



HAL
open science

High-order implicit palindromic discontinuous Galerkin method for kinetic-relaxation approximation

David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger,
Laurent Navoret

► **To cite this version:**

David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger, Laurent Navoret. High-order implicit palindromic discontinuous Galerkin method for kinetic-relaxation approximation. *Computers and Fluids*, 2019, 190, pp.485–502. 10.1016/j.compfluid.2019.06.007 . hal-01706614

HAL Id: hal-01706614

<https://hal.science/hal-01706614>

Submitted on 12 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High-order implicit palindromic discontinuous Galerkin method for kinetic-relaxation approximation

David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger,
Laurent Navoret

IRMA, Univ. Strasbourg, 7 rue Descartes, Strasbourg, France & Inria TONUS

Abstract

We construct a high order discontinuous Galerkin method for solving general hyperbolic systems of conservation laws. The method is CFL-less, matrix-free, has the complexity of an explicit scheme and can be of arbitrary order in space and time. The construction is based on: (a) the representation of the system of conservation laws by a kinetic vectorial representation with a stiff relaxation term; (b) a matrix-free, CFL-less implicit discontinuous Galerkin transport solver; and (c) a stiffly accurate composition method for time integration. The method is validated on several one-dimensional test cases. It is then applied on two-dimensional and three-dimensional test cases: flow past a cylinder, magnetohydrodynamics and multifluid sedimentation.

Keywords: discontinuous Galerkin; implicit scheme; matrix-free; composition method; high order; stiff PDE.

Contents

1	Introduction	2
2	Kinetic relaxation approximation	4
2.1	Kinetic BGK equation	4
2.2	Examples	7
2.2.1	Vectorial kinetic method	7
2.2.2	Other Lattice Boltzmann methods	9

[☆]This work has been supported by Eurofusion, ANR project “EXAMAG” and BPI France project “HOROCHE”.

Email address: helluy@unistra.fr (David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger, Laurent Navoret)

3	Implicit high-order palindromic time-stepping	11
3.1	First order splitting	11
3.2	Second-order stiffly accurate splitting	13
3.3	High-order palindromic splitting	14
4	Implicit discontinuous Galerkin method for linear transport	15
4.1	DG approximation	15
4.2	Triangular structure of the transport matrix	19
5	Parallel implementation	22
6	Numerical results	22
6.1	One-dimensional isothermal Euler test cases	23
6.1.1	Smooth solution	23
6.1.2	Behaviour for discontinuous solutions	23
6.2	MHD flow	24
6.3	Flow past a cylinder (“thick” boundary condition)	26
6.3.1	No-slip boundary condition	28
6.3.2	Numerical simulations	28
6.4	Two-dimensional and three-dimensional two-fluid flow	29
6.4.1	Two-fluid flow with gravity	29
6.4.2	Two-dimensional Rayleigh-Taylor instability in an annulus.	32
6.4.3	Three-dimensional Rayleigh-Taylor instability in a cylinder.	33
7	Conclusion	35
8	Appendix	37
8.1	Second order approximation	37

1. Introduction

Systems of conservation laws are important mathematical tools for modelling many phenomena in physics or engineering.

In several practical applications, only some time scales of the model are interesting and one would like to filter out the smallest time scales. Classical explicit methods require very small time steps, because of the CFL stability condition. A standard way to treat the various time scales is to use an implicit time-stepping scheme. Schemes of this kind, however, are quite challenging from a computational point of view: they require inverting large non-linear systems, which induce high computational and storage costs.

In this paper, we propose an alternative method for solving systems of conservation laws for a large range of time scales on complex geometries. The time-marching procedure is CFL-less, but keeps the complexity of an explicit scheme. In addition, we are able to achieve high order in space and time.

Our method is based on a vectorial kinetic relaxation scheme described in [3, 7, 38, 34]. The vectorial kinetic scheme is a generalization of the relaxation scheme of Jin and Xin [26]. The original system of conservation laws is replaced by an equivalent kinetic system made of a small set of transport equations, coupled through a stiff relaxation source term. The stiffness is measured with a small relaxation time $\tau > 0$. The original system of conservation laws is equivalent to its kinetic representation in the limit $\tau \rightarrow 0$.

Many approaches have been proposed in the literature for solving such kinetic models. It is generally approximated by a splitting method, in which the transport and relaxation steps are treated separately. A simple and natural choice for solving the stiff relaxation step is then to apply a first order implicit scheme in order to avoid instabilities. On the other hand, the transport step can be solved with several different methods: explicit upwind schemes [8, 13, 3]; exact characteristic schemes (which are at the base of the Lattice Boltzmann Method [39, 11, 22]); but also finite volume, finite difference or discontinuous Galerkin methods (see for instance [33, 37, 40, 30]).

From a computational point of view, the splitting approach has several advantages: the transport equations are uncoupled, linear and can be solved with efficient parallel solvers; the relaxation step is also embarrassingly parallel and requires only to considering local ordinary differential equations.

However, the first order splitting introduces too much numerical diffusion for practical applications. Therefore many works have been devoted to the construction of higher order schemes based on improved splitting approaches.

In the Lattice Boltzmann Method, the accuracy is improved if the relaxation step is solved with a Crank-Nicolson scheme [16]. The transport and relaxation steps are then interlaced with a Strang procedure. Because of the stiff relaxation, it has been observed that without special care in the scheme design, one can observe order reduction when the relaxation time $\tau \rightarrow 0$ [24]. It is, however, possible to construct high order Runge-Kutta schemes, mixing implicit and explicit steps [36] that preserve the accuracy when $\tau \rightarrow 0$ (Asymptotic Preserving property [25]).

In all the above approaches, because the transport step is solved by an explicit scheme the whole procedure is still constrained by a CFL condition on the time step.

The first fundamental aspect of our method is to apply an *implicit* Discontinuous Galerkin (DG) method instead of an explicit one for solving the transport equations. In this way, we obtain unconditionally stable schemes and get rid of the CFL condition. The implicit solver has almost no additional cost compared to the explicit one. Indeed, with an upwind numerical flux, the linear system of the implicit DG method is triangular and, in the end, can be solved

explicitly. This kind of ideas is mentioned in several works. See for instance in [6, 42, 12, 35, 32]. In a recent work we have evaluated the parallel scalability of the triangular solver [5].

The second fundamental aspect of our method is the construction of a *symmetric-in-time* integrator that remains second order accurate even for vanishing relaxation time τ (AP property). The construction is based on a modified Crank-Nicolson procedure and on essential reversibility properties of the transport equation. Once a symmetric-in-time integrator is available, it is then very easy to construct arbitrary order methods with the composition method [41, 28, 29, 21]. We apply this method for achieving fourth and sixth order time integration even for vanishing relaxation time τ .

The objective of this paper is first to present the whole construction of the Palindromic Discontinuous Galerkin Method. Then we will establish some rigorous properties of the scheme in the simplified linear case. We will validate the approach on several one-dimensional test cases. Finally, we will apply it in 2D and 3D for computing Von Karmann streets and multi-fluid instabilities.

2. Kinetic relaxation approximation

We consider a system of m conservation laws in D space dimension. The unknown $\mathbf{w}(\mathbf{x}, t) \in \mathbb{R}^m$, depending on space $\mathbf{x} = (x^1, \dots, x^D) \in \mathbb{R}^D$ and time $t > 0$, satisfies the following system

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

where $\mathbf{q}^k(\mathbf{w}) \in \mathbb{R}^m$ are the fluxes in the k -th spatial direction with $1 \leq k \leq D$ and $s(\mathbf{w}) \in \mathbb{R}^m$ is a general source term. For any function $g(\mathbf{x}, t)$, $\partial_k g(\mathbf{x}, t)$ stands for the partial derivative of g with respect to x^k .

The kinetic BGK representation aims at considering (1) as a singular limit of a linear kinetic equation with a source term.

2.1. Kinetic BGK equation

The macroscopic quantity $\mathbf{w}(\mathbf{x}, t) \in \mathbb{R}^m$ is associated to a vectorial distribution function $\mathbf{f}(\mathbf{x}, t) \in \mathbb{R}^{n_v}$, with $n_v > m$, through a linear transformation

$$\mathbf{w} = \mathbf{P}\mathbf{f}, \quad (2)$$

where \mathbf{P} is a constant $m \times n_v$ matrix. Each component of $\mathbf{f}(\mathbf{x}, t)$ corresponds to a discrete velocity, denoted $\mathbf{v}_i = (v_i^1, \dots, v_i^D) \in \mathbb{R}^D$ for $1 \leq i \leq n_v$. This distribution function satisfies the following (kinetic) equation

$$\partial_t \mathbf{f} + \sum_{k=1}^D \mathbf{v}^k \partial_k \mathbf{f} = \frac{1}{\tau} (\mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) - \mathbf{f}) + \mathbf{g}(\mathbf{f}). \quad (3)$$

where, for all $1 \leq k \leq D$, $\mathbf{V}^k \in M_{n_v}(\mathbb{R})$ are diagonal matrices composed of the k -th components of the discrete velocities

$$\mathbf{V}^k = \begin{pmatrix} v_1^k & & & \\ & v_2^k & & \\ & & \ddots & \\ & & & v_{n_v}^k \end{pmatrix}.$$

The right-hand side of (3) involves a generic source term $\mathbf{g}(\mathbf{f})$ and a BGK relaxation term: $\mathbf{N}^\tau \mathbf{f} = (\mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) - \mathbf{f})/\tau$, where $\tau \ll 1$ is a small parameter. In other words, equation (3) is a coupling of n_v transport equations at constant velocities.

The relaxation term is devised so that the macroscopic quantity $\mathbf{w} = \mathbf{P}\mathbf{f}$ converges to the solution to equation (1) as $\tau \rightarrow 0$. To ensure such behaviour, the equilibrium distribution $\mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f})$ only depends on the macroscopic quantity and satisfy:

$$\mathbf{w} = \mathbf{P}\mathbf{f}^{\text{eq}}(\mathbf{w}). \quad (4)$$

Consequently, multiplying equation (3) by \mathbf{P} makes the singular relaxation term vanish and we get

$$\partial_t \mathbf{P}\mathbf{f} + \sum_{k=1}^D \partial_k (\mathbf{P}\mathbf{V}^k \mathbf{f}) = \mathbf{P}\mathbf{g}(\mathbf{f}). \quad (5)$$

Since \mathbf{f} formally tends to $\mathbf{f}^{\text{eq}}(\mathbf{w})$ as $\tau \rightarrow 0$, we recover the system of conservation (1) in the limit provided that we have the following relation

$$\mathbf{q}^k(\mathbf{w}) = \mathbf{P}\mathbf{V}^k \mathbf{f}^{\text{eq}}(\mathbf{w}), \quad (6)$$

$$\mathbf{s}(\mathbf{w}) = \mathbf{P}\mathbf{g}(\mathbf{f}^{\text{eq}}(\mathbf{w})). \quad (7)$$

In the next section, we will give examples of such constructions.

As shown in [3], at the first order in τ , the kinetic relaxation system is consistent with

$$\begin{aligned} \partial_t \mathbf{w} + \sum_{k=1}^D \partial_k (\mathbf{q}^k(\mathbf{w})) &= \mathbf{s} + \tau \sum_{k=1}^D \sum_{j=1}^D \partial_k [\mathcal{D}^{kj}(\mathbf{w}) \partial_j \mathbf{w}] \\ &+ \tau \left[\sum_{k=1}^D \partial_k \left(\mathbf{P}\mathbf{V}^k \left[\nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}(\mathbf{w}) \mathbf{s}(\mathbf{w}) - \mathbf{g}(\mathbf{f}^{\text{eq}}(\mathbf{w})) \right] \right) \right] + \mathcal{O}(\tau^2), \quad (8) \end{aligned}$$

where the diffusion tensor \mathcal{D} is defined by

$$\mathcal{D}^{kj} = \mathbf{P}\mathbf{V}^k \mathbf{V}^j \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}} - \nabla_{\mathbf{w}} \mathbf{q}^k \nabla_{\mathbf{w}} \mathbf{q}^j. \quad (9)$$

For the sake of completeness, the proof of this estimate is provided in Appendix 8.1.

In view of (8), it is particularly interesting to take the kinetic source equal to

$$\mathbf{g}(\mathbf{f}) = \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) \mathbf{s}(\mathbf{P}\mathbf{f}). \quad (10)$$

This choice directly guaranties the consistency of the source (7) since (4) implies that $\mathbf{P}\nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}$ reduces to the identity matrix on \mathbb{R}^m . This particular form of the source term is actually equivalent to only make the macroscopic part of the distribution function evolve (see Remark 1 below).

The stability of kinetic relaxation models is discussed in [3, 7, 10]. The mere dissipation of the L^2 norm, which requires the symmetric part of the diffusion tensor to be positive, is not sufficient for nonlinear hyperbolic systems. For such systems, a more appropriate criterion is the dissipation of an entropy: in its strongest form, it requires the existence of a strictly convex entropy for the kinetic system. A weaker requirement is the dissipation of a macroscopic entropy by the approximated system at the first order in the Chapman-Enskog expansion. We assume the existence of a convex entropy-flux pair $(\eta(\mathbf{w}), \mathbf{Q}^k(\mathbf{w}))$ for (1) and note $\nabla_{\mathbf{w}}^2 \eta$ the Hessian matrix of the entropy. From (8), we have

$$\begin{aligned} \partial_t \eta(\mathbf{w}) + \sum_{k=1}^D \partial_k (\mathbf{Q}^k(\mathbf{w})) + \nabla_{\mathbf{w}} \eta^T \mathbf{s} - \tau \sum_{k=1}^D \sum_{j=1}^D \partial_k [(\nabla_{\mathbf{w}} \eta(\mathbf{w}))^T \mathcal{D}^{kj}(\mathbf{w}) \partial_j \mathbf{w}] \\ = -\tau \sum_{k=1}^D \sum_{j=1}^D (\partial_k \mathbf{w})^T (\nabla_{\mathbf{w}}^2 \eta(\mathbf{w}))^T \mathcal{D}^{kj}(\mathbf{w}) \partial_j \mathbf{w}, \quad (11) \end{aligned}$$

whose r.h.s is dissipative provided the tensor $(\nabla_{\mathbf{w}}^2 \eta)^T \mathcal{D}^{kj}$ is definite non-negative.

Remark 1. The kinetic source term (10) makes the macroscopic variable $\mathbf{w} = \mathbf{P}\mathbf{f}$ evolve according to the macroscopic source dynamics but leaves the out-of-equilibrium part $\tilde{\mathbf{f}} = \mathbf{f} - \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f})$ unchanged. Indeed, considering the differential equation

$$\partial_t \mathbf{f} = \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) \mathbf{s}(\mathbf{P}\mathbf{f}),$$

we easily show that $\mathbf{P}\mathbf{f}$ satisfies the differential equation

$$\partial_t (\mathbf{P}\mathbf{f}) = \mathbf{s}(\mathbf{P}\mathbf{f}),$$

since $\mathbf{P}\nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}$ equals the identity matrix, and then $\tilde{\mathbf{f}}$ satisfies

$$\partial_t \tilde{\mathbf{f}} = \partial_t \mathbf{f} - \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) \partial_t (\mathbf{P}\mathbf{f}) = 0.$$

This will lead to a specific time integration of the source term (see Remark 6).

Remark 2. System (3) has to be supplemented with conditions at the boundary $\partial\Omega$ of the computational domain Ω . We denote by $\mathbf{n} = (n_1 \dots n_D)$ the outward normal vector on $\partial\Omega$. For simplicity, we shall only consider very simple time-independent Dirichlet boundary conditions \mathbf{f}^b . We note

$$\mathbf{V} \cdot \mathbf{n} = \sum_{k=1}^D \mathbf{V}^k n_k, \quad \mathbf{V} \cdot \mathbf{n}^+ = \max(\mathbf{V} \cdot \mathbf{n}, 0), \quad \mathbf{V} \cdot \mathbf{n}^- = \min(\mathbf{V} \cdot \mathbf{n}, 0).$$

A natural boundary condition, which is compatible with the transport operator, is

$$\mathbf{V} \cdot \mathbf{n}^- \mathbf{f}(\mathbf{x}, t) = \mathbf{V} \cdot \mathbf{n}^- \mathbf{f}^b(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \quad (12)$$

Boundary conditions (12) are very natural from the kinetic point of view. However, they are not necessarily natural when we go back to the macroscopic hyperbolic system. For instance, at a given point of the boundary, the number of conditions depends on the lattice velocities, which have no physical meaning. It should rather depend on the number of characteristics of the macroscopic system that are entering the computational domain. Then, it is not surprising that we can observe instabilities arising from the boundary if we apply the boundary condition (12). In one of the test cases proposed in Section 6.3, we will show how we can design appropriate boundary conditions when the macroscopic model requires non-slip boundary conditions.

2.2. Examples

Devising a kinetic approximation consists in giving the discrete velocities and the projection matrix \mathbf{P} such that there exists a equilibrium function \mathbf{f}^{eq} satisfying the compatibility conditions (4)-(6). We first present a generic method, the so-called vectorial kinetic method, and then some specific Lattice-Boltzmann schemes.

2.2.1. Vectorial kinetic method

The principle of the vectorial kinetic representation is to apply an analogue decomposition to each component of the hyperbolic system [19].

We here present the simplest method belonging to this family. It consists in choosing, for each component w_l of macroscopic field $\mathbf{w} = (w_1, \dots, w_m)$, the same velocity set aligned with the Cartesian basis (\mathbf{e}_k , $k = 1, \dots, D$) and a unique velocity scale λ . For each component w_l of macroscopic field, we thus consider the $2D$ velocities

$$\mathbf{v}_{l,k,\pm} = \pm\lambda\mathbf{e}_k, \quad k = 1, \dots, D,$$

and we note $f_{l,k,\pm}$ the corresponding components of the kinetic distribution \mathbf{f} . We thus have $n_v = 2D \times m$ discrete velocities.

The consistency conditions (4) and (6) yield $m \times (D+1)$ equations for the $2D \times m$ unknowns. The projection \mathbf{P} still remains to be defined. One possible choice is to suppose that, for a given l component, each k -th velocity axis components ($f_{l,k,+}$ and $f_{l,k,-}$) contributes to the macroscopic quantity w_l in the same proportion. Hence, relations (4) and (6) write

$$\begin{aligned} f_{l,k,+}^{\text{eq}}(\mathbf{w}) + f_{l,k,-}^{\text{eq}}(\mathbf{w}) &= \frac{w_l}{D}, \quad \forall(l, k) \\ \lambda f_{l,k,+}^{\text{eq}}(\mathbf{w}) - \lambda f_{l,k,-}^{\text{eq}}(\mathbf{w}) &= q^k(\mathbf{w})_l, \end{aligned}$$

With this assumption, the equilibrium functions are uniquely defined by

$$f_{l,j,\pm}^{\text{eq}}(\mathbf{w}) = \frac{w_l}{2D} \pm \frac{q^j(\mathbf{w})_l}{2\lambda}. \quad (13)$$

For these models, the diffusion tensor (9) obtained from the Chapman-Enskog expansion takes a particular simple form. Indeed, the components of the first part of the tensor simplify into

$$(P\mathbf{V}^k\mathbf{V}^j\nabla_{\mathbf{w}}\mathbf{f}^{\text{eq}})_{l,l'} = \frac{\lambda^2}{D}\delta_{kj}\delta_{l,l'}, \quad (14)$$

so that each directional block of the diffusion tensor writes

$$\mathcal{D}^{kj} = \frac{\lambda^2}{D}\text{Id} - \nabla_{\mathbf{w}}\mathbf{q}^k\nabla_{\mathbf{w}}\mathbf{q}^j. \quad (15)$$

Then, from equation (11), considering a convex entropy $\eta(\mathbf{w})$ of the macroscopic system, the limit system is entropy dissipative provided the tensor

$$\sigma_{kj} = \nabla_{\mathbf{w}}^2\eta \left[\frac{\lambda^2}{D}\text{Id} - \nabla_{\mathbf{w}}\mathbf{q}^k\nabla_{\mathbf{w}}\mathbf{q}^j \right] \quad (16)$$

is definite non-negative.

Example 3. (One-dimensional isothermal Euler equations, vectorial method) Let apply the above framework to the one-dimensional isothermal compressible Euler equations. The conservative system is given by $m = 2$ and

$$\mathbf{w} = (\rho, \rho u)^T, \quad (17)$$

$$\mathbf{q}^1(\mathbf{w}) = \mathbf{q}(\mathbf{w}) = (\rho u, \rho u^2 + c^2\rho)^T. \quad (18)$$

where $\rho(x, t)$ is the density, $u(x, t)$ the velocity, and $c > 0$ the sound speed, which is a given parameter. The vectorial kinetic model is given by $n_v = 4$ and

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

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

The diffusion tensor reads

$$\mathcal{D}^{11} = \begin{bmatrix} \lambda^2 - (c^2 - u^2) & -2u \\ -2u(c^2 - u^2) & \lambda^2 - (c^2 + 3u^2) \end{bmatrix}. \quad (19)$$

An entropy for this system is $\eta = \rho \frac{u^2}{2} + \rho c^2 \log(\frac{\rho}{\rho_0})$, with $\rho_0 > 0$ an arbitrary constant. The entropy dissipation tensor reads

$$\nabla_{\mathbf{w}\mathbf{w}}^2 \eta \mathcal{D}^{11} = \frac{1}{\rho} \begin{bmatrix} 2u^2(c^2 - u^2) + (c^2 + u^2)(\lambda^2 - c^2 + u^2) & -u(\lambda^2 + c^2 - u^2) \\ -u(\lambda^2 + c^2 - u^2) & \lambda^2 - c^2 - u^2 \end{bmatrix}, \quad (20)$$

which is definite non-negative provided $\lambda > |u| + c$. The lattice velocity λ has to satisfy the sub-characteristic condition $\lambda > |u| + c$. We note that this representation is equivalent to the Jin and Xin relaxation [26] of the associated hyperbolic system.

2.2.2. Other Lattice Boltzmann methods

We here present the D1Q3 Lattice Boltzmann scheme and the D2Q9 scheme, its extension in two dimensions.

Example 4. (One-dimensional isothermal Euler equations, D1Q3) The D1Q3 scheme is a standard method for the one-dimensional isothermal Euler equations (17)-(18). This model takes advantage of the structure of the Euler equations, which are moments of the Boltzmann equation in the vanishing viscosity limit. The D1Q3 model uses the velocity set $\mathbf{V}^1 = (-\lambda, 0, \lambda)$ and the projection matrix

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 \\ -\lambda & 0 & \lambda \end{bmatrix}.$$

For this model, the diffusion tensor reads

$$\mathcal{D}^{11}(\mathbf{w}) = \begin{bmatrix} 0 & 0 \\ -2u(c^2 - u^2) & \lambda^2 - 3u^2 - c^2 \end{bmatrix}. \quad (21)$$

A notable fact is that there is no diffusion on the density. Let us now consider the same entropy $\eta = \rho \frac{u^2}{2} + \rho c^2 \log(\frac{\rho}{\rho_0})$ as for the vectorial scheme. The entropy dissipation tensor reads

$$\sigma_{11}(\mathbf{w}) = \nabla_{\mathbf{w}}^2 \eta(\mathbf{w}) \mathcal{D}^{11}(\mathbf{w}) = \frac{1}{\rho} \begin{bmatrix} u^2(c^2 - u^2) & -u(\lambda^2 - c^2 - 3u^2) \\ -u(c^2 - u^2) & \lambda^2 - c^2 - 3u^2 \end{bmatrix}. \quad (22)$$

Unfortunately neither \mathcal{D}^{11} nor σ_{11} can be made definite positive by setting the value of λ . Indeed, the symmetric part of σ_{11} (resp. \mathcal{D}^{11}) has always two real eigenvalues of opposite sign, regardless of the value of λ .

Example 5. (Two-dimensional isothermal Euler equations, D2Q9) The extension in two dimension of the previous D1Q3 scheme is the D2Q9 scheme. The number of conservative variables for the two-dimensional isothermal Euler scheme is $m = 3$. The conservative variables are

$$\mathbf{w} = (\rho, \rho u, \rho v)^T,$$

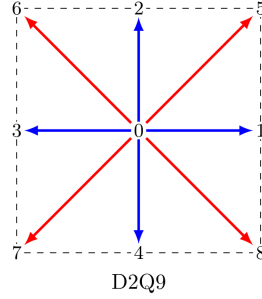


Figure 1: D2Q9 velocity grid.

and the flux is given

$$\mathbf{q}^1(\mathbf{w}) = (\rho u, \rho u^2 + c^2 \rho, \rho uv)^T,$$

$$\mathbf{q}^2(\mathbf{w}) = (\rho v, \rho uv, \rho v^2 + c^2 \rho)^T,$$

where the constant $c > 0$ is the sound speed. The number of kinetic equations is $n = 9$. The kinetic model is based on a lattice of n_v velocities $\mathbf{v}_i = (v_i^1, v_i^2)$, $i = 1, \dots, n_v$, given by

$$(v_j^k) = \lambda \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{pmatrix},$$

with $\lambda > 0$, and represented in Figure 1. The projection matrix is given by

$$\mathbf{P} = \begin{pmatrix} 1 & \cdots & 1 \\ v_1^1 & \cdots & v_9^1 \\ v_1^2 & \cdots & v_9^2 \end{pmatrix}.$$

and the equilibrium distribution is given by

$$\forall 1 \leq i \leq 9, \quad f_i^{\text{eq}}(\mathbf{w}) = \omega_i \rho \left(1 + \frac{v_i^1 u + v_i^2 v}{c^2} + \frac{(v_i^1 u + v_i^2 v)^2}{c^4} - \frac{u^2 + v^2}{2c^2} \right),$$

where the weights are given by:

$$\boldsymbol{\omega} = \left(\frac{4}{9} \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{9} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \quad \frac{1}{36} \right).$$

In the literature, a common choice is to take $\lambda = \sqrt{3}c$. In this case the kinetic model is stable only for low Mach number flows (fluid velocity small compared to the sound speed). It has nevertheless good properties (no diffusion on the density, for instance) and a requires a small number of velocities.

3. Implicit high-order palindromic time-stepping

In this section, we present a scheme for the kinetic-relaxation representation (3), which is implicit and high order in time. We rewrite equation (3) as follows

$$\partial_t \mathbf{f} + \mathbf{L}\mathbf{f} + \mathbf{N}^\tau \mathbf{f} = 0, \quad (23)$$

where $\mathbf{L}\mathbf{f} = \sum_{k=1}^D \mathbf{V}^k \partial_k \mathbf{f}$, and $\mathbf{N}^\tau \mathbf{f} = -(\mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) - \mathbf{f})/\tau$. The transport operator \mathbf{L} is linear, while the relaxation operator \mathbf{N}^τ is non-linear but local. For keeping the explanations simple, we do not include the source term in the general presentation but we will add some specific remarks.

For numerical applications, we shall consider an approximation \mathbf{f}_h of \mathbf{f} in a finite-dimensional space E_h . The parameter h is for instance the size Δx of the cells in the Discontinuous Galerkin mesh. We assume that the approximation error behaves like $O(h^p)$ with $p \geq 1$: the space approximation is at least first order accurate with respect to the discretization parameter h . The kinetic equation (3) is thus approximated by a set of differential equations

$$\partial_t \mathbf{f}_h + \mathbf{L}_h \mathbf{f}_h + \mathbf{N}_h^\tau \mathbf{f}_h = 0, \quad (24)$$

where the operator \mathbf{N}_h^τ actually equals the relaxation operator \mathbf{N}^τ , but the operator \mathbf{L}_h is an approximation of \mathbf{L} . For the approximate transport operator \mathbf{L}_h , several possibilities may be considered: finite differences, finite elements, discrete Fourier transform, Discontinuous Galerkin (DG) approximation, semi-Lagrangian methods, *etc.* In this paper, we adopt an upwind nodal DG approximation [23] (see Section 4).

3.1. First order splitting

The exact flow of the differential equation (24) is given by

$$\mathbf{f}_h(t) = \exp(-t(\mathbf{L}_h + \mathbf{N}_h^\tau)) \mathbf{f}_h(0).$$

The exponential notation can be made completely rigorous here even in the case of non-linear operators thanks to the Lie algebra formalism. For an exposition of this formalism in the context of numerical methods for ordinary differential equations, we refer for instance to [21, 29].

Computing the exact flow is generally not possible. Instead, we apply a splitting method in order to integrate the differential equation (24). We can consider the simple Lie's splitting approximation

$$\mathbf{f}_h(\Delta t) = \mathbf{M}_1(\Delta t) \mathbf{f}_h(0) + O(\Delta t^2). \quad (25)$$

with

$$\mathbf{M}_1(\Delta t) = \mathbf{R}_1^\tau(\Delta t) \mathbf{T}_1(\Delta t)$$

where \mathbf{R}_1^τ and \mathbf{T}_1 are first order approximations of the relaxation and transport exact time integrators. In order to be able to use large time steps, we consider the implicit first order Euler scheme

$$\mathbf{R}_1^\tau(\Delta t) = (\text{Id} + \Delta t \mathbf{N}_h^\tau)^{-1}, \quad \mathbf{T}_1(\Delta t) = (\text{Id} + \Delta t \mathbf{L}_h)^{-1}.$$

For a fixed $\tau > 0$, we actually have the estimates¹

$$\mathbf{R}_1(\Delta t) = \exp(-\Delta t \mathbf{N}_h^\tau) + O(\Delta t^2), \quad \mathbf{T}_1(\Delta t) = \exp(-\Delta t \mathbf{L}_h) + O(\Delta t^2).$$

Let us point out that \mathbf{R}_1^τ is a non-linear operator, because $\mathbf{f} \mapsto \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f})$ is non-linear. The linearity of \mathbf{T}_1 depends on the linearity of \mathbf{L}_h . The transport solver \mathbf{L}_h could be non-linear, even if the transport operator \mathbf{L} is linear. This is the case if slope limiters are activated, for instance.

Finally, let us note that even if \mathbf{R}_1^τ and \mathbf{T}_1 are implicit operators, they can actually be computed with an explicit cost. Indeed, since for all \mathbf{f} we have $\mathbf{P}\mathbf{N}_h^\tau \mathbf{f} = 0$, the macroscopic quantity $\mathbf{w} = \mathbf{P}\mathbf{f}$ is invariant during the relaxation step \mathbf{R}_1^τ . It is then quite standard that \mathbf{R}_1^τ takes the following explicit form

$$\mathbf{R}_1^\tau(\Delta t)\mathbf{f} = \frac{\mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}) + \frac{\tau}{\Delta t}\mathbf{f}}{1 + \frac{\tau}{\Delta t}}.$$

In addition, because the free transport step is solved by an upwind DG solver, then the linear operator $\text{Id} + \Delta t \mathbf{L}_h$ is block-triangular [5] and its inverse \mathbf{T}_1 can also be computed explicitly. We detail the method in Section 4.

Remark 6. When a source term is present in the model, we further compose \mathbf{R}_1^τ and \mathbf{T}_1 with the following local operator

$$\mathbf{G}_1(\Delta t) = (\text{Id} + \Delta t \mathbf{H})^{-1}.$$

where $\mathbf{H}\mathbf{f}_h = \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}_h) \mathbf{s}(\mathbf{P}\mathbf{f}_h)$ is the kinetic source operator. Unlike \mathbf{R}_1^τ and \mathbf{T}_1 , this operator is a priori truly non-linear. However, as noticed in Remark 1, operator \mathbf{S}_h acts only on the macroscopic variables $\mathbf{P}\mathbf{f}_h$. Consequently, we have

$$\mathbf{G}_1(\Delta t)\mathbf{f}_h = \mathbf{f}^{\text{eq}}(\mathbf{S}_1(\Delta t)\mathbf{P}\mathbf{f}_h) + \left(\mathbf{f}_h - \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}_h) \right),$$

where \mathbf{S}_1 is the implicit Euler scheme on the macroscopic variables

$$\mathbf{S}_1(\Delta t) = (\text{Id} + \Delta t \mathbf{s})^{-1}.$$

¹For one single time step the error is $O(\Delta t^2)$. But when the error is accumulated on $t_{\text{max}}/\Delta t$ time steps it indeed produces a first order method.

3.2. Second-order stiffly accurate splitting

Using methods of geometric integration [25], we now consider a second-order in time scheme, that keeps second-order accuracy in the limit $\tau \rightarrow 0$.

We consider the second-order Crank-Nicolson scheme for the transport equation

$$\mathbf{T}_2(\Delta t) = (\text{Id} + \frac{\Delta t}{2} \mathbf{L}_h)(\text{Id} - \frac{\Delta t}{2} \mathbf{L}_h)^{-1}. \quad (26)$$

as well as for the relaxation operator

$$\mathbf{R}_2^\tau(\Delta t) = (\text{Id} + \frac{\Delta t}{2} \mathbf{N}_h^\tau)(\text{Id} - \frac{\Delta t}{2} \mathbf{N}_h^\tau)^{-1}.$$

These transport and relaxation operators can be solved with the cost of an explicit scheme. Indeed, since the macroscopic variables $\mathbf{w} = \mathbf{P}\mathbf{f}$ is unchanged during the relaxation step, the relaxation operator (like \mathbf{R}_1^τ) is only apparently implicit. We actually have the explicit formula:

$$\mathbf{R}_2^\tau(\Delta t)\mathbf{f}_h = \frac{(2\tau - \Delta t)\mathbf{f}_h}{2\tau + \Delta t} + \frac{2\Delta t \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}_h)}{2\tau + \Delta t}. \quad (27)$$

As regards the transport step \mathbf{T}_2 , it involves an explicit and an implicit transport both over a time interval $\Delta t/2$. Like in the first order splitting, the implicit transport solution can be computed at the cost of an explicit solver (see Section 4).

If $\tau > 0$, we observe that the operators \mathbf{T}_2 and \mathbf{R}_2^τ are *time-symmetric*: if we set $\mathbf{O}_2 = \mathbf{T}_2$, $\mathbf{O}_2 = \mathbf{R}_2^\tau$, or $\mathbf{O}_2 = \mathbf{S}_2$ a first order (or more) approximation, then \mathbf{O}_2 satisfies

$$\mathbf{O}_2(-\Delta t) = \mathbf{O}_2(\Delta t)^{-1}, \quad \mathbf{O}_2(0) = \text{Id}. \quad (28)$$

This property implies that, since \mathbf{O}_2 is necessarily a second order approximation of the exact integrator [29, 21]. Let us now note that when $\tau = 0$, the relaxation operator becomes independent of the time step and writes

$$\mathbf{R}_2^0(\Delta t)\mathbf{f}_h = 2\mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}_h) - \mathbf{f}_h, \quad (29)$$

and then \mathbf{R}_2^0 does not satisfy (28) anymore. However, we note that, due to the conservation of the macroscopic variables, it is an involution

$$\mathbf{R}_2^0(\Delta t)\mathbf{R}_2^0(\Delta t) = \text{Id}. \quad (30)$$

This is the key point of the following scheme.

We propose to use the following time-symmetric splitting

$$\mathbf{M}_2(\Delta t) = \mathbf{T}_2\left(\frac{\Delta t}{4}\right)\mathbf{R}_2^\tau\left(\frac{\Delta t}{2}\right)\mathbf{T}_2\left(\frac{\Delta t}{2}\right)\mathbf{R}_2^\tau\left(\frac{\Delta t}{2}\right)\mathbf{T}_2\left(\frac{\Delta t}{4}\right). \quad (31)$$

It can be easily checked that $\mathbf{M}_2(\Delta t)$ is time-symmetric for all $\tau \geq 0$, including the case $\tau = 0$. Consequently, the scheme remains second order accurate in the limit $\tau \rightarrow 0$.

Remark 7. The classical second-order Strang splitting,

$$\tilde{\mathbf{M}}_2(\Delta t) = \mathbf{T}_2 \left(\frac{\Delta t}{2} \right) \mathbf{R}_2^\tau(\Delta t) \mathbf{T}_2 \left(\frac{\Delta t}{2} \right),$$

is time-symmetric for all $\tau > 0$ but not for $\tau = 0$. However for $\tau = 0$, unless the method does not give the identity operator on the kinetic distribution for $\Delta t = 0$, it turns out to be the identity operator on the macroscopic variables: $\mathbf{P}\tilde{\mathbf{M}}_2(0)\mathbf{f} = \mathbf{P}\mathbf{f}$. This might explain why second-order accuracy can be numerically observed at $\tau = 0$ for the macroscopic variables, even though the operator on the full kinetic system is not symmetric.

Remark 8. To take into account source terms, we consider the following second-order scheme

$$\mathbf{G}_2(\Delta t)\mathbf{f}_h = \mathbf{f}^{\text{eq}}(\mathbf{S}_2(\Delta t)\mathbf{P}\mathbf{f}_h) + \left(\mathbf{f}_h - \mathbf{f}^{\text{eq}}(\mathbf{P}\mathbf{f}_h) \right),$$

where \mathbf{S}_2 is the (truly) implicit Crank-Nicolson scheme on the macroscopic variables

$$\mathbf{S}_2(\Delta t) = (\text{Id} + \Delta t \mathbf{s})(\text{Id} + \Delta t \mathbf{s})^{-1}.$$

Then the second-order splitting is modified into

$$\begin{aligned} \bar{\mathbf{M}}_2(\Delta t) = \\ \mathbf{T}_2 \left(\frac{\Delta t}{4} \right) \mathbf{G}_2 \left(\frac{\Delta t}{2} \right) \mathbf{R}_2^\tau \left(\frac{\Delta t}{2} \right) \mathbf{T}_2 \left(\frac{\Delta t}{2} \right) \mathbf{R}_2^\tau \left(\frac{\Delta t}{2} \right) \mathbf{G}_2 \left(\frac{\Delta t}{2} \right) \mathbf{T}_2 \left(\frac{\Delta t}{4} \right). \end{aligned}$$

which is still time-symmetric.

3.3. High-order palindromic splitting

Once defined a second-order accurate time-symmetric scheme, palindromic composition method enables to easily achieve any even order of accuracy [29, 21, 14]. A general palindromic scheme with $s + 1$ steps has the form

$$\mathbf{M}_p(\Delta t) = \mathbf{M}_2(\gamma_0 \Delta t) \mathbf{M}_2(\gamma_1 \Delta t) \cdots \mathbf{M}_2(\gamma_s \Delta t), \quad (32)$$

where the γ_i 's are real numbers satisfying

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

In the following, we will consider the fourth-order Suzuki scheme [41, 21, 29] and the sixth-order Kahan-Li scheme [28], whose intermediate steps are given in Table 1. The Suzuki scheme requires 5 steps, while the Kahan-Li scheme is made of 9 steps.

We note that the two methods require to apply the elementary relaxation or transport \mathbf{R}_2^τ and \mathbf{T}_2 with negative time steps $-\Delta t < 0$. If we were using the exact transport solver \mathbf{L} , negative time steps would not cause any problem. However, the transport approximation \mathbf{L}_h generally introduces a slight

Suzuki coefficients (p=4, s=4)	
$\gamma_0 = \gamma_1 = \gamma_3 = \gamma_4 =$	$\frac{1}{4-4^{1/3}}, \quad \gamma_2 = -\frac{4^{1/3}}{4-4^{1/3}}.$
Kahan-Li coefficients (p=6, s=8)	
$\gamma_0 = \gamma_8 =$	0.392161444007314139275655330038...
$\gamma_1 = \gamma_7 =$	0.332599136789359438604272125325...
$\gamma_2 = \gamma_6 =$	-0.7062461725576393598098453372227...
$\gamma_3 = \gamma_5 =$	0.0822135962935508002304427053341...
$\gamma_4 =$	0.798543990934829963398950353048...

Table 1: Palindromic coefficients.

dissipation to ensure stability (for instance upwinding in DG discretization, see Section 4). In order to ensure stability, we have thus to replace $\mathbf{T}_2(-\Delta t)$ with a more stable operator. This can be done by observing that solving $\partial_t \mathbf{f} + \mathbf{L}\mathbf{f} = 0$ for negative time $t < 0$ is equivalent to solve $\partial_{t'} \mathbf{f} - \mathbf{L}\mathbf{f} = 0$ for $t' = -t > 0$. Therefore, we use

$$\mathbf{T}'_2(\Delta t) = (\text{Id} + \frac{\Delta t}{2}(-\mathbf{L})_h)(\text{Id} - \frac{\Delta t}{2}(-\mathbf{L})_h)^{-1}$$

where $(-\mathbf{L})_h$ is a stable discretization of $-\mathbf{L}$. The numerical relaxation operator \mathbf{R}_τ^2 is time reversible in the limit $\tau \rightarrow 0$: for $\tau = 0$, it actually does not depend on Δt anymore (see (29)). In this stage, negative time steps do not cause any difficulty, at least when $\tau \ll \Delta t$.

4. Implicit discontinuous Galerkin method for linear transport

In this section, we briefly present the approximate linear transport operator \mathbf{L}_h obtained from the Discontinuous Galerkin method. We also show how its matrix-triangular structure enables to solve, with an explicit cost, the implicit operator involved in the second order Crank-Nicolson solver (see (26)).

4.1. DG approximation

As said above, for solving (3) we treat the transport operator $\mathbf{V} \cdot \partial$ and the collision operator \mathbf{N} separately, thanks to the splitting approach. Let us now describe the transport solver.

For a simple exposition, we only consider one single scalar transport equation for $f(\mathbf{x}, t) \in \mathbb{R}$ at constant velocity \mathbf{v}

$$\partial_t f + \mathbf{v} \cdot \nabla f = 0. \tag{33}$$

The general vectorial case is easily deduced.

We consider a mesh \mathcal{M} of Ω made of open sets, called ‘‘cells’’, $\mathcal{M} = \{L_i, i = 1 \dots N_c\}$. In the most general setting, the cells satisfy

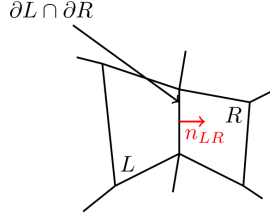


Figure 2: Convention for the L and R cells orientation.

1. $L_i \cap L_j = \emptyset$, if $i \neq j$;
2. $\overline{\cup_i L_i} = \overline{\Omega}$.

In each cell $L \in \mathcal{M}$, we consider a basis of functions $(\varphi_{L,i}(\mathbf{x}))_{i=0 \dots N_d-1}$ constructed from polynomials of order d . We denote by h the maximal diameter of the cells. With an abuse of notation we still denote by f the approximation of f , defined by

$$f(\mathbf{x}, t) = \sum_{j=0}^{N_d-1} f_{L,j}(t) \varphi_{L,j}(\mathbf{x}), \quad \mathbf{x} \in L.$$

The DG formulation then reads: find the $f_{L,j}$'s such that for all cell L and all test function $\varphi_{L,i}$

$$\int_L \partial_t f \varphi_{L,i} - \int_L f \mathbf{v} \cdot \nabla \varphi_{L,i} + \int_{\partial L} (\mathbf{v} \cdot \mathbf{n}^+ f_L + \mathbf{v} \cdot \mathbf{n}^- f_R) \varphi_{L,i} = 0. \quad (34)$$

In this formula (see Figure 43):

- R denotes the neighbouring cell to L along its boundary $\partial L \cap \partial R$, or the exterior of Ω on $\partial L \cap \partial \Omega$.
- $\mathbf{n} = \mathbf{n}_{LR}$ is the unit normal vector on ∂L oriented from L to R .
- f_R denotes the value of f in the neighbouring cell R on $\partial L \cap \partial R$.
- If L is a boundary cell, one may have to use the boundary values instead: $f_R = f^b$ on $\partial L \cap \partial \Omega$.
- $\mathbf{v} \cdot \mathbf{n}^+ f_L + \mathbf{v} \cdot \mathbf{n}^- f_R$ is the standard upwind numerical flux encountered most finite volume or DG methods.

In our applications, we consider hexahedral cells. We have a reference cell

$$\hat{L} =]-1, 1[^D$$

and a smooth transformation $\mathbf{x} = \tau_L(\hat{\mathbf{x}})$, $\hat{\mathbf{x}} \in \hat{L}$, that maps \hat{L} on L

$$\tau_L(\hat{L}) = L.$$

We assume that $\boldsymbol{\tau}_L$ is invertible and we denote by $\boldsymbol{\tau}'_L$ its (invertible) Jacobian matrix. We also assume that $\boldsymbol{\tau}_L$ is a direct transformation

$$\det \boldsymbol{\tau}'_L > 0.$$

In our implementation $\boldsymbol{\tau}_L$ is a quadratic map based on hexahedral curved ‘‘H20’’ finite elements with 20 nodes. The mesh of H20 finite elements is generated by `gmsh` [17].

On the reference cell, we consider the Gauss-Lobatto (GL) points $(\hat{\mathbf{x}}_i)_{i=0\dots N_d-1}$, $N_d = (d+1)^D$ and associated weights $(\omega_i)_{i=0\dots N_d-1}$. They are obtained by tensor products of the $(d+1)$ one-dimensional Gauss-Lobatto (GL) points on $] -1, 1[$. The reference GL points and weights are then mapped to the physical GL points of cell L by

$$\mathbf{x}_{L,i} = \boldsymbol{\tau}_L(\hat{\mathbf{x}}_i), \quad \omega_{L,i} = \omega_i \det \boldsymbol{\tau}'_L(\hat{\mathbf{x}}_i) > 0. \quad (35)$$

In addition, the six faces of the reference hexahedral cell are denoted by F_ϵ , $\epsilon = 1 \dots 6$ and the corresponding outward normal vectors are denoted by $\hat{\mathbf{n}}_\epsilon$. A big advantage of choosing the GL points is that the volume and the faces share the same quadrature points. A special attention is necessary for defining the face quadrature weights. If a GL point $\hat{\mathbf{x}}_i \in F_\epsilon$, we denote by μ_i^ϵ the corresponding quadrature weight on face F_ϵ . We also use the convention that $\mu_i^\epsilon = 0$ if $\hat{\mathbf{x}}_i$ does not belong to face F_ϵ . A given GL point $\hat{\mathbf{x}}_i$ can belong to several faces when it is on an edge or in a corner of \hat{L} . Because of symmetry, we observe that if $\mu_i^\epsilon \neq 0$, then the weight μ_i^ϵ does not depend on ϵ .

We then consider basis functions $\hat{\varphi}_i$ on the reference cell: they are the Lagrange polynomials associated to the Gauss-Lobatto point and thus satisfy the interpolation property

$$\hat{\varphi}_i(\hat{\mathbf{x}}_j) = \delta_{ij}.$$

The basis functions on cell L are then defined according to the formula

$$\varphi_{L,i}(\mathbf{x}) = \hat{\varphi}_i(\boldsymbol{\tau}_L^{-1}(\mathbf{x})).$$

In this way, they also satisfy the interpolation property

$$\varphi_{L,i}(\mathbf{x}_{L,j}) = \delta_{ij}. \quad (36)$$

In this paper, we only consider conformal meshes: the GL points on cell L are supposed to match the GL points of cell R on their common face.

Let L and R be two neighbouring cells. Let $\mathbf{x}_{L,j}$ be a GL point in cell L that is also on the common face between L and R . In the case of conformal meshes, it is possible to define the index j' such that

$$\mathbf{x}_{L,j} = \mathbf{x}_{R,j'}.$$

Applying a numerical integration to (34), using (35) and the interpolation property (36), we finally obtain

$$\begin{aligned} \partial_t f_{L,i} \omega_{L,i} - \sum_{j=0}^{N_d-1} \mathbf{v} \cdot \nabla \varphi_{L,i}(\mathbf{x}_{L,j}) f_{L,j} \omega_{L,j} + \\ \sum_{\epsilon=1}^6 \mu_i^\epsilon (\mathbf{v} \cdot \mathbf{n}_\epsilon(\mathbf{x}_{L,i})^+ f_{L,i} + \mathbf{v} \cdot \mathbf{n}_\epsilon(\mathbf{x}_{L,i})^- f_{R,i'}) = 0. \end{aligned} \quad (37)$$

We have to detail how the gradients and normal vectors are computed in the above formula. Let \mathbf{A} be a square matrix. We recall that the cofactor matrix of \mathbf{A} is defined by

$$\text{co}(\mathbf{A}) = \det(\mathbf{A}) (\mathbf{A}^{-1})^T. \quad (38)$$

The gradient of the basis function is computed from the gradients on the reference cell using (38)

$$\nabla \varphi_{L,i}(\mathbf{x}_{L,j}) = \frac{1}{\det \boldsymbol{\tau}'_L(\hat{\mathbf{x}}_i)} \text{co}(\boldsymbol{\tau}'_L(\hat{\mathbf{x}}_j)) \hat{\nabla} \hat{\varphi}_i(\hat{\mathbf{x}}_j).$$

In the same way, the scaled normal vectors \mathbf{n}_ϵ on the faces are computed by the formula

$$\mathbf{n}_\epsilon(\mathbf{x}_{L,i}) = \text{co}(\boldsymbol{\tau}'_L(\hat{\mathbf{x}}_i)) \hat{\mathbf{n}}_\epsilon.$$

We introduce the following notation for the cofactor matrix

$$\mathbf{c}_{L,i} = \text{co}(\boldsymbol{\tau}'_L(\hat{\mathbf{x}}_i)).$$

The nodal DG scheme then reads

$$\begin{aligned} \partial_t f_{L,i} - \frac{1}{\omega_{L,i}} \sum_{j=0}^{N_d-1} \mathbf{v} \cdot \mathbf{c}_{L,j} \hat{\nabla} \hat{\varphi}_i(\hat{\mathbf{x}}_j) f_{L,j} \omega_j + \\ \frac{1}{\omega_{L,i}} \sum_{\epsilon=1}^6 \mu_i^\epsilon (\mathbf{v} \cdot \mathbf{c}_{L,i} \hat{\mathbf{n}}_\epsilon^+ f_{L,i} + \mathbf{v} \cdot \mathbf{c}_{L,i} \hat{\mathbf{n}}_\epsilon^- f_{R,i'}) = 0. \end{aligned} \quad (39)$$

On boundary GL points, the value of $f_{R,i'}$ is given by the boundary condition

$$f_{R,i'} = f^b(\mathbf{x}_{L,i}), \quad \mathbf{x}_{L,i} = \mathbf{x}_{R,i'}.$$

For practical reasons, it is interesting to also consider $f_{R,i'}$ as an artificial unknown in the fictitious cell. The fictitious unknown is then a solution of the differential equation

$$\partial_t f_{R,i'} = 0. \quad (40)$$

In the end, if we put all the unknowns in a single vector $\mathbf{F}(t)$, (39), (40) read as a large system of coupled differential equations

$$\partial_t \mathbf{F}_h = \mathbf{L}_h \mathbf{F}_h. \quad (41)$$

This defines \mathbf{L}_h the transport matrix. The transport matrix satisfies the following properties:

- $\mathbf{L}_h \mathbf{F}_h = 0$ if the components of \mathbf{F} are all the same.
- Let \mathbf{F}_h be such that the components corresponding to the boundary terms vanish. Then for the scalar product $\langle \mathbf{F}, \mathbf{G} \rangle = \sum_L \sum_i \omega_{L,i} f_{L,i} g_{L,i}$, we have

$$\langle \mathbf{F}_h, \mathbf{L}_h \mathbf{F}_h \rangle \leq 0. \quad (42)$$

This dissipation property is a consequence of the choice of an upwind numerical flux [27]².

- In many cases, and with a good numbering of the unknowns in \mathbf{F}_h , \mathbf{L}_h has a block-triangular structure. This aspect is discussed in Subsection 4.2.

As stated above, we actually have to apply a transport solver for each constant velocity \mathbf{v}_i .

Let L be a cell of the mesh \mathcal{M} and \mathbf{x}_i a GL point in L . As in the scalar case, we denote by $\mathbf{f}_{L,i}$ the approximation of \mathbf{f} in L at GL point i . In the sequel, with an abuse of notation and according to the context, we may continue to note $\mathbf{F}(t)$ the big vector made of all the vectorial values $\mathbf{f}_{L,j}$ at all the GL points j in all the (real or fictitious) cells L .

We may also continue to denote by \mathbf{L}_h the matrix made of the assembly of all the transport operators for all velocities \mathbf{v}_i . With a good numbering of the unknowns it is possible in many cases to suppose that \mathbf{L}_h is block-triangular. More precisely, because in the transport step the equations are uncoupled, we see that \mathbf{L}_h can be made block-diagonal, each diagonal block being itself block-triangular. See next Section 4.2.

4.2. Triangular structure of the transport matrix

Because of the upwind structure of the numerical flux, it appears that the transport matrix is often block-triangular. This is very interesting because this allows to applying implicit schemes to (41) without the costly inversion of linear systems [32]. We can provide the formal structure of \mathbf{L}_h through the construction of a directed graph \mathcal{G} with a set of vertices \mathcal{V} and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The vertices of the graph are associated to the (real or fictitious) cells of \mathcal{M} . Now consider two cells L and R with a common face F_{LR} . We denote by \mathbf{n}_{LR} the normal vector on F_{LR} oriented from L to R . If there is at least one GL point \mathbf{x} on F_{LR} such that

$$\mathbf{n}_{LR}(\mathbf{x}) \cdot \mathbf{v} > 0,$$

then the edge from L to R belongs to the graph:

$$(L, R) \in \mathcal{E},$$

see Figure 3.

²Actually, this dissipation property is true only when the geometrical transformations τ_L are affine maps. For quadratic maps, the Gauss-Lobatto numerical integration is not exact anymore (“aliasing” effect: see [23] for instance). Weak instabilities may develop for long-time numerical simulations.

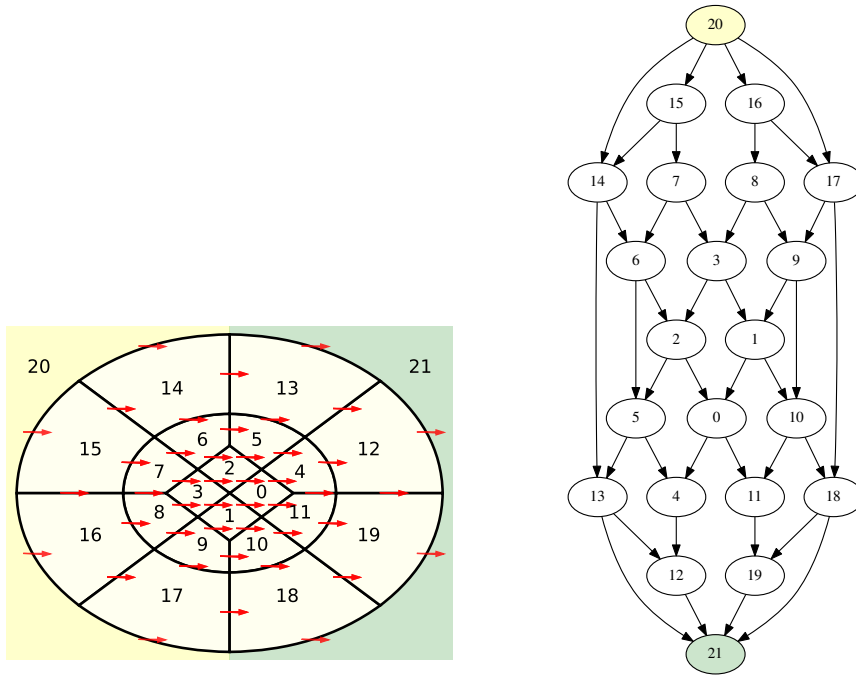


Figure 3: Construction of the dependency graph. Left: example of a simple unstructured mesh with 20 interior cells. The velocity field v is indicated by red arrows. We add two fictitious cells: one for the upwind boundary condition (cell 20) and one for the outflow part of $\partial\Omega$ (cell 21). Right: the corresponding dependency graph \mathcal{G} . By examining the dependency graph, we observe that the values of \mathbf{F}^{n+1} in cell 15 and 16 have to be computed first, using the boundary conditions. Then cells [14, 7, 8, 17] can be computed in parallel, then cells [6, 3, 9] can be computed in parallel, and so on until the downwind boundary is reached.

In (39) we can distinguish between several kinds of terms. We write

$$\partial_t f_L + \Gamma_{L \leftarrow L} f_L + \sum_{(R,L) \in \mathcal{E}} \Gamma_{L \leftarrow R} f_R,$$

with

$$\Gamma_{L \leftarrow L} f_L = -\frac{1}{\omega_{L,i}} \sum_{j=0}^{N_d-1} \mathbf{v} \cdot \mathbf{c}_{L,j} \hat{\nabla} \hat{\varphi}_i(\hat{\mathbf{x}}_j) f_{L,j} \omega_j + \frac{1}{\omega_{L,i}} \sum_{\epsilon=1}^6 \mu_i^\epsilon \mathbf{v} \cdot \mathbf{c}_{L,i} \hat{\mathbf{n}}_\epsilon^+ f_{L,i},$$

and, if $(R, L) \in \mathcal{E}$,

$$\Gamma_{L \leftarrow R} f_R = \frac{1}{\omega_{L,i}} \mu_i^\epsilon \mathbf{v} \cdot \mathbf{c}_{L,i} \hat{\mathbf{n}}_\epsilon^- f_{R,i'}.$$

We can use the following convention

$$(R, L) \notin \mathcal{E} \Rightarrow \Gamma_{L \leftarrow R} = 0. \quad (43)$$

$\Gamma_{L \leftarrow L}$ contains the terms that couple the values of f inside the cell L . They correspond to diagonal blocks of size $(d+1)^D \times (d+1)^D$ in the transport matrix \mathbf{L}_h . $\Gamma_{L \leftarrow R}$ contains the terms that couple the values inside cell L with the values in the neighboring upwind cell R . If R is a downwind cell relatively to L then $\mu_i^\epsilon \mathbf{v} \cdot \mathbf{c}_{L,i} \hat{\mathbf{n}}_\epsilon^- = 0$ and $\Gamma_{L \leftarrow R} = 0$ is indeed compatible with the above convention (43).

Once the graph \mathcal{G} is constructed, we can analyze it with standard tools. If it contains no cycle, then it is called a Directed Acyclic Graph (DAG). Any DAG admits a topological ordering of its nodes. A topological ordering is a numbering of the cells $i \mapsto L_i$ such that if there is a path from L_i to L_j in \mathcal{G} then $j > i$. In practice, it is useful to remove the fictitious cells from the topological ordering. In our implementation they are put at the end of the list.

Once the new ordering of the graph vertices is constructed, we can construct a numbering of the components of \mathbf{F} by first numbering the unknowns in L_0 then the unknowns in L_1 , *etc.* More precisely, we set

$$F_{kN_d+i} = f_{L_k,i}.$$

Then, with this ordering, the matrix \mathbf{L}_h is lower block-triangular with diagonal blocks of size $(d+1)^D \times (d+1)^D$. It means that we can apply implicit schemes to (41) without costly inversion of large linear systems.

As stated above, we actually have to apply a transport solver for each constant velocity \mathbf{v}_i . In the sequel, with another abuse of notation and according to the context, we continue to note \mathbf{F} the big vector made of all the vectorial values $\mathbf{f}_{L,j}$ at all the GL points j in all the (real or fictitious) cells L .

We may also continue to denote by \mathbf{L}_h the matrix made of the assembly of all the transport operators for all velocities \mathbf{v}_i . With a good numbering of the unknown it is still possible to suppose that \mathbf{L}_h is block-triangular. More precisely, as in the transport step the equations are uncoupled, we see that \mathbf{L}_h can be made a block-diagonal matrix, each diagonal block being itself block-triangular.

5. Parallel implementation

Thanks to the splitting procedure described in Section 3 the whole algorithm exhibits several levels of parallelism. First, it is clear that the collision step is purely local to each interpolation point and thus embarrassingly parallel. Second, the transport equations are completely uncoupled from the other ones. They can thus be solved independently in parallel. Finally, as stated above (see Figure 3), inside the resolution of each transport equation it is again possible to detect additional parallelism from the examination of the dependency graph.

We have written a C implementation of the Palindromic Discontinuous Galerkin (PDG) method using a data-based formulation of the parallelism. In this formulation it is essential to distinguish between the input (Read mode) and output (Write mode) data of each elementary computational task. The tasks are then submitted to a runtime system that is able to distribute the work on the available processors. From the data dependency, the runtime system detects the tasks that can be performed in parallel. In our implementation, we rely on the StarPU runtime library, which is especially designed for efficient scientific computing [4]. We use the MPI version of StarPU in order to distribute the computations on clusters of multicore computers.

Submitting a task to the StarPU system induces a slight overhead. It is thus important to submit tasks that are not too small (too much time would be spent into the tasks management) or not too big (which could block the tasks flow). Therefore, we apply what we call a “macrocell” approach. The geometry is first meshed at a coarse level. We call the cells of the coarse mesh the “macrocells”. The macrocells are then refined into several subcells. We apply the task-based transport solver described in Figure 3 at the macrocell level instead of the subcell level. In this way, we can adjust the grain of the parallelism. This approach necessitates solving local transport equations into the macrocells. This is achieved by assembling and solving local block-triangular linear system. Those local sparse linear systems are solved with the KLU library, which is able to detect efficiently block-triangular structures [15]. More details on the implementation are given in [5].

For the moment, the local systems are assembled and factorized at each time-step. It would probably be more efficient to store the local LU decompositions for saving computational time. We have not yet compared the efficiency of our approach with other explicit or implicit DG solvers. However, we have observed a good parallel scaling of the method when the number of computational cores increases [5]. In addition, as it is shown in the numerical sections, the PDG method accepts very high CFL numbers, which makes it a good candidate for avoiding costly non-linear implicit solvers.

6. Numerical results

In this section, we apply the methodology presented in the previous sections. We first numerically demonstrate the accuracy of the scheme on one-dimensional

test cases. We then show how the method applies to two-dimensional models. We will make some remarks on the treatment of the boundary conditions.

An important feature of the PDG method is the possibility to consider large time steps without oscillations. In order to measure this advantage, we have to define precisely how we define the time step and the corresponding CFL number.

6.1. One-dimensional isothermal Euler test cases

In this section, we consider the vectorial kinetic method apply to the one-dimensional isothermal Euler system, presented in Example 3.

6.1.1. Smooth solution

For the first validation of the method we consider a test case with a smooth solution, in the fluid limit $\tau = 0$. The initial condition is given by

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

The sound speed is set to $c = 0.6$ and the lattice velocity to $\lambda = 2$. We define the CFL number $\beta = \lambda\Delta t/\delta$, where δ is the minimal distance between two Gauss-Lobatto points in the mesh. First, the CFL number is fixed to $\beta = 5$. We consider a sufficiently large computational domain $[a, b] = [-2, 2]$ and a sufficiently short final time $t_{\max} = 0.4$ so that the boundary conditions play no role. The reference solution $\mathbf{f}(\cdot, t_{\max})$ is computed numerically with a very fine mesh. In the DG solver the polynomial order in x is fixed to $d = 5$.

On Figure 4 (left picture) we give the results of the convergence study for the smooth solution. The considered error is the L^2 norm of $\mathbf{f}_h(\cdot, t_{\max}) - \mathbf{f}(\cdot, t_{\max})$. We make the same experiment with $\beta = 50$. The convergence study for the Suzuki and Kahan-Li schemes is also presented on Figure 4 (right picture). At high CFL, not only the scheme remains stable, but the high accuracy is also preserved.

6.1.2. Behaviour for discontinuous solutions

We have also experimented the scheme for discontinuous solutions. Of course, in this case the effective order of the method cannot be higher than one and we expect Gibbs oscillations near the discontinuities. On the interval $[a, b] = [-1, 1]$, we consider a Riemann problem with the following initial condition

$$\rho(x, 0) = \begin{cases} 2 & \text{if } x < 0, \\ 1 & \text{otherwise.} \end{cases}, \quad u(x, 0) = 0.$$

We consider numerical results in the fluid limit $\tau = 0$. On Figure 5 we compare the sixth-order numerical solution with the exact one at $t = t_{\max} = 0.4$ for a CFL number $\beta = 3$ and $N_x = 100$ cells. We observe oscillations in the shock wave and at the boundaries of the rarefaction wave, as expected. However, we also observe that the high order scheme is able to capture a precise rarefaction

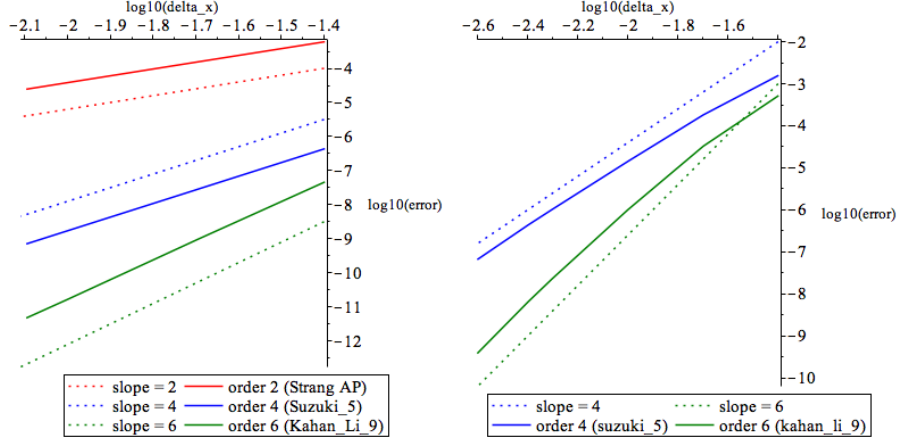


Figure 4: Convergence study for several palindromic methods, order 2 (red), 4 (blue) and 6 (green). The dotted lines are reference lines with slopes 2, 4 and 6 respectively. Left: CFL number $\beta = 5$. Right: CFL number $\beta = 50$.

wave and the correct position of the shock wave. This is a little bit surprising, because in presence of shock waves, the Euler model is no more reversible and we solve it with a palindromic time integrator method that has a reversible structure. The only dissipation is provided by the upwind DG solver (see (42)). Apparently, this slight dissipation is sufficient here for stabilizing the numerical method.

6.2. MHD flow

We now consider a two-dimensional MagnetoHydroDynamics MHD model. The $m = 6$ unknowns of the model are the density $\rho(\mathbf{x}, t) \in \mathbb{R}$, the two-dimensional velocity vector $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^2$, the two-dimensional magnetic field $\mathbf{B}(\mathbf{x}, t) \in \mathbb{R}^2$ and the total energy $Q(\mathbf{x}, t) \in \mathbb{R}$. The pressure $p(\mathbf{x}, t) \in \mathbb{R}$ is given by a perfect gas pressure law

$$p = (\gamma - 1) \left(Q - \rho \frac{\mathbf{u} \cdot \mathbf{u}}{2} - \frac{\mathbf{B} \cdot \mathbf{B}}{2} \right), \quad \gamma = 5/3.$$

The conservative variables are

$$\mathbf{w} = (\rho, \rho \mathbf{u}^T, Q, \mathbf{B}^T)^T.$$

For a two-dimensional direction vector $\mathbf{n} = (n_1, n_2)^T$, the MHD flux is then given by

$$\mathbf{q}(\mathbf{w})\mathbf{n} = \begin{pmatrix} \rho \mathbf{u} \cdot \mathbf{n} \\ \rho(\mathbf{u} \cdot \mathbf{n})\mathbf{u} + (p + \frac{\mathbf{B} \cdot \mathbf{B}}{2})\mathbf{n} - (\mathbf{B} \cdot \mathbf{n})\mathbf{B} \\ (Q + p + \frac{\mathbf{B} \cdot \mathbf{B}}{2})\mathbf{u} \cdot \mathbf{n} - (\mathbf{B} \cdot \mathbf{u})(\mathbf{B} \cdot \mathbf{n}) \\ (\mathbf{u} \cdot \mathbf{n})\mathbf{B} - (\mathbf{B} \cdot \mathbf{n})\mathbf{u} \end{pmatrix}.$$

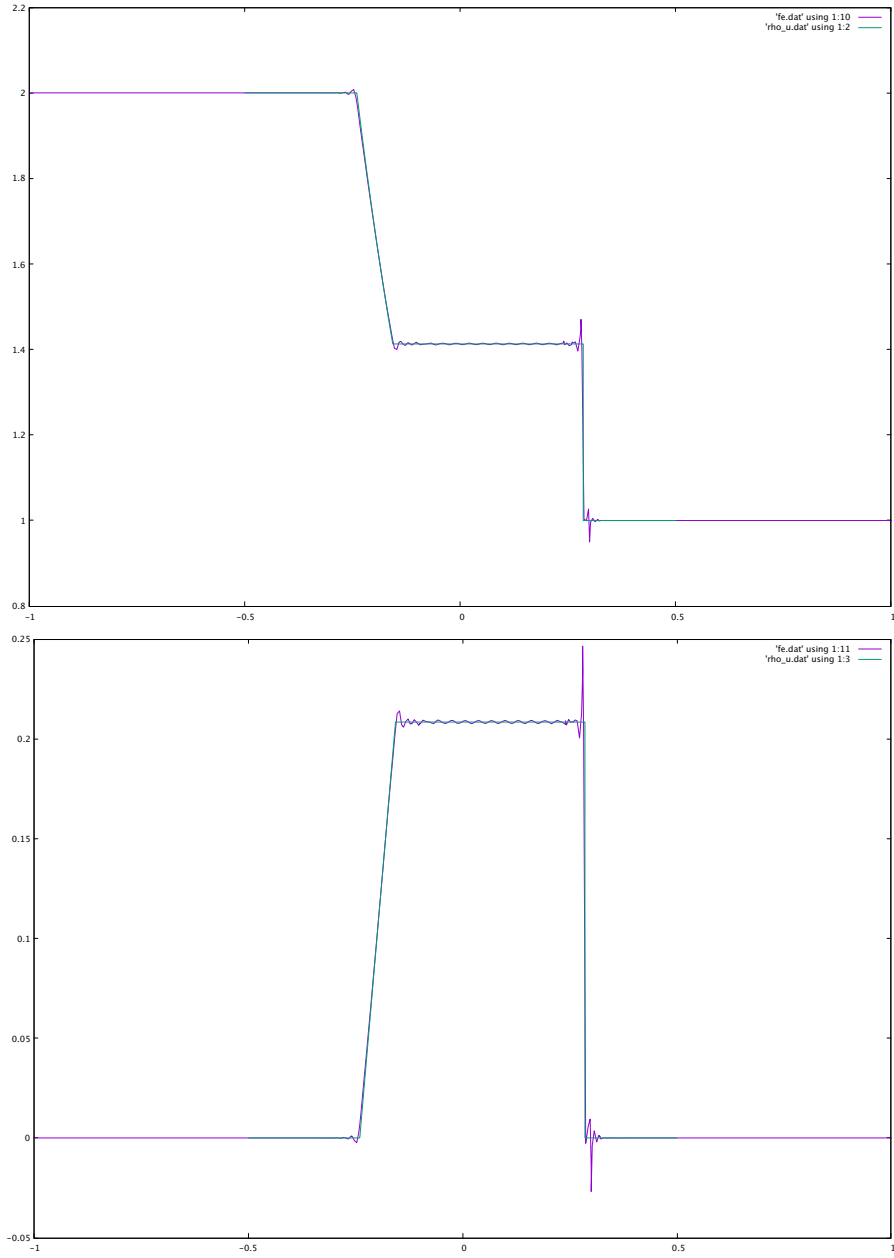


Figure 5: Riemann problem with $\tau = 0$. Comparison of the exact solution (green curve), and the numerical sixth-order solution (purple curve). Left: density. Right: velocity.

We approximate this hyperbolic system with the vectorial kinetic representation described in Section 2.2.1.

The test-case is built upon a single vortex, which is a stationary solution of the MHD system, to which a constant drift velocity is added. In the moving frame centered on $\mathbf{r}_O(t) = t\mathbf{u}_{\text{drift}}$, with $\mathbf{u}_{\text{drift}} \in \mathbb{R}^2$, the analytical solution reads in polar coordinates

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

with $b_0 = \rho_0 u_0^2$. The results shown below are obtained with the parameter set $\rho_0 = p_0 = 1$, $u_0 = b_0 = 0.2$, $\mathbf{u}_{\text{drift}} = (1, 1)^T$, $h(r) = \exp[(1 - r^2)/2]$. The macromesh is the disk made from 20 macrocells. Each macrocell is refined into $8 \times 8 = 64$ subcells with fifth order basis functions, leading to 2304 quadrature points per macrocell. The minimal distance between two quadrature points is around $h_{\min} \approx 0.02$. To each of the six scalar fields $(\rho, \rho u_x, \rho u_y, Q, B_x B_y)$, we associate a four-velocity $D2Q4$ model, with velocities $(-\lambda, 0), (\lambda, 0), (0, -\lambda), (0, \lambda)$, so that there are 24 kinetic fields. The velocity scale λ is set to 4. The vortex is initially centered in $\mathbf{r}_O = (0, 0)^T$ at $t = 0$ and we perform the simulation up to $t = 1$. We test convergence of the first, second and fourth order splitting schemes with time-steps ranging from $\Delta t = 0.2$ to $\Delta t = 0.0125$. This leads to kinetic CFL numbers $(\lambda \Delta t / h_{\min})$ for the transport subsets ranging from 40 to 5 for the first order splitting scheme, 20 to 2.5 for the second order splitting scheme, and 13 to 1.6 for the fourth order Suzuki scheme.

6.3. Flow past a cylinder (“thick” boundary condition)

We here consider the two-dimensional isothermal Euler equation and its D2Q9 approximation presented in Example 5 (see Section 2.2.2).

In this test case, we consider the flow of a fluid in a rectangular duct with a cylindrical solid obstacle, as presented in Figure 7. The simulation domain is the rectangle $[-3, 21] \times [-6, 6]$. The cylindrical obstacle has radius $r = 0.4$. At the boundary of the obstacle, no-slip boundary conditions are applied.

The initial condition is given by the constant state

$$\rho(\mathbf{x}, 0) = 1, \quad u(\mathbf{x}, 0) = u_0 = 0.05, \quad v(\mathbf{x}, 0) = 0.$$

The boundary condition applied at the duct left entry ($x = -3$ axis) for the whole simulation is

$$\rho(\mathbf{x}, t) = 1, \quad u(\mathbf{x}, t) = \tanh(t/5)u_0, \quad v(\mathbf{x}, t) = 0.$$

The progressive growth of the flow at the inlet mitigates the initial unphysical transitory regime during which the initially uniform flow adapts to the no-slip

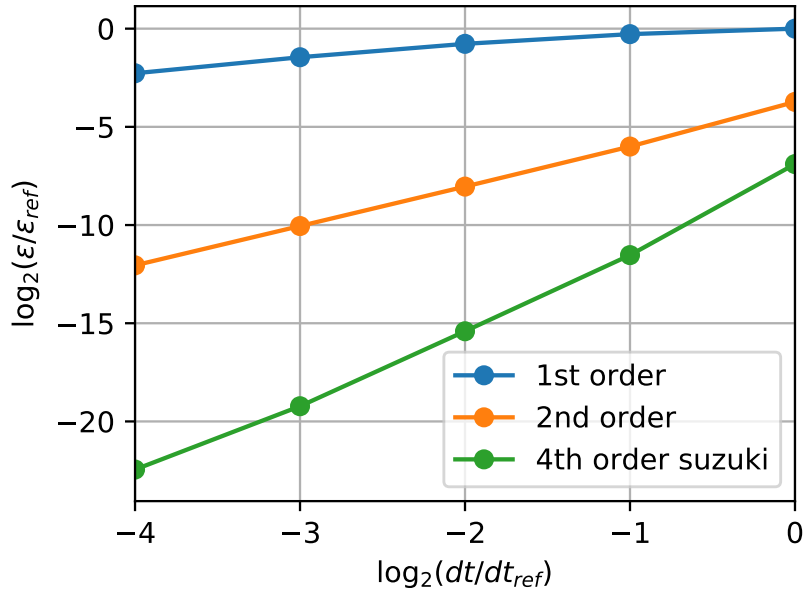


Figure 6: Drifting MHD vortex test-case. Convergence of L^2 error with respect to the analytical solution at $t = 1$. The reference error ϵ_{ref} of the log scale is the error of the first order scheme for $\Delta t = 0.2$.

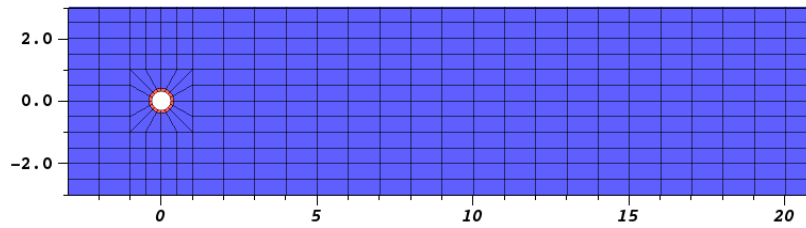


Figure 7: Flow around a cylindrical obstacle. Coarse macromesh with 328 macrocells. The no-slip condition is applied using a stiff relaxation in the thin annulus marked in red.

condition at the obstacle boundary. A second consequence is the transition during the simulation between an initial symmetric recirculation regime (with two vortices in the wake of the obstacle) to the formation of von Karman streets [20].

6.3.1. No-slip boundary condition

To take into account the no-slip boundary condition around the obstacle, we use a fictitious domain approach [31, 2].

Our way to apply the fictitious domain approach is to first mesh a thin annular shell (of width 0.1), at the boundary of the obstacle. The computational domain is thus enlarged with a small part of the obstacle. The boundary condition is then applied by considering a stiff penalization source term in the fluid equations

$$\mathbf{s} = -\kappa \begin{pmatrix} 0 \\ \rho \mathbf{u} \end{pmatrix},$$

with $\kappa = 0$ inside the fluid and $\kappa \gg 1$ in the solid. This amounts to considering the solid as a porous media with a very small porosity.

On the kinetic side, this source term can be represented in many different ways. Our choice is to take

$$\mathbf{g} = -\kappa (0, f_1 - f_3, f_2 - f_4, f_3 - f_1, f_4 - f_2, f_5 - f_7, f_6 - f_8, f_7 - f_5, f_8 - f_6).$$

In other words, each component of the kinetic distribution associated to a given lattice velocity relaxes toward the component associated with the opposite velocity (see Figure 1).

In practice, we observe a very fast decay of the velocity in the obstacle as expected. In addition, this procedure is much more stable than a Dirichlet type boundary condition (12). See the numerical results presented below.

6.3.2. Numerical simulations

The relaxation time has a finite but small value $\tau = 0.0002$. Accounting for the fact that for this model the dimensionless sound speed is $c = 1/\sqrt{3}$, the Mach number of the unperturbed flow is approximately $\frac{u_0}{c} \approx 0.087$. The simulation was performed on a macromesh with 328 macrocells; each macrocell contains 36×36 integration points. The minimal distance between two integration points is about $h_{min} \approx 0.005$. The simulation was run with a time step $\Delta t = 0.1$, up to $t = 3680$, about 7.5 times the macroscopic transit time $L/u_0 = 480$. For the $D2Q9$ kinetic model used herein, the maximal velocity modulus is $\lambda_{max} = \sqrt{2}$. In the second order splitting scheme the transport substep has $\Delta t = 0.05$ at most, so that the maximal kinetic transport CFL number is about 14. On Figure 8, we show the streamlines and velocity field norm at key points of the dynamics: at $t = 50$ when the flow is still essentially symmetric, at $t = 120$ after the onset of the von Karman oscillations in the wake of the obstacle, and at $t = 340$ in the periodic oscillatory regime which starts at about $t = 220$.

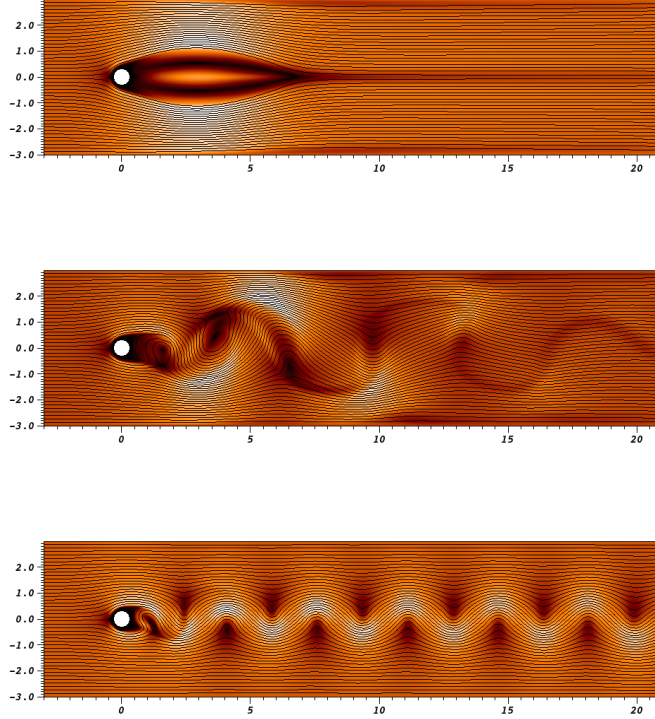


Figure 8: Flow around a cylindrical obstacle. Velocity norm $|u|$ and streamlines at $t = 50$, $t = 120$, $t = 340$.

We observe that the geometry of the flow around the obstacle is well preserved. The velocity is virtually null in the thick boundary (Fig. 9). While the density exhibits a small oscillation inside the thick boundary, its value at the boundary of the computational domain is very close to the nominal value.

6.4. Two-dimensional and three-dimensional two-fluid flow

6.4.1. Two-fluid flow with gravity

We finally apply the methodology to a model of liquid-gas flow with gravity. The model has been studied by several authors, see [1, 9, 18]. In dimension 2 (resp. 3), the $m = 4$ (resp. $m = 5$) unknowns of the model are the mixture density $\rho(\mathbf{x}, t) \in \mathbb{R}$, the two-dimensional (resp. three-dimensional) velocity vector $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^2$ (resp. \mathbb{R}^3) and the mass fraction of gas $\varphi(\mathbf{x}, t) \in \mathbb{R}$. The pressure of the mixture $p(\mathbf{x}, t) \in \mathbb{R}$ is computed by

$$p = p(\rho, \varphi) = \alpha p_1 + (1 - \alpha) p_2,$$

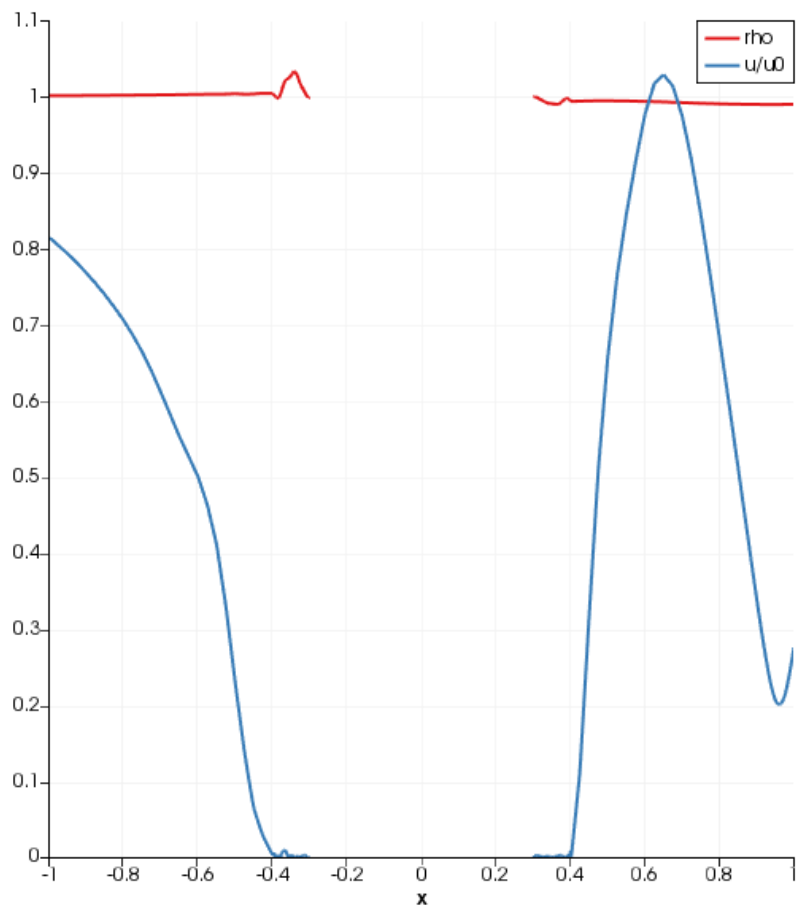


Figure 9: Flow around a cylindrical obstacle. Plot of $|u|/u_0$ and ρ at $t = 340$ on the x axis around the obstacle.

where $\alpha(\mathbf{x}, t) \in [0, 1]$ is the volume fraction of gas and $p_1(\mathbf{x}, t), p_2(\mathbf{x}, t) \in \mathbb{R}$ are the pressures of the gas and the liquid, respectively. For the pressure of each fluid, we take

$$p_1 = p_0 + c^2 \left(\frac{\rho\varphi}{\alpha} - \rho_{0,1} \right), \quad p_2 = p_0 + c^2 \left(\frac{\rho(1-\varphi)}{(1-\alpha)} - \rho_{0,2} \right).$$

The physical constants of the models are the reference pressure p_0 , two reference densities for each fluid $\rho_{0,1}$ and $\rho_{0,2}$ and the sound speed c . Here the sound speed has no physical meaning. It is chosen large enough in order that the flow can be considered as almost incompressible.

The volume fraction α is chosen in such a way that

$$p_1 = p_2.$$

The conservative variables are

$$\mathbf{w} = (\rho, \rho\mathbf{u}^T, \rho\varphi)^T.$$

The flux is given by

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

and the source term

$$\mathbf{s} = \begin{pmatrix} 0 \\ \rho\mathbf{g} \\ 0 \end{pmatrix}.$$

where $\mathbf{g} \in \mathbb{R}^2$ (resp. \mathbb{R}^3) is the gravity vector in dimension 2 (resp. 3). In the following, we consider the vectorial kinetic approximation as described in Section 2.2.1. In dimension 2, unlike the *D2Q9* scheme, each of the four macroscopic fields is the sum of four kinetic fields associated with the velocities $(-\lambda, 0), (\lambda, 0), (0, -\lambda)$ and $(0, \lambda)$, leading to a total of 16 kinetic fields. Similarly, the three-dimensional kinetic relaxation model includes 30 kinetic fields.

The macroscopic source term is represented by the kinetic source $\mathbf{g} = \nabla_{\mathbf{w}}\mathbf{f}^{\text{eqs}}$ of equation (10).

In both test cases, we consider the growth of the Rayleigh-Taylor instability: the light and heavy phases are initially well separated, the heavy phase lying above the light one in the gravity field. At the interface between the two phases, the mass fraction value drops from 1 to 0 over the thin interface width. In order to avoid Gibbs oscillations due to the large gradients, the relaxation time is set to small (around 10^{-5}) but finite values. The resulting numerical viscosity smooths out the flow.

6.4.2. *Two-dimensional Rayleigh-Taylor instability in an annulus.*

For the 2D model, we consider an annular domain of interior radius $r_{min} = 0.2$ and exterior radius $r_{max} = 1$. The gravity field is a radial one pointing inwards, i.e $\mathbf{g} = -g_0\mathbf{e}_r$ with $g_0 = 0.05$. The model parameters are

$$c = 1, \quad p_0 = 1, \quad \rho_{0,1} = 0.9, \quad \rho_{0,2} = 1.1.$$

In the initial unperturbed state, the heavy fluid lies “above” the light one in the annulus $r_0 \leq r \leq 1$, with $r_0 = 0.6$. The interface is perturbed with a single azimuthal Fourier mode i.e.

$$r_{pert} = r_0 + a \sin(m\theta). \quad (44)$$

We will show here the results for a single $m = 5$ azimuthal mode with $a = 0.01$. The sharpness of the transition from the light to the heavy fluid is set using a hyperbolic tangent radial profile, so that the mass fraction at $t = 0$ reads

$$\varphi(r, \theta, t = 0) = 0.5 (1 - \tanh[(r - r_{pert}(\theta))/w_{pert}]). \quad (45)$$

with $w_{pert} = 0.02$.

The initial density is set so that each of the pure phases is at mechanical equilibrium with the gravity field. The macromesh is an annulus, discretized on a regular polar grid with $n_r = 5$ and $n_\theta = 32$. Each macrocell is refined in 25×10 subcells with second order basis functions. With those parameters, the minimal distance between two interpolation points is $h_{min} = 0.002$. The velocity scale parameter is set to $\lambda = 2.5$. The time-step is set to $\Delta t = 0.01$. In the second order palindromic splitting time scheme used here, the maximal time substep for the transport substep is $0.5\Delta t = 0.005$ so that the maximal *CFL* number for the transport of the kinetic fields is 6.25.

The evolution in time of the mass fraction φ (see (Fig. 10 and 11)), the growth of the Rayleigh-Taylor mushrooms is clearly visible. The dispersive errors entail a slight excursion (a few percent) from the pure phase nominal values outside of the interface zone. Those oscillations are mitigated by the diffusive dissipation induced by the finite value of the relaxation time $\tau = 0.00001$. We do not concern ourselves here with the fine tweaking of parameters or model required to strike a particular balance between the conservation of the interface sharpness and the control of dispersive errors.

Assuming a given overall accuracy, the question arises whether the scarcity of the velocity set of the kinetic model induces geometrical artifacts. For this particular test case the mesh, the *D2Q4* velocity set and the continuous system (perturbation included) are all symmetric with respect to the y axis. The fifth-order rotational symmetry of the $m = 5$ mode, is not preserved by the discrete velocity set or the mesh though, and we can expect numerical anisotropy effects to appear. On Figure 12, we compare radial profiles of the mass fractions along the axes of the five Rayleigh-Taylor mushrooms. The symmetry with respect to the vertical axis common to both the excited mode and the velocity set is well

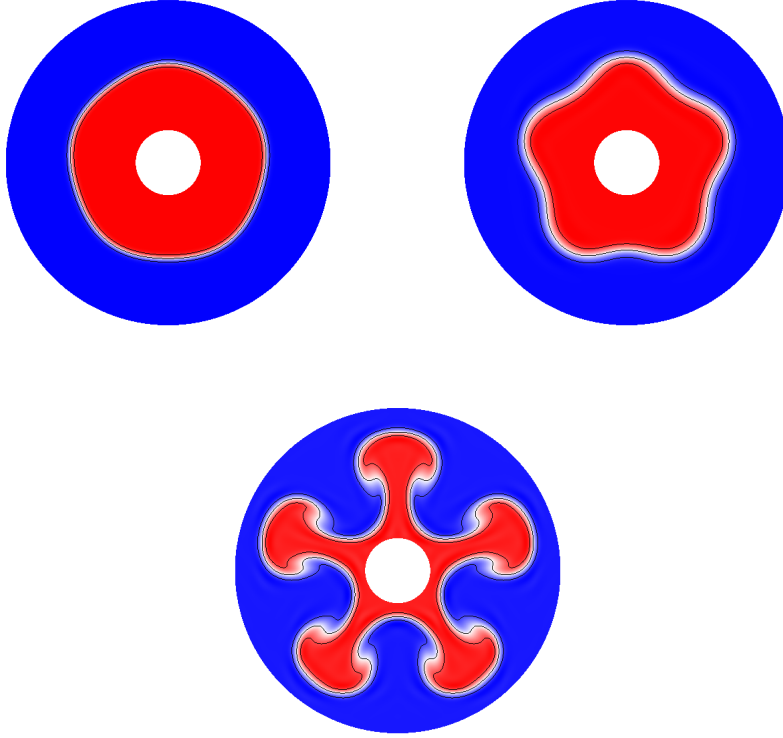


Figure 10: . Two-dimensional Rayleigh-Taylor instability with a single $m = 5$ azimuthal mode perturbation. Mass fraction $\varphi(\mathbf{x}, t)$ at $t = [0, 20, 40]$. Black lines are iso-value contours at values $\varphi = 0.1, 0.5, 0.9$.

preserved by the scheme (the corresponding plots are indistinguishable), while the $2\pi/5$ rotational symmetry is slightly broken due to the anisotropy of the error. We conclude that the anisotropy of the kinetic velocity set has only a very small effect.

6.4.3. Three-dimensional Rayleigh-Taylor instability in a cylinder.

We consider here the three-dimensional version of the two-fluids mixture in a cylindrical duct of height $H = 4.8$ and radius $R = 1$ whose axis is aligned with the constant gravity field $\mathbf{g} = -g_0\mathbf{e}_z$ with $g_0 = 0.04$. The coarse macromesh (Fig. 13) is composed of 1152 macrocells, each of which is refined in $4 \times 4 \times 8 = 128$ subcells and second order basis functions. The buffer zones (yellow, green and red on Fig. 13) are used to apply boundary conditions on the internal cylindrical volume, with a volumic relaxation operator. For this test case, the boundary conditions at $z = \pm 2.4$ are simply obtained by imposing the stationary equilibrium state for each of the fluids, and no operator is required in the relevant

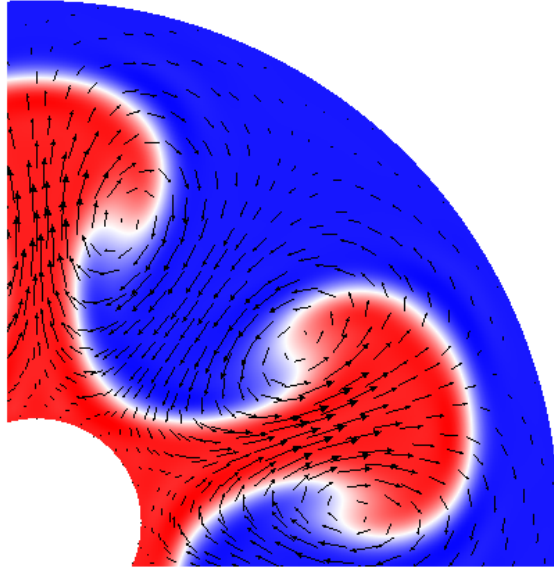


Figure 11: Two-dimensional Rayleigh-Taylor instability with a single $m = 5$ azimuthal mode perturbation. Mass fraction $\varphi(\mathbf{x}, t)$ and velocity field at $t = 40$.

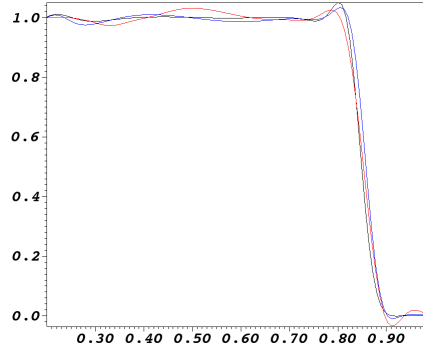


Figure 12: Two-dimensional Rayleigh-Taylor instability with a single $m = 5$ azimuthal mode perturbation. Radial profiles of mass fraction $\varphi(r, \theta, t)$ at time $t = 40$ and azimuthal angles $\theta = \pi/2$ (continuous black line), $\theta = \pi/2 \pm 2\pi/5$ (continuous and dotted blue lines), $\theta = \pi/2 \pm 4\pi/5$ (continuous and dotted red lines). We observe small differences between the radial profiles, because the kinetic velocity set is not aligned with the mesh, nor with the mode.

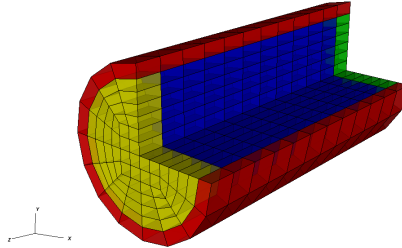


Figure 13: Three-dimensional Rayleigh-Taylor instability. Cylindrical macromesh with 1152 macrocells. The physical domain is the internal cylinder (blue internal zone and red and green buffers). The annular buffer (red) is used to match the non-physical imposed boundary condition on its exterior boundary at $r = 0.2$ with the physical boundary at $r = 1$ using a volumic stiff relaxation operator.

buffers (yellow and green on Fig. 13). In order to mimic the effect of a solid duct at $r = 1$, a no-slip condition is applied on the horizontal directions and a slip condition in the vertical direction.

The initial interface between the two fluids is the plane $h_{ref} = 0.5$. It is perturbed with a single bump centered in $r = 0$ so that the altitude h_{pert} of the interface reads

$$h_{pert}(r) = h_{ref} - a \cos(2\pi r) \exp(-(r/d)^2) \quad (46)$$

with $h_{ref} = 0.5$, $a = 0.2$, $d = 0.3$. The transition between the light and heavy fluid is smoothed out using a hyperbolic tangent profile of typical width $w_{pert} = 0.05$, so that the mass fraction is given in cylindrical coordinates by

$$\varphi(r, z, t = 0) = 0.5 \left(1 - \tanh\left(\frac{z - h_{pert}(r)}{w_{pert}}\right) \right). \quad (47)$$

On Fig. 14 and 15, we observe the development of the Rayleigh-Taylor mushroom.

7. Conclusion

We have presented a new general implicit scheme, the Palindromic Discontinuous Galerkin (PDG) scheme, for solving systems of conservation laws. Despite being formally implicit, the new scheme does not require costly linear solver and has the complexity of an explicit scheme. We have also proposed a new palindromic splitting algorithm that allows us to achieve high accuracy in time, even in the stiff limit. The whole approach remains stable and accurate, even at high CFL numbers.

We have validated the properties of the method on several one-dimensional test cases. We have also tested the approach in higher dimensions, and on different models of conservation laws coming from physics.

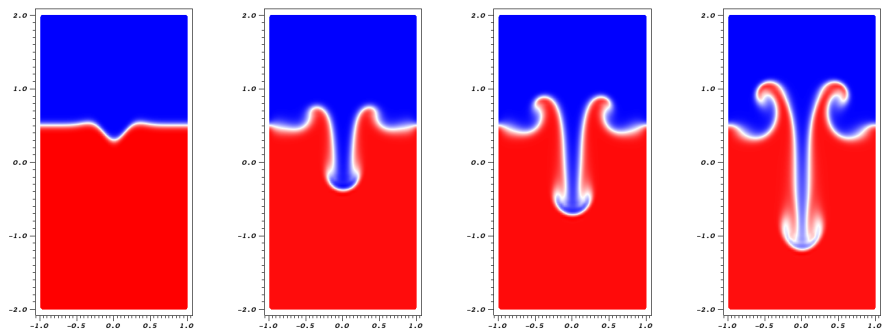


Figure 14: Three-dimensional Rayleigh-Taylor instability. Planar cut at $y = 0$ of the mass fraction ϕ at $t = 0, t = 4, t = 20, t = 25$.

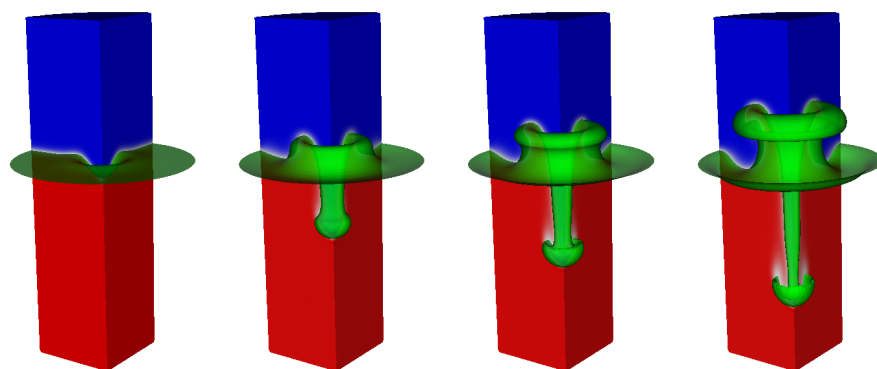


Figure 15: Three-dimensional Rayleigh-Taylor instability. Pseudo-color slice and $\varphi = 0.5$ iso-contour (green) of the mass fraction φ at times $t = 0, t = 4, t = 20, t = 25$.

These first results are very promising for the future.

Many practical and theoretical works are still needed in order to harness the full potential of the PDG method. The most important question is to construct a methodology for handling general boundary conditions in a stable way. A promising approach is to test and analyze in detail the fictitious domain approach that we have sketched in this paper. Another interesting direction of research would be to replace the discontinuous Galerkin resolution of the transport equation by an alternative method, such as semi-Lagrangian approaches. Finally, in many applications, it is important to handle conservation laws with small second order dissipative terms. Those dissipative terms can be of a physical nature or serve a numerical purpose for avoiding oscillations in shock waves, for instance. This can be achieved by considering small, but non-vanishing, relaxation parameter $\tau > 0$. In this direction also, many useful practical extensions of the method can be tested and analyzed.

8. Appendix

8.1. Second order approximation

For the sake of completeness, we recall the proof of (8).

Proof. We decompose \mathbf{f} into its equilibrium and non-equilibrium part setting $\mathbf{f} = \mathbf{f}^{\text{eq}} + \tilde{\mathbf{f}}$. Substituting this formulation in the initial kinetic system, and applying P we get the equivalent coupled system

$$\begin{cases} \partial_t \mathbf{w} + \sum_{k=1}^D \partial_k P \mathbf{V}^k \mathbf{f}^{\text{eq}} &= -\partial_k P \mathbf{V}^k \tilde{\mathbf{f}} + P \mathbf{g} \\ \partial_t \tilde{\mathbf{f}} + \sum_{k=1}^D \partial_k \mathbf{V}^k \tilde{\mathbf{f}} &= -\tau^{-1} \tilde{\mathbf{f}} - \left[\partial_t \mathbf{f}^{\text{eq}} + \sum_{k=1}^D \partial_k \mathbf{V}^k \mathbf{f}^{\text{eq}} \right] + \mathbf{g} \end{cases} \quad (48)$$

We now perform a formal expansion in τ of all quantities, with $\tilde{\mathbf{f}}^{(0)} = 0$: for instance, the kinetic source term is expanded as $\mathbf{g} = \mathbf{g}^{(0)} + \tau \mathbf{g}^{(1)} + \dots$.

At the lowest order we have the limit system

$$\begin{cases} \partial_t \mathbf{w}^{(0)} + \sum_{k=1}^D \partial_k P \mathbf{V}^k \mathbf{f}^{\text{eq},(0)} &= P \mathbf{g}^{(0)} \\ \tilde{\mathbf{f}}^{(0)} &= 0 \end{cases} \quad (49)$$

At the first order we have

$$\begin{cases} \partial_t \mathbf{w}^{(1)} + \sum_{k=1}^D \partial_k P \mathbf{V}^k \mathbf{f}^{\text{eq},(1)} &= -\sum_{k=1}^D \partial_k P \mathbf{V}^k \tilde{\mathbf{f}}^{(1)} + P \mathbf{g}^{(1)} \\ 0 &= -\tilde{\mathbf{f}}^{(1)} - \left[\partial_t \mathbf{f}^{\text{eq},(0)} + \sum_{k=1}^D \partial_k \mathbf{V}^k \mathbf{f}^{\text{eq},(0)} \right] + \mathbf{g}^{(0)} \end{cases} ,$$

and the second equation yields

$$\tilde{\mathbf{f}}^{(1)} = -\left[\nabla_{\mathbf{w}} \mathbf{f}^{\text{eq},(0)} \partial_t \mathbf{w}^{(0)} + \sum_{k=1}^D \mathbf{V}^k \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq},(0)} \partial_k \mathbf{w}^{(0)} \right] + \mathbf{g}^{(0)}. \quad (50)$$

Substituting the expression for $\partial_t \mathbf{w}^{(0)}$ obtained at the lowest order we obtain

$$\tilde{\mathbf{f}}^{(1)} = \sum_{k=1}^D [\nabla_{\mathbf{w}} \mathbf{f}^{\text{eq},(0)} P \mathbf{V}^k \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq},(0)} - \mathbf{V}^k \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq},(0)}] \partial_k \mathbf{w}^{(0)} + [\mathbf{g}^{(0)} - \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq},(0)} P \mathbf{g}^{(0)}]. \quad (51)$$

Recombining terms up to first order in τ , and using the consistency condition $P \mathbf{V}^k \mathbf{f}^{\text{eq}} = \mathbf{q}^k$, the kinetic relaxation system is consistent with

$$\partial_t \mathbf{w} + \sum_{k=1}^D \partial_k \mathbf{q}^k(\mathbf{w}) = \mathbf{s} + \tau \sum_{k=1}^D \sum_{j=1}^D \partial_k [\mathcal{D}^{kj} \partial_j \mathbf{w}] + \tau \sum_{k=1}^D \partial_k P \mathbf{V}^k [\nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}} \mathbf{s} - \mathbf{g}] + \mathcal{O}(\tau^2), \quad (52)$$

with the diffusion tensor given by

$$\mathcal{D}^{kj} = P \mathbf{V}^k \mathbf{V}^j \nabla_{\mathbf{w}} \mathbf{f}^{\text{eq}} - \nabla_{\mathbf{w}} \mathbf{q}^k \nabla_{\mathbf{w}} \mathbf{q}^j. \quad (53)$$

□

Bibliography

- [1] Grégoire Allaire, Sébastien Clerc, and Samuel Kokh. A five-equation model for the simulation of interfaces between compressible fluids. *Journal of Computational Physics*, 181(2):577–616, 2002.
- [2] Philippe Angot, Thomas Auphan, and Olivier Guès. An optimal penalty method for a hyperbolic system modeling the edge plasma transport in a tokamak. *Journal of Computational Physics*, 261:1–22, 2014.
- [3] 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.
- [4] 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 Larson Träff and Jack Dongarra, editors, *EuroMPI 2012*, volume 7490 of *LNCS*. Springer, September 2012. Poster Session.
- [5] Jayesh Badwaik, Matthieu Boileau, David Coulette, Emmanuel Franck, Philippe Helluy, Laura Mendoza, and Herbert Oberlin. Task-based parallelization of an implicit kinetic scheme. *arXiv preprint arXiv:1702.00169*, 2017.

- [6] Jürgen Bey and Gabriel Wittum. Downwind numbering: Robust multigrid for convection-diffusion problems. *Applied Numerical Mathematics*, 23(1):177–192, 1997.
- [7] François Bouchut. A reduced stability condition for nonlinear relaxation to conservation laws. *Journal of Hyperbolic Differential Equations*, 01(01):149–170, 2004.
- [8] Yann Brenier. Averaged multivalued solutions for scalar conservation laws. *SIAM journal on numerical analysis*, 21(6):1013–1037, 1984.
- [9] Guilhem Chantepedrix, Philippe Villedieu, and Jean-Paul Vila. A compressible model for separated two-phase flows computations. In *ASME Fluids Engineering Division Summer Meeting*, number 31141. ASME Montreal, Canada, 2002.
- [10] Gui Qiang Chen, C David Levermore, and Tai Ping Liu. Hyperbolic conservation laws with stiff relaxation terms and entropy. *Communications on Pure and Applied Mathematics*, 47(6):787–830, 1994.
- [11] Shiyi Chen and Gary D Doolen. Lattice Boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364, 1998.
- [12] Frédéric Coquel, Q-L Nguyen, Marie Postel, and Q-H Tran. Large time step positivity-preserving method for multiphase flows. In *Hyperbolic Problems: Theory, Numerics, Applications*, pages 849–856. Springer, 2008.
- [13] F Coron and B Perthame. Numerical passage from kinetic to fluid equations. *SIAM Journal on Numerical Analysis*, 28(1):26–42, 1991.
- [14] David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger, and Laurent Navoret. Palindromic discontinuous galerkin method for kinetic equations with stiff relaxation. *arXiv preprint arXiv:1612.09422*, 2016.
- [15] Timothy A Davis and Ekanathan Palamadai Natarajan. Algorithm 907: KLU, a direct sparse solver for circuit simulation problems. *ACM Transactions on Mathematical Software (TOMS)*, 37(3):36, 2010.
- [16] Paul J Dellar. An interpretation and derivation of the lattice Boltzmann method using Strang splitting. *Computers & Mathematics with Applications*, 65(2):129–141, 2013.
- [17] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [18] Frédéric Golay and Philippe Helluy. Numerical schemes for low mach wave breaking. *International Journal of Computational Fluid Dynamics*, 21(2):69–86, 2007.

- [19] Benjamin Graille. Approximation of mono-dimensional hyperbolic systems: A lattice Boltzmann scheme as a relaxation method. *Journal of Computational Physics*, 266:74–88, 2014.
- [20] A. Grucelski and J. Pozorski. Lattice boltzmann simulations of flow past a circular cylinder and in simple porous media. *Computers & Fluids*, 71:406–416, 2013.
- [21] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.
- [22] Xiaoyi He and Li-Shi Luo. Lattice Boltzmann model for the incompressible Navier–Stokes equation. *Journal of statistical Physics*, 88(3-4):927–944, 1997.
- [23] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.
- [24] Shi Jin. Runge-Kutta methods for hyperbolic conservation laws with stiff relaxation terms. *Journal of Computational Physics*, 122(1):51–67, 1995.
- [25] Shi Jin. Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations. *SIAM Journal on Scientific Computing*, 21(2):441–454, 1999.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] Robert I McLachlan and G Reinout W Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.
- [30] Renwei Mei and Wei Shyy. On the finite difference-based lattice Boltzmann method in curvilinear coordinates. *Journal of Computational Physics*, 143(2):426–448, 1998.
- [31] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

- [32] Salli Moustafa, Mathieu Faverge, Laurent Plagne, and Pierre Ramet. 3D cartesian transport sweep for massively parallel architectures with PAR-SEC. In *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pages 581–590. IEEE, 2015.
- [33] Francesca Nannelli and Sauro Succi. The lattice Boltzmann equation on irregular lattices. *Journal of Statistical Physics*, 68(3-4):401–407, 1992.
- [34] Roberto Natalini. A discrete kinetic approximation of entropy solutions to multidimensional scalar conservation laws. *Journal of Differential Equations*, 148(2):292 – 317, 1998.
- [35] 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.
- [36] 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.
- [37] Gongwen Peng, Haowen Xi, Comer Duncan, and So-Hsiang Chou. Lattice Boltzmann method on irregular meshes. *Physical Review E*, 58(4):R4124, 1998.
- [38] Benoît Perthame. Boltzmann type schemes for gas dynamics and the entropy property. *SIAM Journal on Numerical Analysis*, 27(6):1405–1421, 1990.
- [39] YH Qian, Dominique d’Humières, and Pierre Lallemand. Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.
- [40] Xing Shi, Jianzhong Lin, and Zhaosheng Yu. Discontinuous Galerkin spectral element lattice Boltzmann method on triangular element. *International Journal for Numerical Methods in Fluids*, 42(11):1249–1261, 2003.
- [41] 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.
- [42] Feng Wang and Jinchao Xu. A crosswind block iterative method for convection-dominated problems. *SIAM Journal on Scientific Computing*, 21(2):620–645, 1999.