Kinetic schemes for conservation laws. Applications to MHD and Maxwell

Philippe Helluy

Université de Strasbourg, IRMA CNRS, Inria Tonus

May 19, 2021

Contents

Kinetic algorithm

Conservation laws

System of m conservation laws in dimension d

$$\partial_t W + \sum_{i=1}^{d} \partial_i Q^i(W) = 0,$$

- Unknown: $W(X,t) \in \mathbb{R}^m$, space variable: $X = (x_1 \dots x_d)$, time variable: t.
- $\partial_i = \frac{\partial}{\partial x_i}$, $\partial_t = \frac{\partial}{\partial t}$.
- Hyperbolicity: let $N \in \mathbb{R}^d$ be an arbitrary space direction. The flux

$$Q(W,N) = \sum_{i=1}^{d} N_i Q^i(W)$$

is supposed to be hyperbolic, i.e. the jacobian of the flux $d_W Q(W,N)$ is diagonalizable with real eigenvalues.

Kinetic representation

We consider a set of $d+1$ (or more) kinetic velocities V_k, $k = 0 \dots d+1$, associated to **vectorial** kinetic functions $F_k(W) \in \mathbb{R}^m$. The macroscopic data are related to the kinetic data by

$$W = \sum_k F_k.$$

We also define "Maxwellian" equilibrium functions $M_k(W) \in \mathbb{R}^m$ such that

$$W = \sum_k M_k(W).$$

The kinetic BGK representation is given by transport equations with relaxation source terms [Bou99, ADN00]

$$\partial_t F_k + V_k \cdot \nabla_X F_k = \frac{1}{\tau} \left(M_k(W) - F_k \right).$$

Kinetic representation

When the relaxation time $\tau \to 0^+$, the kinetic model is formally equivalent to the initial system of conservation laws if

$$W = \sum_k M_k(W), \quad \sum_k V_k^i M_k(W) = Q^i(W), \quad i = 1, \ldots, d.$$

- Linear system of size $m(d+1) \times m(d+1)$ for finding the Maxwellian. One expects a unique solution.
- In practice it is difficult to solve directly the BGK system. A splitting is preferable.

Kinetic algorithm

1. Start with $W(\cdot, 0)$. Construct kinetic vectors $F_k(\cdot, 0)$ such that $W = \sum_k F_k$ (not unique);
2. solve the free transport equations $\partial_t F_k + V_k^i \cdot \partial_i F_k = 0$ for a duration of Δt. This gives

$$F_k(X, \Delta t^-) = F_k(X - \Delta t V_k, 0);$$

3. define

$$W(\cdot, \Delta t) = \sum_k F_k(\cdot, \Delta t^-);$$

4. apply a relaxation

$$F_k(\cdot, \Delta t^+) = \omega M_k(W(\cdot, \Delta t)) + (1 - \omega) F_k(\cdot, \Delta t^-).$$

Interesting cases: $\omega = 1$, $\omega = 2$, ω is a matrix.

Flux error representation

During the computations, one expects that $\sum_k V_k^i F_k \simeq Q^i(W)$, we thus introduce the **approximated flux** Z**,** and the **flux error** Y

$$Z^i = \sum_k V_k^i F_k, \quad Y^i = Z^i - Q^i(W).$$

The kinetic algorithm is then a functional operator $\mathcal{M}(\Delta t)$ that maps $(W(\cdot, 0), Y^i(\cdot, 0))$ to $(W(\cdot, \Delta t), Y^i(\cdot, \Delta t^+))$.

1. The operator $\mathcal{M}$ is made of (linear) shift operations and (non-linear) local relaxations.
2. In the (W, Y^i) variables, when $\omega = 2$, the relaxation operation simply reads
$$Y^i(\cdot, \Delta t^+) = -Y^i(\cdot, \Delta t^-).$$
This induces fast oscillations of the flux error. For the analysis it is better to replace $\mathcal{M}$ by $\mathcal{M} \circ \mathcal{M}$.

Equivalent equation

In principle it is now easy, while tedious, to compute the equivalent equation of the kinetic algorithm: simply compute a Taylor expansion of

$$\frac{\mathcal{M}(\Delta t/2) - \mathcal{M}(-\Delta t/2)}{\Delta t}$$

with respect to Δt up to order $O(\Delta t^2)$.

- The term $X - \Delta t V_k$ in the shift operation generates partial derivatives in space.
- Because of symmetries, when $\omega = 2$, the even order terms vanish.
- And finally, the relaxation introduce non linearities. We end up with a system of non-linear conservation laws of first order in (W, Y^i).

One-dimensional case

We consider the case $d = 1$, we have $d + 1 = 2$ kinetic velocities. We take $V_0 = -\lambda$, $V_1 = \lambda$, $W = F_0 + F_1$. The equivalent equation for $\omega = 2$ at order $O(\Delta t^2)$ is

$$\partial_t W + \partial_x Q(W) = 0,$$

(of course) and

$$\partial_t Y - d_W Q(W) \partial_x Y = 0.$$

- We observe that the system is hyperbolic and that the waves for W and Y move in opposite direction.
- No assumption of smallness of Y!
- It is necessary to analyze the third order terms to find the sub-characteristic stability condition

$$\lambda \geq \max_{1 \leq i \leq m} |\lambda_i(W)|,$$

where $\lambda_i(W)$ are the eigenvalues of $d_W Q(W)$.

Two-dimensional case

We consider the case $d = 2$, we have $d + 1 = 3$ kinetic velocities. We take

$$V_k = \lambda \begin{pmatrix} \cos \frac{2k\pi}{3} \\ \sin \frac{2k\pi}{3} \end{pmatrix}$$

The equivalent equation on W for $\omega = 2$ at order $O(\Delta t^2)$ is

$$\partial_t W + \partial_1 Q^1(W) + \partial_2 Q^2(W) = 0,$$

(of course). Setting $A^i(W) = d_W Q^i(W)$, the equation for Y is

$$\partial_t \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} + \begin{pmatrix} \frac{\lambda}{2} I - A^1(W) & 0 \\ -A^2(W) & -\frac{\lambda}{2} \end{pmatrix} \partial_1 \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} + \begin{pmatrix} 0 & -\frac{\lambda}{2} I - A^1(W) \\ -\frac{\lambda}{2} & -A^2(W) \end{pmatrix} \partial_2 \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = O(\Delta t^2),$$

We can prove that the equivalent system is symmetrizable, and thus hyperbolic, if λ is large enough (sub-characteristic condition).

Magnetohydrodynamics

Magnetohydrodynamics (MHD) equations

We consider the MHD equations with Divergence Cleaning [DKK+02]

$$W = \begin{pmatrix} \rho \\ \rho U \\ \mathcal{E} \\ B \\ \psi \end{pmatrix}, \quad Q(W, N) = \begin{pmatrix} \rho U \cdot N \\ \rho (U \cdot N) U + (p + \frac{B \cdot B}{2}) N - (B \cdot N) B \\ (\mathcal{E} + p + \frac{B \cdot B}{2}) U \cdot N - (B \cdot U)(B \cdot N) \\ (U \cdot N) B - (B \cdot N) U + \psi N \\ c_h^2 B \cdot N \end{pmatrix}$$

The velocity and magnetic field are denoted

$$U = (u_1, u_2, u_3)^T, \quad B = (b_1, b_2, b_3)^T,$$

the pressure is given by a perfect-gas law with a constant polytropic exponent $\gamma > 1$

$$p = (\gamma - 1)(\mathcal{E} - \rho \frac{U \cdot U}{2} - \frac{B \cdot B}{2}).$$

Lattice-Boltzmann scheme

In the Lattice-Boltzmann approach, one considers a structured cartesian grid of step Δx and four kinetic velocities

$$V_k = \lambda \begin{pmatrix} \cos \frac{k\pi}{2} \\ \sin \frac{k\pi}{2} \end{pmatrix}, \quad k = 0, \dots, 3.$$

- Because $4 > d + 1$, the Maxwellian is not unique [Fév14]
- With a time-step $\Delta t = \Delta x / \lambda$, the transport steps can be exactly solved by simple shifts.
- The scheme is then the scheme of [Dub14].

GPU implementation

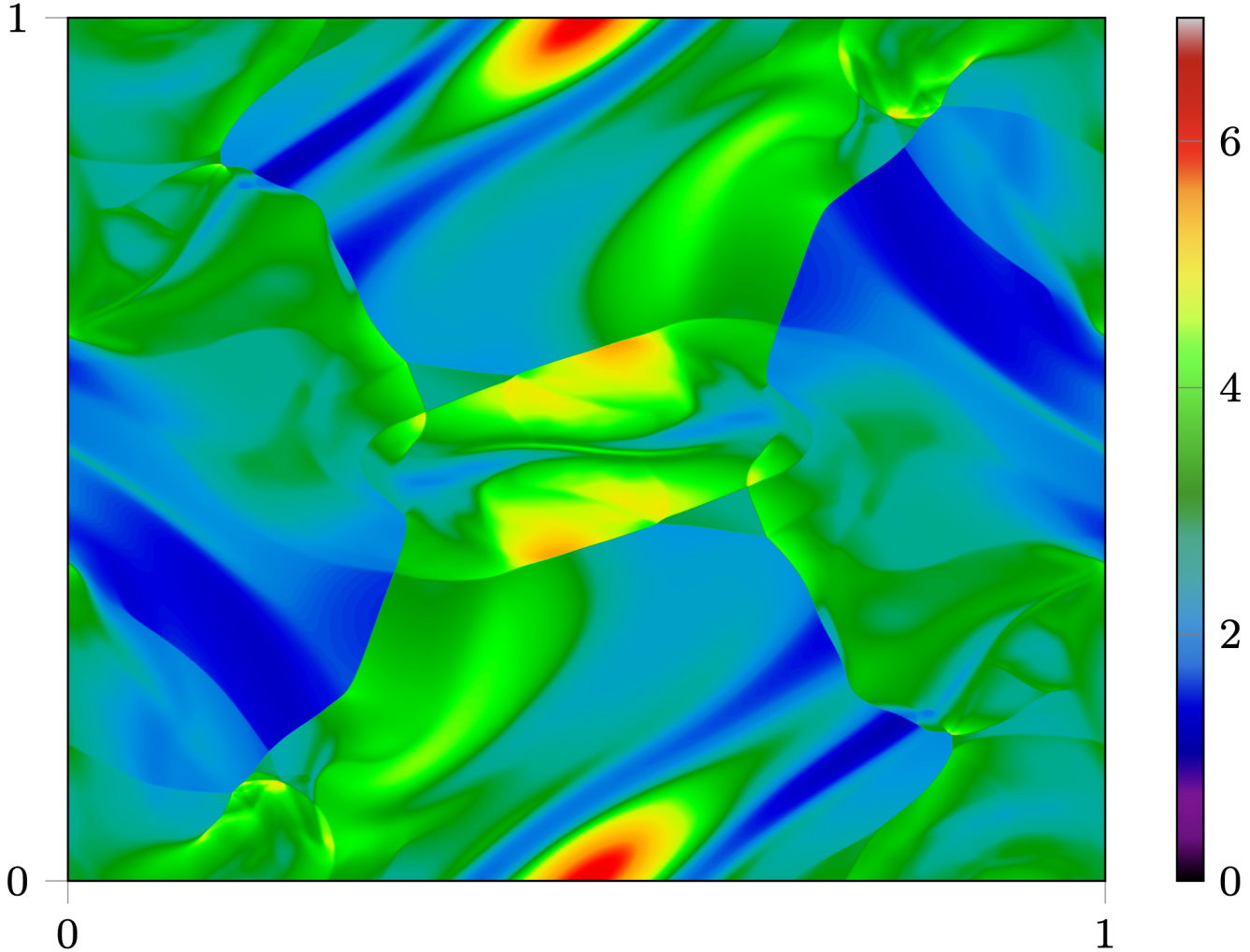
We have implemented the LBM method with PyOpenCL and an efficient memory layout. The algorithm is memory bound. We are not far from the optimal memory bandwidth [BDF+21].

	prec.	b (GB/s, shift-only)	b (GB/s, shift-relax)	max. theoretical b (GB/s)
Intel	float32	17.58	13.38	60
Intel	float64	19.12	17.48	60
Iris 640	float32	26.20	24.98	34
Iris 640	float64	20.08	3.78	34
GTX	float32	147.54	146.94	192
GTX	float64	148.76	49.72	192
Quadro	float32	336.45	329.06	432
Quadro	float64	344.50	127.21	432
V100	float32	692.31	676.44	900
V100	float64	705.88	610.17	900

Table 2 Bandwidth efficiency of the LBM algorithm. Comparison of the data transfer rates of the shift-only algorithm and of the shift-and-relaxation algorithm. The resulting bandwidth is compared with the maximal memory bandwidth advertised by the vendors of the hardware devices.

Numerical results

Orszag-Tang vortex on a 4096×4096 grid and $\omega = 1.8$



Tilt instability

Test case described in [BDF+21].

- Instability with appearance of a current sheet (Dirac measure on a curve).
- We measure the correct instability rate.

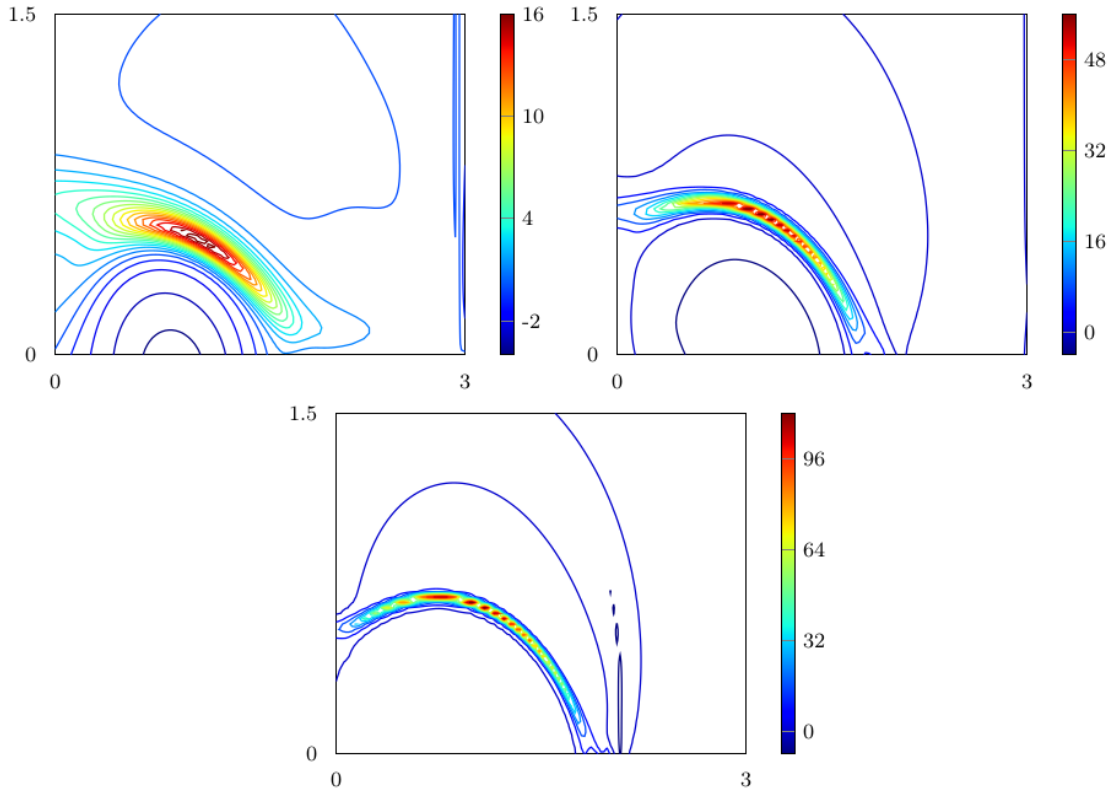


Fig. 9 Snapshots of the magnetic current density recorded at time $t = 6$ in a spatial zone around the current sheets. Grid sizes are $Nx \times Ny = 256 \times 256$, $Nx \times Ny = 1024 \times 1024$, $Nx \times Ny = 4096 \times 4096$.

Explicit, CFL-free, Discontinuous Galerkin

Discontinuous Galerkin (DG) solver

The main task is to solve a single transport equation

$$\partial_t f + V \cdot \nabla f = 0$$

in a domain Ω with a complex geometry. The characteristic method is no more a good idea (problems with stability, conservation, boundaries).

- Unstructured mesh of Ω made of tetrahedral cells.
- The transported function f is approximated in cell L by a linear expansion on basis functions

$$f(x, n\Delta t) \simeq f_L^n(x) = \sum_j f_{L,j}^n \psi_j^L(x), \quad x \in L. \tag{1}$$

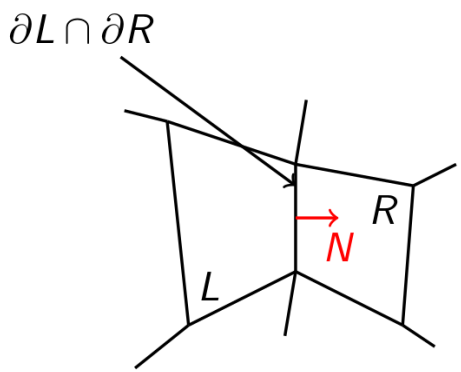
- The unknowns are the coefficients $f_{L,j}^n$ of the linear expansion.

Implicit DG

Implicit DG approximation scheme for going from time step $n-1$ to time step n: for all cell L and $\forall i$,

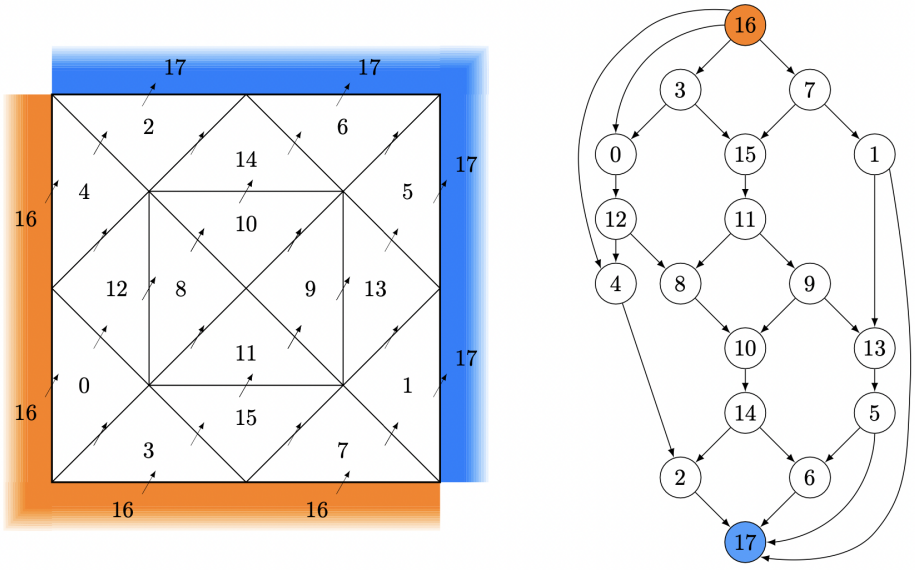
$$\int_L \frac{f_L^n - f_L^{n-1}}{\Delta t}\psi_i^L - \int_L V\cdot\nabla\psi_i^L f_L^n + \int_{\partial L}\left(V\cdot N^+ f_L^n + V\cdot N^- f_R^n\right)\psi_i^L = 0. \tag{2}$$

- R denotes the neighbor cells along ∂L.
- $V\cdot N^+ = \max(V\cdot N, 0)$, $V\cdot N^- = \min(V\cdot N, 0)$. We thus use an upwind numerical flux.
- N is the unit normal vector on ∂L oriented from L to R.



Downwind algorithm

The scheme seems to be implicit, but it is actually explicit.



- The solution can be explicitly computed by following a topological ordering of a Direct Acyclic Graph (DAG), e.g. 3, 7, 0, 15, 1, etc.
- In addition there is parallelism: (3,7) can be computed in parallel, then (0,15,1) can be computed in parallel, etc.
- Low storage: the solution can be replaced in memory during the computations.

Rust implementation

We have implemented the downwind algorithm in Rust:

- Recent programming language (2010) oriented towards concision, speed and security.
- Most common bugs are avoided at compile time: memory leaks and segfaults, uninitialized data, race conditions.
- Automatic parallelization tools: if the sequential code works, the parallel version is guaranteed to be correct.
- Fast. More details in [GHMD21].

			Error e_r		CPU (s)	
Method	CFL β	Δt	$\nu = 2$	$\nu = 5$	1 thread	24 threads
RK3DG	0.37	0.00009	0.00070	0.01238	4,607.95	785.28
D3Q4P	0.37	0.00009	0.00103	0.01467	1,524.45	234.48
RK3DG	0.93	0.00023	0.00070	0.01238	2,189.76	384.79
D3Q4P	0.93	0.00023	0.00103	0.01467	613.44	90.84
RK3DG	1.85	0.00046	0.00070	0.01238	1,121.96	212.60
D3Q4P	1.85	0.00046	0.00103	0.01467	304.41	45.14
D3Q4P	3.70	0.00091	0.00103	0.01468	153.09	22.40
D3Q4P	9.25	0.00228	0.00104	0.01479	61.60	8.96
D3Q4P	18.50	0.00456	0.00115	0.01619	30.76	4.53
D3Q4P	37.00	0.00912	0.00210	0.02992	15.34	2.46
D3Q4P	92.50	0.02281	0.01107	0.16589	6.17	0.92
D3Q4P	185.00	0.04562	0.04509	0.40344	3.10	0.48

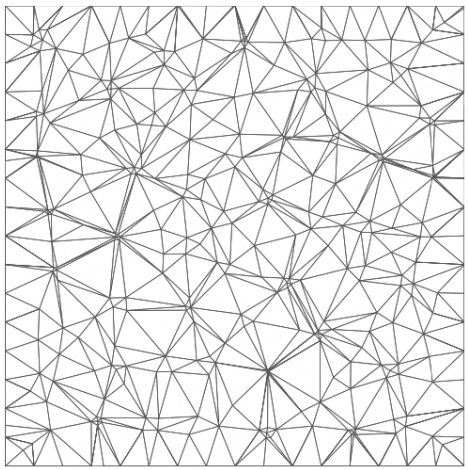
Numerical results

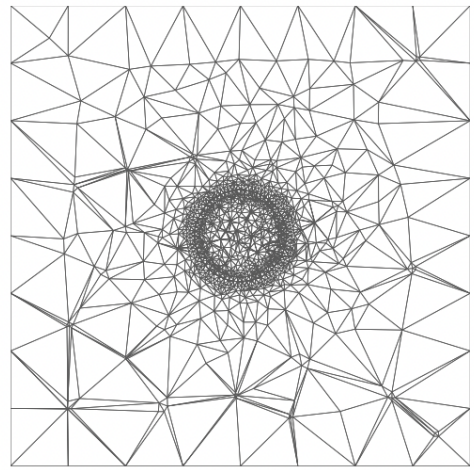
- Maxwell's equation: $W = (E^T, H^T)^T$, with electric field $E \in \mathbb{R}^3$ and magnetic field $H \in \mathbb{R}^3$.
- Maxwell flux:

$$Q(W, N) = \begin{pmatrix} -N \times H \\ N \times E \end{pmatrix}.$$

.

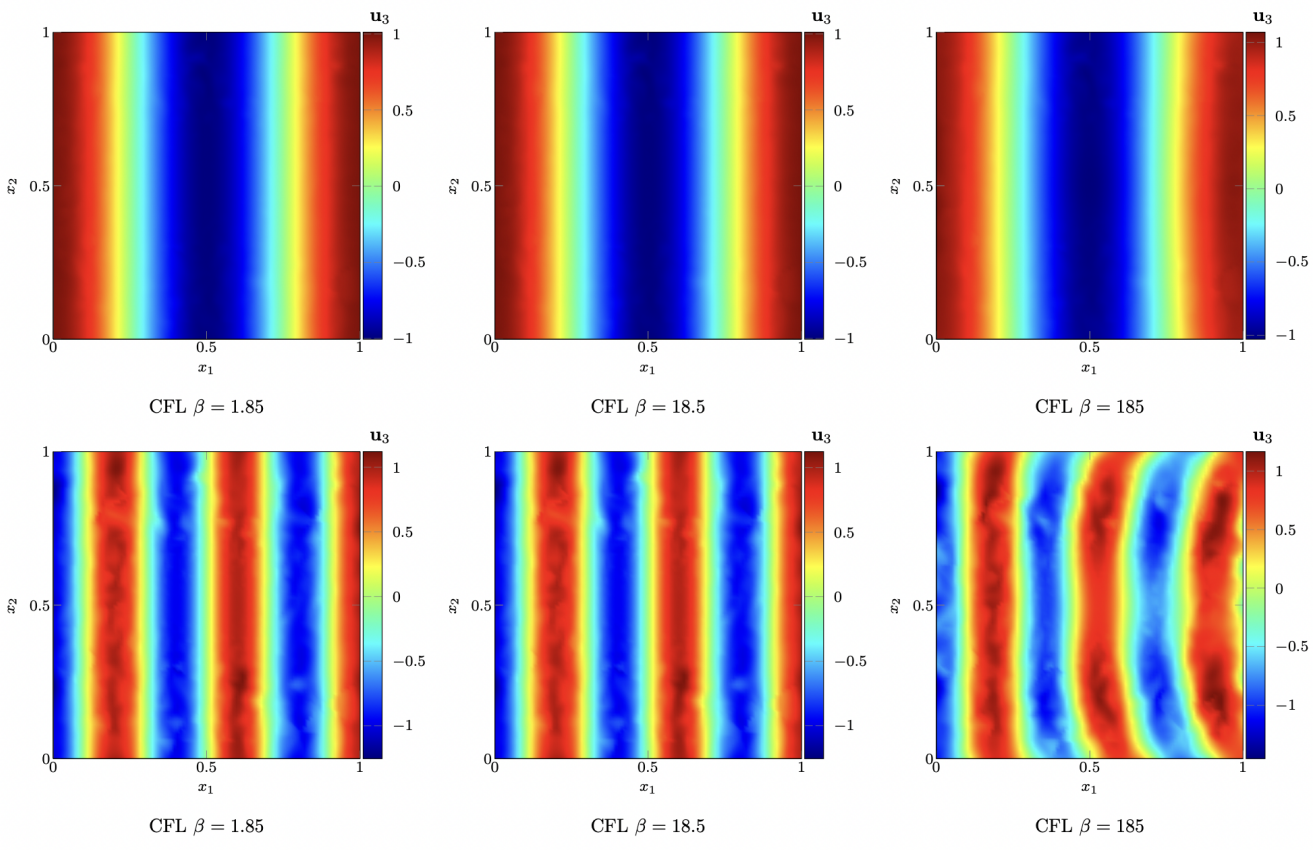
- Unstructured mesh of the unit cube made with large and small cells





Numerical results

- We compute a plane wave on the above mesh.
- We check second order accuracy and the CFL-less feature



Conclusion

The kinetic approach is a generalization of the LBM:

- Can handle arbitrary Mach flows.
- Very fast computations on Cartesian grids.
- Construction of CFL-free explicit DG scheme on unstructured meshes.
- Equivalent equation: useful theoretical tool (stability analysis, boundary conditions, behavior of hidden variables).

Future works: boundary conditions, better parallelism (IMEX approach), other applications (cavitation).

Bibliography I

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.

Hubert Baty, Florence Drui, Emmanuel Franck, Philippe Helluy, Christian Klingenberg, and Lukas Thanhäuser.

A robust and efficient solver based on kinetic schemes for Magnetohydrodynamics (MHD) equations.
https://hal.archives-ouvertes.fr/hal-02965967, April 2021.

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.

Andreas Dedner, Friedemann Kemm, Dietmar Kröner, C-D Munz, Thomas Schnitzer, and Matthias Wesenberg.

Hyperbolic divergence cleaning for the MHD equations.
Journal of Computational Physics, 175(2):645–673, 2002.

François Dubois.

Simulation of strong nonlinear waves with vectorial lattice boltzmann schemes.
International Journal of Modern Physics C, 25(12):1441014, 2014.

Tony Février.

Extension et analyse des schémas de Boltzmann sur réseau: les schémas à vitesse relative.
PhD thesis, Université Paris Sud-Paris XI, 2014.

Pierre Gerhard, Philippe Helluy, and Victor Michel-Dansac.

CFL-less Discontinuous Galerkin solver.
https://hal.archives-ouvertes.fr/hal-03218086, May 2021.