Noname manuscript No.
(will be inserted by the editor)

Space-time Generalized Riemann Problem solvers of order k for linear advection with unrestricted time step

Christophe Berthon · Céline Sarazin · Rodolphe Turpault

Received: date / Accepted: date

Abstract This work concerns high-order approximations of the linear advection equation in very long time. A GRP-type scheme of arbitrary high-order in space and time with no restriction on the time step is developed. In the usual GRP solvers, we consider a polynomial approximation of the solution in space in each cell at the initial time. Here, we add a second polynomial approximation of the solution in time in each interface. Thanks to this double approximation, the resulting scheme is compact. It is proved to be of order $k+1$ in space and time, where k is the degree of the polynomials. Thanks to the compactness of the scheme, a two-dimensional extension is detailed on unstructured meshes made of triangles. Several numerical test-cases and comparison with existing methods illustrate the excellent behaviour of the scheme.

Keywords GRP solvers · linear systems of conservation laws · arbitrary high-order scheme · large time steps

1 Introduction

This work is devoted to the derivation of a high-order numerical scheme able to suitably approximate the long-time solutions of linear hyperbolic systems of conservation laws. Such systems can be written in the following form:

$$\partial_t w + \nabla_{\mathbf{x}} M w = 0, \tag{1}$$

where $w(\mathbf{x}, t) \in \mathbb{R}^n$ is the vector of unknowns and M is a constant matrix which is diagonalizable in $\mathbb{R}$. Here, we are interested in approximations of the solutions after a very long time with respect

C. Berthon
Université de Nantes, Laboratoire de Mathématiques Jean Leray, UMR6629
2 rue de la Houssinière, BP 92208, 44322 Nantes cedex 3, France Tel.: +33251125920
E-mail: christophe.berthon@univ-nantes.fr

C. Sarazin
Tel.: +33251125953
E-mail: celine.sarazin@univ-nantes.fr

R. Turpault
Tel.: +33251125984
E-mail: rodolphe.turpault@univ-nantes.fr

to the characteristic speeds of the system. After such a long time, the classical schemes [35,41] produce solutions which are completely smeared, unless an unreasonably fine grid is considered.

Yet, several applications of physical interest involve coupling between phenomena that occur at very different speeds. An important example concerns kinetic equations such as Boltzmann [24], Fokker-Planck [29], neutronics [14] and radiative transfer equations [40,39,10], (see [15,21, 22,34] for numerous examples). Usual approximations of these equations, for instance discrete ordinates or spherical harmonics decomposition methods [14], lead to systems of the form:

$$\partial_t w + \nabla_{\mathbf{x}} M w = S(w)$$

where $S(w)$ is a source term. These systems may involve drastically different characteristic speeds from one another, for example the fluid speed versus the speed of the light. Limited by the fastest phenomenon, the CFL condition for an explicit numerical approximation for coupled problems may become too restrictive. Thanks to asymptotic-preserving schemes [26,9,8,32], suitable numerical schemes can be written for this system as soon as an efficient scheme for the advection part is known. It is therefore crucial to use a reliable implicit numerical method for (1) in coupling contexts.

During the last decades, several classes of very efficient numerical schemes were developed for hyperbolic systems of conservation laws. Some of these methods, such as Discontinous Galerkin [11,17,16,20,23], ENO/WENO [28,36] and distributive schemes [3,1], have been successfully used in a wide range of applications (see also [31,2,18,38,12,19,30] for a non-exhaustive list). Generalized Riemann Problem (GRP) methods [4,25] are an interesting alternative here since arbitrary high-order explicit GRP schemes can be written by taking advantage of the linear structure of our problem.
However, all these schemes, which were designed for more general (nonlinear) systems, are explicit or involve very sophisticated techniques [42,33,37]. Therefore, due to the CFL condition, their computational cost becomes prohibitive when considering very long times.

The aim of this work is thus to introduce an efficient high-order scheme for linear advection systems which works for any time step. In practice, involving the transformation matrix P, the system (1) can often be transformed into a decoupled system of equations for the variable $u = P^{-1}w$. Therefore, for the sake of simplicity, this article focuses on numerical approximations of the linear advection equation Cauchy problem:

$$\begin{cases} \partial_t u(\mathbf{x}, t) + a\boldsymbol{\Omega}.\nabla_{\mathbf{x}} u(\mathbf{x}, t) = 0, \\ u(\mathbf{x}, t = 0) = u_0(\mathbf{x}), \end{cases} \tag{2}$$

where $a \in \mathbb{R}$ is the speed, and $\boldsymbol{\Omega} = \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix}$ is the normalized direction of the velocity.

The paper is organized as follows. The next section is concerned with the one-dimensional case. First, the principle of the GRP methods is recalled. In particular, arbitrary high-order explicit GRP schemes are built in the linear case. The space-time extension is then introduced. The resulting scheme can be defined with no restriction on the time step. It is also proved to be of order k in space and time for arbitrary large k. In the following section, a two-dimensional extension is developed on unstructured meshes composed of triangles again for an unrestricted time increment. It is also proved to be of order k in space and time. Throughout this paper, the excellent behaviour of the scheme is illustrated on numerical test-cases. The scheme is also compared with usual high-order methods.

2 Presentation of the one-dimensional scheme

2.1 Generalized Riemann Problem

For the sake of completeness, we recall first the principle of the Generalized Riemann Problem method. This method was introduced by Ben-Artzi and Falcovitz in [4] and several extensions have been derived (see for instance [5,25,6,7]). In this paragraph, we recall the construction of a GRP high-order space discretization for linear systems. Indeed, all computations can be carried on due to the linearity of the system.
We consider the equation:

$$\partial_t u + a\partial_x u = 0, \tag{3}$$

where $u(x,t) \in \mathbb{R}$ and the speed a is assumed to be positive. The computations for $a < 0$ are similar and won't be recalled here.
We present the method on a structured mesh with a constant space step Δx and a constant time step Δt. The mesh is composed of the cells $C_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ the centers of which are x_i. Next we introduce the set of functions:

$$\mathbb{P}^k_C = \{u \in L^\infty(\mathbb{R})/\ u_{|C_i} \in \mathbb{R}^i_k, \forall i \in \mathbb{Z}, \},$$

where $\mathbb{R}^i_k$ is the set of polynomials of degree k per cell:

$$\mathbb{R}^i_k = \{u_{|C_i} \in \mathbb{R}_k[X]\}.$$

In each cell C_i, we consider the following canonical basis:

$$\left\{ \left(\frac{x - x_i}{\Delta x} \right)^j \right\}_{j=0,\ldots,k}.$$

This choice of basis was made to simplify the presentation, but another basis such as the Legendre polynomials may be considered for example. The associated dot product on the cell C_i is defined as follows:

$$< f, g >_i = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x)g(x)\ dx,$$

and the associated norm is:

$$\|f\|_i = \sqrt{< f, f >_i}.$$

As opposed to usual techniques like MUSCL or WENO for instance, the GRP scheme only involves two steps: exact evolution of a piecewise polynomial approximation and $\mathbb{P}^k$-projection over each cell of the exactly evolved solution.
Considered approximation: First, we consider an initial data $u^0 \in \mathbb{P}^k_C$. At time $t = t^n$, we assume that an approximation of the solution $u^n(x) \in \mathbb{P}^k_C$ is known. Then we have $u^n_i(x) \in \mathbb{R}^i_k$ for $x \in C_i$ and $i \in \mathbb{Z}$. This approximation is now evolved to define an approximated solution $u^{n+1}(x) \in \mathbb{P}^k_C$ at time $t = t^n + \Delta t$.

Evolution step: The approximation $u^n(x)$ is exactly evolved during a time step Δt which satisfies the CFL condition $\lambda = a\frac{\Delta t}{\Delta x} < 1$. Thanks to the linearity of the equation, this exact solution $u^h(x, t^n + \Delta t)$ is easily determined by adopting the characteristics method (see Fig 1).

$$u^h_i(x, t^n + \Delta t) = \begin{cases} u^n_{i-1}(x - a\Delta t), & \text{if } \dfrac{x - x_{i-\frac{1}{2}}}{\Delta t} < a, \\ u^n_i(x - a\Delta t), & \text{if } \dfrac{x - x_{i-\frac{1}{2}}}{\Delta t} > a. \end{cases}$$

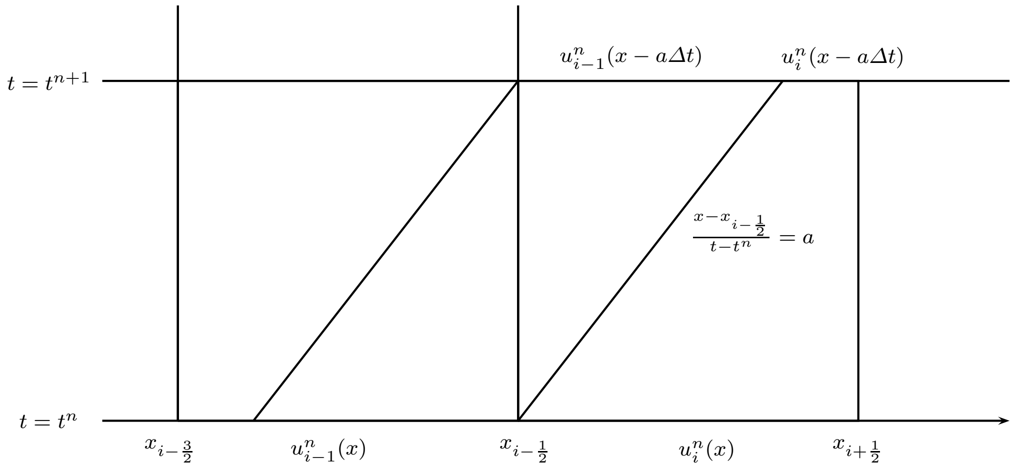

Fig. 1 Exact evolution of the solution thanks to the characteristics method with $a > 0$

Projection step: The exact solution $u^h(x, t^n + \Delta t)$ is projected on the space $\mathbb{P}_C^k$. The updated solution in the cell C_i is therefore the solution of the minimization problem:

$$\|u_i^{n+1} - u_i^h(t^n + \Delta t)\|_i = \inf_{p \in \mathbb{P}_C^k} \|p - u_i^h(t^n + \Delta t)\|_i. \tag{4}$$

Thanks to the Petrov-Galerkin conditions, we can rewrite this minimization problem (4) as:

$$< u_i^{n+1} - u_i^h(t^n + \Delta t), p >_i = 0 \quad \forall p \in \mathbb{P}_C^k. \tag{5}$$

As usual, the Petrov-Galerkin conditions rewrite as a linear system. Indeed, using a decomposition of u_i^{n+1} in the following form:

$$u_i^{n+1}(x) = \sum_{j=0}^{k} \alpha_i^{n+1,j} \left(\frac{x - x_i}{\Delta x} \right)^j,$$

the projection step then consists in solving the following linear system of size $k+1$:

$$\sum_{j=0}^{k} \alpha_i^{n+1,j} < \left(\frac{x - x_i}{\Delta x} \right)^j, \left(\frac{x - x_i}{\Delta x} \right)^l >_i = < u_i^h(t^n + \Delta t), \left(\frac{x - x_i}{\Delta x} \right)^l >_i \quad \forall l = 0, \ldots, k. \tag{6}$$

The left hand side in (6) may be rewritten as:

$$\begin{aligned} \sum_{j=0}^{k} \alpha_i^{n+1,j} < \left(\frac{x - x_i}{\Delta x} \right)^j, \left(\frac{x - x_i}{\Delta x} \right)^l >_i &= \sum_{j=0}^{k} \alpha_i^{n+1,j} \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\frac{x - x_i}{\Delta x} \right)^{j+l} dx \\ &= \sum_{j=0}^{k} \alpha_i^{n+1,j} \frac{1}{j+l+1} \left(\left(\frac{1}{2} \right)^{j+l+1} - \left(\frac{-1}{2} \right)^{j+l+1} \right) \\ &= \sum_{j=0}^{k} \alpha_i^{n+1,j} A_i(j,l). \end{aligned}$$

The right hand side in (6) becomes when $a > 0$:

$$
\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{\Delta x}\right)^l >_i =& \sum_{j=0}^{k} \alpha_{i-1}^{n,j} \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}+a\Delta t} \left(\frac{x-a\Delta t - x_{i-1}}{\Delta x}\right)^j \left(\frac{x-x_i}{\Delta x}\right)^l \, dx \\
&+ \alpha_i^{n,j} \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}+a\Delta t}^{x_{i+\frac{1}{2}}} \left(\frac{x-a\Delta t - x_i}{\Delta x}\right)^j \left(\frac{x-x_i}{\Delta x}\right)^l \, dx \\
=& \sum_{j=0}^{k} \alpha_{i-1}^{n,j} I_i^-(j,l) + \alpha_i^{n,j} I_i^+(j,l) \\
=& \, b_i^{n+1}(l).
\end{aligned}
$$

The coefficients $I_i^-(j,l)$ and $I_i^+(j,l)$ can be recursively computed thanks to successive integrations by parts:

$$
\begin{aligned}
I_i^-(j,l) &= \frac{1}{j+1}\left(\frac{1}{2}\right)^{j+l+1} \left[(-1+2\lambda)^l - (1-2\lambda)^{j+1}\right] - \frac{l}{j+1} I_i^-(j+1,l-1), \\
\text{with} \;\; I_i^-(j+l,0) &= \frac{1}{j+l+1}\left(\frac{1}{2}\right)^{j+l+1} \left[1 - (1-2\lambda)^{j+l+1}\right], \\
I_i^+(j,l) &= \frac{1}{j+1}\left(\frac{1}{2}\right)^{j+l+1} \left[(1-2\lambda)^{j+1} - (-1+2\lambda)^l\right] - \frac{l}{j+1} I_i^+(j+1,l-1), \\
\text{with} \;\; I_i^+(j+l,0) &= \frac{1}{j+l+1}\left(\frac{1}{2}\right)^{j+l+1} \left[(1-2\lambda)^{j+l+1} - (-1)^{j+l+1}\right].
\end{aligned}
$$

Finally, since we have a polynomial per cell approximation of the solution at time $t = t^n$, the scheme simply consists in solving a linear system of size $k+1$ for each cell C_i of the mesh:

$$A_i \alpha_i^{n+1} = b_i^{n+1}, \tag{7}$$

to obtain the updated approximation at time t^{n+1}. Moreover, the approximation $u_i^{n+1}(x)$ of the solution is in $\mathbb{P}_C^k$ by construction. We can note that the matrix A_i of the system (7) is the same for each cell, and does not depend on time, so it has to be computed only once for the whole domain.

The resulting scheme is obviously of order $k+1$ in space. Using the same idea, an scheme of order $k+1$ may be constructed for $\lambda > 1$ though it would require an extended stencil which depends on the time step.

2.2 Space-time Generalized Riemann Problem

For sake of simplicity, the term *implicit* will abusively be used to refer to a situation where $\lambda > 1$ even though the definition of the scheme does not involve the resolution of a nonlinear system. Similarly, the term *explicit* will be used to refer to situations where $\lambda \leq 1$.

In order to obtain a compact implicit scheme, we now consider an extension of the GRP method which can deal with unrestricted time steps. We still consider the equation (3). With the assumptions on the mesh used previously, we denote by $I^{n+\frac{1}{2}} = (t^n, t^{n+1})$ the interface between two time steps and by $K_i^n = C_i \times I^{n+\frac{1}{2}}$ the space-time control volumes.

The scheme consists in the usual: exact evolution of high-order approximations and projection steps.

Considered approximations and notations: The initial solution is once again assumed to be a polynomial approximation in space at time t^n of the solution of degree k; $u^n(x) \in \mathbb{P}_C^k$. Now, we also consider a polynomial approximation of the solution in time on each interface: for $t^n \leq t \leq t^{n+1}, u(x_{i-\frac{1}{2}}, t) \simeq v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t) \in \mathbb{R}_k^n$. The space $\mathbb{R}_k^n$ is defined by:

$$\mathbb{R}_k^n = \{u_{|_{I^{n+\frac{1}{2}}}} \in \mathbb{R}_k[X]\},$$

and we define the set of polynomials per interface:

$$\mathbb{P}_I^k = \{u \in L^\infty(\mathbb{R})/ \ u_{|_{I^{n+\frac{1}{2}}}} \in \mathbb{R}_k^n, \forall n \in \mathbb{N}\}.$$

On each interface $I^{n+\frac{1}{2}}$, we consider the canonical basis:

$$\left\{ \left(\frac{t - t^{n+\frac{1}{2}}}{\Delta t} \right)^r \right\}_{r=0,\ldots,k}.$$

Any other suitable basis of $\mathbb{R}_k^n$ may be considered if necessary. The associated dot product on the interface $I^{n+\frac{1}{2}}$ is defined as:

$$< f, g >^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f(t) g(t) \ dt.$$

Evolution step: We let the solution evolve in an exact way until the next time step. The main difference with the GRP is that we have two approximations of the solution: one on the cells and one on the interfaces. If $a > 0$, the solution at time t^{n+1} and on the interface $x_{i+\frac{1}{2}} \times I^{n+\frac{1}{2}}$ is fully determined by the values of u_i^n and $v_{i-\frac{1}{2}}^{n+\frac{1}{2}}$. If $a < 0$, the solution at time t^{n+1} and on the interface $x_{i-\frac{1}{2}} \times I^{n+\frac{1}{2}}$ is fully determined by the values of u_i^n and $v_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ (see Fig 2). This implies that the scheme will be compact, since all the information required to compute the solution lies inside a single cell. Thanks to the linearity of the equation, the exact evolution of the solution is easy to compute. When $a > 0$ and $\lambda = \frac{a\Delta t}{\Delta x} < 1$, the exact solutions are:

$$u_i^h(x, t^n + \Delta t) = \begin{cases} v_{i-\frac{1}{2}}^{n+\frac{1}{2}} \left(t^{n+1} - \frac{x - x_{i-\frac{1}{2}}}{a} \right), & \text{if } (x - x_{i-\frac{1}{2}}) < a\Delta t \\ u_i^n(x - a\Delta t), & \text{otherwise,} \end{cases}$$

$$v_h^{n+\frac{1}{2}}(t, x_{i+\frac{1}{2}}) = u_i^n(x_{i+\frac{1}{2}} - a(t - t^n)).$$

and when $\lambda > 1$, we have:

$$u_i^h(x, t^n + \Delta t) = v_{i-\frac{1}{2}}^{n+\frac{1}{2}} \left(t^{n+1} - \frac{x - x_{i-\frac{1}{2}}}{a} \right),$$

$$v_h^{n+\frac{1}{2}}(t, x_{i+\frac{1}{2}}) = \begin{cases} u_i^n(x_{i+\frac{1}{2}} - a(t - t^n)), & \text{if } (t - t^n) > \frac{\Delta x}{a} \\ v_{i-\frac{1}{2}}^{n+\frac{1}{2}} \left(t - \frac{\Delta x}{a} \right), & \text{otherwise.} \end{cases}$$

As a convention, we will speak of explicit case when $\lambda < 1$ and implicit case when $\lambda > 1$, although the scheme will be able to handle both of them similarly.

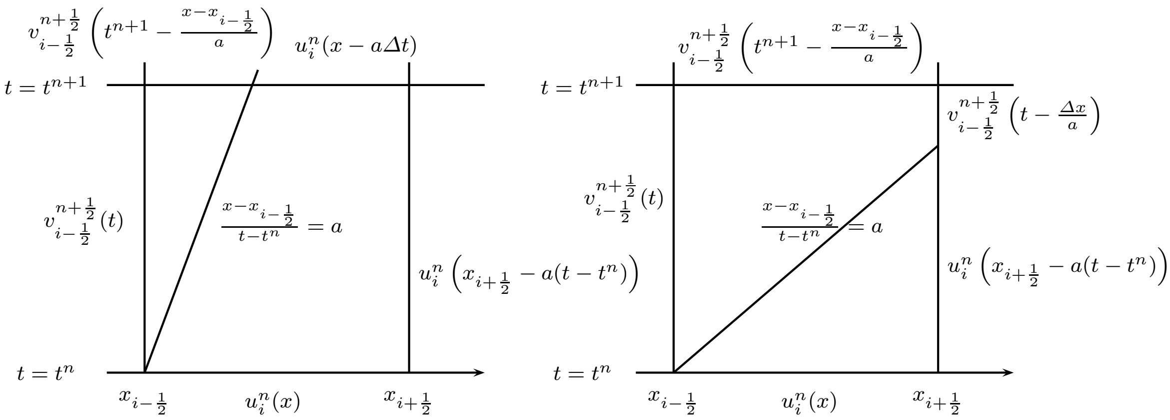

Fig. 2 Exact solutions on the volume K_i^n in explicit (left) and implicit (right) case

Projection step: The exact solutions are projected onto the spaces $\mathbb{P}_C^k$ and $\mathbb{P}_I^k$ respectively. The exact solution is projected on $\mathbb{P}_C^k$ in the cell C_i at time t^{n+1} and the exact solution is projected onto $\mathbb{P}_I^k$ on the interface $x_{i+\frac{1}{2}}\times(t^n,t^{n+1})$. This leads to two minimization problems:

$$\|u_i^{n+1}-u_i^h(t^n+\Delta t)\|_i=\inf_{p\in\mathbb{P}_C^k}\|p-u_i^h(t^n+\Delta t)\|_i, \tag{8}$$

$$\|v_{i+\frac{1}{2}}^{n+\frac{1}{2}}-v_h^{n+\frac{1}{2}}(x_{i+\frac{1}{2}})\|^n=\inf_{q\in\mathbb{P}_I^k}\|q-v_h^{n+\frac{1}{2}}(x_{i+\frac{1}{2}})\|^n. \tag{9}$$

Thanks to the Petrov-Galerkin conditions, we can rewrite (8)-(9) as:

$$< u_i^{n+1}-u_i^h(t^n+\Delta t),\left(\frac{x-x_i}{\Delta x}\right)^l >_i=0 \quad \forall l=0,\dots,k, \tag{10}$$

$$< v_{i+\frac{1}{2}}^{n+\frac{1}{2}}-v_h^{n+\frac{1}{2}}(x_{i+\frac{1}{2}}),\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r >^n=0 \quad \forall r=0,\dots,k. \tag{11}$$

Using the decomposition of the solutions in their corresponding basis:

$$u_i^{n+1}(x)=\sum_{j=0}^{k}\alpha_i^{n+1,j}\left(\frac{x-x_i}{\Delta x}\right)^j,$$

$$v_{i+\frac{1}{2}}^{n+\frac{1}{2}}(t)=\sum_{s=0}^{k}\beta_{i+\frac{1}{2}}^{n+\frac{1}{2},s}\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^s,$$

the minimization problems (10)-(11) are equivalent to solving the two following linear systems of size $k+1$:

$$\sum_{j=0}^{k} \alpha_i^{n+1,j} < \left(\frac{x-x_i}{\Delta x}\right)^j, \left(\frac{x-x_i}{\Delta x}\right)^l >_i = < u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{\Delta x}\right)^l >_i \quad \forall l=0,\dots,k, \tag{12}$$

$$\sum_{s=0}^{k} \beta_{i+\frac{1}{2}}^{n+\frac{1}{2},s} < \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^s, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r >^n = < v_h^{n+\frac{1}{2}}(x_{i+\frac{1}{2}}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r >^n \quad \forall r=0,\dots,k. \tag{13}$$

The left hand sides in (12)-(13) have the same form independently of λ:

$$\begin{aligned}
< \left(\frac{x-x_i}{\Delta x}\right)^j, \left(\frac{x-x_i}{\Delta x}\right)^l >_i =& \frac{1}{j+l+1}\left(\frac{1}{2}\right)^{j+l+1}\left[1-(-1)^{j+l+1}\right] \\
=& A_u(j,l), \\
< \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^s, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r >^n =& \frac{1}{s+r+1}\left(\frac{1}{2}\right)^{s+r+1}\left[1-(-1)^{s+r+1}\right] \\
=& A_v(s,r).
\end{aligned}$$

The right hand sides in (12)-(13) will have two expressions depending on the value of λ. If we assume $a>0$, we have in the explicit case:

$$\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{\Delta x}\right)^l >_i =& \sum_{j=0}^{k} \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},j} \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i-\frac{1}{2}}+a\Delta t} \left(\frac{t^{n+1}-\frac{x-x_{i-\frac{1}{2}}}{a}-t^{n+\frac{1}{2}}}{\Delta t}\right)^j \left(\frac{x-x_i}{\Delta x}\right)^l dx \\
&+ \sum_{j=0}^{k} \alpha_i^{n,j} \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}+a\Delta t}^{x_{i+\frac{1}{2}}} \left(\frac{x-a\Delta t-x_i}{\Delta x}\right)^j \left(\frac{x-x_i}{\Delta x}\right)^l dx \\
=& \sum_{j=0}^{k} \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},j} J_{ex}^1(j,l) + \alpha_i^{n,j} J_{ex}^2(j,l) \\
=& b_u(l), \\
< v_h^{n+\frac{1}{2}}(x_{i+\frac{1}{2}}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r >^n =& \sum_{s=0}^{k} \alpha_i^{n,s} \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \left(\frac{1}{2}-\frac{a(t-t^n)}{\Delta x}\right)^s \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r dt \\
=& \sum_{s=0}^{k} \alpha_i^{n,s} K_{ex}(s,r) \\
=& b_v(r),
\end{aligned}$$

where the coefficients J_{ex}^1, J_{ex}^2 and K_{ex} can be recursively computed:

$$\begin{aligned}
J_{ex}^1(j,l) =& \frac{1}{l+1}\left[\left(\frac{-1}{2}\right)^j\left(\lambda-\frac{1}{2}\right)^{l+1} - \left(\frac{1}{2}\right)^j\left(\frac{-1}{2}\right)^{l+1}\right] + \frac{j}{\lambda(l+1)}J_{ex}^1(j-1,l+1),\\
J_{ex}^2(j,l) =& \frac{1}{l+1}\left[\left(\frac{1}{2}\right)^{l+1}\left(\frac{1}{2}-\lambda\right)^j - \left(\lambda-\frac{1}{2}\right)^{l+1}\left(\frac{-1}{2}\right)^j\right] - \frac{j}{l+1}J_{ex}^2(j-1,l+1),\\
K_{ex}(s,r) =& \frac{1}{r+1}\left[\left(\frac{1}{2}-\lambda\right)^s\left(\frac{1}{2}\right)^{r+1} - \left(\frac{1}{2}\right)^s\left(\frac{-1}{2}\right)^{r+1}\right] + \frac{s\lambda}{r+1}K_{ex}(s-1,r+1).
\end{aligned}$$

For the implicit case, we have:

$$\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{\Delta x}\right)^l >_i =& \sum_{j=0}^k \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},j}\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\left(\frac{t^{n+1}-\frac{x-x_{i-\frac{1}{2}}}{a}-t^{n+\frac{1}{2}}}{\Delta t}\right)^j\left(\frac{x-x_i}{\Delta x}\right)^l dx\\
=& \sum_{j=0}^k \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},j} J_{im}(j,l)\\
=& b_u(l),\\
< v_h^{n+\frac{1}{2}}(x_{i+\frac{1}{2}}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r >^n =& \sum_{s=0}^k \alpha_i^{n,s}\frac{1}{\Delta t}\int_{t^n}^{t^n+\frac{\Delta x}{a}}\left(\frac{x_{i+\frac{1}{2}}-a(t-t^n)-x_i}{\Delta x}\right)^s\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r dt\\
&+ \sum_{s=0}^k \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},s}\frac{1}{\Delta t}\int_{t^n+\frac{\Delta x}{a}}^{t^{n+1}}\left(\frac{t-\frac{\Delta x}{a}-t^{n+\frac{1}{2}}}{\Delta t}\right)^s\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^r dt\\
=& \sum_{s=0}^k \alpha_i^{n,s}K_{im}^1(s,r) + \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},s}K_{im}^2(s,r)\\
=& b_v(r),
\end{aligned}$$

where the coefficients J_{im}, K_{im}^1 and K_{im}^2 can be recursively calculated:

$$\begin{aligned}
J_{im}(j,l) =& \frac{1}{l+1}\left[\left(\frac{1}{2}-\frac{1}{\lambda}\right)^j\left(\frac{1}{2}\right)^{l+1} - \left(\frac{1}{2}\right)^j\left(\frac{-1}{2}\right)^{l+1}\right] + \frac{j}{\lambda(l+1)}J_{im}(j-1,l+1),\\
K_{im}^1(s,r) =& \frac{1}{r+1}\left[\left(\frac{-1}{2}\right)^s\left(\frac{1}{\lambda}-\frac{1}{2}\right)^{r+1} - \left(\frac{1}{2}\right)^s\left(\frac{-1}{2}\right)^{r+1}\right] + \frac{s\lambda}{r+1}K_{im}^1(s-1,r+1),\\
K_{im}^2(s,r) =& \frac{1}{r+1}\left[\left(\frac{1}{2}-\frac{1}{\lambda}\right)^s\left(\frac{1}{2}\right)^{r+1} - \left(\frac{-1}{2}\right)^s\left(\frac{1}{\lambda}-\frac{1}{2}\right)^{r+1}\right] + \frac{s}{r+1}K_{im}^2(s-1,r+1).
\end{aligned}$$

The computations are similar in the case $a < 0$.
In practice, the method can be summarized in the following way:

- At the initial time, the solution $u_i^0(x)$ is known inside each cell, and the solution $v^{\frac{1}{2}}(t)$ is known on the corresponding boundary of the domain (left if $a > 0$, right if $a < 0$).
- The mesh is ran from the left to the right if $a > 0$, from the right to left otherwise.

- Inside each volume K_i^n, two linear systems of size $k+1$ have to be solved to compute the updates $u_i^{n+1}(x)$ and $v_{i+\frac{1}{2}}^{n+\frac{1}{2}}(t)$ if $a>0$, $v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t)$ if $a<0$:

$$A_u \alpha_i^{n+1} = b_u, \tag{14}$$
$$A_v \beta_{i+\frac{1}{2}}^{n+\frac{1}{2}} = b_v. \tag{15}$$

At the end of the iteration, the two approximations $u^{n+1}(x)$ and $v^{n+\frac{1}{2}}(t)$ of the solution are still in their respective spaces $\mathbb{P}_C^k$ and $\mathbb{P}_I^k$ by construction. Since the matrices A_u of the system (14) and A_v of the system (15) do not depend on space and time, they have to be computed and inverted only once.

Example When $k=0$, *i.e.* the constant case, we have:

$$u_i^n(x) = \alpha_i^n \ \in \mathbb{R},$$
$$v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t) = \beta_{i-\frac{1}{2}}^{n+\frac{1}{2}} \ \in \mathbb{R},$$

and the corresponding schemes are:

$$\text{Explicit } (\lambda \le 1)\text{:} \begin{cases} v_{i+\frac{1}{2}}^{n+\frac{1}{2}} = & u_i^n, \\ u_i^{n+1} = & \lambda v_{i-\frac{1}{2}}^{n+\frac{1}{2}} + (1-\lambda) u_i^n, \end{cases}$$
$$\text{Implicit}(\lambda > 1)\text{:} \begin{cases} u_i^{n+1} = & v_{i-\frac{1}{2}}^{n+\frac{1}{2}}, \\ v_{i+\frac{1}{2}}^{n+\frac{1}{2}} = & \frac{1}{\lambda} u_i^n + (1-\frac{1}{\lambda}) v_{i-\frac{1}{2}}^{n+\frac{1}{2}}. \end{cases}$$

We note that the resulting scheme is nothing but the upwind scheme.

When $k=1$, we have:

$$u_i^n(x) = \alpha_i^{n,0} + \alpha_i^{n,1} x,$$
$$v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t) = \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},0} + \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},1} t,$$

and the corresponding explicit scheme reads:

$$\begin{cases} \alpha_i^{n+1,0} = & \alpha_i^{n,0}(1-\lambda) - \frac{\lambda}{2}(1-\lambda)\alpha_i^{n,1} + \lambda \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},0}, \\ \alpha_i^{n+1,1} = & 6\lambda(1-\lambda)[\alpha_i^{n,0} - \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},0}] + (1-\lambda)(1-2\lambda-2\lambda^2)\alpha_i^{n,1} - \lambda^2 \beta_{i-\frac{1}{2}}^{n+\frac{1}{2},1}, \\ \beta_{i+\frac{1}{2}}^{n+\frac{1}{2},0} = & \alpha_i^{n,0} + \frac{1-\lambda}{2}\alpha_i^{n,1}, \\ \beta_{i+\frac{1}{2}}^{n+\frac{1}{2},1} = & -\lambda \alpha_i^{n,0}. \end{cases}$$

It is to note that this explicit scheme turns out to be equivalent to the one developed by Dai and Woodward in [13]. Their scheme, based on moment conservation for non linear equations is however only defined in the explicit case.

In addition to be rather simple to write, we can prove that the scheme we developed is conservative and of order $k+1$ in space and in time.

Theorem 1 *The scheme is conservative on its first moment $\alpha^{n,0}$ and exactly preserves the polynomials of degree k.*

Let us emphasize that the scheme is only conservative with respect to its first moment. Therefore, if it is used in a finite volumes context, the higher order moments have to be understood as high-order corrections.

Proof By definition of the scheme:

$$\begin{aligned}\sum_i \alpha_i^{n+1,0} &= \int_{\mathbb{R}} u^h(x,t^{n+1})dx,\\ &= \int_{\mathbb{R}} u^h(x-a\Delta t,t^n)dx = \sum_i \alpha_i^{n,0},\end{aligned}$$

and therefore the scheme is conservative on its first moment.
To simplify the calculations, we suppose that $a>0$ and use an induction procedure.
We first consider the explicit scheme.

- At time $t=0$, the initial solution is polynomial: $u_0(x) \in \mathbb{R}_k[X]$.
- At time $t=t^n$, we suppose that the solution given by the scheme is the exact solution: $u^n(x)=u_0(x-at^n)$. In particular, $u^n(x)\in\mathbb{R}_k[X]$.
  We want to prove that the exact solution is still preserved by the scheme at time $t=t^n+\Delta t$.
  - Considered approximation: inside each cell C_i of the domain at time t^n, we have $u_i^n(x) = u^n(x)|_{C_i}$, which is the projection of $u^n(x)$ onto the space $\mathbb{P}_C^k$. Since $u^n(x)$ is a polynomial of degree k, its projection $\{u_i^n(x)\}_{i\in\mathbb{Z}}$ onto the space $\mathbb{P}_C^k$ is the exact polynomial written in the corresponding basis.
  - Evolution step: this step consists in doing an exact evolution of the solution. If we focus on the volume K_i^n, the exact solution on the left interface $x_{i-\frac{1}{2}}$ is:

    $$v_h^{n+\frac{1}{2}}(t,x_{i-\frac{1}{2}}) = u_{i-1}^n(x_{i-\frac{1}{2}}-a(t-t^n)) \in \mathbb{R}_k[X],$$

    and on the right interface $x_{i+\frac{1}{2}}$:

    $$v_h^{n+\frac{1}{2}}(t,x_{i+\frac{1}{2}}) = u_{i}^n(x_{i+\frac{1}{2}}-a(t-t^n)) \in \mathbb{R}_k[X].$$

    The exact solution in the cell C_i is:

    $$u_i^h(x,t^n+\Delta t) = \begin{cases} v_{i-\frac{1}{2}}^{n+\frac{1}{2}}\left(t^{n+1}-\frac{x-x_{i-\frac{1}{2}}}{a}\right), & \text{if } (x-x_{i-\frac{1}{2}})<a\Delta t,\\ u_i^n(x-a\Delta t), & \text{otherwise.}\end{cases}$$

  - Projection step: the exact solutions are projected onto the corresponding space $\mathbb{P}^k$. On the left interface $x_{i-\frac{1}{2}}$, since $v_h^{n+\frac{1}{2}}(x_{i-\frac{1}{2}}) \in \mathbb{R}_k[X]$, the projection onto $\mathbb{P}_I^k$ is exact. Therefore the exact solution is a polynomial of degree k. Its projection in $\mathbb{P}_I^k$ is hence the same polynomial. So the solution on the interface $x_{i-\frac{1}{2}}$ is the exact solution:

    $$v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t) = v_h^{n+\frac{1}{2}}(t,x_{i-\frac{1}{2}}) = u_{i-1}^n(x_{i-\frac{1}{2}}-a(t-t^n)).$$

    Consequently:

    $$u_i^h(x,t^n+\Delta t) = \begin{cases} u_{i-1}^n(x-a\Delta t), & \text{if } (x-x_{i-\frac{1}{2}})<a\Delta t,\\ u_i^n(x-a\Delta t), & \text{otherwise.}\end{cases} = u_i^h(x-a\Delta t,t^n)$$

Since $u_i^h(t^n) \in \mathbb{R}_k[X]$, $u_i^h(t^n+\Delta t) \in \mathbb{R}_k[X]$ and the projection onto $\mathbb{P}_C^k$ is once again exact.

$$u_i^{n+1}(x) = u_i^h(x, t^n+\Delta t) = u^n(x - a\Delta t)|_{C_i}.$$

- Finally, we have $u^{n+1}(x) = u^n(x - a\Delta t) = u_0(x - at^{n+1})$, so the explicit scheme gives the exact solution if $u_0 \in \mathbb{R}_k[X]$.

Then we have the implicit case, where we recall that $a > 0$.

- At time $t = 0$, we assume that the initial and left boundary solutions are polynomials of degree k: $u_0(x) \in \mathbb{R}_k[X]$ and $v_L(t) \in \mathbb{R}_k[X]$.
- At time $t = t^n$, we suppose that the solution $u^n(x)$ given by the scheme is the exact solution: $u^n(x) = u_0(x - at^n) \in \mathbb{R}_k[X]$.
  We want to prove that this exact solution is still preserved by the scheme at time $t = t^n + \Delta t$.
  - Considered approximation: inside each cell C_i of the domain at time t^n, we have $u_i^n(x) = u^n(x)|_{C_i}$, which is the projection of $u^n(x)$ onto the space $\mathbb{P}_C^k$. Since $u^n(x)$ is a polynomial of degree k, its projection $\{u_i^n(x)\}_{i\in\mathbb{Z}}$ onto the space $\mathbb{P}_C^k$ is the exact polynomial. On the left boundary, we have $v_{\frac{1}{2}}^{n+\frac{1}{2}}(t) = v_L(t)|_{I^{n+\frac{1}{2}}}$, the exact solution.
  - We focus on the first volume $K_1^{n+\frac{1}{2}}$.

    - Evolution step: the exact solution on the cell C_1 is:

    $$u_1^h(x, t^n+\Delta t) = \ v_{\frac{1}{2}}^{n+\frac{1}{2}}\left(t^{n+1} - \frac{x - x_{\frac{1}{2}}}{a}\right) = v_L\left(t^{n+1} - \frac{x - x_{\frac{1}{2}}}{a}\right)\Big|_{I^{n+\frac{1}{2}}},$$

    and on the interface $x_{\frac{3}{2}}$:

    $$v_h^{n+\frac{1}{2}}(t, x_{\frac{3}{2}}) = \begin{cases} u_1^n(x_{\frac{3}{2}} - a(t-t^n)), & \text{if } (t-t^n) < \frac{\Delta x}{a}, \\ v_{\frac{1}{2}}^{n+\frac{1}{2}}\left(t - \frac{\Delta x}{a}\right), & \text{otherwise.} \end{cases}$$

    - Projection step: the exact solutions are projected onto the corresponding spaces $\mathbb{P}^k$. On the cell C_1, since $v_{\frac{1}{2}}^{n+\frac{1}{2}}(t) \in \mathbb{R}_k[X]$, the projection onto $\mathbb{P}_C^k$ is exact:

    $$u_1^{n+1}(x) = u_1^h(x, t^n+\Delta t) = v_L\left(t^{n+1} - \frac{x - x_{\frac{1}{2}}}{a}\right)$$

    Consequently, on the interface $x_{\frac{3}{2}}$ we have:

    $$v_h^{n+\frac{1}{2}}(t, x_{\frac{3}{2}}) = \begin{cases} v_{\frac{1}{2}}^{n-\frac{1}{2}}\left(t - \frac{\Delta x}{a}\right), & \text{if } (t-t^n) < \frac{\Delta x}{a}, \\ v_{\frac{1}{2}}^{n+\frac{1}{2}}\left(t - \frac{\Delta x}{a}\right), & \text{otherwise} \end{cases} = \begin{cases} v_L(t - \frac{\Delta x}{a})\big|_{I^{n-\frac{1}{2}}}, & \text{if } (t-t^n) < \frac{\Delta x}{a}, \\ v_L(t - \frac{\Delta x}{a})\big|_{I^{n+\frac{1}{2}}}, & \text{otherwise.} \end{cases}$$

    Since $v_L(t) \in \mathbb{R}_k[X]$, $u_1^h(x, t^n+\Delta t) \in \mathbb{R}_k[X]$ and the projection onto $\mathbb{P}_I^k$ is once again exact.
  - We suppose now that we calculated the exact solutions $u_p^{n+1}(x)$ and $v_{p+\frac{1}{2}}^{n+\frac{1}{2}}(t)$ on the volumes $\{K_p^n\}_{p=1,\ldots,i-1}$. We focus on the volume K_i^n.

- Evolution step: the exact solution on the cell C_i at time t^{n+1} is:

$$u_i^h(x, t^n + \Delta t) = \; v_{i-\frac{1}{2}}^{n+\frac{1}{2}} \left(t^{n+1} - \frac{x - x_{i-\frac{1}{2}}}{a} \right),$$

and on the interface $x_{i+\frac{1}{2}}$:

$$v_h^{n+\frac{1}{2}}(t, x_{i+\frac{1}{2}}) = \begin{cases} u_i^n(x_{i+\frac{1}{2}} - a(t - t^n)), & \text{if } (t - t^n) < \frac{\Delta x}{a}, \\ v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t - \frac{\Delta x}{a}), & \text{otherwise.} \end{cases}$$

- Projection step: we project the exact solution in the corresponding spaces $\mathbb{P}^k$. On the cell C_i, since $v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t) \in \mathbb{R}_k[X]$, the projection onto $\mathbb{P}_C^k$ is exact:

$$u_i^{n+1}(x) = u_i^h(x, t^n + \Delta t) = v_{i-\frac{1}{2}} \left(t^{n+1} - \frac{x - x_{\frac{1}{2}}}{a} \right)$$

Since we have the exact solutions on the interfaces $\{x_{p+\frac{1}{2}}\}_{p=1,\dots,i-1}$, we have:

$$u_i^h(x, t^n + \Delta t) = \begin{cases} u_{i-1}^n(x - a\Delta t), & \text{if } x > x_{i-\frac{1}{2}} + a\Delta t - \Delta x, \\ u_{i-2}^n(x - a\Delta t), & \text{if } x_{i-\frac{1}{2}} + a\Delta t - \Delta x > x > x_{i-\frac{1}{2}} + a\Delta t - 2\Delta x \\ \vdots & \\ u_{i-p}^n(x - a\Delta t), & \text{if } x_{i-\frac{1}{2}} + a\Delta t - (p-1)\Delta x > x > x_{i-\frac{1}{2}} + a\Delta t - p\Delta x \\ u_{i-p-1}^n(x - a\Delta t), & \text{if } x_{i-\frac{1}{2}} + a\Delta t - p\Delta x > x > x_{i-\frac{1}{2}} + a\Delta t - (p+1)\Delta x \\ v_{i-p-\frac{3}{2}}^{n+\frac{1}{2}} \left(t^{n+1} - \frac{x - x_{i-p-\frac{3}{2}}}{a} \right), & \text{otherwise.} \end{cases}$$

If we define $j = int\left(\frac{a\Delta t}{\Delta x}\right)$, since we are on the cell C_i, we have:

$$\begin{aligned} u_i^h(x, t^n + \Delta t)\big|_{C_i} &= \begin{cases} u_{i-j}^n(x - a\Delta t), & \text{if } x > x_{i-\frac{1}{2}} + a\Delta t - j\Delta x \\ u_{i-j-1}^n(x - a\Delta t), & \text{if } x_{i-\frac{1}{2}} + a\Delta t - j\Delta x > x \end{cases} \\ &= u^n(x - a\Delta t)\big|_{C_i}. \end{aligned}$$

Since $u_i^h(x, t^n + \Delta t) \in \mathbb{R}_k[X]$, its projection $u_i^{n+1}(x)$ onto $\mathbb{P}_C^k$ is exact. On the right interface, we have:

$$\begin{aligned} v_h^{\frac{1}{2}}(t, x_{i+\frac{1}{2}}) &= \begin{cases} v_{i-\frac{1}{2}}^{n-\frac{1}{2}}(t - \frac{\Delta x}{a}), & \text{if } (t - t^n) < \frac{\Delta x}{a}, \\ v_{i-\frac{1}{2}}^{n+\frac{1}{2}}(t - \frac{\Delta x}{a}), & \text{otherwise.} \end{cases} \\ &= \begin{cases} v_{i-\frac{1}{2}}(t - \frac{\Delta x}{a})\big|_{I^{n-\frac{1}{2}}}, & \text{if } (t - t^n) < \frac{\Delta x}{a}, \\ v_{i-\frac{1}{2}}(t - \frac{\Delta x}{a})\big|_{I^{n+\frac{1}{2}}}, & \text{otherwise.} \end{cases} \end{aligned}$$

Since $v_h^{\frac{1}{2}}(t, x_{i+\frac{1}{2}}) \in \mathbb{R}_k[X]$, its projection $v_{i+\frac{1}{2}}^{n+\frac{1}{2}}(t)$ onto $\mathbb{P}_I^k$ is exact.

◦ Finally, the implicit scheme gives the exact solution if $u_0(x) \in \mathbb{R}_k[X]$ and $v_{\frac{1}{2}}(t) \in \mathbb{R}_k[X]$.

The scheme is therefore of order $k+1$ in space and time for both implicit and explicit cases. Furthermore, it is very compact. This is an important edge for programming and it will also simplify the extension in two dimensions on unstructured meshes. To illustrate the method we developed, some numerical results are presented in the next section.

Table 1 Convergence tables for the explicit scheme (left) and implicit scheme (right)

k	# cells	L^2 error	order
1	64	$2.501.10^{-3}$	
	128	$3.633.10^{-4}$	2.783
	256	$5.798.10^{-5}$	2.648
3	64	$1.613.10^{-8}$	
	128	$1.008.10^{-9}$	4.000
	256	$6.300.10^{-11}$	4.000
5	64	$7.718.10^{-13}$	
	128	$1.205.10^{-14}$	6.001
	256	$1.883.10^{-16}$	6.000

k	# cells	L^2 error	order
1	64	$3.388.10^{-1}$	
	128	$5.452.10^{-2}$	2.635
	256	$7.296.10^{-3}$	2.902
3	64	$4.748.10^{-5}$	
	128	$2.880.10^{-6}$	4.043
	256	$1.570.10^{-7}$	4.197
5	64	$1.875.10^{-7}$	
	128	$2.958.10^{-9}$	5.986
	256	$4.947.10^{-11}$	5.902

2.3 Numerical results

The one-dimensional scheme is now illustrated on test-cases. All of them are done with a speed $a = \frac{e}{2}$ on a regular mesh and with periodic boundary conditions.

Convergence for regular solution
We consider a regular initial solution in order to test the numerical order of accuracy: $u_0(x) = \sin(\pi x)$ for $x \in [-1, 1]$.

Table 1 gives the L^2-error between the exact solution and the solution given by the explicit ($\lambda = 0.5$) and the implicit schemes ($\lambda = 10$) after a time $t = 10$ s. The expected orders are recovered. Moreover, we see that the L^2-errors are very small. In practice for the explicit scheme, the code had to be run in quadruple precision because the errors were smaller than the numerical double precision. Since the considered time step is larger than with the explicit one, the errors are larger, yet still small, for the implicit scheme.

The space-time GRP scheme is then compared with the WENO and Discontinuous Galerkin (DG) methods.
The WENO method is an extension of the ENO scheme introduced by Harten and al. [28] and was developed by Liu and al. [36]. It was extended to higher order and in multi space dimensions (see [31,2,18]). This scheme almost respects the maximum principle, but its computational cost is rather expensive because of the wide stencil and the numerous coefficients it requires.
The Discontinuous Galerkin method was developed by Cockburn and Shu in their series of articles [11,12] (see also [19,30]).
Unlike the space-time GRP scheme, these two methods are designed to approximate non-linear equations, but their corresponding high-order implicit extensions (see [37,42,33]) are not straightforward. In addition, the generalization in any order k with the WENO or DG is not immediate in the sense that there is no general formula.
From a practical point of view, we used for numerical experiments the DG method as it was written in [11]. The WENO scheme is the one proposed in [36] but with a TVD Runge-Kutta method [27].
The L^2-errors of the third order version of these three methods are compared in Table 2. We took the same initial conditions as above but since the DG method has a more restrictive CFL condition, we had to take $\lambda = 0.2$.
The order of accuracy is recovered by each method. However, we see that the errors obtained with the WENO scheme are several orders of magnitude larger than with the two other methods.

Table 2 Comparison of the convergence of the three 3rd order methods

# cells	WENO	DG	Space-time GRP	order
32	$3.158.10^{-1}$	$1.143.10^{-4}$	$1.531.10^{-5}$	
64	$1.078.10^{-1}$	$1.411.10^{-5}$	$1.914.10^{-6}$	3.00002
128	$2.405.10^{-2}$	$1.757.10^{-6}$	$2.394.10^{-7}$	2.99954
256	$3.148.10^{-3}$	$2.194.10^{-7}$	$2.992.10^{-8}$	2.99991

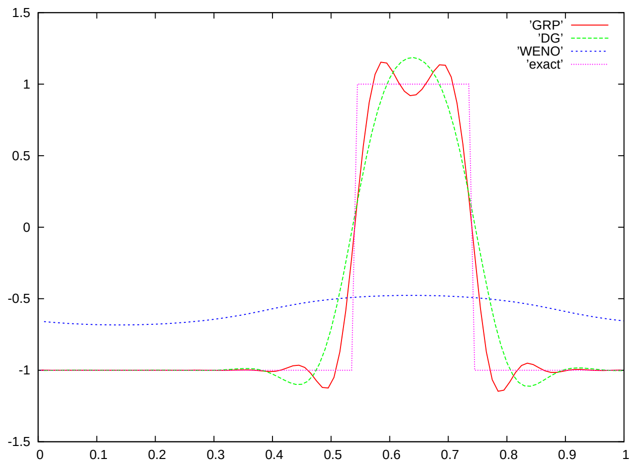

Fig. 3 Comparison of the three methods at $t = 1000\ s$

In practice, this implies that 10 times more cells are required to obtain the same accuracy. The GRP scheme and the DG give comparable results, but we recall that the CFL condition of the DG scheme becomes more restrictive when the order increases.

Discontinuous solution
We now consider the piecewise constant function:

$$u_0(x) = \begin{cases} 1, & \text{if } 0.4 < x < 0.6 \\ -1, & \text{otherwise} \end{cases}$$

as initial solution in the domain $[0, 1]$. The solutions presented here are obtained with 100 cells. The first test-case is a comparison of the three explicit methods of order 3: space-time GRP, WENO and DG in long time. The solutions represented in Figure 3 are obtained with $\lambda = 0.2$ and after a time $t = 1000\ s$. After this very long time, the WENO approximation is smeared and the exact solution can hardly be guessed. On the other hand, the DG and GRP approximations are in a good agreement with the exact solution, with a slight edge for the latter. Concerning the CPU-time, the DG and space-time GRP schemes have a similar computational cost while the WENO method is clearly more expensive.

From now on, we focus on the results of the space-time GRP scheme to check that it has the expected behaviour in long time and with large CFL. The first test-case is devoted to the impact of the degree k of the polynomials in long time, and the second test-case shows some results in a very long time.
The approximations of the solution for different k are compared on Figure 4 after a time $t = 1000\ s$ with $\lambda = 10$ and $\lambda = 50$.

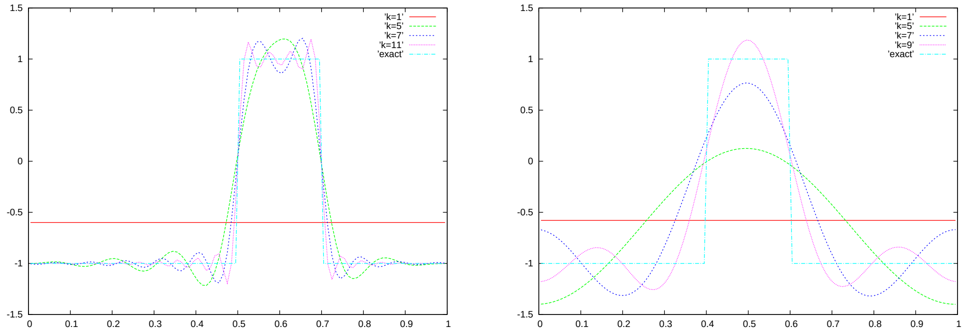

Fig. 4 Solutions at $t = 1000\ s$ with $\lambda = 10$ (left) and $\lambda = 50$ (right)

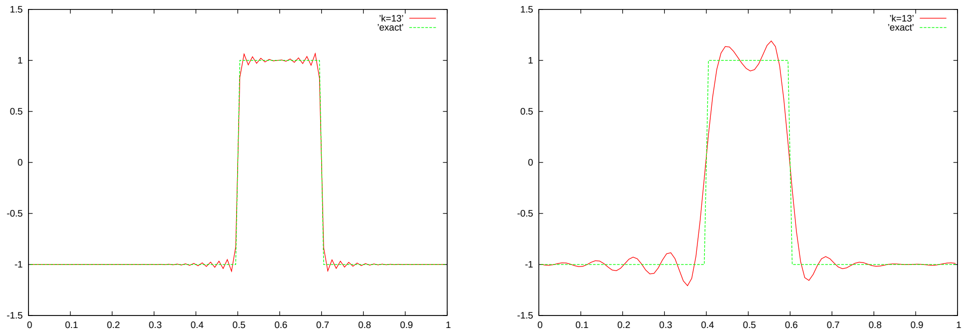

Fig. 5 Solution at $t = 15000\ s$ with $\lambda = 5$ (left) and $\lambda = 20$ (right)

When k increases, in addition to improve the quality of the results, the overshoots are concentrated around the discontinuities and their amplitudes decrease. Taking an increasing degree k, the approximation can be as close to the exact solution as expected.

The scheme is now tested on an extremely long time $t = 15000\ s$. Figure 5 shows the results with $k = 13$ for a CFL number $\lambda = 5$ and $\lambda = 20$.

Despite the difficulty of the test-case (only 100 cells and an extremely long time), the approximations are very close to the exact solution. Furthermore, the computational time of such a benchmark is still reasonable. With $\lambda = 5$, we have a ratio of 8.3 with $k = 3$, of 12.8 with $k = 2$ and of 20 with $k = 1$.
If we want an extremely long time and a very large CFL, k eventually has to be very large. Since the condition number of the matrices increases with k, the numerical errors turn to become important for large values of k. In practice, we can hardly exceed $k = 15 - 20$. In such a case, one issue is to use a basis of orthogonal polynomials.

Linearized Maxwell equations
Now we use our scheme to approximate the solutions of the system of linearized Maxwell equa-

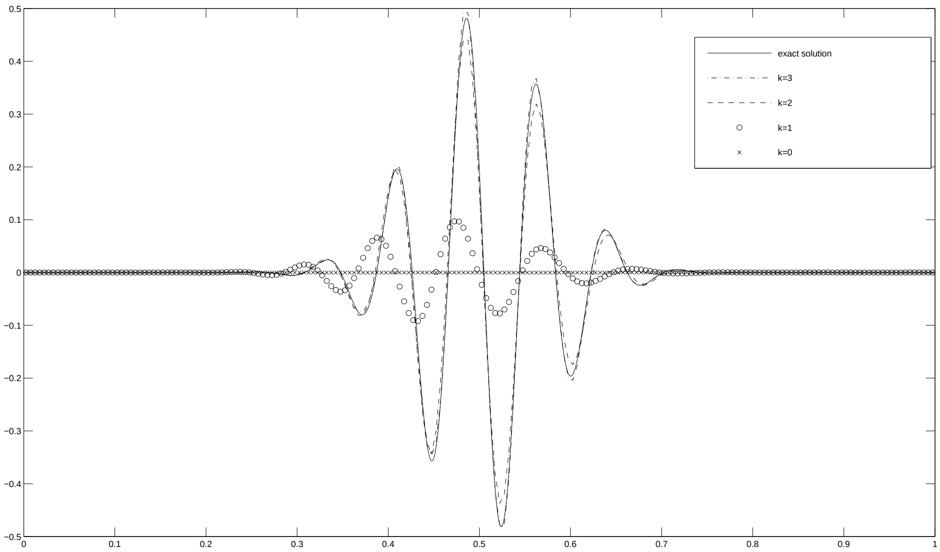

Fig. 6 Maxwell test-case: solutions at $t = 1$ with $\lambda = 5$ for $k = 0, 1, 2, 3$.

tions:

$$\partial_t B_z + \partial_x E_y = 0,$$
$$\partial_t E_y + c^2 \partial_x B_z = 0,$$

where c is the speed of the light, B_z and E_y stand for the z component of the magnetic field and the y component of the electric field respectively.
We perform a simulation involving a left-entering signal representative of a laser impulse:

$$B_z(x = 0, t) = 0.5 \exp\left(-100(t - 0.5)^2\right) \sin(80(t - 0.5)),$$
$$E_y(x = 0, t) = 0,$$

and the initial data is fixed to $B_z(x, t = 0) = E_y(x, t = 0) = 0$. Finally, the speed of the light is normalized: $c = 1$. The main difficulty here lies in the fact that the solution is fully determined by the time-dependent left boundary condition, which is very oscillating. The results for $k = 0, 1, 2$ and 3 for $\lambda = 5$ are shown in Figure 6 and compared to the exact solution.

It is to note that the first order scheme completely fails to reproduce the behavior of the solution. In this case where $\lambda = 5$, the numerical diffusion turns out to be predominant. On the other hand, from $k = 2$ the solution is well preserved. Aside from the general order of the method, in such a situation with a strongly time-dependant boundary condition, having an approximation of order $k + 1$ on the interfaces is an important edge since it greatly improves the approximation of the boundary condition.

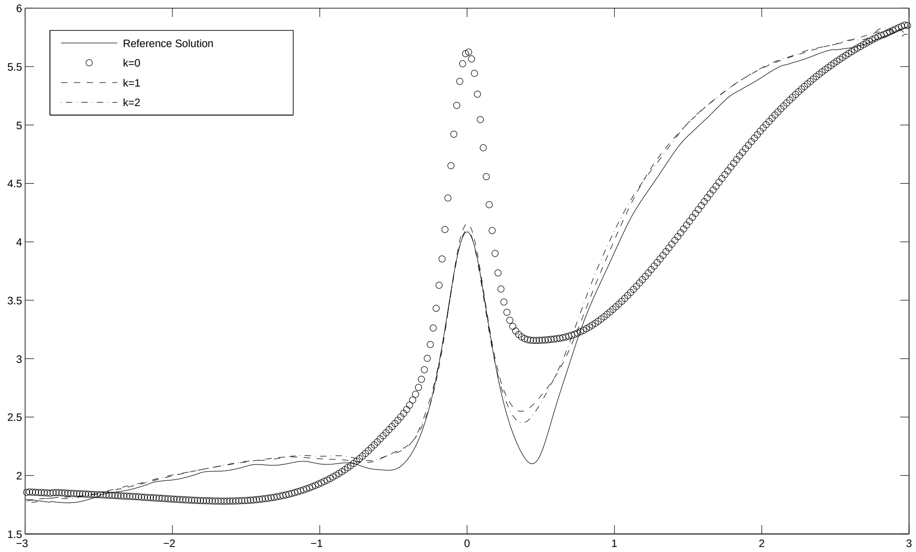

Fig. 7 S_N test-case: solutions at $t = 2.5$ with $\lambda = 16$ for $k = 0, 1, 2$.

S_N model for radiative transfer
The objective of this test is to show that our scheme can easily be coupled with a suitable numerical method for source terms. We hence consider the so-called S_N model for radiative transfer:

$$\partial_t I_j + c\mu_j \partial_x I_j = c\sigma(aT^4 - I_j), \text{ for } j = 1 \dots N,$$

where $I_j(x,t)$ is a radiative intensity in the direction $\mu_j \in [-1 \dots 1]$, $c = 1$ is the speed of the light, a is the Stefan-Boltzmann constant and T is the material temperature. The variable of interest here is the radiative energy $E(x,t) = 2\pi \sum_j I_j \omega_j$ where ω_j are quadrature weights (see [40]).
From a numerical point of view, we here consider a Strang splitting strategy to deal with the source term. For the sake of simplicity, only the first moments $\alpha_i^{n,0}$ are considered when dealing with the source term. This technique is second-order, but higher-order splitting strategies may be used if necessary.

The simulation showed in Figure 7 is obtained with the following setup:

$$\begin{aligned}
aT^4(x) &= e^{-20x^2},\\
\sigma(x) &= e^{-20x^2},\\
I_j(x, t = 0) &= 10^{-2},\\
I_1(x = -3, t) &= 0.3(1 - e^{-t})/\omega_1, \ I_N(x = 3, t) = (1 - e^{-t})/\omega_N,
\end{aligned}$$

the remaining boundary conditions are of Neumann-type. Furthermore, we set $N = 16$ and $\lambda = 16$. Indeed, since the same time step is used for the whole system, this means that the maximum value of λ is 16 (for $\mu = \pm 1$) but λ is smaller in the other directions. We see in Figure 7 that there is a large discrepancy between the reference solution and the first-order scheme. The second-order scheme ($k = 1$) behaves better. Furthermore, using $k = 2$ slightly improves the quality of the solution even though the scheme is still second-order since it is constrained by Strang's splitting.

3 Two-dimensional extension for triangular meshes

3.1 Space-time GRP solver

In this section, we introduce the extension in two dimensions of the space-time GRP scheme we previously developed in one dimension. We still consider the Cauchy problem (2). We assume that we have an unstructured mesh made of triangles and a constant time increment Δt. For the sake of simplicity, the presentation is restricted to triangular meshes, but any other polygons could be considered. We denote by (x_i, y_i) the center of gravity of the triangle T_i and V_i its area. The edges of the triangles T_i are the segments s_j. The segment $s_j = [p_j, q_j],\ p_j \in \mathbb{R}^2, q_j \in \mathbb{R}^2$ is parametrized by:

$$M = (x, y) \in s_j \Leftrightarrow \exists\ \omega(x, y) \in\ [0, 1]/M = p_j + \omega(x, y)\ \overrightarrow{p_j q_j}.$$

We define as in one dimension the interfaces: $I_j^{n+\frac{1}{2}} = (t^n, t^{n+1}) \times s_j$.
Then we define the set of polynomials of degree k per cell (triangle):

$$\mathbb{P}_C^k = \{u \in L^\infty(\mathbb{R}^2)/\ u_{|T_i} \in \mathbb{R}_k[X, Y]\}.$$

We consider the canonical basis of this space:

$$\left\{ \left(\frac{x - x_i}{V_i} \right)^p \left(\frac{y - y_i}{V_i} \right)^q \right\}_{p+q=0,\dots,k}.$$

The associated dot product on the cell T_i is defined by:

$$< f, g >_i = \frac{1}{V_i} \iint_{T_i} f(x, y) g(x, y)\ dx\ dy.$$

We also consider a set of polynomials per interface:

$$\mathbb{P}_I^k = \{u \in L^\infty(\mathbb{R}^2)/\ u_{\big|I_j^{n+\frac{1}{2}}} \in \mathbb{R}_k[X, Y]\}$$

and its basis:

$$\left\{ \left(\frac{t - t^{n+\frac{1}{2}}}{\Delta t} \right)^l \omega^m \right\}_{l+m=0,\dots,k}.$$

The associated dot product on the interface $I_j^{n+\frac{1}{2}}$ is:

$$< f, g >^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \int_0^1 f(t, \omega) g(t, \omega)\ d\omega\ dt.$$

As in the one-dimensional case, the bases of these two sets of polynomials were chosen to simplify the calculations, but other bases such as the Legendre polynomials may be considered for example.

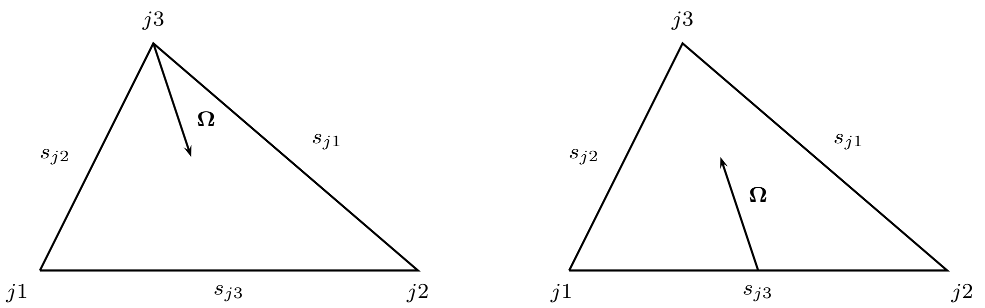

Fig. 8 Case one target (left) and case two targets (right)

This scheme once again consists in the usual exact evolution of high-order approximations and projection steps.

Considered approximations: The initial solution is assumed to be a polynomial approximation of the solution of degree k inside each triangle: $u_i^0(x,y) \in \mathbb{P}_C^k$ for $(x,y) \in T_i$. On each interface, the solution is also assumed to be a polynomial approximation of the solution of degree k: for $t^n \leq t \leq t^{n+1}, (x,y) \in s_j, u(x,y,t) \simeq v_j^{n+\frac{1}{2}}(t,\omega(x,y)) \in \mathbb{P}_I^k$.

Evolution step: The solutions are exactly evolved during a time step. Thanks to the linearity of the equation, the exact evolution of the solutions can still be computed using the characteristics method. In the cell T_i, we consider that the direction of the flux lies between the directions of the two segments s_{j1} and s_{j2}. Depending on the sense of the flux $\mathbf{\Omega}$, we have two cases to deal with (see Fig 8):

- case one target: the flux enters in the cell through s_{j1} and s_{j2},
- case two targets: the flux enters through the third segment s_{j3}.

In each case, we have two subcases to consider (see Fig 9):

- explicit case: some of the characteristic lines coming from the cell T_i at time t^n cross the cell T_i at time t^{n+1},
- implicit case: all the characteristic lines coming from T_i at time t^n cross first an interface.

We can note that, depending on the size of the cells, we can have both explicit and implicit cases during the same time step.
As a convention, we denote by s_j^n the j^{th} segment at time t^n and s_j^{n+1} the j^{th} segment at time t^{n+1}. We call $I_j^{n+\frac{1}{2}}$ the rectangular interface delimited by s_j and (t^n, t^{n+1}). The choice of parametrization of the segments s_j implies that its two edges are ordered: $s_j = [p_j, q_j]$.

In the following, we detail the computations for the implicit case one target. The other three cases are detailed in the appendix.

<u>Case one target (implicit)</u>:

- We know the solution $u_i^n(x,y)$ in the cell T_i at time t^n and the solutions $v_{j1}^{n+\frac{1}{2}}(t,\omega)$ and $v_{j2}^{n+\frac{1}{2}}(t,\omega)$ on the interfaces $I_{j1}^{n+\frac{1}{2}}$ and $I_{j2}^{n+\frac{1}{2}}$.

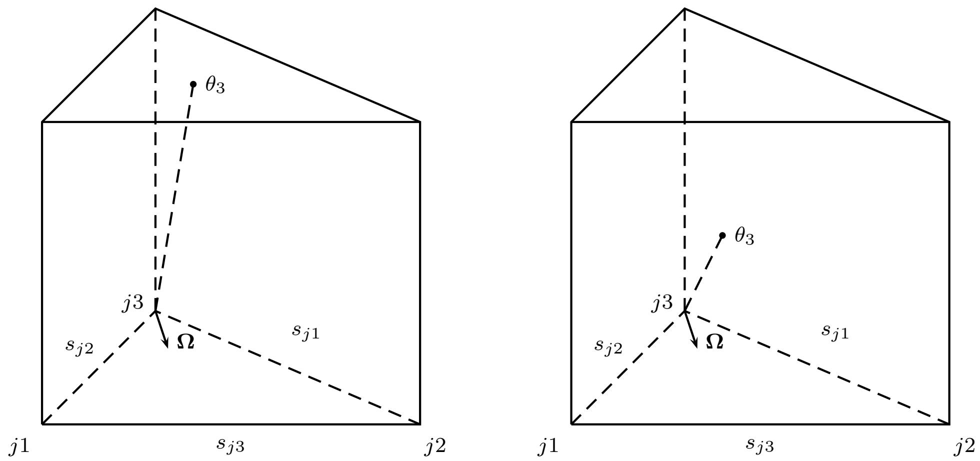

Fig. 9 Explicit case (left) and implicit case (right) for the case one target

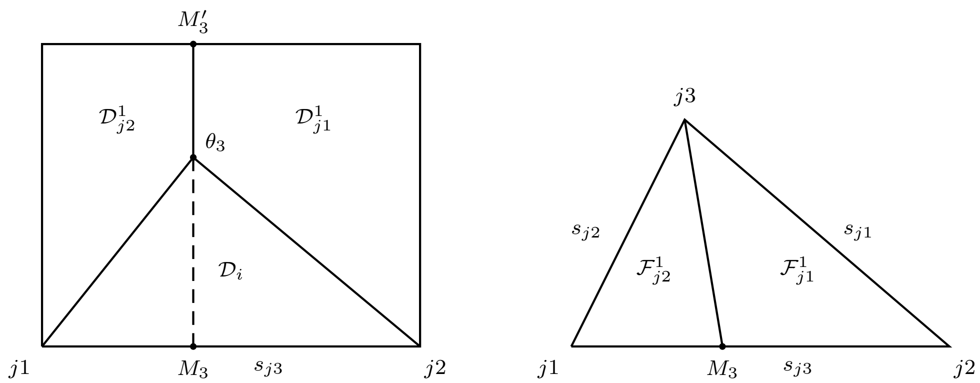

Fig. 10 Incoming solutions on the interface (left) and on the cell (right) for the implicit case one target

- We call θ_3 the intersection between the characteristic line coming from the node $j3$ and the interface $I_{j3}^{n+\frac{1}{2}}$. We call M_3 the projection of θ_3 on s_{j3}^n and M_3' its projection on s_{j3}^{n+1} (see Fig 10).
- The exact solution on the cell T_i at time t^{n+1} is:

$$u_i^h(x,y,t^{n+1}) = \begin{cases} v_{j1}^{n+\frac{1}{2}}(t_1(x,y),\omega_1(x,y)), & \text{if } (x,y) \in \mathcal{F}_{j1}^1 \\ v_{j2}^{n+\frac{1}{2}}(t_2(x,y),\omega_2(x,y)), & \text{if } (x,y) \in \mathcal{F}_{j2}^1 \end{cases}$$

where t_1, ω_1, t_2 and ω_2 are the corresponding changes of variables:

$$\omega_1(x,y) = \frac{\Omega_x(y - y_{p_{j1}} - a\Omega_y(t^n + \Delta t)) - \Omega_y(x - x_{pj1} - a\Omega_x(t^n + \Delta t))}{\Omega_x(y_{q_{j1}} - y_{p_{j1}}) - \Omega_y(x_{q_{j1}} - x_{p_{j1}})}$$

$$t_1(x,y) = \begin{cases} \frac{x_{p_{j1}} + \omega_1(x,y)(x_{q_{j1}} - x_{p_{j1}}) - x}{a\Omega_x} + t^n + \Delta t, & \text{if } \Omega_x \neq 0 \\ \frac{y_{p_{j1}} + \omega_1(x,y)(y_{q_{j1}} - y_{p_{j1}}) - y}{a\Omega_y} + t^n + \Delta t, & \text{otherwise} \end{cases}$$

$$\omega_2(x,y) = \frac{\Omega_x(y - y_{p_{j2}} - a\Omega_y(t^n + \Delta t)) - \Omega_y(x - x_{pj2} - a\Omega_x(t^n + \Delta t))}{\Omega_x(y_{q_{j2}} - y_{p_{j2}}) - \Omega_y(x_{q_{j2}} - x_{p_{j2}})}$$

$$t_2(x,y) = \begin{cases} \frac{x_{p_{j2}} + \omega_2(x,y)(x_{q_{j2}} - x_{p_{j2}}) - x}{a\Omega_x} + t^n + \Delta t, & \text{if } \Omega_x \neq 0 \\ \frac{y_{p_{j2}} + \omega_2(x,y)(y_{q_{j2}} - y_{p_{j2}}) - y}{a\Omega_y} + t^n + \Delta t, & \text{otherwise} \end{cases}$$

- The exact solution on the interface $I_{j3}^{n+\frac{1}{2}}$ is:

$$v_h^{n+\frac{1}{2}}(t, \omega_{j3}) = \begin{cases} u_i^n(x_1(t,\omega), y_1(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_i \\ v_{j1}^{n+\frac{1}{2}}(t_3(t,\omega), \omega_3(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_{j1}^1 \\ v_{j2}^{n+\frac{1}{2}}(t_4(t,\omega), \omega_4(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_{j2}^1 \end{cases}$$

with:

$$\begin{aligned}
x_1(t,\omega) &= x_{p_{j3}} + \omega(x_{q_{j3}} - x_{p_{j3}}) - a\Omega_x(t - t^n) \\
y_1(t,\omega) &= y_{p_{j3}} + \omega(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y(t - t^n) \\
t_3(t,\omega) &= \frac{c_3^1(y_{p_{j1}} - y_{q_{j1}}) + c_3^2(x_{q_{j1}} - x_{p_{j1}})}{d_3} \\
\omega_3(t,\omega) &= \frac{ac_3^2\Omega_x - ac_3^1\Omega_y}{d_3} \\
c_3^1 &= x_{p_{j3}} - x_{p_{j1}} + \omega(x_{q_{j3}} - x_{p_{j3}}) - a\Omega_x t \\
c_3^2 &= y_{p_{j3}} - y_{p_{j1}} + \omega(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y t \\
d_3 &= a\Omega_x(y_{q_{j1}} - y_{p_{j1}}) - a\Omega_y(x_{q_{j1}} - x_{p_{j1}}) \\
t_4(t,\omega) &= \frac{c_4^1(y_{p_{j2}} - y_{q_{j2}}) + c_4^2(x_{q_{j2}} - x_{p_{j2}})}{d_4} \\
\omega_4(t,\omega) &= \frac{ac_4^2\Omega_x - ac_4^1\Omega_y}{d_4} \\
c_4^1 &= x_{p_{j3}} - x_{p_{j2}} + \omega(x_{q_{j3}} - x_{p_{j3}}) - a\Omega_x t \\
c_4^2 &= y_{p_{j3}} - y_{p_{j2}} + \omega(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y t \\
d_4 &= a\Omega_x(y_{q_{j2}} - y_{p_{j2}}) - a\Omega_y(x_{q_{j2}} - x_{p_{j2}})
\end{aligned}$$

Projection step: The exact solutions are now projected onto the space $\mathbb{P}^k$: in the cell T_i at time t^{n+1}, the exact solution is projected onto $\mathbb{P}_C^k$, and on the interfaces $I_j^{n+\frac{1}{2}}$, the solution is projected onto $\mathbb{P}_I^k$. For the implicit case one target, this leads to two minimization problems:

$$\|u_i^{n+1} - u_i^h(t^n + \Delta t)\|_i = \inf_{p \in \mathbb{P}_C^k} \|p - u_i^h(t^n + \Delta t)\|_i, \tag{16}$$

$$\|v_{j3}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j3})\|^n = \inf_{q \in \mathbb{P}_I^k} \|q - v_h^{n+\frac{1}{2}}(\omega_{j3})\|^n, \tag{17}$$

Thanks to the Petrov-Galerkin conditions, we can rewrite (16)-(17) as:

$$< u_i^{n+1} - u_i^h(t^{n+1}), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = 0 \quad \forall\, l+m = 0,\dots,k, \tag{18}$$

$$< v_{j3}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j3}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = 0 \quad \forall\, l+m = 0,\dots,k, \tag{19}$$

Using the decomposition of the solutions in their corresponding basis:

$$\begin{aligned}
u_i^{n+1}(x,y) &= \sum_{p+q=0}^{k} \alpha_i^{n+1,p,q} \left(\frac{x-x_i}{V_i}\right)^p \left(\frac{y-y_i}{V_i}\right)^q, \\
v_{j1}^{n+\frac{1}{2}}(t,\omega) &= \sum_{p+q=0}^{k} \beta_{j1}^{n+\frac{1}{2},p,q} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega^q, \\
v_{j2}^{n+\frac{1}{2}}(t,\omega) &= \sum_{p+q=0}^{k} \beta_{j2}^{n+\frac{1}{2},p,q} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega^q, \\
v_{j3}^{n+\frac{1}{2}}(t,\omega) &= \sum_{p+q=0}^{k} \beta_{j3}^{n+\frac{1}{2},p,q} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega^q,
\end{aligned}$$

the minimization problems (18)-(19) are equivalent to solving the two following linear systems of size $\frac{(k+1)(k+2)}{2}$:

$$\sum_{p+q=0}^{k} \alpha_i^{n+1,p,q} < u_i^{n+1}, \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = < u_i^h(t^{n+1}), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i \quad \forall\, l+m=0,\dots,k,$$

$$\sum_{p+q=0}^{k} \beta_{j3}^{n+\frac{1}{2},p,q} < v_{j3}^{n+\frac{1}{2}}, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = < v_h^{n+\frac{1}{2}}(\omega_{j3}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n \quad \forall\, l+m=0,\dots,k$$

For the left hand sides, we have:

$$\begin{aligned}
< u_i^{n+1}, \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i &= \frac{1}{V_i} \iint_{T_i} \left(\frac{x-x_i}{V_i}\right)^{p+l} \left(\frac{y-y_i}{V_i}\right)^{p+m} dx\ dy \\
&= A^i(p,q,l,m), \\
< v_j^{n+\frac{1}{2}}, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n &= \frac{1}{\Delta t} \iint_{I_j^{n+\frac{1}{2}}} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^{p+l} \omega^{q+m} dt\ d\omega \\
&= \frac{1}{(p+l+1)(q+m+1)} \left[\left(\frac{1}{2}\right)^{p+l+1} \left(-\frac{1}{2}\right)^{q+m+1} \right] \\
&= A_j(p,q,l,m).
\end{aligned}$$

For the right hand sides, we have for the implicit case one target (18)-(19):

$$
\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = & \sum_{p+q=0}^{k} \beta_{j1}^{n+\frac{1}{2},p,q} J_{im,j1}^{p,q,l,m} + \beta_{j2}^{n+1,p,q} J_{im,j2}^{p,q,l,m} \\
= & b^i(l,m), \\
< v_h^{n+\frac{1}{2}}(\omega_{j3}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = & \sum_{p+q=0}^{k} \alpha_i^{n,p,q} K_{im,i}^{p,q,l,m} + \beta_{j1}^{n+\frac{1}{2},p,q} K_{im,j1}^{p,q,l,m} + \beta_{j2}^{n+1,p,q} K_{im,j2}^{p,q,l,m} \\
= & b_{j3}(l,m)
\end{aligned}
$$

where the coefficients are given by:

$$
\begin{aligned}
J_{im,j1}^{p,q,l,m} &= \frac{1}{V_i} \iint_{\mathcal{F}_{j1}^1} \left(\frac{t_1(x,y)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_1(x,y)^q \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m dx\ dy \\
J_{im,j2}^{p,q,l,m} &= \frac{1}{V_i} \iint_{\mathcal{F}_{j2}^1} \left(\frac{t_2(x,y)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_2(x,y)^q \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m dx\ dy \\
K_{im,i}^{p,q,l,m} &= \frac{1}{\Delta t} \iint_{\mathcal{D}_i} \left(\frac{x_1(t,\omega)-x_i}{V_i}\right)^p \left(\frac{y_1(t,\omega)-y_i}{V_i}\right)^q \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega \\
K_{im,j1}^{p,q,l,m} &= \frac{1}{\Delta t} \iint_{\mathcal{D}_{j1}^1} \left(\frac{t_3(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_3(t,\omega)^q \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega \\
K_{im,j2}^{p,q,l,m} &= \frac{1}{\Delta t} \iint_{\mathcal{D}_{j2}^1} \left(\frac{t_4(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_4(t,\omega)^q \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega.
\end{aligned}
$$

Although these numerous coefficients seem complicated, they are all obtained in the same way once we know the changes of variables and the areas for the exact solutions.

In practice, the method can be summarized in the following way:

- At the initial time, the solution $u_0(x,y)$ is known inside each triangle T_i and the solution $v^{\frac{1}{2}}(t,\omega)$ is known on the interfaces s_j of the boundary where the flux enters: $\mathbf{n}_{s_j} \cdot \Omega < 0$, with $\mathbf{n}_{s_j}$ the outgoing normal to the segment s_j.
- An order to run the interfaces is defined to insure that the exact solution can be calculated.
- On each interface $I_j^{n+\frac{1}{2}}$, the corresponding linear system of size $\frac{(k+1)(k+2)}{2}$ has to be solved to compute the update $v_j^{n+\frac{1}{2}}(t,\omega)$:

$$\sum_{p+q=0}^{k} \beta_j^{n+\frac{1}{2},p,q} A_j(p,q,l,m) = \ b_j(l,m) \quad \forall\ l+m=0,\ldots,k.$$

- Inside each cell T_i at time t^{n+1}, the corresponding linear system of size $\frac{(k+1)(k+2)}{2}$ has to be solved to compute the updates $u_i^{n+1}(x,y)$:

$$\sum_{p+q=0}^{k} \alpha_i^{n+1,p,q} A^i(p,q,l,m) = \ b^i(l,m) \quad \forall\ l+m=0,\ldots,k.$$

We note that the matrices do not depend on time. On each cell, the matrix has then to be computed and inverted once. On the interfaces, we also note that the matrices do not depend on space, so only one matrix has to be computed and inverted for all the interfaces.

Theorem 2 *The scheme exactly preserves the polynomials of degree k.*

Proof The proof is similar to the one-dimensional case.

Finally, the resulting scheme is of order $k+1$ in space and time. Moreover, we treat the explicit and implicit cases simultaneously. From a numerical point of view, the main difficulty comes from the computation of the integrals in the matrices and in the right hand sides of the systems. To simplify their computations, some quadrature methods of order $2k$ can be used.

To illustrate the method, some numerical results are presented in the next section.

3.2 Numerical results

Numerical order evaluation
In this first test-case, the numerical order of the 2D scheme is evaluated in the same fashion as it was done in 1D. Indeed, we consider the solution $u(x,y,t) = \sin(\pi(x-t))$ on the domain $[0,1]^2 \times [0,1.5]$. Therefore, the initial and left boundary conditions are fixed accordingly:

$$u_0(x,y) = \sin(\pi x) \text{ and } u(0,y,t) = \sin(-\pi t).$$

The results for $\lambda = 0.9$ are summarized in Table 3.
The meshes were obtained using the *Triangle* software (http://www.cs.cmu.edu/ quake/triangle.html)

Table 3 Convergence rate for the 2D scheme

number of cells	max. cell area	L^2 error ($k=1$)	order	L^2 error ($k=2$)	order
16677	9.998e-5	3.504e-5	2.01	4.59e-7	3.2
20411	8.000e-5	1.388e-5	1.97	1.54e-7	3.92
33242	5.000e-5	8.731e-6	-	4.98e-8	-

and are not refinement of one another.
The expected orders of accuracy are recovered. As in 1D, the L^2 errors are very small.

Advection of a disc
This test-case is done with a speed $a = 5$ on an unstructured mesh. The domain is a 20×2 rectangle meshed with 66498 cells, whose smallest triangle has an area of $1.95.10^{-4}$. For the sake of simplicity, it is chosen long enough to avoid dealing with periodic boundary conditions. We compare here the schemes of order 1, 2 and 3 ($k = 0,1,2$). The initial solution is a disc on the left part of the domain (see Figure 11):

$$u_0(x,y) = \begin{cases} 1, & \text{if } \quad (x-0.6)^2 + (y-1)^2 < 0.25 \\ 0, & \text{otherwise} \end{cases}$$

For each degree k of polynomials, the test-case is run with CFL numbers of 0.5 and 10. The solutions are given at time $t = 3.6\ s$ on Figures 12-13-14. In order to see the mesh and the

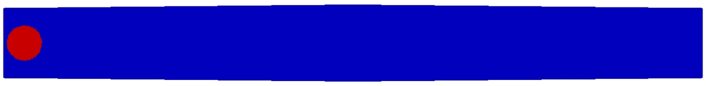

Fig. 11 Initial solution and computational domain

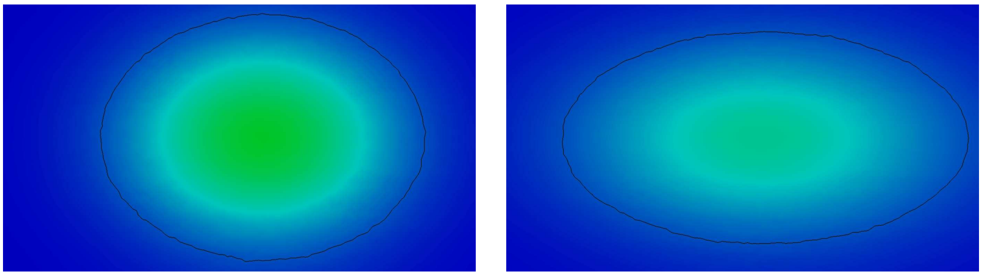

Fig. 12 Zoomed solution and contour lines of 0.1, 0.5, 0.9 for $k = 0$ with $CFL = 0.5$ (left) and 10 (right)

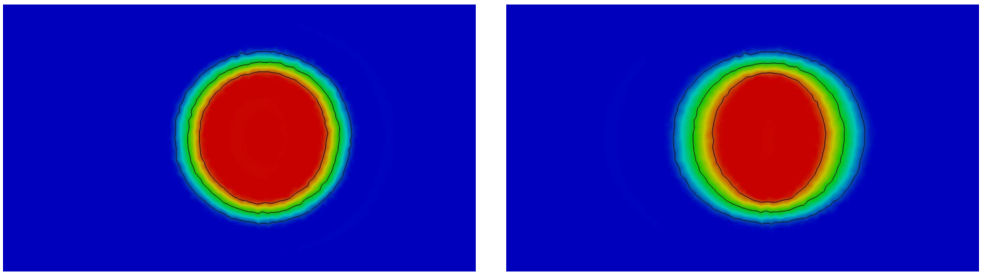

Fig. 13 Zoomed solution and contour lines of 0.1, 0.5, 0.9 for $k = 1$ with $CFL = 0.5$ (left) and 10 (right)

Fig. 14 Zoomed solution and contour lines of 0.1, 0.5, 0.9 for $k = 2$ with $CFL = 0.5$ (left) and 10 (right)

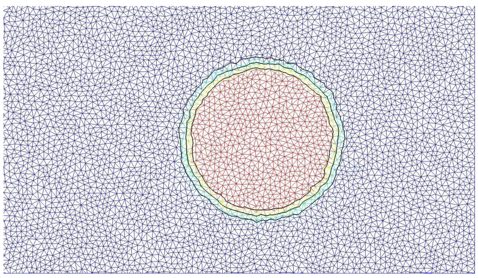

Fig. 15 Mesh, zoomed solution and contour lines of 0.1, 0.5, 0.9 for $k = 2$ with $CFL = 0.5$

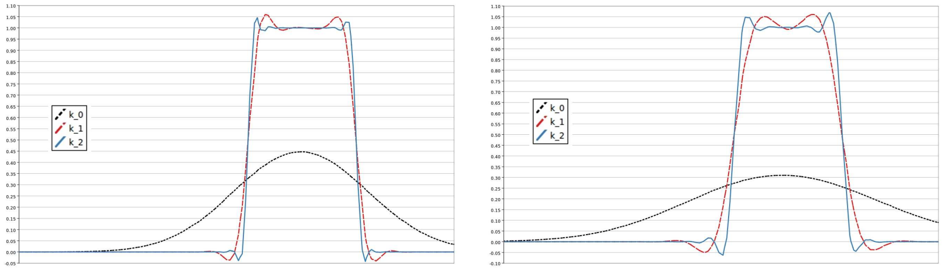

Fig. 16 Cuts of the solution along $y = 1$ with $CFL = 0.5$(left) and 10 (right)

Fig. 17 Zoomed solution and contour lines of $0.1,\ 0.5,\ 0.9$ for $k = 2$ with $CFL = 0.5$ (left) and 10 (right)

number of cells in the discontinuity, Figure 15 shows the same result than Figure 14 with the corresponding mesh. For a better visibility, the results are zoomed in x between 17 and 20 and three contour lines of 0.1, 0.5 and 0.9 are added on each figure. We globally note that the results are more diffused in the direction of the flux. The behaviour of the solutions with the different degrees k is really close to the one-dimensional case.
To have a better comparison of the results, some cuts along the axis $y = 1$ are represented on Figure 16. For a better visibility, the graphs are also zoomed in x between 17 and 20.
The same experiment is done with a square initial solution:

$$u_0(x, y) = \begin{cases} 1, & \text{if} \quad (x, y) \in [0.2, 1.2] \times [0.5, 1.5] \\ 0, & \text{otherwise.} \end{cases}$$

The results are shown on Figure 17. Once again, we observe that this approximation is very close to the exact solution, and especially, the angles are preserved.
Figure 18 is done with the disc initial solution on a domain with three times more cells whose smallest triangle has an area of $4.94.10^{-5}$. We took $k = 2$ and a large CFL number (50).
In order to test the scheme in very long time, the last Figure 19 is done with a speed $a = 10^{-3}$ and a CFL number of 20 and shows the solution after a time $t = 18000\ s$. In spite of the very long time and the large CFL number, the solution given by the scheme is very close to the exact solution.
The last remark concerns the choice between refining the mesh or taking a smaller CFL number. For a fixed quality of approximation and a given number of cells in the mesh, we look for the corresponding CFL number. We still consider the disc initial solution on a rectangular mesh with a speed $a = 5$. In Table 4, three configurations and their computational time are given.
We observe that if we consider a mesh four times finer, the corresponding CFL number is doubled, and the computational time is quadrupled.

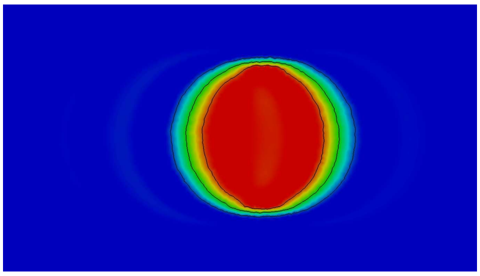

Fig. 18 Zoomed solution and contour lines of $0.1,\ 0.5,\ 0.9$ for $k = 2$ with $CFL = 50$

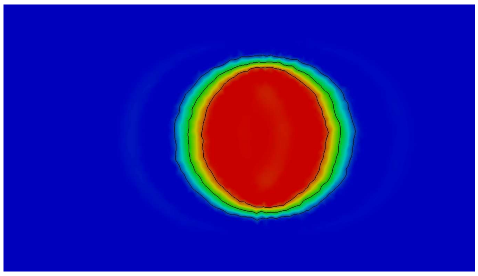

Fig. 19 Zoomed solution and contour lines of $0.1,\ 0.5,\ 0.9$ for $k = 2$ at $t = 18000\ s$ with $a = 10^{-3}$,$CFL = 20$

Table 4 Comparison of computational times for a fixed quality of approximation

number of cells	CFL	CPU-time
33332	13	13.47
66498	20	25.02
133179	30	50.02

S_N model
This last test-case concerns the 2D S_N model previously introduced in 1D:

$$\partial_t I_j + c\Omega_j.\nabla_{\mathbf{x}} I_j = c\sigma(aT^4 - I_j),\ \text{for } j = 1 \dots N,$$

where $\Omega_j \in \mathbb{S}^2$ and the other variables have been introduced in the corresponding 1D numerical experiment. As in the 1D case, the source term was taken into account using a simple Strang splitting strategy. Once again, only the first moment of the solution inside the cells is used for this purpose.

The radiative energy $E = \sum_j I_j \omega_j$, where ω_j are quadrature weights, is computed with the following setup: $N = 16$, $\lambda = 4$, $E(\mathbf{x}, t = 0) = 0$, $aT^4 = 0.01$ on a grid made of 15167 triangles of an hexagonal domain. Two beams are entering the domain with an energy of 1 and two other with an energy of 0.5. Two computations corresponding to $\sigma = 0.1$ and $\sigma = 5$ are shown on Figure 20 using $k = 1$ (which corresponds to second-order as Strang's splitting).
We clearly see that, even with $k = 1$ and $\lambda = 4$, that the numerical diffusion is neglectable in the direction transverse to the beams.

4 Conclusion

In this article, we developed a high-order GRP-type scheme for the linear advection equation with no restriction on the time step. Thanks to the linearity of the equation and our choice of

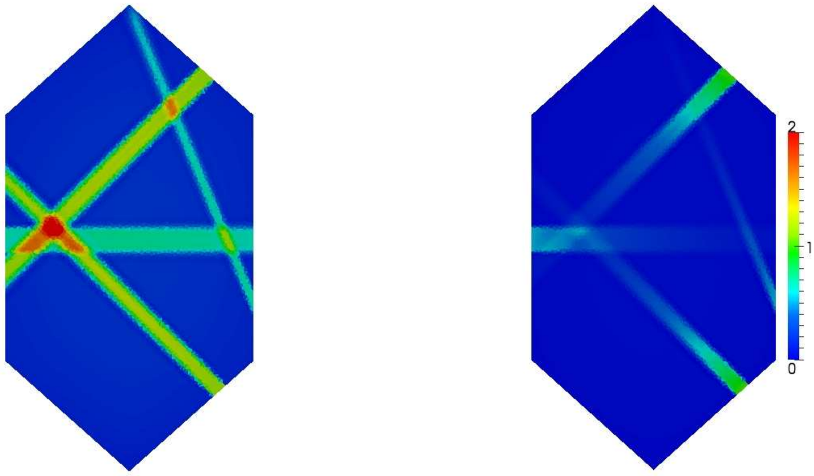

Fig. 20 Radiative energy at time $t = 1.3$ for $\sigma = 0.1$(l) and $\sigma = 5$ (r).

projection, the scheme simply consists in solving one linear system per cell and one linear system per interface. It was proved that the scheme is of order $k + 1$ in space and time for k arbitrary large. Then, a two-dimensional extension of this scheme has been developed on unstructured meshes. Again, the time step can be chosen arbitrary large. Several test-cases and comparisons have been illustrating the excellent behavior of the scheme. Moreover, the source terms can be included using for instance a relevant operator splitting technique.
Compared with the standard GRP technique, it has the advantage of being very compact for any value of the time step while preserving an order $k + 1$. It also naturally handles the time-dependant boundary conditions and the compactness of its stencil makes its 2D extension easier to program. On the other hand, the additional projections occuring for very large values of λ makes it slightly less accurate than the GRP scheme.
Compared with the other existing high-order methods, this space-time GRP scheme only requires an initial effort for taking the geometry into account. Once this price has been paid, it is very simple to code for any value of k (requiring only a relevant quadrature formula). This remark obviously applies to 3D extensions, which will require an effort on the geometry. Finally, this scheme is easily parallelizable: for instance, we observe a factor 8 on 12 processors with nothing but OpenMP routines.
On the other hand, the extension to nonlinear hyperbolic systems is far from being trivial. A BGK-type technique seems a promising candidate.

5 Appendix

Evolution step:
Case two targets (explicit):

- We know the solution $u_i^n(x, y)$ in the cell T_i at time t^n and the solution $v_{j3}^{n+\frac{1}{2}}(t, \omega)$ on the interface $I_{j3}^{n+\frac{1}{2}}$.

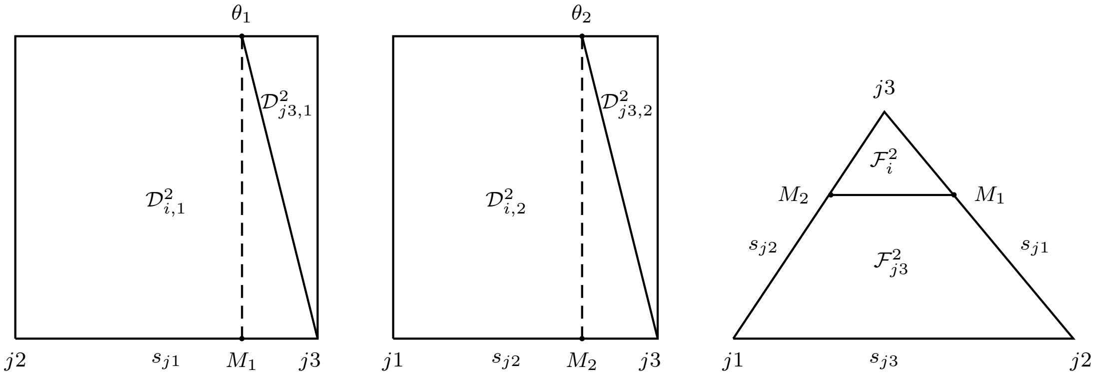

Fig. 21 Incoming solutions on the interface (left) and on the cell (right) for the explicit case two targets

- We define θ_1 as the intersection between the characteristic line coming from s^n_{j3} and the segment s^{n+1}_{j1}. We also define θ_2, the intersection between the characteristic line coming from s^n_{j3} and the segment s^{n+1}_{j2}. Then we call M_1 the projection of θ_1 on s^n_{j1} and M_2 the projection of θ_2 on s^n_{j2} (see Fig 21).
- The exact solution on the cell T_i at time t^{n+1} is:

$$u^h_i(x,y,t^n+\Delta t) = \begin{cases} u^n_i(x_1(x,y),y_1(x,y)), & \text{if } (x,y)\in\mathcal{F}^2_i \\ v^{n+\frac{1}{2}}_{j3}(t_1(x,y),\omega_1(x,y)), & \text{if } (x,y)\in\mathcal{F}^2_{j3} \end{cases}$$

with:

$$\begin{aligned}
x_1(x,y) &= x - a\Omega_x\Delta t \\
y_1(x,y) &= y - a\Omega_y\Delta t \\
\omega_1(x,y) &= \frac{\Omega_x(y-y_{p_{j3}}-a\Omega_y(t^n+\Delta t)) - \Omega_y(x-x_{pj3}-a\Omega_x(t^n+\Delta t))}{\Omega_x(y_{q_{j3}}-y_{p_{j3}})-\Omega_y(x_{q_{j3}}-x_{p_{j3}})} \\
t_1(x,y) &= \begin{cases} \frac{x_{p_{j3}}+\omega_1(x,y)(x_{q_{j3}}-x_{p_{j3}})-x}{a\Omega_x}+t^n+\Delta t, & \text{if } \Omega_x\neq 0 \\ \frac{y_{p_{j3}}+\omega_1(x,y)(y_{q_{j3}}-y_{p_{j3}})-y}{a\Omega_y}+t^n+\Delta t, & \text{otherwise} \end{cases}
\end{aligned}$$

- The exact solution on the interface $I^{n+\frac{1}{2}}_{j1}$ is:

$$v^{n+\frac{1}{2}}_h(t,\omega_{j1}) = \begin{cases} u^n_i(x_2(t,\omega),y_2(t,\omega)), & \text{if } (t,\omega)\in\mathcal{D}^2_{i,1} \\ v^{n+\frac{1}{2}}_{j3}(t_2(t,\omega),\omega_2(t,\omega)), & \text{if } (t,\omega)\in\mathcal{D}^2_{j3,1} \end{cases}$$

with:

$$\begin{aligned}
x_2(t,\omega) &= x_{p_{j1}} + \omega(x_{q_{j1}} - x_{p_{j1}}) - a\Omega_x(t-t^n)\\
y_2(t,\omega) &= y_{p_{j1}} + \omega(y_{q_{j1}} - y_{p_{j1}}) - a\Omega_y(t-t^n)\\
t_2(t,\omega) &= \frac{c_2^1(y_{p_{j3}} - y_{q_{j3}}) + c_2^2(x_{q_{j3}} - x_{p_{j3}})}{d_2}\\
\omega_2(t,\omega) &= \frac{ac_2^2\Omega_x - ac_2^1\Omega_y}{d_2}\\
c_2^1 &= x_{p_{j1}} - x_{p_{j3}} + \omega(x_{q_{j1}} - x_{p_{j1}}) - a\Omega_x t\\
c_2^2 &= y_{p_{j1}} - y_{p_{j3}} + \omega(y_{q_{j1}} - y_{p_{j1}}) - a\Omega_y t\\
d_2 &= a\Omega_x(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y(x_{q_{j3}} - x_{p_{j3}})
\end{aligned}$$

- The exact solution on the interface $I_{j2}^{n+\frac{1}{2}}$ is:

$$v_h^{n+\frac{1}{2}}(t,\omega_{j2}) = \begin{cases} u_i^n(x_3(t,\omega), y_3(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_{i,2}^2\\ v_{j3}^{n+\frac{1}{2}}(t_3(t,\omega), \omega_3(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_{j3,2}^2 \end{cases}$$

with:

$$\begin{aligned}
x_3(t,\omega) &= x_{p_{j2}} + \omega(x_{q_{j2}} - x_{p_{j2}}) - a\Omega_x(t-t^n)\\
y_3(t,\omega) &= y_{p_{j2}} + \omega(y_{q_{j2}} - y_{p_{j2}}) - a\Omega_y(t-t^n)\\
t_3(t,\omega) &= \frac{c_3^1(y_{p_{j3}} - y_{q_{j3}}) + c_3^2(x_{q_{j3}} - x_{p_{j3}})}{d_3}\\
\omega_3(t,\omega) &= \frac{ac_3^2\Omega_x - ac_3^1\Omega_y}{d_3}\\
c_3^1 &= x_{p_{j2}} - x_{p_{j3}} + \omega(x_{q_{j2}} - x_{p_{j2}}) - a\Omega_x t\\
c_3^2 &= y_{p_{j2}} - y_{p_{j3}} + \omega(y_{q_{j2}} - y_{p_{j2}}) - a\Omega_y t\\
d_3 &= a\Omega_x(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y(x_{q_{j3}} - x_{p_{j3}})
\end{aligned}$$

Case one target (explicit):

- We know the solution $u_i^n(x,y)$ in the cell T_i at time t^n and the solutions $v_{j1}^{n+\frac{1}{2}}(t,\omega)$ and $v_{j2}^{n+\frac{1}{2}}(t,\omega)$ on the interfaces I_{j1} and $I_{j2}^{n+\frac{1}{2}}$.
- We define θ_1 as the intersection between the characteristic line coming from the segment s_{j1}^n and the segment s_{j3}^{n+1} (i.e. such that $d(s_{j3}^n, s_{j1}^n) = a\Delta t$). We also define θ_2 as the intersection between the characteristic line coming from the segment s_{j2}^n and the segment s_{j3}^{n+1} (*i.e.* such that $d(s_{j3}^n, s_{j2}^n) = a\Delta t$). Then we call M_1 the projection of θ_1 on s_{j3}^n and M_2 the projection of θ_2 on s_{j3}^n (see Fig 22).
- The exact solution on the cell T_i at time t^{n+1} is:

$$u_i^h(x,y,t^n+\Delta t) = \begin{cases} u_i^n(x_1(x,y), y_1(x,y)), & \text{if } (x,y) \in \mathcal{F}_i\\ v_{j1}^{n+\frac{1}{2}}(t_1(x,y), \omega_1(x,y)), & \text{if } (x,y) \in \mathcal{F}_{j1}^1\\ v_{j2}^{n+\frac{1}{2}}(t_2(x,y), \omega_2(x,y)), & \text{if } (x,y) \in \mathcal{F}_{j2}^1 \end{cases}$$

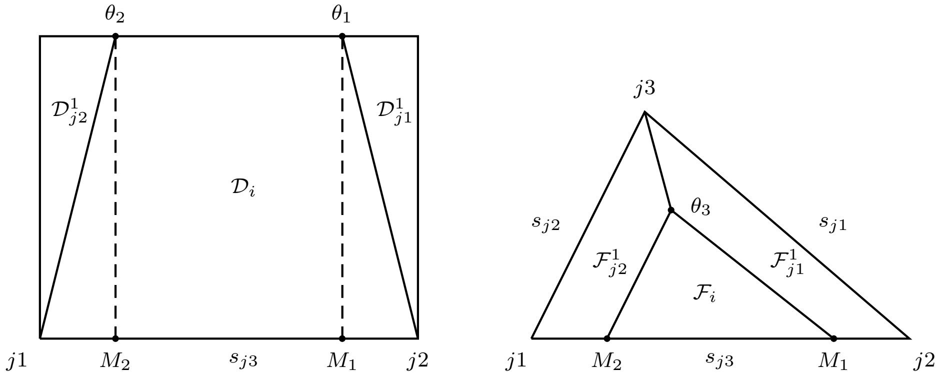

Fig. 22 Incoming solutions on the interface (left) and on the cell (right) for the explicit case 1

with:

$$
\begin{aligned}
x_1(x,y) &= x - a\Omega_x \Delta t \\
y_1(x,y) &= y - a\Omega_y \Delta t \\
\omega_1(x,y) &= \frac{\Omega_x(y - y_{p_{j1}} - a\Omega_y(t^n+\Delta t)) - \Omega_y(x - x_{pj1} - a\Omega_x(t^n+\Delta t))}{\Omega_x(y_{q_{j1}} - y_{p_{j1}}) - \Omega_y(x_{q_{j1}} - x_{p_{j1}})} \\
t_1(x,y) &= \begin{cases} \frac{x_{p_{j1}} + \omega_1(x,y)(x_{q_{j1}} - x_{p_{j1}}) - x}{a\Omega_x} + t^n + \Delta t, & \text{if } \Omega_x \neq 0 \\ \frac{y_{p_{j1}} + \omega_1(x,y)(y_{q_{j1}} - y_{p_{j1}}) - y}{a\Omega_y} + t^n + \Delta t, & \text{otherwise} \end{cases} \\
\omega_2(x,y) &= \frac{\Omega_x(y - y_{p_{j2}} - a\Omega_y(t^n+\Delta t)) - \Omega_y(x - x_{pj2} - a\Omega_x(t^n+\Delta t))}{\Omega_x(y_{q_{j2}} - y_{p_{j2}}) - \Omega_y(x_{q_{j2}} - x_{p_{j2}})} \\
t_2(x,y) &= \begin{cases} \frac{x_{p_{j2}} + \omega_2(x,y)(x_{q_{j2}} - x_{p_{j2}}) - x}{a\Omega_x} + t^n + \Delta t, & \text{if } \Omega_x \neq 0 \\ \frac{y_{p_{j2}} + \omega_2(x,y)(y_{q_{j2}} - y_{p_{j2}}) - y}{a\Omega_y} + t^n + \Delta t, & \text{otherwise} \end{cases}
\end{aligned}
$$

- The exact solution on the interface $I_{j3}^{n+\frac{1}{2}}$ is:

$$
v_h^{n+\frac{1}{2}}(t,\omega_{j3}) = \begin{cases} u_i^n(x_2(t,\omega), y_2(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_i \\ v_{j1}^{n+\frac{1}{2}}(t_3(t,\omega), \omega_3(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_{j1}^1 \\ v_{j2}^{n+\frac{1}{2}}(t_4(t,\omega), \omega_4(t,\omega)), & \text{if } (t,\omega) \in \mathcal{D}_{j2}^1 \end{cases}
$$

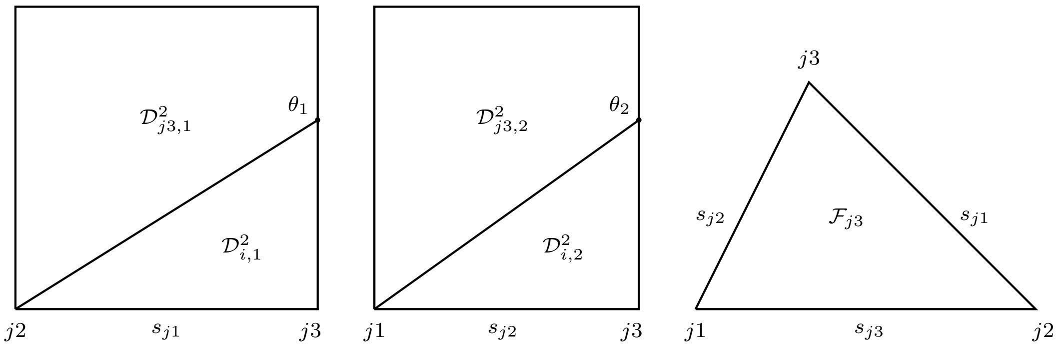

Fig. 23 Incoming solutions on the interface (left) and on the cell (right) for the implicit case 2

with:

$$
\begin{aligned}
x_2(t,\omega) &= x_{p_{j3}} + \omega(x_{q_{j3}} - x_{p_{j3}}) - a\Omega_x(t - t^n)\\
y_2(t,\omega) &= y_{p_{j3}} + \omega(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y(t - t^n)\\
t_3(t,\omega) &= \frac{(c_3^1(y_{p_{j1}} - y_{q_{j1}}) + c_3^2(x_{q_{j1}} - x_{p_{j1}})}{d_3}\\
\omega_3(t,\omega) &= \frac{ac_3^2\Omega_x - ac_3^1\Omega_y}{d_3}\\
c_3^1 &= x_{p_{j3}} - x_{p_{j1}} + \omega(x_{q_{j3}} - x_{p_{j3}}) - a\Omega_x t\\
c_3^2 &= y_{p_{j3}} - y_{p_{j1}} + \omega(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y t\\
d_3 &= a\Omega_x(y_{q_{j1}} - y_{p_{j1}}) - a\Omega_y(x_{q_{j1}} - x_{p_{j1}})\\
t_4(t,\omega) &= \frac{(c_4^1(y_{p_{j2}} - y_{q_{j2}}) + c_4^2(x_{q_{j2}} - x_{p_{j2}})}{d_4}\\
\omega_4(t,\omega) &= \frac{ac_4^2\Omega_x - ac_4^1\Omega_y}{d_4}\\
c_4^1 &= x_{p_{j3}} - x_{p_{j2}} + \omega(x_{q_{j3}} - x_{p_{j3}}) - a\Omega_x t\\
c_4^2 &= y_{p_{j3}} - y_{p_{j2}} + \omega(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y t\\
d_4 &= a\Omega_x(y_{q_{j2}} - y_{p_{j2}}) - a\Omega_y(x_{q_{j2}} - x_{p_{j2}})
\end{aligned}
$$

Case two targets (implicit):

- We know the solution $u_i^n(x,y)$ in the cell T_i at time t^n and the solution $v_{j3}^{n+\frac{1}{2}}(t,\omega)$ on the interface $I_{j3}^{n+\frac{1}{2}}$.
- We define θ_3 as the intersection between the characteristic line coming from s_{j3}^n and the segment $[t^n, t^{n+1}]$ for $(x,y) = (x_{j3}, y_{j3})$ (see Fig 23).

- The exact solution on the cell T_i at time t^{n+1} is:

$$u_i^h(x,y,t^n+\Delta t) = v_{j3}^{n+\frac{1}{2}}(t_1(x,y),\omega_1(x,y)) \quad \text{if } (x,y)\in T_i,$$

with:

$$\omega_1(x,y) = \frac{\Omega_x(y - y_{p_{j3}} - a\Omega_y(t^n+\Delta t)) - \Omega_y(x - x_{pj3} - a\Omega_x(t^n+\Delta t))}{\Omega_x(y_{q_{j3}} - y_{p_{j3}}) - \Omega_y(x_{q_{j3}} - x_{p_{j3}})}$$

$$t_1(x,y) = \begin{cases} \frac{x_{p_{j3}}+\omega_1(x,y)(x_{q_{j3}}-x_{p_{j3}})-x}{a\Omega_x} + t^n + \Delta t, & \text{if } \Omega_x \neq 0 \\ \frac{y_{p_{j3}}+\omega_4(x,y)(y_{q_{j3}}-y_{p_{j3}})-y}{a\Omega_y} + t^n + \Delta t, & \text{otherwise} \end{cases}$$

- The exact solution on the interface $I_{j1}^{n+\frac{1}{2}}$ is:

$$v_h^{n+\frac{1}{2}}(t,\omega_{j1}) = \begin{cases} u_i^n(x_1(t,\omega,y_1(t,\omega)), & \text{if } (t,\omega)\in\mathcal{D}^2_{i,1} \\ v_{j3}^{n+\frac{1}{2}}(t_2(t,\omega),\omega_2(t,\omega)), & \text{if } (t,\omega)\in\mathcal{D}^2_{j3,1} \end{cases}$$

with:

$$\begin{aligned} x_1(t,\omega) &= x_{p_{j1}} + \omega(x_{q_{j1}} - x_{p_{j1}}) - a\Omega_x(t-t^n) \\ y_1(t,\omega) &= y_{p_{j1}} + \omega(y_{q_{j1}} - y_{p_{j1}}) - a\Omega_y(t-t^n) \\ t_2(t,\omega) &= \frac{(c_2^1(y_{p_{j3}} - y_{q_{j3}}) + c_2^2(x_{q_{j3}} - x_{p_{j3}})}{d_2} \\ \omega_2(t,\omega) &= \frac{ac_2^2\Omega_x - ac_2^1\Omega_y}{d_2} \\ c_2^1 &= x_{p_{j1}} - x_{p_{j3}} + \omega(x_{q_{j1}} - x_{p_{j1}}) - a\Omega_x t \\ c_2^2 &= y_{p_{j1}} - y_{p_{j3}} + \omega(y_{q_{j1}} - y_{p_{j1}}) - a\Omega_y t \\ d_2 &= a\Omega_x(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y(x_{q_{j3}} - x_{p_{j3}}) \end{aligned}$$

- The exact solution on the interface $I_{j2}^{n+\frac{1}{2}}$ is:

$$v_h^{n+\frac{1}{2}}(t,\omega_{j2}) = \begin{cases} u_i^n(x_2(t,\omega),y_2(t,\omega)), & \text{if } (t,\omega)\in\mathcal{D}^2_{i,2} \\ v_{j3}^{n+\frac{1}{2}}(t_3(t,\omega),\omega_3(t,\omega)), & \text{if } (t,\omega)\in\mathcal{D}^2_{j3,2} \end{cases}$$

with:

$$\begin{aligned} x_2(t,\omega) &= x_{p_{j2}} + \omega(x_{q_{j2}} - x_{p_{j2}}) - a\Omega_x(t-t^n) \\ y_2(t,\omega) &= y_{p_{j2}} + \omega(y_{q_{j2}} - y_{p_{j2}}) - a\Omega_y(t-t^n) \\ t_3(t,\omega) &= \frac{(c_3^1(y_{p_{j3}} - y_{q_{j3}}) + c_3^2(x_{q_{j3}} - x_{p_{j3}})}{d_3} \\ \omega_3(t,\omega) &= \frac{ac_3^2\Omega_x - ac_3^1\Omega_y}{d_3} \\ c_3^1 &= x_{p_{j2}} - x_{p_{j3}} + \omega(x_{q_{j2}} - x_{p_{j2}}) - a\Omega_x t \\ c_3^2 &= y_{p_{j2}} - y_{p_{j3}} + \omega(y_{q_{j2}} - y_{p_{j2}}) - a\Omega_y t \\ d_3 &= a\Omega_x(y_{q_{j3}} - y_{p_{j3}}) - a\Omega_y(x_{q_{j3}} - x_{p_{j3}}) \end{aligned}$$

Projection step: The exact solutions are respectively projected onto the spaces $\mathbb{P}_C^k$ and $\mathbb{P}_I^k$: in the cell T_i at time t^{n+1}, the exact solution is projected in $\mathbb{P}_C^k$, and on the interfaces $I_j^{n+\frac{1}{2}}$, the solution is projected in $\mathbb{P}_I^k$. For the case n°1 explicit, this leads to the two minimization problems:

$$\|u_i^{n+1} - u_i^h(t^n + \Delta t)\|_i = \inf_{p\in\mathbb{P}_C^k} \|p - u_i^h(t^n + \Delta t)\|_i, \tag{20}$$

$$\|v_{j3}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j3})\|^n = \inf_{q\in\mathbb{P}_I^k} \|q - v_h^{n+\frac{1}{2}}(\omega_{j3})\|^n, \tag{21}$$

for the case n°2 explicit, it leads to the three minimization problems:

$$\|u_i^{n+1} - u_i^h(t^n + \Delta t)\|_i = \inf_{p\in\mathbb{P}_C^k} \|p - u_i^h(t^n + \Delta t)\|_i, \tag{22}$$

$$\|v_{j1}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j1})\|^n = \inf_{q\in\mathbb{P}_I^k} \|q - v_h^{n+\frac{1}{2}}(\omega_{j1})\|^n, \tag{23}$$

$$\|v_{j2}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j2})\|^n = \inf_{q\in\mathbb{P}_I^k} \|q - v_h^{n+\frac{1}{2}}(\omega_{j2})\|^n., \tag{24}$$

and for the case explicit two targets, it also leads to three minimization problems:

$$\|u_i^{n+1} - u_i^h(t^n + \Delta t)\|_i = \inf_{p\in\mathbb{P}_C^k} \|p - u_i^h(t^{n+1})\|_i, \tag{25}$$

$$\|v_{j1}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j1})\|^n = \inf_{q\in\mathbb{P}_I^k} \|q - v_h^{n+\frac{1}{2}}(\omega_{j1})\|^n, \tag{26}$$

$$\|v_{j2}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j2})\|^n = \inf_{q\in\mathbb{P}_I^k} \|q - v_h^{n+\frac{1}{2}}(\omega_{j2})\|^n. \tag{27}$$

Thanks to the Petrov-Galerkin conditions, we can rewrite (20)-(21) as:

$$< u_i^{n+1} - u_i^h(t^n + \Delta t), \left(\frac{x - x_i}{V_i}\right)^l \left(\frac{y - y_i}{V_i}\right)^m >_i = 0 \quad \forall l + m = 0, 1, \ldots, k, \tag{28}$$

$$< v_{j3}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j3}), \left(\frac{t - t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = 0 \quad \forall l + m = 0, 1, \ldots, k. \tag{29}$$

and (22)-(23)-(24) as:

$$< u_i^{n+1} - u_i^h(t^n + \Delta t), \left(\frac{x - x_i}{V_i}\right)^l \left(\frac{y - y_i}{V_i}\right)^m >_i = 0 \quad \forall l + m = 0, 1, \ldots, k, \tag{30}$$

$$< v_{j1}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j1}), \left(\frac{t - t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = 0 \quad \forall l + m = 0, 1, \ldots, k. \tag{31}$$

$$< v_{j2}^{n+\frac{1}{2}} - v_h^{n+\frac{1}{2}}(\omega_{j2}), \left(\frac{t - t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = 0 \quad \forall l + m = 0, 1, \ldots, k. \tag{32}$$

Using the decomposition of the solutions in their corresponding basis:

$$\begin{aligned}
u_i^{n+1}(x,y) &= \sum_{p+q=0}^{k} \alpha_i^{n+1,p,q} \left(\frac{x-x_i}{V_i}\right)^p \left(\frac{y-y_i}{V_i}\right)^q, \\
v_{j1}^{n+\frac{1}{2}}(t,\omega) &= \sum_{p+q=0}^{k} \beta_{j1}^{n+\frac{1}{2},p,q} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega^q, \\
v_{j2}^{n+\frac{1}{2}}(t,\omega) &= \sum_{p+q=0}^{k} \beta_{j2}^{n+\frac{1}{2},p,q} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega^q, \\
v_{j3}^{n+\frac{1}{2}}(t,\omega) &= \sum_{p+q=0}^{k} \beta_{j3}^{n+\frac{1}{2},p,q} \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega^q,
\end{aligned}$$

the minimization problems (28)-(29) are equivalent to solving the two following linear systems of size $\frac{(k+1)(k+2)}{2}$:

$$\begin{aligned}
&\sum_{p+q=0}^{k} \alpha_i^{n+1,p,q} < u_i^{n+1}, \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = < u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i \quad \forall l+m=0,\dots,k \\
&\sum_{p+q=0}^{k} \beta_{j3}^{n+\frac{1}{2},p,q} < v_{j3}^{n+\frac{1}{2}}, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = < v_h^{n+\frac{1}{2}}(\omega_{j3}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n \quad \forall l+m=0,\dots,k.
\end{aligned}$$

and (30)-(31)-(32) are equivalent to solving three linear systems of size $\frac{(k+1)(k+2)}{2}$:

$$\begin{aligned}
&\sum_{p+q=0}^{k} \alpha_i^{n+1,p,q} < u_i^{n+1}, \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = < u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i \quad \forall l+m=0,\dots,k \\
&\sum_{p+q=0}^{k} \beta_{j1}^{n+\frac{1}{2},p,q} < v_{j1}^{n+\frac{1}{2}}, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = < v_h^{n+\frac{1}{2}}(\omega_{j1}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n \quad \forall l+m=0,\dots,k, \\
&\sum_{p+q=0}^{k} \beta_{j2}^{n+\frac{1}{2},p,q} < v_{j2}^{n+\frac{1}{2}}, \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = < v_h^{n+\frac{1}{2}}(\omega_{j2}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n \quad \forall l+m=0,\dots,k.
\end{aligned}$$

The matrices are the same that in the case n°1 implicit and the case n°2 explicit. For the right hand sides, we have first for the case n°1 explicit:

$$\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i =& \sum_{p+q=0}^{k} \alpha_i^{n,p,q} J_{ex,i}^{p,q,l,m} + \beta_{j1}^{n+\frac{1}{2},p,q} J_{ex,j1}^{p,q,l,m} + \beta_{j2}^{n+1,p,q} J_{ex,j2}^{p,q,l,m} \\
=& b^i(l,m), \\
< v_h^{n+\frac{1}{2}}(\omega_{j3}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n =& \sum_{p+q=0}^{k} \alpha_i^{n,p,q} K_{ex,i}^{p,q,l,m} + \beta_{j1}^{n+\frac{1}{2},p,q} K_{ex,j1}^{p,q,l,m} + \beta_{j2}^{n+1,p,q} K_{ex,j2}^{p,q,l,m} \\
=& b_{j3}(l,m),
\end{aligned}$$

then for the case n°2 implicit:

$$
\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = & \sum_{p+q=0}^{k} \beta_{j3}^{n+1,p,q} J_{im,j3}^{p,q,l,m} \\
= & b^i(l,m), \\
< v_h^{n+\frac{1}{2}}(\omega_{j1}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = & \sum_{p+q=0}^{k} \alpha_i^{n,p,q} K_{im,i}^{p,q,l,m} + \beta_{j3}^{n+\frac{1}{2},p,q} K_{im,j3}^{p,q,l,m} \\
= & b_{j1}(l,m), \\
< v_h^{n+\frac{1}{2}}(\omega_{j2}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m > = & \sum_{p+q=0}^{k} \alpha_i^{n,p,q} M_{im,i}^{p,q,l,m} + \beta_{j3}^{n+\frac{1}{2},p,q} M_{im,j3}^{p,q,l,m} \\
= & b_{j2}(l,m)
\end{aligned}
$$

and for the case explicit two targets:

$$
\begin{aligned}
< u_i^h(t^n+\Delta t), \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m >_i = & \sum_{p+q=0}^{k} \alpha_i^{n,p,q} J_{ex,i}^{p,q,l,m} + \beta_{j3}^{n+1,p,q} J_{ex,j3}^{p,q,l,m} \\
= & b^i(l,m), \\
< v_h^{n+\frac{1}{2}}(\omega_{j1}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = & \sum_{p+q=0}^{k} \alpha_i^{n,p,q} K_{ex,i}^{p,q,l,m} + \beta_{j3}^{n+\frac{1}{2},p,q} K_{ex,j3}^{p,q,l,m} \\
= & b_{j1}(l,m), \\
< v_h^{n+\frac{1}{2}}(\omega_{j2}), \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m >^n = & \sum_{p+q=0}^{k} \alpha_i^{n,p,q} M_{ex,i}^{p,q,l,m} + \beta_{j3}^{n+\frac{1}{2},p,q} M_{ex,j3}^{p,q,l,m} \\
= & b_{j2}(l,m),
\end{aligned}
$$

where the coefficients are given by:

$$
\begin{aligned}
J_{ex,i}^{p,q,l,m} &= \frac{1}{V_i} \iint_{\mathcal{F}_i} \left(\frac{x_1(t,\omega)-x_i}{V_i}\right)^p \left(\frac{y_1(t,\omega)-y_i}{V_i}\right)^q \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m dx\ dy \\
J_{ex,j1}^{p,q,l,m} &= \frac{1}{V_i} \iint_{\mathcal{F}_{j1}^1} \left(\frac{t_1(x,y)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_1(x,y)^q \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m dx\ dy \\
J_{ex,j2}^{p,q,l,m} &= \frac{1}{V_i} \iint_{\mathcal{F}_{j2}^1} \left(\frac{t_2(x,y)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_2(x,y)^q \left(\frac{x-x_i}{V_i}\right)^l \left(\frac{y-y_i}{V_i}\right)^m dx\ dy \\
K_{ex,i}^{p,q,l,m} &= \frac{1}{\Delta t} \iint_{\mathcal{D}_i} \left(\frac{x_2(t,\omega)-x_i}{V_i}\right)^p \left(\frac{y_2(t,\omega)-y_i}{V_i}\right)^q \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega \\
K_{ex,j1}^{p,q,l,m} &= \frac{1}{\Delta t} \iint_{\mathcal{D}_{j1}^1} \left(\frac{t_3(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_3(t,\omega)^q \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega \\
K_{ex,j2}^{p,q,l,m} &= \frac{1}{\Delta t} \iint_{\mathcal{D}_{j2}^1} \left(\frac{t_4(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_4(t,\omega)^q \left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega
\end{aligned}
$$

$$J_{im,j3}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{T_i}\left(\frac{t_1(x,y)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_1(x,y)^q\left(\frac{x-x_i}{V_i}\right)^l\left(\frac{y-y_i}{V_i}\right)^m dx\ dy$$

$$K_{im,i}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{i,1}^2}\left(\frac{x_1(x,y)-x_i}{V_i}\right)^p\left(\frac{y_1(x,y)-y_i}{V_i}\right)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$K_{im,j3}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{j3,1}^2}\left(\frac{t_2(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_2(t,\omega)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$M_{im,i}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{i,2}^2}\left(\frac{x_2(x,y)-x_i}{V_i}\right)^p\left(\frac{y_2(x,y)-y_i}{V_i}\right)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$M_{im,j3}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{j3,2}^2}\left(\frac{t_3(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_3(t,\omega)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$J_{ex,i}^{p,q,l,m} = \frac{1}{V_i}\iint_{\mathcal{F}_i^2}\left(\frac{x_1(x,y)-x_i}{V_i}\right)^p\left(\frac{y_1(x,y)-y_i}{V_i}\right)^q\left(\frac{x-x_i}{V_i}\right)^l\left(\frac{y-y_i}{V_i}\right)^m dx\ dy$$

$$J_{ex,j3}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{F}_{j3}^2}\left(\frac{t_1(x,y)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_1(x,y)^q\left(\frac{x-x_i}{V_i}\right)^l\left(\frac{y-y_i}{V_i}\right)^m dx\ dy$$

$$K_{ex,i}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{i,1}^2}\left(\frac{x_2(x,y)-x_i}{V_i}\right)^p\left(\frac{y_2(x,y)-y_i}{V_i}\right)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$K_{ex,j3}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{j3,1}^2}\left(\frac{t_2(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_2(t,\omega)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$M_{ex,i}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{i,2}^2}\left(\frac{x_3(x,y)-x_i}{V_i}\right)^p\left(\frac{y_3(x,y)-y_i}{V_i}\right)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

$$M_{ex,j3}^{p,q,l,m} = \frac{1}{\Delta t}\iint_{\mathcal{D}_{j3,2}^2}\left(\frac{t_3(t,\omega)-t^{n+\frac{1}{2}}}{\Delta t}\right)^p \omega_3(t,\omega)^q\left(\frac{t-t^{n+\frac{1}{2}}}{\Delta t}\right)^l \omega^m dt\ d\omega$$

References

1. Abgrall, R., Deconinck, H., Ricchiuto, M., Villedieu, N., On uniformly high-order accurate residual distribution schemes for advection-diffusion, J. Comput. Appl. Math, 215 (2), 547-556 (2008)
2. Abgrall, R., Larat, A., Ricchiuto, M., Tavé, C., A simple construction of very high order non oscillatory compact schemes on unstructured meshes, Comp. & Fluids., 38(7), 1314-1323 (2009)
3. Abgrall, R., Mezine, M., Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems, J. Comput. Phys., 188, 16-55 (2003)
4. Ben-Artzi, M., Falcovitz, J., Generalized Riemann Problems in Computational Fluid Dynamics, Cambridge Monographs on Applied and Computational Mathematics (2003)
5. Ben-Artzi, M., Birman, A., Application of the Generalized Riemann Problem method to 1-D compressible flows with material interfaces, J. Comput. Phys., 65(1), 170-178 (1986)
6. Ben-Artzi, M., Falcovitz, J., Recent developments of the GRP method, JSME International Journal series B, 38(4), 497-517 (1995)
7. Ben-Artzi, M., Li, J., Warnecke, G., A direct Eulerian GRP scheme for compressible fluid flows, J. Comput. Phys., 218, 19-43 (2006)

8. Berthon, C., Turpault, R., Asymptotic preserving HLL schemes, Numerical Methods for Partial Differential Equations, 27(6), 1396-1422 (2011)
9. Chalons, C., Coquel, F., Godlewski, E., Raviart, P.A., Seguin, N., Godunov-type schemes for hyperbolic systems with parameter dependent source. The case of Euler system with friction, Math. Mod. Math. Appl. Sci., 20(11), (2010)
10. Chandrasekhar, S., Radiative Transfer, Dover (1950)
11. Cockburn, B., Shu, C.W., TVD Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework, Math. Comp., 52, 411- (1989)
12. Cockburn, B., Shu, C.W., The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems, J. Comput. Phys., 141 (2), 199-224 (1998)
13. Dai, W.W., Woodward, P.R., Moment preserving schemes for Euler equations, Comp. & Fluids., 46, 186-196 (2011)
14. Davidson, B., Neutron Transport Theory, Clarendon Press Oxford (1957)
15. Degond, P., Pareschi, L., Russo, G., Modeling and computational methods for kinetic equations, Birkhäuser (2004)
16. Dumbser, M., Balsara, D., Munz, C.D., Toro, E.F., A Unified Framework for the Construction of One-Step Finite-Volume and Discontinuous Galerkin Schemes on Unstructured Meshes, J. Comput. Phys., 227 (18), 8209-8253 (2008)
17. Dumbser, M., Enaux, C., Toro, E.F., Finite Volume Schemes of Very High Order of Accuracy for Stiff Hyperbolic Balance Laws, J. Comput. Phys., 227 (8), 3971-4001 (2008)
18. Dumbser, M., Käser, M., Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, J. Comput. Phys., 221(2), 693-723 (2007)
19. Dumbser, M., Munz, C.D., Arbitrary high-order discontinuous Galerkin schemes, Numerical Methods for Hyperbolic and Kinetic Problems, 295-333 (2005)
20. Dumbser, M., Zanotti, O., Very High Order PNPM Schemes on Unstructured Meshes for the Resistive Relativistic MHD Equations, J. Comput. Phys., 228 (18), 6991-7006 (2009)
21. Filbet, F. Jin, S., An asymptotic preserving scheme for the ES-BGK model of the Boltzmann equation, J. Sci. Comput., 46, 204-224 (2011)
22. Frénod, E., Salvarani, F., Sonnendrücker, E., Long time simulation of a beam in a periodic focusing channel via a two-scale PIC-method, Math. Models Methods Appl. Sci., 19(2), 175-197 (2009)
23. Gassner, G., Dumbser, M., Hindenlang, F., Munz, C.D., Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors, J. Comput. Phys., 230, 4232-4247 (2011)
24. Glassey, R.T., The Cauchy problem in kinetic theory, SIAM (1996)
25. Glimm, J., Marshall, G., Plohr, B., A Generalized Riemann Problem for quasi-one-dimensional gas flows, Advances in Applied Mathematics, 5(1), 1-30 (1984)
26. Gosse, L., Toscani, G., Asymptotic-preserving & well-balanced schemes for radiative transfer and the Rosseland approximation, Numer. Math., 98, 223-250 (2003)
27. Gottlieb, S., Shu, C.W., Total Variation Diminishing Runge-Kutta schemes, Math. Comp, 67, 73-85 (1998)
28. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R., Uniformly high-order accurate essentially non-oscillatory schemes, III,J. Comput. Phys., 71(2), 231-303 (1987)
29. Helffer, B., Nier, F., Hypoelliptic estimates and spectral theory for Fokker-Planck operators and Witten laplacians, Springer (2005)
30. Hoteit, H., Ackerer, P., Mosé, R., Erhel, J., Philippe, B., New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes, Internat. J. Numer. Methods Engrg., 61(14), 2566-2593 (2004)
31. Jiang, G.S., Shu, C.W., Efficient Implementation of Weighted ENO Schemes, J. Comput. Phys., 126(1), 202-228 (1996)
32. Jin, S., Efficient Asymptotic-Preserving (AP) Schemes for Some Multiscale Kinetic Equations, SIAM J. Sci. Comp., 21, 441-454 (1999)
33. Klaij, C., van der Vegt, J.J.W., van der Ven, H., Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations, J. Comput. Phys., 217, 589-611 (2006)
34. Lemou, M., Mieussens, L., Time implicit schemes and fast approximations of the Fokker-Planck-Landau equation, SIAM J. Sci. Comput., 27(3), 809-830 (2007)
35. LeVeque, R.J., Finite volume methods for hyperbolic problems, Cambridge University Press (2002)
36. Liu, T.P., Osher, S., Chan, T., Weighted Essentially Non-oscillatory Schemes, J. Comput. Phys., 115, 200-212 (1994)
37. Lowrie, R.B., Roe, P.L., Van Leer, B., A space-time discontinuous Galerkin method for the time accurate numerical solution of hyperbolic conservation laws, proceedings of 12th AIAA CFD conference, 95-1658 (1995)
38. Lörcher, F., Gassner, G., Munz, C.D., A Discontinuous Galerkin Scheme Based on a Space–Time Expansion. I. Inviscid Compressible Flow in One Space Dimension, J. Sci. Comput., 32(2), 175-199 (2007)
39. Mihalas, D., Weibel-Mihalas, B., Foundations of radiation hydrodynamics, Dover (1999)
40. Pomraning, G.C., The equations of radiation hydrodynamics, Pergamon Press (1973)
41. Toro, E.F., Riemann solvers and numerical methods for fluid dynamics. A practical introduction, Springer-Verlag, Berlin (1999)
42. van der Vegt, J.J.W., van der Ven, H., Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows I. general formulation, J. Comput. Phys., 182, 546-585 (2002)