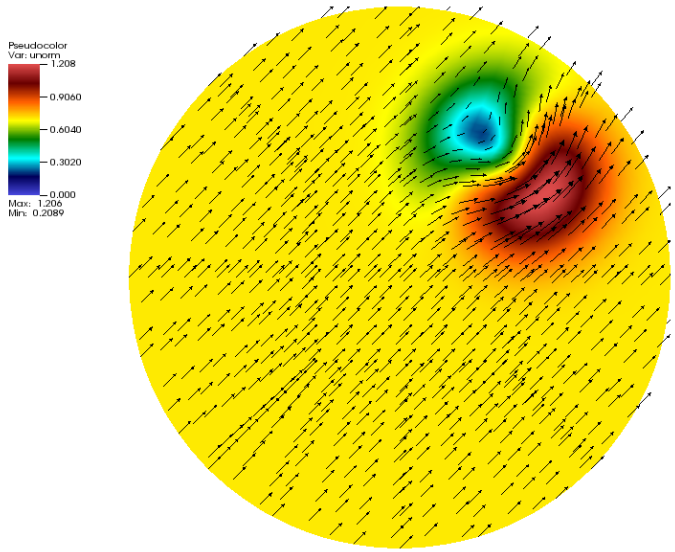
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2-Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1,1]^t$, $h(r) = exp[(1-r^2)/2]$

Magnetic field



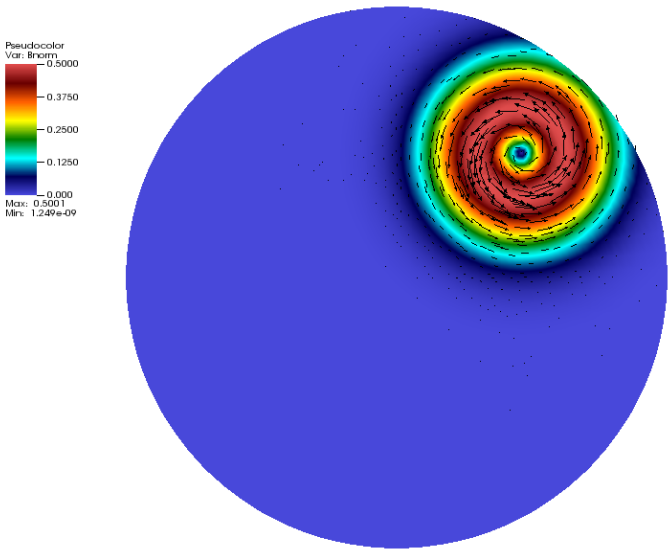
Velocity



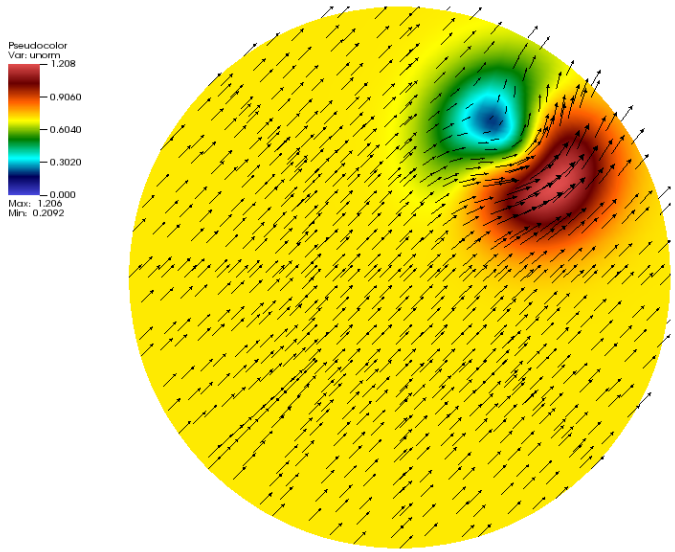
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



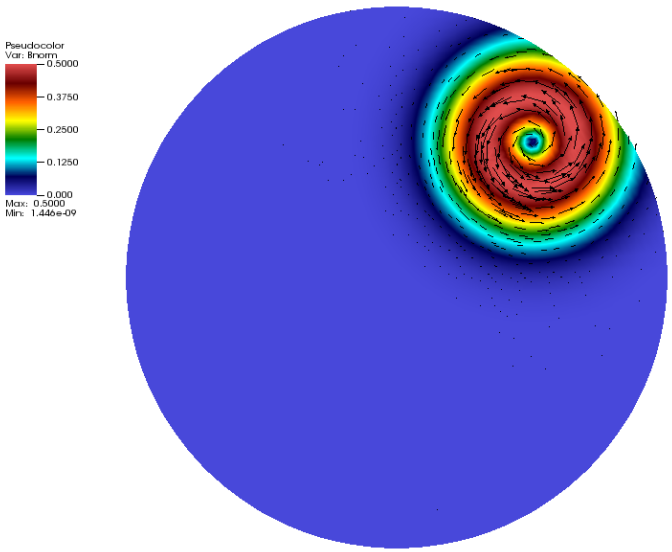
Velocity



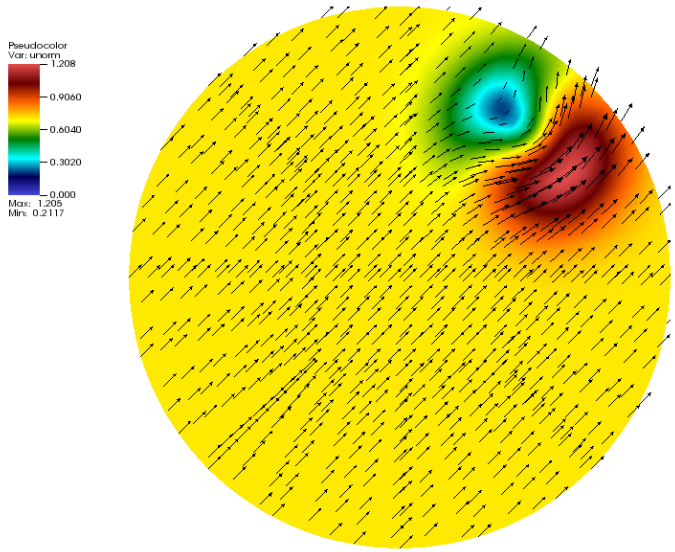
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



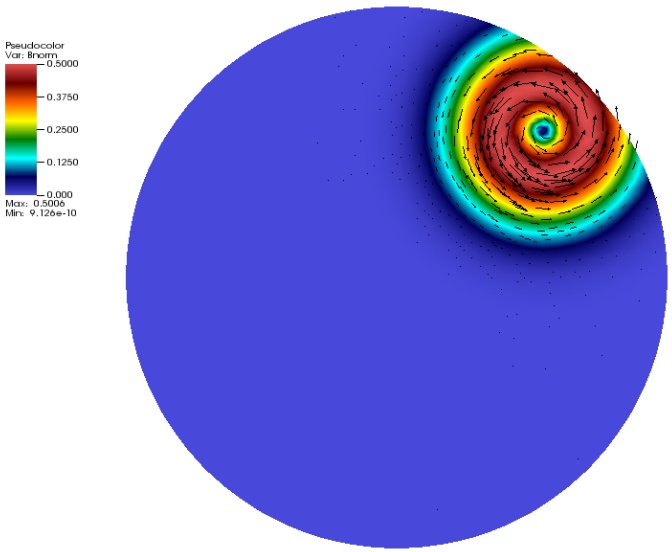
Velocity



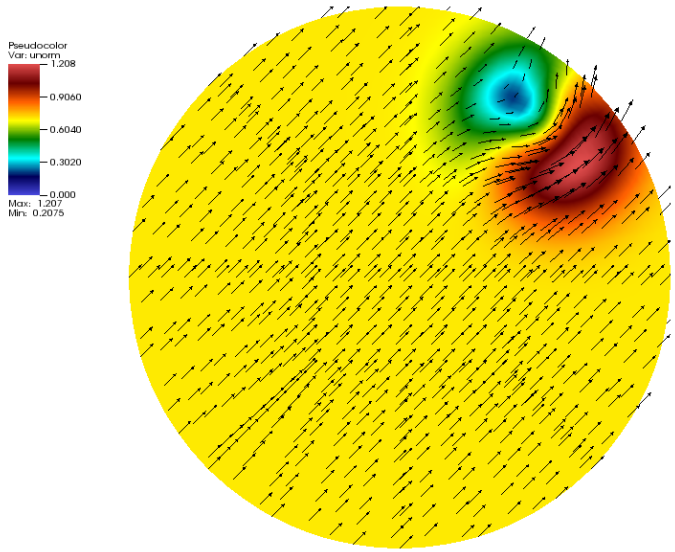
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2-Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1,1]^t$, $h(r) = exp[(1-r^2)/2]$

Magnetic field



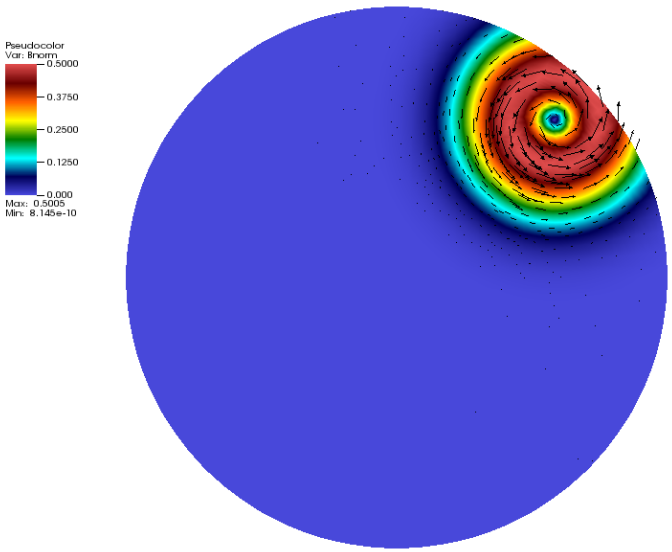
Velocity



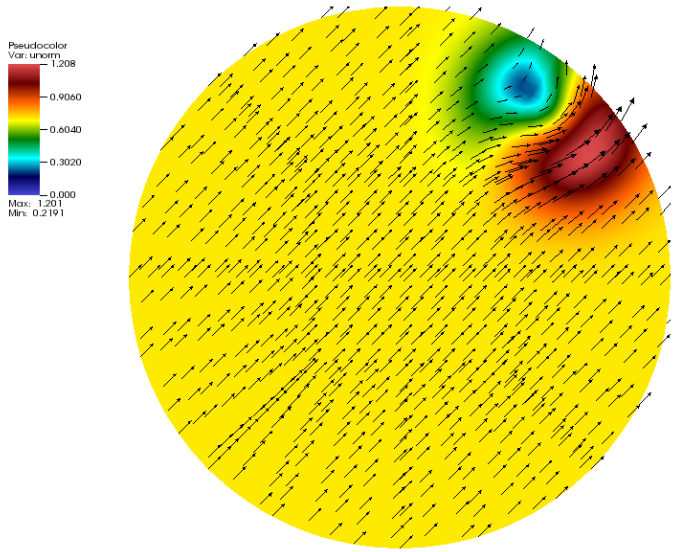
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2-Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1,1]^t$, $h(r) = exp[(1-r^2)/2]$

Magnetic field



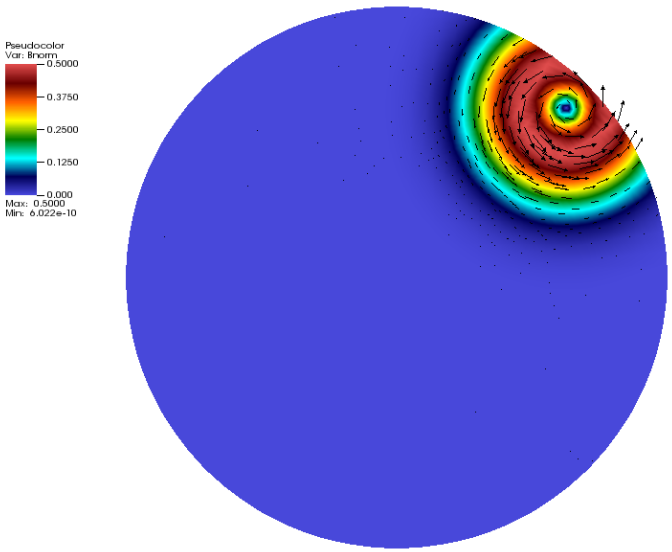
Velocity



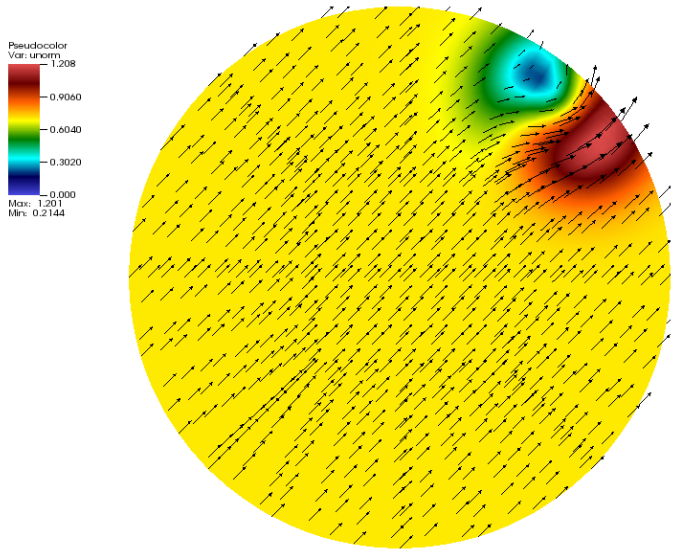
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2-Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1,1]^t$, $h(r) = exp[(1-r^2)/2]$

Magnetic field



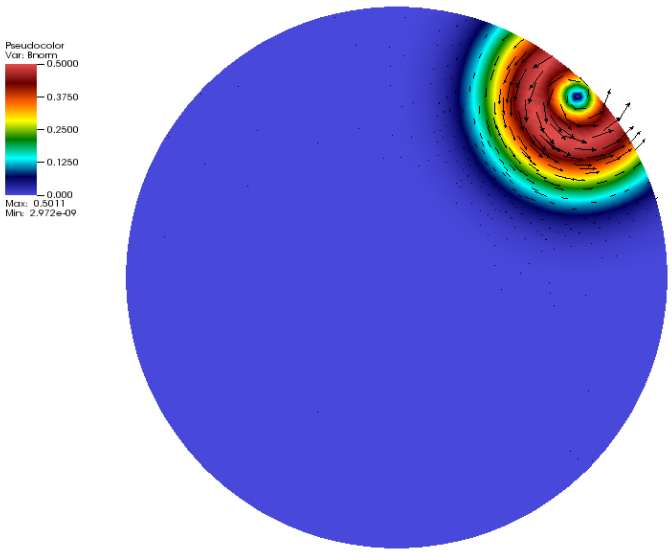
Velocity



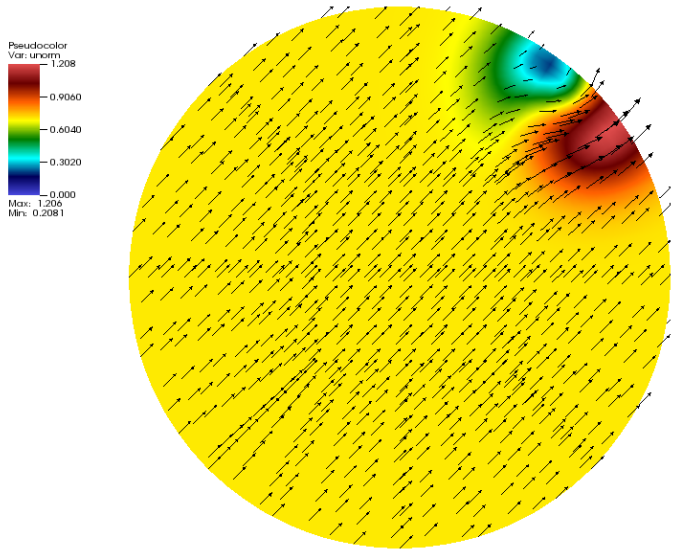
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



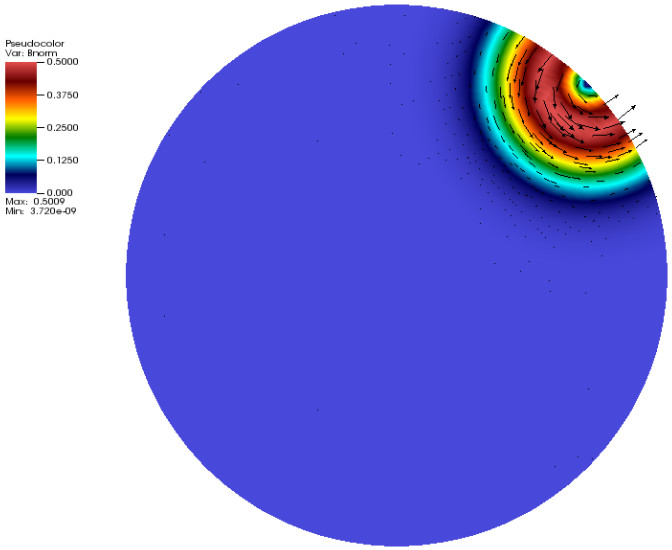
Velocity



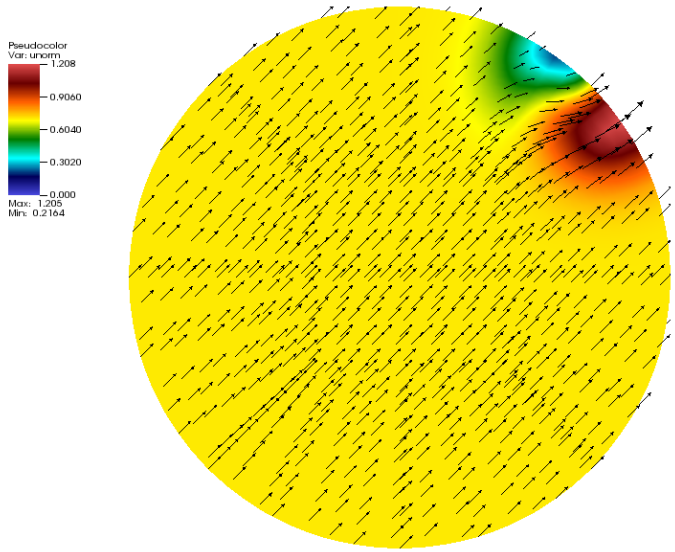
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field



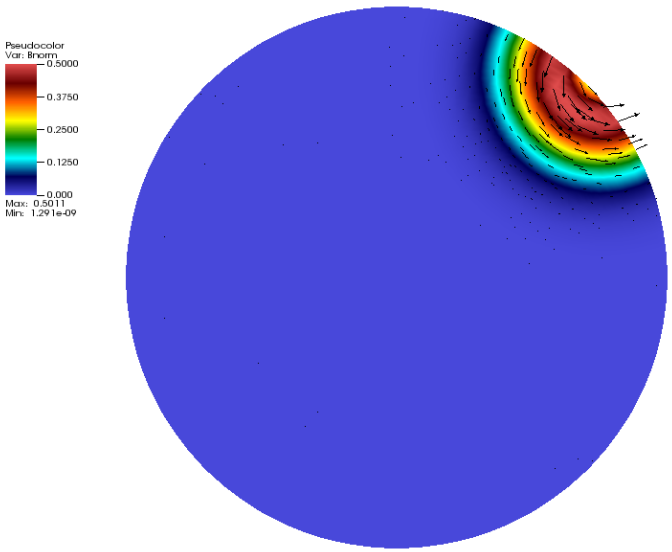
Velocity



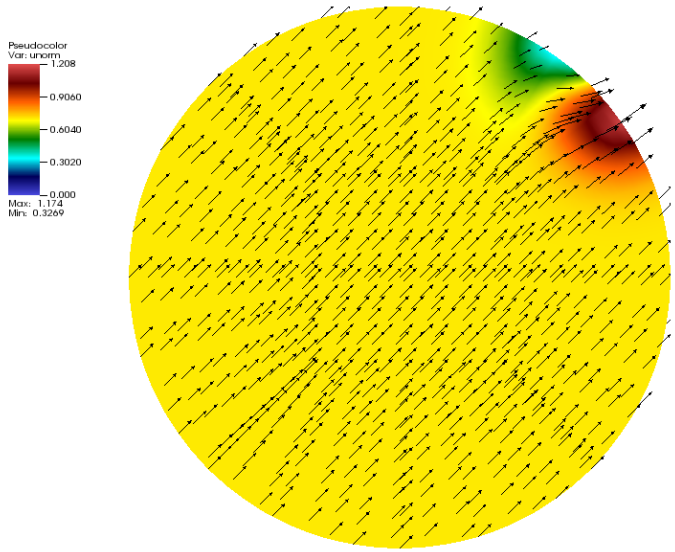
2D MHD drifting vortex

- **Model** : compressible ideal MHD.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $(D2 - Q4)^n$. Symmetric Lattice.
- **Test case** : advection of the vortex (steady state without drift).
- Parameters : $\rho = 1.0$, $p_0 = 1$, $u_0 = b_0 = 0.5$, $u_{drift} = [1, 1]^t$, $h(r) = exp[(1 - r^2)/2]$

Magnetic field

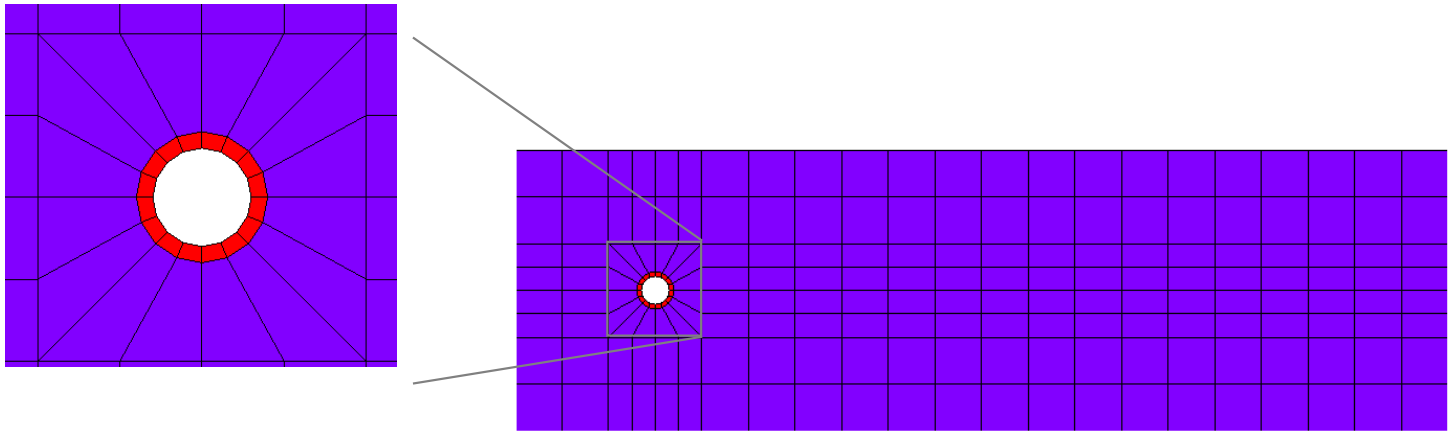


Velocity



2D flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2-Q9$ lattice (specific lattice for this model).
- No-slip ($u=0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i+f_{\bar{i}})$ where $v_{\bar{i}}=-v_i$.
  - CN scheme and ($\tau=0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.



$2D$ flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2-Q9$ lattice (specific lattice for this model).
- No-slip ($u=0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i+f_{\bar{i}})$ where $v_{\bar{i}}=-v_i$.
  - CN scheme and ($\tau=0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.

$2D$ flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2 - Q9$ lattice (specific lattice for this model).
- No-slip ($u = 0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i + f_{\bar{i}})$ where $v_{\bar{i}} = -v_i$.
  - CN scheme and ($\tau = 0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.

$2D$ flow around a cylinder

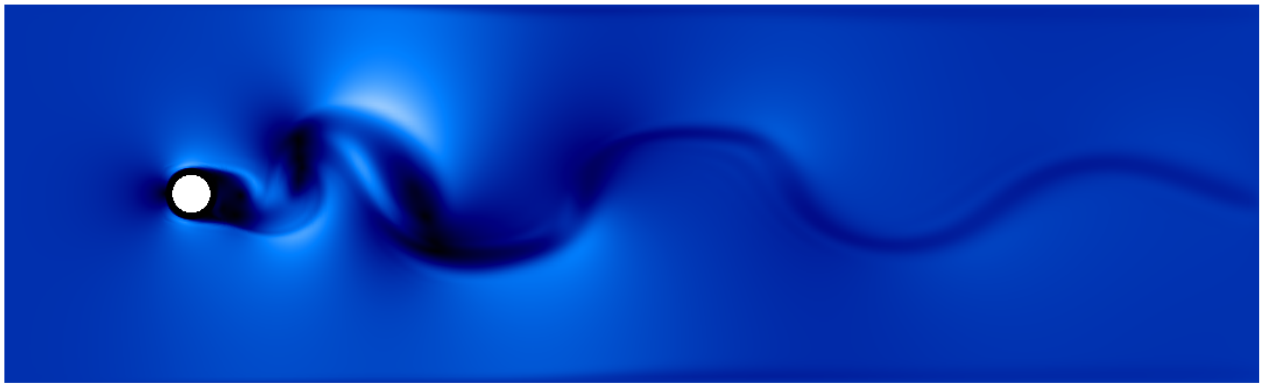
- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2 - Q9$ lattice (specific lattice for this model).
- No-slip ($u = 0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i + f_{\bar{i}})$ where $v_{\bar{i}} = -v_i$.
  - CN scheme and ($\tau = 0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.

$2D$ flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2 - Q9$ lattice (specific lattice for this model).
- No-slip ($u = 0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i + f_{\bar{i}})$ where $v_{\bar{i}} = -v_i$.
  - CN scheme and ($\tau = 0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.

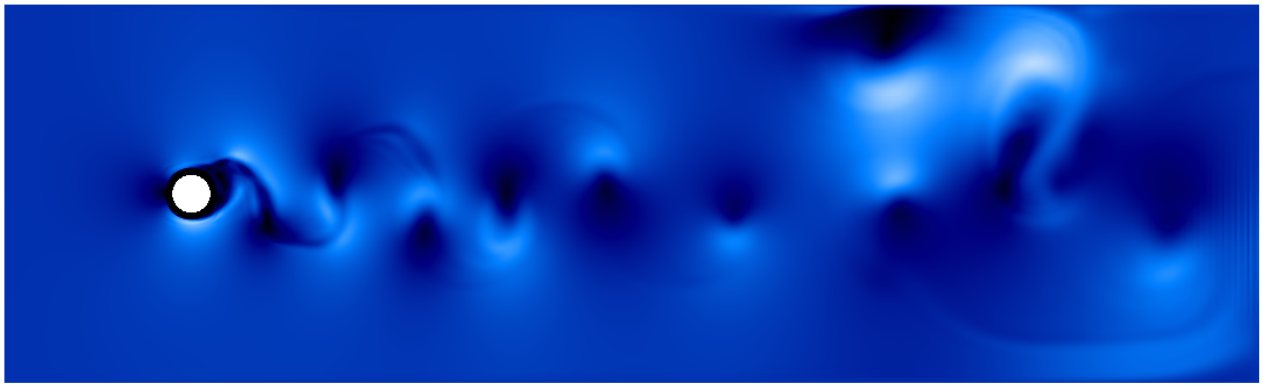
$2D$ flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2 - Q9$ lattice (specific lattice for this model).
- No-slip ($u = 0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i + f_{\bar{i}})$ where $v_{\bar{i}} = -v_i$.
  - CN scheme and ($\tau = 0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.



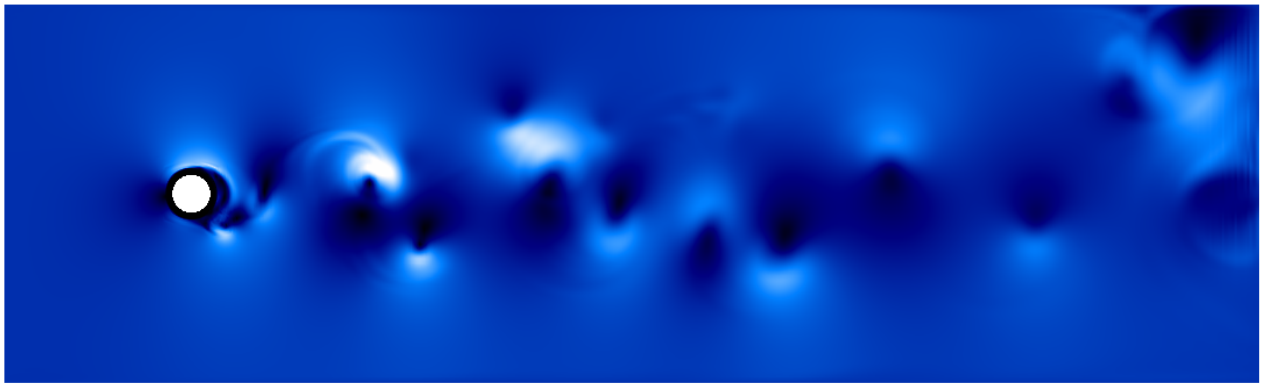
$2D$ flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2 - Q9$ lattice (specific lattice for this model).
- No-slip ($u = 0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i + f_{\bar{i}})$ where $v_{\bar{i}} = -v_i$.
  - CN scheme and ($\tau = 0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.



$2D$ flow around a cylinder

- **Model** : compressible isothermal Euler.
- **Relaxation** : Kinetic relaxation using Schnaps.
- **Lattice** : $D2 - Q9$ lattice (specific lattice for this model).
- No-slip ($u = 0$) condition on the obstacle imposed using a penalization method in a small volume (red ring)
  - Relaxation of each f_i towards $0.5(f_i + f_{\bar{i}})$ where $v_{\bar{i}} = -v_i$.
  - CN scheme and ($\tau = 0$) $\rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- Imposed state at boundaries with a constant low Mach flow.



On going and future works

MHD and reduced MHD

Aim

- Apply these methods to the MHD in Tokamak.
- **MHD Tokamak**: low-Beta, low-Mach in the perpendicular direction, anisotropic heat diffusion.
- **Hierarchy of geometry**: 2D circle, 3D cylinder, 3D Torus, Field - Aligned 3D torus.

- **First model** (in 2D circle or 3D cylinder geometries):

$$\begin{cases} \partial_t \boldsymbol{v}_\perp + \boldsymbol{v}_\perp \cdot \nabla \boldsymbol{v}_\perp = \boldsymbol{B}_\perp \cdot \nabla \boldsymbol{B}_\perp + \partial_Z \boldsymbol{B}_\perp + \nu \Delta \boldsymbol{v}_\perp \\ \partial_t \boldsymbol{B}_\perp + \boldsymbol{v}_\perp \cdot \nabla \boldsymbol{B}_\perp = \boldsymbol{B}_\perp \cdot \nabla \boldsymbol{v}_\perp + \partial_Z \boldsymbol{v}_\perp + \eta(\Delta \boldsymbol{B}_\perp - j_c) \\ \nabla_\perp \cdot \boldsymbol{v}_\perp = 0 \\ \nabla_\perp \cdot \boldsymbol{B}_\perp = 0 \end{cases}$$

- **Following models**: same reduced models in Torus geometry (cylindrical coordinate) and additional parallel velocity.

- **Last model**: full compressible MHD model.

Numerical issue

- Divergence free constraint preserving method.

Diffusion

Issue for large diffusion terms

- Limit equation of the relaxation scheme:

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \nu \partial_x (D(\boldsymbol{U}) \nabla \boldsymbol{U}) + o(\nu^2)$$

- If $\nu = O(1)$ we cannot apply the method.

Diffusion

- Propose relaxation for a diffusion equation (main point the nonlinearity must be local).
- Model:

$$\partial_t \rho - \partial_x (D(\rho) \nabla \rho) = f$$

Baby MHD model

- Propose relaxation for a baby model with the additional difficulties linked to the MHD
- Model:

$$\begin{cases} \partial_t \boldsymbol{B} + \nabla \times \left(\boldsymbol{u} \times \boldsymbol{B} + \frac{1}{\rho_0} \nabla T \right) = \eta \nabla \times (\nabla \times \boldsymbol{B}) \\ \partial_t T - \nabla \cdot \left((k_{\parallel} - k_{\perp})(\boldsymbol{b} \otimes \boldsymbol{b}) \nabla T + k_{\perp} \nabla T \right) = 0 \end{cases}$$

- **Difficulties**: anisotropic diffusion coupled with divergence free constraint.

Equilibrium and other application

Well - Balanced property

- **Preserve** steady solution which comes from of the balance between physical forces.
- **Tokamak**: essential to conserve the magnetic equilibrium.

Numerical issues and future

- Spatial and time errors generate a drift of the steady state.
- The spatial error is assumed small (fine grids, high order in space).
- The time error is not small (large time step). The problem comes from the decoupling of the equations.
- Post-doc of C. Courtes (October 2017) on the subject.

Other work: Full semi-Lagrangian solver for

$$\begin{cases} \partial_t f + v\partial_x f + \partial_v(D(u-v)f) = \nu\partial_{vv} f \\ \\ \partial_t \rho + \partial_x(\rho u) = 0 \\ \partial_t(\rho u) + \partial_x\left(\rho u^2 + \dfrac{p}{\rho_f}\right) = \dfrac{m_d}{\rho_f}\displaystyle\int_v D(v-u)f + \eta\partial_{xx} u \end{cases}$$

Conclusion

Conclusion

Resume

- **Relaxation**: we linearize and decouple the waves overestimating the transport.
- Model error gives the physical diffusion. Numerical error is controlled by high order schemes.

Advantages

- Additional parallelism between the equations.
- The implicit systems are simple to invert.
- Small storage and no matrix assembly at each time.
- High order schemes for small diffusion.

Defaults

- High diffusion, BC and equilibrium preservation more complex.
- Stability depends on the parameter α. Choice of this parameter is essential.
- Less simple for users to add terms or sources.