A stable Spectral Difference approach for computations with triangular and hybrid grids up to the 6th order of accuracy

Adèle Veilleux, Guillaume Puigt, Hugues Deniau, Guillaume Daviller

A stable Spectral Difference approach for computations with triangular and hybrid grids up to the 6^{th} order of accuracy

Adèle Veilleux[a,b,1,*], Guillaume Puigt[a,2], Hugues Deniau[a,2], Guillaume Daviller[b,2]

[a] *ONERA/DMPE, Université de Toulouse, F-31055 Toulouse, France*
[b] *Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS), 42 avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France*

Abstract

In the present paper, a stable Spectral Difference formulation on triangles is defined using a flux polynomial expressed in the Raviart-Thomas basis up to the sixth-order of accuracy. Compared to the literature on the Spectral Difference approach, the present work increases the order of accuracy that the stable formulation can deal with. The proposed scheme is based on a set of flux points defined in the paper. The sets of points leading to a stable formulation are determined using a Fourier stability analysis of the linear advection equation coupled with an optimization process. The proposed Spectral Difference formulation differs from the Flux Reconstruction method on hybrid grids: the distinction between the two approaches is highlighted through the definition of the number of interior flux points. Validation starts from a convergence study using Euler equations and continues with simulations of laminar viscous flows over the NACA0012 airfoil using quadratic triangles and around a cylinder using a hybrid grid.

Keywords: high-order method, Spectral Difference method, Raviart-Thomas space, triangle, hybrid, linear stability analysis.

1. Introduction

Many advancements in high-order discontinuous methods enable accurate and robust simulations on unstructured grids with a good parallel efficiency. Numerical schemes using piecewise continuous polynomials are widely used to obtain high-order accuracy. The aim is to look for a polynomial solution in each mesh cell, but without requiring the solution to be continuous across mesh interfaces. The most popular approach, the Discontinuous Galerkin (DG) method, was successfully implemented in many solvers and leads to very rich research. Without being exhaustive, a partial literature review focused on Computational Fluid Dynamics is available in several books [1–8] and many contributions in Europe also come from projects [9–11] involving research centers and industry [12]. The DG method links the standard Finite Element method and the Finite Volume method: unknowns defined on a polynomial basis are solution of a weak problem as in Finite Element but discontinuities at mesh interfaces are solved using an approximated Riemann solver as in Finite Volume. While DG methods are based on the integral form of equations, other methods directly use the strong form, which results in a simpler formulation and implementation as well as a lower computational cost since no integral needs to be computed [13]. For a standard hyperbolic equation, the solution is sought under the form of a polynomial of degree p defined in any mesh cell. For consistency, it is mandatory to define the flux density divergence as a polynomial of degree p since dealing with the strong formulation

*Corresponding author
Email addresses: Adele.Veilleux@onera.fr (Adèle Veilleux), Guillaume.Puigt@onera.fr (Guillaume Puigt), Hugues.Deniau@onera.fr (Hugues Deniau), Guillaume.Daviller@cerfacs.fr (Guillaume Daviller)
[1] Ph.D. Student
[2] Research engineer

means that the divergence of the flux polynomial is explicitly computed. Today, there are essentially two classes of methods based on the strong formulation.

The first class is called the Correction Procedure for Reconstruction (CPR) or the Flux Reconstruction (FR) approach. Introduced by Huynh in 2007 [14], the method consists in defining a polynomial of degree p for the flux, as it is done for the solution. This flux polynomial loses two mandatory properties: the flux divergence is no longer a polynomial of degree p and since the flux is discontinuous at cell interfaces, the scheme is not conservative. In a second step, a lifting operator defined as a polynomial of degree $p+1$ is introduced to recover these lost properties. The lifting operator plays a central role in the properties of the schemes and enables to link the FR method, the DG formulation and other methods [15–17]. A class of lifting operator can be built especially for specific mathematical properties, such as energy stability [18, 19]. Huynh, Wang and Vincent published in 2014 a reference paper on the proposed techniques [20].

An alternative method named the staggered-grid Chebyshev multidomain method was initiated by Kopriva and Kolias [21] in 1996 and applied to structured quadrilateral grids using a tensor-product framework by Kopriva in [22]. In 2006, Liu *et al.* [23] proposed an extension of Kopriva and Kolias' work to simplex cells and called the approach the Spectral Difference (SD) method. Wang *et al.* [24] adapted the procedure to Euler equations on triangular grids. The method was then extended to Navier-Stokes equations by May and Jameson [25] for triangular meshes and Sun *et al.* [26] for hexahedral grids. It is important to notice that for grids based on tensor product cells, the SD method formulation is identical to the multidomain spectral method introduced in [21]. For tensor product cells, the SD method principle consists in defining two polynomials, one for the solution and one for the flux, leading to an order of accuracy of $p+1$, where p is the solution polynomial degree. This choice of polynomial degrees also ensures the consistency of the formulation. However, contrary to the FR approach, the lifting operator is not introduced in the formulation: two sets of points, the Solution Points (SP) and the Flux Points (FP) enable the definition of the Lagrange interpolation polynomials. An alternative approach was derived very recently by Chen *et al.* [27] for tensor-product cells. This technique and the standard one differ in the definition of the flux derivative. In the new formulation, the flux derivative is built from the set of SP plus the interface FP. Such a formulation avoids the need to interpolate from SP to internal FP. Here, attention is focused on the standard SD formulation and details are provided in Sec. 2.

The SD, FR and standard DG methods were compared by Liang *et al.* in [28]. It was proven that the most efficient method is the FR discretization technique and the slowest one is the DG method. The DG method leads to the more accurate results and the FR to the less accurate ones. For both performance and accuracy, the SD method lies in between. Recently, Cox *et al.* [29] compared the accuracy, stability and performance of the standard SD method compare to the FR approach. Nonlinear stability analysis and numerical experiments show that the SD scheme leads to better accuracy and stability. Finally, the quadrature-free DG scheme and the SD method were proven as equivalent under given conditions (use of a nodal Lagrange basis, the quadrature-free paradigm and the numerical flux) for nonlinear hyperbolic conservation laws by May [30].

The stability of the SD method for tensor product cells was studied by Van den Abeele *et al.* [31, 32]. They showed that the SP position did not influence neither the stability nor the accuracy of the scheme. Jameson confirmed this statement [33] and also showed that for the one-dimensional linear advection case, the SD method is stable for all order of accuracy in a norm of Sobolev-type provided that the interior flux collocation points are placed at the zeros of the corresponding Legendre polynomials.

When considering the standard SD method on simplex cells, stability analysis leads to different conclusions. Van den Abeele *et al.* [32] showed that for an order of accuracy strictly greater than 2, the scheme stability is not ensured for triangular cells. For high-order SD schemes on triangular cells, several FP positions are tested but none of them lead to a stable scheme. This explains why after several papers using the SD approach on triangles (see [23, 24, 34–36] among the possible literature), most researchers focused on unstructured grids composed of hexahedra only. To overcome this limitation, Liang *et al.* [37] proposed to decompose any triangle or quadrilateral into quadrilaterals using cell center and mid-edges, leading to cells of half the size of the one of the original element. Using this option, a 2D hybrid mesh is transformed into an unstructured grid composed of quadrilaterals only but the number of mesh elements is strongly increased.

Balan *et al.* proposed another alternative in [38, 39]. Instead of splitting any mesh cell into sub-cells

to define the computational grid, they build an alternative SD formulation using Raviart-Thomas (RT) elements on triangles, leading to the naming SDRT. The SDRT scheme is proven to be linearly stable up to the 4^{th} order under a Fourier stability analysis originally initiated by May [40] and validated on Euler test cases. The SDRT method was then extended to simulate 2D viscous flows on unstructured hybrid grids up to the fourth-order by Li *et al.* [41] and used for the simulation of vortex-induced vibrations using a sliding-mesh method on hybrid grids by Qiu *et al.* [42].

Finally, one must also mention the work of Meister *et al.* on the SD method on triangles based on Proriol-Koornwinder-Dubiner (PKD) basis on the triangle for both solution and flux polynomials [43, 44]. Such approach will also differ with the present one by the set of FP: here, the authors choose the set of Lobatto points on the triangle, as proposed by Blyth and Pozrikidis [45]. In the latter case, there are FP at triangle vertices and since a triangle vertex is generally shared by more than two triangles, this choice is questionable to properly define the inputs of the Riemann problem. Such a configuration will never appear if the interface FP are located on edges, thus this constraint will be applied to the proposed formulation.

The standard staggered SD approach was chosen to be implemented in the high-order solver JAGUAR (proJect of an Aerodynamic solver using General Unstructured grids And high-ordeR schemes) [46] because of its accuracy [47] and its efficiency [48] for Large Eddy Simulations. The SD method was recently made compatible with the non-reflecting boundary conditions [49], written specifically to cope with the SD algorithm and then coupled with a Time Domain Impedance Boundary Condition formulation [50, 51]. In this context, the present paper focuses on the extension of the JAGUAR solver to deal with 2D hybrid unstructured grids composed of standard element shapes (quadrilaterals and triangles).

In Sec. 2, the SDRT scheme on triangles and its difference with the standard technique on quadrilaterals are highlighted. The linear stability of the SDRT method based on interior FP located at known quadrature rules points is studied using Fourier analysis in Sec. 3. The optimization procedure to find a linearly stable formulation on triangles is then presented and sets of interior FP leading to stable SDRT schemes up to the sixth order are given in Sec. 4. Validation test cases are presented in Sec. 5, starting from a convergence study using the convection of an isentropic vortex test case to simulations of 2D viscous flows on quadratic triangular and hybrid mesh.

2. Spectral Difference Scheme on 2D hybrid grids

2.1. The SD approach for first order PDE

Let us consider the following 2D scalar conservation law under its differential form:

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0, \quad \text{in } \Omega \times [0, t_f], \tag{1}$$

where u is the state variable, $\mathbf{f} = (f, g)$ is the flux vector where f and g are flux densities in the x and y directions respectively and ∇ is the differential operator in the physical domain $\mathbf{x} = (x, y)$. The computational domain Ω is discretized into N non-overlapping cells (triangles or quadrilaterals) and the i-th element is denoted Ω_i:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i. \tag{2}$$

For implementation simplicity, Eq. (1) is solved in the reference domain. Each cell Ω_i of the domain Ω is transformed into a reference element $\mathcal{T} := \{(\xi, \eta) : 0 \leq \xi, \eta \leq 1, \xi + \eta \leq 1\}$ for a triangle or $\mathcal{Q} := \{(\xi, \eta) : 0 \leq \xi, \eta \leq 1\}$ for a quadrilateral. The transformation can be written as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^{N_p} M_i(\xi, \eta) \begin{pmatrix} x_i \\ y_i \end{pmatrix}, \tag{3}$$

where (x_i, y_i) are the Cartesian coordinates of the N_p vertices of the cells and $M_i(\xi, \eta)$ are the shape functions. The Jacobian matrix of the transformation given by Eq. (3) from the physical (x, y) to the

reference domain (ξ, η) takes the following form:

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}. \tag{4}$$

For a non-singular transformation, the inverse transformation is related to the Jacobian matrix according to:

$$\frac{\partial(\xi,\eta)}{\partial(x,y)} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = J^{-1}. \tag{5}$$

In the reference domain, Eq. (1) becomes:

$$\frac{\partial \hat{u}(\boldsymbol{\xi}, t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0, \tag{6}$$

where $\hat{\nabla}$ is the differential operator in the reference domain, $\boldsymbol{\xi} = (\xi, \eta)$ are the coordinates in the reference domain and $\hat{u}$, $\hat{\mathbf{f}}$ are the solution and the flux in the reference domain defined by:

$$\hat{u} = |J| u, \tag{7}$$

and

$$\hat{\mathbf{f}} = |J| J^{-1} \mathbf{f}. \tag{8}$$

2.2. SD scheme on quadrilaterals

For quadrilaterals, the standard SD method follows a tensorial rule approach and the treatment is performed direction per direction, as in [26, 32, 52]. For a polynomial of degree p leading to an accuracy of $p+1$, a number $N_{SP} = p+1$ of SP (denoted $\xi_j, j \in [\![1, N_{SP}]\!]$) are defined as the Gauss-Chebyshev points in the reference domain $[0, 1]$:

$$\xi_j = \frac{1}{2}\left[1 - \cos\left(\frac{(2j-1)\pi}{2p+2}\right)\right], \text{ for } 1 \leq j \leq p+1. \tag{9}$$

A number N_{FP} of FP (denoted $\xi_k, k \in [\![1, N_{FP}]\!]$) are mandatory to define the flux as a polynomial of degree $p+1$. Two FP are located on the element boundary and the remaining p FP are defined as the roots of the Legendre polynomial of degree p. The number of FP is therefore $N_{FP} = p+2$. Solution and flux polynomials are finally computed using the standard Lagrange polynomials based either on the SP or the FP. Finally, it must be highlighted that the position of the FP on any mesh interface follows the position of the SP in the reference element due to the tensorial formulation.

2.3. SDRT scheme on triangles

On triangles, the SD formulation is based on the Raviart-Thomas (RT) polynomial space, as in [38, 39] To obtain a $(p+1)$-th order accurate scheme, a polynomial of degree p is introduced to approximate the solution. As for the standard SD scheme, the solution at FP is computed by a simple interpolation from the solution polynomial. The flux polynomial is then built from the fluxes computed at FP. The main difference of the SDRT scheme with the standard SD formulation comes from the flux approximation. Instead of projecting the flux vector component-wise into a finite-dimensional polynomial space of degree $p+1$, the flux vector is approximated in the RT space, using vectors as basis functions and scalar flux values as coefficients. By nature, the RT space is the smallest polynomial space such that the divergence maps it onto the space of polynomial of order p (see Appendix A for details). This ensures that the solution and the flux divergence will both be polynomials of degree p. Details on implementation are summarized in the following.

2.3.1. Solution polynomial

The solution $\hat{u}$ is approximated on the reference triangle $\mathcal{T}$ by a polynomial of degree p, $\hat{u}_h(\boldsymbol{\xi}) \in \mathbb{P}_p$, through a set of distinct SP $\boldsymbol{\xi}_j, j \in [\![1, N_{SP}]\!]$ where

$$N_{SP} = \frac{(p+1)(p+2)}{2}, \tag{10}$$

and

$$\mathbb{P}_p = \text{span}\{\xi^i \eta^j, 0 \leq i, 0 \leq j \text{ and } i + j \leq p\}. \tag{11}$$

The polynomial $\hat{u}_h(\boldsymbol{\xi})$ can be expanded using a nodal or a modal representation. When using the nodal representation, the polynomial is represented in term of point values by way of a Lagrangian interpolant, which is defined as the polynomial of lowest degree that assumes at each value $\boldsymbol{\xi}_j$ the corresponding value $\hat{u}_j$ so that the function coincides at each point:

$$\hat{u}_h(\boldsymbol{\xi}) = \sum_{j=1}^{N_{SP}} \hat{u}_j \, l_j(\boldsymbol{\xi}), \tag{12}$$

where l_j is a Lagrange polynomial and $\hat{u}_j$ is the known solution value at point $\boldsymbol{\xi}_j$. Since there is not a closed-form expression of the Lagrange polynomials through an arbitrary set of points on the triangular element [53], a solution is to expand the polynomial $\hat{u}_h$ using a modal representation:

$$\hat{u}_h(\boldsymbol{\xi}) = \sum_{m=1}^{N_{SP}} \bar{u}_m \, \Phi_m(\boldsymbol{\xi}), \tag{13}$$

where $\Phi_m(\boldsymbol{\xi}) \in \mathbb{P}_p$ is a complete polynomial basis and $\bar{u}_m$ are the modal basis coefficients, which do not represent the value of a function at a point. Since $\hat{u}_h(\boldsymbol{\xi})$ and $\Phi_m(\boldsymbol{\xi})$ span the same polynomial space, any projection form will recover the exact expansion coefficient $\bar{u}_m$. By performing a collocation projection at the points $\boldsymbol{\xi}_j$ such that

$$\hat{u}_h(\boldsymbol{\xi}_j) = \hat{u}_j = \sum_{m=1}^{N_{SP}} \bar{u}_m \, \Phi_m(\boldsymbol{\xi}_j), \tag{14}$$

the coefficients $\bar{u}_m$ can then be determined as:

$$\bar{u}_m = \sum_{m=1}^{N_{SP}} \hat{u}_j \, \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1}. \tag{15}$$

The term $\Phi_m(\boldsymbol{\xi}_j)$ corresponds to the matrix of basis change, also known as the generalized Vandermonde matrix $\mathcal{V}_{j,m} = \Phi_m(\boldsymbol{\xi}_j)$. The choice of the basis $\Phi_m(\boldsymbol{\xi})$ is of primary importance since a matrix inversion is involved in the polynomial expansion process. The chosen basis will dictate the conditioning of the matrix $\mathcal{V}$ and thus the computational stability. The most straightforward choice would be the monomial basis $\{1, x, y, x^2, xy, y^2, ..., y^p\}$. However, this choice leads to a dense Vandermonde matrix whose condition number rapidly increases with the order p. A solution is to choose a hierarchical orthogonal basis, whose Vandermonde matrices are diagonal and thus better conditioned. An appropriated basis choice is to define $\Phi_m(\boldsymbol{\xi})$ as the PKD basis, which was defined on the triangle by Proriol [54], Koornwinder [55] and Dubiner [56]. For a polynomial approximation of degree p on the reference triangle, the 2D orthonormal PKD basis takes the following form:

$$\Phi_{i,j}(\xi, \eta) = \sqrt{(i + 1/2)(i + j + 1)} \, P_i^{0,0}(\xi) \left(\frac{1 - \eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i + j \leq p. \tag{16}$$

Details on Jacobi polynomials and the PKD basis normalization can be found in Appendix B. For simplicity, the subscript (i, j) can be replaced by the single index m, $m \in [\![1, N_{SP}]\!]$ with any arbitrary bijection $m \equiv m(i, j)$. From the literature [53, 57], three main assets of the PKD basis can be noted. First, it is

based on Jacobi polynomials, which can be evaluated to a high degree using simple recurrence relations. The PKD L^2 orthogonality will then tend to a well-conditioned Vandermonde matrix. Finally, the PKD basis hierarchical nature (the expansion set of order p contains the expansion set of order $p-1$) simplifies the construction of certain finite element spaces, such as the RT space, which will be used to approximate the flux function in the SDRT formulation. The polynomial approximation $\hat{u}_h$ of the solution $\hat{u}$ is thus defined in the reference space by:

$$\hat{u}_h(\boldsymbol{\xi}) = \sum_{m=1}^{N_{SP}} \hat{u}_j \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \Phi_m(\boldsymbol{\xi}). \tag{17}$$

2.3.2. Solution computation at flux points

To compute the flux values at FP, the solution values at those points need to be determined. With the polynomial distribution given by Eq. (17), the solution at the FP (denoted $\boldsymbol{\xi}_k$) can be computed as:

$$\hat{u}_h(\boldsymbol{\xi}_k) = \sum_{m=1}^{N_{SP}} \hat{u}_j \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \Phi_m(\boldsymbol{\xi}_k) = \sum_{m=1}^{N_{SP}} \hat{u}_j \left(\mathcal{V}_{j,m}\right)^{-1} \Phi_m(\boldsymbol{\xi}_k). \tag{18}$$

Numerically, the extrapolation step is represented by the transfer matrix $\mathbf{T}_{kj} = \left[\left(\mathcal{V}_{j,m}\right)^{-1} \Phi_m(\boldsymbol{\xi}_k)\right]$.

2.3.3. Definition of the flux polynomial from the set of fluxes at flux points

Now that solution values at FP are known, the flux values $\hat{f}_k$ at the k-th flux point are assumed to be computed. The details will be given below. The flux function in the reference domain is approximated by $\hat{\mathbf{f}}_h$ in the RT space as:

$$\hat{\mathbf{f}}_h(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}} \hat{f}_k \boldsymbol{\psi}_k(\boldsymbol{\xi}), \tag{19}$$

where N_{FP} is the number of degrees of freedom needed to represent a vector-valued function in the RT_p space:

$$N_{FP} = (p+1)(p+3), \tag{20}$$

and $\boldsymbol{\psi}_k$ are interpolation functions which form a basis in the RT space with the property:

$$\boldsymbol{\psi}_j(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k = \delta_{jk}, \tag{21}$$

where δ is the Kronecker symbol and $\hat{\mathbf{n}}_k$ are the unit normal vectors defined at FP. At this level, it must be highlighted that some flux points will be located inside the triangle and the definition of the normal vector needs to be described accurately. For interior FP, one physical point is associated with two degrees of freedom through the definition of unit vectors in different directions. In 2D, the unit vectors for interior FPs are $\hat{\mathbf{n}} = (1,0)^\top$ and $\hat{\mathbf{n}} = (0,1)^\top$ in the reference element.

The last step is to determine the scalar flux values $\hat{f}_k$ at FP on which the polynomial approximation given by Eq. (19) relies on. In the case of a first-order partial differential equation, as given by Eq. (6), the flux is only function of the solution. For interior FP, the flux values in the reference domain are computed directly from the approximated solution value and projected on the unit normal vector previously defined. For FP located on edges, $\hat{f}_k$ is computed using a standard numerical flux function given as a solution of a Riemann problem using two extrapolated quantities, one on each side of the interface.

$$\hat{f}_k = \begin{cases} \hat{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k = |J| J^{-1} \mathbf{f}_k(u_h(\boldsymbol{\xi}_k)) \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T}, \\ \left(\hat{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k\right)^* = \left(\mathbf{f}_k \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k\right)^*, & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \tag{22}$$

where $\left(\hat{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k\right)^*$ is the standard numerical flux in the reference element and $u_h(\boldsymbol{\xi}_k) = \frac{1}{|J|} \hat{u}_h(\boldsymbol{\xi}_k)$ is the approximated solution in the physical domain.

2.3.4. Differentiation of the flux polynomial in the set of solution points

Once the flux vector is approximated on the reference element by Eq. (19), it can be differentiated at SP:

$$\begin{aligned}\hat{\nabla}\cdot\hat{\mathbf{f}}(u) &= \left(\hat{\nabla}\cdot\hat{\mathbf{f}}_h\right)(\boldsymbol{\xi}_j)\\ &= \hat{f}_k\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j).\end{aligned} \tag{23}$$

The term $\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j)$ in Eq. (23) can be written as a matrix of size $[N_{SP}\times N_{FP}]$ called differentiation matrix defined as $\mathbf{D}_{jk} = \left[\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j)\right]$. To properly define the differentiation matrix, the vector-valued interpolation basis functions $\boldsymbol{\psi}_k$ and their derivatives need to be determined. To do so, the first step is to express the known monomial basis in the RT space $\boldsymbol{\phi}_n,\ n\in[\![1,N_{FP}]\!]$ as a linear combination of the basis functions $\boldsymbol{\psi}_k$:

$$\boldsymbol{\phi}_n(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}} a_{n,k}\boldsymbol{\psi}_k(\boldsymbol{\xi}). \tag{24}$$

To determine the coefficients $a_{n,k}$, Eq. (24) is multiplied by $\hat{\mathbf{n}}_k$ and then by enforcing the condition given by Eq. (21), one gets:

$$\boldsymbol{\phi}_n(\boldsymbol{\xi})\cdot\hat{\mathbf{n}}_k = a_{n,k}\boldsymbol{\psi}_k(\boldsymbol{\xi})\cdot\hat{\mathbf{n}}_k, \tag{25}$$

and

$$\boldsymbol{\phi}_n(\boldsymbol{\xi}_k)\cdot\hat{\mathbf{n}}_k = \sum_{l=1}^{N_{FP}} a_{n,k}\boldsymbol{\psi}_l(\boldsymbol{\xi}_k)\cdot\hat{\mathbf{n}}_k, \tag{26}$$

so

$$a_{n,k} = \boldsymbol{\phi}_n(\boldsymbol{\xi}_k)\cdot\hat{\mathbf{n}}_k. \tag{27}$$

Using Eq. (24), the derivative can be expressed as:

$$\hat{\nabla}\cdot\boldsymbol{\phi}_n(\boldsymbol{\xi}) = \sum_{k=1}^{N_{FP}} a_{n,k}\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}), \tag{28}$$

and therefore

$$\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}) = (a_{n,k})^{-1}\,\hat{\nabla}\cdot\boldsymbol{\phi}_n(\boldsymbol{\xi}). \tag{29}$$

The final form of the SDRT scheme can be written for each degree of freedom of the solution function in each cell i as:

$$\frac{d\hat{u}_j^{(i)}}{dt} + \sum_{k=1}^{N_{FP}} \hat{f}_k^{(i)}\left(\hat{\nabla}\cdot\boldsymbol{\psi}_k\right)(\boldsymbol{\xi}_j) = 0,\quad j\in[\![1,N_{SP}]\!],\quad i\in[\![1,N]\!]. \tag{30}$$

and the solution can be time-integrated using any standard time integration scheme (Runge-Kutta scheme for instance).

2.4. A first comment on the position of FP

Due to the strong desire to perform computations on hybrid grids composed of quadrilaterals and triangles, the position of the FP on triangles must follow the rule for quadrilaterals: there will be $(p+1)$ FP per face so $(3p+3)$ FP are located and the remaining $(p+1)\times(p+3)-3(p+1) = p(p+1)$ FP must be located in the element. In addition, the product $p(p+1)$ is always even, which allows to define $p(p+1)/2$ physical interior FP points associated with two degrees of freedom through the definition of different normal vectors.

2.5. Comparison of SDRT and FR schemes

The FR/CPR technique was introduced as a way to recover SD, DG and other schemes for any linear hyperbolic equation. However, an open question concerns the possible differences between the proposed technique and the FR/CPR scheme. Let us consider the FR/CPR method described by Castonguay and Williams in their respective Ph.D. thesis [58, 59]:

- FR/CPR method: The flux polynomial definition involves $\frac{(p+1)(p+2)}{2}$ SP (located inside the element) and $(p+1)$ FP located on each edge.
- SDRT method: The flux polynomial relies on $(p+1)(p+3)$ FP, including $(p+1)$ FP on each edge. The number of FP located inside the element is thus $p(p+1)$.

SDRT and FR/CPR methods will differ if:

$$\frac{(p+1)(p+2)}{2} \neq p(p+1) \Longrightarrow p \neq 2 \text{ and } p \neq -1. \tag{31}$$

Remark: The present analysis to build a link between SDRT and FR flux polynomial computation is valid for any hyperbolic equation. For the linear advection equation, the authors think that a connection should be established due to the linear relation between the solution and the flux, as in [33]. The definition of this link is out of the scope of the current paper.

2.6. Extension of the SD approach for Navier-Stokes equations

Let us consider the same 2D scalar conservation law in the reference domain:

$$\frac{\partial \hat{u}(\boldsymbol{\xi}, t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0, \tag{32}$$

except now, the flux is defined by :

$$\hat{\mathbf{f}} = |J| J^{-1} \mathbf{f}(u, \nabla u), \tag{33}$$

leading to a second-order PDE. For the Navier-Stokes equations, the flux can be expressed as:

$$\mathbf{f} = \mathbf{f}^i(u) - \mathbf{f}^v(u, \nabla u), \tag{34}$$

where $\mathbf{f}^i$ is the inviscid flux and $\mathbf{f}^v$ is the viscous flux. The viscous flux depends not only on the solution u but also on its first spatial derivative ∇u. Eq. (32) is solved following the very same procedure as for a first-order PDE except for the determination of the flux values at FP $\hat{f}_k$. The scalar flux values are now given by:

$$\hat{f}_k = \hat{f}_k^i - \hat{f}_k^v. \tag{35}$$

The inviscid flux values $\hat{f}_k^i$ are computed using Eq. (22) since the inviscid flux only depends on the solution:

$$\hat{f}_k^i = \begin{cases} \hat{\mathbf{f}}^i \cdot \hat{\mathbf{n}}_k = |J| J^{-1} \mathbf{f}_k^i(u_h(\boldsymbol{\xi}_k)) \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T}, \\ \left(\hat{\mathbf{f}}_k^i \cdot \hat{\mathbf{n}}_k\right)^* = \left(\mathbf{f}_k^i \cdot |J| (J^{-1})^\top \hat{\mathbf{n}}_k\right)^*, & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \tag{36}$$

To compute $\hat{f}_k^v$, which relies on the solution and its gradient, the following procedure, based on a centered formulation [26] is used. From the approximated solution in the reference domain, the physical approximated solution $u_h(\boldsymbol{\xi}_k)$ is first computed at FP:

$$u_h(\boldsymbol{\xi}_k) = \frac{1}{|J|} \hat{u}_h(\boldsymbol{\xi}_k) = \frac{1}{|J|} \mathbf{T}_{kj}\, \hat{u}_j. \tag{37}$$

From those values, a polynomial interpolation of degree $p+1$ should be reconstructed for the solution but this polynomial would be discontinuous at cell interfaces. For this reason, a centered scheme is used to

uniquely define the solution at each flux point by averaging the values from the left and the right cells, leading to a continuous polynomial interpolation u_h^c:

$$u_h^c(\boldsymbol{\xi}_k) = \begin{cases} u_h(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T}, \\ \frac{1}{2}\left(u_h^L(\boldsymbol{\xi}_k) + u_h^R(\boldsymbol{\xi}_k)\right), & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \tag{38}$$

The solution gradient in the reference domain is given as [41]:

$$\left(\hat{\nabla}\hat{u}\right)\left(\boldsymbol{\xi}_j\right) = \mathbf{D}_{jk}\,\hat{\mathbf{n}}_k\,\hat{u}_h^c(\boldsymbol{\xi}_k). \tag{39}$$

In the physical domain, the solution gradient can be expressed as:

$$\nabla u = \frac{1}{|J|}\left(J^{-1}\right)^{\top}\hat{\nabla}\hat{u}, \tag{40}$$

and since

$$\hat{u}_h^c = |J| u_h^c, \tag{41}$$

one gets the expression of the solution gradient (∇u) in the physical domain:

$$(\nabla u)\left(\boldsymbol{\xi}_j\right) = \frac{1}{|J|}\,\mathbf{D}_{jk}\,\left(u_h^c(\boldsymbol{\xi}_k)\left(|J|J^{-1}\right)^{\top}\hat{\mathbf{n}}_k\right). \tag{42}$$

From the solution gradient at SP in the reference domain, the solution gradient in the physical domain can be interpolated at FP:

$$(\nabla u)_h(\boldsymbol{\xi}_k) = \mathbf{T}_{kj}\;(\nabla u)\left(\boldsymbol{\xi}_j\right). \tag{43}$$

The polynomial approximation of the solution gradient $(\nabla u)_h$ is discontinuous at cell interfaces. As it was done for the solution, a centered scheme is used to defined a single value at cell interface:

$$(\nabla u)_h^c(\boldsymbol{\xi}_k) = \begin{cases} (\nabla u)_h(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T}, \\ \frac{1}{2}\left((\nabla u)_h^L(\boldsymbol{\xi}_k) + (\nabla u)_h^R(\boldsymbol{\xi}_k)\right), & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \tag{44}$$

The continuous solution u_h^c and the continuous solution gradient $(\nabla u)_h^c$ in the physical domain are used to compute the viscous flux values:

$$\mathbf{f}_k^v = \mathbf{f}^v(u_h^c(\boldsymbol{\xi}_k), (\nabla u)_h^c(\boldsymbol{\xi}_k)). \tag{45}$$

The viscous flux values in the reference domain are finally given as:

$$\hat{f}_k^v = \begin{cases} |J|J^{-1}\mathbf{f}_k^v \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T}, \\ \mathbf{f}_k^v \cdot |J|(J^{-1})^{\top}\hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \tag{46}$$

The flux polynomial based on the flux values $\hat{f}_k = \hat{f}_k^i - \hat{f}_k^v$ is then differentiated by multiplying it by the differentiation matrix $\mathbf{D}_{jk}$ and the semi-discrete equation is integrated in time.

3. Linear Stability Analysis of the SDRT Formulation

3.1. Importance of the Flux Points Location on the Stability

As presented in Sec. 2, for tensor product cells, the polynomial basis is the Lagrangian basis for both extrapolation and differentiation whereas for simplex cells, the PKD basis and the Raviart-Thomas basis are used for the extrapolation and the differentiation (respectively). Those polynomial bases rely on the SP and FP sets of points and the normal vector associated with each FP. Since it was shown by Van den Abeele

et al. [32] that the SD scheme stability is independent of the SP position, the main concern is to find a set of FP leading to a stable SDRT scheme for all advection angles. The FP location has a direct impact on the SD scheme stability.

In 1D, it was shown by Van den Abeele [32] that if FP are chosen as the Chebyshev-Gauss-Lobatto nodes, the standard 1D SD scheme can be unstable for $p > 2$. Following this work, Jameson [33] has proven that the stability of the SD scheme for all orders of accuracy in the case of a 1D linear advection *'provided that the interior fluxes collocation points are placed at the zeros of the corresponding Legendre polynomial'*.

For triangular elements, it was observed by Balan *et al.* [38] that the placement of FP on edges does not affect the linear stability properties for second- to fourth-order accurate SDRT schemes. To simplify the 2D hybrid implementation, the position of FP located on the edge is set to the Gauss-Chebyshev points given by Eq. (9). FP on edges for a quadrilateral and a triangle are thus located at the same coordinates. By doing so, there is no need to apply mortar techniques (as introduced by Kopriva [22]) for an interface between a quadrilateral and a triangle. Since the edge FP position is chosen to be fixed, the remaining unknown is the interior FP location. For a SDRT scheme, the number of interior FP is given by:

$$N_i = p\,(p+1). \tag{47}$$

The number of physical interior points is reduced from N_i to $N_i/2$ by considering one physical point as two separate DoF with different 'normal' vectors $\hat{\mathbf{n}}$. As mentioned in section 2.3.3, each physical interior flux point is associated with two normal vectors $(1,0)^\top, (0,1)^\top$. The number of physical interior FP, denoted N_{pi}, is thus given as:

$$N_{pi} = \frac{p\,(p+1)}{2} \tag{48}$$

which correspond to the number of SP for a $\mathrm{SDRT}_{(p-1)}$ scheme.

3.2. Flux Points Numbering

For clarity purposes, the FP numbering in the reference triangle needs to be settled and their normal vector defined. On each edge, there are $N_e = (p+1)$ FP which are represented with red circles and numbered as follow:

- on face 1 ($\eta = 0$), $k \in [\![1, N_e]\!]$, k increasing with ξ, $\hat{\mathbf{n}} = (0,-1)^\top$,
- on face 2 ($\eta = 1-\xi$), $k \in [\![N_e+1, 2N_e]\!]$, k increasing with η, $\hat{\mathbf{n}} = (1,1)^\top$,
- on face 3 ($\xi = 0$), $k \in [\![2N_e+1, 3N_e]\!]$, k increasing when η decreases, $\hat{\mathbf{n}} = (-1,0)^\top$.

The remaining $N_i = p\,(p+1)$ FP are located in the interior and represented with blue squares. Since one physical point is considered as two separate DoF with different normal vectors, there are $N_i/2$ physical FP. FP associated with the unit vector $\hat{\mathbf{n}} = (1,0)^\top$ in the reference element are numbered with $k \in [\![3N_e+1, 3N_e+1+N_i/2]\!]$ whereas FP whose unit vector is $\hat{\mathbf{n}} = (0,1)^\top$ are numbered with $k \in [\![3N_e+1+N_i/2, 3N_e+1+N_i]\!]$. An example of the FP numbering and their associated normal vector is given in Fig. 1 for the case $p = 2$.

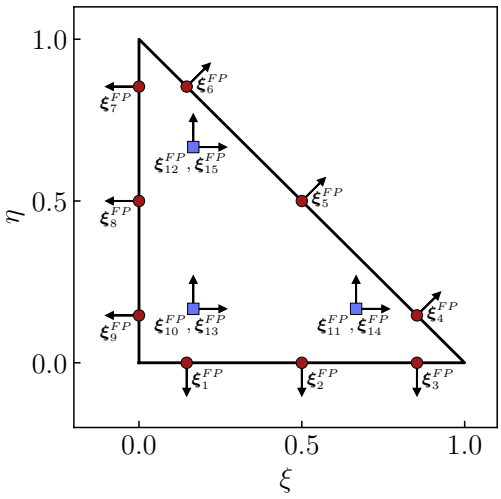


Figure 1: FP numbering in the reference triangular element - Example of FP distribution for $p = 2$ (edge: ●, interior: ■)

3.3. Fourier Stability Analysis

To analyze the linear stability of the SDRT scheme on triangles, the Fourier stability analysis is used. An other method was proposed in [60] and consists in studying the linear stability on a domain composed of a Fixed-Number of Cells (FNC approach) without assuming the numerical solution under the form of a planar harmonic wave; however, it was then shown that the FNC approach is dependent on the properties of the number of cells considered and does not allow to draw conclusion in a general framework, i.e. for larger meshes. In this section, the Fourier analysis presented by Castonguay in his Ph.D. thesis [59] for the FR method is thus adapted to the SDRT scheme and results are presented for $p \in [\![4, 6]\!]$.
Let us consider the linear advection equation:

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \nabla \cdot \mathbf{f} = 0, \quad \text{in } \Omega \times [0, t_f] \tag{49}$$

within a domain Ω, where u is a conserved scalar quantity and $\mathbf{f} = \mathbf{c} \cdot u$ is the flux. The velocity field $\mathbf{c}$ is defined by:

$$\mathbf{c} = (c_x, c_y) = (\cos\theta, \sin\theta), \quad \theta \in [0, 2\pi] \tag{50}$$

Eq. (49) is solved on a square domain $\Omega = [0, L]^2$ with periodic boundary conditions. The domain Ω is meshed as a Cartesian mesh composed of $N_x \times N_y$ quadrilateral elements of size $\Delta x \times \Delta y$. The mesh is distorted using the skew angle μ. Each quadrilateral cell is then divided into two triangles, identified as $T_{i_1,i_2,1}$ and $T_{i_1,i_2,2}$, $i_1 \in [\![1, N_x]\!]$, $i_2 \in [\![1, N_y]\!]$ (Fig. 2). To properly define the mesh pattern, two vectors are introduced: $\mathbf{B}_1 = (\Delta x, 0)$ and $\mathbf{B}_2 = \Delta x(\cos\mu, \sin\mu)$. The mesh is made dimensionless using a scaling by the Cartesian mesh edge length Δx, leading to the dimensionless vectors $\hat{\mathbf{B}}_1 = (1, 0)$ and $\hat{\mathbf{B}}_2 = (\cos\mu, \sin\mu)$.

Defining $\hat{\mathbf{U}}_j^{i_1,i_2} = [\hat{\mathbf{U}}_j^{i_1,i_2,1}, \hat{\mathbf{U}}_j^{i_1,i_2,2}]^\top$ as the vector collecting the solution in the reference domain on the two triangles $T_{i_1,i_2,1}$ and $T_{i_1,i_2,2}$ for each SP $j \in [\![1, N_{SP}]\!]$, the SDRT spatial discretization using an upwind flux on this mesh takes the form:

$$\frac{d\hat{\mathbf{U}}_j^{i_1,i_2}}{dt} = -\frac{\|\mathbf{c}\|}{\Delta x}\Big[\mathbf{M}^{0,0}\hat{\mathbf{U}}_j^{i_1,i_2} + \mathbf{M}^{-1,0}\hat{\mathbf{U}}_j^{i_1-1,i_2} + \mathbf{M}^{+1,0}\hat{\mathbf{U}}_j^{i_1+1,i_2} + \mathbf{M}^{0,-1}\hat{\mathbf{U}}_j^{i_1,i_2-1} + \mathbf{M}^{0,+1}\hat{\mathbf{U}}_j^{i_1,i_2+1}\Big]. \tag{51}$$

In Eq. (51), $\mathbf{M}^{0,0}$, $\mathbf{M}^{-1,0}$, $\mathbf{M}^{+1,0}$, $\mathbf{M}^{0,-1}$ and $\mathbf{M}^{0,+1}$ are matrices of size $[2N_{SP}, 2N_{SP}]$ containing the three steps of the spatial discretization (extrapolation, flux computation and differentiation), which depend on the advection angle θ, the grid angle μ as well as on the SP and FP locations. The exact formulation of

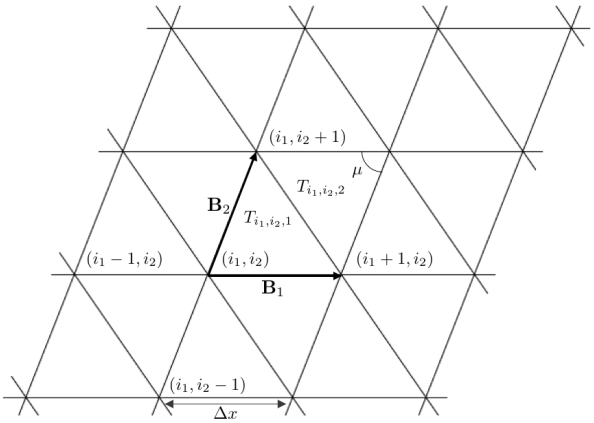


Figure 2: Mesh generating pattern used for the 2D Fourier stability analysis on triangles

those matrices is given in Appendix C. The discrete numerical solution is now assumed under the form of a planar harmonic wave:

$$\hat{\mathbf{U}}^{i_1,i_2} = \tilde{\mathbf{U}} \exp\Big(I\mathbf{k}\,(i_1\mathbf{B}_1 + i_2\mathbf{B}_2)\Big), \tag{52}$$

where $\tilde{\mathbf{U}}$ is a complex vector of dimension $2N_{SP}$, independent of i_1 and i_2, and $\mathbf{k} = k(\cos\vartheta, \sin\vartheta)^\top$, k being the wavenumber of the harmonic wave and ϑ its orientation angle.
Using the non-dimensional quantities previously introduced, the discrete numerical solution is:

$$\hat{\mathbf{U}}^{i_1,i_2} = \tilde{\mathbf{U}} \exp\Big(I\kappa\big((i_1 + i_2\cos\mu)\cos\vartheta + i_2\sin\mu\sin\vartheta\big)\Big), \tag{53}$$

$\kappa = k\Delta x$ being the grid frequency. Injecting Eq. (53) into Eq. (51), one gets:

$$\begin{aligned}\frac{d\tilde{\mathbf{U}}}{dt} = -\frac{\|\mathbf{c}\|}{\Delta x}\Big[&\mathbf{M}^{0,0} + \mathbf{M}^{-1,0}\exp\Big(-I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{+1,0}\exp\Big(I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{0,-1}\exp\Big(-I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\\ &+ \mathbf{M}^{0,+1}\exp\Big(I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\Big]\,\tilde{\mathbf{U}}\\ = \frac{\|\mathbf{c}\|}{\Delta x}&\mathbf{M}_{\mathrm{z}}\tilde{\mathbf{U}}.\end{aligned} \tag{54}$$

The complete spectrum of the SDRT spatial operator can be obtained by computing the eigenvalues of $\mathbf{M}_{\mathrm{z}}$, denoted $\lambda_{\mathbf{M}_{\mathrm{z}}}$. The matrix $\mathbf{M}_{\mathrm{z}}$ depends on:

- the SP location,
- the FP location,
- the advection angle $\theta \in [0, 2\pi]$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu \in [0, \pi/2]$.

Using the eigenvalue analysis, the SDRT spatial discretization is stable under a Fourier stability analysis if the real part of all eigenvalues of the matrix $\mathbf{M}_{\mathrm{z}}$ are non-positive, i.e. if $\mathrm{Re}(\lambda_{\mathbf{M}_{\mathrm{z}}}) \le 0$.

3.4. Interior Flux Points Locations Based on Quadrature Rules

The Fourier analysis was applied to the SDRT scheme for triangular [39, 40] and hybrid grids [41]. The placement of interior FP leading to stable SDRT schemes was given for $p \in [\![1, 3]\!]$ by May and Schöberl [40]. Their conclusions can be summarized as follow:

- For SDRT_1, the interior physical FP is placed at the triangle centroid;
- For SDRT_2, interior physical FP are placed according to the three-points quadrature rule of order 2. This quadrature rule was given by many authors [61–67] as the higher order three-points rules;
- For SDRT_3, interior physical FP are located at the six-point quadrature rule of order 4 given by [62–70].

Efforts were made to determine stable formulations for $p > 3$ but results were not successful. Choosing a quadrature rule associated with the same number of points as the interior physical FP seems to be a promising choice. To be suitable, the quadrature rule should not include integration points located on edges or outside the triangle. Among the available literature, several appropriate quadrature rules are found for $p > 3$:

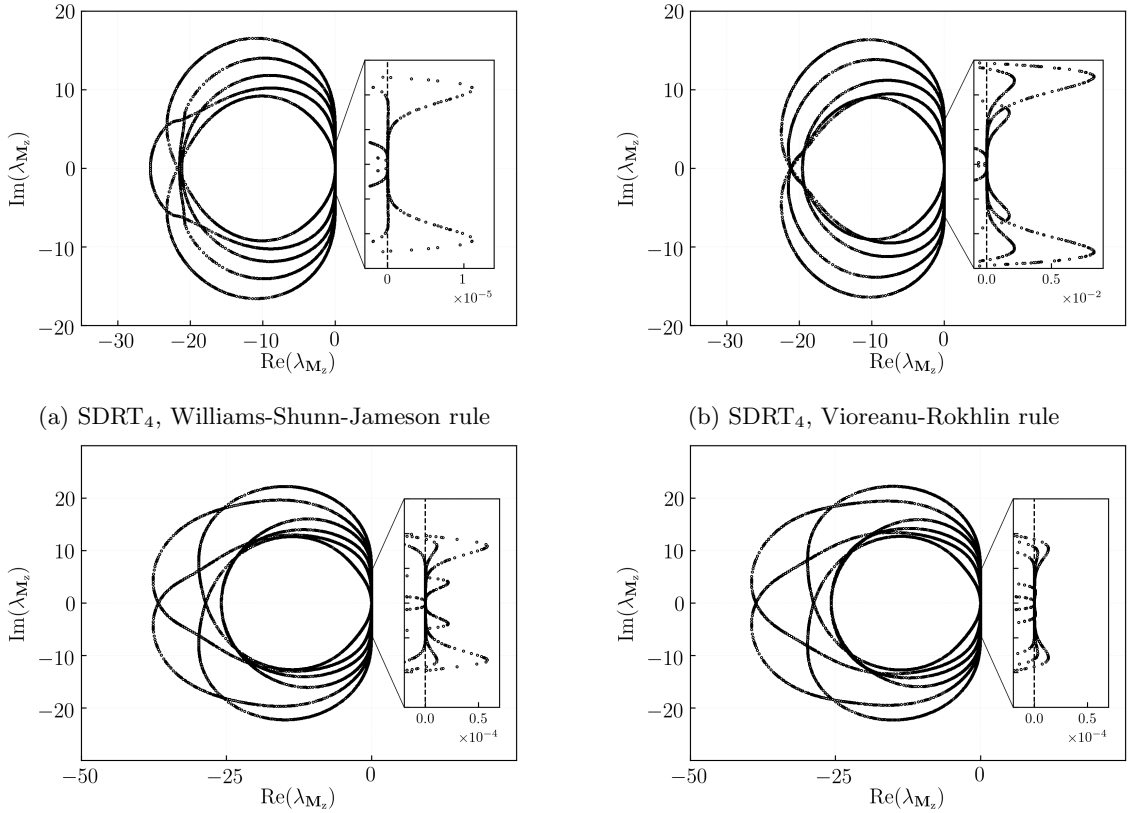
- For SDRT_4, the 10-points quadrature rules of order 5 of Vioreanu-Rokhlin [66] and Williams-Shunn-Jameson [67];
- For SDRT_5, the 15-points quadrature rule of order 7 of Williams-Shunn-Jameson [67], Witherden-Vincent [70], Xiao-Gimbutas [65], Vioreanu-Rokhlin [66], Papanicolopulos [71] and Laursen-Gellert [62].
- For SDRT_6, the 21-points quadrature rule of order 8 of Williams-Shunn-Jameson [67] and Vioreanu-Rokhlin [66] and the 21-points quadrature rule of order 9 of Laursen-Gellert [62]

The spectrum of the spatial SDRT operator is computed for $p \in [\![4, 6]\!]$ using Fourier analysis for different implementations (i.e. different interior FP locations) for $\kappa \in [-\pi, \pi]$, $\vartheta \in [0, 2\pi]$, $\theta = 0$ and $\mu = \pi/2$. The SP location is set to the Williams-Shunn-Jameson quadrature points [67]. Values of $\max(\text{Re}(\lambda_{\mathbf{M}_z}))$ are displayed in Table 1 for each SDRT implementation based on interior FP locations taken as points of quadrature rules presented before. The first observation is that all SDRT implementations show positive values of $\max(\text{Re}(\lambda_{\mathbf{M}_z}))$, indicating that the spatial discretization is unstable. One can then note that only two quadrature rules are appropriated for both $p = 4$, $p = 5$ and $p = 6$: the Williams-Shunn-Jameson and the Vioreanu-Rokhlin. For the three polynomial degrees p, the use of the WSJ quadrature rule as the interior FP leads to smaller values of $\max(\text{Re}(\lambda_{\mathbf{M}_z}))$ compared to the Vioreanu-Rokhlin quadrature rule. For the SDRT_5 scheme, two of the quadrature rules (Laursen-Gellert and Papanicolopulos) lead to very high values of $\max(\text{Re}(\lambda_{\mathbf{M}_z}))$, whereas the smaller value is given by the Witherden-Vincent quadrature rule. For the SDRT_6 scheme, positive values of $\max(\text{Re}(\lambda_{\mathbf{M}_z}))$ are obtained for each quadrature rule.

Quadrature rule	SDRT_4	SDRT_5	SDRT_6
Williams-Shunn-Jameson [67]	$1.11 \cdot 10^{-5}$	$5.85 \cdot 10^{-5}$	$2.87 \cdot 10^{-3}$
Vioreanu-Rokhlin [66]	$8.29 \cdot 10^{-3}$	$1.75 \cdot 10^{-2}$	$2.92 \cdot 10^{-2}$
Laursen-Gellert [62]	-	$> 10^{12}$	$9.82 \cdot 10^{-1}$
Witherden-Vincent [70]	-	$1.31 \cdot 10^{-5}$	-
Xiao-Gimbutas [65]	-	$7.33 \cdot 10^{-2}$	-
Papanicolopulos [71]	-	$> 10^{12}$	-

Table 1: Values of $\max(\text{Re}(\lambda_{\mathbf{M}_z}))$ for $\theta = 0$ using different quadrature rules as the interior FP locations

Spectra of unstable discretizations are plotted in Fig. 3 for SDRT_4 using Williams-Shunn-Jameson (Fig. 3a) and Vioreanu-Rokhlin (Fig. 3b) quadrature rules and for SDRT_5 using Williams-Shunn-Jameson (Fig. 3c) and Witherden-Vincent (Fig. 3d). A closer view on each spectra allows one to clearly see the positive eigenvalues real part of the spatial operator $\mathbf{M}_z$ for $\theta = 0$.



(a) SDRT$_4$, Williams-Shunn-Jameson rule

(b) SDRT$_4$, Vioreanu-Rokhlin rule

(c) SDRT$_5$, Williams-Shunn-Jameson rule

(d) SDRT$_5$, Witherden-Vincent rule

Figure 3: Fourier footprint of the SDRT$_4$ and SDRT$_5$ spatial discretizations on triangles for $\theta = 0$ using different interior FP locations

4. Determination of Stable Formulations through an Optimization Process

4.1. Optimization Algorithm

To determine spatially stable SDRT formulations for orders of accuracy higher than four, the Fourier analysis is used in an optimization problem. The function to minimize is the maximum of the real part of the matrix $\mathbf{M}_z$ eigenvalues and the optimization parameters are the interior FP locations. The optimization process solves the problem of minimizing a function locally using a gradient descent method called the L-BFGS-B method from the SciPy library [72]. The L-BFGS-B algorithm is part of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms, which are iterative methods for solving unconstrained nonlinear optimization problems. The descent direction is determined by preconditioning the gradient with curvature information.

The full algorithm is detailed by Algorithm 1 for the SDRT$_4$ scheme. First, the constant parameters are settled: the polynomial degree p is set to 4, the SP location is set to the position given by the 15-points Williams-Shunn-Jameson quadrature rule and the position of FP located on edges is set to Gauss-Chebyshev points. The interior FP coordinates are then parametrized by a set of coefficients to ensure symmetry. The parametrization is given in Appendix D for $p \in [\![4, 5]\!]$. The initial interior FP location, stored in the initial guess x0, is chosen as the 10-points Williams-Shunn-Jameson quadrature rule, expressed using the optimization parameters. Bounds are given to ensure that all interior FP are included in the triangle. The optimization is then run: the *scipy.optimize.minimize* function based on the L-BFGS-B method called the function MAIN, using x0 as the initial interior FP location and taking bounds into account.

The function MAIN returns the maximum of all eigenvalues real part of the matrix $\mathbf{M}_z$. The interior FP

coordinates are first computed based on the optimization parameters, which allows to compute the transfer matrix $\mathbf{T}$ and the differentiation matrix $\mathbf{D}$. The matrix $\mathbf{M}_z$ is computed for:

- the advection angle $\theta \in [0, \pi], \Delta\theta = \pi/8$,
- the grid frequency $\kappa \in [0, \pi] \Delta\kappa = \pi/32$,
- the harmonic plane orientation $\vartheta \in [0, \pi], \Delta\vartheta = \pi/8$,
- the skew angle $\mu = \pi/2$.

Finally, the maximum of the real part of all the eigenvalues of $\mathbf{M}_z$, denoted rm, is returned and will be minimized by the optimization algorithm.

Algorithm 1 Fourier analysis optimization algorithm on triangles for SDRT_4

Constants

$p \leftarrow 4$

SP location: $\boldsymbol{\xi}^{SP}_{1:15} \leftarrow$ WSJ 15-points rule

Edge FP location: $\boldsymbol{\xi}^{FP}_{1:5} \leftarrow$ Gauss-Chebyshev points ▷ Edge 1

$\boldsymbol{\xi}^{FP}_{6:10} \leftarrow$ Gauss-Chebyshev points ▷ Edge 2

$\boldsymbol{\xi}^{FP}_{11:15} \leftarrow$ Gauss-Chebyshev points ▷ Edge 3

Optimization Parameters

$\alpha_1 = 0.333333333333333, \alpha_2 = 0.055564052669793$

$\beta_1 = 0.365789252254277, \gamma_1 = 0.112639085608754$

$\beta_2 = 0.704466288264107, \gamma_2 = 0.281977603613669$ ▷ WSJ 10-points rule

$\beta_3 = 0.929744459481616, \gamma_3 = 0.169338518004915$

$\beta_4 = 0.944435947330207, \gamma_4 = 0.416653920995311$

x0 = $(\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4)$

Bounds = $(\alpha_1, \alpha_2 \in]0, 0.5[, \beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in]0, 1[)$

Optimization Process

call scipy.optimize.minimize(MAIN, x0, Bounds, method='L-BFGS-B')

function MAIN

$\boldsymbol{\xi}_{16} = (\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$, $\boldsymbol{\xi}_{17} = (\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$

$\boldsymbol{\xi}_{18} = (\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$, $\boldsymbol{\xi}_{19} = (\alpha_2, \alpha_2)$

$\boldsymbol{\xi}_{20} = (\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$, $\boldsymbol{\xi}_{21} = (\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$

$\boldsymbol{\xi}_{22} = (\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$, $\boldsymbol{\xi}_{23} = (\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$

$\boldsymbol{\xi}_{24} = (\alpha_1, \alpha_1)$, $\boldsymbol{\xi}_{25} = (\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$

$\boldsymbol{\xi}^{FP}_{26:35} = \boldsymbol{\xi}^{FP}_{16:25}$

Compute Transfer Matrix $\mathbf{T}$

Compute Differentiation Matrix $\mathbf{D}$

for $\mu = \pi/2$ **do** ▷ Skew angle

for $\vartheta \in [0, \pi], \Delta\vartheta = \pi/8$ **do** ▷ Harmonic plane orientation

for $\theta \in [0, \pi], \Delta\theta = \pi/8$ **do** ▷ Advection angle

Compute $\mathbf{M}^{0,0}, \mathbf{M}^{-1,0}, \mathbf{M}^{+1,0}, \mathbf{M}^{0,-1}, \mathbf{M}^{0,+1}$

for $\kappa \in [0, \pi], \Delta\kappa = \pi/32$ **do** ▷ Grid frequency

Compute $\mathbf{M}_z$ using:

$$\begin{aligned}\mathbf{M}_z = -\Big[&\mathbf{M}^{0,0} + \mathbf{M}^{-1,0}\exp\Big(-I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{+1,0}\exp\Big(I\kappa\cos\vartheta\Big)\\ &+ \mathbf{M}^{0,-1}\exp\Big(-I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\\ &+ \mathbf{M}^{0,+1}\exp\Big(I\kappa(\cos\mu\cos\vartheta + \sin\mu\sin\vartheta)\Big)\Big]\end{aligned}$$

Compute $\max(\mathrm{Re}(\lambda_{\mathbf{M}_z}))$

rm = $\max(\mathrm{rm}, \max(\mathrm{Re}(\lambda_{\mathbf{M}_z})))$

end for

end for

end for

end for

return rm

end function

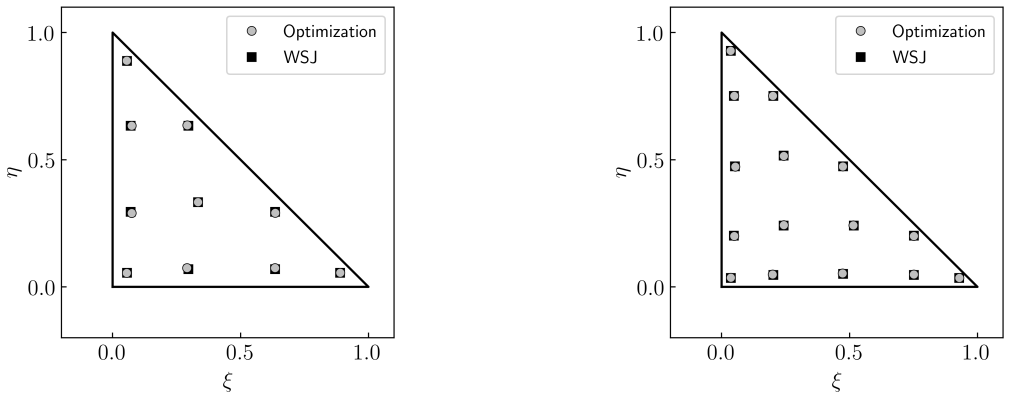
4.2. Spatially Stable $SDRT_4$ and $SDRT_5$ Formulations

4.2.1. Sets of Interior Flux Points

The optimization process based on the L-BFGS-B algorithm was able to determined spatially stable $SDRT_4$ and $SDRT_5$ formulations. Several locations can be found since the algorithm is searching for the minimum and there is no stopping criteria when they go negative. In practice, several locations are found, thus the uniqueness of the stable SDRT scheme is not demonstrated. Due to the local optimization algorithm, the difference between the coordinates of two different sets of points is very small ($\sim 10^{-5}$). Given this small difference, the optimization loop was stopped after approximately ten sets of points leading to a stable SDRT formulation were found and the one associated with the smaller real part of all eigenvalues was chosen. The corresponding parameters are given in Table 2. The interior FP coordinates leading to stable SDRT formulations are actually very close to the coordinates given by the Williams-Shunn-Jameson quadrature rule. This was an expected result due to the local optimization process that looks for a stable formulation close from the initial guess. Both sets of points are compared in Fig. 4. However, as shown in the next section, stability conclusions are quite different.

	$SDRT_4$	
α_1	0.33366214220365053577666003548 1	–
α_2	0.05502032327765691427368111021 7	–
β_1, γ_1	0.36505900941934221748397249029 9	0.10825744697505322589048404324 8
β_2, γ_2	0.70838121841272838619119056602 4	0.28017810320268821122624558483 9
β_3, γ_3	0.92672898300009898253648543686 7	0.17186473732812543313563935498 7
β_4, γ_4	0.94480877497865967118428898174 9	0.41703166521315820913784477852 5
	$SDRT_5$	
α_1	0.03601638717092110059114773434 9	–
α_2	0.24288371116316528897094428884 9	–
α_3	0.47330280861806123260393519558 4	–
β_1, γ_1	0.24865327212126914213641271089 7	0.07537555948630439428548299929 4
β_2, γ_2	0.52610716826649672750448871738 6	0.20953863720661883296436656110 0
β_3, γ_3	0.75746307239073784600691396917 7	0.13620750036029358187583682138 2
β_4, γ_4	0.80019811864053436156751786256 8	0.35127172764319664066690052095 5
β_5, γ_5	0.95099538178119114029129832488 3	0.27556778867665415733156919486 6
β_6, γ_6	0.96387254267775313021360261700 5	0.44671648161944355059915778838 3

Table 2: Coordinate parameters of interior FP determined using the optimization process on Fourier analysis



(a) $SDRT_4$ (b) $SDRT_5$

Figure 4: Sets of FP determined using the optimization process on Fourier analysis compared to Williams-Shunn-Jameson sets

4.2.2. Fourier Analysis of the Spatial Discretization

The Fourier analysis of the spatial SDRT discretization based on the interior FP determined by the optimization process is conducted. The spectrum of $\mathbf{M}_z$ is first computed for:

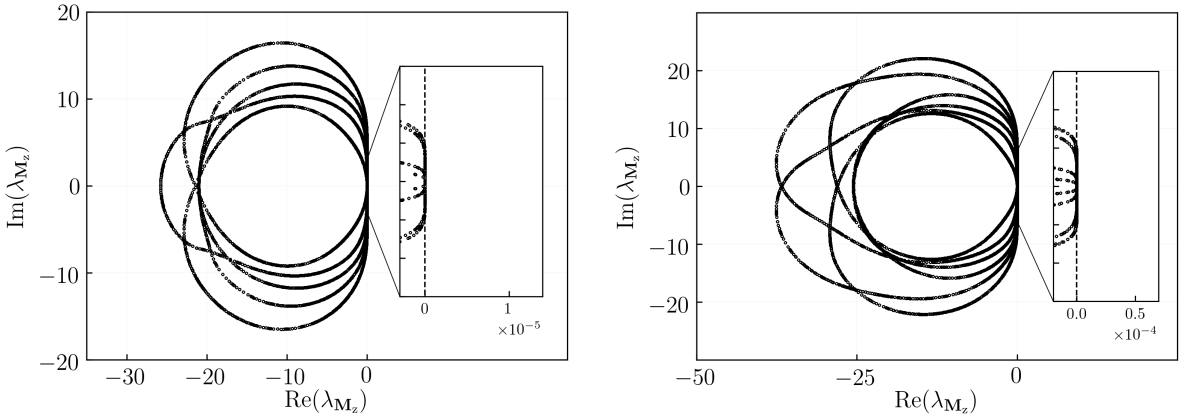
- the advection angle $\theta = 0$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu = \pi/2$.

These conditions are exactly the same as the ones used in Sec. 3.4, where the SDRT discretization was proven as unstable for $p = 4$ and $p = 5$. The spectrum of $\mathbf{M}_z$ using the interior FP determined by the optimization process is displayed in Fig. 5a for $SDRT_4$ and in Fig. 5b for $SDRT_5$. For each order, the Fourier footprint obtained using the SDRT is similar to the one obtained using interior FP located at quadrature rules points (Fig. 3a and Fig. 3c) except that positive eigenvalues have been pushed to the negative side, leading to stable formulations.

Spectra are then computed in the general case, i.e. for:

- the advection angle $\theta \in [0, \pi]$,
- the grid frequency $\kappa \in [-\pi, \pi]$,
- the harmonic plane orientation $\vartheta \in [0, 2\pi]$,
- the skew angle $\mu \in [\pi/2, \pi/3, \pi/4]$.

Corresponding Fourier footprints are shown in Fig. 5c for $SDRT_4$ and in Fig. 5d for $SDRT_5$. From these spectra, the linear stability of the SDRT spatial discretization is clearly established since the real part of all eigenvalues is negative.



(a) SDRT$_4$, $\theta = 0$

(b) SDRT$_5$, $\theta = 0$

(c) SDRT$_4$, General Case

(d) SDRT$_5$, General Case

Figure 5: Fourier footprint of the SDRT$_4$ (left) and SDRT$_5$ (right) spatial discretizations on triangles

4.2.3. Fourier Analysis of the Coupled Time-Space Discretization

To investigate the linear stability of the coupled time-space discretization, the semi-discretized form needs to be integrated in time. Considering a differential equation:

$$\frac{\partial u}{\partial t} = R(u), \tag{55}$$

a general m-stage Runge-Kutta (RK) method for Eq. (55) can be written as in [73]:

$$\begin{cases} u^{(l)} & = \sum_{k=0}^{l-1} \left(\alpha_{lk} u^{(k)} + \Delta t \beta_{lk} R(u^{(k)}) \right), \quad l = 1, ..., m, \\ u^{(0)} & = u^{(n)}, \quad u^{(m)} = u^{(n+1)}, \end{cases} \tag{56}$$

where, in the case of Eq. (49), $R(u^{(k)}) = -\nabla \cdot \mathbf{f}(u^{(k)})$. In this paper, the time-integration scheme used is the SSP3s3o of Gottlieb and Shu, which is part of the family of Total Variation Diminishing (TVD) time discretization [73], later called Strong Stability Preserving (SSP) schemes [74]. The nonlinear stability property of these methods makes them particularly appropriated for the time-integration of hyperbolic partial differential equations.

The semi-discretized matrix form containing the planar harmonic wave given by Eq. (54) integrated in time using Eq. (56) is:

$$\begin{cases} \mathbf{G}^{(l)} & = \left(\sum_{k=0}^{l-1} \left(\alpha_{lk} \mathbf{I} + \nu \beta_{lk} \mathbf{M}_z \right) \right) \mathbf{G}^{(k)}, \quad l = 1, ..., m, \\ \mathbf{G}^{(0)} & = \mathbf{I}, \quad \tilde{\mathbf{U}}^{(n+1)} = \mathbf{G}\, \tilde{\mathbf{U}}^{(n)}, \end{cases} \tag{57}$$

where $\mathbf{G} = \mathbf{G}^{(m)}$ and ν is the CFL number defined by:

$$\nu = \frac{\|\mathbf{c}\| \Delta t}{\Delta x}. \tag{58}$$

The stability condition on the coupled time-space discretization is thus obtained by requiring that the amplitude of any harmonic does not grow in time, i.e.:

$$|\mathbf{G}| = \left| \frac{\tilde{\mathbf{U}}^{(n+1)}}{\tilde{\mathbf{U}}^{(n)}} \right| \leq 1. \tag{59}$$

In other words, to ensure a stable discretization, the spectral radius of the matrix $\mathbf{G}$, denoted $\rho_{\mathbf{G}}$ should be lower than 1, meaning that all the eigenvalues $\lambda_{\mathbf{G}}$ should be in the unit circle of the complex plane. The transfer matrix $\mathbf{G}$ between time steps n and $n+1$ is the amplification factor (or the Fourier symbol) of the full discretization.

The linear stability of the coupled time-space discretization is now analyzed through the study of the spectral radius of the amplification factor $\rho_{\mathbf{G}}$. The following parameters are considered in the study:

- $\theta \in [0, 2\pi], \Delta\theta = \pi/8$,
- $\kappa \in [-\pi, \pi], \Delta\kappa = \pi/32$,
- $\vartheta \in [0, 2\pi], \Delta\vartheta = \pi/8$,
- $\mu = (\pi/2, \pi/3, \pi/4)$.

The CFL stability limits are given for those parameters in Table 3. Note that values are associated with the classical definition of the CFL number given by Eq. (58). To compare CFL numbers used in high-order discontinuous methods with classical methods (like Finite Volume, Finite Element or Finite Difference), one can introduced an equivalent CFL number $\hat{\nu}$ defined as $\hat{\nu} = (p+1)\nu$ [47]. This equivalent CFL number

makes sense in the one-dimensional case since $\Delta x/(p+1)$ is a length scale corresponding to the mean distance between two adjacent SP. However, this definition is not necessarily the most adequate one on triangles, as shown by Chalmers and Krivodonova [75] for the DG method. To the authors' knowledge, there is no consensus on the definition on an equivalent CFL number for high-order discontinuous methods on simplex cells. The classical CFL definition is thus preferred here.

	$\theta = 0$	$\theta = \pi/8$	$\theta = \pi/4$	$\theta \in [0, 2\pi], \Delta\theta = \pi/8$
$SDRT_4$	0.097	0.081	0.075	0.075
$SDRT_5$	0.065	0.057	0.054	0.054

Table 3: CFL stability limits ν for SDRT schemes on triangles coupled with the SSP3s3o temporal schemes

320

5. Numerical Experiments

5.1. Convection of an isentropic vortex

A nonlinear case is now studied by considering the two-dimensional Euler equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = 0, \ \text{ in } \Omega \times [0, t_f], \tag{60}$$

where $\mathbf{u}$, $\mathbf{f}$ and $\mathbf{g}$ are given by:

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho U \\ \rho V \\ E \end{pmatrix}, \ \mathbf{f} = \begin{pmatrix} \rho U \\ \rho U^2 + P \\ \rho UV \\ U(E+P) \end{pmatrix}, \ \mathbf{g} = \begin{pmatrix} \rho V \\ \rho VU \\ \rho V^2 + P \\ V(E+P) \end{pmatrix}. \tag{61}$$

In Eq. (61), ρ is the density, U (respectively V) is the velocity component in the x (respectively y) direction, E is the total energy and P is the pressure determined from the following equation of state:

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho(U^2 + V^2)\right), \tag{62}$$

where the constant ratio of specific heats γ is equal to 1.4 for air.

To assess the SDRT scheme accuracy and capability to preserve vorticity in an unsteady inviscid flow, the convection of an isentropic vortex (COVO) test case from the International Workshop on High-Order CFD Methods [76] is studied. An isentropic vortex is transported by an inviscid uniform flow defined by $P_\infty = 10^5$ Pa, $T_\infty = 300$ K, $M_\infty = U_\infty/\sqrt{\gamma R_{gas} T_\infty} = 0.5$. The fluid is assumed to be a perfect gas, with a constant ratio of specific heats $\gamma = 1.4$. An isentropic vortex of characteristic radius $R = 0.005$ m and strength $\beta = 0.2$ is added to this mean flow around the point of coordinates $(X_c, Y_c) = (0.05, 0.05)$ through perturbation in U, V and the temperature T. The computation is thus initialized by the local velocity components U_0 and V_0 as well as temperature T_0:

$$U_0 = U_\infty \left(1 - \beta \, \frac{(y - Y_c)}{R} \, \exp\left(-r^2/2\right)\right), \tag{63}$$

$$V_0 = U_\infty \beta \, \frac{(x - X_c)}{R} \, \exp\left(-r^2/2\right), \tag{64}$$

$$T_0 = T_\infty - \frac{U_\infty^2 \beta^2}{2R_{gas}} \frac{(\gamma - 1)}{\gamma} \exp\left(-r^2\right), \tag{65}$$

where

$$r = \frac{\sqrt{(x - X_c)^2 + (y - Y_c)^2}}{R}. \tag{66}$$

Since the vortex is isentropic, the density can be computed using:

$$\rho_0 = \rho_\infty \left(\frac{T_0}{T_\infty} \right)^{\frac{1}{\gamma - 1}}, \tag{67}$$

where $\rho_\infty = \frac{P_\infty}{R_{gas} T_\infty}$. Euler equations are solved on the computational domain $\Omega = [0, L]^2$ where $L = 0.1\,\mathrm{m}$. Translational periodic boundary conditions are imposed for the left/right and top/bottom boundaries respectively. The SSP3s3o scheme introduced for the coupled time-space analysis is considered for the simulations. The time step is chosen sufficiently small so that the error from the time discretization is negligible compared to the spatial discretization error by setting the CFL number to 10^{-2}. At interfaces, Roe's Riemann solver [77] was used to compute the numerical flux. Figure 6 gives the initial solution by showing ρV contours (product of the density ρ and the y-velocity V) on a regular mesh composed of $2N^2$ triangles using a SDRT_5 scheme.

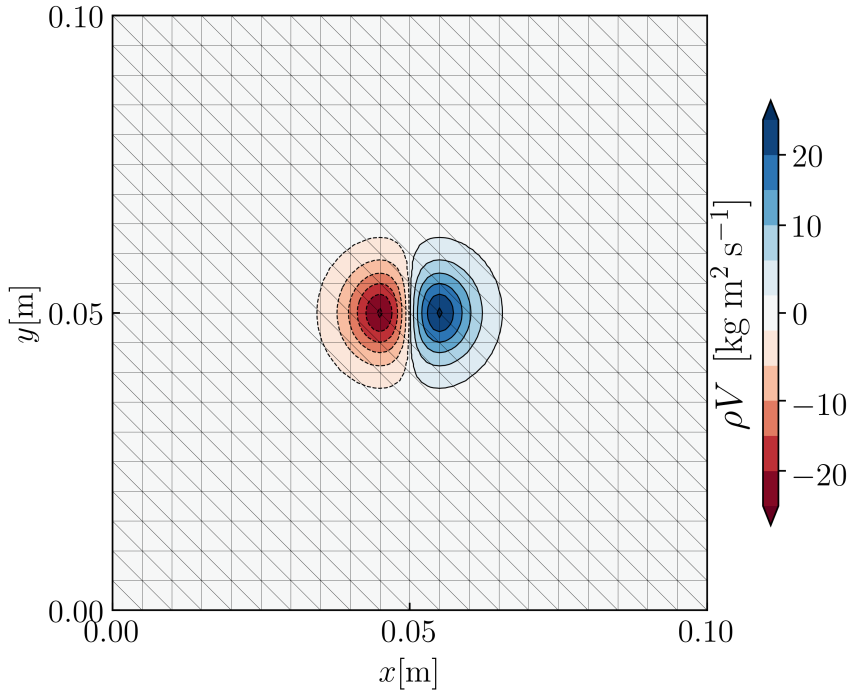


Figure 6: Initialization of the COVO test case using SDRT_5 and $N = 20$ on a triangular grid

A mesh refinement study is performed using regular triangular and hybrid grids of different size at the time corresponding to 5 periods. To generate hybrid grids, the left part of the computational domain ($x \in [0, 0.05]$) is meshed using quadrilateral elements whereas the right part ($x \in [0.05, 0.1]$) is meshed with triangles. The mesh is composed of $\frac{1}{2}N^2$ quadrilaterals and N^2 triangles. The numerical scheme used for quadrilateral cells is the standard SD method based on the interior FP located at the zeros of the corresponding Legendre polynomials. The very same polynomial degree is used for both triangular and quadrilateral elements.

To verify the order of accuracy of the SDRT scheme, the L_2 error on the density can be computed at each period on the domain as:

$$L_2 = \sqrt{\frac{\int_\Omega \left(\rho_{h_0} - \rho_{num}\right)^2 d\Omega}{\int_\Omega d\Omega}}, \tag{68}$$

where ρ_{h_0} is the polynomial approximation of the initial solution ρ_0.

In Eq. (68), the integral on the top can be expressed as the following sum on each cell:

$$\int_\Omega \left(\rho_{h_0} - \rho_{num}\right)^2 d\Omega = \sum_{i=1}^{N} \int_{\Omega_i} \left(\rho_{h_0}^{(i)} - \rho_{num}^{(i)}\right)^2 d\Omega, \tag{69}$$

where N is the number of cells on the domain Ω. Integration is then performed in the reference domain using a quadrature rule such that:

$$\sum_{i=1}^{N} \int_{\Omega_i} \left(\rho_{h_0}^{(i)} - \rho_{num}^{(i)}\right)^2 d\Omega = \sum_{i=1}^{N} \sum_{j=1}^{N_q} A \left|J^{(i,j)}\right| \omega_j \left(\rho_{h_0}^{(i)}(\boldsymbol{\xi}_j) - \rho_{num}^{(i)}(\boldsymbol{\xi}_j)\right)\Big)^2, \tag{70}$$

where A is the reference element area ($A = 1$ for a quadrilateral element and $A = 1/2$ for a triangular element), $|J^{(i,j)}|$ is the Jacobian determinant at the j-th integration point of the i-th cell and N_q is the number of quadrature points. The quadrature points are located at $\boldsymbol{\xi}_j$ and associated with the weight ω_j. Since ρ_{h_0} and ρ_{num} are polynomials of degree p, the term $\left(\rho_{h_0}^{(i)} - \rho_{num}^{(i)}\right)^2$ should be approximated using a quadrature of degree $2p$. On triangles, the integration is carried out using the 175-points symmetric quadrature given by Wandzura and Xiao [78], which can be used up to degree 30. On quadrilaterals, the integration is performed using the tensor product of two 1D integration at SP, with the appropriate Gauss-Chebyshev weights.

The L_2 norm of the density error is plotted in Fig. 7 after 5 time-periods and compared with the expected order slope. For both triangular and hybrid grids, the expected order of accuracy $p + 1$ is retrieved. The precise overall order of accuracy, computed using a least squares polynomial fit of degree one, is given in Table 4)

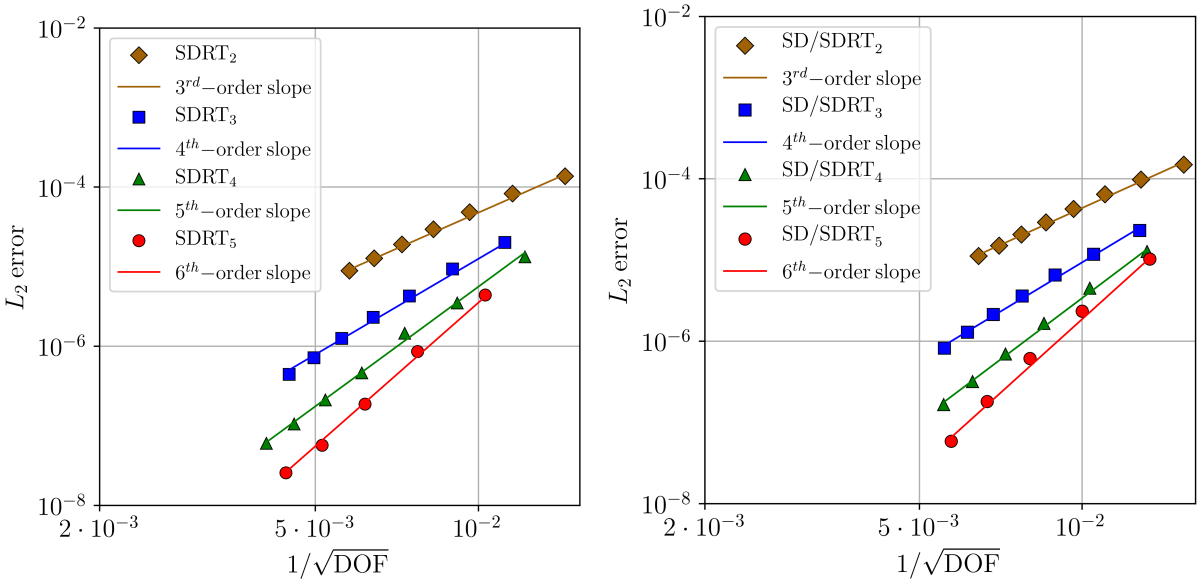
Scheme	Order of accuracy
$SDRT_2$	3.03
$SDRT_3$	4.02
$SDRT_4$	4.98
$SDRT_5$	6.19

(a) Triangular Grids

Scheme	Order of accuracy
$SD/SDRT_2$	3.00
$SD/SDRT_3$	4.02
$SD/SDRT_4$	5.05
$SD/SDRT_5$	6.09

(b) Hybrid Grids

Table 4: Overall accuracy orders for the COVO test case after 5 periods



(a) Triangular Grids

(b) Hybrid Grids

Figure 7: L_2 error and theoretical order of accuracy slopes for the COVO test case after 5 time-periods in log scale

5.2. Viscous flow over an NACA0012 airfoil

This test case aims to validate the method for the computation of viscous flow with a high-order triangular curved boundary representation. The compressible Navier-Stokes equations are solved and a laminar viscous flow over the NACA0012 airfoil is considered. The computational setup is defined by the angle of attack α, the Mach number M_∞ and the Reynolds number $Re = \rho_\infty U_\infty C/\mu_{d\infty}$, where C is the airfoil chord. Three different laminar flow conditions chosen from the NASA technical report [79] are considered:

- Case A: Symmetric subsonic flow, $M_\infty = 0.5$, $\alpha = 0°$, $Re = 5000$,
- Case B: Asymmetric subsonic flow, $M_\infty = 0.5$, $\alpha = 2°$, $Re = 5000$,
- Case C: Transonic flow, $M_\infty = 0.8$, $\alpha = 10°$, $Re = 500$.

The NACA0012 airfoil equation used is:

$$y = \pm 0.6\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4\right), \tag{71}$$

so the trailing edge has a zero thickness. At the airfoil, a no-slip adiabatic wall condition is imposed. To avoid spurious reflections on the boundary conditions, the farfield boundary is located 50 chords away from the airfoil. On the farfield boundary, pressure and temperature are imposed at $P_\infty = 101325\,\text{Pa}$ and $T_\infty = 293.15\,\text{K}$ and the velocity is imposed depending on the Mach number. Interface flux is then obtained by applying the approximated Riemann solver at the interface using the prescribed state outside and the extrapolated internal state. The computational domain is meshed with a C-type topology and has a total number of 2407 quadratic triangular elements (with 62 cells on the airfoil). A close view of the resulting mesh is shown in Fig. 8. Solutions are time-integrated using the SSP3s3o temporal scheme and the convection flux is Roe's scheme. Results are presented for SDRT schemes from the third- to the sixth-order ($p \in [\![2, 5]\!]$). The CFL number is chosen based on the maximum one affordable using $p = 5$, i.e. $\nu = 0.05$. The visualization process on triangular grids is done by extrapolating the solution at SP to high-order Lagrange elements nodes.

5.2.1. Thorough Analysis of Case C

Results obtained for Case C are first presented in details since the transonic flow can be considered as the 'most critical' test case. For the subsonic flows (Case A and B), briefer results will be given in the next section.

To monitor the computation convergence, the L_2 norm of the residual on the density between iteration n and $n+1$ is computed using:

$$||\text{Res}||_2 = \sqrt{\frac{\int_\Omega \left(\rho^{n+1} - \rho^n\right)^2 d\Omega}{\int_\Omega d\Omega}}. \tag{72}$$

Integration is performed using the 175-points symmetric quadrature given by Wandzura and Xiao [78]. The decay of the residual against number of iteration for SDRT_p schemes, $p \in [\![2,5]\!]$, is shown for the transonic Case C in Fig. 9. Despite the use of an explicit time-integration scheme, a very good convergence is obtained, starting from the constant flow field corresponding to the inflow conditions for each SDRT_p scheme, $p \in [\![2,5]\!]$.

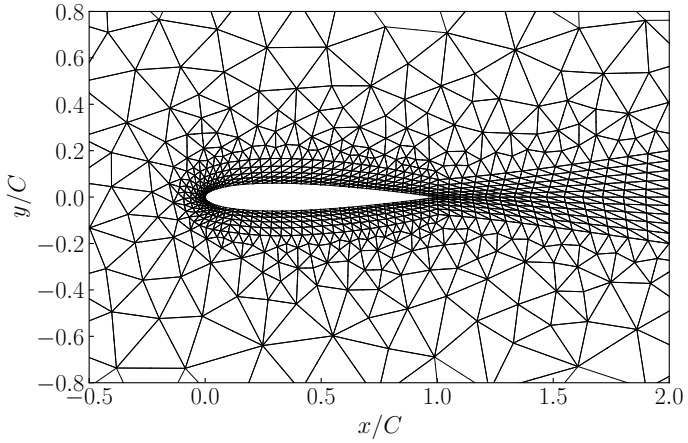


Figure 8: Close view of the unstructured mesh around a NACA0012airfoil

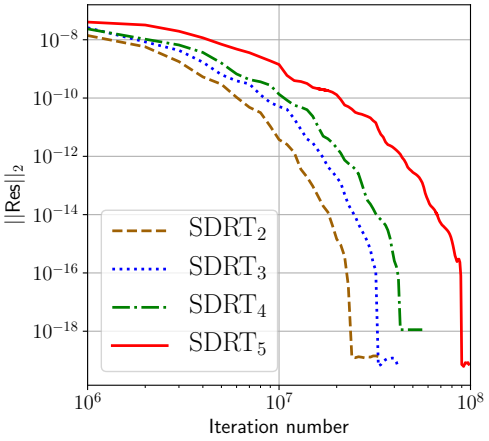
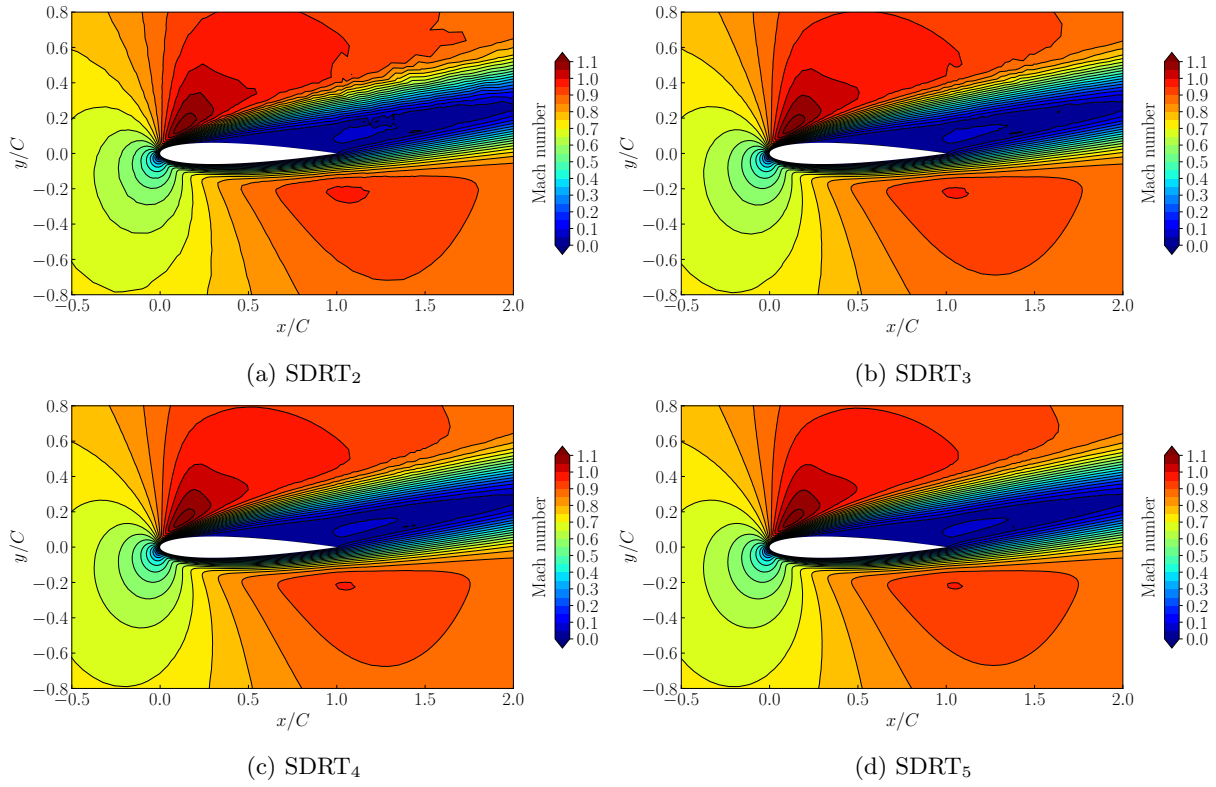


Figure 9: Convergence of the residual for the transonic viscous flow over a NACA0012 airfoil

Fig. 10 shows the Mach contours obtained with SDRT schemes of 3^{rd} to 6^{th} order of accuracy for Case C. The flow is accelerated at the airfoil upper surface and create a small supersonic zone ($M > 1$). However, as expected for this case, there is no shock wave developing. As the degree of the polynomial reconstruction increases, the solution becomes smoother and thus more accurate. For SDRT_2 and SDRT_3 schemes (Fig. 10a, 10b), discontinuous contour lines can be observed. Those discontinuities are induced by the visualization process, which is done independently on each triangular element, leading to different solution values at cell interfaces, and express a low resolution. The Mach contours given by the SDRT_4 and SDRT_5 schemes (Fig. 10c, 10d) show continuous lines for most of the domain. The remaining discontinuities located around the position $(x/C, y/C) = (1.4, 0.4)$ are due to the fact that the mesh used is refined for the wake given with an angle of attack $\alpha = 0^o$. Apart from this region, the Mach contours obtained show that the 5^{th} and 6^{th} order SDRT schemes converge to the same solution.



(a) SDRT$_2$ (b) SDRT$_3$

(c) SDRT$_4$ (d) SDRT$_5$

Figure 10: Case C: Mach number contours using SDRT$_p$ schemes, $p \in [\![2, 5]\!]$

The surface skin-friction coefficient C_f and the surface pressure coefficient C_p distributions are plotted in Fig. 11 and Fig. 12. For both coefficients, there is good agreement between the results obtained from SDRT$_p$ schemes and the NASA data [79]. In Fig. 11, a closer view shows that all SDRT schemes were able to capture the maximum value of the surface skin-friction coefficient at the leading edge. Small discontinuities between cells are observed, resulting from the interpolation post-processing step performed independently on each cell. The peak at the leading edge is quite accurately represented. A difference with the NASA data can be noticed at the trailing edge, where SDRT schemes did not manage to capture the maximum values due to low mesh refinement. A close view indicates that the SDRT C_f value gets a bit closer to the reference one when the order of accuracy increases. The surface pressure coefficient plot (Fig. 12) shows that all SDRT$_p$ schemes lead to excellent agreement with the NASA data, including at leading and trailing edges. A closer view shows the convergence of the results when the order increases.

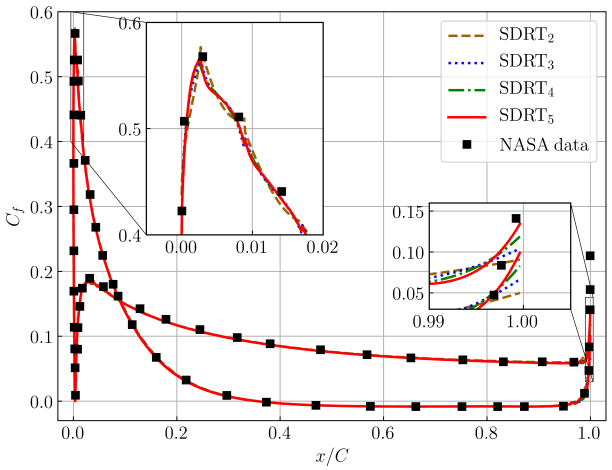


Figure 11: Case C: Surface skin-friction coefficient C_f

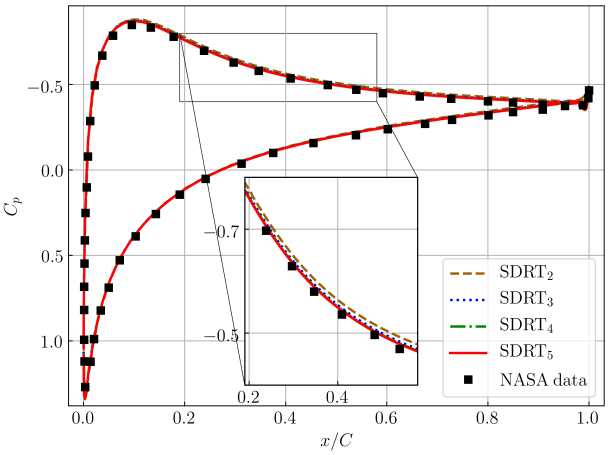


Figure 12: Case C: Surface pressure coefficient C_p

5.2.2. Aerodynamic Coefficients Computation for Three Different Laminar Flow Conditions

For each case, aerodynamic coefficients are computed for each SDRT scheme and compared to NASA reference data in Table 5.

For Case A, an additional comparison is made with results obtained using a standard fifth-order SD scheme in [80]. Values of the different coefficients converge as the order of accuracy increases and get closer to the reference data. Results using SDRT$_4$ are in excellent agreement with the NASA data using the refined grid (difference of 0.1%), whereas SDRT$_5$ matches results obtained using the fifth-order SD scheme (difference of 0.7%). All of the separation point locations predicted by SDRT schemes lies in the interval given by the two references up to the third decimal.

For Case B, the SDRT$_2$ scheme leads to a good prediction of C_D (1% of difference), but the surface skin-friction and pressure coefficients are not accurately determined ($\sim 4.5\%$ of difference). For higher-order SDRT schemes, the C_D value converges to the NASA reference (13% for SDRT$_3$, 2.8% for SDRT$_4$ and 0.8% for SDRT$_5$), with close values for $(C_D)_p$ and $(C_D)_f$. The same behavior is observed for the separation point location, even if it is slightly overestimated (2.8% for $p = 5$).

Finally, for Case C, results for the total drag coefficient are in excellent agreement with the NASA reference, with a difference of $< 2\%$ for all schemes. For $p > 2$, all coefficients are converged up to the third decimal. The location of the separation point is slightly smaller than the reference ones but remains in good agreement.

		DoF Number	C_D	$(C_D)_p$	$(C_D)_f$	x_{sep}/C
Case A	$SDRT_2$	14,442	0.0583644	0.0220679	0.0362965	0.809
	$SDRT_3$	24,070	0.0573619	0.0239848	0.0333771	0.808
	$SDRT_4$	36,105	0.0554990	0.0225962	0.0329028	0.810
	$SDRT_5$	50,547	0.0543702	0.0218255	0.0325447	0.814
	NASA [79]	1024×512 (524,288)	0.0555743	0.0227887	0.0327855	0.808
	Fifth-order SD [80]	360×120 (43,200)	0.05476	0.02225	0.03251	0.814
Case B	$SDRT_2$	14,442	0.0574735	0.0233123	0.0341613	0.584
	$SDRT_3$	24,070	0.0644525	0.0298255	0.0346270	0.580
	$SDRT_4$	36,105	0.0584788	0.0252599	0.0332190	0.579
	$SDRT_5$	50,547	0.0573627	0.0245549	0.0328078	0.577
	NASA [79]	1024×512 (524,288)	0.0568914	0.0243173	0.0325741	0.561
Case C	$SDRT_2$	14,442	0.272783	0.148275	0.124508	0.357
	$SDRT_3$	24,070	0.270953	0.147087	0.123867	0.356
	$SDRT_4$	36,105	0.270255	0.146632	0.123622	0.355
	$SDRT_5$	50,547	0.270224	0.146715	0.123509	0.355
	NASA [79]	1024×512 (524,288)	0.275155	0.147544	0.127611	0.362

Table 5: Comparison of drag coefficients and separation point location

5.3. Viscous flow around a circular cylinder

The last test case aims to validate the method for the computation of viscous flow using 2D hybrid mesh. A steady laminar viscous flow at Re = 20 around a cylinder is considered. The Mach number is $M_\infty = 0.1$ and the Reynolds number is defined by $\mathrm{Re} = \rho_\infty U_\infty d/\mu_{d\infty}$, where the dynamic viscosity is $\mu_{d\infty} = 1.853 \cdot 10^{-3}$ Pa s, and the cylinder diameter is $d = 1$ m. The density ρ_∞ and the velocity U_∞ can be deduced from the temperature $T = 300$ K and the constant ratio of specific heats $\gamma = 1.4$. The cylinder is placed in a rectangular domain. The farfield boundaries are located 10 diameters away from the cylinder in the upstream, upward and downward directions and 30 diameters away in the downstream direction. A hybrid mesh of 3427 elements is used, with 196 quadrilateral elements near the cylinder and 3231 triangles in the rest of the domain. A close view of the mesh is provided in Fig. 13. On the farfield boundary, the pressure, temperature and velocity are settled. At the cylinder surface, a no-slip isothermal wall condition is imposed at $T_W = 300$ K. On quadrilateral elements, the standard SD method based on the interior FP located at the zeros of the corresponding Legendre polynomials is used. The same polynomial degree is used for both triangular and quadrilateral elements. Roe's Riemann solver is used to compute flux at interface flux points and the CFL number is set to 0.05 (the maximum one affordable using $p = 5$). The computation convergence is monitored by computing the L_2 norm of the residual on the density between iteration n and $n+1$ using:

$$||\mathrm{Res}||_2 = \sqrt{\frac{\int_\Omega \left(\rho^{n+1} - \rho^n\right)^2 d\Omega}{\int_\Omega d\Omega}}. \tag{73}$$

Integration is performed using the 175-points symmetric quadrature given by Wandzura and Xiao [78]. The decay of the residual against number of iteration for $\mathrm{SD/SDRT}_p$ schemes, $p \in [\![2, 5]\!]$, is shown in Fig. 14. As for NACA test cases, computations start from the constant flow field corresponding to the inflow conditions and computations are very well converged.

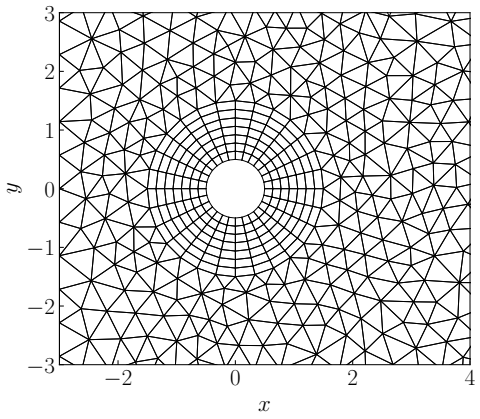


Figure 13: Close view of the hybrid mesh for the viscous flow around a cylinder

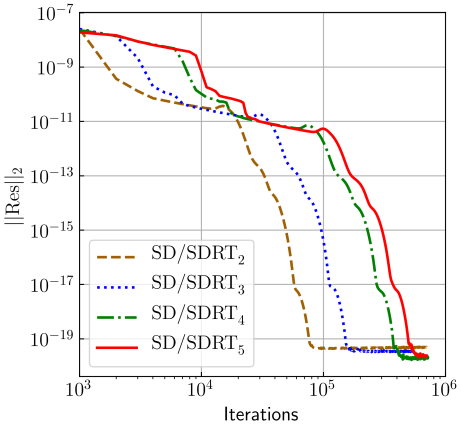
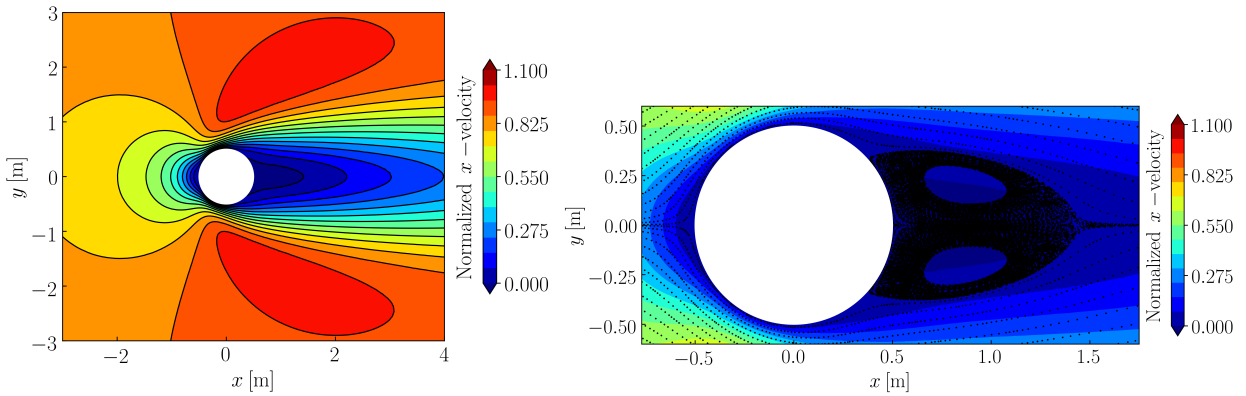


Figure 14: Convergence of the residual for the viscous flow around circular cylinder

Figure 15 shows the normalized x-velocity contours and streamlines around the cylinder obtained using a SD/SDRT$_5$ scheme. Streamlines show a recirculation zone within the wake of the cylinder where two vortices are generated.



(a) Normalized x-velocity contours

(b) Close view of streamlines and normalized x-velocity contours

Figure 15: Normalized x-velocity contours and streamlines around the cylinder using a SD/SDRT$_5$ scheme

Values of the drag coefficients as well as the separation angle θ_{sep} and the normalized reattachment length L/d obtained using SD/SDRT schemes are compared with different reference values in Table 6. Values presented are computed after applying a Savitzky-Golay filter; however, note that the maximum difference between values on all coefficients obtained with and without the filter is around 0.2%. Reference results [81–83] are based on finite difference approximation and Cartesian grids. Compared to data from Dennis and Chang, the SD/SDRT scheme overestimates drag coefficients values. This overestimation could be due to the fact that linear quadrilateral elements were used. A mesh composed of quadratic elements could lead to better results. Additionally, using a no-slip adiabatic wall condition at the cylinder surface instead of an isothermal wall condition could be more appropriate. However, compared to other reference data, SD/SDRT schemes lead to a proper estimation of drag coefficients, separation angle and reattachment length.

	C_D	$(C_D)_p$	$(C_D)_f$	θ_{sep}	L/d
SD/SDRT$_2$	2.135	1.275	0.860	45.5°	0.915
SD/SDRT$_3$	2.145	1.284	0.860	45.4°	0.915
SD/SDRT$_4$	2.146	1.286	0.860	45.4°	0.916
SD/SDRT$_5$	2.147	1.287	0.860	45.3°	0.916
Dennis and Chang [81]	2.045	1.233	0.812	43.7°	0.94
Russell and Z. Jane Wang [82]	2.13	-	-	43.3°	0.94
Calhoun [83]	2.19	-	-	45.5°	0.91

Table 6: Comparison of drag coefficients and separation angle for flow over a cylinder

6. Conclusion

For simulations that need a low-dissipation low-dispersion scheme such as vortex-dominated flow and unstructured grids to cope with complex geometry, the tendency today is to consider numerical schemes using piecewise continuous polynomials inside mesh cells, but without requiring continuity at mesh interfaces. While the most popular approach is certainly the DG method, alternatives were proposed during the last ten years. Among them, the SD method solves the strong form of equations, as the standard Finite Difference approach, but accounts for discontinuities of the solution at mesh interface using an approximated Riemann solver to compute the flux, as the Finite Volume method. It is today applied routinely to segments, quadrilaterals and hexahedra following a tensor approach. The weakness of the method lies in its inability to deal with hybrid grids.

In this context, the goal of this paper is to propose an extension of the SD method to 2D hybrid grids composed of triangles and quadrilaterals. Following the pioneering work of Balan *et al.* [38, 39], accounting for triangles is possible by introducing the RT space to approximate the flux. Compared to the literature, the proposed work gives new results since SDRT formulations based on new sets of interior FP were shown as spatially stable for $p = 4$ and $p = 5$. These sets of points were determined through an optimization process based on the Fourier analysis. Proof of the spatial stability of this formulation was given through a study of the spatial operator spectrum and the coupled time-space discretization was then studied to determine CFL stability limits. A theoretical analysis shows that the present formulation differs with the FR/CPR approaches. There are then two alternative approaches to the DG formulation valid on both quadrilaterals and simplex cells. The last effort deals with validation and it is shown that the formulation proposed in the paper is able to perform simulations for Euler and Navier-Stokes equations. The theoretical order of accuracy is recovered and coherent results on academic test cases are obtained.

The present work is the first effort towards the definition of the SD method for 3D simulations on hybrid grids and many questions are still open. Future work should deal with the definition of a stable SD formulation for tetrahedra and on spectral properties of the proposed schemes (dissipation and dispersion). In the later case, a specific analysis is necessary due to the difficulty to treat all directions at the same time in a regular grid with a given reference length scale in all directions. Finally, the demonstration of the stability in the non-linear case could be addressed through an entropy stable stability analysis.

Acknowledgments

Dr. Guillaume Puigt is partially supported by LMA2S (Laboratoire de Mathématiques Appliquées à l'Aéronautique et au Spatial), the Applied Mathematics Lab of ONERA.

Appendix A. Definition of the Raviart-Thomas (RT) space

The Raviart-Thomas (RT) finite element spaces were originally introduced by Raviart and Thomas [84] to approximate the Sobolev space H(div) defined by:

$$\mathrm{H(div)} = \{u \in \left(L^2(K)\right)^d,\ \nabla \cdot u \in L^2(K)\}, \tag{A.1}$$

where d is the dimension, K is a bounded open subset of $\mathbb{R}^d$ with a Lipshitz continuous boundary, $L^2(K)$ is the Hilbert space of square integrable function defined on K. The extension to the three-dimensional case considering K as a tetrahedron or a cube was proposed by Nedelec [85]. The space RT_k spanned by the Raviart-Thomas basis functions of degree k is the smallest polynomial space such that the divergence maps RT_k onto $\mathbb{P}_k$, the space of piecewise polynomials of degree $\leq k$. Considering the reference triangle $\mathcal{T}$, the RT space of order k is defined in 2D by:

$$\mathrm{RT}_k = (\mathbb{P}_k)^2 + \begin{pmatrix} x \\ y \end{pmatrix} \bar{\mathbb{P}}_k, \tag{A.2}$$

where $\mathbb{P}_k$ is the space of polynomials of degree at most k:

$$\mathbb{P}_k(x, y) = span\{x^i y^j,\ i, j \geq 0, i + j \leq k\}, \tag{A.3}$$

$\bar{\mathbb{P}}_k$ is the space of polynomials of degree k:

$$\bar{\mathbb{P}}_k(x, y) = span\{x^i y^j,\ i, j \geq 0, i + j = k\}, \tag{A.4}$$

and $(\mathbb{P}_k)^2 = (\mathbb{P}_k, \mathbb{P}_k)^\intercal$ is the two dimensional vector space for which each component is a polynomial of degree at most p. The dimension of each space is $\dim \mathbb{P}_k = \frac{(k+1)(k+2)}{2}$, $\dim \mathbb{P}_k^2 = (k+1)(k+2)$, $\dim \bar{\mathbb{P}}_k = k+1$ and thus $\dim \mathrm{RT}_k = (k+1)(k+3)$. Let us denote $\boldsymbol{\phi}_n, n \in [\![1, N_{FP}]\!]$ the monomials which form a basis in the RT_k space where

$$N_{FP} = (k+1)(k+3). \tag{A.5}$$

Determination of $\boldsymbol{\phi}_n$ for $\mathbf{RT}_1$, $N_{FP} = 8$

$$\mathbb{P}_1(x, y) = span\left\{1, x, y\right\}, \tag{A.6}$$

$$\mathbb{P}_1^2(x, y) = span\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix}\right\}, \tag{A.7}$$

$$\bar{\mathbb{P}}_1(x, y) = span\{x, y\}, \tag{A.8}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \bar{\mathbb{P}}_1 = span\left\{\left(\begin{pmatrix} x \\ y \end{pmatrix} x\right), \left(\begin{pmatrix} x \\ y \end{pmatrix} y\right)\right\} = \left\{\begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix}\right\}, \tag{A.9}$$

$$\boldsymbol{\phi}_n = \{\phi_1, \cdots, \phi_8\} = \left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix}, \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix}\right\}. \tag{A.10}$$

Appendix B. Proriol-Koornwinder-Dubiner (PKD) basis

The PKD basis is obtained by forming a tensor product of one-dimensional Jacobi polynomials based on a Cartesian coordinate system. Since the Jacobi polynomials are defined on the reference interval $[-1, 1]$, the 2D PKD basis obtained using tensor product is defined on the reference quadrilateral bounded by the same constant limits, i.e. $\mathcal{Q} := \{(x, y) : -1 \leq x, y \leq 1\}$. This section follows the generalised tensor product modal expansion notations defined in [53].

Appendix B.1. Collapsed coordinated system

To express the PKD basis on the reference triangle $\mathcal{T} := \{(x, y) : -1 \le x, y \le 0, x + y \le 0\}$, the first step is to map the reference triangle onto the reference quadrilateral. This transformation is defined by:

$$\begin{cases} \xi = 2\frac{(1+x)}{(1-y)} - 1, \\ \eta = y, \end{cases} \quad \text{or} \quad \begin{cases} x = \frac{(1+\xi)(1-\eta)}{2} - 1, \\ y = \eta. \end{cases} \tag{B.1}$$

The triangular element can now be describe with the new coordinates as $\mathcal{T} := \{(\xi, \eta) : -1 \le \xi, \eta \le 1\}$, bounded by the very same constant limits. The system of coordinates (ξ, η) is referred to as the collapsed coordinate system or Duffy coordinates.

Appendix B.2. PKD basis

Using the collapsed coordinates, the PKD basis is constructed as a family of L^2-orthogonal polynomials on $\mathcal{T}$ derived from a warped product of one-dimensional Jacobi polynomials:

$$\Phi_{i,j} = P_i^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i + j \le p. \tag{B.2}$$

In Eq. (B.2), $P_i^{\alpha,\beta}$ denotes the corresponding n-th order Jacobi polynomials on the interval $[-1, 1]$ which, under the Jacobi weight $(1-x)^\alpha(1+x)^\beta$ are orthogonal polynomials, i.e.,

$$\int_{-1}^{1}(1-x)^\alpha(1+x)^\beta P_i^{\alpha,\beta}(x)P_j^{\alpha,\beta}(x)dx = \frac{2^{\alpha+\beta+1}}{2i+\alpha+\beta+1}\frac{\Gamma(i+\alpha+1)\Gamma(i+\beta+1)}{i!\Gamma(i+\alpha+\beta+1)}\delta_{ij}, \tag{B.3}$$

where Γ is the Gamma function

$$\Gamma(n) = (n-1)!. \tag{B.4}$$

In the case of $\beta = 0$, Eq. (B.3) can be simplify as

$$\int_{-1}^{1}(1-x)^\alpha P_i^{\alpha,0}(x)P_j^{\alpha,0}(x)dx = \frac{2^{\alpha+1}}{2i+\alpha+1}\delta_{ij}. \tag{B.5}$$

Remark: The 2D PKD basis functions are polynomials in both (x, y) and (ξ, η) spaces.
Remark: The 'wrapped product' property refers to the fact that the 2D PKD basis functions can be expressed as the product of two polynomials, one in ξ and one in η.

Appendix B.3. Demonstration of L^2 orthogonality

To demonstrate the L^2 orthogonality of the PKB basis functions, the following integral must be evaluated:

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \int_{-1}^{1}\int_{-1}^{-y} \Phi_{i,j}\Phi_{k,l}dxdy, \tag{B.6}$$

which can be written in terms of the collapse coordinate system (ξ, η):

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \int_{-1}^{1}\int_{-1}^{1} P_i^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta)P_k^{0,0}(\xi)\left(\frac{1-\eta}{2}\right)^k P_l^{2k+1,0}(\eta)\,|J|\,d\xi d\eta, \tag{B.7}$$

where the Jacobian determinant is

$$|J| = \left|\frac{\partial(x,y)}{\partial(\xi,\eta)}\right| = \frac{1-\eta}{2}. \tag{B.8}$$

Eq. (B.7) can be written:

$$\begin{aligned} \langle \Phi_{i,j} | \Phi_{k,l} \rangle &= \int_{-1}^{1}\int_{-1}^{1}\left(\frac{1-\eta}{2}\right)^{i+k+1} P_i^{0,0}(\xi)P_j^{2i+1,0}(\eta)P_k^{0,0}(\xi)P_l^{2k+1,0}(\eta)d\xi d\eta \\ &= \frac{1}{2^{i+k+1}}\int_{-1}^{1} P_i^{0,0}(\xi)P_k^{0,0}(\xi)d\xi \int_{-1}^{1}(1-\eta)^{i+k+1} P_j^{2i+1,0}(\eta)P_l^{2k+1,0}(\eta)d\eta. \end{aligned} \tag{B.9}$$

The first integral value follow from the orthogonality of Legendre polynomials, which are the special case $(\alpha, \beta) = (0, 0)$ of Jacobi polynomials:

$$\int_{-1}^{1} P_i^{0,0}(\xi) P_k^{0,0}(\xi) d\xi = \frac{2\delta_{ik}}{2i+1}. \tag{B.10}$$

The first integral is equal to zero if $i \neq k$. When $i = k$, using Eq. (B.5), the second integral becomes:

$$\int_{-1}^{1} (1-\eta)^{i+k+1} P_j^{2i+1,0}(\eta) P_l^{2k+1,0}(\eta) d\eta = \int_{-1}^{1} (1-\eta)^{2i+1} P_j^{2i+1,0}(\eta) P_l^{2i+1,0}(\eta) d\eta = \frac{2^{2i+2}\delta_{jl}}{2j+2i+2}. \tag{B.11}$$

Eq. B.9 becomes:

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \frac{\delta_{ik}\delta_{jl}}{(i+1/2)(i+j+1)}. \tag{B.12}$$

From Eq. (B.12), if $i \neq k$ and $j \neq l$, $\langle \Phi_{i,j} | \Phi_{k,l} \rangle$ is necessarily equal to zero, which shows the orthogonatily of the PKD basis in $\mathcal{T}$.

Appendix B.4. Normalized PKD basis

From Eq. (B.12), it comes:

$$\|\Phi_{i,j}\|_{\mathcal{T}}^2 = \langle \Phi_{i,j} | \Phi_{i,j} \rangle = \frac{1}{(i+1/2)(i+j+1)}, \tag{B.13}$$

which leads to the expression of the L^2 normalized 2D PKD basis:

$$\Phi_{i,j} = \sqrt{(i+1/2)(i+j+1)}\, P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p. \tag{B.14}$$

Appendix C. Matrices Formulation for the Fourier Analysis

The matrices $\mathbf{M}^{0,0}$, $\mathbf{M}^{-1,0}$, $\mathbf{M}^{+1,0}$, $\mathbf{M}^{0,-1}$ and $\mathbf{M}^{0,+1}$ involved in the SDRT spatial discretization for the Fourier analysis on triangular elements (51) are detailed in this appendix. Those matrices are given as:

$$\mathbf{M}^{0,0} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,0} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{C.1}$$

$$\mathbf{M}^{-1,0} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{-1,0} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{C.2}$$

$$\mathbf{M}^{+1,0} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{+1,0} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{C.3}$$

$$\mathbf{M}^{0,-1} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,-1} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{C.4}$$

$$\mathbf{M}^{0,+1} = \begin{bmatrix} \mathbf{D}_{jk} & O_{N_{SP},N_{FP}} \\ O_{N_{SP},N_{FP}} & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C}^{0,+1} \begin{bmatrix} \mathbf{T}_{kj} & O_{N_{FP},N_{SP}} \\ O_{N_{FP},N_{SP}} & \mathbf{T}_{kj} \end{bmatrix}, \tag{C.5}$$

with $j \in [\![1, N_{SP}^{tri}]\!], k \in [\![1, N_{FP}^{tri}]\!]$ and $O_{m,n}$ is the zero matrix of size $m \times n$. The transfer matrix is given by Eq. (18):

$$\mathbf{T}_{kj} = \sum_{m=1}^{N_{SP}^{tri}} \left(\Phi_m(\boldsymbol{\xi}_j)\right)^{-1} \Phi_m(\boldsymbol{\xi}_k), \tag{C.6}$$

and the differentiation matrix by Eq. (23):

$$\mathbf{D}_{jk} = \sum_{n=1}^{N_{FP}^{tri}} \left(\boldsymbol{\phi}_n(\boldsymbol{\xi}_k)\cdot\hat{\mathbf{n}}_k\right)^{-1} \hat{\nabla}\cdot\boldsymbol{\phi}_n(\boldsymbol{\xi}_j). \tag{C.7}$$

The velocity matrices $\mathbf{C}^{0,0}, \mathbf{C}^{-1,0}, \mathbf{C}^{+1,0}, \mathbf{C}^{0,-1}$ and $\mathbf{C}^{0,+1}$ are given by:

$$\mathbf{C}^{0,0} = \begin{bmatrix}\begin{bmatrix} \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & \mathbf{C}^I_{N_i} \end{bmatrix} & \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} \\ \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} & \begin{bmatrix} \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^L & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & \mathbf{C}^I_{N_i} \end{bmatrix} \end{bmatrix}, \tag{C.8}$$

$$\mathbf{C}^{-1,0} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} \\ \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{C.9}$$

$$\mathbf{C}^{0,+1} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \\ \begin{bmatrix} O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & \mathbf{C}^R & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{C.10}$$

$$\mathbf{C}^{0,-1} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} \\ \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{C.11}$$

$$\mathbf{C}^{+1,0} = \begin{bmatrix} \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \\ \begin{bmatrix} \mathbf{C}^R & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_e} & O_{N_e,N_i} \\ O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_e} & O_{N_i,N_i} \end{bmatrix} & \begin{bmatrix} O_{N_{FP},N_{FP}} \end{bmatrix} \end{bmatrix}, \tag{C.12}$$

where $\mathbf{C}^I$, $\mathbf{C}^L$ and $\mathbf{C}^R$ are defined by:

$$\mathbf{C}^I = \left[\mathrm{diag}(|J|J^{-1}(\mathbf{c}\cdot\hat{\mathbf{n}}))\right]_{N_i,N_i}, \tag{C.13}$$

$$\mathbf{C}^L = (\mathbf{c}\cdot\mathbf{n})\left[\mathrm{diag}(\frac{1+\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2})\right]_{N_e,N_e}, \tag{C.14}$$

and

$$\mathbf{C}^R = (\mathbf{c}\cdot\mathbf{n})\begin{bmatrix} 0 & \cdots & \frac{1-\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} \\ \vdots & \iddots & \vdots \\ \frac{1-\mathrm{sign}(\mathbf{c}\cdot\mathbf{n})}{2} & \cdots & 0 \end{bmatrix}_{N_e,N_e}. \tag{C.15}$$

Appendix D. Interior Flux Points Parametrization for the Optimization Algorithm

Flux Point	(ξ, η)
$\boldsymbol{\xi}_{16}$	$(\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$
$\boldsymbol{\xi}_{17}$	$(\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$
$\boldsymbol{\xi}_{18}$	$(\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$
$\boldsymbol{\xi}_{19}$	(α_2, α_2)
$\boldsymbol{\xi}_{20}$	$(\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$
$\boldsymbol{\xi}_{21}$	$(\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$
$\boldsymbol{\xi}_{22}$	$(\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$
$\boldsymbol{\xi}_{23}$	$(\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$
$\boldsymbol{\xi}_{24}$	(α_1, α_1)
$\boldsymbol{\xi}_{25}$	$(\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$
$\boldsymbol{\xi}_{26:35}$	$\boldsymbol{\xi}_{16:25}$

Table D.7: Interior FP coordinate parameters in a triangular element for SDRT_4

Flux Point	(ξ, η)	Flux Point	(ξ, η)
$\boldsymbol{\xi}_{19}$	(α_1, α_1)	$\boldsymbol{\xi}_{27}$	$(\beta_3/2 - \gamma_3, \beta_3/2 + \gamma_3)$
$\boldsymbol{\xi}_{20}$	$(\beta_6/2 - \gamma_6, \beta_6/2 + \gamma_6)$	$\boldsymbol{\xi}_{28}$	$(\beta_2/2 + \gamma_2, \beta_2/2 - \gamma_2)$
$\boldsymbol{\xi}_{21}$	$(\beta_6/2 + \gamma_6, \beta_6/2 - \gamma_6)$	$\boldsymbol{\xi}_{29}$	$(\beta_3/2 + \gamma_3, \beta_3/2 - \gamma_3)$
$\boldsymbol{\xi}_{22}$	(α_3, α_3)	$\boldsymbol{\xi}_{30}$	$(\beta_5/2 + \gamma_5, \beta_5/2 - \gamma_5)$
$\boldsymbol{\xi}_{23}$	(α_2, α_2)	$\boldsymbol{\xi}_{31}$	$(\beta_4/2 + \gamma_4, \beta_4/2 - \gamma_4)$
$\boldsymbol{\xi}_{24}$	$(\beta_2/2 - \gamma_2, \beta_2/2 + \gamma_2)$	$\boldsymbol{\xi}_{32}$	$(\beta_1/2 - \gamma_1, \beta_1/2 + \gamma_1)$
$\boldsymbol{\xi}_{25}$	$(\beta_5/2 - \gamma_5, \beta_5/2 + \gamma_5)$	$\boldsymbol{\xi}_{33}$	$(\beta_4/2 - \gamma_4, \beta_4/2 + \gamma_4)$
$\boldsymbol{\xi}_{26}$	$(\beta_1/2 + \gamma_1, \beta_1/2 - \gamma_1)$	$\boldsymbol{\xi}_{34:48}$	$\boldsymbol{\xi}_{19:33}$

Table D.8: Interior FP coordinate parameters in a triangular element for SDRT_5

References

[1] B. Cockburn, G. E. Karniadakis, C.-W. Shu (Eds.), Discontinuous Galerkin Methods, Springer Berlin Heidelberg, 2000. doi:10.1007/978-3-642-59721-3.

[2] J. S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods, Springer New York, 2008. doi:10.1007/978-0-387-72067-8.

[3] D. Di Pietro, A. Ern, Mathematical Aspects of Discontinuous Galerkin Methods, Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-22980-0.

[4] X. Feng, O. Karakashian, Y. Xing (Eds.), Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations, Springer International Publishing, 2014. doi:10.1007/978-3-319-01818-8.

[5] V. Dolejší, M. Feistauer, Discontinuous Galerkin Method, Springer International Publishing, 2015. doi:10.1007/978-3-319-19267-3.

[6] M. Uzunca, Adaptive Discontinuous Galerkin Methods for Non-linear Reactive Flows, Springer International Publishing, 2016. doi:10.1007/978-3-319-30130-3.

[7] A. Cangiani, Z. Dong, E. H. Georgoulis, P. Houston, hp-Version Discontinuous Galerkin Methods on Polygonal and Polyhedral Meshes, Springer International Publishing, 2017. doi:10.1007/978-3-319-67673-9.

[8] S. Du, F.-J. Sayas, An Invitation to the Theory of the Hybridizable Discontinuous Galerkin Method, Springer International Publishing, 2019. doi:10.1007/978-3-030-27230-2.

[9] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. Ven, K. Sørensen (Eds.), ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications, Vol. 13 of Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-03707-8.

[10] 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Von Karman Institute, VKI LS 2010-01, 2010.

[11] N. Kroll, C. Hirsch, F. Bassi, C. Johnston, K. Hillewaert (Eds.), IDIHOM: Industrialization of High-Order Methods - A Top-Down Approach, Results of a Collaborative Project Funded by the European Union, Vol. 128 of Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer, 2015. doi:10.1007/978-3-319-12886-3.

[12] V. Brunet, E. Croner, A. Minot, J. de Laborderie, E. Lippinois, S. Richard, J.-F. Boussuge, J. Dombard, F. Duchaine, L. Gicquel, T. Poinsot, G. Puigt, G. Staffelbach, L. Segui, O. Vermorel, N. Villedieu, J.-S. Cagnone, K. Hillewaert, M. Rasquin, G. Lartigue, V. Moureau, T. Roger, V. Couaillier, E. Martin, M. de la Llave Plata, J.-M. Le Gouez, F. Renac, Comparison of various CFD codes for LES simulations of turbomachinery: from inviscid vortex convection to multi-stage compressor, in: ASME Turbo Expo 2018: Turbomahinery Technical Conference and Exposition, Oslo, Norway, 2018.

[13] M. Yu, Z.J. Wang, Y. Liu, On the accuracy and efficiency of discontinuous Galerkin, spectral difference and correction procedure via reconstruction methods, Journal of Computational Physics 259 (2014) 70–95. doi:10.1016/j.jcp.2013.11.023.

[14] H. T. Huynh, A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods, in: 18th AIAA Computational Fluid Dynamics Conference, 2007. doi:10.2514/6.2007-4079.

[15] Z.J. Wang, H. Gao, A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, Journal of Computational Physics 228 (21) (2009) 8161–8186. doi:10.1016/j.jcp.2009.07.036.

[16] Y. Allaneau, A. Jameson, Connections between the filtered discontinuous Galerkin method and the flux reconstruction approach to high order discretizations, Computer Methods in Applied Mechanics and Engineering 200 (2011) 3626–3636. doi:10.1016/j.cma.2011.08.019.

[17] R. Abgrall, E. L. Mldo, P. ffner, P. Offner, On the Connection between Residual Distribution Schemes and Flux Reconstruction, preprint (2018).
URL hal.archives-ouvertes.fr/hal-01820176

[18] P. E. Vincent, P. Castonguay, A. Jameson, A New Class of High-Order Energy Stable Flux Reconstruction Schemes, Journal of Scientific Computing 47 (1) (2011) 50–72. doi:10.1007/s10915-010-9420-z.

[19] P. Castonguay, P. Vincent, A. Jameson, A New Class of High-Order Energy Stable Flux Reconstruction Schemes for Triangular Elements, Journal of Scientific Computing 51 (1) (2012) 224–256. doi:10.1007/s10915-011-9505-3.

[20] H. Huynh, Z. Wang, P. Vincent, High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids, Computers & Fluids 98 (2014) 209–220. doi:10.1016/j.compfluid.2013.12.007.

[21] D. A. Kopriva, J. H. Kolias, A Conservative and Staggered-Grid Chebyshev and Multidomain Method and for Compressible and Flows, Journal of Computational Physics 125 (1996) 244–261. doi:10.1006/jcph.1996.0091.

[22] D. A. Kopriva, A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows. II. A Semi-Structured Method, Journal of Computational Physics 128 (2) (1996) 475–488. doi:10.1006/jcph.1996.0225.

[23] Y. Liu, M. Vinokur, Z.J. Wang, Spectral Difference Method for Unstructured Grids I: Basic Formulation, Journal of Computational Physics 216 (2) (2006) 780–801. doi:10.1016/j.jcp.2006.01.024.

[24] Z.J. Wang, Y. Liu, G. May, A. Jameson, Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations, Journal of Scientific Computing 32 (1) (2007) 45–71. doi:10.1007/s10915-006-9113-9.

[25] G. May, A. Jameson, A Spectral Difference Method for the Euler and Navier-Stokes Equations on Unstructured Meshes, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006. doi:10.2514/6.2006-304.

[26] Y. Sun, Z.J. Wang, Y. Liu, High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, Communications in Computational Physics 2 (2) (2007) 310–333.

[27] W. Chen, Y. Ju, C. Zhang, A collocated-grid spectral difference method for compressible flows, Computers & Fluids 196 (2020) 104341. doi:10.1016/j.compfluid.2019.104341.

[28] C. Liang, C. Cox, M. Plesniak, A comparison of computational efficiencies of spectral difference method and correction procedure via reconstruction, Journal of Computational Physics 239 (2013) 138–146. doi:10.1016/j.jcp.2013.01.001.
[29] C. Cox, W. Trojak, T. Dzanic, F. Witherden, A. Jameson, Accuracy, stability, and performance comparison between the spectral difference and flux reconstruction schemes (2020). arXiv:2009.06072.
[30] G. May, On the Connection Between the Spectral Difference Method and the Discontinuous Galerkin Method, Communications in Computational Physics 9 (4) (2011) 1071–1080. doi:10.4208/cicp.090210.040610a.
[31] K. Van den Abeele, C. Lacor, Z.J. Wang, On the connection between the spectral volume and the spectral difference method, Journal of Computational Physics 227 (2) (2007) 877–885. doi:10.1016/j.jcp.2007.08.030.
[32] K. Van den Abeele, C. Lacor, Z.J. Wang, On the Stability and Accuracy of the Spectral Difference Method, Journal of Scientific Computing 37 (2) (2008) 162–188. doi:10.1007/s10915-008-9201-0.
[33] A. Jameson, A Proof of the Stability of the Spectral Difference Method for All Orders of Accuracy, Journal of Scientific Computing 45 (1-3) (2010) 348–358. doi:10.1007/s10915-009-9339-4.
[34] Z.J. Wang, Y. Liu, The Spectral Difference Method for the 2D Euler Equations on Unstructured Grids, in: 17th AIAA computational Fluid Dynamics Conference, AIAA Paper 2005-5112, 2005.
[35] Y. Liu, M. Vinokur, Z.J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, in: Computational Fluid Dynamics 2004, Proceedings of the Third International Conference on Computational Fluid Dynamics (ICCFD3), Toronto, 12-16 July 2004, 2006, pp. 449–454.
[36] C. Liang, R. Kannan, Z.J. Wang, A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured grids, in: 18th AIAA Computational Fluid Dynamics Conference, 25-28 June, Miami, FL, AIAA Paper 2007-4326, 2007. doi:10.2514/6.2007-4326.
[37] C. Liang, A. Jameson, Z.J. Wang, Spectral difference method for compressible flow on unstructured grids with mixed elements, Journal of Computational Physics 228 (8) (2009) 2847–2858. doi:10.1016/j.jcp.2008.12.038.
[38] A. Balan, G. May, J. Schberl, A Stable and Spectral Difference and Method for Triangles, in: 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2011. doi:10.2514/6.2011-47.
[39] A. Balan, G. May, J. Schberl, A stable high-order Spectral Difference method for hyperbolic conservation laws on triangular elements, Journal of Computational Physics 231 (5) (2012) 2359–2375. doi:10.1016/j.jcp.2011.11.041.
[40] G. May, J. Schöberl, Analysis of a Spectral Difference Scheme with Flux Interpolation on Raviart-Thomas Elements, Tech. rep., Aachen Institute for Advanced Study in Computational Engineering Science (2010).
[41] M. Li, Z. Qiu, C. Liang, M. Sprague, M. Xu, C. A. Garris, A new high-order spectral difference method for simulating viscous flows on unstructured grids with mixed-element meshes, Computers & Fluids 184 (2019) 187–198. doi:10.1016/j.compfluid.2019.03.010.
[42] Z. Qiu, B. Zhang, C. Liang, M. Xu, A high-order solver for simulating vortex-induced vibrations using sliding-mesh spectral difference method and hybrid grids, International Journal for Numerical Methods in Fluids 90 (2019) 171–194. doi:10.1002/fld.4717.
[43] A. Meister, S. Ortleb, T. Sonar, M. Wirz, A comparison of the Discontinuous-Galerkin and Spectral-Difference Method on triangulations using PKD polynomials, Journal of Computational Physics 231 (2012) 7722–7729. doi:10.1016/j.jcp.2012.07.025.
[44] M. Pawlak, T. Sonar, A Spectral Difference Method with Dubiner Polynomials on Triangulations, in: Series in Contemporary Applied Mathematics, Co-Published with Higher Education Press, 2012, pp. 610–616. doi:10.1142/9789814417099_0063.
[45] M. G. Blyth, C. Pozrikidis, A Lobatto interpolation grid over the triangle, IMA Journal of Applied Mathematics 71 (1) (2006) 153–169. doi:10.1093/imamat/hxh077.
[46] A. Cassagne, J.-F. Boussuge, N. Villedieu, G. Puigt, I. DAst, A. Genot, JAGUAR: a new CFD code dedicated to massively parallel high-order LES computations on complex geometry, in: 50th 3AF International Conference on Applied Aerodynamics, 2015.
[47] J. Vanharen, G. Puigt, X. Vasseur, J.-F. Boussuge, P. Sagaut, Revisiting the spectral analysis for high-order spectral discontinuous methods, Journal of Computational Physics 337 (2017) 379–402. doi:10.1016/j.jcp.2017.02.043.
[48] A. Cassagne, G. Puigt, J. Boussuge, High-order Method for a New Generation of Large Eddy Simulation Solver, Tech. rep., Partnership for Advanced Computing in Europe (PRACE), available online at www.prace-ri.eu (2015).
[49] T. Poinsot, S. Lele, Boundary Conditions for Direct Simulations of Compressible Viscous Flows, Journal of Computational Physics 101 (1992) 104–129. doi:10.1016/0021-9991(92)90046-2.
[50] R. Fiévet, H. Deniau, J. Brazier, E. Piot, Numerical Study of Hypersonic Boundary-Layer Transition Delay through Second-Mode Absorption, in: AIAA Scitech 2020 Forum, AIAA Paper 2020-2061. doi:10.2514/6.2020-2061.
[51] R. Fiévet, H. Deniau, E. Piot, Strong compact formalism for characteristic boundary conditions with discontinuous spectral methods, Journal of Computational Physics 408 (2020) 109276. doi:10.1016/j.jcp.2020.109276.
[52] Z.J. Wang, High-order methods for the Euler and NavierStokes equations on unstructured grids, Progress in Aerospace Sciences 43 (2007) 1–41. doi:10.1016/j.paerosci.2007.05.001.
[53] G. E. Karniadakis, S. J. Sherwin, Spectral/HP Element Methods for Computational Fluid Dynamics, Oxford University Press, Oxford, 2005.
[54] J. Proriol, Sur une famille de polynômes à deux variables orthogonaux dans un triangle, Sci. Paris 257 (1957) 24592461.
[55] T. Koornwinder, Two-variable analogues of the classical orthogonal polynomials, in: R. Askey (Ed.), Theory and Applications of Special Functions, San Diego, 1975.
[56] M. Dubiner, Spectral methods on triangles and other domains, Journal of Scientific Computing 6 (4) (1991) 345–390. doi:10.1007/BF01060030.
[57] G. Pena, Spectral element approximation of the incompressible Navier-Stokes equations in a moving domain and applica-

tions, Ph.D. thesis, Ecole Polytechnique Fdrale de Lausanne (2009).
[58] D. Williams, Energy stable high-order methods for simulating unsteady, viscous, compressible flows on unstructured grids, Ph.D. thesis, Stanford University (2013).
[59] P. Castonguay, High-order energy stable flux reconstruction schemes for fluid flow simulations on unstructured grids, Ph.D. thesis, Stanford University (2012).
[60] A. Veilleux, Extension of the Spectral Difference method to simplex cells and hybrid grids, Ph.D. thesis, Cerfacs and Onera (2021).
[61] P. Hammer, O. Marlowe, A. Stroud, Numerical Integration Over Simplexes and Cones, Mathematical Tables and Other Aids to Computation 10 (55) (1956) 130–137. doi:10.1090/S0025-5718-1956-0086389-6.
[62] M. Laursen, M. Gellert, Some criteria for numerically integrated matrices and quadrature formulas for triangles, International Journal for Numerical Methods in Engineering 12(1) (1978) 67–76.
[63] D. Dunavant, High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle, International Journal for Numerical Methods in Engineering 21 (6) (1985) 1129–1148. doi:10.1002/nme.1620210612.
[64] L. Zhang, T. Cui, H. Liu, A set of symmetric quadrature rules on triangles and tetrahedra, J. Comput. Math. 27 (2009) 89–96.
[65] H. Xiao, Z. Gimbutas, A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions, Computers and Mathematics with Applications 59 (2) (2010) 663–676. doi:10.1016/j.camwa.2009.10.027.
[66] B. Vioreanu, V. Rokhlin, Spectra of Multiplication Operators as a Numerical Tool, SIAM Journal on Scientific Computing 36 (2014) 267–288. doi:10.1137/110860082.
[67] D. Williams, L. Shunn, A. Jameson, Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements, Journal of Computational and Applied Mathematics 266 (2014) 18–38. doi:10.1016/j.cam.2014.01.007.
[68] G. Cowper, Gaussian quadrature formulas for triangles, Numerical Methods in Engineering 7(3) (1973) 405–408. doi:10.1002/nme.1620070316.
[69] J. Lyness, D. Jespersen, Moderate degree symmetric quadrature rules for the triangle, IMA Journal of Applied Mathematics 15 (1) (1975) 19–32. doi:10.1093/imamat/15.1.19.
[70] F. Witherden, P. Vincent, On the identification of symmetric quadrature rules for finite element methods, Computers and Mathematics with Applications 69 (2015) 1232–1241. doi:10.1016/j.camwa.2015.03.017.
[71] S.-A. Papanicolopulos, Computation of moderate-degree fully symmetric cubature rules on the triangle using symmetric polynomials and algebraic solving, Computer and Mathematics with Applications 69 (2015) 650–666. doi:10.1016/j.camwa.2015.02.014.
[72] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.
[73] S. Gottlieb, C. Shu, Total Variation Diminishing Runge-Kutta Schemes, Mathematics of Computation 67 (221) (1998) 7385. doi:10.1090/S0025-5718-98-00913-2.
[74] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong Stability-Preserving High-Order Time Discretization Methods, SIAM Review 43 (1) (2001) 89–112. doi:10.1137/S003614450036757X.
[75] N. Chalmers, L. Krivodonova, A robust cfl condition for the discontinuous galerkin method on triangular meshes, Journal of Computational Physics 403. doi:10.1016/j.jcp.2019.109095.
[76] https://how5.cenaero.be/. [link].
URL https://how5.cenaero.be/
[77] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (2) (1981) 357–372. doi:10.1016/0021-9991(81)90128-5.
[78] S. Wandzura, H. Xiao, Symmetric quadrature rules on a triangle, Computers and Mathematics with Applications 45 (2003) 1829–1840. doi:10.1016/S0898-1221(03)90004-6.
[79] R. C. Swanson, S. Langer, Comparison of NACA 0012 Laminar Flow Solutions: Structured and Unstructured Grid Methods, Technical Memorandum NASA-TM-2016-219003, NASA Langley Research Center.
[80] Y. Sun, Z.J. Wang, Y. Liu, Efficient Implicit Non-linear LU-SGS Approach for Compressible Flow Computation Using High-Order Spectral Difference Method, Communications in Computational Physics 5 (2-4) (2009) 760–778. doi:10.2514/6.2007-4322.
[81] S. C. R. Dennis, G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, Journal of Fluid Mechanics 42 (3) (1970) 471–489. doi:10.1017/s0022112070001428.
[82] D. Russell, Z. Jane Wang, A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, Journal of Computational Physics 191 (1) (2003) 177–205. doi:10.1016/S0021-9991(03)00310-3.
[83] D. Calhoun, A Cartesian Grid Method for Solving the Two-Dimensional Streamfunction-Vorticity Equations in Irregular Regions, Journal of Computational Physics 176 (2) (2002) 231–275. doi:10.1006/jcph.2001.6970.
[84] P. Raviart, J. Thomas, A mixed finite element method for 2nd order elliptic problem, in: Lecture Notes in Mathematics 606, 1977, p. 292315.
[85] J. Nedelec, Mixed finite elements in R3, Numerische Mathematik 35 (1980) 315–341.