Compact high-order numerical schemes for scalar hyperbolic partial differential equations

G. Capdeville

▶ **To cite this version:**

G. Capdeville. Compact high-order numerical schemes for scalar hyperbolic partial differential equations. Journal of Computational and Applied Mathematics, 2020, 363, pp.171-210. 10.1016/j.cam.2019.05.029 . hal-03143725

HAL Id: hal-03143725
https://hal.science/hal-03143725

Submitted on 25 Oct 2021

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

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

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S037704271930264X
Manuscript_c36873c15686dc0d5b56553250e90905

Compact high-order numerical schemes for scalar hyperbolic partial differential equations.

G. Capdeville *
Email: guy.capdeville@ec-nantes.fr

14th May 2019

Abstract

This work is a comparative study on the numerical behavior of compact high-order schemes for discretizing scalar hyperbolic problems. To this aim, we investigate two possibilities: a high-order Hermitian scheme (HUPS), based upon the scalar variable and its first derivatives in space, and a "classical" high-order upwind scheme (CUPS), based upon the scalar variable and a wide numerical stencil, as a reference scheme. To study the theoretical properties of both schemes we use a spectral analysis based on a Fourier transform of the numerical solution. In a one-dimensional context, this analysis enables us to investigate the impact of the "spurious" components generated by the Hermitian schemes. As demonstrated by the spectral analysis, in some circumstances, this "spurious" component may deeply alter the accuracy of the scheme and even, in some cases, destroy its consistency. Thus, by identifying the salient features of an Hermitian scheme, we try to design efficients Hermitian schemes for multi-dimensional purposes. A two-dimensional Fourier analysis and some scalar numerical tests allow to highlight two high-order versions of an Hermitian scheme (HUPS4,5) of which the numerical properties are compared with a classical fifth-order scheme (CUPS5) based upon a wider stencil.

Keywords: Upwind discretization, Hyperbolicity, high-order accuracy, compact stencil, spurious component, Hermite polynomial.

1 Introduction

Nowadays, many fields of computational fluid dynamics require high-order numerical schemes for modeling the physics: aeroacoustics, magnetohydrodynamics, turbulence modeling, are all domains where a low frequency mean solution coexists with rapidly varying components of very weak amplitude.

*Ecole Centrale de Nantes, LHEEA Lab.(ECN/CNRS), Nantes, France.

© 2019 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

Usually, the simpler way to improve accuracy of a numerical method is to enlarge its numerical stencil. However, the lack of compactness of a stencil may be problematic when introducing boundary conditions or when the mesh is highly distorted; to remedy that, a solution is to reduce the numerical stencil by employing a Hermitian strategy.

The use of Hermite polynomials to generate a compact high-order discretization scheme is not really a novelty. Qiu and al. made a seminal contribution about this matter those last years; see, for example, [1], [2] and [3].

More recently, pursuing their effort in this direction, Qiu and his co-workers, proposed a new class of high-order compact schemes that are formulated on staggered meshes and involve HWENO spatial discretization, [4]. With this latter work, their schemes do not require the use of numerical fluxes while accurately computing smooth and non-smooth numerical solutions. In [5], the authors proposed a different strategy for the spatial reconstruction with a dimension by dimension algorithm when the spatial dimension is higher than one. See also, the work of Zahran and Abdalla, [6], who extended the fifth-order HWENO scheme of Qiu, [1], to seventh-order in combination with a high-order Runge-Kutta positivity-preserving scheme (SSPRK) for the time integration.

Based on a spectral analysis, we propose, in this document, another way to design a high order Hermitian method. This kind of analysis is not a novelty since it was used early on by V. Leer, [7], to construct compact finite-difference schemes for compressible flows. Later on, Lele, [8], and Mahesh, [9], used a spectral analysis for the design of Hermitian schemes for smooth solutions. More recently, Sengupta et al., [10], generated central and upwind compact schemes by using a Von Neumann matrix spectral analysis. However, all this work was limited to a one-dimensional analysis. The spectral analysis we use allows us to analyze two-dimensional schemes.
In addition, upon using the work initiated in [11], we are able to introduce into the method, monotonicity constraints that allow non-monotonous solutions to be effectively treated, without entirely destroying the overall accuracy of the resulting method. The benefits expected from this new approach are manifold:

- A resulting multi-dimensional discretization method that is not restricted to Cartesian geometries.

- A simple introduction of monotonicity constraints into the Hermitian procedure for computing solutions with embedded discontinuities.

- An alternative to the now classic WENO strategy for generating high-order non-oscillatory interpolation polynomials.

In order to achieve these objectives, using the work done in [12], [13] and [11], we have designed and compared, both theoretically and numerically, models whose main characteristics can be summarized as follows:

- A finite-volume formulation.

- A high-order interpolation polynomials based upon the cell-averaged variable and its first derivatives in space.

- A least-square procedure to compute the polynomial coefficients over possible non Cartesian geometries.

- A limitation procedure of the polynomial reconstruction that prevents spurious oscillations in regions of discontinuous solutions and ensures a maximum principle in space without altering the formal accuracy.

- A positivity-preserving Runge-Kutta method (SSPRK) that provides a maximum principle in time.

- A HLL numerical flux to introduce upwind principles into the discretization.

The numerical model resulting from these features is an Hermite Upwind Positive Scheme (HUPS), which is designed to be monotonicity-preserving over compact numerical stencils and of which the accuracy can be varied from fourth-order to sixth-order.

This article is organized as follows:

In a first section, we analyze, in the Fourier space, the spectral properties of several possible Hermitian solutions when discretizing either the one-dimensional or the two-dimensional scalar advection equation. Firstly, in the one-dimensional framework, the truncation error, the phase and amplitude errors, the damping factor and -last but not least -, the "spurious" component of the numerical solution, are the main quantities that enable to assess the peculiarities of an Hermitian method when compared with a more classical discretization scheme, (CUPS). Theoretical details and numerical features concerning the spectral analysis are developed in Appendices B and C.

Next, by using the teachings of the preceding sub-section, we propose three HUPS versions for discretizing two-dimensional scalar conservation laws. The resulting schemes are analyzed and compared by the means of a two-dimensional spectral analysis.

In section 3, we develop the numerical procedure for a practical computation of two-dimensional scalar laws. We also present the extension of the procedure detailed in [11] to generate a monotonicity-preserving scheme. Next, we perform numerical tests to verify the specific choices we have made. Then these results are compared with those obtained from a more classical definition of a high-order scheme (CUPS).

Lastly, conclusions and perspectives about the possible improvements and extensions of this work, are drawn in a final section.

2 Spectral analysis.

The results obtained for this section required an intensive use of the MAPLE-2017.3 symbolic mathematical computer package.
The purpose of this section is twofold: firstly to study the spectral properties of an Hermitian discretization (HUPS) in terms of stability, phase/amplitude errors and consistency; secondly, to compare those latter properties with a more classical discretization scheme (CUPS).

2.1 One-dimensional analysis

For the sake of simplicity, in a one-dimensional context, we suppose that the mesh is uniform: ($\Delta x \equiv \text{Cte} \equiv h$).

For this study, we consider the discretization of the scalar linear advection equation:

$$u_t + \text{au}_x = 0 \quad (a \equiv \text{Cte} > 0) \tag{2.1}$$

This equation is discretized by using the "method of lines" along with a finite-volume formulation.

With the choice of a Classical Upwind Scheme (CUPS), we get then the following ODE in time, at each discrete point, $x = x_i$:

$$\frac{d\bar{u}_i}{\text{dt}} + a\frac{u^L_{i+1/2} - u^L_{i-1/2}}{h} = 0 \tag{2.2}$$

where the pointwise quantities $u^L_{i\pm1/2}$ are drawn from the local interpolation polynomials, $\tilde{u}_{i\pm1}(x)$; for example, $\tilde{u}_i(x)$ is defined over the discrete "volume", $I_i \equiv [x_{i-1/2}, x_{i+1/2}]$ and $u^L_{i+1/2} \equiv \tilde{u}_i(x_{i+1/2})$, is the pointwise value, at the left of the point $x_{i+1/2}$. In addition, $\bar{u}_i$, represents the cell-averaged value of the numerical solution over I_i.

Practically, we define $\tilde{u}_i(x)$ in the following way, for a p-th order spatial accuracy of the resulting scheme:

$$\tilde{u}_i(x) \equiv \bar{u}_i + \sum_{l=1}^{p-1} \frac{1}{h^l l!} \left[(x - x_i)^l - \bar{x}^l_i \right] \times D'_l \tag{2.3}$$

with the following definitions: $\bar{x}^l_i \equiv \dfrac{1}{h}\displaystyle\int_{I_i} (x - x_i)^l \, dx$, to ensure the mean preservation of $\tilde{u}_i(x)$ over I_i, and $D'_l \equiv h^l \dfrac{\partial^l u}{\partial x^l} + \mathcal{O}(h^p)$.

If we adopt a classical choice (CUPS scheme), the polynomial coefficients of $\tilde{u}_i(x)$, D'_l in (2.3), may be computed according to the condition of "mean preservation", which is consistent with the finite-volume formulation:

$$\frac{1}{h}\int_{I_j} \tilde{u}_i(x)dx = \bar{u}_j, \;\; j \in N_i - \{i\} \tag{2.4}$$

where N_i, stands for the numerical stencil over which $\tilde{u}(x)$ is computed. The size of N_i is selected to generate at least as many relationships as unknowns, D'_l.

Now, if we select, instead, an Hermitian procedure to discretize (2.1) (HUPS scheme), firstly we decide to define the following system for the new discrete unknowns ($\bar{u}_i, \bar{r}_i \equiv \displaystyle\int_{I_i} \frac{d\tilde{u}_i}{dx} dx$) or, equivalently $\left(\bar{u}_i, \bar{r}_i \equiv \tilde{u}_i(x_{i+1/2}) - \tilde{u}_i(x_{i-1/2})\right)$, as follows :

$$\begin{cases} \dfrac{d\bar{u}_i}{dt} + a\dfrac{u^L_{i+1/2} - u^L_{i-1/2}}{h} = 0 \\ \dfrac{d\bar{r}_i}{dt} + a\dfrac{r^L_{i+1/2} - r^L_{i-1/2}}{h} = 0 \end{cases} \tag{2.5}$$

with the following pointwise definition: $r^L_{i+1/2} \equiv h \cdot \dfrac{d\tilde{u}_i}{dx}(x_{i+1/2})$.

Secondly, the polynomial coefficients of $\tilde{u}_i(x)$, are computed according to the double conservative conditions:

$$\begin{cases} \dfrac{1}{h}\displaystyle\int_{I_j} \tilde{u}_i(x)dx = \bar{u}_j, j \in N_i - \{i\} \\ \displaystyle\int_{I_j} \frac{d\tilde{u}_i(x)}{dx}dx = \bar{r}_j, j \in N_i \end{cases} \tag{2.6}$$

Consequently, by increasing the number of relationships for the polynomial coefficients, it now becomes possible to reduce the size of N_i for computing the D'_l: doing so, we define a "compact" stencil over which $\tilde{u}_i(x)$ may be computed.

To illustrate this presentation, let us define the CUPS scheme over the numerical stencil, N_i, typified by Figure 1, below:

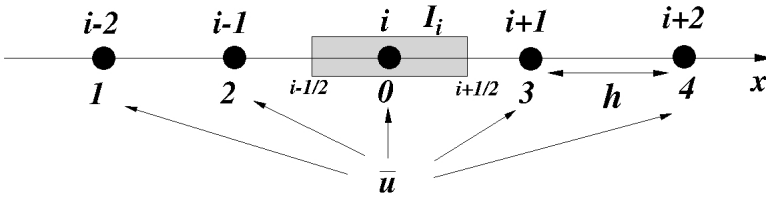


Figure 1: Numerical stencil for the CUPS5 scheme

This symmetrical stencil is selected to generate a polynomial, $\tilde{u}_i(x)$, of degree four ($p = 5$), at best.

Indeed, over this stencil, with five relationships arising from (2.4) for determining the D'_l, one may define, at best, an interpolating polynomial of degree four: the resulting finite-volume form (2.2) becomes, then, fifth-order accurate in space. As a result, we call this solution the "CUPS5" scheme.

Now, if we define $\tilde{u}_i(x)$ over the more compact numerical stencil, Figure 2 above, with an Hermitian definition, then, upon considering the six degrees of freedom given by the variable and its first space derivative, a sixth-order scheme may be defined, at best, over this stencil: this is the "HUPS6 scheme".

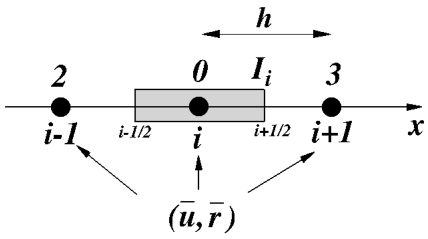


Figure 2: Numerical stencil for the HUPS scheme

However, to be consistent with the two-dimensional computations of the next sub-section, we use also a least-square procedure to define interpolating polynomials of degree three and four: we call the resulting schemes, respectively, the "HUPS4" and "HUPS5" schemes.
Note that, in all what follows, all linear least-square problems are analytically inverted from MAPLE-2017.3 by using a Gram-Schmidt process.

Lastly, if we choose a SSPRK(5,4) time-integration procedure, [14] (see also Appendix A), either for (2.2) (CUPS scheme) or for (2.5) (HUPS), we get the final algebraic form that is fourth-order time accurate and, depending on the form and the algorithm selected, either fourth-order, fifth-order or sixth-order accurate in space.

For the analysis that follows, all the computed quantities are formulated in terms of two main parameters: the number of cells-per-wavelength, N, where $N \times \beta = 2\pi$, (β, phase angle: see Appendix B) for the sinusoidal solution, and the CFL number, ν, where CFL $\equiv \nu = a\Delta t/h$.

To begin this comparative analysis, let us compute the discrete characteristics of the CUPS scheme, for $p = 5$: namely, the CUSP5 scheme. Thus, upon using condition (2.4) over the stencil typified by Figure1, we obtain a 4×4 linear problem of which the solution for the D'_l gives the following pointwise interpolated quantity at $x = x_{i+1/2}$, from the polynomial, $\tilde{u}_i(x)$:

$$\widetilde{u_i}(x_{i+1/2}) \equiv u^L_{i+1/2} = \frac{1}{60}\left[2\bar{u}_{i-2} - 13\bar{u}_{i-1} + 47\bar{u}_i\right] + \frac{1}{20}\left[9\bar{u}_{i+1} - \bar{u}_{i+2}\right] \quad (2.7)$$

If we introduce this latter result into the semi-discrete form, (2.2) and if we use a discrete Fourier transform of the resulting numerical solution, then, we can formulate the scalar "amplification factor", $\mathcal{G}(\beta, \nu)$, for the scheme (see Appendix B), when employing the fourth-order accurate SSPRK(5,4) time-integration algorithm.

This complex amplification factor, $\mathcal{G}(\beta, \nu)$, is the key element to analyse the spectral properties of any numerical scheme (see Appendix B).

Thus, for the CUPS5 scheme, we get the following result for the truncation error, $\tau(\beta,\nu)$, in the Fourier space (see Appendix B):

$$\tau(\beta,\nu) = -\frac{j\nu^5}{120}\left(\frac{\beta^5}{\Delta t}\right) - \frac{\nu}{720}(5\nu^5 - 12)\left(\frac{\beta^6}{\Delta t}\right) + \mathcal{O}\left(\frac{\beta^7}{\Delta t}\right) \tag{2.8}$$

This first result for the CUPS5 scheme gives us many informations:

- The resulting scheme remains fourth-order accurate, overall (term $\frac{\beta^5}{\Delta t}$), due to the time-integration process.

- The leading error generated by this scheme is a phase error.

- The term that quantifies this phase error (see Appendix B), may be identified to give:

$$a_1(\nu) \equiv \frac{1}{120}\nu^5 \tag{2.9}$$

- The second term in (2.8), which is a fifth-order term, gives us informations about stability of the scheme. Indeed, if we formulate the modified equation of (2.1) by the CUPS5 scheme, we get the following result (see Appendix B):

$$u_t + \mathrm{a}u_x = \frac{\nu^5}{120}\left(\frac{\Delta x^5}{\Delta t}\right)\frac{\partial^5 u}{\partial x^5} - \frac{\nu}{720}(5\nu^5 - 12)\left(\frac{\Delta x^6}{\Delta t}\right)\frac{\partial^6 u}{\partial x^6} + \mathcal{O}\left(\frac{\Delta x^7}{\Delta t}\right) \tag{2.10}$$

Since the high-order quantity, $a_2(\nu) \equiv \frac{\nu}{720}(5\nu^5 - 12)$, is always negative, whatever $\nu \in [0,1]$, we deduce that the quantity in front of $\frac{\partial^6 u}{\partial x^6}$, in (2.10), is always positive and, therefore, it introduces a slight source of damping (damping of the round-off errors) into the computation of the numerical solution: there are no numerical sources of instabilities, even minor, within the CUPS5 scheme.

Figure 3, which plots the modulus of the amplification factor, $\mathcal{G}(\beta, \nu = 3/4)$, for N varying, illustrates this first finding:

As one can see it, the amplification factor is everywhere lower than one ($|\mathcal{G}(\beta,\nu = 3/4)| \equiv |\lambda| < 1$), for a Courant number of $3/4$ and its lowest value is obtained for $N = 5$ points/wavelength, with ($|\mathcal{G}(\beta = 2\pi/5, \nu = 3/4)| = 0.967$.
In addition, upon using the computation of $\mathcal{G}(\beta,\nu)$, we can plot numerical errors for $\nu \in [0,1]$ and $N \in [5,20]$ (see Appendix B).
Figures 4 that follow, display those first results.

As one can see from these results, for $\nu = 1/2$, for example, less than 10 cells-per-wavelength are necessary to get an accuracy level of 1×10^{-3}, both in amplitude and phase error.
Also, we can notice that the best results are obtained for $\nu < 1/2$; this behavior is particularly sensitive in what concerns the phase error, Figure 4b.

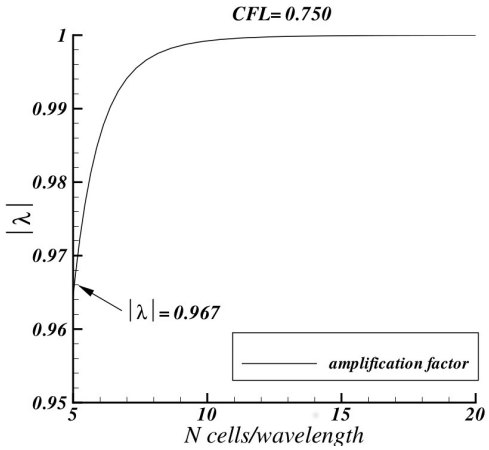
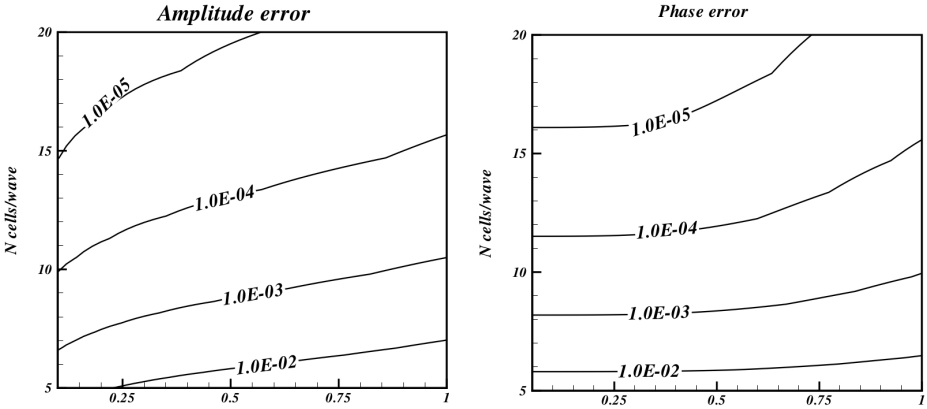


Figure 3: CUPS5 scheme: stability range for $\nu = 0.75$



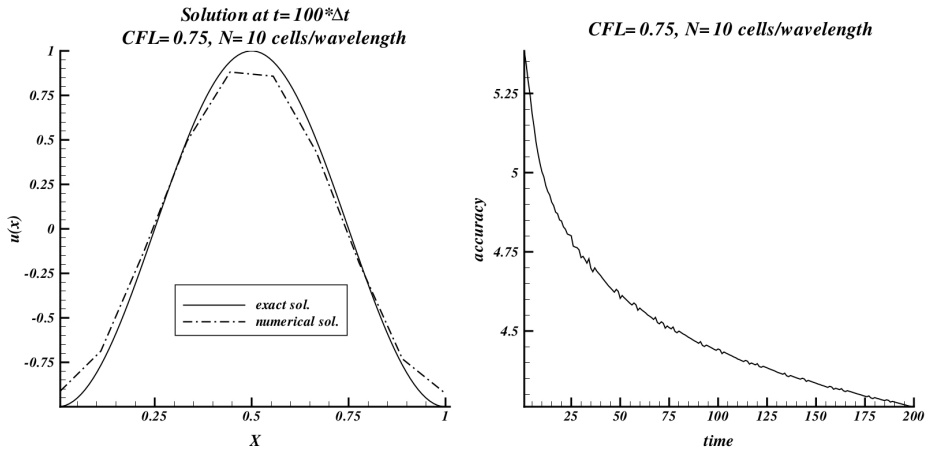
(a) Amplitude error, $A(\beta, \nu)$ (b) Phase error, $P(\beta, \nu)$

Figure 4: CUPS5 scheme errors, $A(\beta, \nu)$ and $P(\beta, \nu)$

Now, if we compute the time-evolution of an initial sinusoidal solution, $u(x, t = 0) = sin2\pi x, \forall x \in [0, 1]$, by using the relationship for the first component of equation (B.28), we get the results displayed by Figure 5a, at $t = 100\Delta t$ and for the choices: $\nu = 3/4$, $N = 10$ cells-per-wavelength; similarly, Figure 5b gives us the accuracy time-evolution, computed from the definition given in Appendix B: the accuracy converges asymptotically towards the theoretical value, 4. Practically, this latter result tells us that there are no numerical instabilities that arise during the time-evolution procedure and that might alter the scheme accuracy.
Thus, those results gives us two significant informations about the spectral beha-

vior of the CUPS5 scheme when employing a moderate space resolution ($N = 10$ points per wavelength):

- The dispersive behavior of the scheme is slight, although noticeable.
- The dissipative behavior of the scheme is very weak.



(a) Solution at $t = 100\Delta t$

(b) accuracy time-evolution

Figure 5: CUPS5 scheme: time-evolution of the solution, $u(x, t = 0) = sin2\pi x, \forall x \in [0, 1]$, and accuracy, for $\nu = 3/4, N = 10$ cells/wavelength, at $T = 100\Delta t$.

Now, let us study the spectral characteristics of the first Hermitian scheme: the HUPS4 scheme. This scheme is obtained for $p = 4$ in equation (2.3) and over the stencil pictured by Figure 2: in that case, we obtain a linear system of 5 equations for the 3 unknowns, D'_l, in equation (2.3); this system is then solved by the Gramm-Schmidt method from MAPLE-2017.3. This way, we get the following pointwise interpolated quantities at $x = x_{i+1/2}$, from the Hermite polynomial, $\tilde{u}_i(x)$:

$$\begin{cases} \widetilde{u_i}(x_{i+1/2}) \equiv u^L_{i+1/2} = \frac{1}{30}\left[\frac{11}{2}\bar{u}_{i+1} + 29\bar{u}_i\right] - \frac{3}{20}\bar{u}_{i-1} - \frac{1}{45}\left[\bar{r}_{i+1} + 4\bar{r}_{i-1}\right] + \frac{5}{18}\bar{r}_i \\ \\ \widetilde{r_i}(x_{i+1/2}) \equiv r^L_{i+1/2} = \frac{1}{120}\left[47\bar{u}_{i+1} - 23\bar{u}_{i-1}\right] - \frac{1}{5}\bar{u}_i + \frac{1}{360}\left[67\bar{r}_{i+1} - 77\bar{r}_{i-1}\right] + \frac{4}{9}\bar{r}_i \end{cases} \tag{2.11}$$

The truncation error, in the Fourier space, may be computed from these results and from the computation of the "accurate" eigenvalue, $\lambda_1(\beta, \nu)$, of the "amplification matrix", $\mathcal{G}(\beta, \nu)$ (see Appendix B), to give the first useful result for the HUPS4 scheme:

$$\tau(\beta,\nu) = -\frac{j\nu\left(21\nu^4 - 89\right)}{2520}\left(\frac{\beta^5}{\Delta t}\right) - \frac{\nu}{176400}(245\nu^5 - 6230\nu - 4101)\left(\frac{\beta^6}{\Delta t}\right) + \mathcal{O}\left(\frac{\beta^7}{\Delta t}\right) \tag{2.12}$$

This latter result gives us the following informations:

- The resulting scheme is designed to be fourth-order accurate, overall (term $\frac{\beta^5}{\Delta t}$).
- The leading error generated by this scheme is a phase error.
- The term that quantifies this phase error, is:

$$a_1(\nu) \equiv \frac{\nu}{2520}\left(21\nu^4 - 89\right)$$

This latter quantity is always negative, whatever $\nu \in [0,1]$.
In addition,the modified equation of (2.1) by the HUPS4 scheme, is as follows:

$$\begin{aligned} u_t + au_x = & \frac{\nu\left(21\nu^4 - 89\right)}{2520}\left(\frac{\Delta x^5}{\Delta t}\right)\frac{\partial^5 u}{\partial x^5} \\ & - \frac{\nu}{176400}(245\nu^5 - 6230\nu - 4101)\left(\frac{\Delta x^6}{\Delta t}\right)\frac{\partial^6 u}{\partial x^6} + \mathcal{O}\left(\frac{\Delta x^7}{\Delta t}\right) \end{aligned} \tag{2.13}$$

The function $a_2(\nu) \equiv \frac{\nu}{176400}(245\nu^5 - 6230\nu - 4101)$ being always negative, $\forall \nu \in [0,1]$, the HUPS4 scheme does not present numerical source of instabilities, even minor, in its design (see Appendix B).

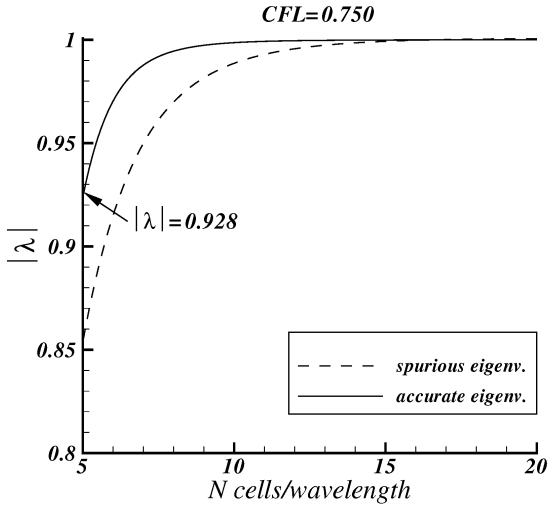
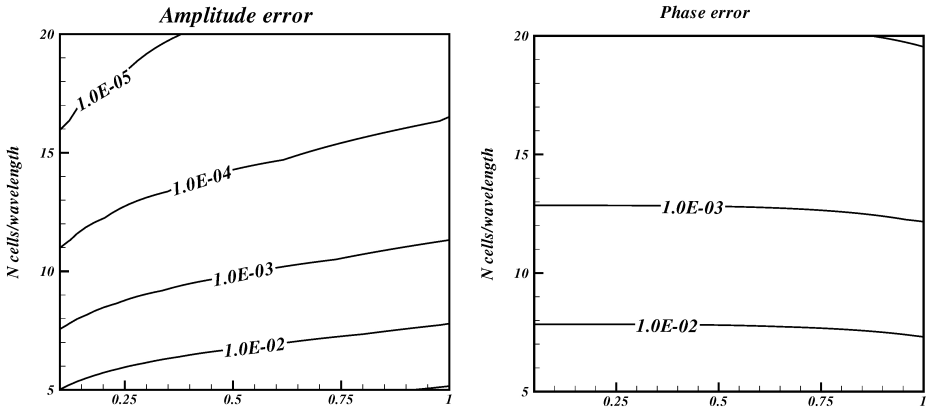


Figure 6: HUPS4 scheme: stability range for $\nu = 0.75$

Figure 6, illustrates the amplification factor modulus, $|g(\beta,\nu)|$, (see appendix B) computed for the HUPS4 scheme with $\nu = 3/4$.

This plot gives us more informations than that obtained with the CUPS5 scheme.
Firstly, we see that the HUPS4 scheme remains linearly stable, whatever the number of cells per wave length, N, used; the minimum value for the accurate eigenvalue is obtained for $N = 5$ with $|\lambda_1(\beta = 2\pi/5, \nu = 0.75)| = 0.928$: this value is lower than that obtained by using the CUPS5 scheme and, consequently, the damping of the high frequencies in the numerical solution (highly varying components of the solution), will be sligthly stronger with this space resolution. Moreover, Figure 6 shows us that the modulus of the "spurious" eigenvalue is below 1, whatever N: this means, in particular, that the consistency of the numerical scheme cannot be destroyed by the spurious component of the solution (see equ. (B.34), Appendix B).

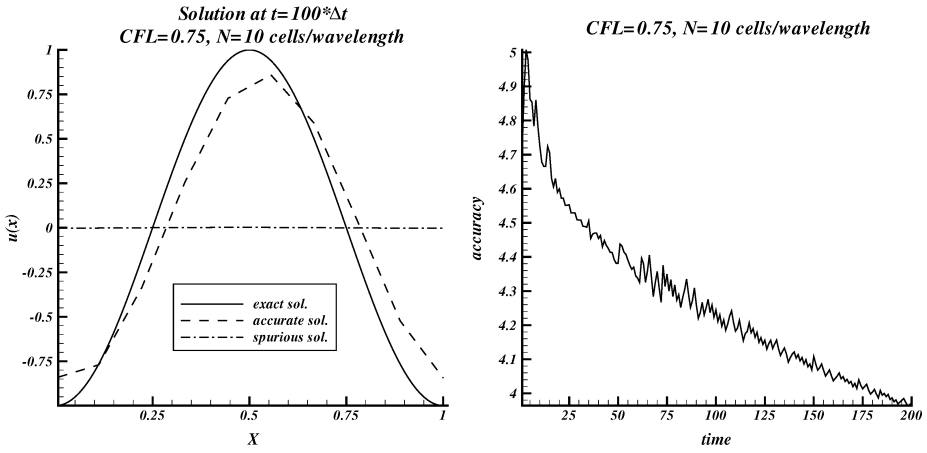
Now, if we look at the amplitude and phase errors of the HUPS4 scheme, we get numerical results displayed by Figures 7 that follow:



(a) Amplitude error, $A(\beta, \nu)$ (b) Phase error, $P(\beta, \nu)$

Figure 7: HUPS4 scheme errors, $A(\beta, \nu)$ and $P(\beta, \nu)$

Clearly, by comparing these results with those obtained in Figures 4, it appears that the HUPS4 scheme is largely less accurate, in phase errors, than the CUPS5 scheme: the higher accuracy of the interpolation polynomial, $\tilde{u}_i(x)$, seems to have a significant role to explain this first result.
On the other side, if we look at the time evolution of the numerical solution, we get results illustrated by Figure 8, below, for $\nu = 0.75$ and $N = 10$ cells per wavelength.



(a) Solution at $t = 100\Delta t$ (b) accuracy time-evolution

Figure 8: HUPS4 scheme: time-evolutions, at $T = 100\Delta t$, of solution $u(x, t = 0) = sin2\pi x, \forall x \in [0, 1]$, and accuracy, for $\nu = 0.75$ and $N = 10$ cells/wavelength.

The phase error of the numerical solution is obvious by examining Figure 8a. However, one can note that the spurious component of the solution is everywhere 0, Figure 8a: this means that this component has no influence over the accuracy of the numerical solution, during the computation time.
Indeed, according to equation (B.34) of Appendix B, the numerical solution, $U_i^n \equiv [u_i^n, r_i^n]^t$, of the discretized form of (2.5), may be written as follows, after a Fourier transform of the resulting algebraic form:

$$U_i^n = \left(\lambda_1^n \hat{w}_1^0 r_1 + \lambda_2^n \hat{w}_2^0 r_2\right) e^{jkx_i} \tag{2.14}$$

where, (λ_p, r_p), represent, respectively, the $p-th$ eigenvalue and its associated right eigenvector of the complex "amplification matrix", $\mathcal{G}(\beta, \nu)$, coming from the Fourier transform of the algebraic form (Appendix B); $\hat{w}_p^0$ is the $p-th$ complex amplitude associated to λ_p and it is computed from the initial solution of (2.1).
Thus, if the eigenvalue, $\lambda_1(\beta, \nu)$, is defined as the consistent approximation of the exact amplification factor, $e^{-j\nu\beta}$, of (2.1) (see Appendix B), one may say that the numerical solution at $t = t_n$, is the sum of two components: an "accurate" component, $\lambda_1^n \hat{w}_1^0 r_1$, which should be compared to the exact solution of the Hermitian form of (2.1), and a "spurious" component, $\lambda_2^n \hat{w}_2^0 r_2$, that does not have counterpart in the differential problem; the amplitude of this latter component is given by the modulus of the complex quantity, $\hat{w}_2^0$.
Numerical results illustrated by Figure 8a show us that the modulus of this "spurious" component is zero, everywhere.
Besides, the accuracy of the computations is not far from the theoretical value

of 4, when time evolves, Figure 8b.

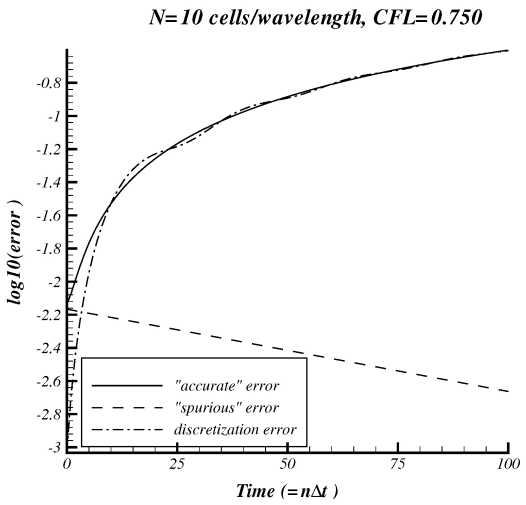


Figure 9: HUPS4 scheme: time-evolution of errors for $\nu = 0.75$, $N = 10$ and $\bar{r}_i^0 = Im\left\{j\beta \times e^{j2\pi x_i}\right\}$

In addition, it is possible to compute the time-evolution of the "spurious" and "accurate" discretization errors, by making the comparison between the exact solution of (2.1) and, respectively, the spurious and accurate components of this solution (see Appendix B, equ.(B.34) and associated definitions). Figure 9, displays this result.

As one can see, the spurious error generated, at $t = 0$, by an exact computation of the space derivative ($\bar{r}_i^0 = Im\left\{j\beta \times e^{j2\pi x_i}\right\}$, see Appendix B) is not negligible. However, because $|\lambda_2|(\beta, \nu) < 1$, this spurious error is damped as the time evolves and the "accurate" error swiftly prevails; in that case, the discretization error closely follows the time-evolution of this "accurate" error and the numerical solution has the spectral properties given by the "accurate" eigenvalue, $\lambda_1(\beta, \nu)$.

This result is interesting because it allows to understand the temporal behavior of the numerical solution of an Hermitian scheme and the influence of the initial error on this behavior.

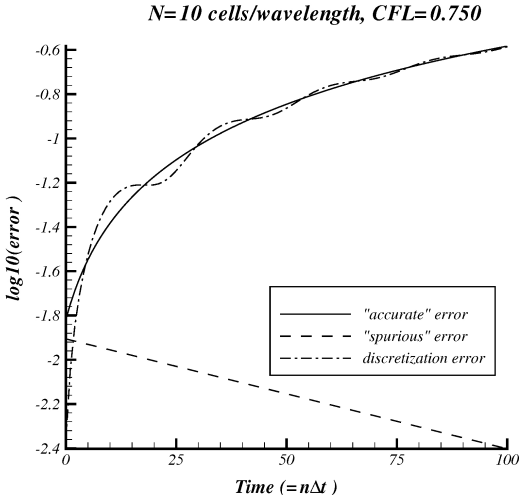


Figure 10: HUPS4 scheme: time-evolution of errors for $\nu = 0.75$, $N = 10$ and $\bar{r}_i^0 = Im\left\{jsin\beta \times e^{j2\pi x_i}\right\}$

For example, if we decide to use a centered initialization of the space derivative by setting: $\bar{r}_i^0 \equiv Im\left\{jsin\beta \times e^{j2\pi x_i}\right\}$, then, we get the new results displayed by Figure 10.

Now, the "spurious" error generated by this initialization, is influent, even for $t > 0$, and it significantly alters the discretization error, Figure 10. But, once again, thanks to $|\lambda_2(\beta, \nu)| < 1$, the discretization error ultimately follows the "accurate" error as time evolves.

Let us pursue this analysis by studying the numerical behavior of the HUPS5 scheme.

This scheme is obtained for $p = 5$ in equation (2.3): in that case, we obtain a linear system of 5 equations for the 4 unknowns, D'_l, in equation (2.3); this system remains solved by a least-square procedure and we get the following pointwise interpolated quantities at $x = x_{i+1/2}$, as result:

$$\begin{cases} \widetilde{u_i}(x_{i+1/2}) \equiv u^L_{i+1/2} = \frac{1}{30}\left[13\bar{u}_{i+1} - 2\bar{u}_{i-1} + 19\bar{u}_i\right] - \frac{1}{30}\left[\bar{r}_{i-1} + 4\bar{r}_{i+1} - 5\bar{r}_i\right] \\ \\ \widetilde{r_i}(x_{i+1/2}) \equiv r^L_{i+1/2} = \frac{1}{2}\left[3\bar{u}_{i+1} + \bar{u}_{i-1} - 4\bar{u}_i\right] + \frac{1}{6}\left[\bar{r}_i + \bar{r}_{i-1} - 2\bar{r}_{i+1}\right] \end{cases} \tag{2.15}$$

Then, the truncation error may be computed to give the following result for the

HUPS5 scheme:

$$\tau(\beta,\nu) = -\frac{j\nu^5}{120}\left(\frac{\beta^5}{\Delta t}\right) - \frac{1}{720}\nu(\nu^5-2)\left(\frac{\beta^6}{\Delta t}\right) + \mathcal{O}\left(\frac{\beta^7}{\Delta t}\right) \tag{2.16}$$

To begin, a first interesting result may be noted: the truncation error is very similar to the truncation error obtained for the CUPS5 scheme, equ.(2.8).
Indeed, only the second term typified by the function, $a_2(\nu)$, is slightly different from that obtained for the CUPS5 scheme since we have, now: $a_2(\nu) \equiv \frac{1}{720}\nu(\nu^5-2)$; this quantity remains negative, whatever $\nu \in [0,1]$.
Therefore, the HUPS5 scheme does not present high-order source of instabilities and moreover, by considering the truncation error, one may expect accuracy properties very close to those of the CUPS5 scheme.

This latter remark is confirmed by results of Figure 11 that follows.
Indeed, the modulus of the amplification factor is sligthly larger than that of the CUPS5 scheme; for $N = 5$ cells-per-wavelength and $\nu = 0.75$, we get: $|g(\beta = 2\pi/5, \nu = 0.75)| = 0.991$.

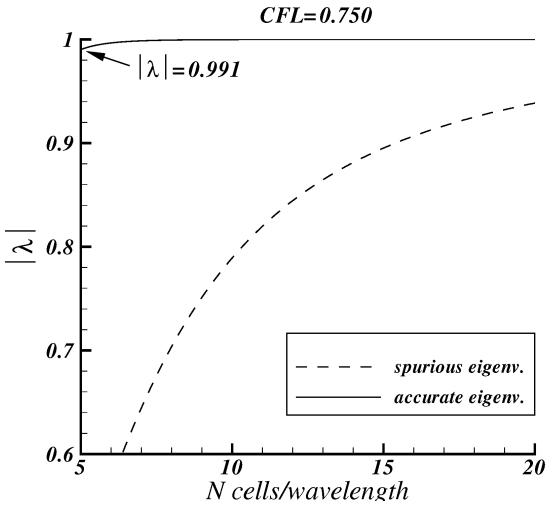
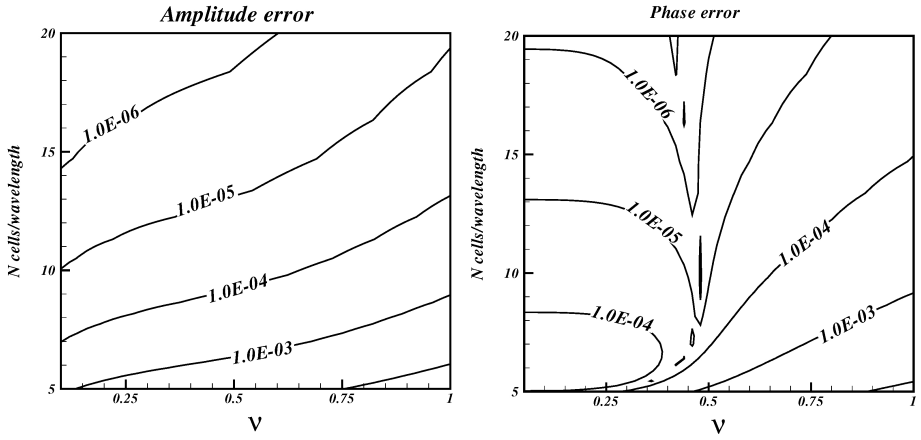


Figure 11: HUPS5 scheme: stability range for $\nu = 0.75$

In addition, compared to Figure 6 for the HUPS4 scheme, the modulus of the "spurious" eigenvalue is substantially lower; consequently, the influence of the spurious component of the solution will be damped more swiftly than with the HUPS4 scheme.
Now, if we look at the amplitude error, $A(\beta, \nu)$, and phase error, $P(\beta, \nu)$, we get results displayed by Figures 12.

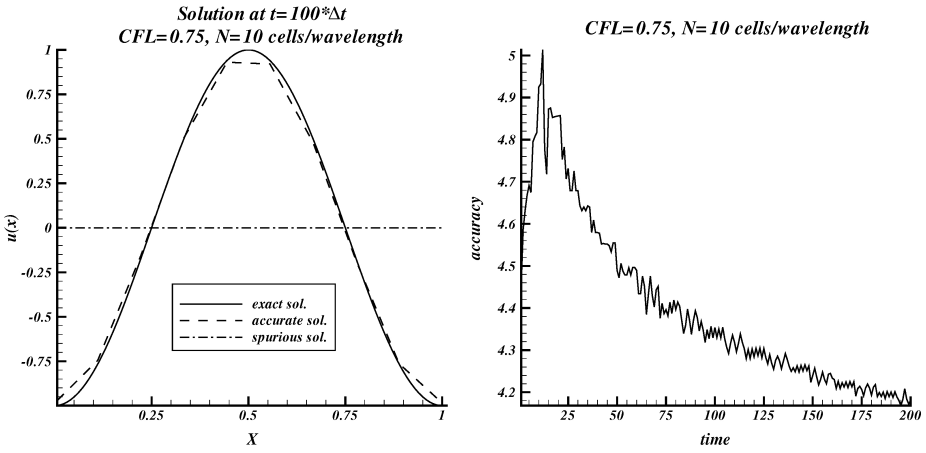


(a) Amplitude error, $A(\beta, \nu)$ (b) Phase error, $P(\beta, \nu)$

Figure 12: HUPS5 scheme errors, $A(\beta, \nu)$ and $P(\beta, \nu)$

The results for the amplitude and phase errors, Figure 12a and Figure 12b, respectively, are lower by an order of magnitude than that obtained with the CUPS5 scheme, Figure 7a and Figure 7b; for example, with $\nu = 1/2$ and 10 cells-per-wavelength, we get an error level below 1×10^{-4}. However, one can note that the phase error of the HUPS5 scheme is more sensitive to the CFL number, Figure 12b, and values higher than 0.5 generate a significant increase of the phase error.

Figures 13 that follow present the time-evolution of the solution and of the effective accuracy.



(a) Solution at $t = 100\Delta t$ (b) accuracy time-evolution

Figure 13: HUPS5 scheme: time-evolutions, at $T = 100\Delta t$, of solution, $u(x, t = 0) = sin2\pi x$, and accuracy, for $\nu = 0.75, N = 10$ cells/wavelength and $\bar{r}_i^0 = Im\left\{j\beta \times e^{j2\pi x_i}\right\}$.

The numerical accuracy asymptotically converges towards the value 4, Figure 13b. The numerical solution, Figure 13a, does not exhibit amplitude and phase errors as substantial as that obtained with the CUPS5 scheme, Figure 5.
Compared to results obtained with HUPS4, Figure 8, these results are better for the solution and almost identical for the effective accuracy.
If we look at the time evolution of the errors, we get results displayed by Figure 14.

As one can see, the spurious error decreases very rapidly with, as consequence, a discretization error identical to the "accurate" error: the advantage in having a spurious eigenvalue, $\lambda_2(\beta, \nu)$, of which the modulus is much smaller than 1, is obvious, here.

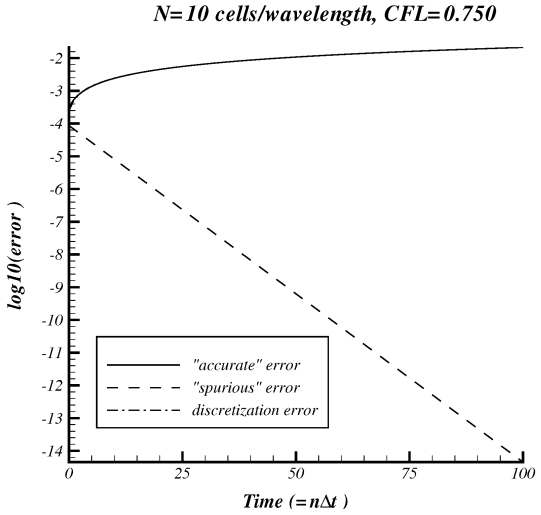


Figure 14: HUPS5 scheme: time-evolution of errors for $\nu = 0.75$, $N = 10$ and $\bar{r}_i^0 = Im\left\{j\beta \times e^{j2\pi x_i}\right\}$

Lastly, if we compute the numerical solution with a centered approximation of the initial space derivative, $\bar{r}_i^0$, the numerical results for the errors are exactly the same as those displayed by Figure 14: the HUPS5 scheme is insensitive to the way the first space derivative is initialized.

To end with this one-dimensional spectral analysis, it is left to consider the best obtainable accuracy for an Hermite polynomial, on the stencil pictured by Figure 2. This accuracy is obtained for $p = 6$ in equ. (2.3). This way, we have to solve a linear system of 5 equations for the 5 unknowns, D'_l, which is directly inverted. The result gives the following pointwise interpolated quantities at $x = x_{i+1/2}$:

$$\begin{cases} \widetilde{u_i}(x_{i+1/2}) \equiv u^L_{i+1/2} = \frac{1}{60}\left[11\left(\bar{u}_{i+1} + \bar{u}_{i-1}\right) + 38\bar{u}_i\right] + \frac{1}{20}\left[40\bar{r}_i + \bar{r}_{i-1} - \bar{r}_{i+1}\right] \\ \widetilde{r_i}(x_{i+1/2}) \equiv r^L_{i+1/2} = \frac{1}{4}\left[7\bar{u}_{i+1} - 8\bar{u}_i + \bar{u}_{i-1}\right] + \frac{1}{12}\left[\bar{r}_{i-1} - 5\bar{r}_{i+1} - 2\bar{r}_i\right] \end{cases} \tag{2.17}$$

And the truncation error, in the Fourier space, writes as:

$$\tau(\beta,\nu) = -\frac{j\nu^5}{120}\left(\frac{\beta^5}{\Delta t}\right) - \frac{\nu^6}{720}\left(\frac{\beta^6}{\Delta t}\right) + \mathcal{O}\left(\frac{\beta^7}{\Delta t}\right) \tag{2.18}$$

Once again, since we used a fourth-order time-integration procedure, the resulting scheme is fourth-order, overall, although the interpolation polynomial, $\tilde{u}_i(x)$, is a fifth degree polynomial.

This truncation error is very similar to that obtained with the CUPS5 or the HUPS5 scheme, previously studied. However, one must note an interesting difference: the function $a_2(\nu) \equiv \frac{\nu^6}{120}$, is always a positive quantity, whatever $\nu \in [0,1]$; according to the form of the modified equation linked to this truncation error (see Appendix B), this means that instabilities may appear, since the second term of the truncation error becomes, then, an amplification term.
Of course, this does not mean that the resulting scheme is linearly unstable, since the first term of the truncation error, which introduces phase errors into the computations, prevails, but, due to this second term, some minor instabilities may appear and alter the convergence rate of the numerical algorithm.

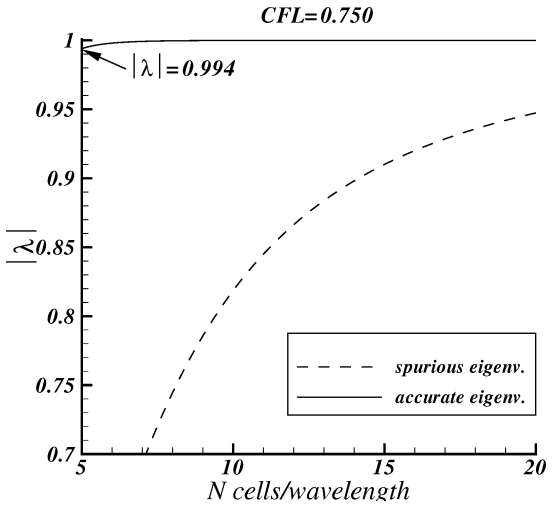
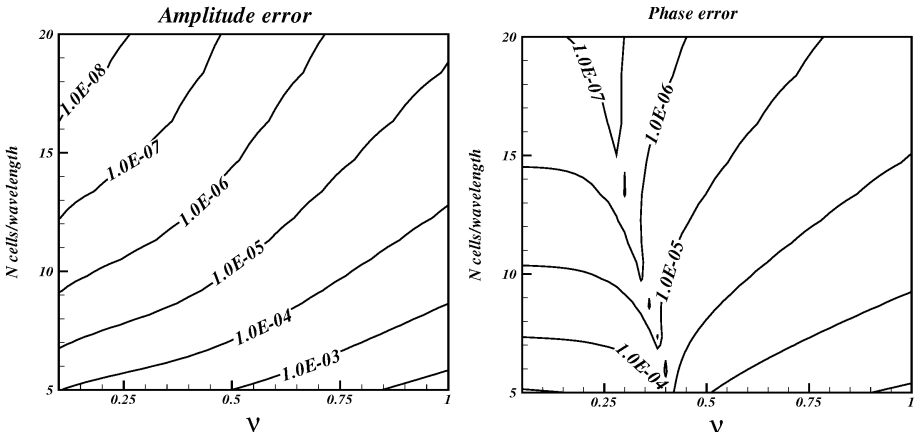


Figure 15: HUPS5 scheme: stability range for $\nu = 0.75$

Figure 15, gives us the first informations concerning the spectral properties of the HUPS6 scheme. The amplification factor modulus is almost identical to that of the HUPS5 scheme, Figure 11. Indeed, we have $|g(\beta = 2\pi/5, \nu = 0.75)| = 0.994$; similarly, the size of the modulus of the "spurious" eigenvalue is the same as that of the HUPS5 scheme.

If we consider, now, the Figures for the amplitude and phase errors, the improvments in accuracy are obvious by comparing with Figure 7 (HUPS4) or Figure 12 (HUPS5).

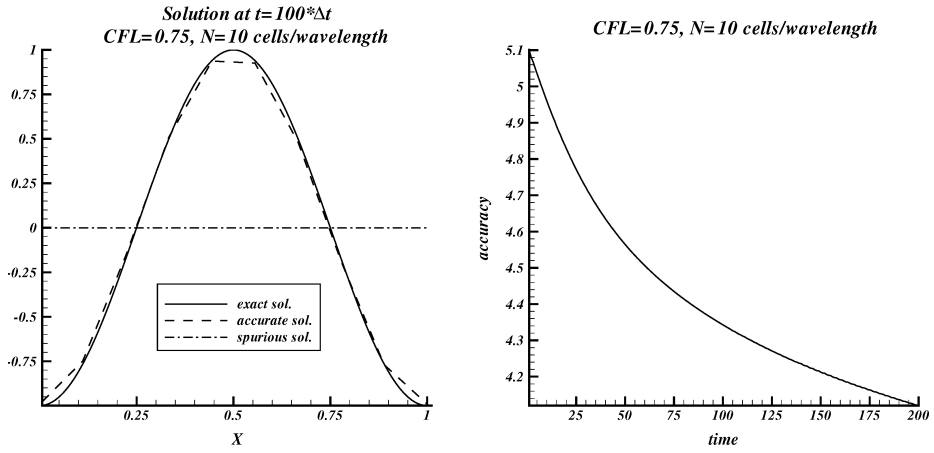


(a) Amplitude error, $A(\beta, \nu)$ (b) Phase error, $P(\beta, \nu)$

Figure 16: HUPS6 scheme errors, $A(\beta, \nu)$ and $P(\beta, \nu)$

Indeed, as one can see from Figure 16a, the amplitude error is lower by more than an order than that of Figure 12a (HUPS5); the phase error, Figure 16b, has a similar behavior but it is more sensitive to the CFL number: the phase error is very low in the range $[0.30, 0.40]$, whereas this quantity rapidly grows up as soon as $\nu > 1/2$.
This latter result means that the better accuracy properties of the scheme are obtained at the cost of a smaller CFL number: this is what we may consider as the main drawback of the HUPS6 scheme.
The time-evolution of the numerical solution and the accuracy are given in figures that follow.

The temporal evolution of the accuracy of the numerical solution, Figure 17b, asymptotically converges towards the theoretical value of 4, as time grows up.
If we look at the time-evolution of the errors, we get results illustrated by Figure 18.



(a) Solution at $t = 100\Delta t$ (b) accuracy time-evolution

Figure 17: HUPS6 scheme: time-evolutions, at $T = 100\Delta t$ for the solution, $u(x, t = 0) = sin2\pi x$, and $T = 200\Delta t$ for the accuracy, for $\nu = 0.75, N = 10$ cells/wavelength and $\bar{r}_i^0 = Im\left\{j\beta \times e^{j2\pi x_i}\right\}$.

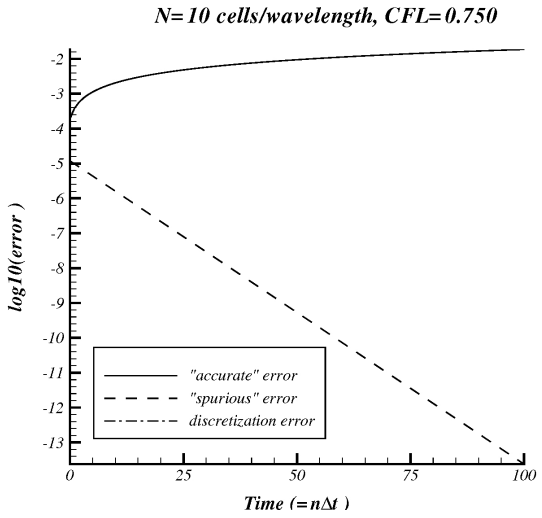


Figure 18: HUPS6 scheme: time-evolution of errors for $\nu = 0.75$, $N = 10$ and $\bar{r}_i^0 = Im\left\{j\beta \times e^{j2\pi x_i}\right\}$

This result is pretty similar to that of Figure 14, for the HUPS5 scheme. In addition, things are almost unchanged if we approximate the space derivative,

$\bar{r}_i^0$, by a centered discretization, Figure 19.

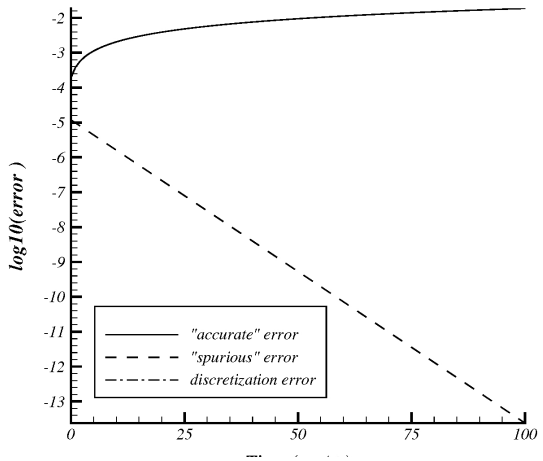


Figure 19: HUPS6 scheme: time-evolution of errors for $\nu = 0.75$, $N = 10$ and $\bar{r}_i^0 = Im\left\{jsin\beta \times e^{j2\pi x_i}\right\}$

Therefore, just as for the HUPS5 scheme, the HUPS6 scheme is not sensitive to the initialization of the space derivative, $\bar{r}_i^0$, because the modulus of the spurious eigenvalue, $\lambda_2(\beta, \nu)$, remains largely below the unit value, whatever the number of cells per wavelength, N, selected.

Now, upon using these first results, we can extend our comparative study to two-dimensional linear advection problems.

2.2 Two-dimensional analysis

We want to discretize the two-dimensional form of the scalar linear advection equation.
In a Cartesian reference frame, this equation reads as:

$$u_t + au_x + bu_y = 0 \qquad (a \equiv Cte > 0, b \equiv Cte > 0) \tag{2.19}$$

Once again, we select an uniform Cartesian mesh, in both directions, with: $\Delta x = \Delta y \equiv h$. In addition , we define the "convection angle", $\theta \equiv Atan(b/a)$, as a new parameter to typify the two-dimensional nature of the physical problem.

With the same choices we made previously, the CUPS scheme gives us the following ODE in time, at the reference discrete point, $o \equiv (x_{i,j}, y_{i,j})$,

upon starting from an integration of equation (2.19) over the discrete cell $I_o \equiv [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$:

$$\frac{d\bar{u}_o}{dt} + a\frac{u^L_{i+1/2} - u^L_{i-1/2}}{h} + b\frac{u^L_{j+1/2} - u^L_{j-1/2}}{h} = 0 \tag{2.20}$$

Once again, all the space-derivatives are approximated in an upwind manner, according to the sign of the convection velocities, (a, b).
With those notations, $\bar{u}_o$, is the cell averaged value of the numerical solution over I_o and $u^L_{i+1/2}$, for example, loses its pointwise meaning to become the following averaged value, at the left of the face $x = x_{i+1/2}$:

$$u^L_{i+1/2} \equiv \frac{1}{h^2}\left[\int_{y_{j-1/2}}^{y_{j+1/2}} \tilde{u}_o(x_{i+1/2}, y)dy\right] \tag{2.21}$$

where, $\tilde{u}_o(\vec{x})$ stands for the two-dimensional interpolation polynomial defined within the discrete cell, I_o.
More specifically, for a spatial $p-th$ order accuracy of the resulting scheme, we define $\tilde{u}_o(\vec{x})$ as follows:

$$\tilde{u}_o(\vec{x}) \equiv \bar{u}_o + \sum_{k=1}^{p-1}\sum_{l=0}^{k} \frac{1}{h^k k!} \times \binom{k}{l} \times \left[(x - x_o)^l (y - y_o)^{k-l} - \overline{(x^l y^{k-l})}_o\right] \times D'_{l,k-l} \tag{2.22}$$

with the following definition for $D'_{l,k-l}$:

$$D'_{l,k-l} \equiv h^k \frac{\partial^k u}{\partial x^l \partial y^{k-l}} + \mathcal{O}(h^p)$$

and for $\overline{(x^l y^{k-l})}_o$:

$$\overline{(x^l y^{k-l})}_o \equiv \frac{1}{|I_o|}\int_{I_o} (x - x_o)^l (y - y_o)^{k-l} dxdy$$

which is designed to ensure that the average of $\tilde{u}_o(\vec{x})$, over I_o, is equal to $\bar{u}_o$.

Practically, the quantities, $D'_{l,k-l}$, are computed by using the mean preservation condition, over each discrete cell, I_j:

$$\frac{1}{|I_j|}\int_{I_j} \tilde{u}_o(\vec{x})dxdy = \bar{u}_j, \forall j \in S_o - \{o\} \tag{2.23}$$

where S_o represents the two-dimensional numerical stencil over which $\tilde{u}_o$ is defined.
In order to maintain the fifth-order accuracy of the CUPS scheme, in each spatial direction, we selected the numerical stencil, S_o, which is illustrated by Figure 20 that follows:

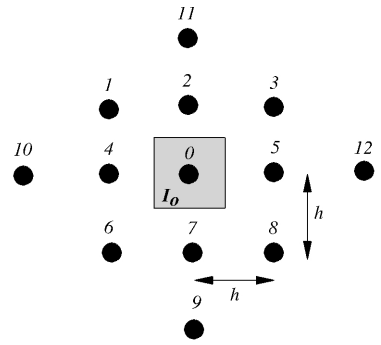


Figure 20: Two-dimensional CUPS5 scheme: Numerical stencil, S_o, and the control volume, I_o.

Thus, if we discard the central point, o, which already appears in equation (2.22), this stencil generates 12 relationships in using condition (2.23).

Consequently, if we want to preserve the fifth-order accuracy of the interpolation procedure, at least along each spatial direction, one must add the following terms to equation (2.22), initially defined for $p = 4$:

$$\frac{1}{24h^4}\left[(x-x_o)^4 - \overline{(x^4y^0)}_o\right] \times D'_{4,0} + \frac{1}{24h^4}\left[(y-y_o)^4 - \overline{(x^0y^4)}_o\right] \times D'_{0,4} + \frac{1}{4h^4}\left[(x-x_o)^2(y-y_o)^2 - \overline{(x^2y^2)}_o\right] \times D'_{2,2} \tag{2.24}$$

so that a system of 12 equations for the 12 unknowns, $D'_{l,k-l}$, results. This latter system is analytically inverted from MAPLE-2017.3 to generate the interpolated quantities, $u^L_{i\pm1/2}$ and $u^L_{j\pm1/2}$ that appear into equation (2.20) in using conditions such as (2.23).
Note that we did not choose the derivatives $D'_{3,1}$ and $D'_{1,3}$ in (2.24) because these terms cannot be computed consistently over the numerical stencil, S_o.

Lastly, the SSPRK(5,4) time-integration procedure enables us to generate the final algebraic form that is fourth-order accurate, overall. However, since the spatial fifth-order accuracy is preserved along each Cartesian direction, we call the resulting CUPS scheme, the CUPS5 scheme. The algebraic formulas that characterize $u^L_{i\pm1/2}$ and $u^L_{j\pm1/2}$, are not given, here, because they are too intricate.
Practically, the system of order 12 for the unknowns, $D'_{l,k-l}$, is directly solved, at each time-step, by a Gauss pivot method. The metric matrix of this system is pre-inverted and the result is stored at the very begining of the computations.

Then, by knowing $D'_{l,k-l}$, the computations of $u^L_{i\pm1/2}$ and $u^L_{j\pm1/2}$, from $\tilde{u}_o$, become straightforward.
To study the spectral properties of the resulting CUPS5 scheme, we start from the integrated form of (2.20) by the SSPRK(5,4) algorithm and we extend the one-dimensional procedure by using a two-dimensional discrete Fourier transform of the numerical solution. Then, it becomes possible to formulate the complex amplification factor of the scheme, $\mathcal{G}(\beta, \nu, \theta, \varphi)$ (see Appendix C).
By doing that, the first result we obtain is the truncation error of the CUPS5 scheme in the Fourier space. Indeed, this latter quantity writes as:

$$\tau(\beta_x, \beta_y, \nu_x, \nu_y) = -\frac{j}{120}\frac{(\nu_x\beta_x + \nu_y\beta_y)^5}{\Delta t} + \mathcal{O}\left(\frac{\beta^6}{\Delta t}\right) \tag{2.25}$$

where we defined the following practical quantities:

$$\begin{cases} \nu_x \equiv \nu \times cos\theta, & \beta_x \equiv k_x \times h \equiv \beta \times cos\varphi \\ \nu_y \equiv \nu \times sin\theta, & \beta_y \equiv k_y \times h \equiv \beta \times sin\varphi \end{cases}$$

with: $\nu \equiv \frac{\sqrt{a^2+b^2}\Delta t}{h}$, which represents the CFL number of the discretization scheme; therefore, ν_x and ν_y are, respectively, the CFL number in the x and y-directions. Similarly, $\beta \equiv \sqrt{\beta_x^2 + \beta_y^2}$, represents the equivalent phase angle of the wave solution of (2.19) and φ, indicates the direction of this wave-like solution; this way, the relationship $N \times \beta = 2\pi$, keeps the same meaning as in the one-dimensional context.
Equation (2.25) is interesting, since, by comparing with equation (2.8), it shows that the CUPS5 scheme is, indeed, fifth-order accurate along each spatial direction; however, due to the fourth-order accuracy of the time-integration and an imperfect interpolation polynomial, the CUPS5 scheme remains fourth-order accurate (term: $\mathcal{O}(\beta^5_{x,y}/\Delta t)$), overall.
Now, If we plot the modulus of the complex amplification factor, $\mathcal{G}(N, \nu, \psi)$ ($\psi \equiv \theta - \varphi$, see appendix C), then, we get results displayed by Figure (21) that follows.

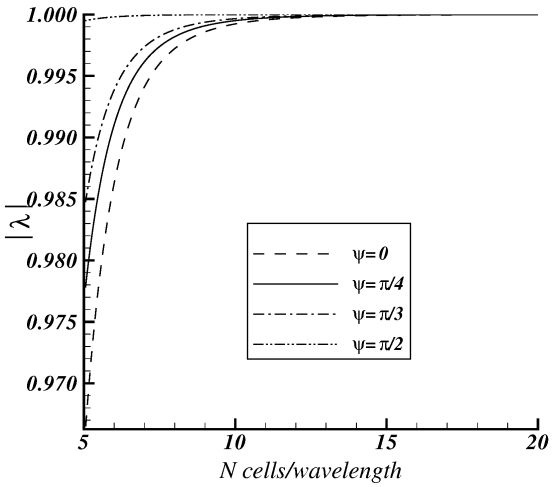
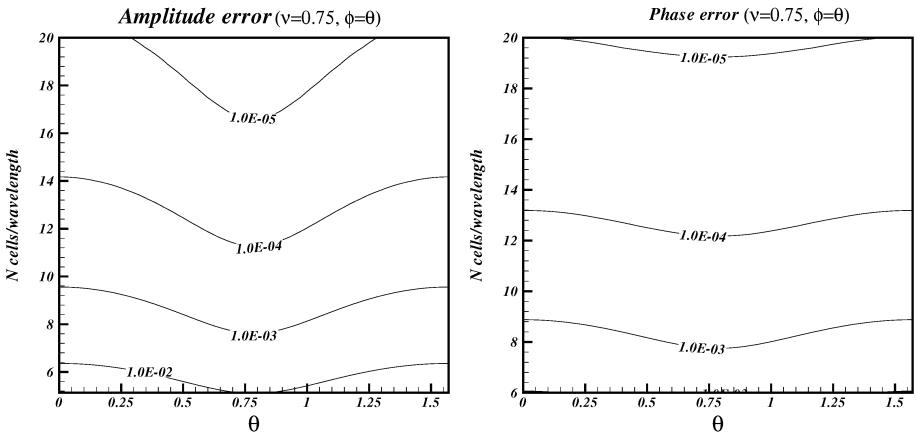


Figure 21: CUPS5 scheme: modulus of the amplification factor, $\mathcal{G}(N, \nu, \psi)$

This result gives us useful insights: first, when $\psi \equiv 0$ (convection and wave directions both aligned), we find the results displayed by Figure 3, for the amplification factor: this is the solution obtained in the one-dimensional context. On the other side, the solutions for which the convection and wave direction are not aligned ($\psi \neq 0$) give numerical results that are better since the values of the amplification factor are higher. The best results are obtained for $\psi = \pi/2$, or, in other words, when the wave and convection directions are orthogonal.
However, it is not obvious to interpret results in terms of the wave direction; for this reason, we only present numerical results for which $\psi \equiv 0$, in what follows: this is what we may consider as the most unfavorable case for the CUPS5 scheme, according to Figure 21.

Summarizing those first results, one may say that the two-dimensional nature of the problem does not alter the spectral properties of the CUPS5 scheme, at least in what concerns the amplification factor.
To check these first findings, let us consider the Figures that follow:

Figures 22 represent the iso-lines of the amplitude, Figure 22a, and of the phase, Figure 22b, for $\nu = 0.75$ and whatever $\theta \in [0, \pi/2]$.
Firstly, it is interesting to note that, in both cases, the minimum value for the errors (at a given spatial resolution, N) is obtained for $\theta = \pi/4$: this a truly multi-dimensional solution. However, the variation of errors with the convection angle, θ, is not very significant: this is particularly meaningful in what concerns the phase error, Figure 22b.
Secondly, by comparing with the one-dimensional results of Figures 4, we can see that the error levels generated by the CUPS5-2D scheme are very similar,



(a) Amplitude error (b) Phase error

Figure 22: CUPS5 scheme: Amplitude and Phase errors for: $\varphi = \theta$

for the same spatial resolution, N, to those of the CUPS5-1D version.
Overall, those results demonstrate that, by discretizing the two-dimensional advection problem, (2.19), with a genuine multi-dimensional scheme that shares the main characteristics of the one-dimensional version, the high-order accuracy properties, are preserved: it is a very hopeful result for the future extensions of the HUPS scheme.
Let us begin those extensions with the two-dimensional version of the HUPS4 scheme. For a straightforward extension of the one-dimensional version of this scheme, we select the numerical stencil, S_o, typified by Figure 23 that follows.

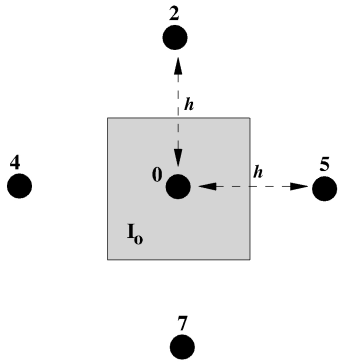


Figure 23: Two-dimensional HUPS4 scheme: Numerical stencil, S_o, and the control volume, I_o.

The very compact nature of this scheme is obvious when we compare its stencil with that of the CUPS5 scheme, Figure 20. Once again, in order to maintain the fifth-order spatial accuracy along each spatial direction, we add to (2.22) the following fourth-order terms, with $p = 4$:

$$\frac{1}{24h^4}\left[(x-x_o)^4 - \overline{(x^4y^0)}_o\right] \times D'_{4,0} + \frac{1}{24h^4}\left[(y-y_o)^4 - \overline{(x^0y^4)}_o\right] \times D'_{0,4} \quad (2.26)$$

This latter choice is the only one that allows to compute all the derivatives, $D'_{l,k-l}$, which define the interpolation polynomial over S_o.
However, we keep calling the resulting scheme, the "HUPS4 scheme", in order to distinguish it from the higher order version that follows.
Therefrom, the derivatives of the resulting polynomial are computed by solving the following system, $\forall j \in S_o - \{o\}$:

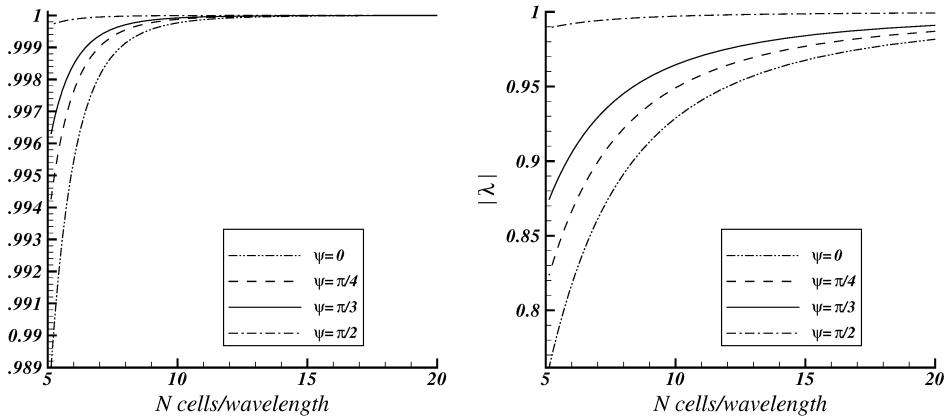
$$\begin{cases} \dfrac{1}{|I_j|}\displaystyle\int_{I_j} \tilde{u}_o(\vec{x})dxdy = \bar{u}_j \\ \dfrac{h}{|I_j|}\displaystyle\int_{I_j} \frac{\partial \tilde{u}_o}{\partial x}(\vec{x})dxdy = \bar{r}_j \\ \dfrac{h}{|I_j|}\displaystyle\int_{I_j} \frac{\partial \tilde{u}_o}{\partial y}(\vec{x})dxdy = \bar{s}_j \end{cases} \quad (2.27)$$

This is a system of 12 equations, for the 11 unknowns, $D'_{l,k-l}$ ($D'_{0,0} \equiv \bar{u}_o$, is already included into the initial definition of $\tilde{u}_o(\vec{x})$). Once solved by a least-square procedure, the interpolation polynomial, $\tilde{u}_o(\vec{x})$, may be computed in order to generate the pointwise interpolated values that appear in the following semi-discrete Hermitian system:

$$\begin{cases} \dfrac{d\bar{u}_o}{dt} + a\dfrac{u^L_{i+1/2} - u^L_{i-1/2}}{h} + b\dfrac{u^L_{j+1/2} - u^L_{j-1/2}}{h} = 0 \\ \dfrac{d\bar{r}_o}{dt} + a\dfrac{r^L_{i+1/2} - r^L_{i-1/2}}{h} + b\dfrac{r^L_{j+1/2} - r^L_{j-1/2}}{h} = 0 \\ \dfrac{d\bar{s}_o}{dt} + a\dfrac{s^L_{i+1/2} - s^L_{i-1/2}}{h} + b\dfrac{s^L_{j+1/2} - s^L_{j-1/2}}{h} = 0 \end{cases} \quad (2.28)$$

Lastly, by integrating this system of ODEs by the means of the SSPRK(5,4) time-integration procedure, we get an algebraic system for the discrete variables $(\bar{u}_j, \bar{r}_j, \bar{s}_j)$.
Consequently, if we do a discrete Fourier transform of this latter system, we get the 3×3 amplification matrix, $\mathcal{G}(\beta, \nu, \psi)$, of which the complex eigenvalues give

the spectral properties of any HUPS scheme (see Appendix C).
This time, the amplification matrix generates not only the "accurate" eigenvalue, $\lambda_1(\beta, \nu, \psi)$, which represents the approximation of the exact amplification factor of (2.19) (see Appendix C), but also two spurious eigenvalues, $\lambda_{2,3}(\beta, \nu, \psi)$, which have no counterpart in the differential problem, (2.19).
Since we have to solve for a third-order complex polynomial to get the $\lambda' s$, it is not possible to derive the complex truncation error in a closed analytical form; for this reason, we only presents in what follows numerical results obtained from the spectral analysis.
To begin, we consider results obtained by computing the modulus of the complex eigenvalues of $\mathcal{G}(\beta, \nu, \psi)$: those are Figures 24.

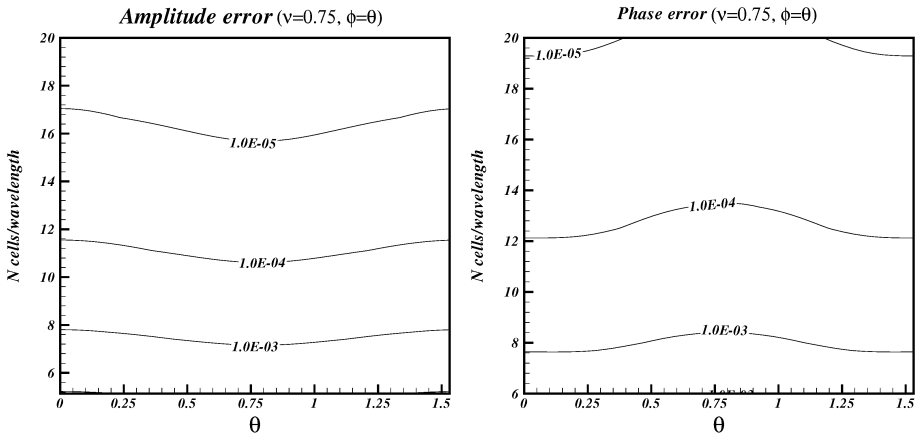


(a) $|\lambda_1(N, \nu = 0.75, \psi)|$ (b) $|\lambda_{2,3}(N, \nu = 0.75, \psi)|_{max}$

Figure 24: HUPS4 scheme: Modulus of the eigenvalues of the complex amplification matrix, $\mathcal{G}(\beta, \nu, \psi)$.

Figure 24a presents numerical results obtained with the accurate eigenvalue, $\lambda_1(\beta, \nu, \psi)$, for $\nu = 0.75$ and $\psi \in \{0, \pi/4, \pi/3, \pi/2\}$.
As one can see it, we obtain the same trends as with the CUPS5-2D scheme, Figure 21: the best value for the modulus of $\lambda_1(\beta, \nu, \psi)$ is obtained for $\psi = \pi/2$, whereas $\psi = 0$ (convection and wave directions aligned) gives the lowest value for this quantity. In addition, always by comparing with Figure 21, results obtained with the HUPS4 scheme are more accurate since the modules are nearer of the unit value for $N = 5$: we have, indeed, $|\lambda_1| = 0.9998$, for $\psi = 0$.
Now, if we look at the behavior of the spurious eigenvalues of the HUPS4 scheme, Figure 24b, we plot the most detrimental case for the study of the stability of the scheme: $max(|\lambda_2|, |\lambda_3|)$.
We note the significant result that the modulus of the spurious eigenvalues is below 1, whatever the values of ψ; hence, there are no minor sources of instabilities within the HUPS4 scheme. In addition, the case for which the convection

and the wave directions are aligned ($\psi \equiv 0$) is the most stable among cases investigated.
The genuine accuracy of the HUPS4 may be evaluated by considering the magnitude of the amplitude and phase errors. These errors are obtained for the most detrimental case, $\psi = 0$ ($\varphi \equiv \theta$), in terms of accuracy. Figures 25 that follow plot these latter quantities for $\nu = 0.75$ and the convection angle $\theta \in [0, \pi/2]$.



(a) Amplitude error (b) Phase error

Figure 25: HUPS4 scheme: Amplitude and Phase errors for: $\varphi = \theta$

As one can see from Figure 25a, the HUPS4 scheme gives a better amplitude error level than the CUPS5 scheme does, Figure 22a; moreover, this latter quantity is less sensitive to the convection angle, θ, than with the CUPS5 scheme. In reality, the HUPS4 scheme is almost free of the convection angle in what concerns the amplitude error: this is a good proof for a genuine multi-dimensional discretization.

The phase error, Figure 25b, is almost the same as with CUPS5, Figure 22b; however, this quantity is slightly more sensitive to the convection angle, θ.
Therefore, those first results show that, with the same overall accuracy and despite a more compact numerical stencil, a HUPS scheme performs better than a classical upwind scheme (CUPS).

Now, if we want to improve the spatial accuracy of the Hermitian scheme, one must resort to a larger numerical stencil, S_o: this improvement is typified by Figure 26 that follows.
Indeed, by selecting a stencil of 9 points, it becomes possible to build a complete interpolation polynomial of degree four: in that case, the fifth-order spatial accuracy ($p = 5$) may be reached if we generate, at least, 14 relationships for

the 14 unknowns, $D'_{l,k-l}$ ($D'_{0,0} \equiv \bar{u}_o$, is already included into the initial definition of $\tilde{u}_o(\vec{x})$).
The choice we put forward consists in using only the variable, $\bar{u}$, for the nodes $\{1, 3, 6, 8\}$, Figure 26, for generating the missing relationships. This way, we are able to yield a system of 16 equations for the 14 unknowns, which is always solved by means of a least-square procedure.

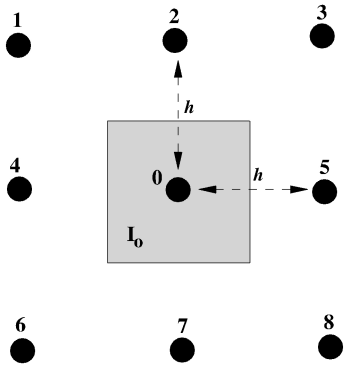
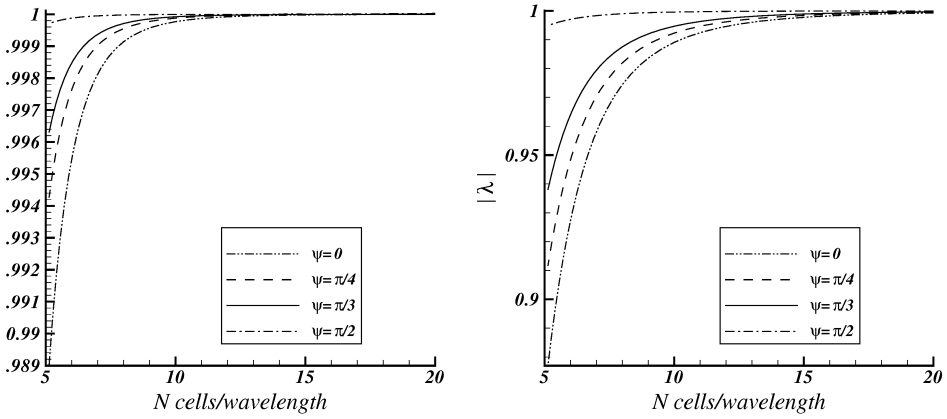


Figure 26: Two-dimensional HUPS5 scheme: Numerical stencil, S_o, and the control volume, I_o.

Note that this is by no means the best accuracy attainable over this stencil: however, since we want to pursue the one-dimensional analysis by comparing many versions of the HUPS scheme, this choice may constitute a good accommodation between a higher spatial accuracy and a moderate increase of the algebraic complexity.
Since we have, now, a complete polynomial of degree four, we call the resulting numerical scheme, the HUPS5 scheme.
Therefore, once the coefficients, $D'_{l,k-l}$, of $\tilde{u}_o(\vec{x})$ are computed, we may generate, as we did previously, the discrete Hermitian system for the unknowns $(\bar{u}, \bar{r}, \bar{s})$ and then, a discrete Fourier transform allows to formulate the complex amplification matrix, $\mathcal{G}(\beta, \nu, \psi)$.

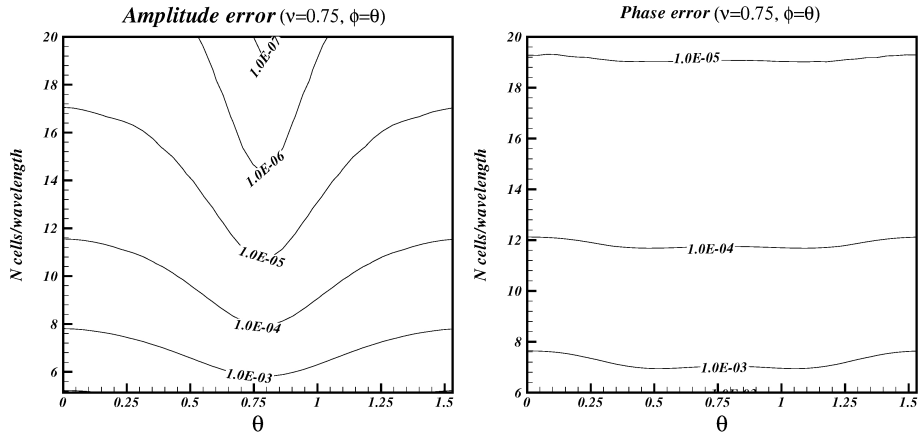
From this amplification matrix, we can withdraw the first useful insights, displayed by Figures 27.
The main difference between these results and those illustrated by Figures 24 appears into the behavior of the spurious eigenvalues. Indeed, those quantities, when computed for the HUPS5 scheme, are closer to the unit value, whatever the value of ψ, Figure 27b: one may expect a more significant sensitivity of the numerical solution to slight disturbances, during the computations.



(a) $|\lambda_1(N, \nu = 0.75, \psi)|$ (b) $|\lambda_{2,3}(N, \nu = 0.75, \psi)|_{max}$

Figure 27: HUPS5 scheme: Modulus of the eigenvalues of the complex amplification matrix, $\mathcal{G}(\beta, \nu, \psi)$.

Practically, the consequence of this increasing in sensitivity is in a necessary lowering of the CFL number in order to preserve the accuracy properties. Then, Figures 28 below, display numerical results obtained in computing the amplitude and phase errors of the numerical solution in the Fourier space.

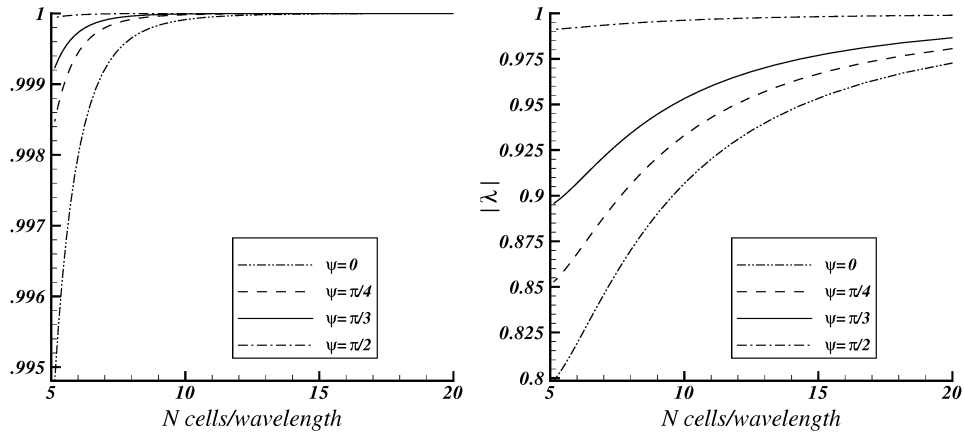


(a) Amplitude error (b) Phase error

Figure 28: HUPS5 scheme: Amplitude and Phase errors for: $\varphi = \theta$

The utmost sensitivity of the amplitude error to the convection angle, θ, appears obviously, here, Figure 28a, whereas the phase error is, now, almost

independent of θ, Figure 28b; however, except this change, the errors levels generated by the HUPS5 scheme are tantamount to those of the HUPS4 scheme, Figures 25.
Consequently, upon considering those results, the advantage in using a fifth-order version of the HUPS scheme over an enlarged stencil is not obvious if we only consider a spectral analysis.
Thus, one might think that the special choice we made in selecting a stencil of which the corner points are only associated to the mean quantity, $\bar{u}$, as parameter, is not optimum.
For this reason, we investigate an ultimate choice based on the stencil illustrated by Figure 26 but, this time, we associate the three variables $(\bar{u}, \bar{r}, \bar{s})$ to each node constituting the numerical stencil, S_o; this way, we can generate 26 relationships ($\bar{u}_o$ not included) for the unknowns coefficients of (2.22). Thus, we can build an interpolation polynomial of degree five ($p = 6$) that necessitates the computation of 20 coefficients ($D'_{0,0} \equiv \bar{u}_o$) to be uniquely defined. The resulting numerical scheme is called the HUPS6 scheme, accordingly: this is the best accuracy we can hope by using a HUPS scheme over a 9 points stencil.
With the same procedure as previously, we can analyze the spectral properties of the HUPS6 scheme. Firstly, Figures 29 display the modulus of the eigenvalues of this scheme.

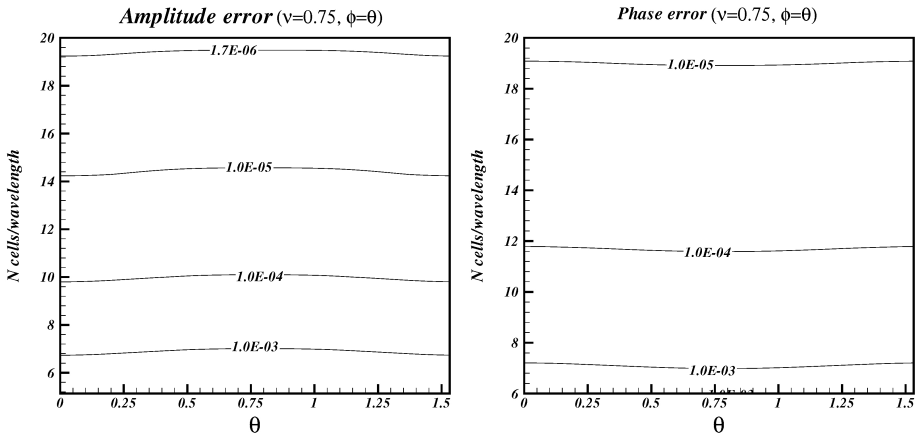


(a) $|\lambda_1(N, \nu = 0.75, \psi)|$ (b) $|\lambda_{2,3}(N, \nu = 0.75, \psi)|_{max}$

Figure 29: HUPS6 scheme: Modulus of the eigenvalues of the complex amplification matrix, $\mathcal{G}(\beta, \nu, \psi)$.

As expected and due to the increased spatial accuracy of the HUPS6 scheme, we get an amplification factor of which the values are very close to the unit value, whatever the values taken by ψ, Figure 29a: by comparing with Figure 27a the

improvement is real although slight. On the other hand, Figure 29b gives unexpected results since the maximum modulus of the spurious eigenvalues is lower than that of the HUPS5 scheme, Figure 27b: as a matter of fact, results of Figure 29b are almost identical with those obtained in studying the HUPS4 scheme, Figure 24b.
Therefore, upon considering these first results, one might say that the HUPS6 scheme constitutes an intermediary scheme between HUPS4 and HUPS5 schemes.
To end this two-dimensional spectral analysis, let us study results displayed by Figures 30.



(a) Amplitude error (b) Phase error

Figure 30: HUPS6 scheme: Amplitude and Phase errors for: $\varphi = \theta$

These results present the computation of amplitude and phase errors, for $\varphi \equiv \theta$, $\nu = 0.75$ and $\theta \in [0, \pi/2]$. The major teaching from these computations is that the errors are independent of the convection angle, θ. Nevertheless, by comparing with Figures 28 the error level is not substantially improved.
Consequently, one may consider that the meaning improvement brought by passing to a sixth-order spatial interpolation is the independence of the discretization from the convection angle.

On summarizing the main teachings of this sub-section, two schemes seem to emerge and to constitute a good compromise between accuracy and algebraic simplicity: the CUPS5 and HUPS4 schemes. To be more specific, the most efficient scheme seems to be, despite everything, the CUPS5 scheme.
Unexpectedly, the high-order accuracy combined with a compact discretization is not an absolute guarantee for obtaining outstanding performances. Moreover, the behavior of spurious eigenvalues is difficult to foresee when one selects a high-order HUPS scheme.

These conclusions should be contrasted with those coming from the one-dimensional spectral analysis, which demonstrates that the most accurate scheme is also the most efficient; with the emergence of two supplementary parameters, θ and φ, a genuinely two-dimensional scheme has a peculiar numerical behavior.

However, these initial findings should be weighted by the following remarks: the scope of a spectral analysis is limited to linear problems without any influence of the boundary conditions: the use of a compact discretization is not negligible in that case. In addition, the computation effort is a significant parameter that is not at all addressed by a spectral analysis.
Therefore, in order to be as exhaustive as possible, we go beyond this spectral analysis and we study the whole numerical behavior of the schemes we designed by computing two-dimensional scalar non-linear problems.

3 Two-dimensional high-order schemes.

3.1 HLLE approximate Riemann solver for the convective fluxes.

By employing the numerical methods we previously presented, we want to study the discretization of the scalar, non-linear, two-dimensional, conservation law, over the computation domain, $\mathcal{D}$:

$$u_t + f(u)_x + g(u)_y = 0 \quad \forall \vec{x} \in \mathcal{D} \tag{3.1}$$

Consequently, if we use a CUPS finite-volume scheme, we can generate the following semi-discrete ODE equation, over a Cartesian mesh of typical size $h \equiv \Delta x \equiv \Delta y$ and for the discrete point $o \equiv (x_{i,j}, y_{i,j})$:

$$\frac{d\bar{u}_o}{dt} + \frac{\left(\bar{f}_{i+1/2} - \bar{f}_{i-1/2}\right)}{h} + \frac{\left(\bar{g}_{j+1/2} - \bar{g}_{j-1/2}\right)}{h} \tag{3.2}$$

In order to generate this equation, we introduced the following approximations:

$$\begin{cases} \bar{f}_{i+1/2} \approx \displaystyle\int_{S_{i+1/2}} f(u(x_{i+1/2}, y))dy \\ \\ \bar{g}_{j+1/2} \approx \displaystyle\int_{S_{j+1/2}} g(u(x, y_{j+1/2}))dx \end{cases} \tag{3.3}$$

Practically, these approximations are realized by using a 2-point Gaussian integration formula, which is exact for a polynomial up to degree 3 ($p = 4$).
For example, along the face $x = x_{i+1/2}$, referenced in what follows as $S_{i+1/2}$, we write, by using the interpolated values $u_q^L \equiv \tilde{u}_o(\vec{x}_q)$ and $u_q^R \equiv \tilde{u}_{i+1}(\vec{x}_q)$ (see Figure 31):

$$\bar{f}_{i+1/2} \equiv \frac{1}{2}\left(\tilde{f}(u^L_{q_1}, u^R_{q_1}) + \tilde{f}(u^L_{q_2}, u^R_{q_2})\right) \tag{3.4}$$

where q_1 represents, for example, the quadrature point: $(x_{i+1/2}, w \times y_{j-1/2} + (1-w) \times y_{j+1/2})$, with: $w \equiv \frac{1}{2}\left(1 + \frac{\sqrt{3}}{3}\right)$; q_2 is computed in the same way, by: $(x_{i+1/2}, w \times y_{j+1/2} + (1-w) \times y_{j-1/2})$. Consequently, the resulting spatial discretization will be, at best, fourth-order accurate.

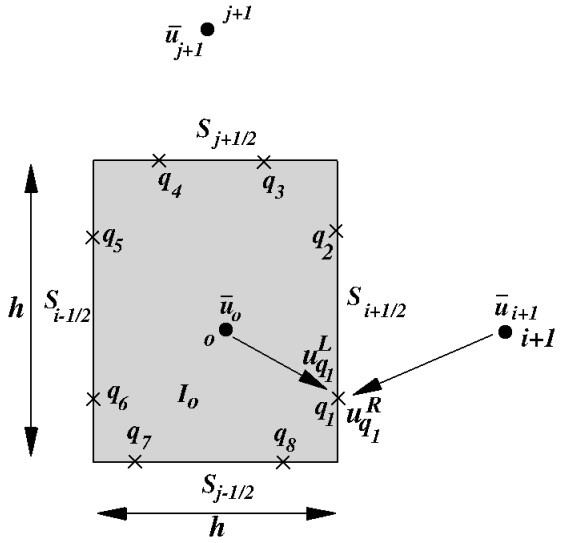


Figure 31: Computation of the interpolated values (u^L_q, u^R_q) at the cell-face, $S_{i+1/2}$, from the interpolating polynomial, $\tilde{u}_o(\vec{x})$.

Lastly, to introduce upwind principles into the discretization scheme, spatially typified by (3.4), we use an approximate Riemann solver for computing the numerical flux, $\tilde{f}(u^L, u^R)$; specifically, if we use the now "classical" HLLE scheme, [15], the computation of this numerical flux is as follows:

$$\begin{aligned} \tilde{f}(u^L, u^R) &= f(u^L) + a_L \cdot (u^* - u^L) \\ &= f(u^R) - a_R \cdot (u^R - u^*) \end{aligned} \tag{3.5}$$

where the mean quantity, u^*, is calculated from the "conservativity" condition (see [15]):

$$u^* = \frac{a_R u^R - a_L u^L}{a_R - a_L} - \frac{f(u^R) - f(u^L)}{a_R - a_L} \tag{3.6}$$

with:

$$\begin{cases} a_L \equiv min(f'(u^L), \bar{a}, 0) \\ a_R \equiv max(f'(u^R), \bar{a}, 0) \end{cases} \tag{3.7}$$

The set of conditions (3.7) for the characteristic velocities was early designed by Einfeldt, [15], in the context of Euler equations.
On the other hand, if we define the mean characterisitc velocity, $\bar{a}$, as:

$$\bar{a} \equiv \begin{cases} \dfrac{f(u^R) - f(u^L)}{u^R - u^L}, \quad if \quad |u^R - u^L| \leq \varepsilon \\ f'(u^R) \quad else \end{cases} \tag{3.8}$$

then, the numerical flux, $\tilde{f}(u^L, u^R)$, which is computed according to equations (3.5), (3.6), (3.7), and (3.8), is a Lipschitz continuous function of its arguments, [15].
This set of definitions and properties constitute the main features of the "HLLE Riemann solver" designed in [15]; this solver constitutes the building block for the spatial discretization of the convective terms of (3.1) by the CUPS scheme.
Now, to extend the HLLE algorithm to an Hermitian formulation, we have to consider the following Hermitian system for the unknowns $(u, r \equiv h\dfrac{\partial u}{\partial x}, s \equiv h\dfrac{\partial u}{\partial y})$ from (3.1):

$$\begin{cases} u_t + f(u)_x + g(u)_y = 0 \\ r_t + [a(u).r]_x + [b(u).r]_y = 0 \\ s_t + [a(u).s]_x + [b(u).s]_y = 0 \end{cases} \tag{3.9}$$

with the "characteristic velocities": $(a(u) \equiv f'(u), b(u) \equiv g'(u))$. Hence, if we introduce, for the variable r or s, the following definitions:

$$\begin{cases} f_1(u, \alpha) \equiv [a(u) \cdot \alpha] \\ g_1(u, \alpha) \equiv [b(u) \cdot \alpha] \end{cases} \tag{3.10}$$

with $\alpha \in \{r, s\}$, we can note the following property:

$$\begin{cases} \dfrac{\partial f_1}{\partial \alpha} = a(u) \\ \dfrac{\partial g_1}{\partial \alpha} = b(u) \end{cases} \tag{3.11}$$

Literally, this latter property means that the equation for the first derivatives, (r, s), in (3.9), have the same characteristic velocities as equation (3.1).
Consequently, we can use the HLLE algorithm for computing, for example, the

numerical flux $\tilde{f}_1(u^L, u^R, \alpha^L, \alpha^R)$ that approximates $f_1(u, \alpha)$, in the following way:

$$\begin{aligned}\tilde{f}_1(u^L, u^R, \alpha^L, \alpha^R) &= f_1(u^L, \alpha^L) + a_L \cdot (\alpha^* - \alpha^L) \\ &= f_1(u^R, \alpha^R) - a_R \cdot (\alpha^R - \alpha^*)\end{aligned} \tag{3.12}$$

with:

$$\alpha^* = \frac{a_R \alpha^R - a_L \alpha^L}{a_R - a_L} - \frac{f_1(u^R, \alpha^R) - f_1(u^L, \alpha^L)}{a_R - a_L} \tag{3.13}$$

Therefore, by extending this procedure to the whole Hermitian system, we can produce the following semi-discrete Hermitian system, which generalizes (3.2), and is tantamount to (2.28) in the non-linear case:

$$\begin{cases} \dfrac{d\bar{u}_o}{dt} + \dfrac{\bar{f}_{i+1/2} - \bar{f}_{i-1/2}}{h} + \dfrac{\bar{g}_{j+1/2} - \bar{g}_{j-1/2}}{h} = 0 \\ \dfrac{d\bar{\alpha}_o}{dt} + \dfrac{\bar{f}_{1\,i+1/2} - \bar{f}_{1\,i-1/2}}{h} + \dfrac{\bar{g}_{1\,j+1/2} - \bar{g}_{1\,j-1/2}}{h} = 0 \\ \bar{f}_{1\,i+1/2} \equiv \dfrac{1}{2}\left(\tilde{f}_1(u^L_{q_1}, u^R_{q_1}, \alpha^L_{q_1}, \alpha^R_{q_1}) + \tilde{f}(u^L_{q_2}, u^R_{q_2}, \alpha^L_{q_2}, \alpha^R_{q_2})\right) \\ \alpha \in \{r, s\} \end{cases} \tag{3.14}$$

Indeed, if we set $a(u) \equiv Cte$ and $b(u) \equiv Cte$, then, system (3.14) identifies with (2.28). Finally, by integrating equation (3.2) or system (3.14) by means of the fourth-order accurate SSPRK (5,4) algorithm, we get the finite-volume discretization of the conservation law, (3.1), either from the CUPS or from the HUPS scheme.

3.2 Polynomial reconstruction: monotonicity-preservation principles.

In order to compute the interpolated values, (u_q, α_q), at the quadrature points, we use the definition of the interpolation polynomial, $\tilde{u}_o(\vec{x})$, given by equation (2.22); this way, according to the accuracy selected, $p \in \{4, 5, 6\}$, we can design a numerical scheme of which the interpolated quantities are computed with an accuracy that evolves between the fourth and the sixth-order. Practically, the polynomial coefficients are computed according to the procedure described in the preceding section; depending on the choice for the accuracy, those coefficients are computed either by a direct inversion of the resulting system or by a least-square procedure (Housseholder's algorithm)

However, the interpolated quantities do not guarantee the non-oscillatory behavior of the resulting numerical scheme: some specific conditions must be added for that.
To begin, we start from the monotonicity-preserving procedure we early developed in [11]. This procedure ensures the following monotonicity property, for the computed quantity, $\bar{u}_o$:

$$\begin{aligned} &\bar{u}_o^{n+1} \in [m_o, M_o], \quad \forall o \\ &\text{with:} \quad M_o \equiv \max_{j \in S_o} \left\{ \bar{u}_j^n \right\} \quad \text{and} \quad m_o \equiv \min_{j \in S_o} \left\{ \bar{u}_j^n \right\} \end{aligned} \tag{3.15}$$

This is a classical property of what is usually called a "positive" numerical scheme, [11], [16].
To enforce this latter property, we designed the following two-step procedure.
In a first step, for each interface, S_k, $k \in \{i-1/2, i+1/2, j-1/2, j+1/2\}$, of the control volume, I_o, we compute the limiter, φ_{ok}, which reads as:

$$\varphi_{ok} \equiv \min \left\{ \left| \frac{\frac{1}{2}\left(u_{q_1}^R + u_{q_2}^R\right) - \bar{u}_o}{\frac{1}{2}\left(u_{q_1}^L + u_{q_2}^L\right) - \bar{u}_o} \right|, 1 \right\} \tag{3.16}$$

In the interpolation process, this latter condition ensures that the interpolated values, u_q^L, limited by φ_{ok}, will follow the monotonicity of the mean solution (see [11], for the theoretical derivation of this condition).
To this aim, we define the following modified quantity, $\Delta u_o^1(\vec{x}_q)$, for every quadrature point, q, located on each surface, S_k, delimiting the control volume, I_o (see Figure(31)):

$$\Delta u_o^1(\vec{x}_q) \equiv \varphi_{ok}.\Delta u_o(\vec{x}_q) \equiv \varphi_{ok}.(u_q^L - \bar{u}_o) \quad \forall q \in S_k \tag{3.17}$$

The second step of the monotonicity-preserving procedure uses the preceding computation and lies upon the definition of the limiter, ψ_o, which writes as follows:

$$\psi_o \equiv \min \left\{ \left| \frac{M - \bar{u}_o}{\max_q \Delta u_o^1(\vec{x}_q)} \right|, \left| \frac{m - \bar{u}_o}{\min_q \Delta u_o^1(\vec{x}_q)} \right|, 1 \right\} \tag{3.18}$$

The bounds (m, M) are defined from the initial solution, $u_o(\vec{x})$, of (2.19):

$$m \equiv \min_{\vec{x} \in \mathcal{D}} u_o(\vec{x}) \quad \text{and} \quad M \equiv \max_{\vec{x} \in \mathcal{D}} u_o(\vec{x})$$

Hence, by using this latter limiter, the quantity, $\psi_o.\Delta u_o^1(\vec{x}_q)$, lies inside the range $[m - \bar{u}_o, M - \bar{u}_o], \forall q$, and, therefore, a global maximum principle is ensured for the interpolation procedure.
Finally, the overall procedure that permits to compute monotone interpolated

values, henceforth noted $\tilde{u}_q^L$ in what follows, is synthesized in the following formula:

$$\tilde{u}_q^L = \bar{u}_o + \psi_o.\Delta u_o^1(\vec{x}_q) \tag{3.19}$$

with: $\Delta u_o^1(\vec{x}_q) \equiv \varphi_{ok}.\Delta u_o(\vec{x}_q) \equiv \varphi_{ok}.(u_q^L - \bar{u}_o) \quad \forall q \in S_k$.

By combining these two steps, one can prove (see [11]) that the positivity condition, (3.15), is ensured when discretizing (3.1).
In what concerns the Hermitian scheme, the interpolated derivatives, (r_q^L, s_q^L) at the cell interface, S_k, are simply computed as:

$$\begin{cases} r_q^L \equiv h \cdot \dfrac{\partial \tilde{u}_o}{\partial x}(\vec{x}_q) \\ s_q^L \equiv h \cdot \dfrac{\partial \tilde{u}_o}{\partial y}(\vec{x}_q) \end{cases} \tag{3.20}$$

Thus, doing so, we have designed the ultimate version of a high-order numerical scheme for discretizing, with monotonicity principles, non-linear scalar conservation laws.
Now, in the sub-section that follows, we present numerical tests in order to study and to compare the salient features of each scheme.

3.3 Numerical tests

A Cartesian grid ($h \equiv \Delta x \equiv \Delta y$), is used for all the computations of this sub-section. The CFL number is defined as:

$$CFL \equiv \max_o \left(\left| f'(\bar{u}_o^n) \right|, \left| g'(\bar{u}_o^n) \right| \right) . \Delta t / h \tag{3.21}$$

This latter definition is used for all the tests that follow, except for linear accuracy tests where we selected: $\Delta t \equiv CFL \times h^{p/4}$, for either the CUPS5/HUPS5 scheme ($p = 5$) or the HUPS6 scheme ($p = 6$) to verify spatial accuracy.
In all that follows and for each numerical scheme, the optimum CFL value that generates the lowest error level obtainable, was selected.

Test1 : Two-dimensional linear advection.

We begin this series of numerical tests with an accuracy study. To this aim, we compute the linear form of equation (3.1), $\mathcal{D} \equiv [-2, 2]^2$, with:

$$f(u) \equiv u, \quad g(u) \equiv u$$

The initial conditions are defined as follows:

$$\begin{cases} u_o(\vec{x}) \equiv u(\vec{x}, t = 0) = sin\left(\frac{\pi}{2}(x + y)\right) \\ r_o(\vec{x}) \equiv h \cdot \dfrac{\partial u}{\partial x}(\vec{x}, t = 0) = h \cdot \dfrac{\pi}{2} \cdot cos\left(\dfrac{\pi}{2}(x + y)\right) \\ s_o(\vec{x}) \equiv r_o(\vec{x}) \end{cases}$$

Computations are run up to the dimensionless time, $T = 0.20$, with periodic boundary conditions.
Table 1 displays the numerical results we obtained by using the CUPS5 scheme (equations (2.22) and (2.24) for the definition of the interpolation polynomial, with $p = 4$), with an optimum CFL number of 0.50. The Cartesian mesh is gradually refined from $N = 20$ to $N = 320$ grid points, in each space direction.

N	L_1-error	order	L_∞-error	order
20	2×10^{-5}	-	3.10×10^{-5}	-
40	6.35×10^{-7}	5	1.0×10^{-6}	5
80	2.0×10^{-8}	5	3.20×10^{-8}	5
160	6.65×10^{-10}	5	1.04×10^{-9}	5
320	2.43×10^{-11}	4.8	3.82×10^{-11}	4.7

Table 1: Linear advection test: $u_t + u_x + u_y = 0$, $u_0(x, y) = \sin\left(\frac{\pi}{2}(x + y)\right)$, at $T = 0.20$. CUPS5 scheme.

These results are compared with those obtained by using the HUPS4 ($CFL_{opt} = 0.125$), Table 2, the HUPS5, Table 3 ($CFL_{opt} = 0.250$) and the HUPS6 scheme ($CFL_{opt} = 0.250$), Table 4.
Thus, we get many valuable informations, from these first simulations:

- The HUPS6 scheme excepted, all schemes achieve their designed spatial accuracy, both in L_1 and L_∞ norms.
- The monotonicity-preservation procedure typified by (3.16) and (3.18), does not deteriorate the numerical schemes accuracy.
- Due to its compact nature, the HUPS4 scheme gives better error levels than the CUPS5 scheme does, but its rate of convergence is weaker and the optimum CFL number is lower than that of CUPS5 scheme.
- Despite a better error level, the numerical performances of the HUPS schemes deteriorates as soon as the mesh resolution becomes dense. The behavior of the spurious eigenvalues is a possible explanation of this fact (see Figs. (24b), (27b) and (29b)): the spurious round-off errors are insufficiently damped, during the time-procedure, by the numerical algorithm; hence, despite a CFL condition that ensures the overall numerical stability of the scheme, there exist some minor instabilities, which are supported by the spurious modes of the HUPS scheme, and which "pollute" its numerical solution.
- This latter behavior seems particularly exacerbated with the HUPS6 scheme since its rate of convergence is not at all in agreement with its targeted accuracy. Neither a decrease of the CFL number nor a deactivation of the monotonicity-preservation procedure, alleviates that flaw.

N	L_1-error	order	L_∞-error	order
10	1.27×10^{-4}	-	1.96×10^{-4}	-
20	4.33×10^{-6}	4.9	6.75×10^{-6}	4.9
40	1.54×10^{-7}	4.8	2.42×10^{-7}	4.8
80	6.30×10^{-9}	4.6	9.90×10^{-9}	4.6
160	3.0×10^{-10}	4.4	4.75×10^{-10}	4.4
320	1.65×10^{-11}	4.3	2.60×10^{-11}	4.3

Table 2: HUPS4 scheme.

N	L_1-error	order	L_∞-error	order
10	4.79×10^{-4}	-	7.40×10^{-4}	-
20	1.64×10^{-5}	4.9	2.56×10^{-5}	4.9
40	5.28×10^{-7}	5	8.29×10^{-7}	5
80	1.69×10^{-8}	5	2.66×10^{-8}	5
160	5.65×10^{-10}	5	8.87×10^{-10}	5
320	2.14×10^{-11}	4.7	3.35×10^{-11}	4.7

Table 3: HUPS5 scheme.

N	L_1-error	order	L_∞-error	order
20	1.22×10^{-6}	-	1.91×10^{-6}	-
40	4.14×10^{-8}	4.9	6.51×10^{-8}	4.9
80	3.20×10^{-9}	3.7	5.04×10^{-9}	3.7
160	2.15×10^{-10}	4	3.37×10^{-10}	4

Table 4: HUPS6 scheme.

In addition, Figure 32 that follows, gives us supplementary insights about the computation effort supported by each scheme.

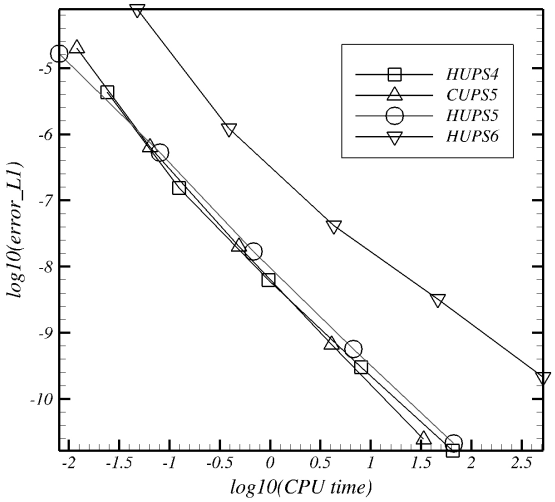


Figure 32: Linear advection test: Comparisons of the computation effort for the CUPS5, HUPS4, HUPS5 and HUPS6 schemes

Clearly, the HUPS6 scheme generates a significant computation effort when compared with the remaining schemes; for a given error level, the computation effort of the CUPS5 scheme is tantamount to that of the HUPS4 and HUPS5 schemes: this property can be explained by knowing that the CUPS5 scheme can accomodate a CFL number that is much greater than that of the HUPS schemes.
Therefore, by only considering these first findings, one might conclude that the increased algebraic complexity associated with a HUPS scheme does not bring any substantial advantage over a more classical scheme. To confirm or to balance this characteristic, we study the two-dimensional linear advection of a discontinuity, in what follows.
Now, the initial solution of the linear advection problem, with $\mathcal{D} \equiv [-1,1]^2$, is typified as follows:

$$\begin{cases} u_o(\vec{x}) \equiv u(\vec{x}, t=0) = \begin{cases} 1 & \text{if} \quad \vec{x} \in S \\ 0 & \text{otherwise} \end{cases} \\ \\ r_o(\vec{x}) \equiv s_o(\vec{x}) \equiv 0 \end{cases}$$

where $S \equiv \left\{(x,y)/\, |x-y| < 1/\sqrt{2}, |x+y| < 1/\sqrt{2}\right\}$ represents an unit square centered at the origin and rotated by an angle of $\pi/4$.
For this problem, the mesh resolution selected is $h = 1/40$ and computations are run up to $T = 2$ (one period in time).

To be as accurate as possible, both the maximum and minimum values of the numerical solution as well as the error in L_1 norm, are computed, every time.
Numerical results obtained with the CUPS5 scheme are displayed by Figure 33, for an horizontal cut along the line $y = 0$ and an optimum CFL value of 0.125.

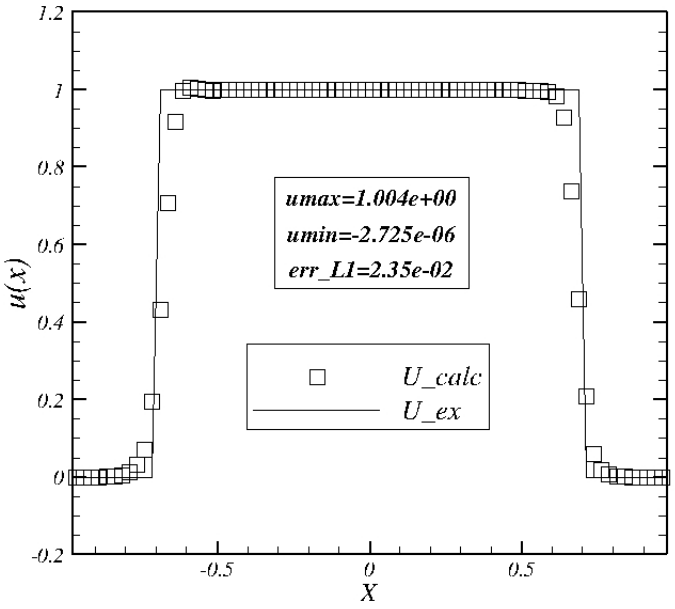


Figure 33: CUPS5 scheme: Two-dimensional linear advection of a discontinuity at $T = 2$, for $h = 1/40$ and $CFL_{opt} = 0.125$.

As one can see, the monotonicity of the numerical solution is ensured and the discontinuities are captured with accuracy.
The numerical solution verifies the maximum principle, since there are no values that lie outside the range $[u_{min} = 0, u_{max} = 1]$.
Now, we compare these first results with those obtained by employing the HUPS4 scheme, with an optimum CFL number of 0.125.
Figure 34, displays those results.

Obviously, the discontinuities are captured with a better accuracy, this without any oscillation. The monotonicity-preservation condition is almost ensured although the minimum value of the solution ($u_{min} = -1.15 \times 10^{-3}$) lies outside the monotonicity range; however, the overall discretization error is lower ($|err|_{L_1} = 1.97 \times 10^{-2}$) than that obtained with the CUPS5 scheme ($|err|_{L_1} = 2.35 \times 10^{-2}$).

This latter result can even be improved if we look at numerical results obtained by using the HUPS5 scheme, Figure 35, with $CFL_{opt} = 0.125$.

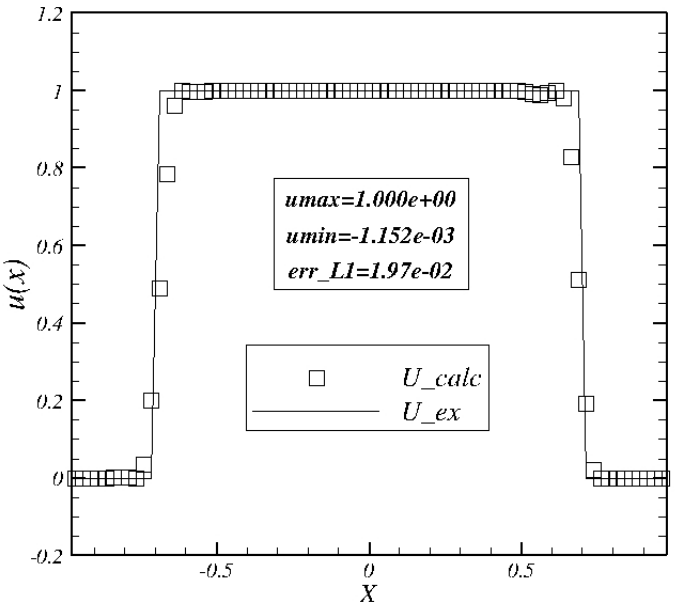


Figure 34: HUPS4 scheme: Two-dimensional linear advection of a discontinuity at $T = 2$ for $h = 1/40$ and $CFL_{opt} = 0.125$.

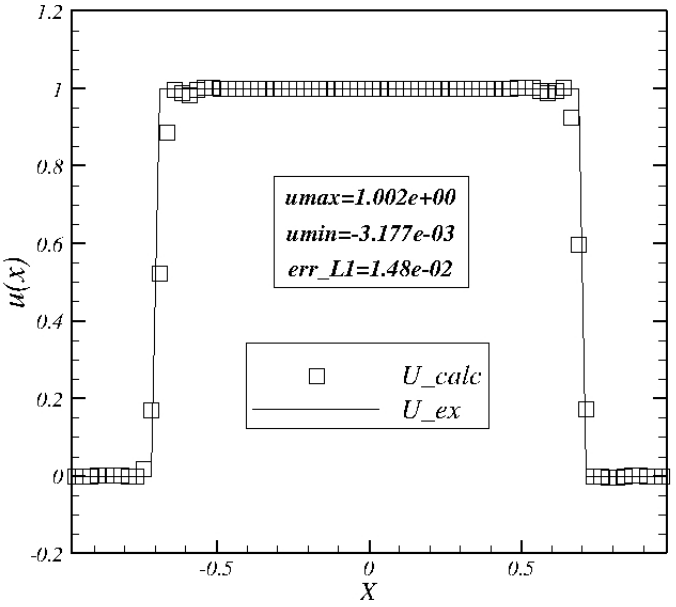


Figure 35: HUPS5 scheme: Two-dimensional linear advection of a discontinuity at $T = 2$ for $h = 1/40$ and $CFL_{opt} = 0.125$.

Indeed, the discontinuities are captured with no more than 3 points; in addition, the discretization error is substantially lower, since we have: $|err|_{L_1} = 1.48 \times 10^{-2}$.
However, the price we have to pay for this better resolution seems to be in the property of monotonicity-preservation since $u_{min} = -3.2 \times 10^{-3}$ and $u_{max} = 1.002$.
Lastly, numerical results given by the HUPS6 scheme, are displayed by Figure 36.

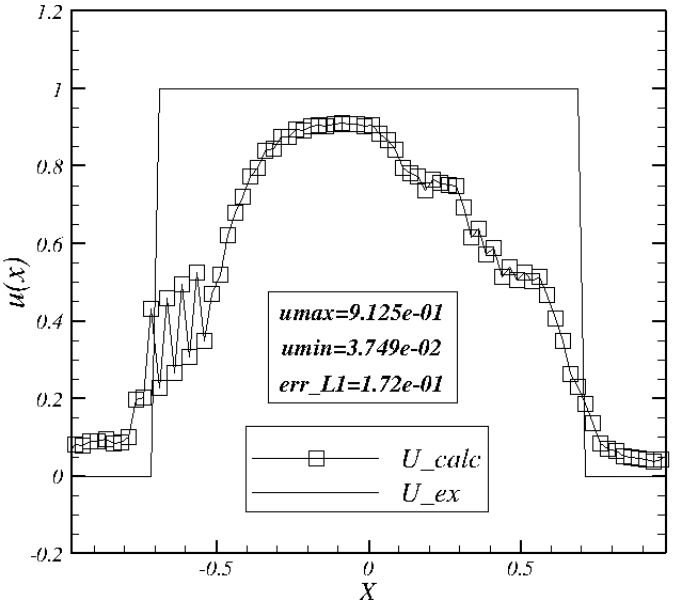


Figure 36: HUPS6 scheme: Two-dimensional linear advection of a discontinuity at $T = 2$ for $h = 1/40$.

As one can see it, not only the accuracy but also the structure of the solution are entirely lost; whatever the CFL number selected, spurious errors develop during the time-procedure. For the reasons identified in the spectral analysis, these errors cannot be damped by the scheme itself and, consequently, the monotonicity-preservation procedure must be activated to preserve the monotonicity of the solution.
In some sense, this procedure is successful since no value lies outside the monotonicity range, however the damping necessary for that purpose, is definitely too strong.
Thus, this second test case brings significant novelties. Indeed, the compact nature of HUPS schemes (HUPS6 excepted) enables to capture a linear discontinuity with a better accuracy; this accuracy is even improved by increasing the degree of the interpolation polynomial. On the other side, in terms of computa-

tion effort, the CUPS5 schemes remains advantageous since the computations last 11 s, in CPU time, while the HUPS4 and HUPS5 schemes necessitate 20s and 35s, respectively.

To get the same error level as the HUPS5 scheme, in using the CUPS5 alternative solution, we need to use a spatial resolution of $h = 1/60$, Figure 37; then, the computation time climbs to 36s, with $CFL_{opt} = 0.125$. However, although this mesh resolution gives an error level tantamount to that of the HUPS5 scheme, the capture of the shock remains less good when compared to this latter scheme, Figure 35: obviously, this is the main advantage in using an Hermitian formulation of problem (3.1).

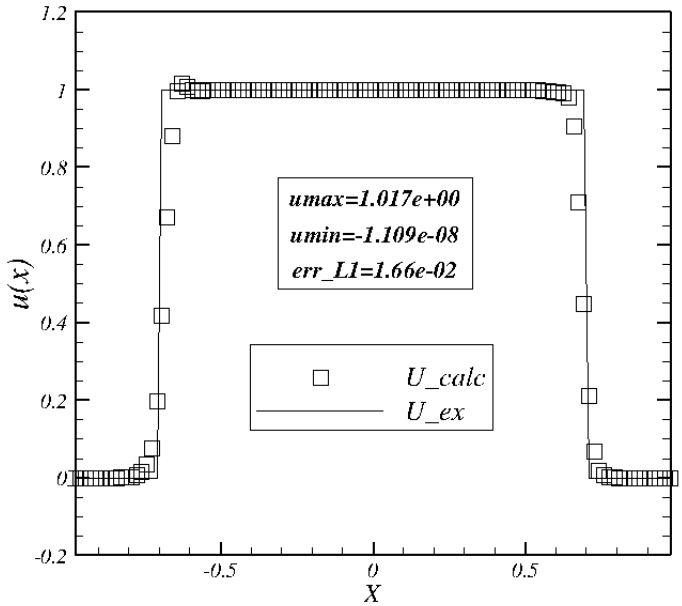


Figure 37: CUPS5 scheme: Two-dimensional linear advection of a discontinuity at $T = 2$, for $h = 1/60$ and $CFL_{opt} = 0.125$.

The case of the HUPS6 scheme is somewhat particular. Indeed, considering its significant sensitivity to spurious errors and since the spurious modes are not sufficiently damped, its numerical solution becomes inconsistent with the actual solution.
Therefore, whatever the kind of solution, the HUPS6 scheme does not perform well: for all these reasons, this latter scheme is discarded in the remaining simulations.

Pursuing this series of linear tests, we consider the scalar equation, (3.1), with the following modifications for the fluxes:

$$f(u) \equiv y \cdot u, \quad g(u) \equiv -x \cdot u$$

The initial data is Gaussian: $u_o(\vec{x}) = exp\left[-100 \cdot \left[(x-1/2)^2 + y^2\right]\right]$ over the square domain, $\mathcal{D} \equiv [-1,1]^2$. The initial conditions for the spatial derivatives, $r_o(\vec{x})$ and $s_o(\vec{x})$, are directly derived from $u_o(\vec{x})$. This choice for the fluxes and the initial condition enables to model the rotation of a Gaussian hill.
This test-case is significant to evaluate the dissipative and dispersive properties of a numerical scheme. Indeed, to avoid significant extrema damping, one must resort to high-order schemes; however, in that case, slight oscillations often appear behind the convected hill. These oscillations come from phase errors and often illustrate a local loss of monotonicity.
The mesh size selected is $h = 1/40$ and computations are run for one period, up to $T = 2\pi$. Figure 38, displays the first results obtained in using the CUPS5 scheme, with $CFL_{opt} = 0.90$.

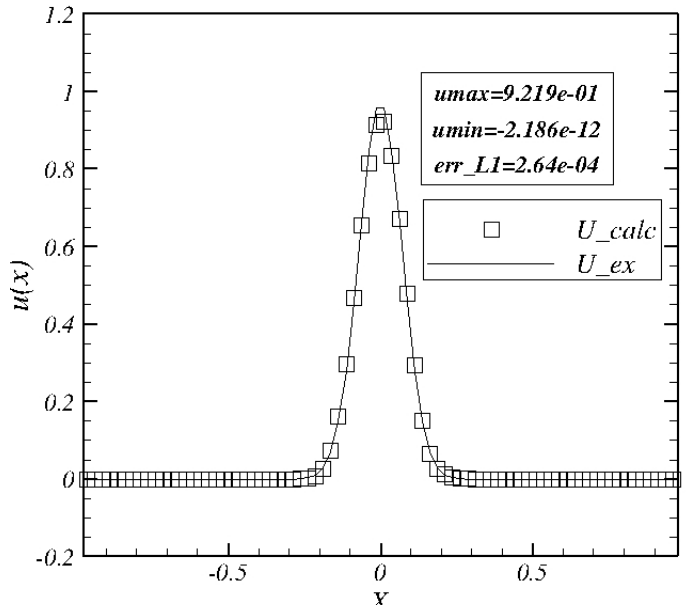


Figure 38: CUPS5 scheme: Rotation of a Gaussian hill over one period, for $h = 1/40$ and $CFL_{opt} = 0.90$.

There are no noticeable oscillations and the extrema damping is weak since $u_{max} = 0.922$; the discretization error is: $|err|_{L_1} = 2.64 \times 10^{-4}$.
Upon comparing with the HUPS4 scheme, we get results illustrated by Figure 39, for $CFL_{opt} = 1.0$: this high value for the CFL number is permitted by the

five stages of the SSPRK procedure (see [14] and appendix A).
The improvement appears when considering the maximum value computed: $u_{max} = 0.947$. Again, there are no oscillations behind the signal and the discretization error is lower: $|err|_{L_1} = 9.33 \times 10^{-5}$.

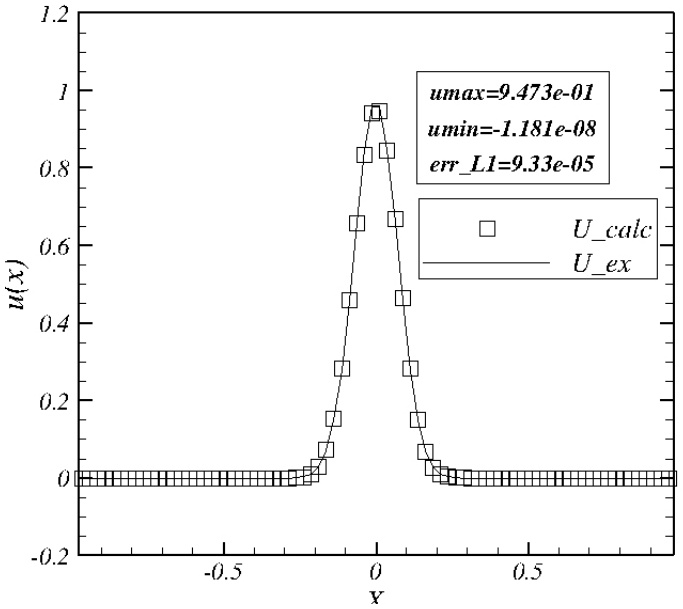


Figure 39: HUPS4 scheme: Rotation of a Gaussian hill over one period, for $h = 1/40$ and $CFL_{opt} = 1.0$.

Lastly, Figure 40 displays numerical results for the HUPS5 scheme.
This time, the optimum CFL number may be selected greater than one and we have: $CFL_{opt} = 1.30$. These are the best results among the numerical solutions investigated since we have: $u_{max} = 0.956$ and $|err|_{L_1} = 2.94 \times 10^{-5}$.
In terms of computation effort, we have the following results: $t_{cpu} = 10.4s$, (HUPS5), $t_{cpu} = 8s$, (HUPS4) and $t_{cpu} = 4.8s$, (CUPS5); but, once again, to get, approximately, the same error level as HUPS5 when using the CUPS5 scheme, we need $h = 1/60$ and then we now have: $t_{cpu} = 15.5s$.
Consequently, there is an advantage in using a HUPS scheme when dealing with isolated extrema; in some sense, this finding is in agreement with the previous one concerning the good shock capturing capabilities of a HUPS scheme.

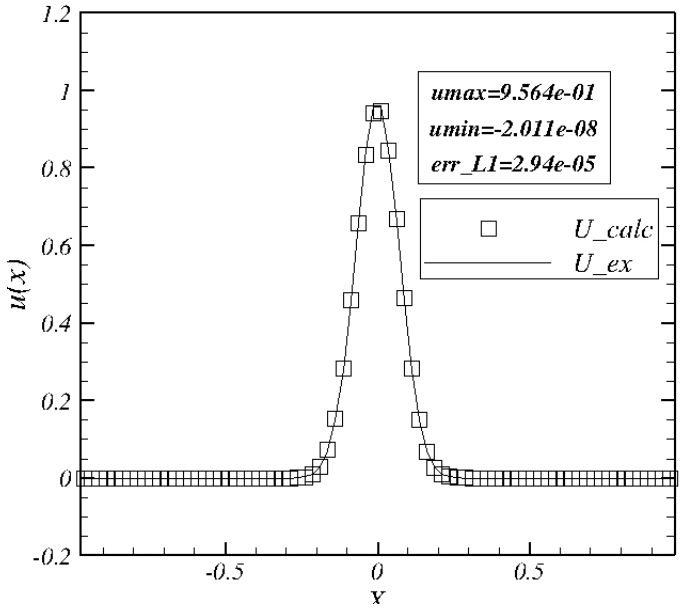


Figure 40: HUPS5 scheme: Rotation of a Gaussian hill over one period, for $h = 1/40$ and $CFL_{opt} = 1.30$.

Test2 : Two-dimensional non-linear advection.

Now, we want to check the previous conclusions in a non-linear context. To this aim, we study the 2D Burgers equation. In such a case, the non-linear form of equation (3.1) is obtained by the following fluxes:

$$f(u) \equiv g(u) \equiv \frac{u^2}{2}$$

with the initial data, defined for $\mathcal{D} \equiv [-2, 2]^2$:

$$\begin{cases} u_o(\vec{x}) = \frac{1}{4} + \frac{1}{2} sin\left(\frac{\pi}{2}(x+y)\right) \\ r_o(\vec{x}) = s_o(\vec{x}) = h \cdot \frac{\pi}{4} cos\left(\frac{\pi}{2}(x+y)\right) \end{cases}$$

The exact solution for this problem is computed from the "method of characteristics".
Firstly, the L_1- and $L_\infty-$norms of the discretization error, are collected for the dimensionless time, $T = 0.30$, when the exact solution is still smooth. Table 5, synthesizes those first results for the CUPS5 scheme, with $CFL_{opt} = 0.50$.

Both in L_1- and $L_\infty-$norms, the accuracy asymptotically converges towards the value 4: the error made in approximating the unsteady term predominates in that case because we used definition (3.21) for computing the discrete time-step.

N	L_1-error	order	L_∞-error	order
20	9.36×10^{-5}	-	5.60×10^{-4}	-
40	5.30×10^{-6}	4.1	4.60×10^{-5}	3.6
80	2.52×10^{-7}	4.4	2.30×10^{-6}	4.4
160	1.20×10^{-8}	4.4	1.60×10^{-7}	4
320	7.96×10^{-10}	4	9.60×10^{-9}	4

Table 5: CUPS5 scheme. 2D Burgers equation at $T = 0.30$.

Now, if we compute the same problem with the HUPS4 scheme, we get numerical results typified by Table 6, for $CFL_{opt} = 0.25$.

N	L_1-error	order	L_∞-error	order
20	4.15×10^{-5}	-	2.60×10^{-4}	-
40	2.39×10^{-6}	4.1	2.90×10^{-5}	3.2
80	1.0×10^{-7}	4.6	2.60×10^{-6}	3.2
160	3.70×10^{-9}	4.8	2.30×10^{-7}	3.5
320	1.60×10^{-10}	4.6	2.10×10^{-8}	3.5

Table 6: HUPS4 scheme. 2D Burgers equation at $T = 0.30$.

This time, the rate of convergence of the discretization error, in L_1-norm, is greater, by more than an order, than in $L_\infty-$norm. This discrepancy, which does not come from the monotonicity-preserving procedure, is difficult to explain. A decrease of the CFL number does not modify significantly this result. Some minor nonlinear instabilities might appear into the numerical solution and might trigger the spurious modes of the numerical scheme with, as consequence, a local increase of the spurious error and, then, an alteration of the resulting accuracy.
Despite this detrimental behavior, one must note that the error levels are lower in L_1-norm than those of the CUPS5 scheme and almost equivalent in $L_\infty-$norm.
Lastly, Table 7 shows numerical results obtained displayed by the HUPS5 scheme, for CFL_{opt}=0.125.

N	L_1-error	order	L_∞-error	order
20	6.60×10^{-5}	-	5.90×10^{-4}	-
40	4.30×10^{-6}	4	4.50×10^{-5}	3.7
80	1.90×10^{-7}	4.5	2.50×10^{-6}	4.2
160	7.50×10^{-9}	4.7	9.70×10^{-8}	5
320	2.90×10^{-10}	4.8	2.10×10^{-9}	4.8

Table 7: HUPS5 scheme. 2D Burgers equation at $T = 0.30$.

As one can see it, the improvement essentially appears in the $L_\infty-$error when passing from HUPS4 to HUPS5 scheme, since the error level is lower and the rate of convergence of this quantity is higher, when using the HUPS5 scheme. As the accuracy of the interpolation polynomial is pretty much the same in both cases, the main difference between HUPS4 and HUPS5 schemes lies on the discretization stencil; a wider stencil seems to contribute to reducing the nonlinear excitation of the spurious components of the numerical solution, at least for the test-case investigated, here.
The rate of convergence of the discretization error, in both norms, seems to indicate that errors made by the spatial discretization predominate over temporal errors. Note, also, that the monotonicity-preserving procedure does not alter the accuracy of the scheme.
Lastly the three schemes may be compared in terms of the computation effort, for a given accuracy; Figure 41, displays these results.

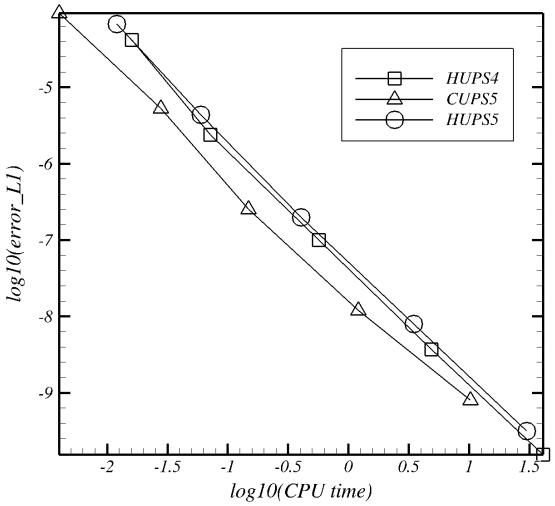


Figure 41: 2D Burgers equation: Comparisons of the computation effort for the CUPS5, HUPS4 and HUPS5 schemes

Both HUPS4 and HUPS5 schemes give equivalent results; on the other side, the CUPS5 scheme is more advantageous in terms of computation effort.
Therefore, it appears that this latter quantity is an important parameter in the rating of a high-order numerical method. Indeed, despite higher error levels and lower rates of convergence, the CUPS5 scheme is less costly in CPU time than a HUPS scheme with an equivalent accuracy.

Now, if we compute the numerical solution at $T = 0.80$, a moving shock develops and interacts with a rarefaction wave. The numerical solution is drawn in Figures 42, 43, 44, for the CUPS5, HUPS4, HUPS5 schemes, respectively, with $h = 1/20$ and $CFL_{opt} = 0.250$.
We see that the monotonicity of the solution is preserved between the shocks that are accurately captured without any spurious oscillation.
The solution computed from the CUPS5 scheme is visually identical with either that of the HUPS4, Figure 43, or the HUPS5 scheme, Figure 44.
Similarly, upon looking at the discretization error, the differences are very weak: $|err|_{L_1} = 1.45 \times 10^{-2}$ (CUPS5), $|err|_{L_1} = 1.46 \times 10^{-2}$ (HUPS4), $|err|_{L_1} = 1.40 \times 10^{-2}$ (HUPS5).

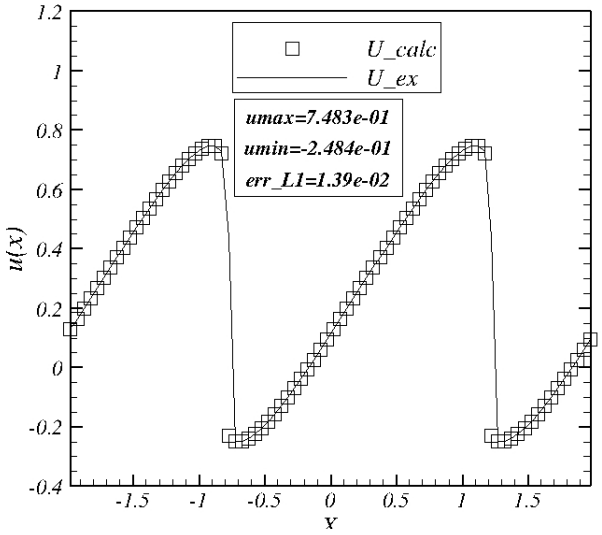


Figure 42: CUPS5 scheme: 2D Burgers equation at $T = 0.80$, for $h = 1/20$ and $CFL_{opt} = 0.250$.

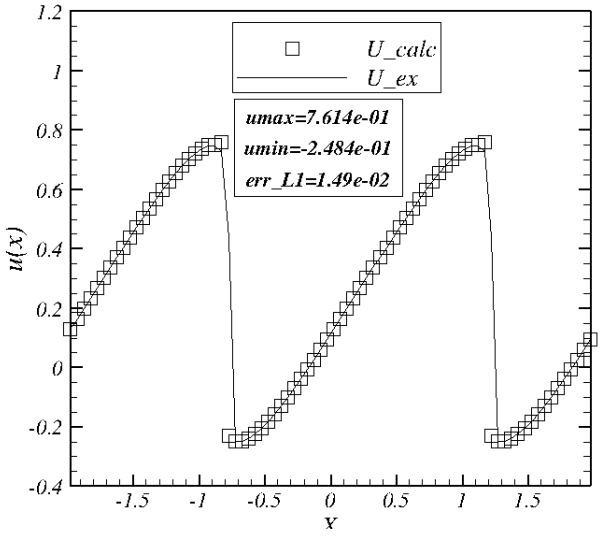


Figure 43: HUPS4 scheme: 2D Burgers equation at $T = 0.80$, for $h = 1/20$ and $CFL_{opt} = 0.250$.

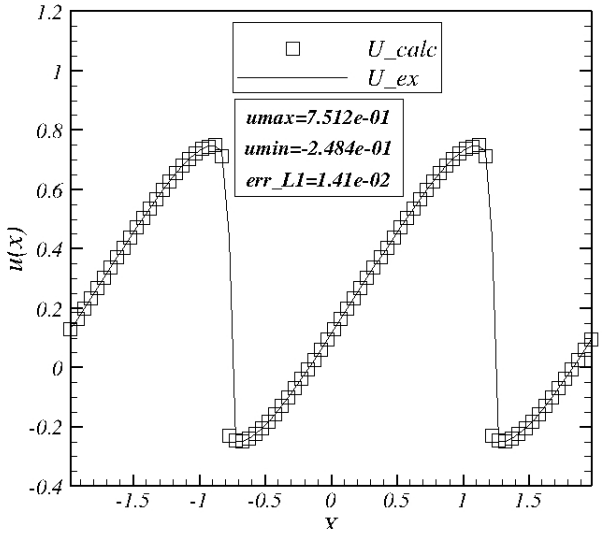


Figure 44: HUPS5 scheme: 2D Burgers equation at $T = 0.80$, for $h = 1/20$ and $CFL_{opt} = 0.250$.

Consequently, the choice of the high-order version of a scheme to compute this test-case does not appear to be a crucial point; the choice of the monotonicity-preserving procedure is undoubtedly more prominent.

4 Conclusions and perspectives

In this paper we designed a high-order and compact numerical method that is weakly dissipative and dispersive, while remaining non-oscillatory.
An Hermitian formulation of the differential problem allows to generate a compact numerical scheme (HUPS scheme) without sacrificing numerical accuracy; in addition, an upwind formulation of the numerical fluxes (HLL formulation) reduces phase errors and increases the stability of the numerical method.
When there are discontinuities in the solution or when the solution presents isolated extrema, a HUPS formulation becomes advantageous. This is particularly true for the computation of linear discontinuities for which we demonstrated that an HUPS scheme gives better results than a classical scheme (CUPS) of the same order.
On the other hand, a CUPS scheme generally performs better than a HUPS scheme of the same order as long as the solution remains smooth and provided that boundary conditions do not play a critical role in the accuracy of that solution; the CUPS5 scheme (fifth-order accuracy in space) is a good choice in that case.
In cases of complex solutions with embedded discontinuities, a HUPS scheme is preferable; the HUPS5 scheme we designed in this paper is a good candidate

for this kind of problems.

However, an Hermitian formulation of a differential problem has some drawbacks, also.
Firstly, at the discrete level, any Hermitian formulation introduces spurious components into the numerical solution; these components have no equivalent in the differential problem. Depending on the formulation of the Hermitian scheme, these spurious components may deeply alter not only the theoretical convergence properties of the resulting scheme, but also the consistency of the numerical solution.
Secondly, an Hermitian formulation of a differential problem results in an increased algebraic complexity when compared with a classical formulation; this has consequences on the computation effort.

These drawbacks trace the way for future improvements of an Hermitian numerical method.
For instance, the leading equations for the spatial derivatives might be linearized in order to decrease the algebraic complexity and to weaken the non-linearities that are responsible for a triggering of the spurious components of the numerical solution.
On the other side, extension of this work to non Cartesian geometries may be developed by using the least-square procedure designed in [12], [13].

Appendix A Preserving positivity time-integration method.

In order to preserve the positivity properties of the scheme in its high-order time-integration version, we made the choice of the fourth-order, five stages, SSPRK(5,4) scheme of Gottlieb,[14].

For example, by writing formally equation (2.2) as:

$$\left(\frac{d\bar{u}}{\mathrm{dt}}\right)_i \equiv \mathcal{F}(\bar{u}(t)) \tag{A.1}$$

integration of this equation by the SSPRK(5,4) scheme, gives the following algorithm:

Algorithm SSPRK(5,4), [14].

$$\begin{aligned}
u_i^{(1)} &\equiv \bar{u}_i^n + 0.391752226571890\Delta t\mathcal{F}(\bar{u}(t_n)) \\
u_i^{(2)} &\equiv 0.444370493651235\bar{u}_i^n + 0.555629506348765u_i^{(1)} \\
&\quad + 0.368410593050371\Delta t\mathcal{F}(u^{(1)}) \\
u_i^{(3)} &\equiv 0.620101851488403\bar{u}_i^n + 0.379898148511597u_i^{(2)} \\
&\quad + 0.251891774271694\Delta t\mathcal{F}(u^{(2)}) \\
u_i^{(4)} &\equiv 0.178079954393132\bar{u}_i^n + 0.821920045606868u_i^{(3)} \\
&\quad + 0.544974750228521\Delta t\mathcal{F}(u^{(3)}) \\
\bar{u}_i^{n+1} &\equiv 0.517231671970585u_i^{(2)} + 0.096059710526147u_i^{(3)} \\
&\quad + 0.063692468666290\Delta t\mathcal{F}(u^{(3)}) + 0.386708617503269u^{(4)} \\
&\quad + 0.226007483236906\Delta t\mathcal{F}(u^{(4)})
\end{aligned} \tag{A.2}$$

The advantage in using this algorithm is twofold: firstly, the resulting scheme is fourth-order accurate in time and, moreover, it remains monotonicity-preserving; secondly, this time-integration method is more efficient than the popular SSPRK(3,3) method (see also [14]), with an effective CFL number ($\equiv$ CFL $/m$, m:number of computations of $\mathcal{F}(\bar{u}(t))$ required per time step) of 0.377 instead of 1/3 for SSPRK(3,3).

Appendix B Spectral analysis for the discretization of the linear advection equation.

B.1 Classical Upwind Scheme (CUPS)

We start this analysis from the linear semi-discrete form (2.2) obtained over an uniform Cartesian mesh ($\Delta x \equiv \text{Cte} \equiv h$) :

$$\frac{d\bar{u}_i}{\text{dt}} + a\frac{u^L_{i+1/2} - u^L_{i-1/2}}{h} = 0 \quad (a \equiv \text{Cte} > 0) \tag{B.1}$$

This form is supposed to be the semi-discrete form obtained by a classical upwind scheme over the stencil typified by Figure 1. The pointwise values, $u_{i\pm1/2}$ are obtained, at $t = t_n$, from the interpolation polynomials, $\tilde{u}_i(x)$ and $\tilde{u}_{i-1}(x)$, computed according to conditions (2.4).

Now, assuming that the computation domain is periodic, we decompose the semi-discrete solution, $\bar{u}_i(t)$, in Fourier series, according to the following definition:

$$\overline{u}_i(t) \equiv \hat{u}(t)e^{\text{jk}\,x_i} \quad (j^2 = -1) \tag{B.2}$$

where $\hat{u}(t)$ represents the complex amplitude of the numerical solution and k, the wave number that characterizes the periodic signal.

Inserting (B.1) into the linear form, (B.1), produces the following result:

$$\frac{d\hat{u}}{\text{dt}} = G(\beta,\nu).\hat{u}(t) \tag{B.3}$$

$G(\beta,\nu)$ is a complex factor that represents the discrete Fourier transform of the discrete spatial operator in (B.1). This quantity only depends upon the phase angle, β ($\equiv kh$) and the CFL number, ν ($\equiv a\Delta t/h$).

Finally, upon integrating (B.3) over the time-interval $[t_n, t_{n+1}]$, we get, formally, the following result, in the Fourier space ($\hat{u}^{n+1} \equiv \hat{u}(t_{n+1}), \hat{u}^n \equiv \hat{u}(t_n)$)

$$\hat{u}^{n+1} = \mathcal{G}(\beta,\nu)\hat{u}^n \tag{B.4}$$

When it is a scalar quantity, $\mathcal{G}(\beta,\nu)$ is called the "amplification factor" of the numerical scheme; $\mathcal{G}(\beta,\nu)$ is a complex quantity.

To formalize $\mathcal{G}(\beta,\nu)$, let us suppose that (B.3) is integrated by using a Runge-Kutta positive scheme, [14] (Appendix A). Then, each step, m, of the integration procedure, will produce the following recursive result:

$$\hat{u}^m = \hat{u}^n\text{-}\alpha_m G(\beta,\nu)\hat{u}^{m-1} \tag{B.5}$$

where, α_m, represents the coefficient of the Runge-Kutta scheme at the m-th stage.

Combining those results for the whole procedure, we have, therefore, for a p-th order Runge-Kutta scheme, the final result:

$$\hat{u}^{n+1} = (1\text{-}\alpha_p G.\,(1\text{-}\alpha_{p\text{-}1}G.(\,1\text{-}\ldots(1\text{-}\alpha_1 G)\,)\,.\hat{u}^n \tag{B.6}$$

and the amplification factor, $\mathcal{G}(\beta,\nu)$, can now be directly identified as:

$$\mathcal{G}(\beta,\nu) \equiv (1\text{-}\alpha_p G.\,(1\text{-}\alpha_{p\text{-}1} G.(\,1\text{-}\ldots(1\text{-}\alpha_1 G)\,)\,(\beta,\nu) \tag{B.7}$$

Thus, $\mathcal{G}(\beta,\nu)$ can be compared with the exact amplification factor obtained from a continuous Fourier transform of the linear advection equation, (2.1). Indeed, if we introduce into (2.1), for a given wave number, k, the decomposition: $u(x,t) = \hat{u}(t)e^{\mathrm{jk}\,x}$, then, we get the following ODE:

$$\frac{d\hat{u}}{\mathrm{dt}} + \mathrm{jka}\,\hat{u}(t) = 0 \tag{B.8}$$

By integrating exactly this result over $[t_n, t_{n+1}]$, we obtain:

$$\hat{u}^{n+1} = e^{-j\nu\beta}\hat{u}^n \tag{B.9}$$

This latter result is unchanged if we consider the integrated form of (2.1) with the decomposition $\overline{u}_i(t) \equiv \hat{u}(t)e^{\mathrm{jk}\,x_i}$, for the variable, $\bar{u}(x,t)$.
Therefore, since equation (B.4) represents the approximation of equation (B.9), in the Fourier space, $\mathcal{G}(\beta,\nu)$ may be considered as the result of the numerical approximation by the numerical scheme, of the exact amplification operator, $e^{-j\nu\beta}$, obtained from equation (B.9).

In agreement with this statement, one may define the following useful quantities for the study of the spectral properties of a numerical scheme, with $\nu \in [0,1]$, $\beta \in [0,\pi]$:

- The amplitude error : $A(\beta,\nu) \equiv 1 - |\mathcal{G}(\beta,\nu)|$
- The relative phase error: $P(\beta,\nu) \equiv [\nu\beta + \mathrm{Arg}(\mathcal{G}(\beta,\nu))]\,/\nu\beta$
- The truncation error: $\tau(\beta,\nu) \equiv \frac{1}{\Delta t}[e^{-j\nu\beta} - \mathcal{G}(\beta,\nu)]$
- The discretization error, at $t = t_n$: $e_n(\beta,\nu) \equiv |\mathcal{G}^n(\beta,\nu) - e^{-jn\nu\beta}|$
- The accuracy at $t = t_n$: $\dfrac{[log_{10}\,(e_n(\beta_0,\nu)) - log_{10}\,(e_n(\beta_1,\nu))]}{(log_{10}(\beta_0) - log_{10}(\beta_1))}$

In addition, according to the classical Von-Neumann linear stability analysis, the quantity, $|\mathcal{G}(\beta,\nu)|$, makes it possible to evaluate the stability of the numerical scheme discretizing (2.1).

Of course, the truncation error, $\tau(\beta,\nu)$, in the Fourier space, may be linked with the classical definition of the truncation error, $\tau(x,t)$, for a numerical scheme, by the following relationship:

$$\mathrm{TF}(\tau(x,t)) = \tau(\beta,\nu)\hat{u}e^{\mathrm{jk}\,x} \tag{B.10}$$

Indeed, if we integrate (B.1) over $[t_n, t_{n+1}]$, we get the following condensed algebraic form:

$$\bar{u}_i^{n+1} = \mathcal{H}_{\Delta t}(\bar{u}^n; i) \tag{B.11}$$

From this result, the local truncation error is usually defined as:

$$\tau(x_i, t_n) \equiv \frac{1}{\Delta t}\left[\bar{u}(x_i, t_{n+1}) - \mathcal{H}_{\Delta t}(\bar{u}^n; i)\right] \tag{B.12}$$

If we introduce decomposition (B.2), at $t = t_{n+1}$, into this latter definition and if we use result (B.9), then we get the following result:

$$TF(\tau(x_i, t_n)) = \frac{1}{\Delta t}\left[e^{-j\nu\beta} - \mathcal{G}(\beta, \nu)\right]\hat{u}^n e^{jkx_i} \tag{B.13}$$

where we used the following result: $TF(\mathcal{H}_{\Delta t}(\bar{u}^n; i)) = \mathcal{G}(\beta, \nu)$.
Then, it becomes easy to identify $\tau(\beta, \nu)$ in a practical way.
Of course, we keep on having, in the Fourier space, the consistency condition:

$$\tau(\beta, \nu) \underset{\beta \to 0}{\to} 0 \tag{B.14}$$

More specifically, for the high frequency components of the numerical solution ($\beta \ll 1$), a Taylor series expansion of $\tau(\beta, \nu)$ gives the following typical result, for a $p - th$ order (space and time) discretization scheme of (B.1):

$$\tau(\beta, \nu) = \frac{1}{\Delta t}\left[-a_1(\nu)\,(j\beta)^{p+1} - a_2(\nu)\,(j\beta)^{p+2}\right] + \mathcal{O}\left(\frac{\beta^{p+3}}{\Delta t}\right) \tag{B.15}$$

Upon starting from this result, we can identify the truncation error in the physical space, $\tau(x, t)$, from an inverse Fourier transform ($\tau(x, t) = TF^{-1}\left(\tau(\beta, \nu)\hat{u}e^{\mathrm{j}\,\mathrm{k}\,x}\right)$), to produce the following useful result:

$$\begin{aligned}\tau(x, t) = &-a_1(\nu)\,(-1)^p\,\frac{\Delta x^{p+1}}{\Delta t}\frac{\delta^{p+1}u}{\delta x^{p+1}} \\ &- a_2(\nu)\,(-1)^{p+1}\,\frac{\Delta x^{p+2}}{\Delta t}\frac{\delta^{p+2}u}{\delta x^{p+2}} + \mathcal{O}\left(\frac{\Delta x^{p+3}}{\Delta t}\right)\end{aligned} \tag{B.16}$$

Consequently, the modified equation of (2.1), by the discretization scheme that generates the two first term of $\tau(x, t)$, may be defined as:

$$u_t + \mathrm{a}u_x = a_1(\nu)\,(-1)^p\,\frac{\Delta x^{p+1}}{\Delta t}\frac{\delta^{p+1}u}{\delta x^{p+1}} + a_2(\nu)\,(-1)^{p+1}\,\frac{\Delta x^{p+2}}{\Delta t}\frac{\delta^{p+2}u}{\delta x^{p+2}} \tag{B.17}$$

Finally, by using, once again, a continuous Fourier transform of the exact solution of this modified equation, we may appreciate the characteristic behavior of the numerical solution.

Indeed, the Fourier transform of (B.17) ($u(x, t) = \hat{u}(t)e^{jkx}$) integrated over $[t_n, t_{n+1}]$, produces the following exact result for the complex amplitude, $\hat{u}$:

$$\hat{u}^{n+1} = e^{-j\nu\beta}\hat{u}^n e^{(-1)^p a_1(\nu)\frac{(j\beta)}{\Delta t}^{p+1}} e^{(-1)^{p+1} a_2(\nu)\frac{(j\beta)}{\Delta t}^{p+2}} \tag{B.18}$$

Now, this latter result must be compared with that of equation (B.9).

Thus, if p is even, the leading error (first term of the truncation error), between exact solution (B.9) and solution (B.18), which typifies the numerical

scheme, is a phase error; whereas the higher order term of (B.16) gives us information about the dissipative behavior of this scheme, depending on the sign of $a_2(\nu)$.
On the other side, if p is odd, the leading error will be, henceforth, an amplitude error; in that case the second term of (B.16) will introduce a phase error of higher order.

B.2 Hermitian Upwind Scheme (HUPS)

Now, if we suppose that equation (2.1) is discretized by an Hermitian upwind scheme, for example over the compact stencil typified by Figure 2, some significant modifications must be made to the above considerations.

First of all, we must start from the following semi-discrete linear system:

$$\frac{d\bar{U}_i}{\mathrm{dt}} + a\frac{U^L_{i+1/2} - U^L_{i-1/2}}{h} = 0 \quad (a \equiv \mathrm{Cte} > 0) \tag{B.19}$$

with: $U \equiv \left(u, r \equiv h.\frac{\partial u}{\partial x}\right)^t$ and $\bar{U} \equiv \left(\bar{u}, \bar{r} \equiv u(x_{i+1/2}) - u(x_{i-1/2})\right)^t$.

Now, if we do a Fourier transform of the spatial operators of this system:

$$\bar{U}_i(t) = \left[\begin{array}{c} \hat{u}(t) \\ \hat{r}(t) \end{array} \right] \times e^{jkx_i} \equiv \hat{U}(t) \times e^{jkx_i} \tag{B.20}$$

we obtain, formally, a result equivalent to equation (B.3):

$$\frac{d\hat{U}}{\mathrm{dt}} = G(\beta, \nu) \cdot \hat{U}(t) \tag{B.21}$$

But now, the quantity, $G(\beta, \nu)$, becomes a 2×2 complex matrix and $\hat{U}$ is the vector of complex amplitudes, $(\hat{u}, \hat{r})^t$.

Then, upon integrating (B.21) over $[t_n, t_{n+1}]$ with a positive Runge-Kutta algorithm, we get:

$$\hat{U}^{n+1} = \mathcal{G}(\beta, \nu) \cdot \hat{U}^n \tag{B.22}$$

where $\mathcal{G}(\beta, \nu)$ is a 2×2 complex matrix; $\mathcal{G}(\beta, \nu)$ is called the "amplification matrix" of the Hermitian scheme and this result is formally tantamount to the scalar equation (B.4), for a classical scheme.

Secondly, the spectral properties of the Hermitian scheme are influenced by the complex eigenvalues of $\mathcal{G}(\beta, \nu)$.
Indeed, the complex scalar amplification factor, $g(\beta, \nu)$, for the Hermitian system is such that we have:

$$\hat{U}^{n+1} = g(\beta, \nu) \times \hat{U}^n \tag{B.23}$$

Therefore, if we introduce this definition into (B.22), we get the following condition:

$$(\mathcal{G}(\beta,\nu) - g(\beta,\nu) \times \mathcal{I})\,\hat{U}^n = 0 \tag{B.24}$$

($\mathcal{I}$: identity matrix) showing that the only condition for having a solution different from the trivial solution, $\hat{U}^n = 0$, is the quantity, $g(\beta,\nu)$, to be an eigenvalue of $\mathcal{G}(\beta,\nu)$.

If we calculate those eigenvalues, defining the first eigenvalue, $\lambda_1(\beta,\nu)$, as approximating the exact amplification factor, $e^{-j\nu\beta}$, of (2.1) in the complex plane, then $\lambda_1(\beta,\nu)$ is identified, practically, by the consistency condition:

$$\lambda_1(\beta,\nu) \underset{\beta\to 0}{\to} 1 \tag{B.25}$$

In that case, the second eigenvalue, $\lambda_2(\beta,\nu)$, represents a "spurious component", introduced by an Hermitian discretization into the solution, and has no equivalent in the continuous problem (2.1).

In order to illustrate this latter property, let us suppose that the initial solution of (2.1) is: $u(x,t=0) = \hat{u}^0 e^{jkx}$, a sinusoidal solution with a wavelength, $2\pi/k$ and an amplitude $\hat{u}^0$.
Then, the exact solution of (2.1), at any time t, is as follows:

$$u(x,t) = \hat{u}^0 e^{-j\nu\beta\frac{t}{\Delta t}} e^{jkx} \tag{B.26}$$

and, for an Hermitian problem, this exact solution can be written as:

$$U(x,t) \equiv \begin{bmatrix} u \\ h\frac{\partial u}{\partial x} \end{bmatrix} = e^{-j\nu\beta\frac{t}{\Delta t}} \hat{u}^0 \begin{bmatrix} 1 \\ j\beta \end{bmatrix} e^{jkx} \tag{B.27}$$

or, equivalently, at $x = x_i$ and $t = t_n \equiv n\Delta t$ and by setting: $\hat{U}^0 \equiv \hat{u}^0 \times [1, j\beta]^t$:

$$\boxed{U(x_i, t = t_n) = \hat{U}^0 e^{-jn\nu\beta} e^{jkx_i}} \tag{B.28}$$

The vector, $\hat{U}^0 \equiv \hat{u}^0 \times [1, j\beta]^t$, is the complex amplitude of the full solution and it also represents the exact eigenvector of the Hermitian system, in the Fourier space.
Now, let us compare this exact solution with the numerical solution, U_i^n, generated by an HUPS scheme.
To begin, let us suppose that the initial first space derivative, $\bar{r}_i^0$, is approximated by a centered discretization at $x = x_i$; then, we write:

$$\bar{r}_i^0 \approx \left(\bar{u}_{i+1}^0 - \bar{u}_{i-1}^0\right)/2 \tag{B.29}$$

or, by using the Fourier decomposition: $\bar{u}_i^0 = \hat{u}^0 e^{jkx_i}$:

$$\bar{r}_i^0 = jsin\beta\hat{u}^0 e^{jkx_i} \tag{B.30}$$

Consequently, the first approximation generated by the HUPS scheme in initializing the numerical solution, reads as:

$$\hat{u}^0 \cdot \begin{bmatrix} 1 \\ jsin\beta \end{bmatrix} \sim \hat{U}^0 \tag{B.31}$$

Now, by using relationship, (B.22), it becomes possible to formulate, at $t = t_n$, the approximate solution, U_i^n, generated by an HUPS discretization typified by $\mathcal{G}(\beta, \nu)$:

$$U_i^n = \mathcal{G}^n \hat{u}^0 \cdot \begin{bmatrix} 1 \\ jsin\beta \end{bmatrix} e^{jkx_i} \tag{B.32}$$

If we use the decomposition: $\mathcal{G} \equiv R\Lambda R^{-1}$, where R stands for the right eigenvectors matrix of $\mathcal{G}$ and Λ its matrix of eigenvalues, we can re-formulate (B.32) as follows:

$$U_i^n = R\Lambda^n R^{-1} \hat{u}^0 \cdot \begin{bmatrix} 1 \\ jsin\beta \end{bmatrix} e^{jkx_i} \tag{B.33}$$

Looking at this latter result, one may see that two sources of approximation appear into the solution:

- a spatial and a time approximations, typified by $\mathcal{G}^n(\beta, \nu) = R\Lambda^n R^{-1}$
- an initialization error, which is represented by the approximate vector, $\hat{u}^0 \cdot [1, jsin\beta]^t$.

Lastly, by setting:

$$\hat{W}^0 \equiv R^{-1} \hat{u}^0 \cdot \begin{bmatrix} 1 \\ jsin\beta \end{bmatrix} \equiv \begin{bmatrix} \hat{w}_1^0 \\ \hat{w}_2^0 \end{bmatrix}$$

the following significant result is obtained:

$$\boxed{U_i^n = R\Lambda^n \hat{W}^0 e^{jkx_i} = \left(\lambda_1^n \hat{w}_1^0 r_1 + \lambda_2^n \hat{w}_2^0 r_2\right) e^{jkx_i}} \tag{B.34}$$

where, r_p, represents the $p-th$ right eigenvector of $\mathcal{G}(\beta, \nu)$, associated with the $p-th$ eigenvalue, λ_p.

Therefore, if we compare (B.34) with equation (B.28), since the eigenvalue, $\lambda_1(\beta, \nu)$, is defined as the consistent approximation of the exact amplification factor, $e^{-j\nu\beta}$, one may say that the numerical solution at $t = t_n$ is the sum of two components: an "accurate" component, $\lambda_1^n \hat{w}_1^0 r_1$, which should be compared to $\hat{U}^0 e^{-jn\nu\beta}$, and a "spurious" component, $\lambda_2^n \hat{w}_2^0 r_2$, that does not have counterpart in the differential problem; the amplitude of this latter component is given by the modulus of the complex quantity, $\hat{w}_2^0$.

Thus, one can define in the Fourier space the following new quantities, for the first component, u, of U:

- an "accurate" error: $|\lambda_1^n \hat{w}_1^0 r_{11} - e^{-jn\nu\beta}|$.

- a "spurious" error: $|\lambda_2^n \hat{w}_2^0 r_{21}|$.

- a discretization error: $|\lambda_1^n \hat{w}_1^0 r_{11} + \lambda_2^n \hat{w}_2^0 r_{21} - e^{-jn\nu\beta}|$

If $|\lambda_2| < 1$, whatever $\nu \in [0, 1]$ and whatever $\beta \in [0, \pi]$, then the "spurious" error sized by $|\hat{w}_2^0|$ has a decreasing influence as time evolves, even though this latter error is significant at $t = 0$ (initialization).
The issue becomes problematic as soon as $|\lambda_2| > 1$: in that case, if $|\hat{w}_2^0| \neq 0$, then, according to (B.34), the numerical solution may become inconsistent when the spurious component overwhelms the accurate component of the solution. Therefore, at least theoretically, an Hermitian scheme, even though it is designed to be linearly stable ($|\lambda_1| < 1, \quad \forall \nu \in [0, 1], \quad \forall \beta \in [0, \pi]$) and consistent with the differential problem, may generate spurious components within its solution, with as consequence the need to lower the CFL number more than necessary, to damp this unwelcome behavior.
To end, it is significant to note that the way the numerical solution is initialized plays a role in the definition of the discretization error, contrarily to the definition we gave for this latter quantity when considering the CUPS scheme.

Appendix C Spectral analysis: two-dimensional linear advection equation.

To begin, we start from the classical (CUPS) numerical form that discretizes the two-dimensional linear advection equation, (2.19):

$$\frac{d\bar{u}_i}{dt} + a\frac{u^L_{i+1/2} - u^L_{i-1/2}}{h} + b\frac{u^L_{j+1/2} - u^L_{j-1/2}}{h} = 0 \quad (a \equiv Cte > 0, b \equiv Cte > 0) \tag{C.1}$$

with: $h \equiv \Delta x = \Delta y \equiv Cte$.

Once again, by introducing the decomposition: $u(\vec{x}, t) = \hat{u}(t)e^{j\vec{k}\vec{x}}$, into equation (2.19), for the given wave vector, $\vec{k} \equiv (k_x, k_y)^t$, and after integrating over $[t_n, t_{n+1}]$, we get the following exact solution:

$$\begin{cases} \hat{u}^{n+1} = \mathcal{G}_e(\nu, \beta, \psi)\hat{u}^n \\ \\ \mathcal{G}_e(\nu, \beta, \psi) \equiv e^{-j\nu\beta cos\psi} \end{cases} \tag{C.2}$$

Where we defined the following quantities:

$$\beta \equiv \left(\beta_x^2 + \beta_y^2\right)^{1/2}, \ \nu \equiv \frac{\sqrt{a^2 + b^2}\Delta t}{h}, \ \psi \equiv \theta - \varphi$$

with: $\beta_{x,y} \equiv k_{x,y}h$, and: $\theta = \arctan(b/a)$, $\varphi \equiv \arctan(\beta_y/\beta_x)$.

Thus, to deal with a two-dimensional spectral analysis, we have to add, now, a new parameter, ψ, which links together, the convection angle, θ, and the wave angle, φ.

Note that, the parameter, N, remains associated with the two-dimensional phase angle, β, by the relationship: $N \times \beta = 2\pi$, with the same meaning as in the one-dimensional case.

Similarly, a discrete Fourier transform of the numerical solution of (C.1) computed by the CUPS scheme and integrated by a Runge-Kutta scheme over $[t_n, t_{n+1}]$, gives the following relationship:

$$\hat{u}^{n+1} = \mathcal{G}(\beta, \nu, \psi)\hat{u}^n \tag{C.3}$$

The scalar numerical amplification factor, $\mathcal{G}(\beta, \nu, \psi)$, must now be compared with $\mathcal{G}_e(\beta, \nu, \psi)$, to compute the quantities defined in Appendix B.

On the other side, when we approximate (2.19) by an Hermitian scheme (HUPS), we get the following semi-discrete result:

$$\frac{d\bar{U}_i}{dt} + a\frac{U^L_{i+1/2} - U^L_{i-1/2}}{h} + b\frac{U^L_{j+1/2} - U^L_{j-1/2}}{h} = 0 \tag{C.4}$$

with: $\bar{U}_i \equiv \left(\bar{u}_i, \bar{r}_i \equiv \frac{h}{|I_i|}\int_{I_i}\frac{\partial u}{\partial x}dxdy, \bar{s}_i \equiv \frac{h}{|I_i|}\int_{I_i}\frac{\partial u}{\partial y}dxdy\right)^t$.

Therefore, a discrete Fourier transform of the numerical solution, will display the following relationship:

$$\hat{U}^{n+1} = \mathcal{G}(\beta, \nu, \psi)\hat{U}^n \tag{C.5}$$

with, now, the following meaning for $\hat{U}$: $\hat{U} \equiv [\hat{u}, \hat{r}, \hat{s}]^t$, vector of the complex amplitudes of the variable, the first derivative in x and the first derivative in y, respectively.

Consequently, $\mathcal{G}(\beta, \nu, \psi)$ becomes a 3×3 complex matrix when computing a two-dimensional Hermitian scheme.
Once again, by analyzing the eigenvalues of $\mathcal{G}(\beta, \nu, \psi)$, we can identify an "accurate" and two "spurious" components into the numerical solution, respectively associated to the eigenvalues, $\lambda_1(\beta, \nu, \psi)$ and $\lambda_{2,3}(\beta, \nu, \psi)$.

Thus, $\lambda_1(\beta, \nu, \psi)$ is now considered as the numerical approximation of $\mathcal{G}_e(\nu, \beta, \psi)$, introduced in (C.2). Practically, to study the two-dimensional properties of an HUPS scheme discretizing, (2.19), we define the following discretization errors, for $\nu \in [0, 1]$, $\beta \in [0, \pi]$ and $\psi \in [0, \pi/2]$:

- The amplitude error: $A(\beta, \nu, \psi) \equiv 1 - |\lambda_1(\beta, \nu, \psi)|$

- The relative phase error: $P(\beta, \nu, \psi) \equiv [\nu\beta + Arg(\lambda_1(\beta, \nu, \psi))]/\nu\beta$.

Unfortunately, the truncation error, due to the complexity of the algebraic form for $\lambda_1(\beta, \nu, \psi)$, cannot be computed in a simple closed form as in the one-dimensional problem.

Acknowledgments . The author wishes to thank warmly, Adrien Grellier and Vincent Degat (ECN-France), who provided a technical support essential to this work.

References

[1] Shu C.-W. Qiu J. Hermite weno schemes and their application as limiters for runge-kutta discontinuous galerkin method: one-dimensional case. *J. of Comp. Phys.*, 193:115–135, 2004.

[2] Shu C.-W. Qiu J. Hermite weno schemes and their application as limiters for runge-kutta discontinuous galerkin method ii: two-dimensional case. *Comp. Fluids*, 34:642–663, 2005.

[3] Zhu J. Qiu J. A class of the fourth-order finite volume hermite weighted essentially non-oscillatory schemes. *Sci. China Ser. A*, 51:1549–1560, 2008.

[4] Qiu J. Tao Z., Li F. High-order central hermite weno schemes on staggered meshes for hyperbolic conservation laws. *J. of Comp. Phys.*, 281:148–176, 2015.

[5] Qiu J. Tao Z., Li F. High-order central hermite weno schemes:dimension-by-dimension moment-based reconstructions. *J. of Comp. Phys.*, 318:222–251, 2016.

[6] A. H. Abdalla Y. H. Zahran. Seventh order hermite weno schemes for hyperbolic conservation laws. *Comp. and Fluids*, 131:66–80, 2016.

[7] B. van Leer. Towards the ultimate conservative difference scheme iii. upstream-centered finite-difference schemes for ideal compressible flow. *J. of Comp. Phys.*, 23:262–275, 1977.

[8] S. K. Lele. Compact finite difference scheme with spectral-like resolution. *J. of Comp. Phys.*, 103:16–42, 1992.

[9] K. Mahesh. A family of high-order finite difference schemes with good spectral resolution. *J. of Comp. Phys.*, 145:332–358, 1998.

[10] T. K. Sengupta and al. Analysis of central and upwind compact schemes. *J. of Comp. Phys.*, 192:677–694, 2003.

[11] Capdeville G. A high-order monotonicity-preserving scheme for hyperbolic conservation laws. *Comp. Fluids*, 144:86–116, 2016.

[12] Capdeville G. Towards a compact high-order method for non-linear hyperbolic systems. i: the hermite least square monotone (hlsm) reconstruction. *J. of Comp. Phys.*, 228:3762–3788, 2009.

[13] Capdeville G. Towards a compact high-order method for non-linear hyperbolic systems. ii: the hermite hllc scheme. *J. of Comp. Phys.*, 227:9428–9462, 2008.

[14] S. Gottlieb. On high-order strong stability preserving runge-kutta and multi step time discretizations. *J. of Sci. Comp.*, 25:105–128, 2005.

[15] B. Einfeldt and al. On godunov-type methods near low densities. *J. of Comp. Phys.*, 92:273–295, 1991.

[16] S. Spekreijse. Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws. *Math. of Comp.*, 49:135–155, 1987.