CFL-less explicit schemes for conservation laws based on a kinetic approach

Philippe Helluy

Université de Strasbourg, CNRS, Inria

"Structure preserving numerical methods for hyperbolic equations", 13th October, 2020.

Motivations

Hyperbolic system of conservation laws

$$\partial_t \mathbf{w} + \sum_{k=1}^{D} \partial_k \mathbf{q}^k(\mathbf{w}) = 0,$$

- space variable: $\mathbf{x} \in \mathbb{R}^D$, time: $t \in [0, T]$.
- unknown: $\mathbf{w}(\mathbf{x}, t) \in \mathbb{R}^m$.
- conservative flux: $\mathbf{q}(\mathbf{w}) \cdot \mathbf{n} = \sum_{k=1}^{D} \mathbf{q}^k(\mathbf{w}) n_k$.
- hyperbolic: $\mathbf{A}(\mathbf{w}, \mathbf{n}) = D_{\mathbf{w}} q(\mathbf{w}) \cdot \mathbf{n}$ has real eigenvalues λ_i, $i = 1 \dots m$.
- Discontinuities, turbulence.
- Explicit numerical schemes are constrained by a stability condition $\Delta t \leq \Delta x / \lambda$, with $\lambda = \max_i |\lambda_i|$.
- Issues: numerical precision, long time behavior, oscillations, ...

Example I: Two-phase flow model

Fluid of density ρ, velocity $\mathbf{u}$, pressure p. Color function ϕ: $\phi = 0$ in the liquid and $\phi = 1$ in the gas.

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho\varphi \end{pmatrix}, \quad \mathbf{q}(\mathbf{w})\cdot\mathbf{n} = \begin{pmatrix} \rho\mathbf{u}\cdot\mathbf{n} \\ \rho(\mathbf{u}\cdot\mathbf{n})\mathbf{u} + p\mathbf{n} \\ \rho\phi\mathbf{u}\cdot\mathbf{n} \end{pmatrix}. \tag{1}$$

The pressure is a function of ρ and ϕ, $p = \pi(\rho, \phi)$.

Example II: MagnetoHydroDynamics (MHD)

Magnetized fluid of density ρ, velocity $\mathbf{u}$, energy Q, magnetic field $\mathbf{B}$, pressure p, internal energy e.

$$\partial_t \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ Q \\ \mathbf{B} \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + (p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{I} - \mathbf{B}\otimes\mathbf{B} \\ (Q + p + \frac{\mathbf{B}\cdot\mathbf{B}}{2})\mathbf{u} - (\mathbf{B}\cdot\mathbf{u})\mathbf{B} \\ \mathbf{u}\otimes\mathbf{B} - \mathbf{B}\otimes\mathbf{u} \end{pmatrix} = 0, \qquad (2)$$

$$Q = \rho e + \rho\frac{\mathbf{u}\cdot\mathbf{u}}{2} + \frac{\mathbf{B}\cdot\mathbf{B}}{2}.$$

$$p = \pi(\rho, e) = (\gamma - 1)\rho e, \quad \gamma > 1.$$

Demo: Orszag-Tang vortex

Introduction

Construction of a general numerical method:

- that works for any hyperbolic system of conservation laws;
- explicit;
- unconditionally stable;
- of arbitrary order;
- low-storage (stores only one time-step of the solution);
- naturally parallel.

Outlines

The construction is based on a generalization of the Lattice Boltzmann Method (LBM)

Upwind transport solver

Vectorial kinetic representation

Structure preserving time integration

Implementation

Upwind transport solver

1D transport equation

$f(x,t)$ solution of the simple transport equation at **constant** velocity $v > 0$.

$$\partial_t f + v\partial_x f = 0, \quad v > 0.$$

Approximation by an "implicit" upwind finite volume method.
$f_i^p \simeq f(i\Delta x, p\Delta t)$.

$$\frac{f_i^p - f_i^{p-1}}{\Delta t} + v\frac{f_i^p - f_{i-1}^p}{\Delta x} = 0.$$

The scheme is stable for arbitrary CFL number $\beta = \frac{v\Delta t}{\Delta x}$.
Actually, it is explicit: f_0^p is given by the left boundary condition and

$$f_i^p = \frac{f_i^{p-1} + \beta f_{i-1}^p}{1+\beta}, \quad i = 1, 2, \ldots$$

Low-storage: Only one time-step of the solution is stored and directly updated in memory.

Higher order in x: Discontinuous Galerkin (DG)

- N_x cells of size h. In cell C_ℓ, we consider Gauss-Lobatto points $x_{\ell,i}$.
- DG basis function $\varphi_{\ell,i}$ with support in cell ℓ.
- interpolation property $\varphi_{\ell,i}(x_{\ell,j}) = \delta_{ij}$.
- Expansion on the DG basis
  $f(x,t) \simeq f_h(x,t) = \sum_{j=0}^{d} f_{\ell,j}(t)\varphi_{\ell,j}(x), \quad x \in C_\ell.$
- DG formulation for $v > 0$: for all cell C_ℓ and all test function $\varphi_{\ell,i}$

$$\int_{C_\ell} (\partial_t f_h + v\partial_x f_h)\varphi_{\ell,i} + v\left(f_{\ell,0} - f_{\ell-1,d}\right)\varphi_{\ell,i}(x_{\ell,0}) = 0. \tag{3}$$

We obtain a set of linear differential equations

$$\partial_t \mathbf{f}_h + \mathbf{L}_h \mathbf{f}_h = 0, \tag{4}$$

where $\mathbf{L}_h$ is a **lower block-triangular** matrix because of the upwind flux. Here also, the "implicit" scheme is explicit.

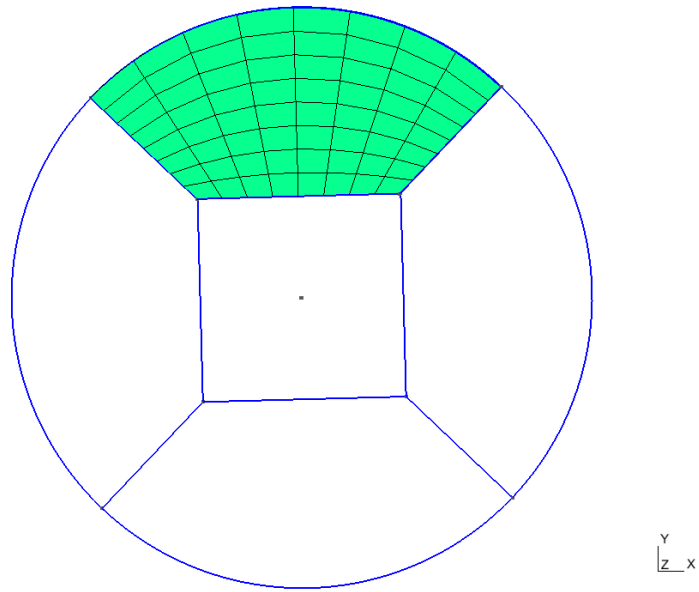
Extension to dimension $D = 2$ or 3

$f(\mathbf{x}, t)$ solution of the transport equation

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f = 0, \quad \mathbf{x} \in \mathbb{R}^D.$$

Arbitrary mesh. (We use meshes made of hexahedral blocks, but it is not necessary). The velocity $\mathbf{v}$ is **constant**.

- In each cell L we consider polynomial basis functions ψ_k^L of degree d.
- Expansion on the polynomial basis: discontinuous approximation of f.
- Possible non-conformity in "h" and "d".



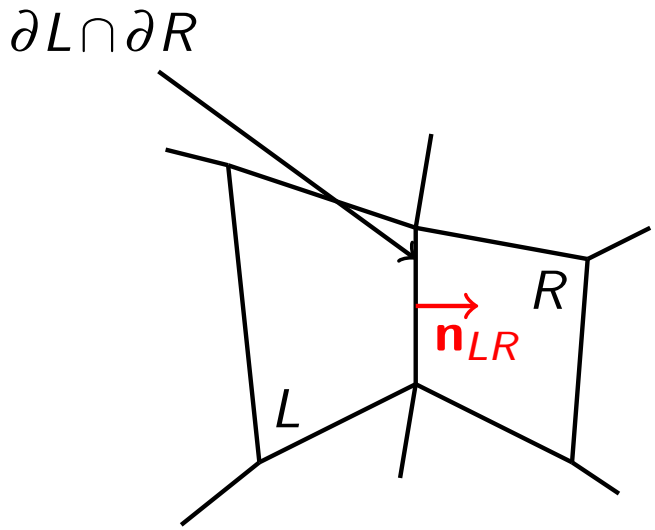
$$f(\mathbf{x}, p\Delta t) \simeq f_L^p(\mathbf{x}) = \sum_k f_{L,k}^p \psi_k^L(\mathbf{x}), \quad \mathbf{x} \in L.$$

DG approximation

Implicit DG approximation scheme: $\forall L, \forall k$

$$\int_L \frac{f_L^p - f_L^{p-1}}{\Delta t} \psi_k^L - \int_L \mathbf{v} \cdot \nabla \psi_k^L f_L^p + \int_{\partial L} \left(\mathbf{v} \cdot \mathbf{n}^+ f_L^p + \mathbf{v} \cdot \mathbf{n}^- f_R^p \right) \psi_k^L = 0.$$

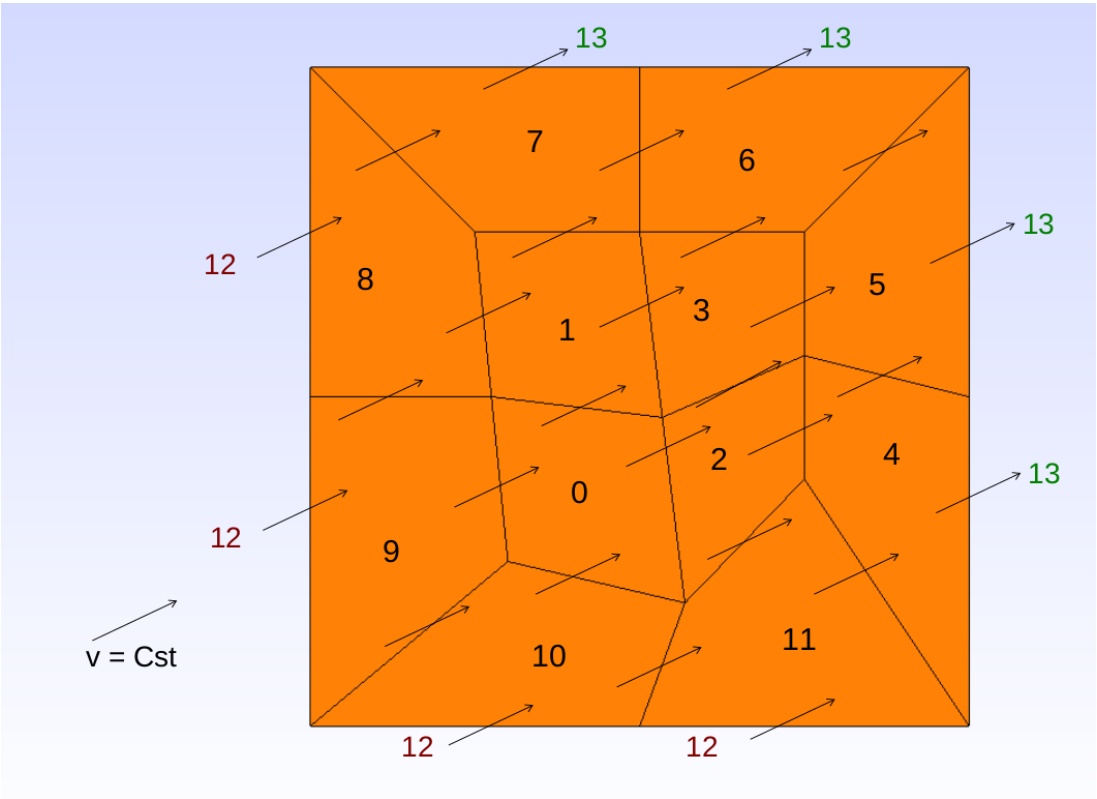
- time step index: p
- R denotes the neighbor cells along ∂L.
- $\mathbf{v} \cdot \mathbf{n}^+ = \max(\mathbf{v} \cdot \mathbf{n}, 0)$, $\mathbf{v} \cdot \mathbf{n}^- = \min(\mathbf{v} \cdot \mathbf{n}, 0)$.
- $\mathbf{n}_{LR}$ is the unit normal vector on ∂L oriented from L to R.



Computing f^p from f^{p-1} requires the resolution of a large linear system. But here again the matrix has a block-triangular structure that makes the scheme explicit.

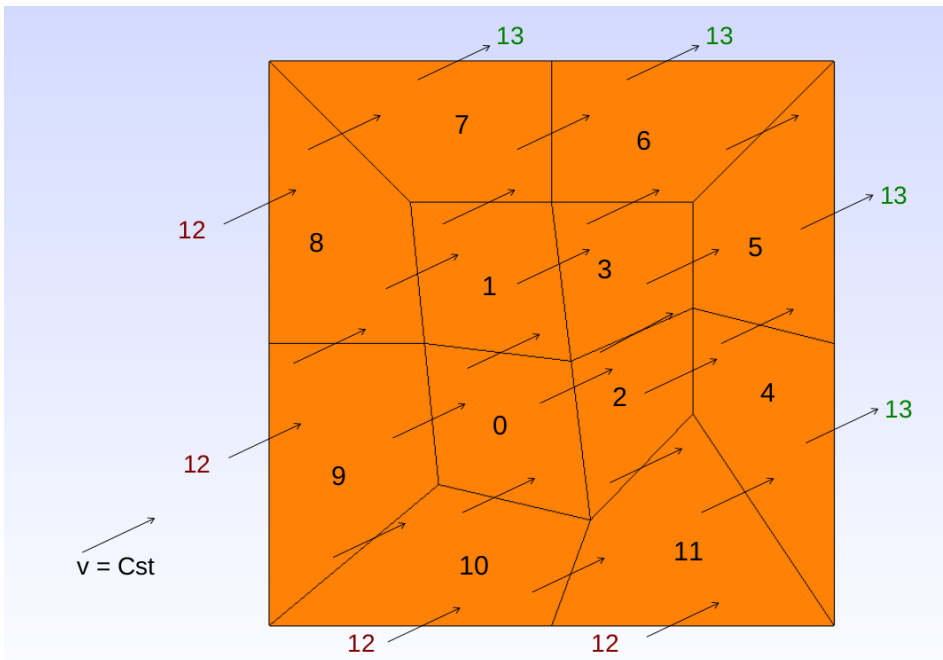
Upwind numbering

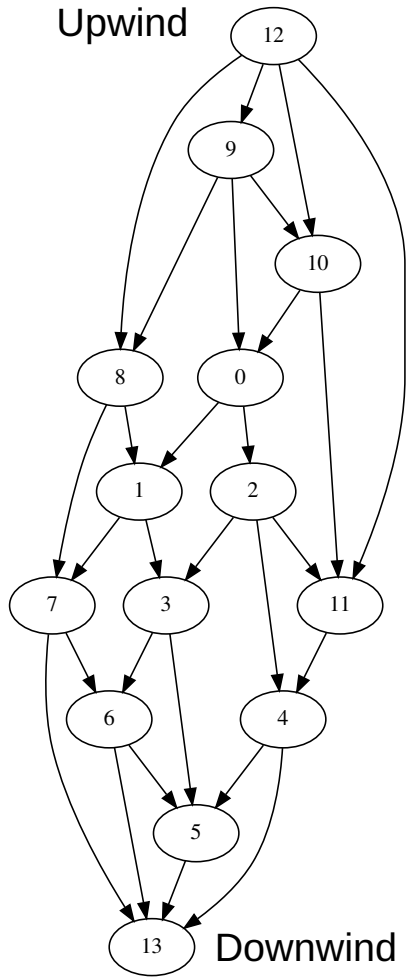
- L is *upwind* with respect to R if $\mathbf{v} \cdot \mathbf{n}_{LR} > 0$ on $\partial L \cap \partial R$: the velocity $\mathbf{v}$ cross the common edge from L to R.
- In a cell R, the solution depends only on the values of f in the upwind cells: the linear system is block-triangular with small blocks.



Dependency graph

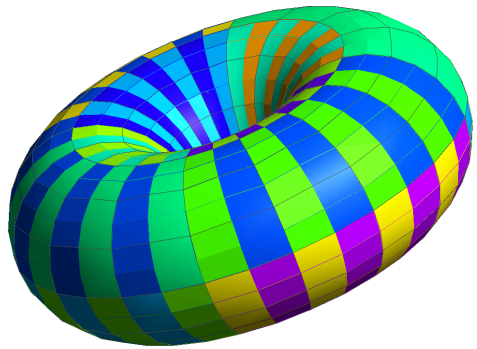
For a given velocity **v** we can build a dependency graph. The graph vertices are associated with cells, and graph edges with interfaces or boundaries. We consider two fictitious additional vertices: the "Upwind" vertex and the "Downwind" vertex.



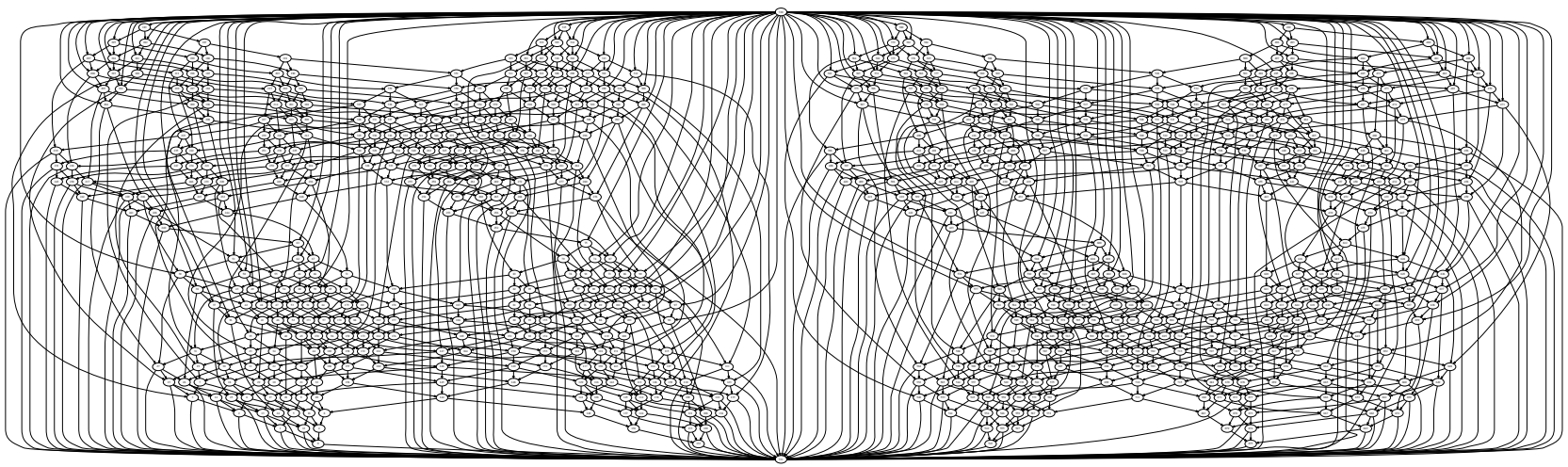


Example of transport graph

Mesh of a torus : 720 cells - 2064 interfaces



Mesh of a torus - transport graph for $(1,0,0)$ velocity.



Algorithm

Bibliography: [DR78, JNP84, WX99, NL08]
Transport algorithm:

- Topological ordering of the dependency graph.
- For each cell (in topological order):
  - Compute volumic terms contributions.
  - Compute fluxes from upwind interfaces/boundaries.
  - Solve local-to-the-cell linear systems.
  - Extract the results to the downwind cells interfaces.

Comments

- Parallelization is easily achieved thanks to a task-based runtime system (StarPU, OpenMP 4.0, etc.)
- Grouping the cells into macrocells allows to fine-tune the grain of the parallelism.

The kinetic approach allows to represent any hyperbolic of conservation laws by a set of coupled transport equations. This leads to natural unconditionally stable explicit solvers.

Vectorial kinetic representation

General kinetic model

- Vectorial kinetic equation

$$\partial_t \mathbf{f} + \sum_{k=1}^{D} \mathbf{V}^k \partial_k \mathbf{f} = \frac{1}{\tau}(\mathbf{f}^{eq}(\mathbf{f}) - \mathbf{f}). \tag{5}$$

  $\mathbf{f}(\mathbf{x}, t) \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^D$.

- The matrices $\mathbf{V}^k$, $1 \leq k \leq D$ are **diagonal** and **constant**.
- $\mathbf{w} = \mathbf{P}\mathbf{f}$ where $\mathbf{P}$ is a constant $m \times n$ matrix, $m < n$.
- The equilibrium distribution $\mathbf{f}^{eq}(\mathbf{f})$ is such that $\mathbf{P}\mathbf{f} = \mathbf{P}\mathbf{f}^{eq}(\mathbf{f})$.
- When $\tau \to 0$, approximation of

$$\partial_t \mathbf{w} + \sum_{k=1}^{D} \partial_k \mathbf{q}^k(\mathbf{w}) = 0,$$

  where the flux is given by

$$\mathbf{q}^k(\mathbf{w}) = \mathbf{P}\mathbf{V}^k \mathbf{f}^{eq}(\mathbf{f}).$$

  Stable under a "subcharacteristic" condition [Bou99, ADN00, Gra14].

Example: 1D system

$D = 1$, $m \geq 1$:

$$\partial_t \mathbf{w} + \partial_x \mathbf{q}(\mathbf{w}) = 0.$$

The kinetic model is given by $n = 2m$ and

$$\mathbf{P} = \begin{pmatrix} I & I \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} -\lambda I & 0 \\ 0 & \lambda I \end{pmatrix},$$

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_- \\ \mathbf{f}_+ \end{pmatrix}$$

$$\mathbf{f}_+^{eq}(\mathbf{f}) = \frac{\mathbf{f}_+ + \mathbf{f}_-}{2} + \frac{\mathbf{q}(\mathbf{f}_+ + \mathbf{f}_-)}{2\lambda}, \quad \mathbf{f}_-^{eq}(\mathbf{f}) = \frac{\mathbf{f}_+ + \mathbf{f}_-}{2} - \frac{\mathbf{q}(\mathbf{f}_+ + \mathbf{f}_-)}{2\lambda}.$$

Link with Jin-Xin relaxation [JX95]

Jin-Xin relaxation system for approximating a system of m conservation laws:

$$\partial_t \mathbf{w} + \partial_x \mathbf{z} = 0, \tag{6}$$

$$\partial_t \mathbf{z} + \lambda^2 \partial_x \mathbf{w} = \frac{1}{\tau}(\mathbf{q}(\mathbf{w}) - \mathbf{z}). \tag{7}$$

$\mathbf{z} \simeq \mathbf{q}(\mathbf{w})$ is the approximated flux.
Equivalent to the previous representation with the change of variables

$$\mathbf{w} = \mathbf{f}_+ + \mathbf{f}_-, \quad \mathbf{z} = \lambda \mathbf{f}_+ - \lambda \mathbf{f}_-.$$

Stable if $\lambda > \rho(\mathbf{q}'(\mathbf{w}))$.

Example: isothermal Euler

isothermal compressible flow of density ρ and velocity u. The sound speed is fixed to c. The conservative system is given by $D = 1$, $m = 2$ and $w = (\rho, \rho u), q(w) = (\rho u, \rho u^2 + c^2 \rho)$. The kinetic model is given by $n = 4$ and

$$\mathbf{V} = \mathrm{diag}(-\lambda, \lambda, -\lambda, \lambda), \quad \mathbf{P} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

$$\mathbf{w} = \mathbf{P}\mathbf{f}.$$

$$f_{2k-1}^{eq} = \frac{w_k}{2} - \frac{q(w)_k}{2\lambda}, \quad f_{2k}^{eq} = \frac{w_k}{2} + \frac{q(w)_k}{2\lambda}, \quad k = 1, 2.$$

Stable under the sub-characteristic condition $|u| + c < \lambda$.

Actually any system of conservation laws in dimension D admits an infinite numbers of such representations. In general

$$n \geq m(D+1).$$

For fluids, smaller n is sometimes possible because the mass flux is also a conserved quantity.
Applications: Magnetohydrodynamics (MHD), elastodynamics, compressible Euler, etc.

First order splitting algorithm

For each time step of duration Δt,

- free transport step: solve a set of transport equations with constant velocities
$$\partial_t \mathbf{f} + \mathbf{V} \cdot \nabla_x \mathbf{f} = 0 \Leftrightarrow \partial_t \mathbf{f} = \mathbf{L}\mathbf{f}.$$
- relaxation or "collision" step: compute $\mathbf{w} = \mathbf{P}\mathbf{f}$ and $\mathbf{f}^{eq}$ (that depends only on $\mathbf{w}$), and solve
$$\partial_t \mathbf{f} = \frac{1}{\tau}(\mathbf{f}^{eq} - \mathbf{f}) \Leftrightarrow \partial_t \mathbf{f} = \mathbf{N}(\mathbf{f}).$$

The resulting scheme is $O(\Delta t)$ with high numerical viscosity

Structure preserving time integration

Time-symmetric scheme

Let us consider a differential equation

$$v'(t) = g(v(t)), \quad v(0) = v_0.$$

First order numerical method Φ

$$\Phi(\Delta t)v_0 = v(\Delta t) + O(\Delta t^2).$$

Another method (Crank-Nicolson):

$$\Psi(\Delta t) = \Phi(\frac{\Delta t}{2}) \circ \Phi(\frac{-\Delta t}{2})^{-1}.$$

The CN method is symmetric:

$$\Psi(-\Delta t) = \Psi(\Delta t)^{-1}, \quad \Psi(0) = I.$$

A symmetric method is necessarily second order.

Usual Strang splitting

We have to solve $\partial_t \mathbf{f}_h = \mathbf{L}_h \mathbf{f}_h + \mathbf{N}_h \mathbf{f}_h$. Each step is solved by the Crank-Nicolson scheme:

- Transport solver: $\exp(\Delta t \mathbf{L}_h) \simeq T_2(\Delta t) := (\mathbf{I} + \frac{\Delta t}{2}\mathbf{L}_h)(\mathbf{I} - \frac{\Delta t}{2}\mathbf{L}_h)^{-1}$.
- Collision integrator:
  $\exp(\Delta t \mathbf{N}_h)\mathbf{f}_h \simeq C_2(\Delta t)\mathbf{f}_h := \frac{(-1+\frac{2\tau}{\Delta t})\mathbf{f}_h}{1+\frac{2\tau}{\Delta t}} + \frac{2\mathbf{f}_h^{eq}(\mathbf{f}_h)}{1+\frac{2\tau}{\Delta t}}$.

T_2 and C_2 are time-symmetric when $\tau > 0$.
Usual Strang splitting

$$\exp(\Delta t(\mathbf{L}_h + \mathbf{N}_h)) \simeq T_2(\frac{\Delta t}{2}) C_2(\Delta t) T_2(\frac{\Delta t}{2}).$$

It is time-symmetric for $\tau > 0$ and thus second order.

Asymptotic preserving Strang splitting

- When $\tau \to 0$, $C_2(\Delta t)\mathbf{f}_h = 2\mathbf{f}_h^{eq}(\mathbf{f}_h) - \mathbf{f}_h$ and C_2 is no more symmetric for $\tau = 0$ ($C_2(0) \neq I$).
- The usual Strang splitting is **no more** time-symmetric for $\tau = 0$.
- We observe order reduction [Jin95, PR05].
- Time-symmetric splitting:

$$M_2(\Delta t) = T_2(\frac{\Delta t}{4})C_2(\frac{\Delta t}{2})T_2(\frac{\Delta t}{2})C_2(\frac{\Delta t}{2})T_2(\frac{\Delta t}{4}).$$

Remark: equivalent equation

We apply the previous scheme to the Jin-Xin relaxation in the limit $\tau \to 0$:

$$\begin{aligned} \partial_t \mathbf{w} + \partial_x \mathbf{z} &= 0, \\ \partial_t \mathbf{z} + \lambda^2 \partial_x \mathbf{w} &= \frac{1}{\tau}(\mathbf{q}(\mathbf{w}) - \mathbf{z}). \end{aligned}$$

By Taylor expansions we can compute the equivalent equation of the numerical scheme. Introducing $\mathbf{y} = \mathbf{z} - \mathbf{f}(\mathbf{w})$, we find

$$\partial_t \begin{pmatrix} \mathbf{w} \\ \mathbf{y} \end{pmatrix} + \begin{pmatrix} \mathbf{f}'(\mathbf{w}) & 0 \\ 0 & -\mathbf{f}'(\mathbf{w}) \end{pmatrix} \partial_x \begin{pmatrix} \mathbf{w} \\ \mathbf{y} \end{pmatrix} = O(\Delta t^2).$$

- Cannot be obtained by usual Chapman-Enskog expansion (?).
- Useful analysis for devising stable boundary conditions.

Palindromic composition (I)

General palindromic scheme [MQ02, HLW06, CFH+17] with $s+1$ steps has the form

$$M_s(\Delta t) = M_2(\gamma_0 \Delta t) M_2(\gamma_1 \Delta t) \cdots M_2(\gamma_s \Delta t), \tag{8}$$

where the γ_i's are real numbers such that

$$\gamma_i = \gamma_{s-i}, \quad 0 \leq i \leq s.$$

The method is low-storage: only one time-step of the solution is stored and directly updated in memory.
Example: the fourth-order Suzuki scheme (see [Suz90, HLW06, MQ02])

$$\gamma_0 = \gamma_1 = \gamma_3 = \gamma_4 = \frac{1}{4 - 4^{1/3}}, \quad \gamma_2 = -\frac{4^{1/3}}{4 - 4^{1/3}}. \tag{9}$$

Palindromic composition (II)

Example 2: sixth-order Kahan-Li scheme [KL97]:

$$\begin{aligned}
\gamma_0 = \gamma_8 &= 0.39216144400731413927565533003 8\ldots \\
\gamma_1 = \gamma_7 &= 0.33259913678935943860427212532 5\ldots \\
\gamma_2 = \gamma_6 &= -0.70624617255763935980984533722 27\ldots \\
\gamma_3 = \gamma_5 &= 0.08221359629355080023044270533 41\ldots \\
\gamma_4 &= 0.79854399093482996339895035304 8\ldots
\end{aligned} \tag{10}$$

The methods (9) and (10) require to apply the elementary collision or transport bricks C_2 and T_2 with negative time steps $-\Delta t < 0$.

- Collision operator: $C_2(\Delta t)\mathbf{f}_h = 2\mathbf{f}_h^{eq}(\mathbf{f}_h) - \mathbf{f}_h$, reversible when $\tau = 0$. Works also if $\tau \ll \Delta t$.
- Transport: we simply solve a transport with the opposite velocity.

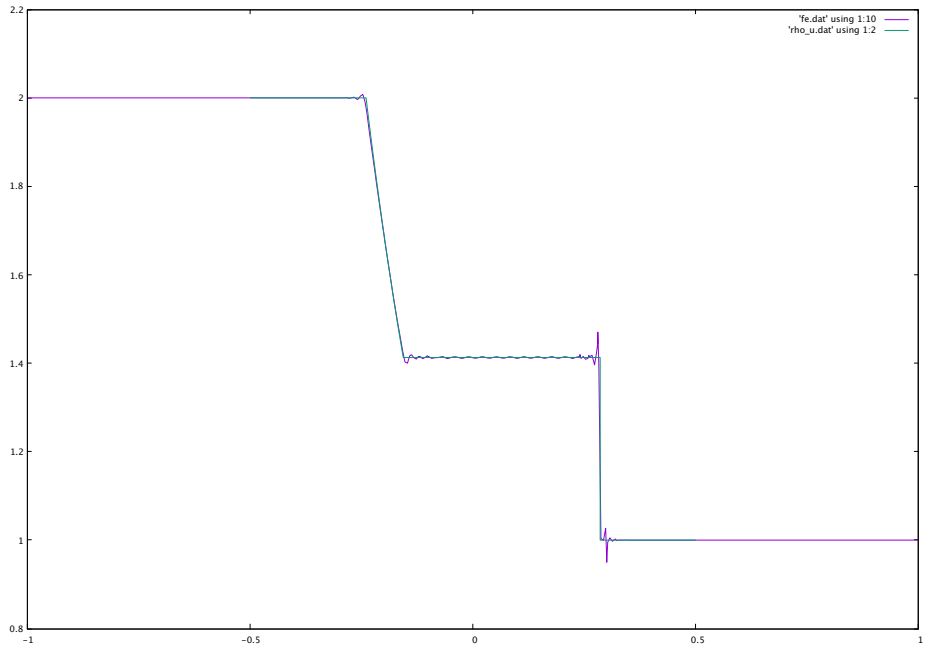
Numerical results in 1D

- Isothermal Euler.
- Riemann problem $\rho_L = 2$, $\rho_R = 1$, $u_L = u_R = 0$.
- Implicit Discontinuous Galerkin scheme of order $d = 5$.
- 100 cells (600 nodes).
- CFL number

$$\beta = \frac{\lambda \Delta t}{\Delta x},$$

where Δx is the minimal distance between two Gauss-Lobatto points in a cell ($\Delta x \leq h/d$).

density, CFL=3



Smooth solution

Test case with a smooth solution, in the fluid limit $\tau = 0$.

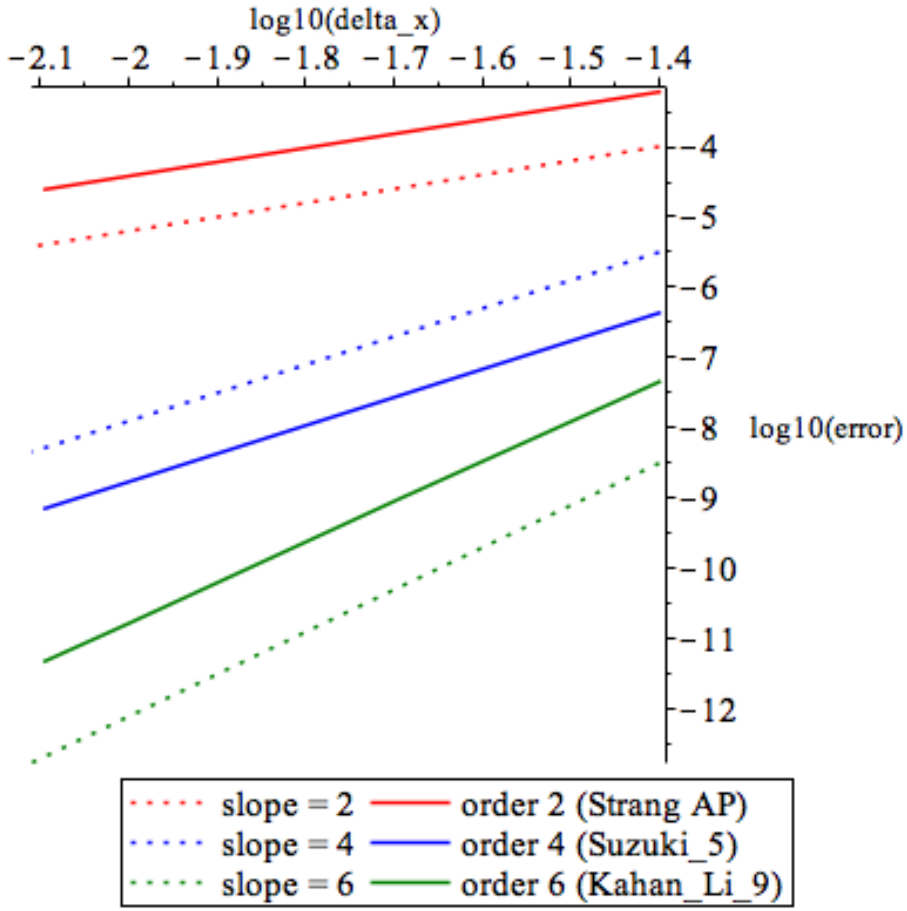
- Initial condition given by

$$\rho(x,0) = 1 + e^{-30x^2}, \quad u(x,0) = 0.$$

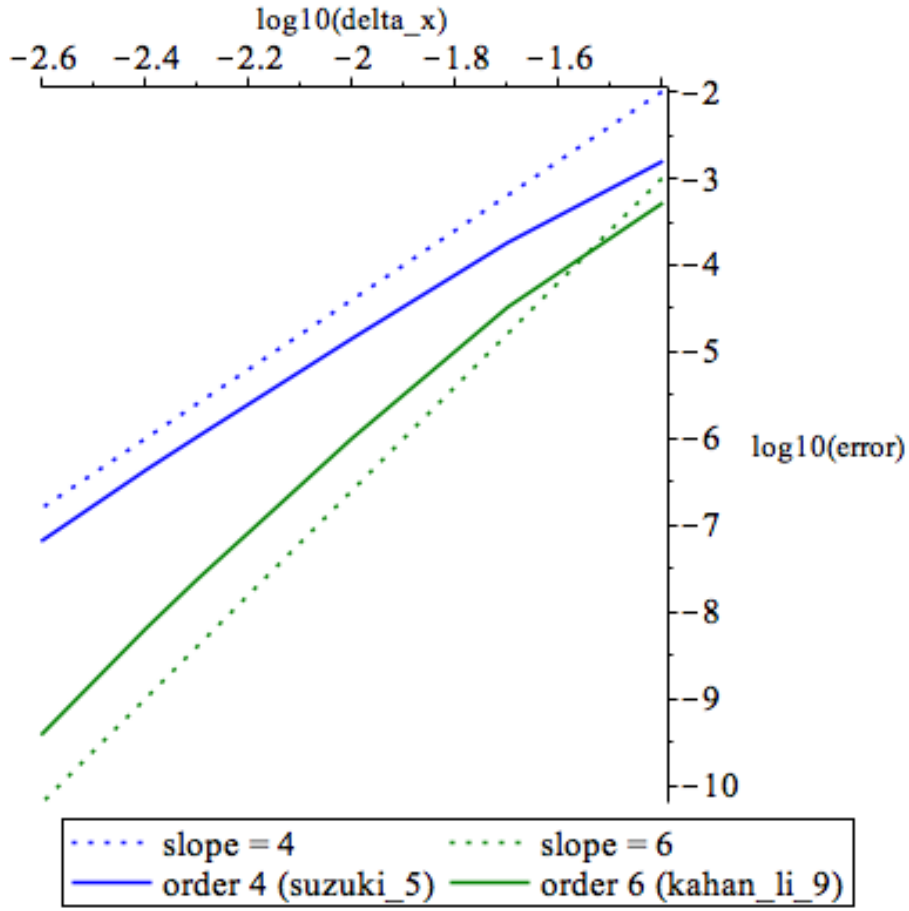
  sound speed $c = 0.6$, lattice velocity $\lambda = 2$.

- $t_{\max} = 0.4$ so that the boundary conditions play no role.
- error = L^2 norm of $\mathbf{f}_h(\cdot, t_{\max}) - \mathbf{f}(\cdot, t_{\max})$.

Smooth solution, CFL=5



Smooth solution, CFL=50



Implementation

Higher dimensions: StarPU parallelization

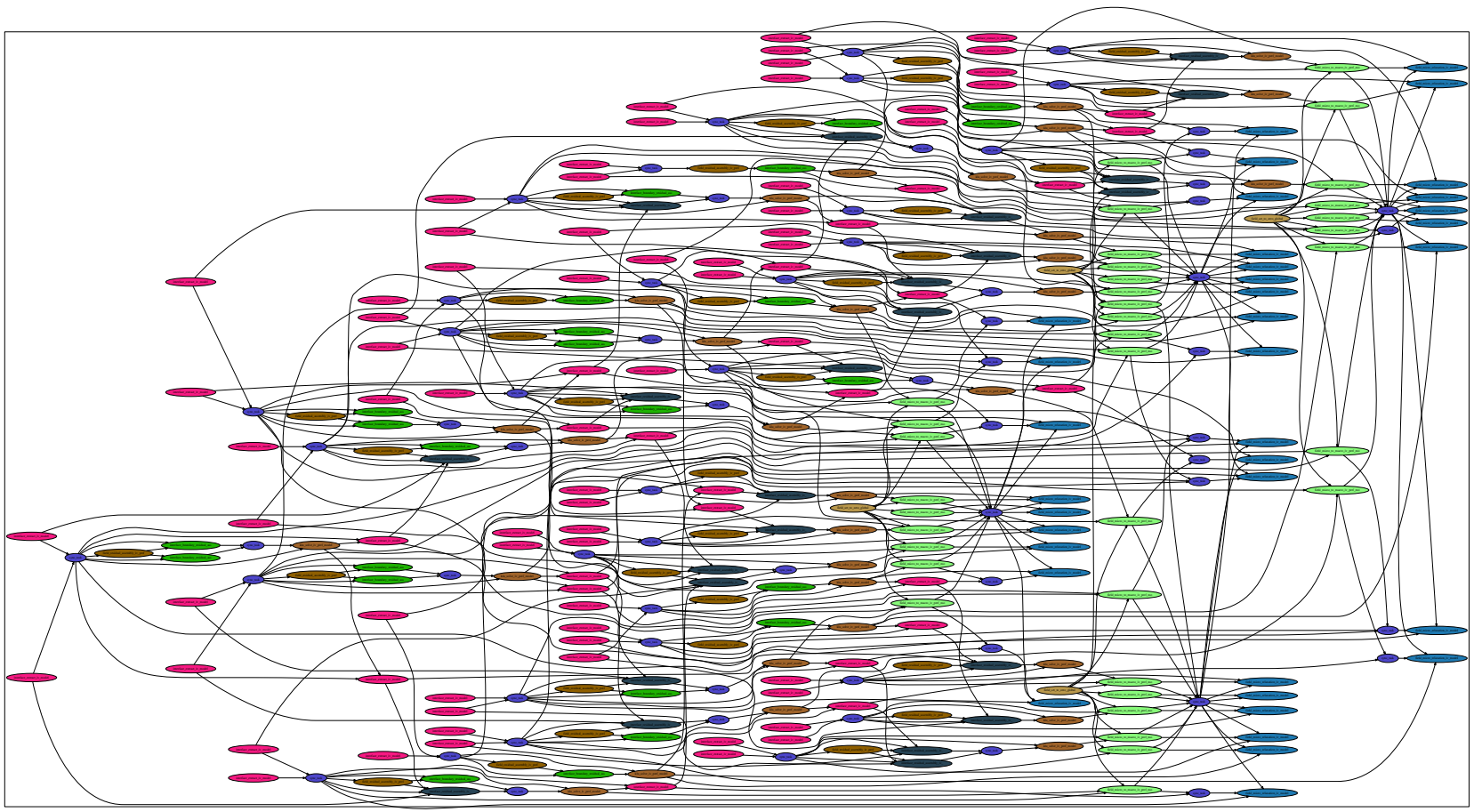
- StarPU is a library developed at Inria Bordeaux [AAF+12]: `http://starpu.gforge.inria.fr`
- Task-based parallelism.
- Task description: codelets, inputs (R), outputs (W or RW).
- Several implementations of the same task are possible.
- The user submits tasks in a correct sequential order.
- StarPU schedules the tasks in parallel if possible.

Implementation in SCHNAPS

- SCHNAPS: "Solveur Conservatif Hyperbolique Non-linéaire Appliqué aux PlaSmas"
- `http://schnaps.gforge.inria.fr` : DG for general hyperbolic systems.
- Implementation of the implicit transport scheme.
- Implementation of a generic collision stage.
- Task parallel distributions managed by StarPU.
- Palindromic high order extension.

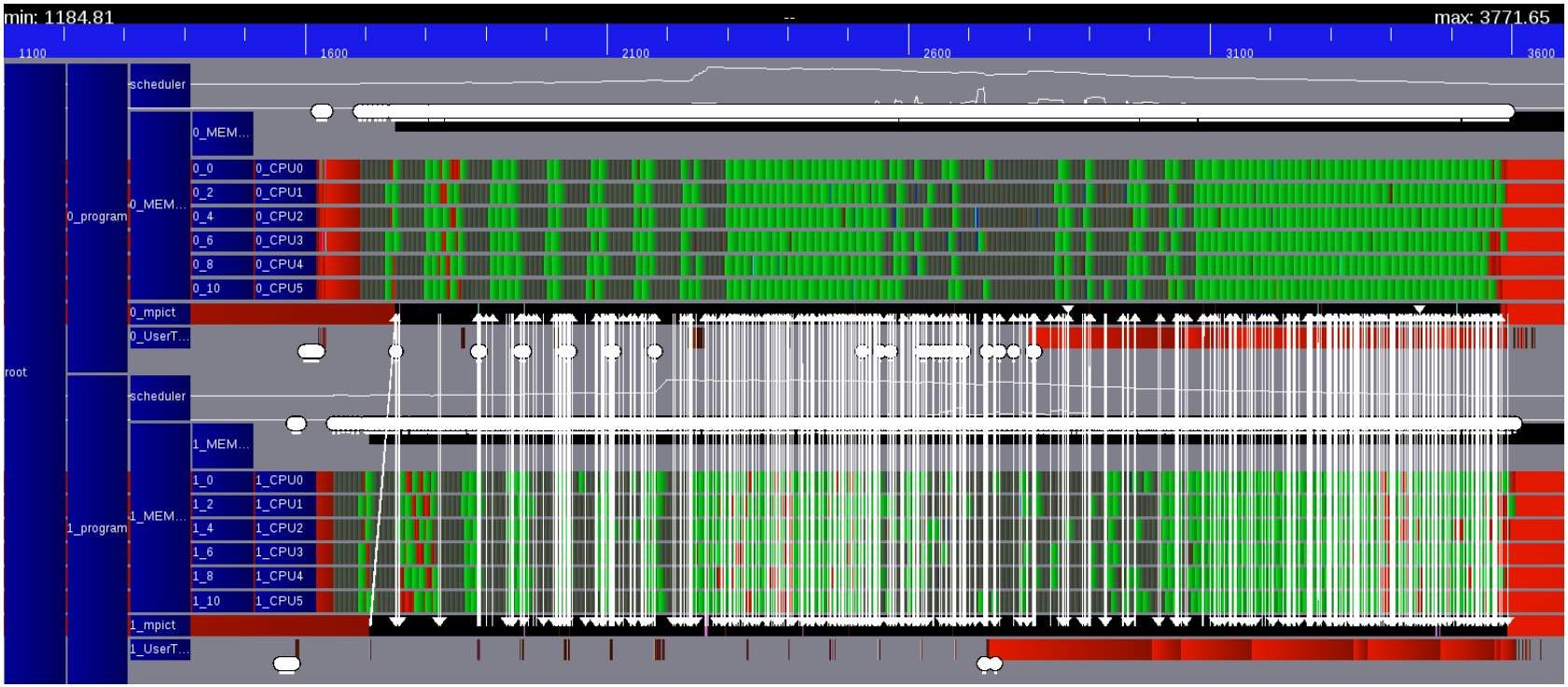
Task graph generated by StarPU

- 4 subdomains in a 2D square
- a single time step of the first order scheme (T + C)



Gantt diagram

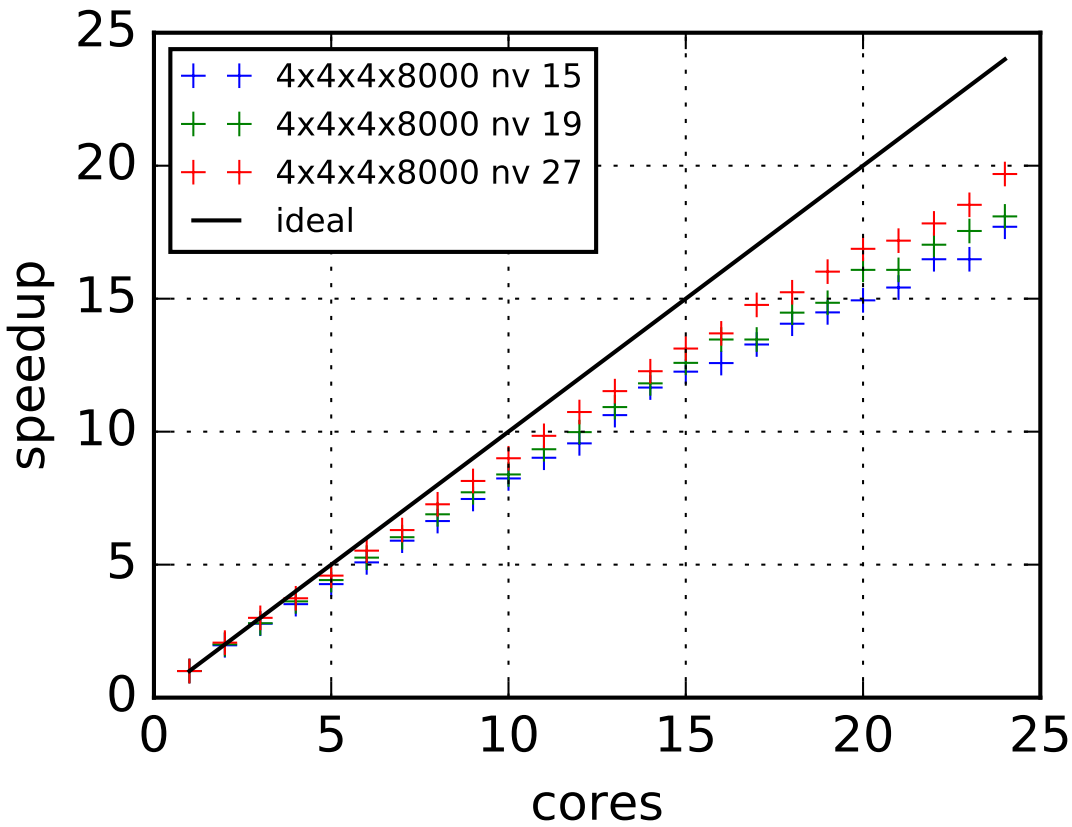
D2Q9 model - 2x6 threads



Low overhead/sleep time

StarPU multithread scaling

D3Q15,D3Q19,D3Q27 models on a cube with 4x4x4 elements and 8000 dof per elements with eager scheduler.

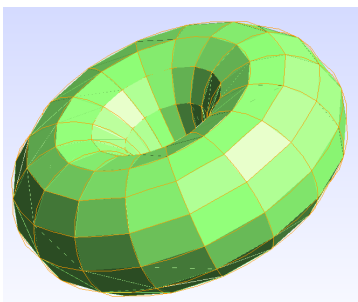
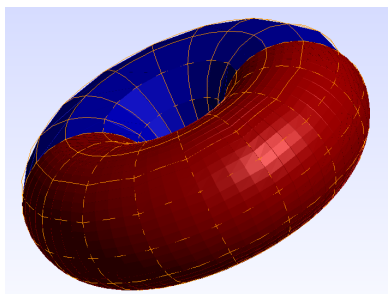
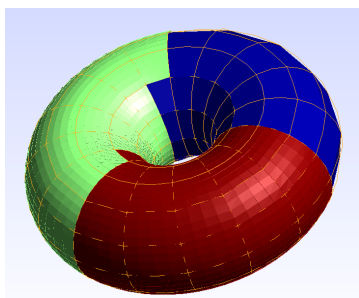
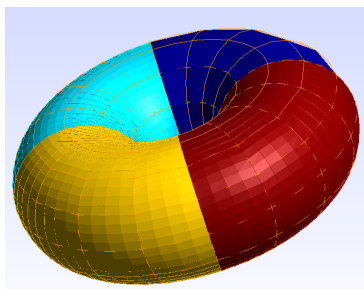


StarPU MPI Scaling

Toroidal mesh : 720 macroelements x 3335 dof
2064 interfaces - 192 boundary faces

Wall time in sec for 100 interations.

Nthreads/Nmpi	1	2	3	4
14	6862	2772	1491	1014

Hybrid CPU GPU (explicit solver)

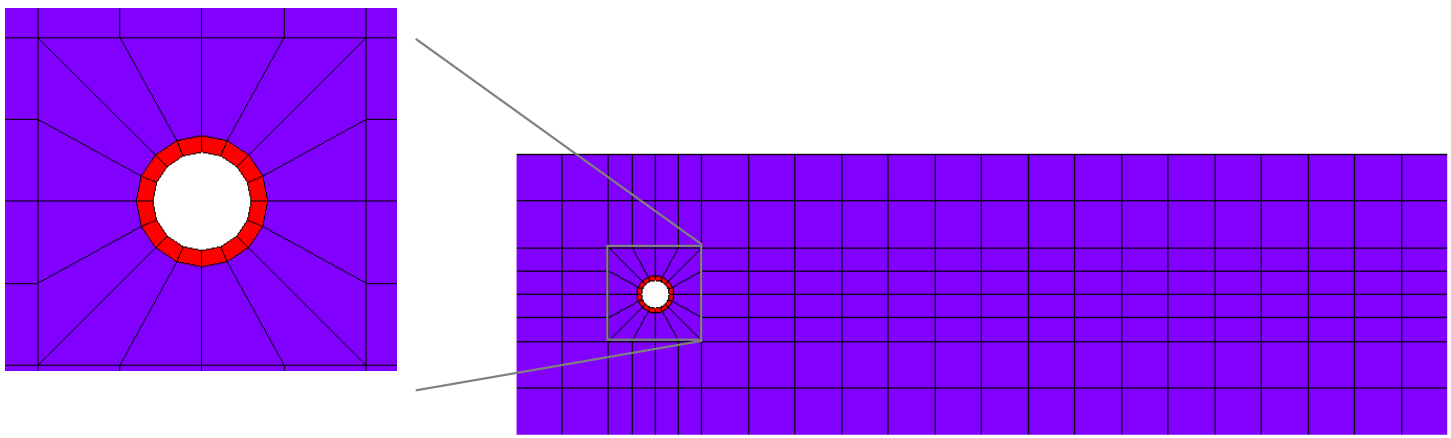
Traditional explicit DG solver (no kinetic representation). C and OpenCL codelets: StarPU can harness the available CPUs and GPUs.

	1 GPU	2 GPU	4 GPU	4 GPU + 1 CPU
128^2	14.2	8.8	6.1	4.8
256^2	19.7	12.6	9.8	10.4
512^2	57	28.8	23.5	24.3
1024^2	208	96	60	49
2048^2	740	380	250	207

The GPUs are faster, but the CPUs give n additional perfs.
StarPU codelets for the kinetic solver: work in progress...

$2D$ flow around a cylinder

- D2Q9 model for Isothermal Euler
- mesh is adapted to the geometry of the obstacle
- 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 $v_{\bar{i}} = -v_i$.
  - with CN scheme and $(\tau = 0) \rightarrow$ "bounce-back" operator : simply swap f_i values between opposite velocities.
- imposed state at boundaries with constant low Mach flow.



- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm

- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm

- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm

- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

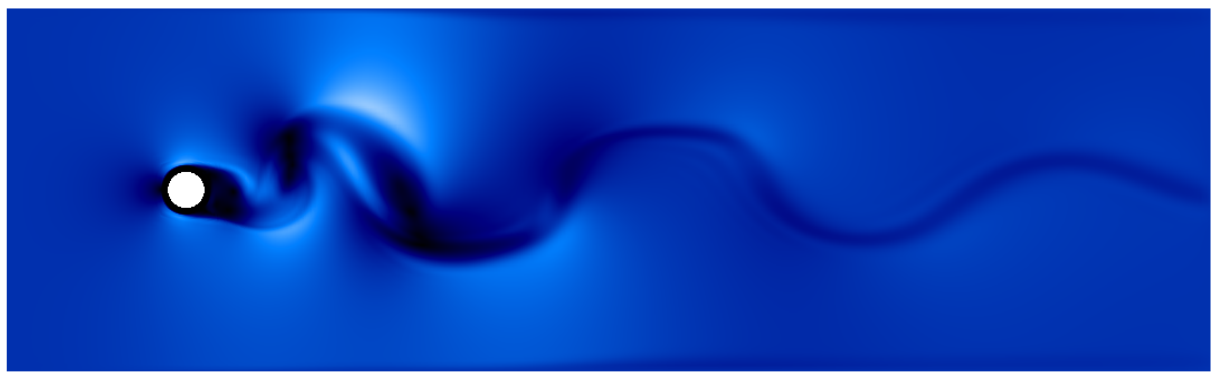
Velocity field norm

- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm

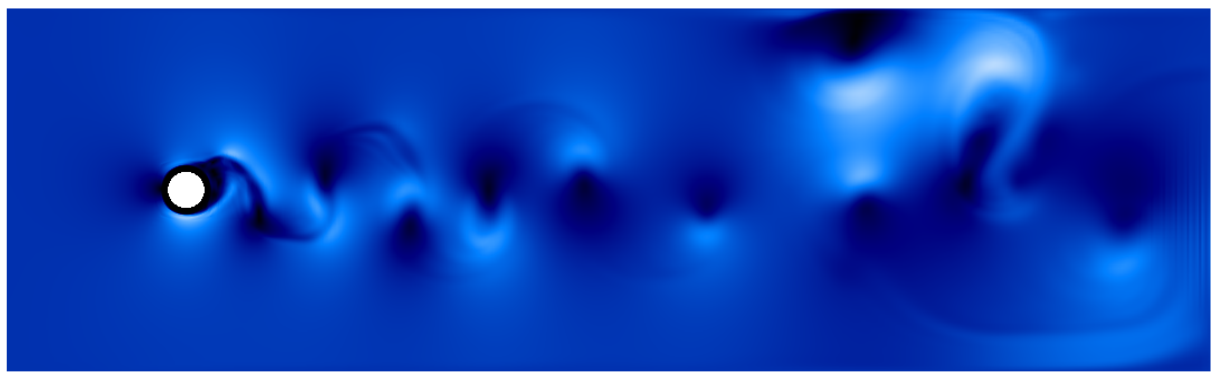
- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm



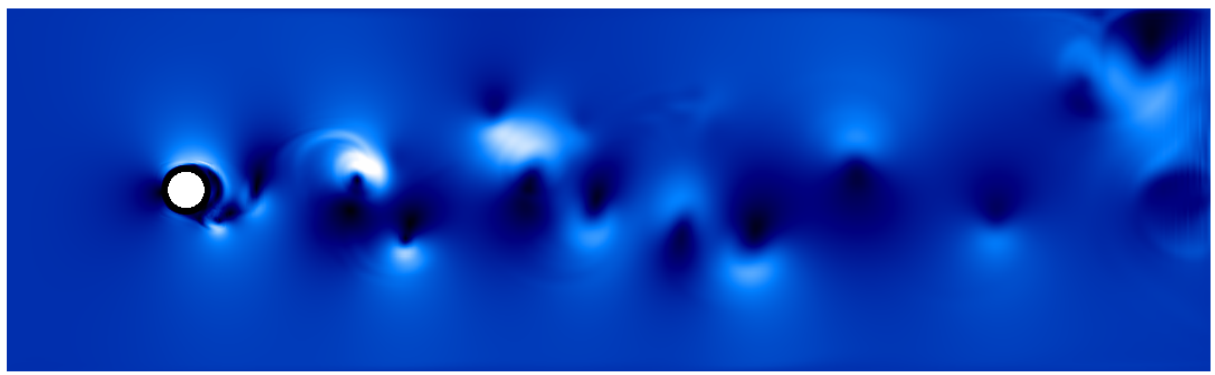
- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm



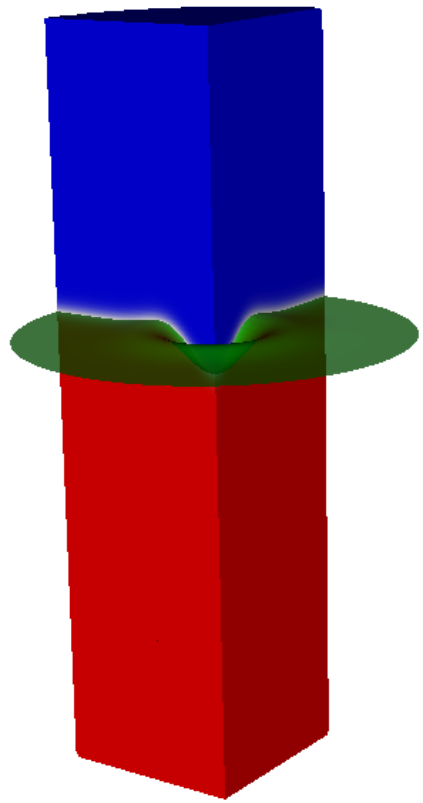
- imposed velocity field at inlet $u \approx 0.07c$.
- finite $\tau = 0.0001$.

Velocity field norm



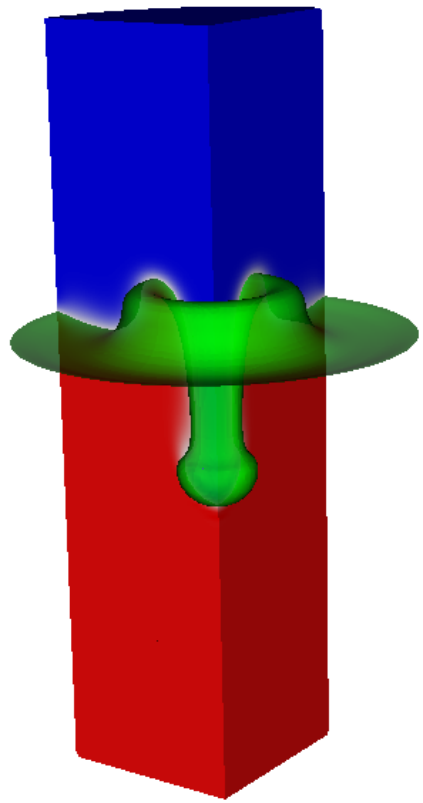
3D Multiphase flow

Rayleigh-Taylor instability. Two immiscible fluids with gravity.



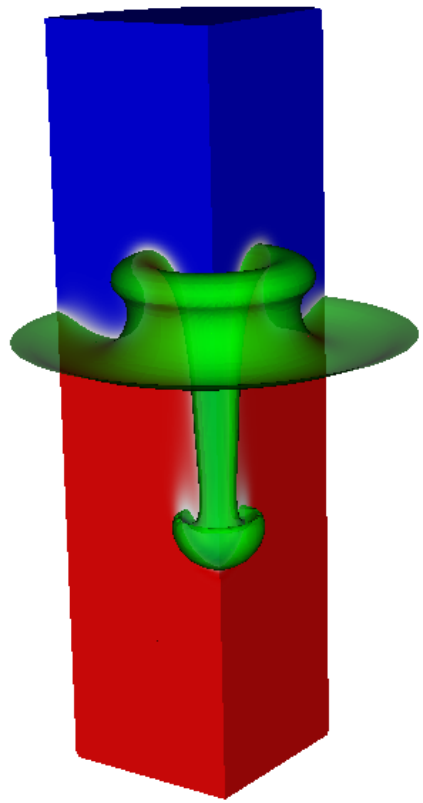
3D Multiphase flow

Rayleigh-Taylor instability. Two immiscible fluids with gravity.



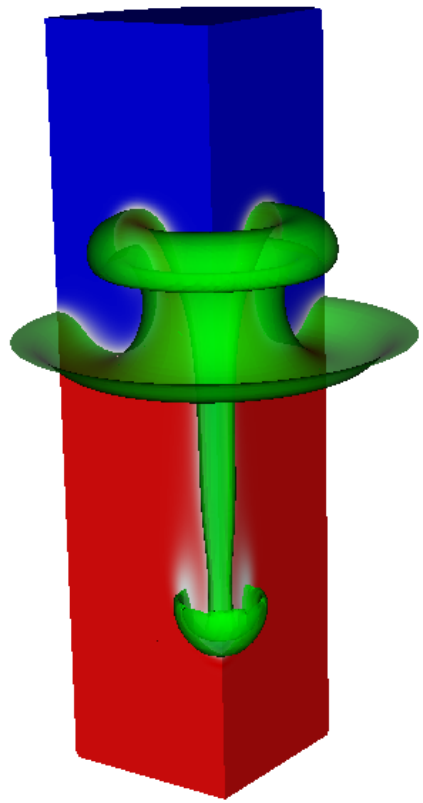
3D Multiphase flow

Rayleigh-Taylor instability. Two immiscible fluids with gravity.



3D Multiphase flow

Rayleigh-Taylor instability. Two immiscible fluids with gravity.



Drifting MHD stationary vortex : $7 \times D2Q4$

$$\text{Basic MHD}\\ \text{(+divergence cleaning)} \quad \begin{cases} \partial_t \rho + \nabla \cdot \rho \mathbf{u} & = 0 \\ \partial_t \rho \mathbf{u} + \nabla \cdot \left[\rho \mathbf{u}\mathbf{u} + (p + \frac{B^2}{2}) I - \mathbf{B}\mathbf{B} \right] & = 0 \\ \partial_t Q + \nabla \cdot \left[(Q + p + \frac{B^2}{2}) \mathbf{u} - (\mathbf{B} \cdot \mathbf{u}\mathbf{B}) \right] & = 0 \\ \partial_t \mathbf{B} + \nabla \cdot (\mathbf{B}\mathbf{u} - \mathbf{u}\mathbf{B} + \psi I) & = 0 \\ \partial_t \psi + \nabla \cdot (c_h^2 \mathbf{B}) & = 0 \end{cases}$$

Closure: $p = (\gamma - 1)\rho e = (\gamma - 1)\left[Q - \rho \frac{u^2}{2} + \frac{B^2}{2} \right]$ Simple 2D stationary solution (azimuthal symmetry) + constant drift velocity

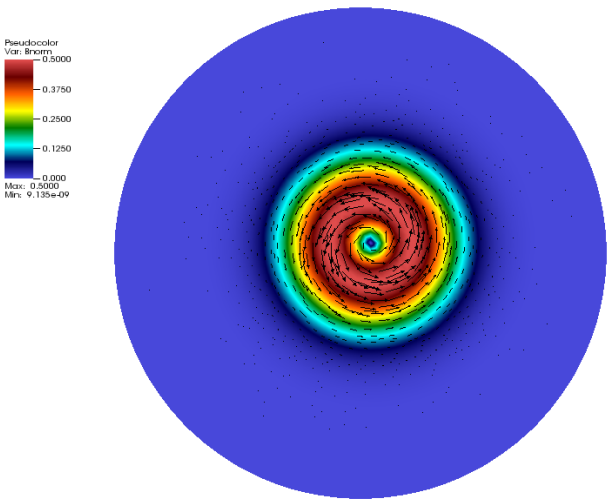
$$\begin{cases} \rho & = \rho_0 \\ \mathbf{u} & = u_0 \left[\mathbf{u}_{drift} + h(r) \mathbf{e}_\vartheta \right] \\ \mathbf{B} & = b_0 h(r) \mathbf{e}_\vartheta \\ p(r) & = p_0 + \frac{b_0^2}{2}(1 - h^2(r)) \\ b_0^2 & = \rho u_0^2 \end{cases}$$

expressed in cylindrical coordinates in the drifting frame ($\mathbf{r_0} = \mathbf{u}_{drift} t$).

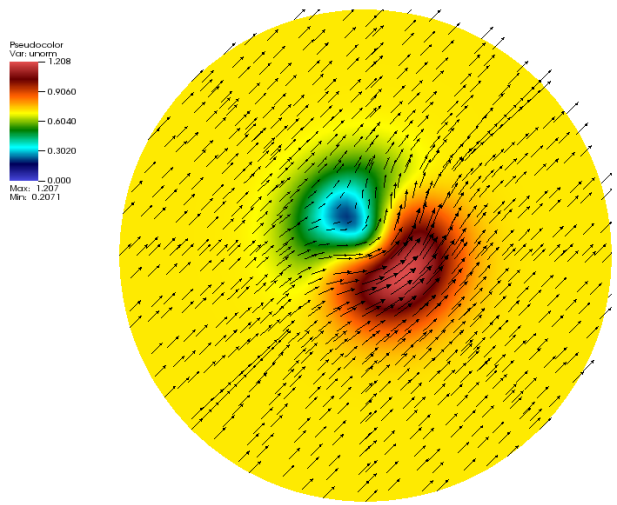
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



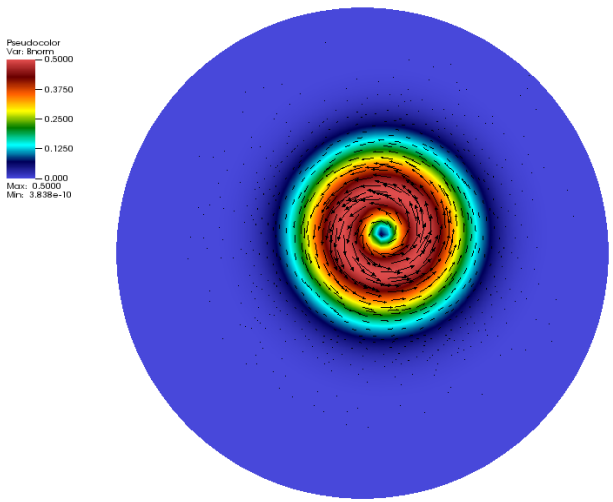
Velocity



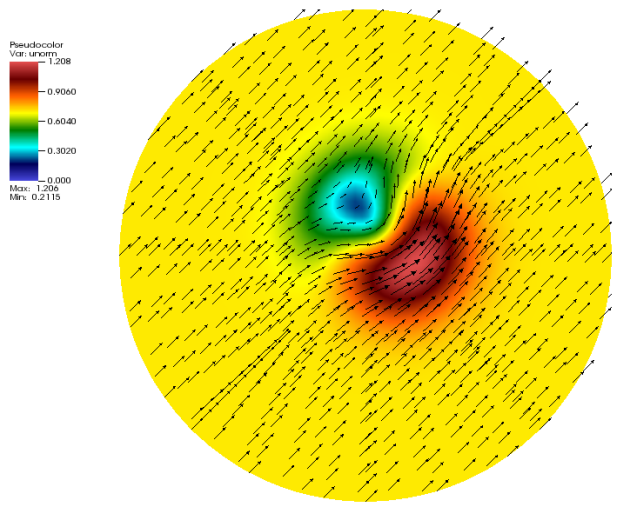
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



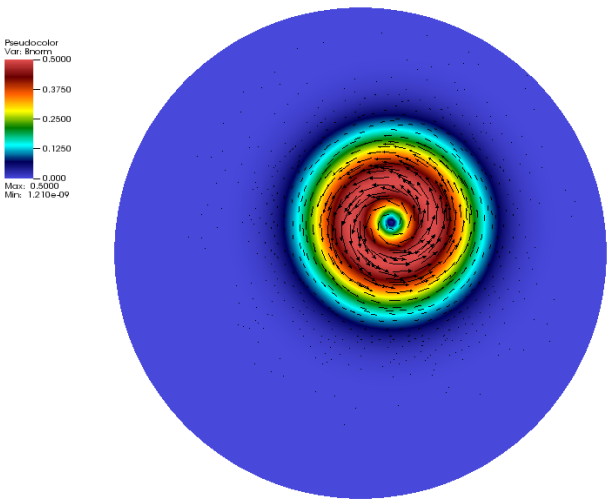
Velocity



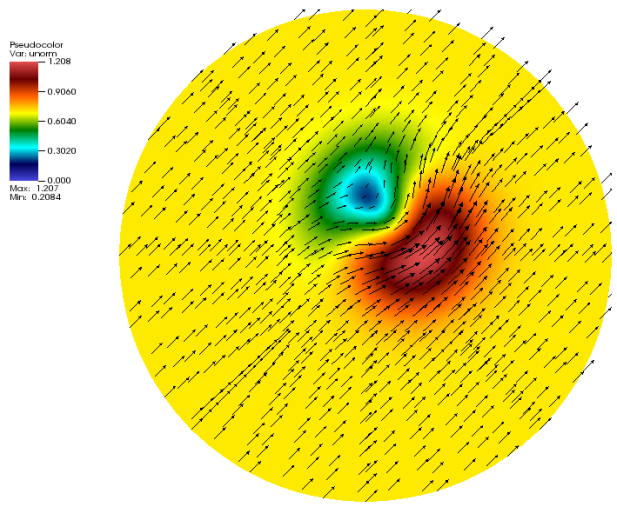
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



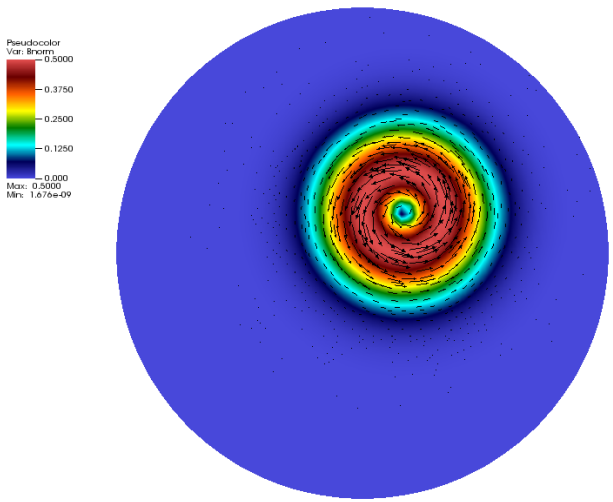
Velocity



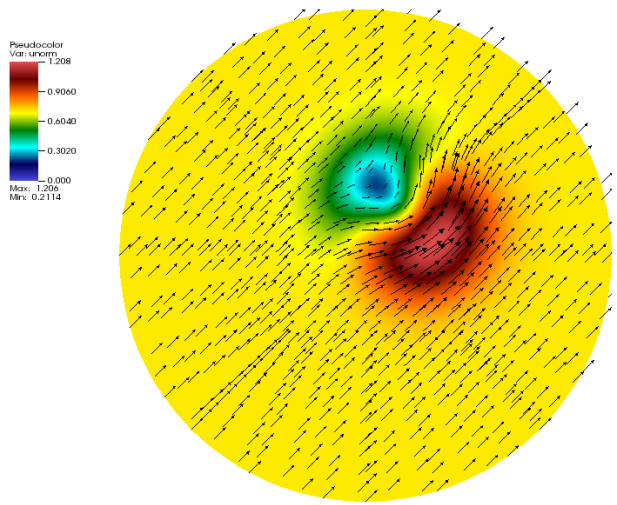
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1-r^2)/2]$

Magnetic field



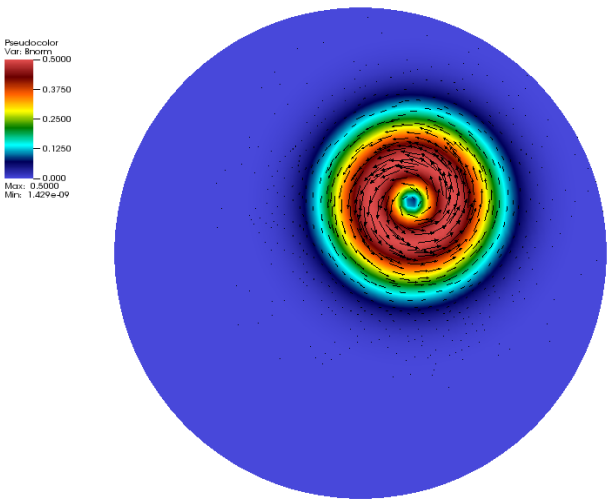
Velocity



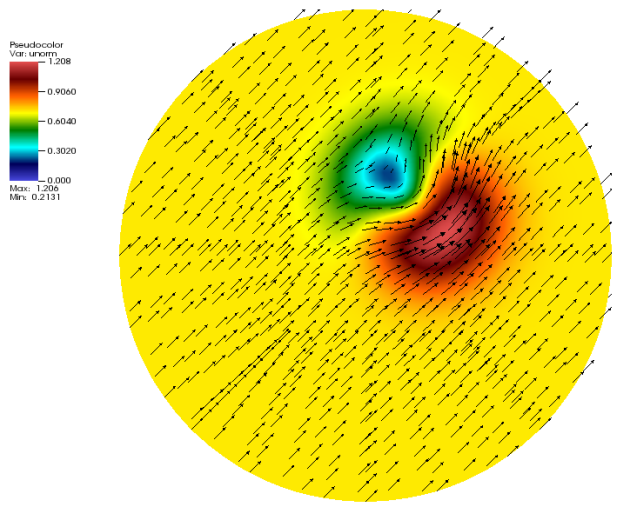
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



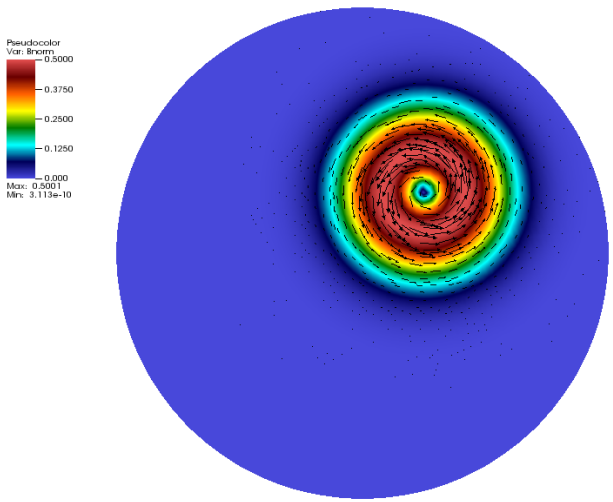
Velocity



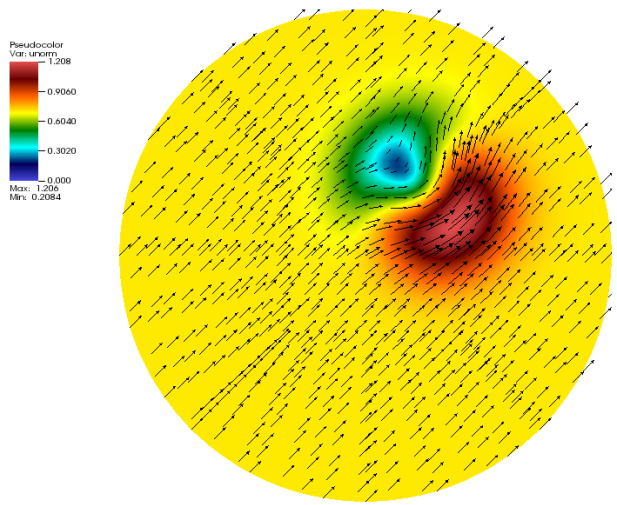
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



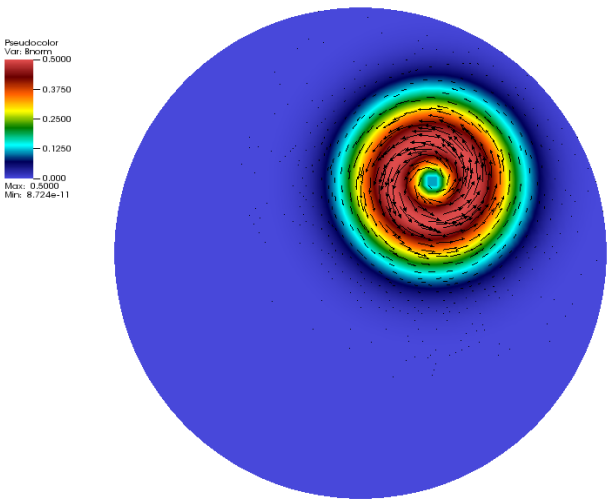
Velocity



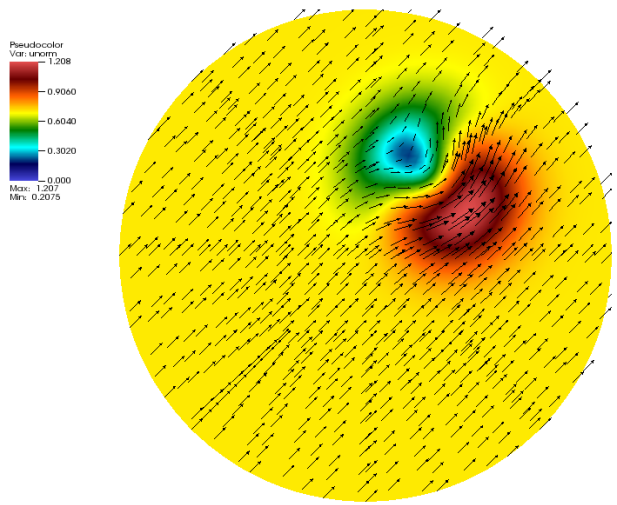
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



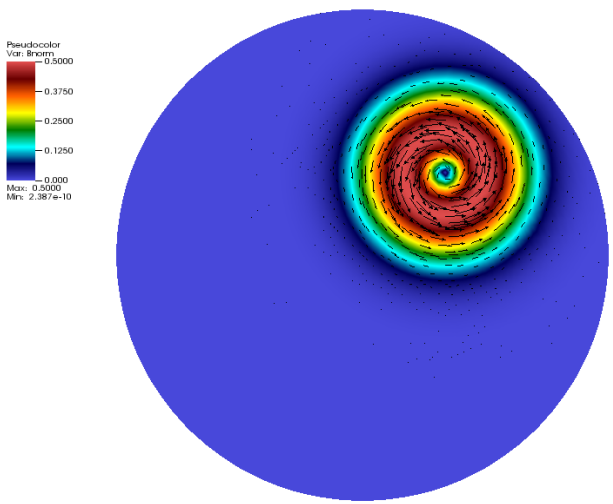
Velocity



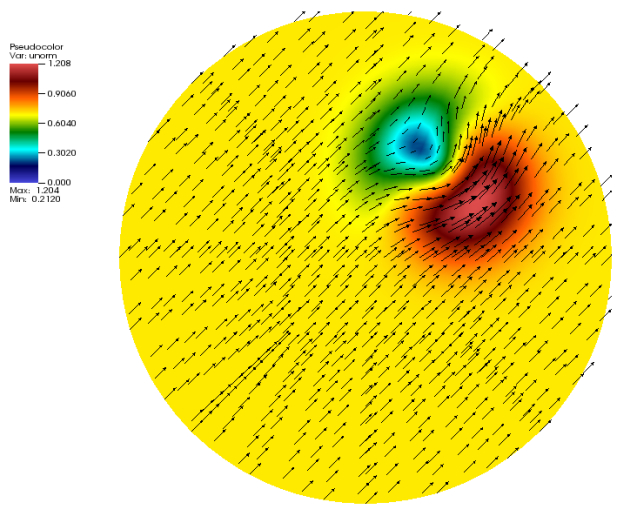
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



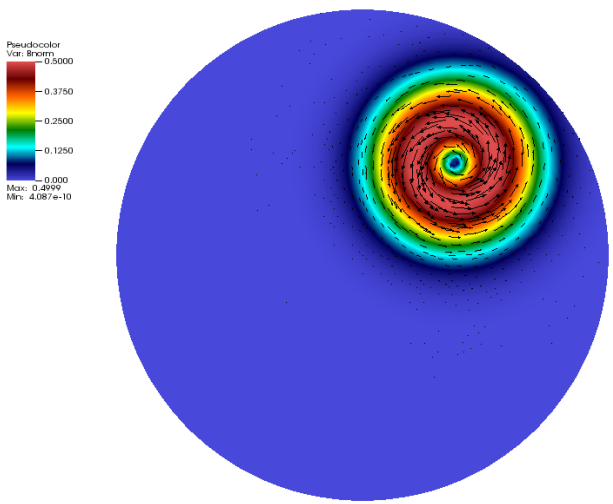
Velocity



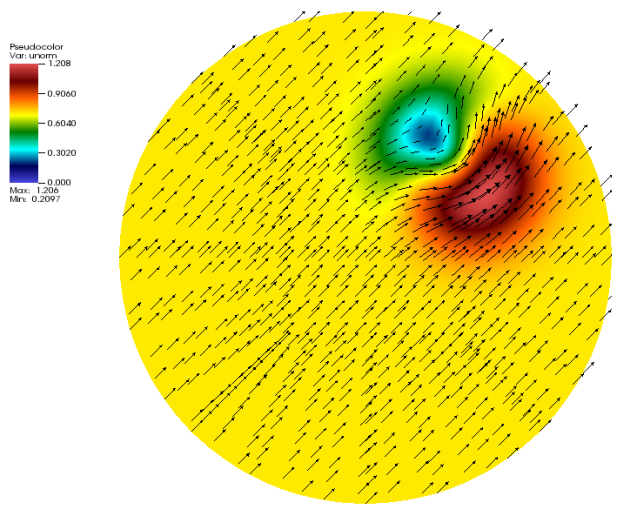
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



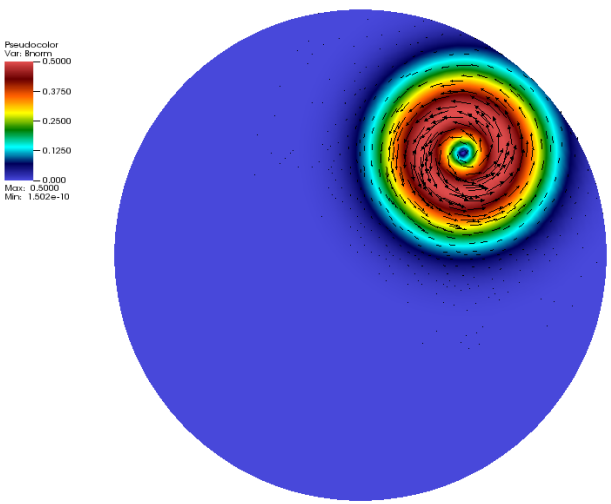
Velocity



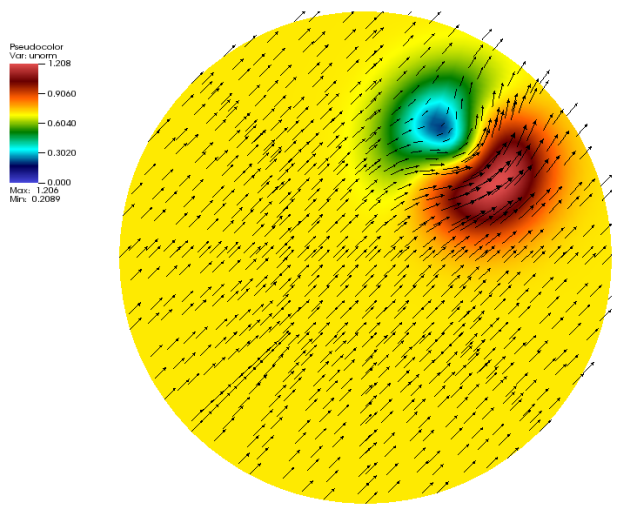
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



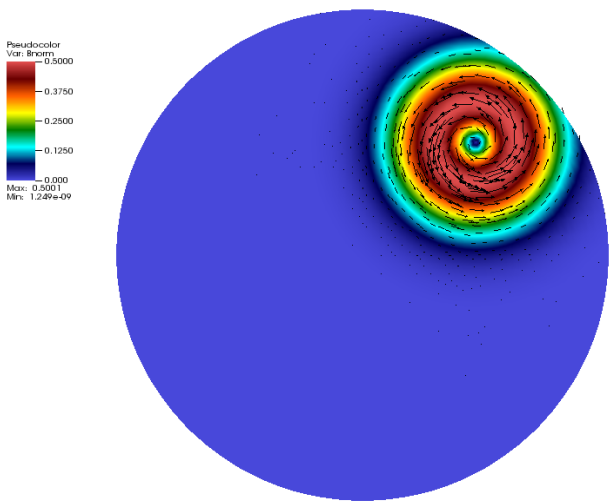
Velocity



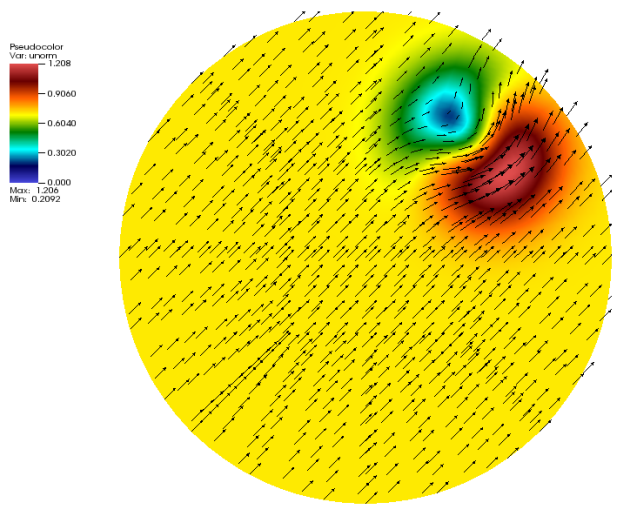
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



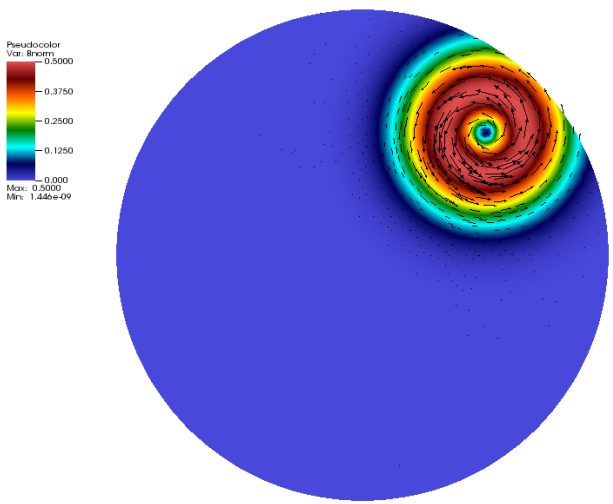
Velocity



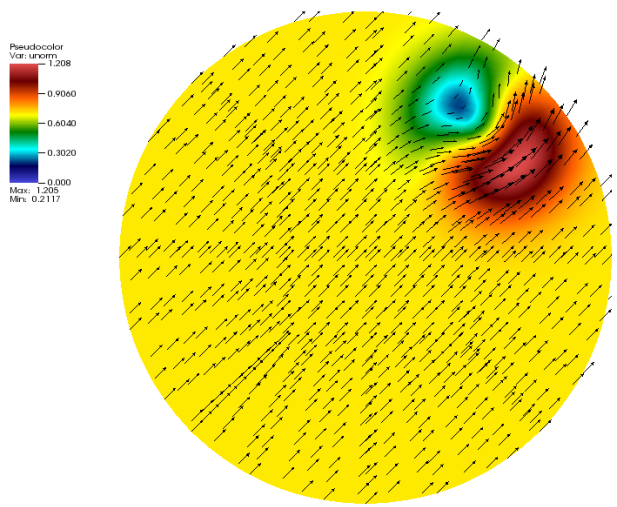
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1-r^2)/2]$

Magnetic field



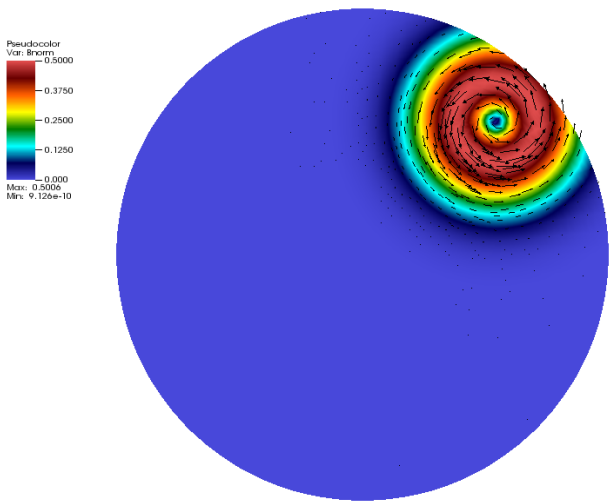
Velocity



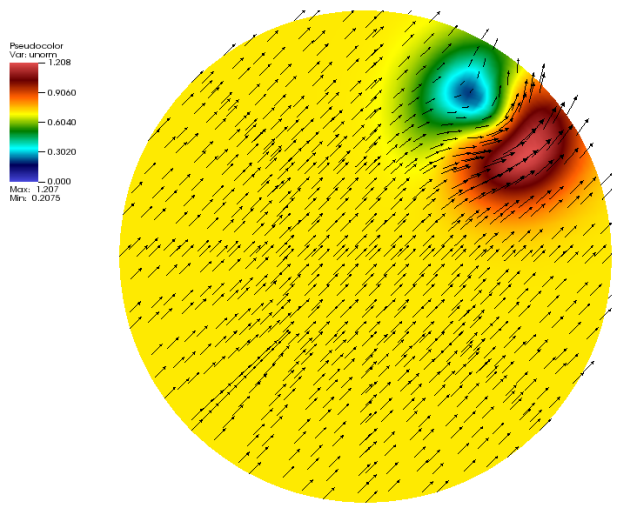
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



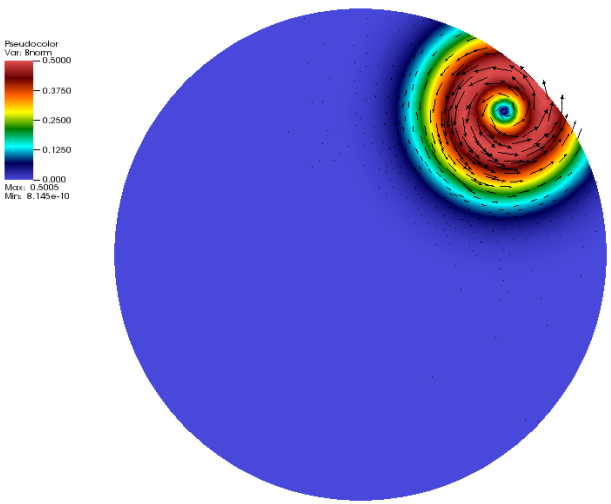
Velocity



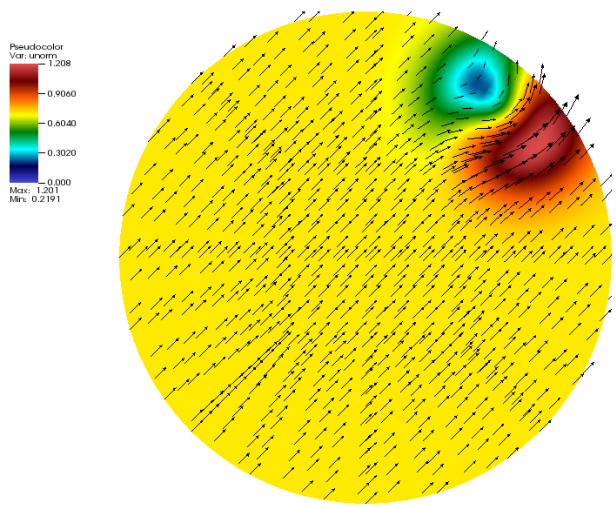
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



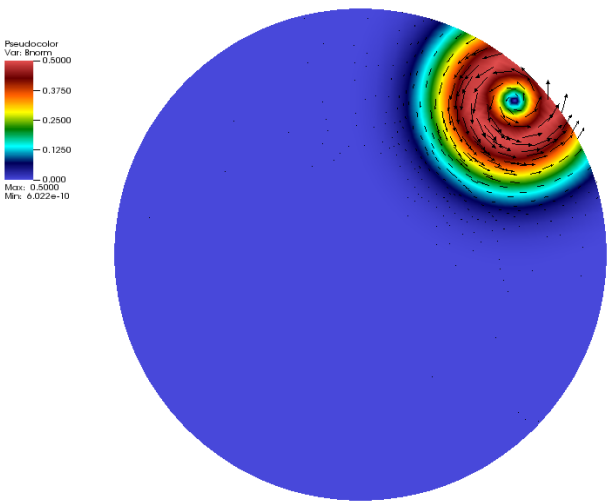
Velocity



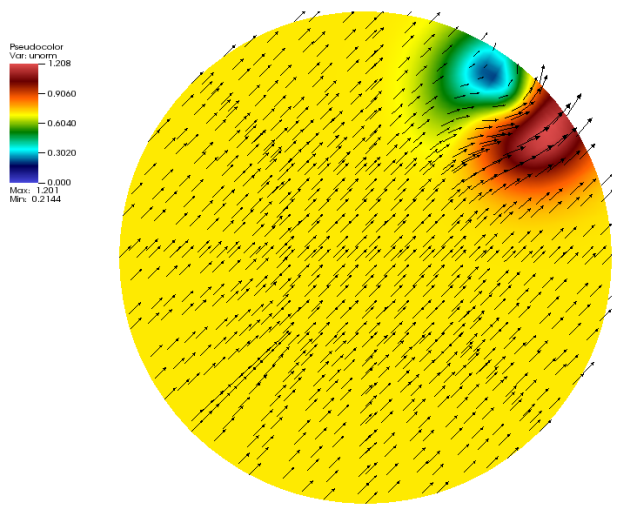
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



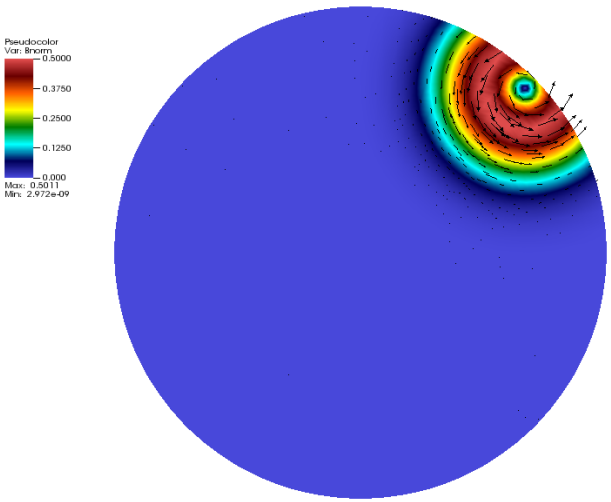
Velocity



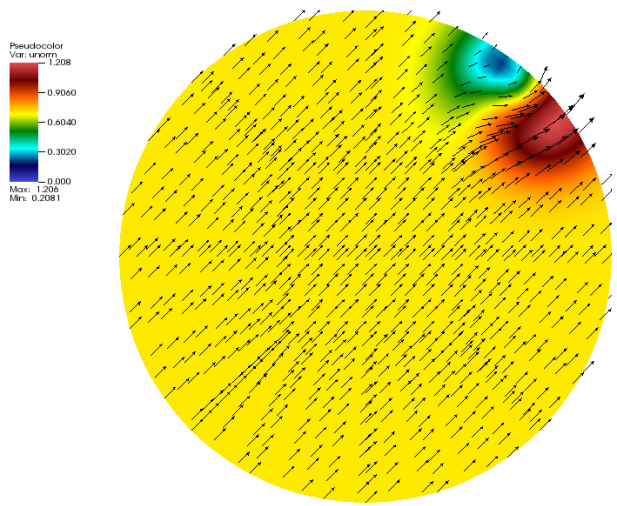
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



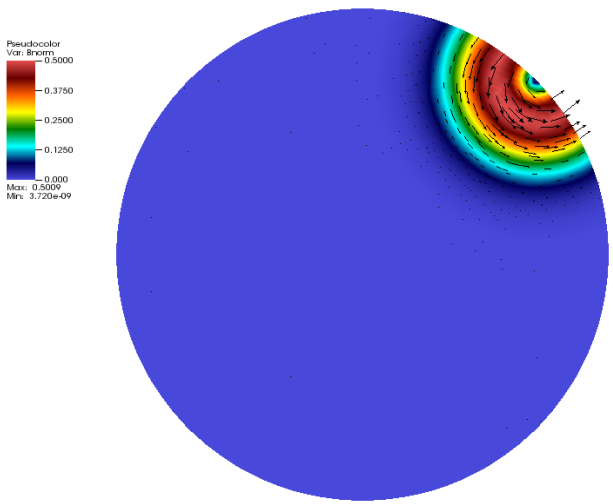
Velocity



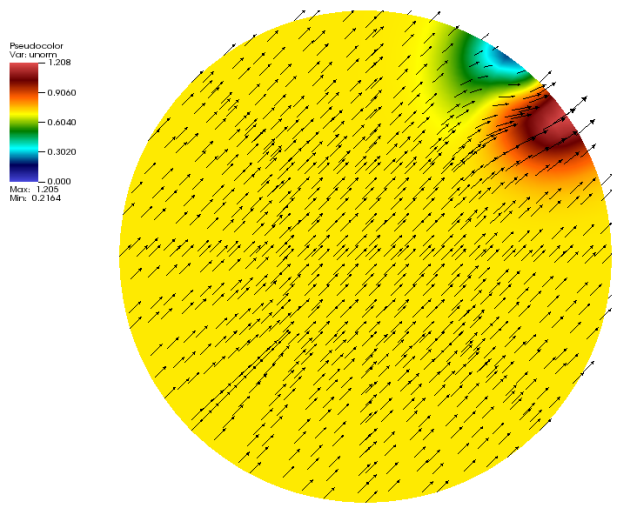
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field



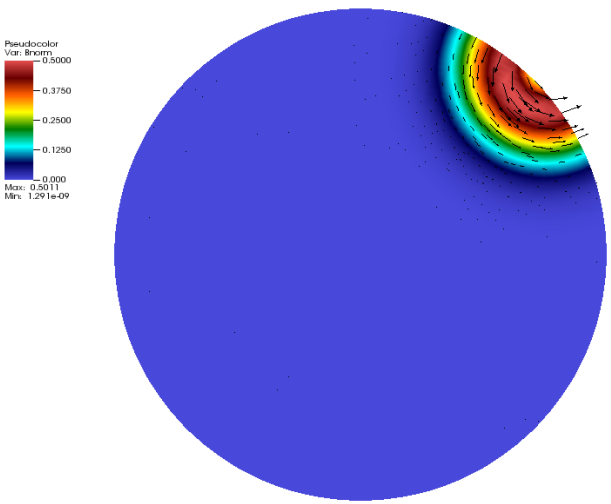
Velocity



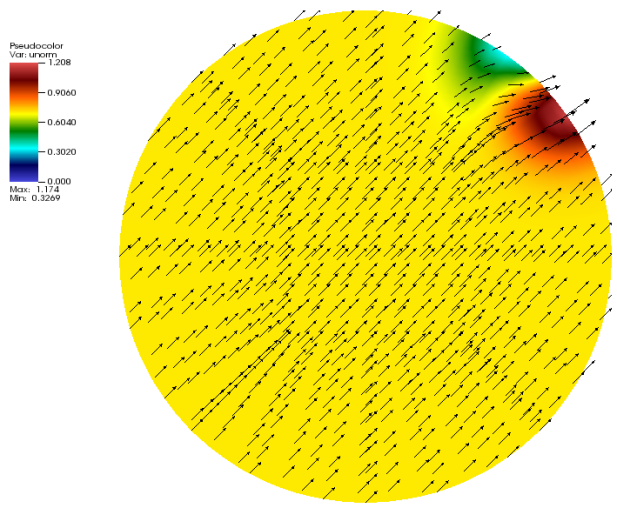
2D MHD drifting vortex

Parameters : $\rho = 1.0, p_0 = 1,\ u_0 = b_0 = 0.5,\ \mathbf{u}_{drift} = [1,1]^t,$
$h(r) = exp[(1 - r^2)/2]$

Magnetic field

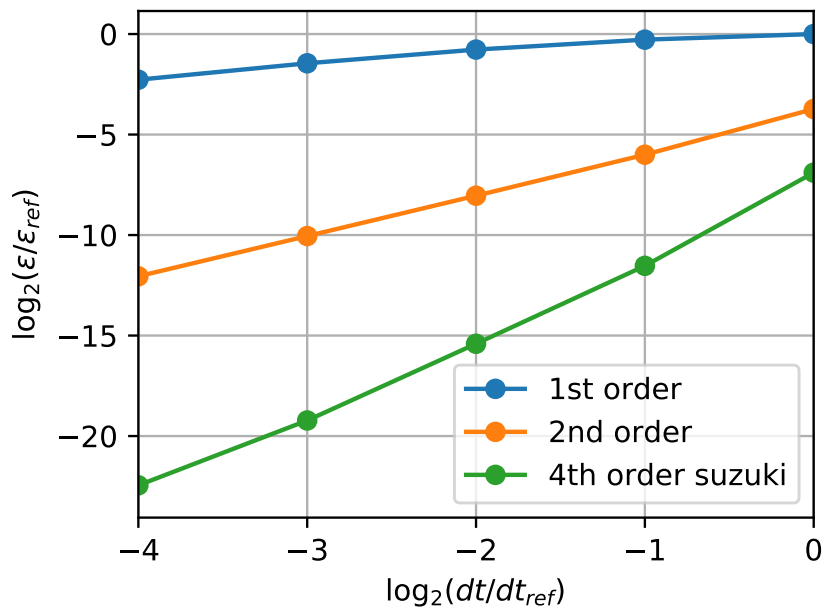


Velocity



MHD drifting vortex : time order check

- 2D disk mesh : 20 macro-elements
- refinement 8 , degree 5 : 2304 quadrature points per element
- six scalar fields (no div B cleaning)
- dt ranges from $dt_{ref} = 0.2$ to $dt = 0.006250$
- $hmin \approx 0.02$, discrete velocity norm $\|v_i\| = 4$;



Conclusion and prospects

- Explicit kinetic schemes with no CFL condition.
- low-storage : stores only one time-step of the solution.
- High order time integrator compatible with $\tau \to 0$.
- StarPU Task-based parallelization of the solver.

Work in progress:

- 3D MagnetoHydroDynamics.
- Boundary conditions.
- Optimizations: GPU codelets.
- Replacement of the DG solver with a Semi-Lagrangian solver.

Bibliography I

[AAF+12] Cédric Augonnet, Olivier Aumage, Nathalie Furmento, Raymond Namyst, and Samuel Thibault.

StarPU-MPI: Task Programming over Clusters of Machines Enhanced with Accelerators.

In Siegfried Benkner Jesper Larsson Träff and Jack Dongarra, editors, *EuroMPI 2012*, volume 7490 of *LNCS*. Springer, September 2012.

Poster Session.

[ADN00] Denise Aregba-Driollet and Roberto Natalini.

Discrete kinetic schemes for multidimensional systems of conservation laws.

SIAM Journal on Numerical Analysis, 37(6):1973–2004, 2000.

[Bou99] François Bouchut.

Construction of BGK models with a family of kinetic entropies for a given system of conservation laws.

Journal of Statistical Physics, 95(1-2):113–170, 1999.

[CFH+17] David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger, and Laurent Navoret.

Palindromic Discontinuous Galerkin Method, pages 171–178.

Springer International Publishing, Cham, 2017.

[DR78] Iain S Duff and John Ker Reid.

An implementation of tarjan's algorithm for the block triangularization of a matrix.

ACM Transactions on Mathematical Software (TOMS), 4(2):137–147, 1978.

Bibliography II

[Gra14] Benjamin Graille.

Approximation of mono-dimensional hyperbolic systems: A lattice Boltzmann scheme as a relaxation method.

Journal of Computational Physics, 266:74–88, 2014.

[HLW06] Ernst Hairer, Christian Lubich, and Gerhard Wanner.

Geometric numerical integration: structure-preserving algorithms for ordinary differential equations, volume 31.

Springer Science & Business Media, 2006.

[Jin95] Shi Jin.

Runge-kutta methods for hyperbolic conservation laws with stiff relaxation terms.

Journal of Computational Physics, 122(1):51–67, 1995.

[JNP84] Claes Johnson, Uno Nävert, and Juhani Pitkäranta.

Finite element methods for linear hyperbolic problems.

Computer methods in applied mechanics and engineering, 45(1):285–312, 1984.

[JX95] Shi Jin and Zhouping Xin.

The relaxation schemes for systems of conservation laws in arbitrary space dimensions.

Communications on pure and applied mathematics, 48(3):235–276, 1995.

[KL97] William Kahan and Ren-Cang Li.

Composition constants for raising the orders of unconventional schemes for ordinary differential equations.

Mathematics of Computation of the American Mathematical Society, 66(219):1089–1099, 1997.

Bibliography III

[MQ02] Robert I McLachlan and G Reinout W Quispel.

Splitting methods.

Acta Numerica, 11:341–434, 2002.

[NL08] Jostein R Natvig and Knut-Andreas Lie.

Fast computation of multiphase flow in porous media by implicit discontinuous galerkin schemes with optimal ordering of elements.

Journal of Computational Physics, 227(24):10108–10124, 2008.

[PR05] Lorenzo Pareschi and Giovanni Russo.

Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation.

Journal of Scientific computing, 25(1-2):129–155, 2005.

[Suz90] Masuo Suzuki.

Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations.

Physics Letters A, 146(6):319–323, 1990.

[WX99] Feng Wang and Jinchao Xu.

A crosswind block iterative method for convection-dominated problems.

SIAM Journal on Scientific Computing, 21(2):620–645, 1999.