

A high-order relaxation method with projective integration for solving nonlinear systems of hyperbolic conservation laws

Pauline Lafitte ^{*} Ward Melis [†] Giovanni Samaey [‡]

March 28, 2017

Abstract

We present a general, high-order, fully explicit relaxation scheme which can be applied to any system of nonlinear hyperbolic conservation laws in multiple dimensions. The scheme consists of two steps. In a first (relaxation) step, the nonlinear hyperbolic conservation law is approximated by a kinetic equation with stiff BGK source term. Then, this kinetic equation is integrated in time using a projective integration method. After taking a few small (inner) steps with a simple, explicit method (such as direct forward Euler) to damp out the stiff components of the solution, the time derivative is estimated and used in an (outer) Runge-Kutta method of arbitrary order. We show that, with an appropriate choice of inner step size, the time step restriction on the outer time step is similar to the CFL condition for the hyperbolic conservation law. Moreover, the number of inner time steps is also independent of the stiffness of the BGK source term. We discuss stability and consistency, and illustrate with numerical results (linear advection, Burgers' equation and the shallow water and Euler equations) in one and two spatial dimensions.

1 Introduction

Hyperbolic conservation laws arise in numerous physical applications, such as fluid dynamics, plasma physics, traffic modeling and electromagnetism (see, for instance, [26,37]). They express the conservation of physical quantities (such as mass, momentum, or energy) and may be supplemented with boundary conditions that control influx or outflux at the boundaries of the physical domain [26]. In this paper, we consider a system of hyperbolic conservation laws in multiple spatial dimensions:

$$\partial_t \mathbf{u} + \nabla_{\mathbf{x}} \cdot \mathbf{F}(\mathbf{u}) = \mathbf{0}, \quad (1)$$

or, equivalently,

$$\partial_t \mathbf{u} + \sum_{d=1}^D \partial_{x^d} \mathbf{F}^d(\mathbf{u}) = \mathbf{0}, \quad (2)$$

in which $\mathbf{x} = (x^d)_{d=1}^D \in \mathbb{R}^D$ represents the space variables (D being the number of spatial dimensions), $\mathbf{u}(\mathbf{x}, t) := (u_m(\mathbf{x}, t))_{m=1}^M \in \mathbb{R}^M$ denotes the conserved quantities, and $\mathbf{F}(\mathbf{u}) \in \mathbb{R}^{M \times D}$ corresponds to the flux functions.

Hyperbolic conservation laws are often solved using a finite volume method [26,29], which is derived from the integral expression of the conservation law. To that end, in a scalar, one-dimensional setting and with a spatially uniform grid, the domain is divided in I cells $\mathcal{C}_i = [x_{i-1/2}, x_{i+1/2}]$ with constant cell width Δx over which the cell average of the solution $u(x, t)$ to the conservation law

$$\partial_t u + \partial_x F(u) = 0, \quad (3)$$

^{*}Laboratoire de Mathématiques Appliquées aux Systèmes, Ecole Centrale Paris, Grande Voie des Vignes, 92290 Châtenay-Malabry, France (pauline.lafitte@ecp.fr).

[†]Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium (ward.melis@cs.kuleuven.be).

[‡]Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium (giovanni.samaey@cs.kuleuven.be).

is approximated at time $t = t^n$ by

$$U_i^n \approx \frac{1}{\Delta x} \int_{\mathcal{C}_i} u(x, t^n) dx. \quad (4)$$

Note that boldface is removed whenever the quantities are scalar. A numerical scheme is then constructed by integrating the conservation law (3) in space over the cell \mathcal{C}_i and in time from t^n to t^{n+1} to obtain

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \left(F_{i+1/2}^n - F_{i-1/2}^n \right), \quad (5)$$

in which $\Delta t = t^{n+1} - t^n$ and the *numerical flux* satisfies

$$F_{i\pm 1/2}^n \approx \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F(u(x_{i\pm 1/2}, t)) dt. \quad (6)$$

Clearly, equation (5) is conservative by construction. The numerical fluxes $F_{i\pm 1/2}^n$ can be obtained by constructing an (approximate) Riemann solver, based on a (possibly high-order) reconstruction of the solution in each of the cells using interpolation over the neighboring cells [26,32]. However, in the general nonlinear case, these spatial discretizations require the (possibly tedious) computation of the solutions of local Riemann problems.

Relaxation methods offer an interesting alternative in which the nonlinear hyperbolic conservation law is replaced by a linear transport equation with a stiff nonlinear (but local) source term, see, for instance, discrete kinetic schemes in [18,19,27] and, in particular, [1] which also contains a brief historical overview. In a relaxation method, the conservation law (1) is approximated by a problem of higher dimension containing a small relaxation parameter ε such that, when ε tends to zero, the original problem is recovered. In this paper, we will consider the relaxation problem to be a kinetic BGK equation. In a scalar one-dimensional setting, this equation describes the evolution of a distribution function $f^\varepsilon(x, v, t)$ of particles at position x with velocity v at time t and takes the following form:

$$\partial_t f^\varepsilon + v \partial_x f^\varepsilon = \frac{1}{\varepsilon} (\mathcal{M}_v(u^\varepsilon) - f^\varepsilon). \quad (7)$$

The left hand side of equation (7) describes the transport of particles, whereas the right hand side represents collisions between particles, which is modeled as a linear relaxation to the Maxwellian $\mathcal{M}_v(u^\varepsilon)$ with a relaxation time ε . The idea is that some of the difficulties associated with the original problem are avoided, while, for sufficiently small ε , the relaxation problem is a good approximation of the problem of interest. In particular, the advantage of the kinetic equation (7) over the conservation law (3) is the fact that the advection term in (7) is now linear, removing the difficulties associated with the high-order discretization of a nonlinear flux term. The disadvantage is the appearance of a stiff source term, which requires special care during time integration. The first methods, proposed in [1,19] are based on splitting techniques. As a consequence, the order in time is restricted to 2 and can only be improved by nontrivial manipulations, see [9]. More recently, several *asymptotic-preserving* methods based on IMEX techniques (in the sense of Jin [17]) have been proposed. An appealing idea along this line of thought, based on IMEX Runge-Kutta methods, is presented in [5,6] for general hyperbolic systems with relaxation. We refer to [10,14] for specific methods for the Boltzmann equation in the hyperbolic and diffusive regimes with a computational cost that is independent of ε . We note that the principle of a kinetic relaxation scheme also bears resemblance to the method of transport [13], see also [38].

In this paper, we propose to use a projective integration method to solve the stiff relaxation systems with an arbitrary order of accuracy in time. We will show that the resulting scheme constitutes a flexible, robust and fully explicit alternative to splitting and IMEX methods, while avoiding the construction of complicated and problem-specific (approximate) Riemann solvers. Projective integration methods were proposed in [15] for stiff systems of ordinary differential equations and analyzed in [23] for kinetic equations with a diffusive scaling. An arbitrary order version, based on Runge-Kutta methods, has been proposed recently in [22], where it was also analyzed for kinetic equations with an advection-diffusion limit. Projective integration is particularly suited for stiff problems with a clear spectral gap. In such stiff problems, the fast modes, corresponding to the Jacobian eigenvalues with large negative real parts, decay quickly, whereas the slow modes correspond to eigenvalues of smaller magnitude and are the solution components of practical interest. Projective integration allows a stable yet explicit integration of such

problems by first taking a few small (inner) steps with a simple, explicit method, until the transients corresponding to the fast modes have died out, and subsequently projecting (extrapolating) the solution forward in time over a large (outer) time step. Besides being robust and fully explicit, the resulting projective integration relaxation method is very appealing for nonlinear hyperbolic conservation laws because of its flexibility: once a solver is available, applying it to a different nonlinear hyperbolic conservation law merely amounts to changing the definition of the Maxwellian function $\mathcal{M}_v(u)$ in equation (7), leaving both the space and time discretizations untouched.

Projective integration fits within recent research efforts on numerical methods for multiscale simulation [11, 20, 21]. In this context, projective integration is a useful technique to effectively deal with problems in which there is a macroscopic (slow) dynamics whose mathematical formulation is not known and that can be captured “on-the-fly” by a short (appropriately initialized) microscopic simulation. Then, a few small steps of the full microscopic dynamics are combined with an extrapolation of the macroscopic, slow degrees of freedom only, and the resulting method is called *coarse projective integration*. Examples are, amongst others, bacterial chemotaxis [31], chemical reactions [30] and disease modeling [8]. For more examples, we refer to [21]. To conclude, we also mention alternative approaches to obtain a higher-order projective integration scheme which have been proposed in [24, 30]; see also [12, 34, 35] for related work.

The remainder of this paper is structured as follows. In section 2, we introduce the kinetic equations that form the basis of the relaxation method, and discuss their asymptotic equivalence with the original hyperbolic problem. In section 3, we describe the projective integration method that will be used to integrate these kinetic equations. We then analyze convergence of the resulting projective integration relaxation method for hyperbolic conservation laws in section 4, including the choice of appropriate method parameters. This analysis is based on the results in [22], for which we provide a number of alternative, simplified proofs that are specific for the relaxation systems of section 2. Section 5 reports the results of extensive numerical tests for a set of benchmark problems in both one and two space dimensions: linear advection, nonlinear conservation, the dam-break problem and Sod’s shock test. We conclude in section 6 with a brief discussion and some ideas for future work.

2 Relaxation systems

2.1 Kinetic equation and hydrodynamic limit

To solve equation (1), we introduce, as in [1], the (hyperbolically scaled) kinetic equation:

$$\partial_t \mathbf{f}^\varepsilon + \mathbf{v} \cdot \nabla_{\mathbf{x}} \mathbf{f}^\varepsilon = \frac{1}{\varepsilon} (\mathcal{M}_{\mathbf{v}}(\mathbf{u}^\varepsilon) - \mathbf{f}^\varepsilon), \quad (8)$$

or, equivalently,

$$\partial_t \mathbf{f}^\varepsilon + \sum_{d=1}^D v^d \partial_{x^d} \mathbf{f}^\varepsilon = \frac{1}{\varepsilon} (\mathcal{M}_{\mathbf{v}}(\mathbf{u}^\varepsilon) - \mathbf{f}^\varepsilon), \quad (9)$$

which models the evolution of a vector of particle distribution functions $\mathbf{f}^\varepsilon(\mathbf{x}, \mathbf{v}, t) = (f_m^\varepsilon(\mathbf{x}, \mathbf{v}, t))_{m=1}^M \in \mathbb{R}^M$ with particle positions $\mathbf{x} \in \mathbb{R}^D$ and velocities $\mathbf{v} \in V \subset \mathbb{R}^D$. The right hand side of (9) contains the BGK relaxation operator [3] that describes linear relaxation of \mathbf{f}^ε to a Maxwellian equilibrium $\mathcal{M}_{\mathbf{v}}(\mathbf{u}^\varepsilon) \in \mathbb{R}^M$, in which the argument $\mathbf{u}^\varepsilon(\mathbf{x}, t) = \langle \mathbf{f}^\varepsilon(\mathbf{x}, \mathbf{v}, t) \rangle$ is obtained via averaging over the measured velocity space (V, μ) :

$$\mathbf{u} := \langle \mathbf{f} \rangle = \int_V \mathbf{f} d\mu(\mathbf{v}). \quad (10)$$

The advantage of the kinetic formulation is that the advection term is now linear, and therefore easier to discretize. The disadvantage is the increased dimension, as well as the introduction of the stiff source term of size $O(1/\varepsilon)$. The projective integration scheme that we will propose in section 3 allows to integrate this stiff source term using an explicit method of arbitrary order.

To ensure that the kinetic equation (9) converges to the conservation law (1) in the hydrodynamic limit $\varepsilon \rightarrow 0$, one requires:

$$\begin{cases} \langle \mathcal{M}_{\mathbf{v}}(\mathbf{u}) \rangle = \mathbf{u}, \\ \langle v^d \mathcal{M}_{\mathbf{v}}(\mathbf{u}) \rangle = \mathbf{F}^d(\mathbf{u}), \quad 1 \leq d \leq D. \end{cases} \quad (11)$$

Then, one can show [1] that, in the limit of $\varepsilon \rightarrow 0$, the kinetic model (9) is approximated by the following equation:

$$\partial_t \mathbf{u}^\varepsilon + \sum_{d=1}^D \partial_{x^d} \mathbf{F}^d(\mathbf{u}^\varepsilon) = \varepsilon \sum_{d=1}^D \partial_{x^d} \left(\sum_{d'=1}^D \mathbf{B}_{dd'}(\mathbf{u}^\varepsilon) \partial_{x^{d'}} \mathbf{u}^\varepsilon \right) + O(\varepsilon^2), \quad (12)$$

in which $\mathbf{B}_{dd'}(\mathbf{u}^\varepsilon) \in \mathbb{R}^{M \times M}$ is generally called the diffusion matrix belonging to velocity components v^d and $v^{d'}$, and is defined as:

$$\mathbf{B}_{dd'}(\mathbf{u}) := \left\langle v^d v^{d'} \partial_{\mathbf{u}} \mathcal{M}_{\mathbf{v}}(\mathbf{u}) \right\rangle - \partial_{\mathbf{u}} \mathbf{F}^d \partial_{\mathbf{u}} \mathbf{F}^{d'}, \quad (13)$$

in which the $M \times M$ matrices $\partial_{\mathbf{u}} \mathcal{M}_{\mathbf{v}}(\mathbf{u})$ and $\partial_{\mathbf{u}} \mathbf{F}^d$ represent the Jacobian matrices of $\mathcal{M}_{\mathbf{v}}(\mathbf{u})$ and $\mathbf{F}(\mathbf{u})$, respectively.

Clearly, equation (9) is consistent with equation (1) to order 1 in ε . Moreover, the analysis reveals an additional condition on the choice of $\mathcal{M}_{\mathbf{v}}(\mathbf{u})$ and V . Indeed, to ensure the parabolicity of (12), the diffusion matrix $\mathbf{B}(\mathbf{u})$ should be positive definite. This leads to the so-called *subcharacteristic condition* [1, 7]:

$$\sum_{d,d'=1}^D \left(\mathbf{B}_{dd'}(\mathbf{u}) \boldsymbol{\xi}^d \cdot \boldsymbol{\xi}^{d'} \right) \geq 0, \quad (14)$$

which must be satisfied for all sets of vectors $\boldsymbol{\xi} = (\boldsymbol{\xi}^d)_{d=1}^D \in \mathbb{R}^{DM}$ with $\boldsymbol{\xi}^d \in \mathbb{R}^M$.

In what follows, we will always assume that the velocity space is discrete and of the form:

$$V = \{\mathbf{v}_j\}_{j=1}^J, \quad d\mu(\mathbf{v}) = \sum_{j=1}^J w_j \delta(\mathbf{v} - \mathbf{v}_j) d\mathbf{v}, \quad (15)$$

with \mathbf{v}_j denoting the chosen velocities and w_j the corresponding weights for which we have $\sum_{j=1}^J w_j = 1$. Due to this choice of V the system of kinetic equations (8) breaks up into a system of J coupled partial differential equations:

$$\partial_t \mathbf{f}_j^\varepsilon + \mathbf{v}_j \cdot \nabla_{\mathbf{x}} \mathbf{f}_j^\varepsilon = \frac{1}{\varepsilon} (\mathcal{M}_j(\mathbf{u}^\varepsilon) - \mathbf{f}_j^\varepsilon) \quad (1 \leq j \leq J), \quad (16)$$

in which $\mathbf{f}_j^\varepsilon(\mathbf{x}, t) \equiv \mathbf{f}^\varepsilon(\mathbf{x}, \mathbf{v}_j, t)$, and the only coupling between different velocities \mathbf{v}_j is through the computation of \mathbf{u}^ε , see equation (10). As $\varepsilon \rightarrow 0$, a Chapman-Enskog expansion allows to write:

$$\mathbf{f}_j^\varepsilon = \mathcal{M}_j(\mathbf{u}^\varepsilon) + O(\varepsilon), \quad (17)$$

so that, injecting it in (16) and taking the mean value over V , we get:

$$\partial_t \langle \mathcal{M}_j(\mathbf{u}^\varepsilon) \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v}_j \cdot \mathcal{M}_j(\mathbf{u}^\varepsilon) \rangle = O(\varepsilon).$$

Finally, the compatibility conditions (11) imply:

$$\partial_t \mathbf{u}^\varepsilon + \sum_{d=1}^D \partial_{x^d} \mathbf{F}^d(\mathbf{u}^\varepsilon) = O(\varepsilon). \quad (18)$$

Remark 2.1 (Minimal number of velocities). *In the relaxation problem (16), the minimal number J of discrete velocities depends on the spatial dimension of the problem. In particular, one needs to ensure – at least – that there is a velocity associated to each possible direction of motion. There are $J = 2$ possible velocity directions in 1D (left, right) and $J = 4$ possible directions in 2D (left, right, up, down). Since the computational complexity of the relaxation system is proportional to J , we will not consider higher values of J here. Hence, the one-dimensional stability analysis in section 4 will be performed specifically for $J = 2$, and in the experiments in section 5, we will only use $J = 2$ in 1D and $J = 4$ in 2D (the effect of including more velocities is taken up in section 4.3). Finally, we stress that when choosing $J = 2$ in 1D, the form of the discrete kinetic system given in equation (16) coincides precisely with the relaxation system introduced by Jin and Xin in [19].*

Remark 2.2 (Choice of ε). From equation (18), it is clear that the relaxation system (16) contains a modeling error that is proportional to ε . In our projective integration schemes, however, a finite value of ε will need to be chosen. This choice will also be dictated by the numerical schemes; in particular, roundoff errors will become important as $\varepsilon \rightarrow 0$, see section 4.4.

Remark 2.3 (Boundary conditions). We point out that, in general, the main difficulty of imposing boundary conditions arises when replacing the hyperbolic system (1) with the relaxation system (8). This issue is the subject of several papers in its own right (see, for instance, [2, 25]) and will not be treated in this work. We confine ourselves to periodic boundaries in the analysis (section 4) and to both periodic and Neumann boundary conditions in the numerical experiments (section 5).

2.2 One-dimensional examples

In one space dimension, we write system (1) as:

$$\partial_t \mathbf{u} + \partial_x \mathbf{F}(\mathbf{u}) = \mathbf{0}, \quad (19)$$

with $x \in \mathbb{R}$ the spatial variable, $\mathbf{u}(x, t) = (u_m(x, t))_{m=1}^M \in \mathbb{R}^M$ the set of conserved quantities, and $\mathbf{F}(\mathbf{u}) = (F_m(\mathbf{u}))_{m=1}^M \in \mathbb{R}^M$ the set of flux functions. Correspondingly, the system of kinetic equations (9) becomes:

$$\partial_t \mathbf{f}^\varepsilon + v \partial_x \mathbf{f}^\varepsilon = \frac{1}{\varepsilon} (\mathcal{M}_v(\mathbf{u}^\varepsilon) - \mathbf{f}^\varepsilon), \quad (20)$$

where $\mathbf{f}^\varepsilon(x, v, t) = (f_m^\varepsilon(x, v, t))_{m=1}^M \in \mathbb{R}^M$ denotes the vector of particle distribution functions and the particle velocities are represented as $v \in V \subset \mathbb{R}$.

In equation (20), the Maxwellian \mathcal{M}_v is as yet not completely defined, as we only require the conditions (11) to be satisfied. A physically relevant Maxwellian that corresponds to these conditions, is:

$$\mathcal{M}_v(\mathbf{u}^\varepsilon) = \mathbf{u}^\varepsilon + \frac{v \mathbf{F}(\mathbf{u}^\varepsilon)}{\langle v^2 \rangle}, \quad (21)$$

with corresponding velocity space $V = \mathbb{R}$ and measure:

$$d\mu(v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{v^2}{2\sigma^2}\right) dv. \quad (22)$$

To satisfy the subcharacteristic condition given in (14), the variance $\sigma^2 \in \mathbb{R}^+$ of the velocity measure $d\mu(v)$ needs to be chosen appropriately. For instance, in the scalar case ($M = 1$), using equations (13) and (21), we obtain the following constraint:

$$\left\langle v^2 \left(1 + \frac{v F'(u)}{\langle v^2 \rangle} \right) \right\rangle \geq (F'(u))^2.$$

Since this velocity space is odd symmetric, meaning that $\int_V h(v) d\mu(v) = 0$ for every odd function $h(v)$, this condition further reduces to:

$$\langle v^2 \rangle \geq (F'(u))^2.$$

For the Gaussian measure in equation (22) we have $\langle v^2 \rangle = \sigma^2$. Consequently, we require $\sigma \geq \max_u |F'(u)|$ to ensure parabolicity of equation (12). Based on (22), we choose a discrete measured symmetric velocity space with an even number J of velocities that satisfy $v_{J-j+1} \equiv -v_j$. From the measure given in (22), these discrete velocities $(v_j)_{j=1}^J$ and weights $(w_j)_{j=1}^J$ are derived as the nodes and weights of the corresponding Gauss-Hermite quadrature. In particular, for $J = 2$, this results in $v_j = \pm\sigma$, with corresponding weights $w_j = 1/2$.

An alternative suggestion, that is identical to the above choice for $J = 2$, was proposed in [7] and is of the following form:

$$\mathcal{M}_v(\mathbf{u}^\varepsilon) = \mathbf{u}^\varepsilon + \frac{\mathbf{F}(\mathbf{u}^\varepsilon)}{v}. \quad (23)$$

For both choices (21) and (23), the conditions (11) can be seen to be satisfied. In case of the Maxwellian given in equation (23), the specific values of the velocities v_j need to be chosen such that the subcharacteristic condition (14) is satisfied. When we further restrict to a scalar case ($M = 1$):

$$\partial_t u + \partial_x F(u) = 0, \quad (24)$$

and choosing an odd symmetric velocity space, the subcharacteristic condition again gives rise to the condition $\langle v^2 \rangle \geq (F'(u))^2$, which is always satisfied when choosing the discrete velocities as:

$$|v_j| \geq \max_u |F'(u)| \quad (1 \leq j \leq J). \quad (25)$$

The corresponding weights are chosen as $w_j = 1/J$. Note again that all boldfaced typesetting is removed for a scalar case.

For the numerical illustrations, we choose concretely the following examples:

Example 2.4. *The scalar linear advection equation,*

$$F(u) = a \cdot u, \quad a \in \mathbb{R}. \quad (26)$$

Example 2.5. *The scalar (inviscid) Burgers' equation,*

$$F(u) = u^2/2. \quad (27)$$

Example 2.6. *The one-dimensional Euler equations,*

$$\mathbf{u} = (\rho, \rho\bar{v}, E), \quad \mathbf{F}(\mathbf{u}) = (\rho\bar{v}, \rho\bar{v}^2 + P, (E + P)\bar{v}), \quad (28)$$

which can be closed by the polytropic equation of state:

$$P = (\gamma - 1) \left(E - \frac{1}{2} \rho\bar{v}^2 \right). \quad (29)$$

2.3 Two-dimensional examples

In two space dimensions, we write system (1) as

$$\partial_t \mathbf{u} + \partial_x \mathbf{F}^x(\mathbf{u}) + \partial_y \mathbf{F}^y(\mathbf{u}) = \mathbf{0}, \quad (30)$$

with $\mathbf{x} = (x, y) \in \mathbb{R}^2$ the spatial variables, $\mathbf{u}(\mathbf{x}, t) = (u_m(\mathbf{x}, t))_{m=1}^M \in \mathbb{R}^M$ the vector of conserved quantities, and $\mathbf{F}^{x,y}(\mathbf{u}) = (F_m^{x,y}(\mathbf{u}))_{m=1}^M \in \mathbb{R}^M$ the sets of fluxes in the x - and y - directions. Correspondingly, the kinetic system (9) becomes:

$$\partial_t \mathbf{f}^\varepsilon + v^x \partial_x \mathbf{f}^\varepsilon + v^y \partial_y \mathbf{f}^\varepsilon = \frac{1}{\varepsilon} (\mathcal{M}_{\mathbf{v}}(\mathbf{u}^\varepsilon) - \mathbf{f}^\varepsilon), \quad (31)$$

with $\mathbf{f}^\varepsilon(x, y, v^x, v^y, t) = (f_m^\varepsilon(x, y, v^x, v^y, t))_{m=1}^M \in \mathbb{R}^M$ the vector of particle distribution functions and the particle velocities given by $\mathbf{v} = (v^x, v^y) \in V \subset \mathbb{R}^2$, with $v^{x,y}$ the velocity of particles in the x - and y -direction, respectively.

Compared to the one-dimensional setting, the description of the discrete velocity space is considerably more elaborate, and many options have been documented, see, for instance, [1, 4, 28]. In the numerical examples in this paper, we choose the orthogonal velocities method (see [1]), which we now detail for the scalar case ($M = 1$). In this method, we choose a set of velocities with varying length and direction. More specifically, we first fix a maximal velocity length v_{\max} . We then consider R different velocity lengths:

$$\rho_r = \frac{r}{R} v_{\max}, \quad 1 \leq r \leq R,$$

and $4S$ different velocity directions:

$$\theta_s = \frac{s}{S} \frac{\pi}{2}, \quad 1 \leq s \leq 4S,$$

with $R, S \geq 1$. We then obtain $J = 4RS$ velocities $\mathbf{v}_j = (v_j^x, v_j^y)$, $1 \leq j \leq J$, by assigning an index $j = (r-1)4S + s$ to every length-direction pair (r, s) , and writing

$$v_j^x = \rho_r \cos(\theta_s), \quad v_j^y = \rho_r \sin(\theta_s), \quad 1 \leq r \leq R, \quad 1 \leq s \leq 4S. \quad (32)$$

The Maxwellian function \mathcal{M}_j for the j^{th} equation of system (16) is then chosen as:

$$\mathcal{M}_j(u) = u + v_j^x \frac{F^x(u)}{\langle (\mathbf{v}^x)^2 \rangle} + v_j^y \frac{F^y(u)}{\langle (\mathbf{v}^y)^2 \rangle}, \quad (33)$$

with $\mathbf{v}^{x,y} = (v_j^{x,y})_{j=1}^J$ the velocity components along the x - and y -directions. It can be shown that $\langle (\mathbf{v}^x)^2 \rangle = \langle (\mathbf{v}^y)^2 \rangle$ for the orthogonal velocities method (32).

The generalization to $M > 1$ is straightforward. In [1] it is proven that the subcharacteristic condition induces the following constraint on the choice of v_{\max} :

$$v_{\max}^2 \geq \frac{12R^2 (\|\partial_{\mathbf{u}} \mathbf{F}^x\|^2 + \|\partial_{\mathbf{u}} \mathbf{F}^y\|^2)}{(R+1)(2R+1)}, \quad (34)$$

where $\|\cdot\|$ is the matrix norm associated with the classical 2-norm when $M > 1$.

Remark 2.7. *As pointed out in remark 2.1, in the experiments in section 5 we will always choose $J = 4$ velocities in 2D. This is accomplished by setting $R = S = 1$ in the orthogonal velocities method (32).*

3 Projective integration

In this section, we construct a fully explicit, asymptotic-preserving, arbitrary order time integration method for the stiff system (16). The asymptotic-preserving property [17] implies that, in the limit when ε tends to zero, an ε -independent time step constraint, of the form $\Delta t = O(\Delta x)$, can be used, in agreement with the classical hyperbolic CFL constraint for the limiting equation (12). To achieve this, we will use a projective integration method [15, 23], which combines a few small time steps with a naive (*inner*) timestepping method, such as a direct forward Euler discretization, with a much larger (*projective, outer*) time step. The idea is sketched in figure 1.

The inner and outer integrator can be selected independently. In section 3.1, we discuss the inner integrator. Afterwards, in section 3.2, we consider the outer integrator, before studying their numerical properties in section 4.

3.1 Inner integrators

We intend to integrate (16) on a uniform, constant in time, periodic spatial mesh with spacing Δx , consisting of I mesh points $x_i = i\Delta x$, $1 \leq i \leq I$, with $I\Delta x = 1$, and a uniform time mesh with time step δt and discrete time instants $t^k = k\delta t$. (The generalization to space-time adaptive grids is of course straightforward.) The numerical solution on this mesh is denoted by $\mathbf{f}_{i,j}^k$, where we have dropped the superscript ε for conciseness. After discretizing in space, we obtain a semidiscrete system of ordinary differential equations:

$$\dot{\mathbf{f}} = \mathbf{D}_t(\mathbf{f}), \quad \mathbf{D}_t(\mathbf{f}) = -\mathbf{D}_{\mathbf{x},\mathbf{v}}(\mathbf{f}) + \frac{1}{\varepsilon}(\mathcal{M}_{\mathbf{v}}(\mathbf{u}) - \mathbf{f}), \quad (35)$$

where $\mathbf{D}_{\mathbf{x},\mathbf{v}}(\cdot)$ represents a suitable discretization of the convective derivative $v\partial_x$ (for instance, using upwind differences), and \mathbf{f} is a vector of size $M \cdot I \cdot J$ that results from discretizing the kinetic system (16) in space and velocity.

As inner integrator, we choose an explicit scheme, for which we will, later on, use the shorthand notation:

$$\mathbf{f}^{k+1} = S_{\delta t}(\mathbf{f}^k), \quad k = 0, 1, \dots \quad (36)$$

The forward Euler (FE) method and Runge-Kutta (RK) methods immediately come to mind.

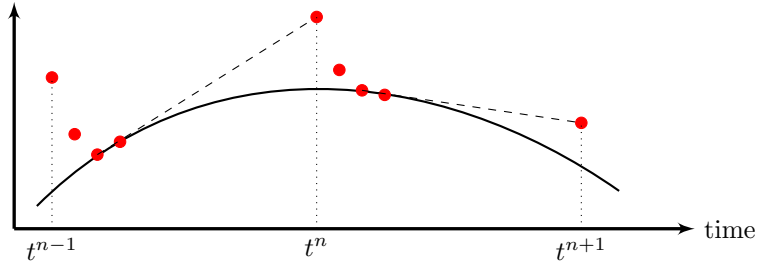


Figure 1: Sketch of projective integration. At each time instant, an explicit method is applied over a number of small time steps (red dots) so as to stably integrate the fast modes. As soon as these modes are sufficiently damped the solution is extrapolated using a much larger time step (dashed lines).

Forward Euler (FE). The simplest time discretization routine is the forward Euler method:

$$\mathbf{f}^{k+1} = \mathbf{f}^k + \delta t D_t(\mathbf{f}^k). \quad (37)$$

Higher-order Runge-Kutta (RK). To obtain higher-order accuracy in time in the inner integrator, one could also employ any Runge-Kutta method [16, 36], such as the second-order method:

$$\mathbf{k}_1 = D_t(\mathbf{f}^k), \quad (38)$$

$$\mathbf{k}_2 = D_t\left(\mathbf{f}^k + \frac{\delta t}{2} \mathbf{k}_1\right), \quad (39)$$

$$\mathbf{f}^{k+1} = \mathbf{f}^k + \delta t \mathbf{k}_2. \quad (40)$$

However, in section 4, we will show that higher-order inner Runge-Kutta methods of even order have spectral properties that make them unsuitable for use in conjunction with projective integration. Hence, in the following sections, we will always use forward Euler as inner integrator.

3.2 Outer integrators

In system (16), the small parameter ε in the relaxation term leads to the classical time step restriction of the form $\delta t = O(\varepsilon)$ for the inner integrator. However, as ε goes to 0, we obtain the limiting equation (18) for which a standard finite volume/forward Euler method only needs to satisfy a stability restriction of the form $\Delta t \leq C \Delta x$, with C a constant that depends on the specific choice of the scheme and the parameters of the equation.

In [23], it was proposed to use a projective integration method to accelerate such a brute-force integration; the idea, originating from [15], is the following. Starting from a computed numerical solution \mathbf{f}^n at time $t^n = n\Delta t$, one first takes $K + 1$ *inner* steps of size δt :

$$\mathbf{f}^{n,k+1} = S_{\delta t}(\mathbf{f}^{n,k}), \quad 0 \leq k \leq K, \quad (41)$$

in which the superscript pair (n, k) represents the numerical solution at time $t^{n,k} = n\Delta t + k\delta t$. The aim is to obtain a discrete derivative to be used in the *outer* step to compute $\mathbf{f}^{n+1} = \mathbf{f}^{n+1,0}$ via extrapolation in time, that is:

$$\mathbf{f}^{n+1} = \mathbf{f}^{n,K+1} + (\Delta t - (K + 1)\delta t) \frac{\mathbf{f}^{n,K+1} - \mathbf{f}^{n,K}}{\delta t}. \quad (42)$$

This method is called projective forward Euler (PFE), and it is the simplest instantiation of this class of integration methods [15].

Higher-order projective integration methods can be constructed by replacing each time derivative evaluation \mathbf{k}_s in a classical Runge-Kutta method by $K + 1$ steps of an inner integrator as follows (with

$\mathbf{f}^{n,0} = \mathbf{f}^n$ for consistency) [22]:

$$s = 1 : \begin{cases} \mathbf{f}^{n,k+1} = \mathbf{f}^{n,k} + \delta t \mathbf{D}_t(\mathbf{f}^{n,k}), & 0 \leq k \leq K \\ \mathbf{k}_1 = \frac{\mathbf{f}^{n,K+1} - \mathbf{f}^{n,K}}{\delta t} \end{cases} \quad (43)$$

$$2 \leq s \leq S : \begin{cases} \mathbf{f}_s^{n+c_s,0} = \mathbf{f}^{n,K+1} + (c_s \Delta t - (K+1)\delta t) \sum_{l=1}^{s-1} \frac{a_{s,l}}{c_s} \mathbf{k}_l, \\ \mathbf{f}_s^{n+c_s,k+1} = \mathbf{f}_s^{n+c_s,k} + \delta t \mathbf{D}_t(\mathbf{f}_s^{n+c_s,k}), & 0 \leq k \leq K \\ \mathbf{k}_s = \frac{\mathbf{f}_s^{n+c_s,K+1} - \mathbf{f}_s^{n+c_s,K}}{\delta t} \end{cases} \quad (44)$$

$$\mathbf{f}^{n+1} = \mathbf{f}^{n,K+1} + (\Delta t - (K+1)\delta t) \sum_{s=1}^S b_s \mathbf{k}_s. \quad (45)$$

To ensure consistency, the Runge-Kutta matrix $\mathbf{a} = (a_{s,l})_{s,l=1}^S$, weights $\mathbf{b} = (b_s)_{s=1}^S$, and nodes $\mathbf{c} = (c_s)_{s=1}^S$ satisfy (see, for instance, [16]) the conditions $0 \leq b_s \leq 1$ and $0 \leq c_s \leq 1$, as well as:

$$\sum_{s=1}^S b_s = 1, \quad \sum_{l=1}^{S-1} a_{s,l} = c_s, \quad 1 \leq s \leq S. \quad (46)$$

(Note that these assumptions imply that $c_1 = 0$ using the convention that $\sum_1^0 \cdot = 0$.)

In the numerical experiments, we will specifically use projective Runge-Kutta methods of orders 2 and 4, whose data is summarized in the Butcher tableaux in figure 2.

3.3 Stability of projective integration

We now briefly discuss the main stability properties of projective Runge-Kutta methods as derived in [22]. To that end, we introduce the test equation and its corresponding inner integrator:

$$\dot{y} = \lambda y, \quad y^{k+1} = \tau(\lambda \delta t) y^k, \quad \lambda \in \mathbb{C}. \quad (47)$$

As in [15], we call $\tau(\lambda \delta t)$ the *amplification factor* of the inner integrator. (For instance, if the inner integrator is the forward Euler scheme, we have $\tau(\lambda \delta t) = 1 + \lambda \delta t$.) The inner integrator is stable if $|\tau| \leq 1$. The question then is for which subset of these values the projective integration method is also stable.

Considering projective forward Euler with amplification factor, it can easily be seen from (42) that the projective forward Euler method is stable if:

$$\left| \left[\left(\frac{\Delta t - (K+1)\delta t}{\delta t} + 1 \right) \tau - \frac{\Delta t - (K+1)\delta t}{\delta t} \right] \tau^K \right| \leq 1, \quad (48)$$

for all eigenvalues τ of the inner integrator for the kinetic equation (20). The goal is to take a projective time step $\Delta t = O(\Delta x)$, whereas $\delta t = O(\varepsilon)$ necessarily to ensure stability of the inner brute-force forward Euler integration. Since we are interested in the limit $\varepsilon \rightarrow 0$ for fixed Δx , we look at the limiting stability regions as $\Delta t/\delta t \rightarrow \infty$. In this regime, it is shown in [15] that the values τ for which the condition (48) is satisfied lie in the union of two separated disks $\mathcal{D}_1^{\text{PFE}} \cup \mathcal{D}_2^{\text{PFE}}$ with:

$$\mathcal{D}_1^{\text{PFE}} = \mathcal{D}\left(1 - \frac{\delta t}{\Delta t}, \frac{\delta t}{\Delta t}\right) \text{ and } \mathcal{D}_2^{\text{PFE}} = \mathcal{D}\left(0, \left(\frac{\delta t}{\Delta t}\right)^{1/K}\right), \quad (49)$$

where $\mathcal{D}(c, r)$ denotes the disk with center $(c, 0)$ and radius r . One then aims at positioning the eigenvalues that correspond to modes that are quickly damped by the timestepper in $\mathcal{D}_2^{\text{PFE}}$, whereas the eigenvalues in $\mathcal{D}_1^{\text{PFE}}$ should correspond to slowly decaying modes. The projective integration method then allows for accurate integration of the modes in $\mathcal{D}_1^{\text{PFE}}$ while maintaining stability for the modes in $\mathcal{D}_2^{\text{PFE}}$.

c	a
	b^T

0	1/2
1/2	0 1

0	1/2	1/2	1/2
1/2	0	0	1
1	1/6	1/3	1/3 1/6

Figure 2: Butcher tableaux for Runge-Kutta methods. Left: general notation; middle: RK2 method; right: RK4 method.

In [22], this analysis is extended to the projective Runge-Kutta case, showing that, in the limit when $\delta t/\Delta t$ tends to 0, the stability region of the projective Runge-Kutta method also breaks up into two regions $\mathcal{D}_1^{\text{PRK},q}$ and $\mathcal{D}_2^{\text{PRK},q}$. Moreover, these regions satisfy $\forall q \in \{1, 2, \dots\}$:

$$\mathcal{D}_1^{\text{PRK},q+1} \supseteq \mathcal{D}_1^{\text{PRK},q} \supseteq \mathcal{D}_1^{\text{PFE}} \quad \text{and} \quad \mathcal{D}_2^{\text{PRK},q+1} \supseteq \mathcal{D}_2^{\text{PRK},q} \supseteq \mathcal{D}_2^{\text{PFE}},$$

in which the constant q indicates the order of the specific Runge-Kutta method. This implies that the stability regions of lower-order projective integration methods are contained within those of the higher-order ones. The main conclusion is that, whereas the stability regions of higher-order projective Runge-Kutta methods differ from those of projective forward Euler in their precise shape, their qualitative dependence on the parameters of projective integration (δt , K and Δt) is identical, and method parameters that are suitable for projective forward Euler will also be suitable for the higher-order projective Runge-Kutta methods.

4 Numerical properties

Now we are ready to use the projective integration method on the relaxation system (35). The parameters to determine are then the time scale separation parameter ε in the relaxation system (35), as well as the projective integration parameters: the inner time step δt , the outer time step Δt and the number of inner steps K . The projective integration parameters δt , K , and Δt can be determined by imposing that all the eigenvalues of the selected inner integrator scheme fall into the stability region of the projective integration method. While the numerical experiments also deal with systems of nonlinear hyperbolic conservation laws in multiple space dimensions, the analysis is restricted to a one-dimensional, scalar, linear setting. In section 4.1, we will calculate the spectrum of the inner integrators constructed for the relaxation system given in (35), and this in the specific case $J = 2$ (see Remark 2.1). This result is a special case (with an adapted proof) of the more general result for any (even) number of velocities J that was obtained in [22]: for $J = 2$, we are able to derive explicit asymptotic expansions for both the fast and slow eigenvalues in the spectrum. Then, we derive suitable choices for the projective integration parameters in the specific setting of this paper (section 4.2). Thereafter, in section 4.3, we regard the influence of adding more velocities to the kinetic relaxation system. Finally, we elaborate on the consistency of the resulting method for the hyperbolic conservation law (3) in section 4.4. There, we will also see how to properly choose the relaxation parameter ε .

4.1 Spectrum of inner integrators

To compute bounds on the spectrum of the inner integrator for the scalar kinetic equation (20) with a linear Maxwellian,

$$\mathcal{M}_v(u) = u + \frac{u}{v},$$

we first rewrite the semi-discretized kinetic equation (35) in the (spatial) Fourier domain:

$$\partial_t \hat{\mathbf{F}}(\zeta_m) = \mathcal{B} \hat{\mathbf{F}}(\zeta_m), \quad \mathcal{B} = \frac{1}{\varepsilon} (-\varepsilon \mathbf{D} + \mathbf{M}\mathbf{P} - \mathbf{I}), \quad (50)$$

with $\hat{\mathbf{F}} \in \mathbb{C}^2$, \mathbf{B} , $\mathbf{D} \in \mathbb{C}^{2 \times 2}$, \mathbf{M} , $\mathbf{P} \in \mathbb{R}^{2 \times 2}$, and \mathbf{I} the identity matrix of dimension $J = 2$. In (50), the matrix \mathbf{D} represents the (diagonal) Fourier matrix of the spatial discretization chosen for the convection part, \mathbf{P} is the Fourier matrix of the averaging of f^ε over the positive and negative velocities,

$$\mathbf{P} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad (51)$$

and the matrix \mathbf{M} represents the Fourier transform of the Maxwellian:

$$\mathbf{M} = \mathbf{I} + \mathbf{V}^{-1}, \quad (52)$$

with $\mathbf{V} = \text{diag}([v_*, -v_*])$ and v_* the chosen (discrete) velocity component.

Since we are using a symmetric velocity space, we have the following property on the diagonal elements of the matrix \mathbf{D} : $D_1 = \overline{D_2}$. Hence, from now on, we write the diagonal elements of \mathbf{D} as:

$$D_{1,2} = \alpha \pm i\beta, \quad (53)$$

in which α and β depend on the spatial discretization, the velocity value v_* and the (discrete) Fourier variable ζ_m , $m = 1, \dots, I$, see table 1. As a result, the matrix \mathbf{B} in (50) also depends on ζ_m through \mathbf{D} .

Since in the Fourier domain we are calculating the eigenvalues of (2×2) matrices, we can compute all eigenvalues of the matrix \mathbf{B} in the following theorem.

Theorem 4.1. *Under the above assumptions, the spectrum of the matrix $\mathbf{B} = \frac{1}{\varepsilon}(\mathbf{MP} - \mathbf{I} - \varepsilon\mathbf{D})$ contains one slow eigenvalue λ_1 and one fast eigenvalue λ_2 which can be expanded in terms of ε as:*

$$\begin{aligned} \lambda_1 &= \alpha + \beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon + O(\varepsilon^3) + i \left(\frac{\beta}{v_*} + 2 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) \right) \\ \lambda_2 &= -\frac{1}{\varepsilon} + \alpha - \beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon + O(\varepsilon^3) - i \left(\frac{\beta}{v_*} + 2 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) \right). \end{aligned} \quad (54)$$

Consequently, when choosing $v_* = 1$ (and also for $\varepsilon \rightarrow 0$ with $v_* \neq 1$), the spectrum of \mathbf{B} can be written as:

$$\text{Sp}(\mathbf{B}) \subset \left\{ \mathcal{D} \left(-\frac{1}{\varepsilon}, \max_{\zeta_m} \sqrt{\alpha^2 + \frac{\beta^2}{v_*^2}} \right) \cup \{\lambda_1\} \right\}. \quad (55)$$

Proof. Since we discretized in space, the Fourier variable ζ is a discrete variable, $\zeta_m = 2\pi m \Delta x$, $m = 1, \dots, I$. The matrix product \mathbf{MP} and the matrix $\mathbf{D}(\zeta_m)$ in the Fourier domain become:

$$\mathbf{MP} = \frac{1}{2} \begin{pmatrix} 1 + 1/v_* & 1 + 1/v_* \\ 1 - 1/v_* & 1 - 1/v_* \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} z^+ & 0 \\ 0 & z^- \end{pmatrix}, \quad (56)$$

where we used the matrices \mathbf{M} and \mathbf{P} given in equations (52) and (51). Furthermore, in (56) we adopted the shorthand notation $z^\pm = \alpha \pm i\beta$, where α and β depend on the spatial discretization method, the velocity value v_* and the Fourier variable ζ_m , see table 1.

We start by calculating the spectrum of the matrix $\mathcal{A} = \varepsilon\mathbf{B}$. The eigenvalues $\tilde{\lambda}$ of \mathcal{A} can be obtained as the roots of its characteristic polynomial, denoted by $\chi_{\mathcal{A}}(\tilde{\lambda})$, which is given by:

$$\begin{aligned} \chi_{\mathcal{A}}(\tilde{\lambda}) &= \left| \mathcal{A} - \tilde{\lambda}\mathbf{I} \right| \\ &= \tilde{\lambda}^2 + (1 - \varepsilon(z^+ + z^-))\tilde{\lambda} + \varepsilon \left(\frac{1 - v_*}{2v_*} z^- - \frac{1 + v_*}{2v_*} z^+ \right) + \varepsilon^2 z^+ z^- = 0. \end{aligned} \quad (57)$$

Using the equalities

$$z^+ + z^- = 2\alpha, \quad z^+ - z^- = i2\beta, \quad z^+ z^- = \alpha^2 + \beta^2,$$

the roots of equation (57) can be calculated as:

$$\tilde{\lambda}_{1,2} = \frac{-1 + 2\alpha\varepsilon \pm \sqrt{\Delta}}{2}, \quad (58)$$

	upwind 1	upwind 2	upwind 3
α	$-v_* \frac{2 \sin^2(\zeta_m/2)}{\Delta x}$	$-v_* \frac{3 - 4 \cos(\zeta_m) + \cos(2\zeta_m)}{2\Delta x}$	$-v_* \frac{3 - 4 \cos(\zeta_m) + \cos(2\zeta_m)}{6\Delta x}$
β	$-v_* \frac{\sin(\zeta_m)}{\Delta x}$	$-v_* \frac{4 \sin(\zeta_m) - \sin(2\zeta_m)}{2\Delta x}$	$-v_* \frac{8 \sin(\zeta_m) - \sin(2\zeta_m)}{6\Delta x}$

Table 1: Dependence of α and β on v_* , ζ and the chosen spatial discretization technique.

where the discriminant Δ of equation (57) is written as:

$$\Delta = 1 - 4\beta^2\varepsilon^2 + \iota 4 \frac{\beta}{v_*} \varepsilon.$$

Performing a Taylor series expansion of $\sqrt{\Delta}$ for $\varepsilon \rightarrow 0$ in (58) leads to:

$$\sqrt{\Delta} = 1 + 2\beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) + \iota \left(2 \frac{\beta}{v_*} \varepsilon + 4 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^3 + O(\varepsilon^5) \right).$$

Plugging this expansion into equation (58), we obtain:

$$\begin{aligned} \tilde{\lambda}_1 &= \frac{-1 + 2\alpha\varepsilon + \sqrt{\Delta}}{2} \\ &= \alpha\varepsilon + \beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) + \iota \left(\frac{\beta}{v_*} \varepsilon + 2 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^3 + O(\varepsilon^5) \right) \\ \tilde{\lambda}_2 &= \frac{-1 + 2\alpha\varepsilon - \sqrt{\Delta}}{2} \\ &= -1 + \alpha\varepsilon - \beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) - \iota \left(\frac{\beta}{v_*} \varepsilon + 2 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^3 + O(\varepsilon^5) \right). \end{aligned}$$

Finally, the dominant and fast eigenvalues, λ_1 and λ_2 , of the matrix $\mathbf{B} = \frac{1}{\varepsilon} \mathbf{A}$ are then given by:

$$\begin{aligned} \lambda_1 &= \alpha + \beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon + O(\varepsilon^3) + \iota \left(\frac{\beta}{v_*} + 2 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) \right) \\ \lambda_2 &= -\frac{1}{\varepsilon} + \alpha - \beta^2 \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon + O(\varepsilon^3) - \iota \left(\frac{\beta}{v_*} + 2 \frac{\beta^3}{v_*} \left(\frac{1 - v_*^2}{v_*^2} \right) \varepsilon^2 + O(\varepsilon^4) \right). \end{aligned}$$

□

When we write the Fourier transform of the inner forward Euler scheme (37) as:

$$\hat{\mathbf{F}}^{k+1} = \mathbf{S}_{\delta t} \hat{\mathbf{F}}^k = (\mathbf{I} + \delta t \mathbf{B}) \hat{\mathbf{F}}^k, \quad (59)$$

it is clear that the amplifications factors $\boldsymbol{\tau} = (\tau_1, \tau_2)$ of the forward Euler scheme, which are the eigenvalues of $\mathbf{S}_{\delta t}$, and the eigenvalues $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ of the matrix \mathbf{B} are related via

$$\tau_j = 1 + \lambda_j \delta t, \quad j \in \{1, 2\}. \quad (60)$$

Thus, the spectrum of an inner forward Euler timestepper satisfies:

$$\text{Sp}(\mathbf{I} + \delta t \mathbf{B}) \subset \left\{ \mathcal{D} \left(1 - \frac{\delta t}{\varepsilon}, \delta t \max_{\zeta_m} \sqrt{\alpha^2 + \frac{\beta^2}{v_*^2}} \right) \cup \{1 + \lambda_1 \delta t\} \right\}, \quad (61)$$

with λ_1 given in theorem 4.1.

For higher-order Runge-Kutta inner integrators (of order Q), we have

$$\tau_j = 1 + \sum_{q=1}^Q \frac{(\lambda_j \delta t)^q}{q!}, \quad j \in \{1, 2\}, \quad (62)$$

and the corresponding spectrum reads:

$$\text{Sp} \left(\mathbf{I} + \sum_{q=1}^Q \frac{(\delta t \mathbf{B})^q}{q!} \right) \subset \left\{ \mathcal{D} \left(1 + \sum_{q=1}^Q \frac{(-1)^q}{q!} \left(\frac{\delta t}{\varepsilon} \right)^q, C \delta t \max_{\zeta_m} \sqrt{\alpha^2 + \frac{\beta^2}{v_*^2}} \right) \cup \left\{ 1 + \sum_{q=1}^Q \frac{(\lambda_1 \delta t)^q}{q!} \right\} \right\}, \quad (63)$$

in which the constant C depends on the order of the Runge-Kutta method. The spectrum in (63) follows from transforming the spectrum of \mathbf{B} in (54) by the particular expression of the Runge-Kutta inner integrator amplification factor given by (62).

4.2 Method parameters

In this section the projective integration method parameters will be determined by ensuring that the spectrum of the inner integrator falls within the stability region of the projective forward Euler method. First, we select a suitable inner time step δt such that the fast modes are quickly damped (section 4.2.1). Then, we choose the outer time step Δt commensurate with the slow part of the evolution (section 4.2.2). Finally, we fix K to ensure overall stability (section 4.2.3).

4.2.1 Choice of inner integrator and time step

Let us first discuss the effect of the inner integrator choice. To that end, we look at the discretization error and the desired stability properties.

Concerning stability, we deduce from the stability properties of the projective integration method (see section 3.3) that it is preferable to center the part of the inner timestepper spectrum corresponding to quickly damped modes around 0. Since, for forward Euler, these fast modes are given by (61), we choose, for inner forward Euler, $\delta t = \varepsilon$.

For higher-order inner integrators of even order, one cannot center eigenvalues τ_2 corresponding to quickly damped modes around 0. For instance, for a second-order Runge-Kutta method we have:

$$\min_{\lambda_2 \delta t} \tau_2 = \min_{\lambda_2 \delta t} \left(1 + \lambda_2 \delta t + \frac{1}{2} (\lambda_2 \delta t)^2 \right) = \frac{1}{2}, \quad (64)$$

and this value is reached for $\lambda_2 \delta t = -1$ or, equivalently, $\delta t = \varepsilon$. Although Runge-Kutta methods of odd order can center the eigenvalues around 0, as we will show below, the discretization error of the projective integration scheme is dominated by the error of the outer integrator, whereas the discretization error due to the inner integrator is negligible. As a consequence, we conclude that there is no point in using a higher-order time discretization for the inner integrator.

4.2.2 Outer time step

Given the inner time step $\delta t = \varepsilon$, we choose Δt such that the dominant eigenvalue τ_1 of the inner integrator lies inside the stability region $\mathcal{D}_1^{\text{PRK},q}$. Let us first look at the projective forward Euler method, with stability regions (49). We have the following condition on Δt such that τ_1 is contained within $\mathcal{D}_1^{\text{PFE}}$:

$$\left(\Re(\tau_1) - \left(1 - \frac{\varepsilon}{\Delta t} \right) \right)^2 + \left(\Im(\tau_1) \right)^2 \leq \left(\frac{\varepsilon}{\Delta t} \right)^2. \quad (65)$$

Using the expressions of τ_1 in (60) and λ_1 given by (54) in theorem 4.1, for which we both consider only terms up to order ε , we compute:

$$\begin{aligned} \left((1 + \varepsilon\alpha) - \left(1 - \frac{\varepsilon}{\Delta t}\right) \right)^2 + \left(\frac{\beta}{v_*} \varepsilon \right)^2 &\leq \left(\frac{\varepsilon}{\Delta t} \right)^2 \\ \left(\alpha + \left(\frac{1}{\Delta t} \right) \right)^2 + \left(\frac{\beta}{v_*} \right)^2 &\leq \left(\frac{1}{\Delta t} \right)^2 \\ \alpha^2 + \frac{2\alpha}{\Delta t} + \frac{\beta^2}{v_*^2} &\leq 0. \end{aligned}$$

From the last inequality, we deduce a bound on Δt of the form:

$$\Delta t \leq \min_{\zeta_m} \left(\frac{-2\alpha v_*^2}{\alpha^2 v_*^2 + \beta^2} \right). \quad (66)$$

For the projective forward Euler method and first-order upwind discretization, using table 1, the restriction in (66) produces a CFL-like bound:

$$\Delta t \leq \frac{\Delta x}{v_*}.$$

As indicated in section 2.2, we need to choose $v_* \geq \max_u |F'(u)|$ to satisfy the subcharacteristic condition (14). When choosing $v_* = \max_u |F'(u)|$, we obtain exactly the stability condition for the forward Euler method applied to the original hyperbolic conservation law (3).

When combining a higher-order upwind method with forward Euler timestepping, the time step restriction becomes much more severe, in exactly the same way as would be the case with a direct higher-order upwind/forward Euler discretization of equation (3). Using table 1 and equation (66), we find:

$$\begin{aligned} \Delta t &\leq \frac{2\pi^2 v_* (3 - 10\pi^2 \Delta x^2)}{3} \Delta x^3 && \text{(second-order upwind)} \\ \Delta t &\leq \frac{2\pi^2 v_* (3 - 2\pi^2 \Delta x^2)}{9} \Delta x^3 && \text{(third-order upwind)}. \end{aligned}$$

The reason for these severe conditions is that, for higher-order upwind methods, the dominant eigenvalues do not lie on a circle anymore. In fact, they belong to a region that is much steeper close to (1,0) in the complex τ -plane. Since the dominant stability region of projective forward Euler is always a circle, we need to choose its radius sufficiently large such that even the steepest eigenvalues fall into this circular stability region. When we use a higher-order projective Runge-Kutta method, we should determine the bound on Δt using its proper dominant stability region (see, for instance, [22]). For example, in case of the second-order projective Runge-Kutta method, we obtain the following CFL-like bound:

$$\Delta t \leq \nu \Delta x,$$

where ν is the CFL-number which depends on the coefficients α and β . It is important to notice, however, that in all above cases Δt never depends on ε .

4.2.3 The number K of inner time steps

The only parameter that remains to be chosen is the number K of inner steps such that all fast eigenvalues τ_2 (see equation (60) or (62)) corresponding to quickly damped modes are contained within the stability region $\mathcal{D}_2^{\text{PFE}}$ in (49). For given time steps δt and Δt , this is achieved by expressing that the radius of the fast eigenvalues zone may not exceed the radius of the stability region $\mathcal{D}_2^{\text{PFE}}$. For forward Euler as inner integrator, we obtain a condition for K of the form:

$$c\varepsilon \leq \left(\frac{\varepsilon}{\Delta t} \right)^{1/K}, \quad c = \max_{\zeta} \sqrt{\alpha^2 + \frac{\beta^2}{v_*^2}}.$$

From this we obtain the following lower bound on K :

$$K \geq \frac{1}{1 + \frac{\log(c)}{\log(\varepsilon)}} + \frac{\log(\Delta t)}{\log\left(\frac{1}{\varepsilon c}\right)}.$$

Therefore, for ε small, a safe choice is taking $K \geq 2$, uniformly in ε . The computational cost of the method is then independent of ε .

Note that, when choosing a higher-order inner Runge-Kutta method of even order, the fast eigenvalues τ_2 cannot be centered around 0, see (64). In that case, we need to impose that the radius of the stability region $\mathcal{D}_2^{\text{PFE}}$ in (49) is sufficiently large to encompass the fast eigenvalues zone of the inner Runge-Kutta method, yielding:

$$\tilde{c} \leq \left(\frac{\delta t}{\Delta t}\right)^{1/K}, \quad \tilde{c} = 1 + \sum_{q=1}^Q \frac{(-1)^q}{q!} \left(\frac{\delta t}{\varepsilon}\right)^q + C \delta t \max_{\zeta} \sqrt{\alpha^2 + \frac{\beta^2}{v_*^2}}.$$

Taking the logarithm of both sides and rearranging terms ultimately leads to

$$K \geq \frac{\log\left(\frac{1}{\delta t}\right)}{\log\left(\frac{1}{\tilde{c}}\right)} + \frac{\log(\Delta t)}{\log\left(\frac{1}{\tilde{c}}\right)},$$

which results in a condition of the form $K \geq \log(1/\varepsilon)$ since $\delta t = O(\varepsilon)$. Hence, using an inner integrator of even order destroys the asymptotic-preserving nature of the projective integration method.

4.3 Number of velocities

In remark 2.1, we stated that we use $J = 2$ discrete velocities in 1D as using more velocities increases the computational cost of the method. Nonetheless, here, we investigate the effect of using more discrete velocity components in 1D, for which we always assume an even number J . More specifically, we consider the influence on the dominant eigenvalues λ_1 since the artificial diffusion of the kinetic relaxation system depends on the position of these components. In [22], it is shown that, when using the Maxwellian (23) with linear flux function $F(u) = a \cdot u$, the dominant eigenvalues (to order ε) are given by:

$$\lambda_1 = \langle \boldsymbol{\alpha} \rangle + O(\varepsilon) + \iota \left(a \left\langle \frac{\boldsymbol{\beta}}{\mathbf{v}} \right\rangle + O(\varepsilon) \right),$$

in which $\boldsymbol{\alpha} = (\alpha_j)_{j=1}^J$ and $\boldsymbol{\beta} = (\beta_j)_{j=1}^J$ represent the real and imaginary parts appearing in the diagonal Fourier domain matrix $\mathbf{D} = \text{diag}(\boldsymbol{\alpha} + \iota \boldsymbol{\beta})$ of the spatial discretization. They are retrieved from table 1 by replacing v_* by $|v_j|$ for α_j and v_* by v_j for β_j . When repeating the proof in [22] using the Maxwellian in (21), we now obtain:

$$\lambda_1 = \langle \boldsymbol{\alpha} \rangle + O(\varepsilon) + \iota \left(\frac{a}{\langle \mathbf{v}^2 \rangle} \langle \boldsymbol{\beta} \mathbf{v} \rangle + O(\varepsilon) \right),$$

from which we find that only the imaginary part of λ_1 is affected when changing the Maxwellian.

Another effect induced by these Maxwellians is the set of corresponding discrete velocities and weights. As discussed in section 2.2, for the first Maxwellian, the weights are chosen as $w_j = 1/J$ together with the following uniform velocity discretization:

$$v_j = \begin{cases} v_* + \Delta v(j-1) & (1 \leq j \leq J/2) \\ v_{J-j+1} & (J/2+1 \leq j \leq J), \end{cases} \quad (67)$$

where $v_* \geq |a|$ is chosen such that the discretization complies with the subcharacteristic condition (25) and $\Delta v > 0$ denotes the velocity increment. For the Maxwellian in (21) with velocity measure (22), the velocities and weights are chosen as the nodes and weights from Gauss-Hermite quadrature, and the variance of the velocity measure $\sigma^2 = \langle \mathbf{v}^2 \rangle$ must satisfy the subcharacteristic condition $\sigma^2 \geq a^2$.

We now regard the influence of J on the position of the dominant eigenvalues. We first note that, since β_j can be written as $\beta_j = v_j h(\zeta_m)$ (see table 1), for both Maxwellians, the imaginary part (to order ε) of the dominant eigenvalue λ_1 is independent of J . This can be seen as follows:

$$a \left\langle \frac{\beta}{v} \right\rangle = \frac{a}{J} \sum_{j=1}^J h(\zeta_m) = ah(\zeta_m), \quad \frac{a}{\langle v^2 \rangle} \langle \beta v \rangle = \frac{a}{\langle v^2 \rangle} \sum_{j=1}^J w_j v_j^2 h(\zeta_m) = ah(\zeta_m).$$

On the other hand, we have $\alpha_j = -|v_j|g(\zeta_m)$, giving:

$$\langle \alpha \rangle = -g(\zeta_m) \sum_{j=1}^J w_j |v_j|. \quad (68)$$

It is clear that for the Maxwellian (23) with uniform velocity discretization (67) and weights $w_j = 1/J$, the sum in equation (68) increases with J . This is seen by explicitly computing the sum as follows:

$$\frac{1}{J} \sum_{j=1}^J |v_j| = \frac{2}{J} \sum_{j=1}^{J/2} |v_j| = v_* + \frac{2\Delta v}{J} \sum_{j=1}^{J/2} (j-1) = v_* + \frac{\Delta v}{J} \frac{J(J-2)}{4},$$

where we used the known result on sums of successive natural numbers to arrive at the last equality. Substituting this into equation (68) yields:

$$\langle \alpha \rangle = -g(\zeta_m) \left(v_* + \frac{\Delta v}{4} (J-2) \right),$$

which indeed reveals a linear dependence on the number of velocities J . We conclude that, when increasing the number of velocities for Maxwellian (23), not only does the method become more expensive, the dominant eigenvalues of the kinetic relaxation system are also stretched out to the left in the complex plane (however, preserving their imaginary part), thus increasing the artificial diffusion of the problem.

For the Maxwellian (21) with Gauss-Hermite velocity discretization, it is known that integration with respect to the measure (22) of polynomials of v up to order $2J-1$ yields an exact result. However, since the function $|v|$ in equation (68) is not a polynomial, the value $\langle |v| \rangle$ computed by Gauss-Hermite quadrature will only be an approximation that converges to the true value $\langle |v| \rangle = \sigma \sqrt{2/\pi}$ as $J \rightarrow \infty$. In figure 3, we plot the computed values $\langle |v| \rangle$ (blue line) as a function of J together with the exact value $\langle |v| \rangle$ (black line). It is seen that the computed values alternately over- and underestimate the true value. However, since we always require an even number J , we observe convergence to the true value from above (red line). In conclusion, for the Maxwellian (21), the addition of more velocities compresses the dominant eigenvalues in the complex plane (however, preserving their imaginary part) thus reducing the artificial diffusion of the kinetic relaxation system at the cost of increasing its size. As indicated in figure 3, the reduction in diffusion is expected to be most significant for low values of J . The above observations seem to extend naturally to the nonlinear case. However, a detailed analysis would lead us too far.

4.4 Consistency analysis

We now examine the consistency behavior of the proposed method in a scalar, one-dimensional setting ($M = D = 1$). To that end, we regard the local truncation error of the method. The exposition in this section is based on the consistency derivation in [23], which becomes simpler in the hyperbolic case. We introduce the following notation for $N \geq 0$ and $0 \leq k \leq K+1$:

1. $\tilde{\mathbf{f}}^{N,k}$ denotes the exact solution at time $t^{N,k} = N\Delta t + k\delta t$ and is a vector of length $I \cdot J$ obtained by collecting $f^\varepsilon(x_i, v_j, t^{N,k})$, $\forall 1 \leq i \leq I, 1 \leq j \leq J$;
2. $\mathbf{f}^{N,k}$ denotes the numerical solution at time $t^{N,k}$ starting from the exact solution, that is, $\mathbf{f}^{N,k} = S_{\delta t}^k(\tilde{\mathbf{f}}^N)$. It is a vector of length $I \cdot J$ obtained by collecting $f_{i,j}^{N,k}$, $\forall 1 \leq i \leq I, 1 \leq j \leq J$;
3. $\mathbf{u}^N = \langle \mathbf{f}^N \rangle$ and $\tilde{\mathbf{u}}^N = \langle \tilde{\mathbf{f}}^N \rangle$ represent the numerical and exact conserved quantity at time $t^N \equiv t^{N,0}$ obtained by averaging over velocity space. Both are vectors of length I .

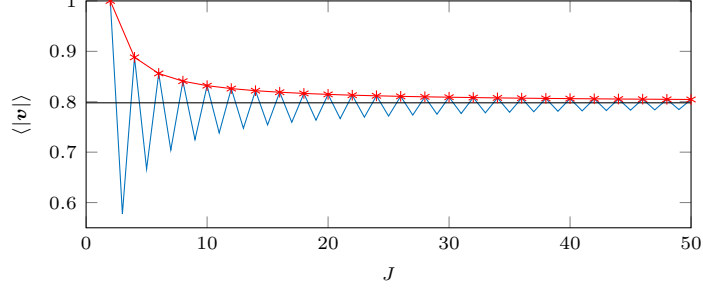


Figure 3: Convergence of $\langle |v| \rangle$ (blue line) to $\langle |v| \rangle = \sigma\sqrt{2/\pi}$ (black line) for Gauss-Hermite quadrature and $\sigma = 1$. The red line shows convergence for an even number J .

Since the goal of the simulations is to obtain $\mathbf{u}(x, t) = \langle \mathbf{f}^\varepsilon(x, v, t) \rangle$, we define the local truncation error at time t^{N+1} as:

$$E^{N+1} = \frac{\tilde{\mathbf{u}}^{N+1} - \mathbf{u}^{N+1}}{\Delta t}. \quad (69)$$

We focus here on the consistency analysis for PFE. The analysis can be straightforwardly extended to higher-order PRK methods, see [22], since \mathbf{k}_s on each stage s in the projective Runge-Kutta methods are computed as finite difference approximations of the time derivative in which the function values are obtained by the PFE method, see equations (43)-(44). By averaging equation (42) over velocity space and substituting the result into (69), we find:

$$E^{N+1} = \frac{\tilde{\mathbf{u}}^{N+1} - \mathbf{u}^{N,K+1}}{\Delta t} - \left(\frac{\Delta t - (K+1)\delta t}{\Delta t} \right) \frac{\mathbf{u}^{N,K+1} - \mathbf{u}^{N,K}}{\delta t}. \quad (70)$$

Equation (70) shows that the local truncation error of the PFE method depends on its inner integrator. Therefore, we also introduce the local truncation error of the inner integrator, defined as:

$$e_f^{N,k+1} = \frac{\tilde{\mathbf{f}}^{N,k+1} - \mathbf{f}^{N,k+1}}{\delta t}, \quad (71)$$

for which we now calculate an estimate. We rewrite both quantities $\tilde{\mathbf{f}}^{N,k+1}$ and $\mathbf{f}^{N,k+1}$ in terms of their respective solutions at time $t^{N,k}$. The numerical solution $\mathbf{f}^{N,k+1}$ is reformulated by recalling that we choose the FE scheme (37) as inner integrator (see section 4.2.1), and the fact that we require $\delta t = \varepsilon$ for stability (see section 4.2.1), giving:

$$\begin{aligned} \mathbf{f}^{N,k+1} &= S_{\delta t}(\mathbf{f}^{N,k}) = \mathbf{f}^{N,k} + \varepsilon \left(-D_{\mathbf{x},v}(\mathbf{f}^{N,k}) + \frac{1}{\varepsilon} (\mathcal{M}_v(\mathbf{u}^{N,k}) - \mathbf{f}^{N,k}) \right) \\ &= -\varepsilon D_{\mathbf{x},v}(\mathbf{f}^{N,k}) + \mathcal{M}_v(\mathbf{u}^{N,k}). \end{aligned} \quad (72)$$

The exact solution $\tilde{\mathbf{f}}^{N,k+1}$ is computed by applying a Taylor series expansion of $\tilde{\mathbf{f}}^{N,k+1}$ around $t^{N,k}$, yielding:

$$\begin{aligned} \tilde{\mathbf{f}}^{N,k+1} &= \tilde{\mathbf{f}}^{N,k} + \varepsilon \partial_t \tilde{\mathbf{f}}^{N,k} + O(\varepsilon^2) \\ &= \tilde{\mathbf{f}}^{N,k} + \varepsilon \left(-\widetilde{\mathbf{v} \partial_{\mathbf{x}} \mathbf{f}}^{N,k} + \frac{1}{\varepsilon} (\mathcal{M}_v(\mathbf{u}^{N,k}) - \tilde{\mathbf{f}}^{N,k}) \right) + O(\varepsilon^2) \\ &= S_{\delta t}(\tilde{\mathbf{f}}^{N,k}) + \varepsilon \left(D_{\mathbf{x},v}(\tilde{\mathbf{f}}^{N,k}) - \widetilde{\mathbf{v} \partial_{\mathbf{x}} \mathbf{f}}^{N,k} \right) + O(\varepsilon^2). \end{aligned} \quad (73)$$

The second equality is obtained by using the expression of the kinetic system (20), where $\widetilde{\mathbf{v} \partial_{\mathbf{x}} \mathbf{f}}^{N,k}$ denotes the evaluation of the convective derivative $v \partial_x f^\varepsilon$ at time $t^{N,k}$, which is subsequently collected on all discrete points x_i and v_j . The last equality follows from adding and subtracting $D_{\mathbf{x},v}(\tilde{\mathbf{f}}^{N,k})$ and subsequently using the forward Euler timestepper expression (37). Substituting equations (72) and (73) into (71), we find:

$$e_f^{N,k+1} = S_{\delta t}(e_f^{N,k}) + \left(D_{\mathbf{x},v}(\tilde{\mathbf{f}}^{N,k}) - \widetilde{\mathbf{v} \partial_{\mathbf{x}} \mathbf{f}}^{N,k} \right) + O(\varepsilon). \quad (74)$$

Working out the recursion in (74) gives:

$$e_f^{N,K+1} = \sum_{k=0}^K S_{\delta t}^k \left(D_{\mathbf{x},\mathbf{v}}(\tilde{\mathbf{f}}^{N,K-k}) - \widetilde{\mathbf{v}} \partial_{\mathbf{x}} \mathbf{f}^{N,K-k} \right) + O((K+1)\varepsilon). \quad (75)$$

Since the difference between brackets in equation (75) precisely corresponds to the spatial discretization error, we find the following estimate for the inner integrator local truncation error:

$$e_f^{N,K+1} = O((K+1)\Delta x^p) + O((K+1)\varepsilon), \quad (76)$$

where p denotes the order of accuracy of the spatial discretization method. Then, using equations (71) and (76), the local truncation error (70) of the PFE method is computed as:

$$\begin{aligned} E^{N+1} &= \frac{\tilde{\mathbf{u}}^{N+1} - \tilde{\mathbf{u}}^{N,K+1} + \varepsilon \langle e_f^{N,K+1} \rangle}{\Delta t} - \left(\frac{\Delta t - (K+1)\varepsilon}{\Delta t} \right) \frac{\tilde{\mathbf{u}}^{N,K+1} - \tilde{\mathbf{u}}^{N,K} - \varepsilon \langle e_f^{N,K+1} - e_f^{N,K} \rangle}{\varepsilon} \\ &= O(\Delta t) + O\left(\frac{\varepsilon^2}{\Delta t}\right) + \frac{\varepsilon}{\Delta t} \langle e_f^{N,K+1} \rangle + \left(\frac{\Delta t - (K+1)\varepsilon}{\Delta t} \right) \left(O(\Delta x^p) + O((K+1)\varepsilon) \right) \\ &= O(\Delta t) + O(\Delta x^p) + O((K+1)\varepsilon) + O\left(\frac{\varepsilon^2}{\Delta t}\right) + \frac{\varepsilon}{\Delta t} O(\Delta x^p), \end{aligned} \quad (77)$$

where we used Taylor expansions of all quantities $\tilde{\mathbf{u}}$ around t^N in the first equality.

Ultimately, we obtain the following expression for the local truncation error for PFE and FE as inner integrator:

$$E^{N+1} = O(\Delta t) + O(\Delta x^p) + O((K+1)\varepsilon) + \varepsilon O\left(\frac{\Delta x^p + \varepsilon}{\Delta t}\right). \quad (78)$$

It can then easily be shown, following the proof of [22, Theorem 5.1], that a projective Runge-Kutta method of order q has the following discretization error:

$$E^N = O(\Delta t^q) + O(\Delta x^p) + O((K+1)\varepsilon) + \varepsilon O\left(\frac{\Delta x^p + \varepsilon}{\Delta t}\right). \quad (79)$$

The first term in (79) is due to the time discretization error made in the outer Runge-Kutta integrator, whereas the next two terms are the space and time discretization error respectively due to the inner integrator. The last term results from the time derivative operator approximation. We remark that, in the limit of Δt going to 0, for fixed ε , this last term would result in divergence. However, since the goal is to create an asymptotic-preserving scheme, valid for fixed Δt (independent of ε), while ε tends to 0, this is not an issue: the last term then becomes of $O(\varepsilon)$, and hence negligible. Consequently, the scheme is indeed consistent and asymptotic-preserving in the limit $\varepsilon \rightarrow 0$.

Finally, we mention that the choice of ε in this work is determined by the finite difference approximation to the time derivative appearing in the projective Runge-Kutta formulations which have the following form:

$$\frac{\mathbf{f}^{n,K+1} - \mathbf{f}^{n,K}}{\varepsilon}. \quad (80)$$

As a consequence, the numerical error of this approximation is $O(\varepsilon)$ and is bounded below by $O(\sqrt{\varepsilon_{mach}})$ with $\varepsilon_{mach} \approx 10^{-16}$ being the machine precision. For that reason, we choose $\varepsilon = 10^{-8}$. Clearly, the above reasoning assumes implicitly that the distribution function f^ε is not very small or very large. This can always be realized by a proper non-dimensionalization of the system under study.

5 Applications

In this section, we illustrate the relaxation method with projective integration on a number of example systems. We first examine the one-dimensional case. In section 5.1, we consider the linear advection equation, and demonstrate the spatial and temporal order of the methods. Subsequently, we investigate nonlinear conservation laws: Burgers' equation in section 5.2 and the Euler equations (Sod's shock test) in section 5.3. Afterwards, we consider linear advection, the dam-break problem and the Euler equations in the two-dimensional case (sections 5.4–5.6).

5.1 Linear advection in 1D

Let us first illustrate the order of the relaxation method with projective integration, both in space and time. To this end, we consider the linear advection equation, that is, equation (3) with the linear flux function (26):

$$\partial_t u + \partial_x(a \cdot u) = 0, \quad (81)$$

in which the macroscopic unknown function $u(x, t)$ denotes the conserved quantity that propagates with velocity $a \in \mathbb{R}$. We compute the solution for $t \in [0, T]$ and $x \in [0, 1]$, using $a = 1$. We impose periodic boundary conditions and choose a smooth initial condition:

$$u(x, 0) = \exp(-100(x - 0.5)^2). \quad (82)$$

We compute the global error E^N at time $t^N = T$ which is defined as $E^N = \|\mathbf{E}^N\|$, with $\mathbf{E}^N = (E_i^N)_{i=1}^I$, and E_i^N denoting the global space-time discretization error at time $t = t^N$ and grid location $x = x_i$ given by:

$$E_i^N = \Delta x |\mathbf{u}_i^N - \mathbf{u}(x_i, t^N)|, \quad (83)$$

(For the linear advection equation (81), the exact solution is known analytically.) Here, we shall always choose the 1-norm to calculate the global error E^N .

For the relaxation method, we use the kinetic system (20), in which we discretize velocity space using $J = 2$ velocities. Taking into account the subcharacteristic condition (25), the velocities are fixed as: $v_1 = -|a|$ and $v_2 = |a|$. We point out that, in this purely academic first test case, we recover the Jin-Xin relaxation system (see also Remark 2.1), and additionally, the distribution corresponding to $v_1 = -|a|$ vanishes in the limit of ε tending to zero. As the Maxwellian, we choose (23), with $F(u) = a \cdot u$. The inner integrator is a space-time discretization of system (20), in which we choose the standard upwind spatial discretizations of order 1, 2 and 3 with grid spacing Δx (that will vary throughout the experiments), combined with a forward Euler time discretization with $\delta t = \varepsilon$ and $\varepsilon = 10^{-8}$. The projective integration method uses $K = 2$ inner steps, and an outer time step of size Δt (that will also vary).

Numerical spatial order (figures 4 and 5) To illustrate the spatial order of accuracy, we calculate the global error at time $T = 0.02$ and vary the grid spacing Δx as:

$$\Delta x = [0.04, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001]. \quad (84)$$

Correspondingly, depending on the spatial order of accuracy p and the temporal order of accuracy q of the outer integrator, we choose the outer time step Δt as:

$$\begin{cases} \Delta t = O(\Delta x^{p/q}) = C_p \Delta x^{p/q} & (p \geq q) \\ \Delta t = C_p \Delta x & (p < q). \end{cases} \quad (85)$$

The choice of outer time step for $p \geq q$ ensures that the temporal discretization error and the spatial discretization error display the same asymptotic behavior as Δx tends to zero, while for $p < q$ the spatial discretization error dominates in (79). The constants C_p in (85) should be chosen such that the projective integration method remains stable for all choices of Δx . Figure 4 shows the error as a function of Δx . In the left plot, time integration is done using the projective forward Euler method (PFE), for which the time order $q = 1$. The constants in (85) are then chosen as $C_1 = 0.5$, $C_2 = 20$ and $C_3 = 100$. We clearly observe the expected spatial order. This is confirmed by fitting a least squares line through the calculated error points. The slopes of these lines correspond to the numerical order which in this case were found to be 0.98, 1.94 and 2.99. These indeed lie sufficiently close to the expected spatial order.

In the middle plot, the experiment is repeated using a second-order projective Runge-Kutta method (PRK2, $q = 2$). For spatial orders $p \in \{1, 2\}$, we choose $\Delta t = 0.5\Delta x$. In that case, the first term in expression (79) will be dominant and the order in space can be observed. For $p = 3$ we put $\Delta t = C_3 \Delta x^{3/2}$ and choose $C_3 = 4$. On these plots, we observe that, for the third-order upwind discretization, the error curves start to level off for small values of Δx . This is due to the fact that the contribution of the spatial discretization error in (79) becomes negligible, and the $O(\varepsilon)$ term becomes dominant. As indicated in section 4.4, this term results from the time derivative approximation in the projective step by a finite

difference expression, see equation (80). When calculating the slopes of the least squares fit we now obtain 0.99, 1.99 and 3.07 which correspond to the expected spatial orders.

In the right plot, we use a second-order Runge-Kutta method as inner integrator. We note that, in agreement with the observations in section 4.2, the number K of inner steps can now no longer be chosen independently of ε . For stability, we are required to choose (at least) $K = 21$ for $p \in \{1, 2\}$ and $K = 22$ for $p = 3$. The numerical orders are 1.00, 1.99 and 3.04. Moreover, we again observe that the error levels off, since the remaining error is due to the finite difference approximation of the time derivative, and not due to the time discretization error of the inner integrator. Note that the error curve now levels off to a value which is about 10 times higher than in the forward Euler case, due to a less efficient damping of the fast eigenvalues, as can be seen in (63). This observation is supported by looking at equation (79) in which the value of K needs to be taken 10 times higher for RK2 as inner integrator so as to guarantee a stable functioning of the method. From these findings we conclude that it is not useful to select a higher-order inner integrator within the projective integration framework. Therefore, in what follows we will always select FE as inner integrator.

Next, we repeat this experiment using a fourth-order projective Runge-Kutta method (PRK4). We choose $\Delta t = C\Delta x$ for each $p \in \{1, 2, 3\}$ and put C equal to 0.4. The result is depicted on the left hand side plot of figure 5, where we again see the expected behavior (numerical orders: 0.99, 1.99 and 2.97). To avoid unphysical oscillations associated with higher-order upwind schemes, we also performed the same experiments using an essentially non-oscillatory (ENO) spatial discretization [32], which uses an adaptive stencil to avoid stencils with large variations in the solution values. The order test of the PRK4 scheme with ENO is shown on the right hand side plot of figure 5. The calculated numerical orders are 0.99, 1.86 and 2.86 which are in agreement with the expected spatial orders of the ENO scheme.

Numerical time order (figure 6) The temporal order of the projective integration methods is demonstrated in a slightly different manner as outlined above. Now, we fix the grid spacing $\Delta x = 0.05$ and vary the outer time step Δt as

$$\Delta t = [0.04, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002, 0.0001]. \quad (86)$$

We will calculate the error at time $T = 0.04$. The other simulation parameters remain the same as above. The error is now calculated by taking the 1-norm of the difference between the numerical solution and the analytical solution of the (linear) semi-discretized system (35). By doing so we take into account the discretization error in space such that we only look at the error in time. The simulations are run for PFE, PRK2 and PRK4 with FE as inner integrator and upwind differences of order 3 in space. The results can be seen in figure 6 in which we also look at the influence of the value of ε by choosing $\varepsilon = 10^{-5}$ (left hand side plot) and $\varepsilon = 10^{-8}$ (right hand side plot). It is clearly indicated on the plots that for small values of Δt the error curves level off towards the value of the dominant term of $O(\varepsilon)$ in expression (79) since the other terms in (79) are negligible. For $\varepsilon = 10^{-5}$ the numerical orders are 1.07 and 2.00 for PFE and PRK2, respectively. In this case there were too few meaningful points to reliably estimate the numerical order of the PRK4 method. For $\varepsilon = 10^{-8}$ the numerical orders are 1.02, 2.00 and 4.07, corresponding to the expected time order of the different methods.

5.2 Burgers' equation in 1D

As a second example, we consider the inviscid Burgers' equation in one spatial dimension,

$$\partial_t u + \partial_x \left(\frac{u^2}{2} \right) = 0. \quad (87)$$

We compute the solution for $t \in [0, 1]$ and $x \in [0, 2]$. We impose periodic boundary conditions and consider three different initial conditions: a Gaussian pulse $u_1(x)$, a sinc wave packet $u_2(x)$ and a sine wave $u_3(x)$ given by:

$$u_1(x) = \exp(-25(x-1)^2), \quad u_2(x) = \text{sinc}(5(x-1)), \quad u_3(x) = \sin(\pi x). \quad (88)$$

For the relaxation method, we use the kinetic system (20) together with the more realistic form of the Maxwellian given in (21) with $F(u) = u^2/2$. (Note that, compared to the previous example with

Order test (space) for linear advection

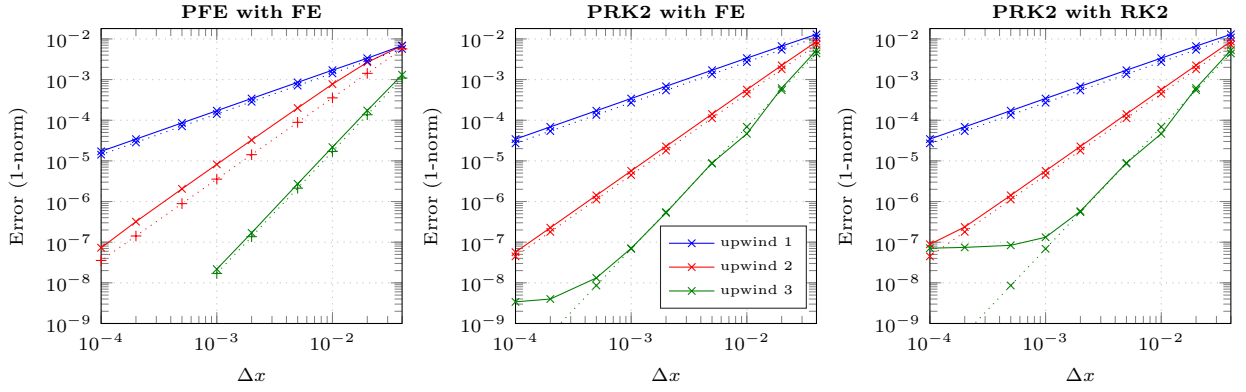


Figure 4: Spatial order test for PFE with FE as inner integrator (left) and PRK2 with FE (middle) and RK2 (right) as inner integrators and three different spatial orders. The error is computed using the 1-norm. On each plot, the solid lines represent the calculated error whereas the dotted line shows the expected error.

Order test (space) for linear advection: PRK4 + FE

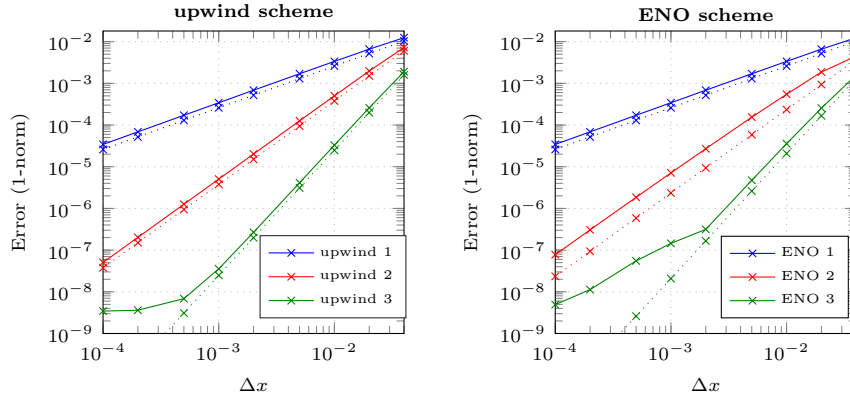


Figure 5: Spatial order test for PRK4 with FE as inner integrator and three different spatial orders using a) upwind differences (left plot) and b) the ENO scheme (right plot). The error is computed using the 1-norm. On each plot, the solid lines represent the calculated error whereas the dotted line shows the expected error.

Order test (time) for linear advection

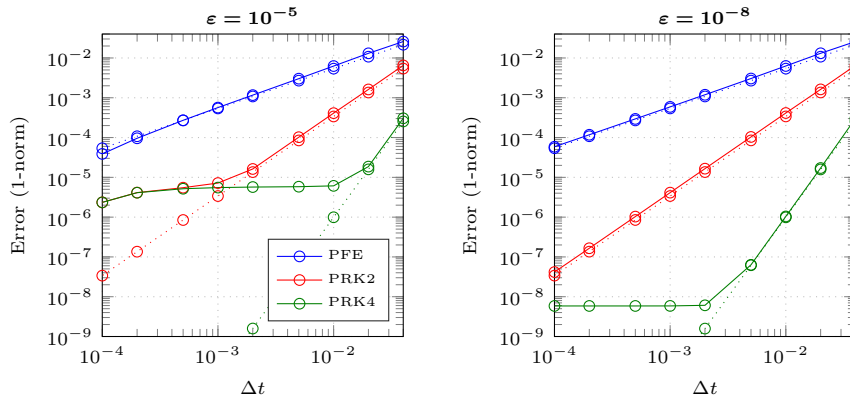


Figure 6: Temporal order test for the different projective integration methods under study: PFE, PRK2 and PRK4 with FE as inner integrator and upwind differences of order 3 in space comparing results for $\varepsilon = 10^{-5}$ (left plot) and $\varepsilon = 10^{-8}$ (right plot). The error is computed using the 1-norm. On each plot, the solid lines represent the calculated error whereas the dotted line shows the expected error.

linear advection, this only requires a change in one line of the code.) We discretize the velocity space using $J = 2$ velocities which are obtained as the nodes of Gauss-Hermite quadrature with $\sigma^2 = 1$ in equation (22). The inner integrator is a space-time discretization of system (20) using the standard upwind spatial discretizations of order 1, 2 and 3 with grid spacing Δx (that will vary throughout the experiments), combined with a forward Euler time discretization with $\delta t = \varepsilon$ and $\varepsilon = 10^{-8}$. We also consider the third-order ENO scheme. The projective integration method uses $K = 2$ inner steps, and an outer time step of size Δt (that will also vary).

We first perform a numerical simulation using a third-order ENO spatial discretization and a fourth-order projective Runge-Kutta method (PRK4) with $\Delta x = 10^{-2}$ and $\Delta t = 0.5\Delta x$. The results are depicted in figure 7. We clearly see that the discontinuities are nicely captured without the appearance of spurious oscillations.

Next, we also investigate the temporal order of the methods. Since the analytical solution of Burgers' equation is not available explicitly, the error is computed with respect to a reference solution that is obtained using a high-order simulation of (87) with the PRK4 method and FE as inner integrator using upwind differences of order 3 in space with grid spacing $\Delta x = 0.05$ and time step $\delta t = 10^{-8}$. The outer time step is chosen to be $\Delta t = 10^{-6}$. Then, we vary Δt as given by (86) and we again examine the influence of the value of ε by choosing $\varepsilon = 10^{-5}$ (left hand side plot) and $\varepsilon = 10^{-8}$ (right hand side plot) in figure 8. It is observed on the plots that for small values of Δt the error curves level off towards the value of the dominant term of $O(\varepsilon)$ in expression (79) since the other terms in (79) are negligible. For $\varepsilon = 10^{-5}$ the numerical orders are 1.07 and 2.00 whereas for $\varepsilon = 10^{-8}$ the numerical orders are 1.00, 2.00 and 4.18 which are in agreement with the expected time order of the different methods. For the temporal order test the initial solution consists of a Gauss curve centered around the middle of the domain $x \in [0, 2]$ (that is, function $u_1(x)$ in (88)) and the error is calculated at $t = 0.04$. The results are comparable to the linear advection case, see figure 8.

5.3 Sod's shock test in 1D

Sod's shock test is an important numerical test to check how a numerical method captures shock waves [33]. The test involves the Euler equations in one spatial dimension for mass, momentum and energy, $\mathbf{u} = (\rho, \rho\bar{v}, E)$,

$$\begin{cases} \partial_t \rho + \partial_x(\rho\bar{v}) = 0, \\ \partial_t(\rho\bar{v}) + \partial_x(\rho\bar{v}^2 + P) = 0, \\ \partial_t E + \partial_x((E + P)\bar{v}) = 0, \end{cases} \quad (89)$$

in which the pressure P is determined via the polytropic equation of state:

$$P = (\gamma - 1) \left(E - \frac{1}{2} \rho\bar{v}^2 \right), \quad (90)$$

in which the constant γ equals $7/5$ in case of a diatomic perfect gas [33].

Sod's shock test consists of an initial centered Riemann problem with the following left and right state values [33]:

$$\begin{pmatrix} \rho_L \\ \bar{v}_L \\ P_L \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \rho_R \\ \bar{v}_R \\ P_R \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0 \\ 0.1 \end{pmatrix}. \quad (91)$$

Furthermore, we impose outflow boundary conditions and perform simulation for $x \in [0, 1]$ and $t \in [0, 0.22]$. The particular choice of the time interval allows for a clear visualization of the three different characteristic waves (see below) and an easy comparison with the available literature.

For the relaxation method, we use the kinetic system (20), in which we discretize the velocity space with $J = 2$ velocities corresponding to the nodes of Gauss-Hermite quadrature. The Maxwellian is chosen as:

$$\mathcal{M}_v(\mathbf{u}^\varepsilon) = \begin{pmatrix} \rho \\ \rho\bar{v} \\ E \end{pmatrix} + \frac{v}{\langle v^2 \rangle} \begin{pmatrix} \rho\bar{v} \\ \rho\bar{v}^2 + P \\ E + P\bar{v} \end{pmatrix}. \quad (92)$$

The inner integrator uses a third-order spatial ENO discretization with $\Delta x = 5 \cdot 10^{-3}$ and a forward Euler time discretization with $\delta t = \varepsilon = 10^{-8}$. As the outer method, we choose the fourth-order projective Runge-Kutta method (PRK4), using $K = 2$ inner steps and an outer step of size $\Delta t = 0.5\Delta x$.

Evolution of Burgers' equation for several waves (PRK4 + FE + ENO3)

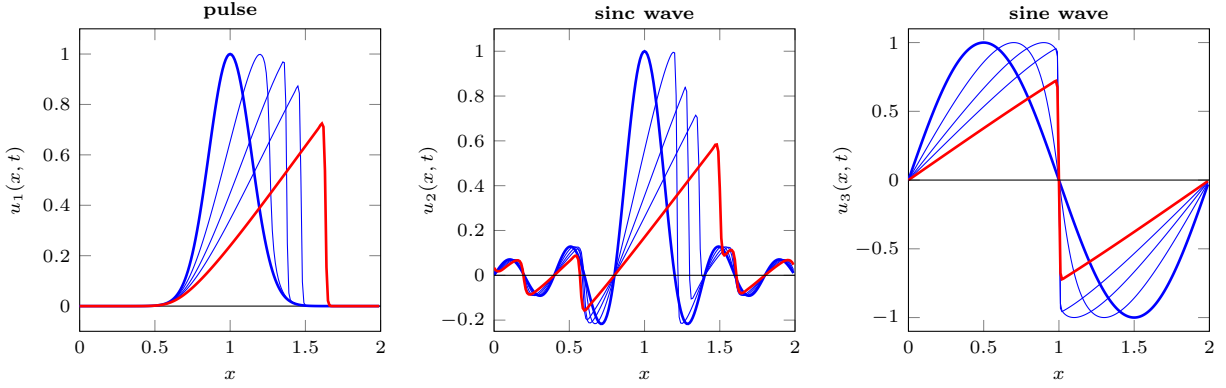


Figure 7: Numerical solution of Burgers' equation (87) obtained with PRK4 and FE as inner integrator using a third-order ENO scheme with $\Delta x = 10^{-2}$. The thick blue and red line represent the solution at $t = 0$ and $t = 1$.

Order test (time) for Burgers' equation

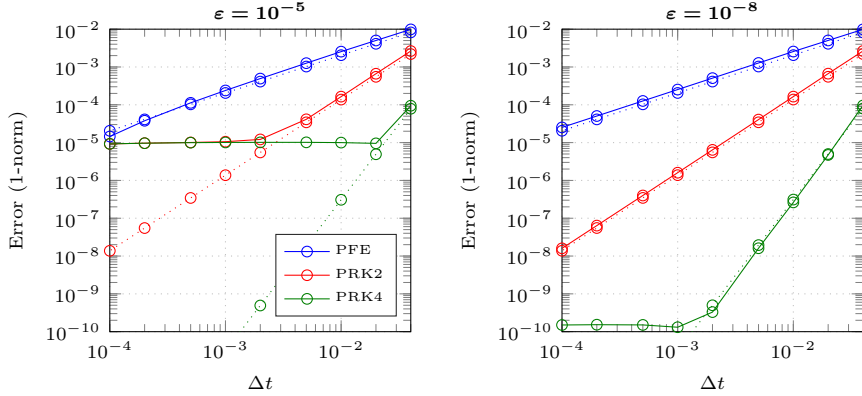


Figure 8: Temporal order test for Burgers' equation for the different projective integration methods under study: PFE, PRK2 and PRK4 with FE as inner integrator and upwind differences of order 3 in space comparing results for $\varepsilon = 10^{-5}$ (left plot) and $\varepsilon = 10^{-8}$ (right plot). The error is computed using the 1-norm. On each plot the solid lines represent the calculated error whereas the dotted line shows the expected error.

The result is illustrated in figure 9, where we plot density, velocity, energy and pressure at time $t = 0.22$, along with the analytical solution, calculated with an exact Riemann solver, for comparison.

We clearly see the development of three characteristic waves. The first wave corresponds to a rarefaction wave propagating to the left since initially the pressure and density on the left half side of the shock tube are higher than on the right side. Secondly, a contact discontinuity is observed. This right propagating wave corresponds to the initial discontinuity of (91). Finally, a shock wave propagating to the right has also appeared. It appears that only pressure and velocity remain continuous over the contact discontinuity and exhibit a very flat state. Furthermore, it can be observed that all the quantities are discontinuous over the shock wave. The projective integration method captures all these phenomena, without developing undesired oscillations and without smoothing out the discontinuities too much.

5.4 Linear advection in 2D

We now turn to problems in two spatial dimensions. We again start with the linear advection equation, which, in two dimensions, reads:

$$\partial_t u + a \partial_x u + b \partial_y u = 0, \quad (93)$$

in which $a, b \in \mathbb{R}$ are the constant advection speeds along the x - and y -direction, respectively. The macroscopic unknown function $u(x, y, t)$ denotes the two-dimensional conserved quantity. In the simula-

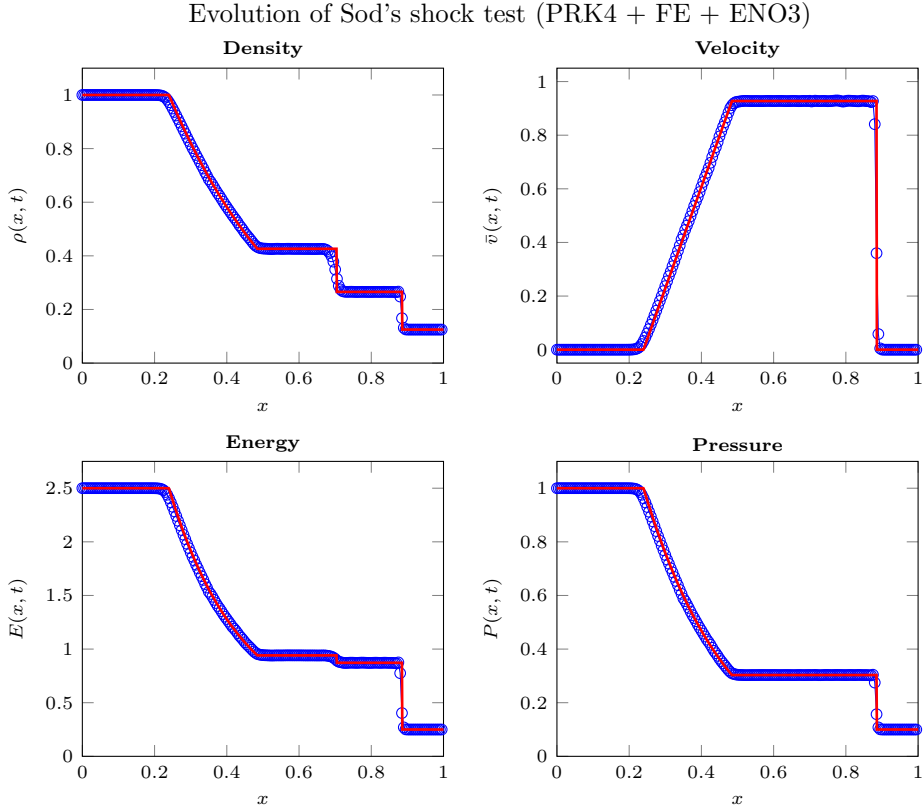


Figure 9: Fluid properties in Sod's shock test at $t = 0.22$ obtained with PRK4 and FE as inner integrator and a third-order ENO discretization in space using $\Delta x = 5 \cdot 10^{-3}$.

tions we integrate over $t \in [0, 1]$ and $(x, y) \in [0, 1]^2$, and we set $a = b = 1$. We impose periodic boundary conditions and start from a Gaussian pulse centered in the middle of the domain:

$$u(x, y, 0) = \exp(-50(x - 0.5)^2) \exp(-50(y - 0.5)^2). \quad (94)$$

As described in section 2.3 we now solve the two-dimensional kinetic system (31) with Maxwellian given by (33). The velocity discretization is determined by the orthogonal velocity method, in which we fix $R = S = 1$ (see remark 2.7). To satisfy the subcharacteristic condition, we calculate v_{\max} by choosing the (integer) lower bound from expression (34):

$$v_{\max} = \left\lceil \sqrt{\frac{12R^2(a^2 + b^2)}{(R+1)(2R+1)}} \right\rceil = 2.$$

The inner integrator is a space-time discretization of the kinetic system (31), in which we take the ENO scheme of order 1, 2 and 3 in space with $\Delta x = \Delta y = 0.04$ and the forward Euler scheme in time with $\delta t = \varepsilon = 10^{-8}$. The outer integrator is the fourth-order projective Runge-Kutta (PRK4) method, using $K = 2$ and $\Delta t = 0.3\Delta x$.

We compare the obtained numerical results for increasing order in space from 1 to 3. This is shown in figure 10.

5.5 Dam-break problem in 2D

In dam-break problems, one is interested in the evolution of two (or more) regions of water that are separated by a dam, which then is suddenly removed. Such problems are modeled by shallow water equations which, in general, describe the evolution and development of waves that are shallow with respect to their wavelength [26]. In two space dimensions, the shallow water equations take the following

2D linear advection (PRK4 + FE + ENO)

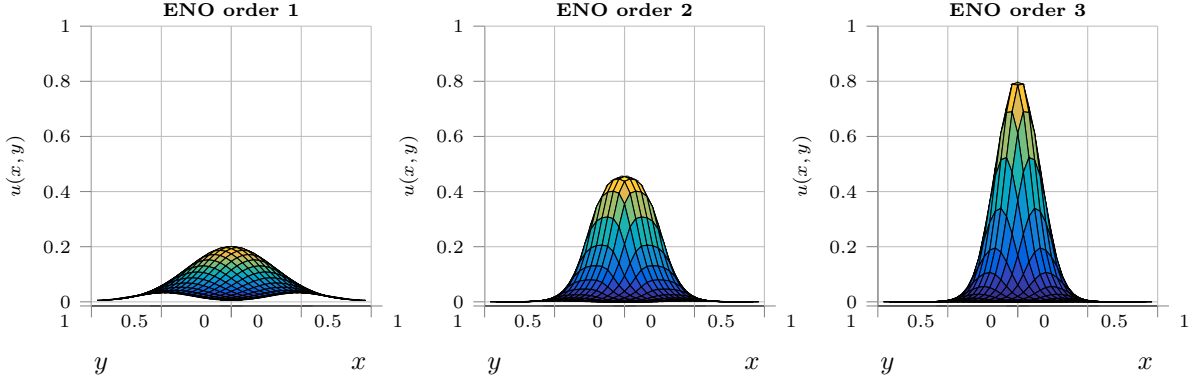


Figure 10: Comparison of the numerical solution for linear advection in 2D at $t = 1$ obtained by PRK4 and FE as inner integrator for different orders in space of the ENO scheme with $\Delta x = \Delta y = 0.04$.

form:

$$\begin{cases} \partial_t h + \partial_x(h\bar{v}^x) + \partial_y(h\bar{v}^y) = 0 \\ \partial_t(h\bar{v}^x) + \partial_x(h(\bar{v}^x)^2 + P) + \partial_y(h\bar{v}^x\bar{v}^y) = 0 \\ \partial_t(h\bar{v}^y) + \partial_x(h\bar{v}^x\bar{v}^y) + \partial_y(h(\bar{v}^y)^2 + P) = 0, \end{cases} \quad (95)$$

in which h is the water depth, $\bar{\mathbf{v}} = (\bar{v}^x, \bar{v}^y)$ is the (macroscopic) velocity vector, and P is the pressure. All these unknown functions depend on x, y and t . The system in (95) contains three equations for four unknown functions: h, \bar{v}^x, \bar{v}^y and P . It is closed by the following (hydrostatic) equation of state [26]:

$$P = \frac{1}{2}gh^2. \quad (96)$$

The shallow water system is considered over the domain $(x, y) \in [-2.5, 2.5]^2$ and $t \in [0, 1.5]$. We impose outflow boundary conditions and define the initial solution as a cylindrical basin of water surrounded by a dam (see [26]):

$$h(x, y, 0) = \begin{cases} 2, & x^2 + y^2 \leq 0.5 \\ 1, & \text{otherwise} \end{cases}, \quad \bar{\mathbf{v}}(x, y, 0) = \mathbf{0}. \quad (97)$$

For the relaxation method, we use the kinetic system (31) together with the Maxwellian given by (33), which now becomes:

$$\mathcal{M}_j(\mathbf{u}^\varepsilon) = \begin{pmatrix} h \\ h\bar{v}^x \\ h\bar{v}^y \end{pmatrix} + \frac{v_j^x}{\langle (\mathbf{v}^x)^2 \rangle} \begin{pmatrix} h\bar{v}^x \\ h(\bar{v}^x)^2 + P \\ h\bar{v}^x\bar{v}^y \end{pmatrix} + \frac{v_j^y}{\langle (\mathbf{v}^y)^2 \rangle} \begin{pmatrix} h\bar{v}^y \\ h\bar{v}^x\bar{v}^y \\ h(\bar{v}^y)^2 + P \end{pmatrix}. \quad (98)$$

The velocity discretization is determined by the orthogonal velocity method, in which we fix $R = S = 1$ and $v_{\max} = 2$. In the projective integration framework, the inner integrator is a space-time discretization of the kinetic equation (31), in which we take the WENO3 scheme in space with $\Delta x = \Delta y = 0.02$ and the forward Euler method in time with $\delta t = \varepsilon = 10^{-8}$. The outer integrator is the fourth-order projective Runge-Kutta (PRK4) method, using $K = 2$ and $\Delta t = 0.5\Delta x$. Due to the radial symmetry of the problem, in figure 11, we only plot the cross-section $h(x, 0, t)$ of the water depth. As in [26], we first observe an inflow of water, which eventually drops below $h = 1$, followed by a strong outflow around $t = 1$. After $t = 1$, two shock waves that weaken over time start propagating away from the origin. At $t = 1.5$, the water depth near the origin stabilizes around $h \approx 0.96$.

Shallow water equations in 2D (PRK4 + FE + WENO3)

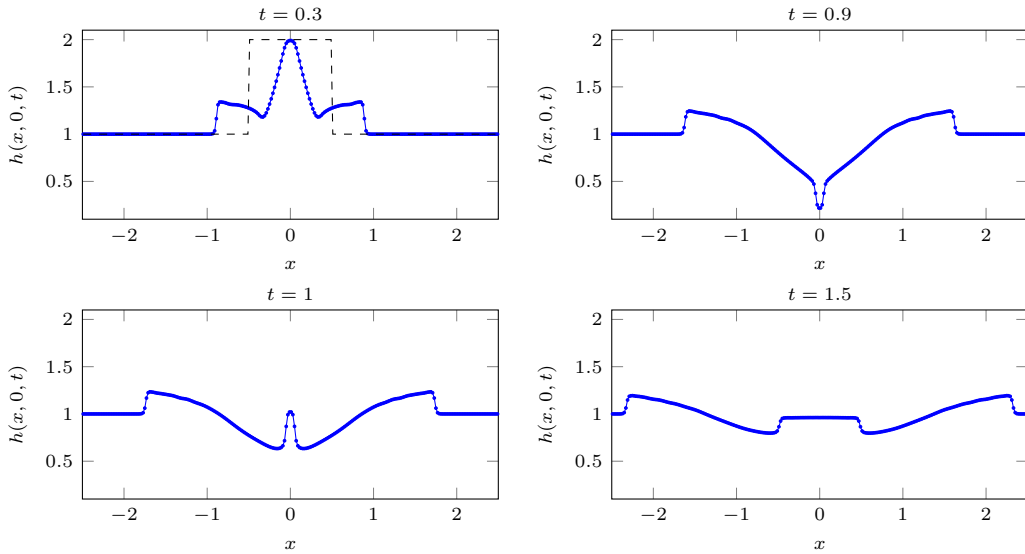


Figure 11: Time evolution of the water depth $h(x, 0, t)$ in the cylindrical dam-break problem obtained with PRK4 and FE as inner integrator using the WENO3 scheme with $\Delta x = \Delta y = 0.02$ and $\Delta t/\Delta x = 0.5$. The black dashed line (top left) represent the initial water depth in (97).

5.6 Euler equations in 2D

When extending the Euler equations from one to two spatial dimensions, we obtain a system of four equations, given by:

$$\begin{cases} \partial_t \rho + \partial_x(\rho \bar{v}^x) + \partial_y(\rho \bar{v}^y) = 0 \\ \partial_t(\rho \bar{v}^x) + \partial_x(\rho(\bar{v}^x)^2 + P) + \partial_y(\rho \bar{v}^x \bar{v}^y) = 0 \\ \partial_t(\rho \bar{v}^y) + \partial_x(\rho \bar{v}^x \bar{v}^y) + \partial_y(\rho(\bar{v}^y)^2 + P) = 0 \\ \partial_t E + \partial_x((E + P)\bar{v}^x) + \partial_y((E + P)\bar{v}^y) = 0. \end{cases} \quad (99)$$

In system (99) the unknown functions ρ , $\bar{\mathbf{v}} = (\bar{v}^x, \bar{v}^y)$, P and E all depend on x, y and t . Similarly to the one-dimensional case, we close the Euler system by the two-dimensional polytropic equation of state:

$$P = (\gamma - 1) \left(E - \frac{1}{2} \rho |\bar{\mathbf{v}}|^2 \right), \quad (100)$$

with $\gamma = 7/5$. We consider the Euler system over the domain $(x, y) \in [-0.5, 0.5]^2$ and $t \in [0, 0.16]$.

The extension of Sod's shock test (91) of Sod's shock test in 1D to two dimensions is given by (see [1]):

$$\begin{pmatrix} \rho_1 \\ \bar{\mathbf{v}}_1 \\ P_1 \end{pmatrix} = \begin{pmatrix} 0.1 \\ \mathbf{0} \\ 0.1 \end{pmatrix} \quad (xy \leq 0), \quad \text{and,} \quad \begin{pmatrix} \rho_2 \\ \bar{\mathbf{v}}_2 \\ P_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{0} \\ 1 \end{pmatrix} \quad (\text{otherwise}), \quad (101)$$

which is also called a double Sod tube. Furthermore, we impose outflow boundary conditions.

For the relaxation method and velocity discretization, we use the same setup as in section 5.5. In the projective integration framework, the inner integrator is a space-time discretization of the kinetic system (31), in which we take the WENO3 scheme in space with $\Delta x = \Delta y = 0.01$ and the forward Euler scheme in time with $\delta t = \varepsilon = 10^{-8}$. The outer integrator is the fourth-order projective Runge-Kutta (PRK4) method, using $K = 2$ and $\Delta t = 0.4\Delta x$. The results can be seen in figure 12 and correspond to those in [1].

6 Conclusions

We presented a general, high-order, fully explicit, relaxation scheme for systems of nonlinear hyperbolic conservation laws in multiple dimensions, by approximating the nonlinear hyperbolic conservation law by

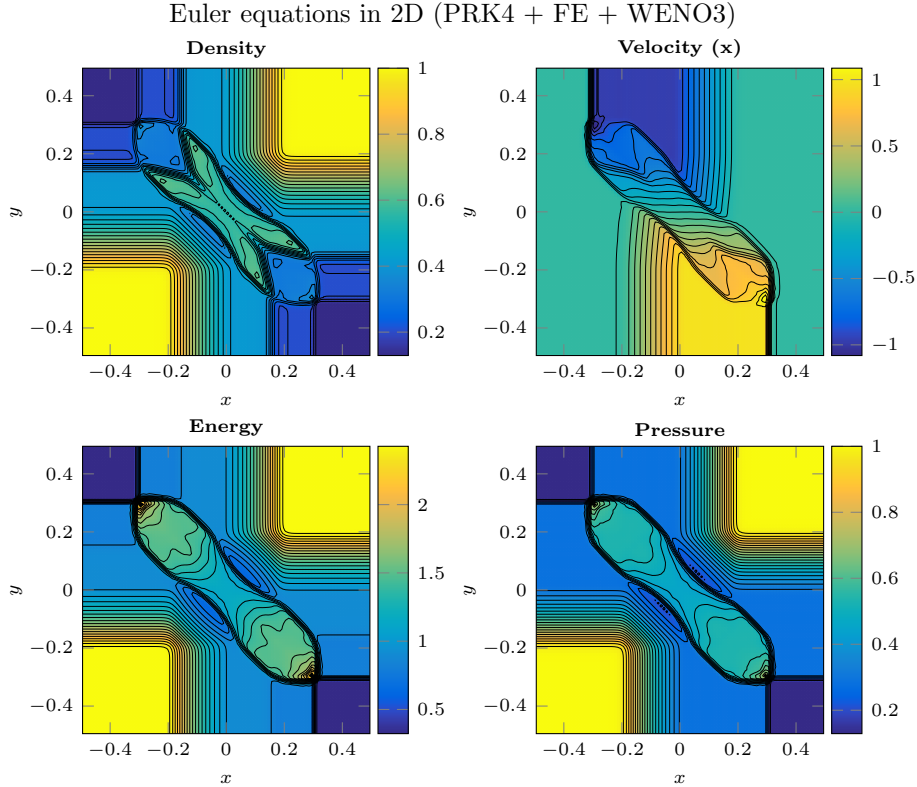


Figure 12: Contour plots of the numerical solution of Sod's double shock tube problem at $t = 0.16$ obtained with PRK4 and FE as inner integrator using the WENO3 spatial discretization with $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.4\Delta x$.

a kinetic equation with BGK source term, which is, in turn, discretized and integrated using a projective integration method. After taking a few small (inner) steps with the direct forward Euler method, an estimate of the time derivative is used in an (outer) Runge-Kutta method of arbitrary order.

Unlike other methods based on relaxation [1, 19], the projective integration method does not rely on a splitting technique, but only on an appropriate selection of time steps using a naive explicit discretization method. Its main advantage is its generality and ease of use: implementing the method for a different system of hyperbolic conservation laws only requires changing the definition of the Maxwellian.

We showed that, with an appropriate choice of inner step size, the time step restriction on the outer time step is similar to the CFL condition for the hyperbolic conservation law. Moreover, the number of inner time steps is also independent of the scaling parameter. We analyzed stability and consistency, and illustrated with numerical results on a set of test problems of varying complexity.

For future work, it is interesting to extend and analyze the proposed technique to quasilinear systems of conservation laws and hyperbolic equations with source terms.

References

- [1] 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.
- [2] Christophe Besse, Saja Borghol, Thierry Goudon, Ingrid Lacroix-Violet, and Jean Paul Dudon. Hydrodynamic regimes, knudsen layer, numerical schemes: Definition of boundary fluxes. *Advances in Applied Mathematics and Mechanics*, 3(5):519–561, 2011.

- [3] Prabhu Lal Bhatnagar, Eugene P. Gross, and Max Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3), 1954.
- [4] Alexander Bobylev and Niclas Bernhoff. Discrete velocity models and dynamical systems. *Series on Advances in Mathematics for Applied Sciences*, 63:203–222, 2003.
- [5] Sebastiano Boscarino, Lorenzo Pareschi, and Giovanni Russo. Implicit-explicit Runge–Kutta schemes for hyperbolic systems and kinetic equations in the diffusion limit. *SIAM Journal on Scientific Computing*, 35(1):22–51, 2013.
- [6] Sebastiano Boscarino and Giovanni Russo. On A Class Of Uniformly Accurate IMEX Runge–Kutta Schemes And Applications To Hyperbolic Systems With Relaxation. *SIAM Journal on Scientific Computing*, 31(3):1926–1945, 2009.
- [7] François Bouchut. Construction of BGK models with a family of kinetic entropies for a given system of conservation laws. *Journal of Statistical Physics*, 95(1-2):113–170, 1999.
- [8] Jaime Cisternas, C. William Gear, Simon A. Levin, and Ioannis G. Kevrekidis. Equation-free modeling of evolving diseases: Coarse-grained computations with individual-based models. *Proceedings of the Royal Society of London A: Mathematical, Physical & Engineering Sciences*, 460:2761–2779, 2004.
- [9] Petra Csomós and István Faragó. Error analysis of the numerical solution of split differential equations. *Mathematical and Computer Modelling*, 48(7-8):1090–1106, 2008.
- [10] Giacomo Dimarco and Lorenzo Pareschi. Asymptotic-Preserving Implicit-Explicit Runge-Kutta Methods for Nonlinear Kinetic Equations. *SIAM Journal on Numerical Analysis*, 51(2):1064–1087, 2013.
- [11] Weinan E and Bjorn Engquist. The Heterogeneous Multiscale Methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.
- [12] Kenneth Eriksson, Claes Johnson, and Anders Logg. Explicit Time-Stepping for Stiff ODEs. *SIAM Journal on Scientific Computing*, 25(4):1142–1157, 2004.
- [13] Michael Fey, Rolf Jeltsch, Jochen Maurer, and Anne-Thérèse Morel. The method of transport for nonlinear systems of hyperbolic conservation laws in several space dimensions. Technical report, 1997.
- [14] Francis Filbet and Shi Jin. A class of asymptotic-preserving schemes for kinetic equations and related problems with stiff sources. *Journal of Computational Physics*, 229(20):7625–7648, 2010.
- [15] C. William Gear and Ioannis G. Kevrekidis. Projective Methods for Stiff Differential Equations: Problems with Gaps in Their Eigenvalue Spectrum. *SIAM Journal on Scientific Computing*, 24(4):1091–1106, 2003.
- [16] Ernst Hairer, Syvert Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I*. Springer Berlin Heidelberg, 1993.
- [17] Shi Jin. Efficient Asymptotic-Preserving (AP) Schemes For Some Multiscale Kinetic Equations. *SIAM Journal on Scientific Computing*, 21(2):441–454, 1999.
- [18] Shi Jin, Lorenzo Pareschi, and Giuseppe Toscani. Diffusive relaxation schemes for multiscale discrete-velocity kinetic equations. *SIAM Journal on Numerical Analysis*, 35(6):2405–2439, 1998.
- [19] 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.
- [20] Ioannis G. Kevrekidis, C. William Gear, James M. Hyman, Panagiotis G. Kevrekidis, Olof Runborg, and Constantinos Theodoropoulos. Equation-Free, Coarse-Grained Multiscale Computation: enabling microscopic simulators to perform system-level tasks. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

- [21] Ioannis G. Kevrekidis and Giovanni Samaey. Equation-free multiscale computation: algorithms and applications. *Annual review of physical chemistry*, 60:321–344, 2009.
- [22] Pauline Lafitte, Annelies Lejon, and Giovanni Samaey. A High-Order Asymptotic-Preserving Scheme for Kinetic Equations Using Projective Integration. *SIAM Journal on Numerical Analysis*, 54(1):1–33, 2016.
- [23] Pauline Lafitte and Giovanni Samaey. Asymptotic-preserving projective integration schemes for kinetic equations in the diffusion limit. *SIAM Journal on Scientific Computing*, 34(2):579–600, 2012.
- [24] Steven L. Lee and C. William Gear. Second-order accurate projective integrators for multiscale problems. *Journal of Computational and Applied Mathematics*, 201(1):258–274, 2007.
- [25] Mohammed Lemou and Florian Méhats. Micro-macro schemes for kinetic equations including boundary layers. *SIAM Journal on Scientific Computing*, 34(6):B734–B760, 2012.
- [26] Randall J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge University press, 2002.
- [27] Tai-Ping Liu. Hyperbolic Conservation Laws with Relaxation. *Communications in Mathematical Physics*, 108:153–175, 1987.
- [28] Luc Mieussens. Discrete Velocity Model and Implicit Scheme for the BGK Equation of Rarefied Gas Dynamics. *Mathematical Models and Methods in Applied Sciences*, 10(8):1121–1149, 2000.
- [29] Keith W. Morton and David Francis Mayers. *Numerical solution of partial differential equations: an introduction*. Cambridge University press, 2005.
- [30] Ramiro Rico-Martínez, C. William Gear, and Ioannis G. Kevrekidis. Coarse projective kMC integration: forward/reverse initial and boundary value problems. *Journal of Computational Physics*, 196(2):474–489, 2004.
- [31] Simas Setayeshgar, C. William Gear, Hans Othmer, and Ioannis G. Kevrekidis. Application of coarse integration to bacterial chemotaxis. *Multiscale Modeling & Simulation*, 4(1):307–327, 2005.
- [32] Chi-Wang Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. Technical report, NASA/CR-97-206253, ICASE Report No. 97-65, 1998.
- [33] Gary A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, 1978.
- [34] Ben P. Sommeijer. Increasing the real stability boundary of explicit methods. *Computers & Mathematics with Applications*, 19(6):37–49, 1990.
- [35] Christophe Vandekerckhove, Dirk Roose, and Kurt Lust. Numerical stability analysis of an acceleration scheme for step size constrained time integrators. *Journal of Computational and Applied Mathematics*, 200(2):761–777, 2007.
- [36] Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*. Springer-Verlag, Berlin, 1991.
- [37] Gerald Beresford Whitham. *Linear and nonlinear waves*. Wiley Interscience, 1974.
- [38] Susanne Amy Zimmerman. The method of transport for the Euler equations written as a kinetic scheme. In *Hyperbolic Problems: Theory, Numerics, Applications*, pages 999–1008. Springer, 1999.