



HAL
open science

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

► **To cite this version:**

Adèle Veilleux, Guillaume Puigt, Hugues Deniau, Guillaume Daviller. A stable Spectral Difference approach for computations with triangular and hybrid grids up to the 6th order of accuracy. 2020. hal-02933792v1

HAL Id: hal-02933792

<https://hal.science/hal-02933792v1>

Preprint submitted on 8 Sep 2020 (v1), last revised 5 Jan 2022 (v3)

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

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

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

Adèle Veilleux^{*,1,2}, Guillaume Puigt^{†,1}, Hugues Deniau^{‡,1}, and Guillaume Daviller^{§,2}

¹ONERA/DMPE, Université de Toulouse, F-31055 Toulouse, France

²Centre 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. Compared to the literature on the Spectral Difference approach, the present work outperforms published results in the order of accuracy that the stable formulation can deal with ($p = 4$ and $p = 5$). Moreover, the present approach differs with today reference approach called Flux Reconstruction method on hybrid grids. The proposed scheme is based on a set of flux points that are defined in the paper. The sets of point leading to a stable formulation are determined using a linear stability analysis. Two techniques are introduced, based on an optimization process or using published cubature points. For each order of accuracy, different sets of points lead to stable Spectral Difference schemes using Raviart-Thomas elements. Validation starts from the linear advection equation and ends with the laminar subsonic and transonic Navier-Stokes solutions over the NACA0012 airfoil using high order triangles and the laminar flow around a cylinder using a hybrid grid.

Keywords: high-order method, Spectral Difference method, triangle, quadrangle, 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 any mesh cell, but without requiring the solution to be continuous across mesh interfaces. The most popular approach, the Discontinuous Galerkin (DG) method, has been successfully implemented in many solvers and leads to a very rich research. Without being exhaustive, a partial literature review focused on Computational Fluid Dynamics is available in several books [1, 2, 3, 4, 5, 6, 7, 8] and many contributions in Europe also come from projects [9, 10, 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 . Indeed, dealing with the strong formulation 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 with the aim to recover these lost properties. The lifting operator plays a central role in the

*Corresponding author, Adele.Veilleux@onera.fr

†Guillaume.Puigt@onera.fr

‡Hugues.Deniau@onera.fr

§Guillaume.Daviller@cerfacs.fr

properties of the schemes and enables to link the FR method, the DG formulation and other methods [15, 16]. A class of lifting operator can be built especially for specific mathematical properties, such as energy stability [17, 18]. Huynh, Wang and Vincent published in 2014 a reference paper on the proposed techniques [19].

An alternative method named the staggered-grid Chebyshev multidomain method was initiated by Kopriva and Kalias [20] in 1996 and applied to structured quadrilateral grids using a tensor-product framework by Kopriva in [21]. In 2006, Liu *et al.* [22] proposed an extension of Kopriva and Kalias' work to simplex cells and called the approach the Spectral Difference (SD) method. Wang *et al.* [23] adapted the procedure to Euler equations on triangular grids. Then, the method was extended to Navier-Stokes equations by May and Jameson [24] for triangular meshes and Sun *et al.* [25] 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 [20]. 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.* [26] 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 stability of the SD method for tensor product cells was studied by Van den Abeele *et al.* [27, 28]. They showed that the SP position did not influence neither the stability nor the accuracy of the scheme. Jameson confirmed this statement [29] 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.* [28] 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 [30, 31, 22, 32, 23] among the possible literature), most researchers focused on unstructured grids composed of hexahedra only. To overcome this limitation, Liang *et al.* [33] proposed to decompose any triangular or quadrangular cell into quadrangles using cell center and mid-edges, leading to quadrangle 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 quadrangles only but the number of mesh elements is strongly increased. Balan *et al.* proposed another alternative in [34, 35]. Instead of splitting any mesh cell in 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 4th order under a Fourier stability analysis originally initiated by May [36] and validated on Euler test cases. The SDRT method was then extended to simulate 2D viscous flows on unstructured hybrid grids up to the 4th order in [37]. Finally, one must also mention the work of Meister *et al.* on the SD method on triangles based on Proriot-Koornwinder-Dubiner (PKD) basis on the triangle for both solution and flux polynomials [38, 39]. Such approach will also differ with ours by the set of FP: here, the authors choose the set of Lobatto points on the triangle, as proposed by Blyth and Pozrikidis [40]. In the later 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 our 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) [41] because of its accuracy [42] and its efficiency [43] for Large Eddy Simulations. The SD method was recently made compatible with the non reflecting boundary conditions [44], written specifically to cope with the SD algorithm and then coupled with a Time Domain Impedance Boundary Condition formulation [45, 46]. 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 (quadrangles and triangles). In Sec. 2, the SDRT scheme on triangles and its difference with the standard technique on quadrangles are highlighted. Then, our procedure to find a linearly stable formulation on triangles is introduced in Sec. 3. Validation test cases are presented in Sec. 4, starting from the linear advection to simulations of 2D viscous flows on high-order-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:

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

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 quadrangles) and the i -th element is denoted Ω_i :

$$\Omega = \bigcup_{i=1}^N \Omega_i. \quad (2)$$

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 quadrangle. The transformation can be written as:

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

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:

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

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

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

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

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

where $\hat{\nabla}$ is the differential operator in the reference domain and \hat{u} , $\hat{\mathbf{f}}$ are the solution and the flux in the reference domain defined by:

$$\hat{u} = |J|u \quad (7)$$

and

$$\hat{\mathbf{f}} = |J|J^{-1}\mathbf{f}. \quad (8)$$

2.2 SD scheme on quadrangles

For quadrangles, the standard SD method follows a tensorial rule approach and the treatment is performed direction per direction, as in [25, 47, 28]. For a polynomial of degree p leading to an accuracy of $p+1$, a number $N_{SP} = p+1$ of SP (denoted $\xi_j, j \in \llbracket 1, N_{SP} \rrbracket$) are defined as the Gauss-Chebyshev points in the reference domain $[0, 1]$:

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

A number of $N_{FP} = p+2$ FP (denoted $\xi_k, k \in \llbracket 1, N_{FP} \rrbracket$) 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 . Solution and flux polynomials are finally computed using the standard Lagrange polynomials based either on the SP or on 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 RT polynomial space, as in [34, 35]. In order 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 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 $\boldsymbol{\xi}_j, j \in \llbracket 1, N_{SP} \rrbracket$ where

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

and

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

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:

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

where l_j is a Lagrange polynomial and \hat{u}_j are the known solution values 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 [48], a solution is to expand the polynomial \hat{u}_h using a modal representation:

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

where $\Phi_m(\boldsymbol{\xi}) \in \mathbb{P}_p$ is a complete polynomial basis and \tilde{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 \tilde{u}_m . Then, by performing a collocation projection at the points $\boldsymbol{\xi}_j$ such that

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

the coefficients \tilde{u}_m can be determined as:

$$\tilde{u}_m = \sum_{j=1}^{N_{SP}} \hat{u}_j (\Phi_m(\boldsymbol{\xi}_j))^{-1}. \quad (15)$$

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 $\{1, x, y, x^2, xy, y^2, \dots, y^p\}$. 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 has been defined on the triangle by Prorior [49], Koornwinder [50] and Dubiner [51]. For a polynomial approximation of degree p on the reference triangle, the 2D orthonormal PKD basis takes the following form:

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

Details on Jacobi polynomials and the PKD basis normalization can be found in B.

For simplicity, the subscript (i, j) can be replaced by the single index m , $m \in \llbracket 1, N_{SP} \rrbracket$ with any arbitrary bijection $m \equiv m(i, j)$.

From the literature [48, 52], three main assets of the PKD basis can be noted. First, it is based on Jacobi polynomials, which can be evaluated to high degree using simple recurrence relations. Then, the PKD L^2 orthogonality will 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:

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

2.3.2 Solution computation at flux points

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

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

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

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:

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

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

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

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

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

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 to 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.

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

where $(\hat{\mathbf{f}}_k \cdot \hat{\mathbf{n}}_k)^*$ 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:

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

The term $(\hat{\nabla} \cdot \psi_k)(\xi_j)$ in Eq. (23) can be written as a matrix of size $[N_{SP} \times N_{FP}]$ called differentiation matrix. To properly define the differentiation matrix, the vector-valued interpolation basis functions ψ_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 ϕ_n , $n \in \llbracket 1, N_{FP} \rrbracket$ as a linear combination of the basis functions ψ_k :

$$\phi_n(\xi) = \sum_{k=1}^{N_{FP}} a_{n,k} \psi_k(\xi) \quad (24)$$

For instance, in the case of $p = 1$, the monomial basis takes the form:

$$\phi = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix}, \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix}. \quad (25)$$

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), we get:

$$\phi_n(\xi) \cdot \hat{\mathbf{n}}_k = \sum_{k=1}^{N_{FP}} a_{n,k} \psi_k(\xi) \cdot \hat{\mathbf{n}}_k \quad (26)$$

and

$$\phi_n(\xi_k) \cdot \hat{\mathbf{n}}_k = \sum_{k=1}^{N_{FP}} a_{n,k} \psi_k(\xi_k) \cdot \hat{\mathbf{n}}_k \quad (27)$$

so

$$a_{n,k} = \phi_n(\xi_k) \cdot \hat{\mathbf{n}}_k. \quad (28)$$

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

$$\hat{\nabla} \cdot \phi_n(\xi) = \sum_{k=1}^{N_{FP}} a_{n,k} (\hat{\nabla} \cdot \psi_k)(\xi) \quad (29)$$

and therefore

$$(\hat{\nabla} \cdot \psi_k)(\xi) = (a_{n,k})^{-1} \hat{\nabla} \cdot \phi_n(\xi). \quad (30)$$

Eventually, the differentiation matrix at SP is computed as:

$$\mathbf{D}_{jk} = [(\hat{\nabla} \cdot \psi_k)(\xi_j)] = [(\phi_n(\xi_k) \cdot \hat{\mathbf{n}}_k)^{-1} \hat{\nabla} \cdot \phi_n(\xi_j)]. \quad (31)$$

2.3.5 Time increment

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

$$\frac{d\hat{u}_j^{(i)}}{dt} + \sum_{k=1}^{N_{FP}} \hat{f}_k^{(i)} (\hat{\nabla} \cdot \psi_k)(\xi_j) = 0, \quad j \in \llbracket 1, N_{SP} \rrbracket, \quad i \in \llbracket 1, N \rrbracket \quad (32)$$

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 quadrangles and triangles, the position of the FP on triangles must follow the rule for quadrangles: 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 to 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. But an open question concerns the possible differences between the proposed technique and the FR/CPR scheme. In order to ease the explanations, we consider the FR/CPR method described in [53]:

- For $p = 2$: three FP per face are introduced for the lifting operator and the very same number of FP is introduced for the SDRT. Moreover, the full flux polynomial definition for the FR/CPR involves the flux computed in 6 SP located inside the element and the 9 fluxes on the boundary. Here, we have 9 fluxes on the boundary and $p(p+1) = 6$ fluxes from internal FP.

- For $p = 1$: two FP per face are introduced for the lifting operator, as for the SDRT method. Moreover, the flux polynomial before correction needs the flux computed at the three SP inside the element. With the SDRT method, only $p(p + 1) = 2$ internal FP are needed.
- More generally, SDRT and FR/CPR methods will differ once the number of SP inside the element is not equal to $p(p + 1)$, so:

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

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 [29]. 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:

$$\frac{\partial \hat{u}(\boldsymbol{\xi}, t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0 \quad (33)$$

except that now, the flux is defined by :

$$\hat{\mathbf{f}} = |J|J^{-1}\mathbf{f}(u, \nabla u), \quad (34)$$

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

$$\mathbf{f} = \mathbf{f}^i(u) - \mathbf{f}^v(u, \nabla u) \quad (35)$$

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. (33) 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:

$$\hat{f}_k = \hat{f}_k^i - \hat{f}_k^v \quad (36)$$

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

$$\hat{f}_k^i = \begin{cases} \hat{\mathbf{f}}^i \cdot \hat{\mathbf{n}}_k = |J|J^{-1}\mathbf{f}_k^i(u_h(\boldsymbol{\xi}_k)) \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \\ (\hat{\mathbf{f}}_k^i \cdot \hat{\mathbf{n}}_k)^* = (\mathbf{f}_k^i \cdot |J|(J^{-1})^\top \hat{\mathbf{n}}_k)^*, & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \quad (37)$$

In order to compute \hat{f}_k^v , which relies on the solution and its gradient, the following procedure, based on a centered formulation [25] is used.

From the approximated solution in the reference domain, the physical approximated solution $u_h(\boldsymbol{\xi}_k)$ is first computed at FP:

$$u_h(\boldsymbol{\xi}_k) = \frac{1}{|J|}\hat{u}_h(\boldsymbol{\xi}_k) = \frac{1}{|J|}\mathbf{T}_{kj}\hat{u}_j \quad (38)$$

From those values, a polynomial of degree $p + 1$ can be reconstructed for the solution. However, this polynomial is discontinuous at cell interfaces. A centered scheme is used to define a single values at each flux point by averaging the values from the left and the right cells, leading to a continuous polynomial approximation u_h^c .

$$u_h^c(\boldsymbol{\xi}_k) = \begin{cases} u_h(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \\ \frac{1}{2}(u_h^L(\boldsymbol{\xi}_k) + u_h^R(\boldsymbol{\xi}_k)), & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \quad (39)$$

Eq. (39) is then differentiated to compute the solution gradient (∇u) at SP.

$$(\nabla u)(\boldsymbol{\xi}_j) = \frac{1}{|J|}\mathbf{D}_{jk}\left(u_h^c(\boldsymbol{\xi}_k)(|J|J^{-1})^\top \hat{\mathbf{n}}_k\right) \quad (40)$$

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

$$(\nabla u)_h(\boldsymbol{\xi}_k) = \mathbf{T}_{kj}(\nabla u)(\boldsymbol{\xi}_j) \quad (41)$$

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

$$(\nabla u)_h^c(\boldsymbol{\xi}_k) = \begin{cases} (\nabla u)_h(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \\ \frac{1}{2} \left((\nabla u)_h^L(\boldsymbol{\xi}_k) + (\nabla u)_h^R(\boldsymbol{\xi}_k) \right), & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \quad (42)$$

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:

$$\mathbf{f}_k^v = \mathbf{f}^v(u_h^c(\boldsymbol{\xi}_k), (\nabla u)_h^c(\boldsymbol{\xi}_k)) \quad (43)$$

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

$$\hat{f}_k^v = \begin{cases} |J|J^{-1}\mathbf{f}_k^v \cdot \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \mathcal{T} \setminus \partial\mathcal{T} \\ \mathbf{f}_k^v \cdot |J|(J^{-1})^\top \hat{\mathbf{n}}_k, & \boldsymbol{\xi}_k \in \partial\mathcal{T}. \end{cases} \quad (44)$$

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

The stability analysis presented in this section follows the eigenvalue analysis provided in [54, 48, 42]: in order to study the scheme stability, the (coupled) space and time discretizations are written under a matrix form. The eigenvalue analysis is performed using a coarse grid taking into account the block-diagonal structure identified in [54, 48].

3.1 Matrix form of the SDRT scheme

We consider the linear advection equation

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} + \nabla \cdot \mathbf{f} = 0, \quad \text{in } \Omega \times [0, t_f] \quad (45)$$

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:

$$\mathbf{c} = (c_x, c_y) = (\cos \theta, \sin \theta), \quad \theta \in [0, 2\pi] \quad (46)$$

The domain Ω is a regular mesh composed of two triangles T_1 and T_2 (Fig. 1) with periodic boundary conditions. Using symmetry properties, the choice of the advection angle can be reduced to $\theta \in [0, \pi/4]$. 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 ($\eta = 0$), $k \in \llbracket 1, N_e \rrbracket$, k increasing with ξ , $\hat{\mathbf{n}} = (0, -1)^\top$
- On face 2 ($\eta = 1 - \xi$), $k \in \llbracket N_e + 1, 2N_e \rrbracket$, k increasing with η , $\hat{\mathbf{n}} = (1, 1)^\top$
- On face 3 ($\xi = 0$), $k \in \llbracket 2N_e + 1, 3N_e \rrbracket$, k increasing when η decreases, $\hat{\mathbf{n}} = (-1, 0)^\top$

The remaining $N_i = p(p+1)$ flux nodes are located in the interior and represented with blue squares. As mentionned in Sec. 2, one physical point is considered as two separate degrees of freedom with different normal vectors, thus there are $N_i/2$ physical FP. Flux nodes associated with the unit vector $\hat{\mathbf{n}} = (1, 0)^\top$ in the reference element are numbered with $k \in \llbracket 3N_e + 1, 3N_e + 1 + N_i/2 \rrbracket$ whereas flux nodes whose unit vector is $\hat{\mathbf{n}} = (0, 1)^\top$ are numbered with $k \in \llbracket 3N_e + 1 + N_i/2, 3N_e + 1 + N_i \rrbracket$. An example of the FP numbering and their associated normal vector is given on Fig. 2 for the case $p = 2$.

3.2 Linear stability analysis for the spatial discretization

3.2.1 Matrix form of the SDRT scheme

Each element is transformed into a standard element and Eq. (45) becomes:

$$\frac{\partial \hat{u}(\boldsymbol{\xi}, t)}{\partial t} + \hat{\nabla} \cdot \hat{\mathbf{f}} = 0, \quad (47)$$

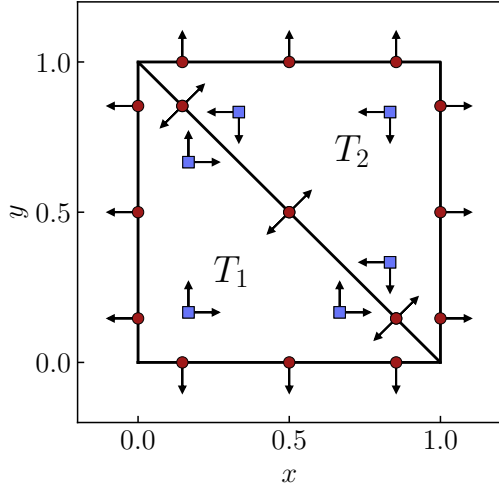


Figure 1: Computational domain for the linear stability analysis - Example of FP distribution for $p = 2$

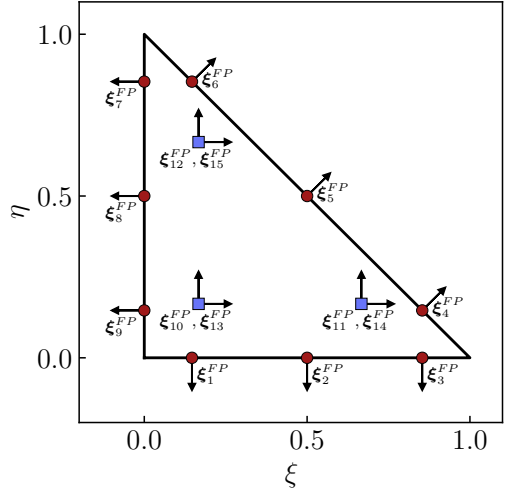


Figure 2: Flux points numbering in the reference element - Example of FP distribution for $p = 2$

with $\hat{u} = |J|u$ and $\hat{\mathbf{f}} = |J|J^{-1}\mathbf{f}$.

We introduce $\hat{\mathbf{U}}_j^i$, the column vector of size N_{SP} whose components are the solution values in the reference element at SP in the i -th cell:

$$\hat{\mathbf{U}}_j^i = [\hat{u}_i(\boldsymbol{\xi}_j)]_{1 \leq j \leq N_{SP}}. \quad (48)$$

The solution values at FP are obtained by multiplying $\hat{\mathbf{U}}_j^i$ by the transfer matrix \mathbf{T}_{kj} , which represents the extrapolation step and is given as:

$$\mathbf{T}_{kj} = \sum_{m=1}^{N_{SP}} (\Phi_m(\boldsymbol{\xi}_j))^{-1} \Phi_m(\boldsymbol{\xi}_k), \quad (49)$$

leading to the column vector $\hat{\mathbf{V}}_k^i$ of size N_{FP} :

$$\hat{\mathbf{V}}_k^i = [\hat{u}_i(\boldsymbol{\xi}_k)]_{1 \leq k \leq N_{FP}} = \mathbf{T}_{kj} \hat{\mathbf{U}}_j^i. \quad (50)$$

From the solution values at FP, the flux values are then computed. At edges, two different solution values are available at the same point. An upwind Godunov scheme is used as the numerical flux to ensure the flux continuity. The flux scalar values read:

$$\hat{f}_k = \begin{cases} |J|J^{-1}(\mathbf{c} \cdot \hat{\mathbf{n}}_k^i) u_i(\boldsymbol{\xi}_k), & \boldsymbol{\xi}_k \in \Omega \setminus \partial\Omega \\ (\mathbf{c} \cdot \mathbf{n}_k^i) \left(\frac{1 + \text{sign}(\mathbf{c} \cdot \mathbf{n}_k^i)}{2} u_i(\boldsymbol{\xi}_k) + \frac{1 - \text{sign}(\mathbf{c} \cdot \mathbf{n}_k^i)}{2} u_{nei}(\boldsymbol{\xi}_k) \right), & \boldsymbol{\xi}_k \in \partial\Omega, \end{cases} \quad (51)$$

where $u_{nei}(\boldsymbol{\xi}_k)$ is the solution value coming from the neighboring cell. Given that the flux is linear, the flux computation can be expressed as a matrix product vector between a velocity matrix denoted \mathbf{C} and the vector containing the solution values at FP $\hat{\mathbf{V}}_k^i$:

$$\hat{\mathbf{F}}_k^i = \mathbf{C} \cdot \hat{\mathbf{V}}_k^i \quad (52)$$

The velocity matrix \mathbf{C} should verify Eq. (51) while taking into account the mesh connectivity and the periodic boundary conditions. Following the FP numbering previously settled, it can be expressed as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^L & O_{3N_e, N_i} & \mathbf{C}^R & O_{3N_e, N_i} \\ O_{N_i, 3N_e} & \mathbf{C}_{N_i}^I & O_{N_i, 3N_e} & O_{N_i, N_i} \\ \mathbf{C}^R & O_{3N_e, N_i} & \mathbf{C}^L & O_{3N_e, N_i} \\ O_{N_i, 3N_e} & O_{N_i, N_i} & O_{N_i, 3N_e} & \mathbf{C}_{N_i}^I \end{bmatrix} \quad (53)$$

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

$$\mathbf{C}^I = [\text{diag}(|J|J^{-1}(\mathbf{c} \cdot \hat{\mathbf{n}}))]_{N_i, N_i} \quad (54)$$

$$\mathbf{C}^L = (\mathbf{c} \cdot \mathbf{n}) \left[\text{diag}\left(\frac{1 + \text{sign}(\mathbf{c} \cdot \mathbf{n})}{2}\right) \right]_{3N_e, 3N_e} \quad (55)$$

and

$$\mathbf{C}^R = (\mathbf{c} \cdot \mathbf{n}) \left[\text{bdiag} \left(\begin{bmatrix} 0 & \cdots & \frac{1-\text{sign}(\mathbf{c} \cdot \mathbf{n})}{2} \\ \vdots & \ddots & \vdots \\ \frac{1-\text{sign}(\mathbf{c} \cdot \mathbf{n})}{2} & \cdots & 0 \end{bmatrix}_{N_e, N_e} \right) \right]_{3,3} \quad (56)$$

where, when applied to an arbitrary square matrix \mathbf{A} , the operator $\text{bdiag}()$ gives a block diagonal matrix such that the main-diagonal blocks are the square matrix \mathbf{A} and all off-diagonal blocks are zeros matrices. Eventually, the flux values at FP are differentiated at SP by multiplying by the differentiation matrix given as:

$$\mathbf{D}_{jk} = \sum_{n=1}^{N_{FP}^{tri}} (\phi_n(\boldsymbol{\xi}_k) \cdot \hat{\mathbf{n}}_k)^{-1} \hat{\mathbf{v}} \cdot \phi_n(\boldsymbol{\xi}_j) \quad (57)$$

On the computational domain $\Omega = (T_1, T_2)$, Eq. (45) takes the following matrix form:

$$\begin{bmatrix} \frac{\partial \hat{\mathbf{U}}_j^{T_1}(t)}{\partial t} \\ \frac{\partial \hat{\mathbf{U}}_j^{T_2}(t)}{\partial t} \end{bmatrix} = - \begin{bmatrix} \mathbf{D}_{jk} & 0 \\ 0 & \mathbf{D}_{jk} \end{bmatrix} \mathbf{C} \begin{bmatrix} \mathbf{T}_{kj} & 0 \\ 0 & \mathbf{T}_{kj} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{U}}_j^{T_1}(t) \\ \hat{\mathbf{U}}_j^{T_2}(t) \end{bmatrix}, \quad j \in \llbracket 1, N_{SP}^{tri} \rrbracket, k \in \llbracket 1, N_{FP}^{tri} \rrbracket \quad (58)$$

where $\hat{\mathbf{U}}_j^{T_1}$ and $\hat{\mathbf{U}}_j^{T_2}$ are the vectors collecting the solution values at SP for the triangle T_1 and T_2 . Now that all matrices are defined on the computational domain, the stability of the scheme can be studied through the matrix $\mathbf{M} = -\text{diag}(\mathbf{D}) \mathbf{C} \text{diag}(\mathbf{T})$. Using Eq. (58), the linear stability of the spatial discretization can be investigated. The spatial discretization is fully expressed through the square matrix \mathbf{M} , which contains the different steps of the SDRT scheme: extrapolation, flux computation and derivation. This matrix can thus be seen as the amplification factor of the spatial discretization. It must be highlighted here that the matrix \mathbf{M} does not depend on the initial solution. Consequently, if the real part of all \mathbf{M} 's eigenvalues are negatives, the linear stability of the SDRT scheme is ensured regardless of the initial solution. According to Eq. (58), the matrix \mathbf{M} depends on the transfer matrix \mathbf{T} , the velocity matrix \mathbf{C} and the differentiation matrix \mathbf{D} :

- The **transfer matrix** is expressed from the polynomial basis (PKD basis for simplex cells) at SP and FP locations.
- The **velocity matrix** relies on the velocity vector \mathbf{c} and the normal vector defined at each flux point.
- The **differentiation matrix** involves the polynomial basis in which the flux is differentiated (RT basis for simplex cells) at SP and FP locations as well as on the normal vector defined at each flux point.

The velocity vector components values are set through the definition of the advection angle θ . The linear stability will be studied for several values of the advection angle. As explained in Sec. 2, the PKD basis and the RT basis are used respectively for the extrapolation and the differentiation. Those polynomial basis rely on the SP and FP sets of points and on the normal vector associated to FP. Since it has been shown by Van den Abeele *et al.* [28] that the SD scheme stability is independent of the SP position, our 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 has been shown by Van den Abeele [55] that if the FP are chosen as the Chebyshev-Gauss-Lobatto nodes, the standard 1D SD scheme can be unstable. Following this work, Jameson [29] 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 has been observed in [34] that the placement of FP on edges does not affect the linear stability properties for second to fourth order accurate SDRT schemes. In order to simplify the 2D hybrid implementation, we decide to settle the position of FP located on the edge to the Gauss-Chebyshev points given by Eq. (9). By doing so, the FP on edges for a quadrangle and a triangle are located at the exact same coordinates, avoiding the need to apply mortar techniques. Since the edge FP position is chosen to be fixed, the interior FP location is the parameter that we will study here in order to establish stable SDRT discretization for all advection angles.

3.3 Establishment of stable SDRT discretizations

In order to determine a set of interior FP leading to a stable SDRT_p formulation for each degree p , the interior FP position inside the element first needs to be set up. Then, a differential evolution algorithm is used to minimize the real part of \mathbf{M} 's eigenvalues. To set up the interior FP position, a symmetry along $y = x$ is imposed in the triangle. The position of the interior FP located on this line is defined by a parameter $\alpha_i \in [0, 0.5]$. To set up the position of the remaining interior FP, two parameters

$(\beta_i, \gamma_i) \in [0, 1] \times [0, \beta_i/2]$ per points are needed. The parameter β_i is used to browse the triangle from 0 to $y = -x + 1$ using $y = -x + \beta_i$. The parameter γ_i is used to adjust the position of the point on the line $y = -x + \beta_i$. Because of the symmetry rule, one set (β_i, γ_i) gives two FP located at $(\beta_i/2 + \gamma_i, \beta_i/2 - \gamma_i)$ and $(\beta_i/2 - \gamma_i, \beta_i/2 + \gamma_i)$, thus the number of parameters in the optimization problem is equal to $N_i/2$. An example of the set up for $p = 4$ is given on Fig. 3.

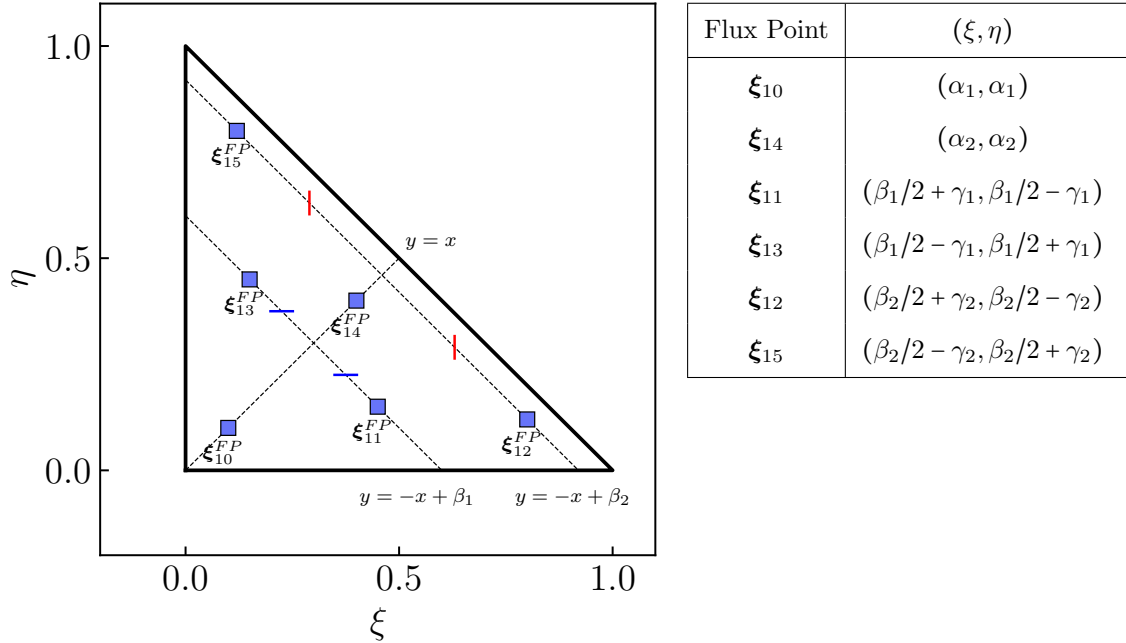


Figure 3: Interior FP set up in the reference element - Example for SDRT₄

The differential evolution algorithm is run for SDRT schemes up to the sixth order. For each scheme, several sets of interior FP lead to stable formulations. This result shows that the definition of a stable SDRT formulation is not unique. One of the possible sets of coefficients to built the position of the interior FP is given in Tab. 1 for the new SDRT₄ and SDRT₅ stable formulations. They are plotted in the reference domain on Fig. 4.

SDRT ₄		
α_1	0.2049431286797799	—
α_2	0.0685150418264187	—
β_1, γ_1	0.4720289928855974	0.2003096571609239
β_2, γ_2	0.8340689076462250	0.2954293172666666
β_3, γ_3	0.9296725246067598	0.3496263405633941
β_4, γ_4	0.9340656356895457	0.2448926406908226
SDRT ₅		
α_1	0.3822665100426330	—
α_2	0.4550603088554616	—
α_3	0.0400736317911791	—
β_1, γ_1	0.9270536098373276	0.3116771986988879
β_2, γ_2	0.1982183938046974	0.0339405048273691
β_3, γ_3	0.6939139683747306	0.0819761640225505
β_4, γ_4	0.9573143940984170	0.4534034282622463
β_5, γ_5	0.8424456505829385	0.3345523352287995
β_6, γ_6	0.6305741301837382	0.2523643148948984

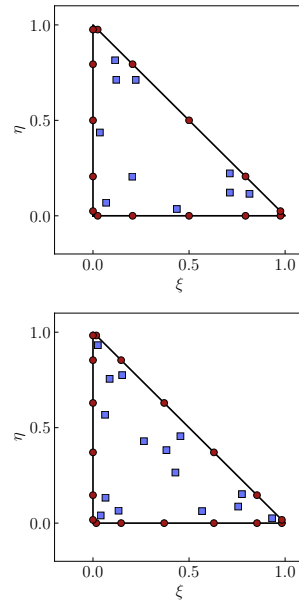


Table 1: Coordinates of stable interior FP for SDRT₄ and SDRT₅

Figure 4: Complete stable set of FP for SDRT₄ and SDRT₅

The parametrization process leads to stable formulations but the time for analysis is quite long. While the process was running, stability using the position of control points for several cubature rules was also

analyzed. For $p = 3$, we tested several existing cubatures rules for interior FP location and those from [56, 57, 58, 59, 60, 61, 62] were found stable.

For $p = 4$, the Williams-Shunn-Jameson [60] and the Vioreanu-Rokhlin [59] cubature rules are found to be stable while for $p = 5$, stable formulations are obtained using the position from Williams-Shunn-Jameson [60] and the Witherden-Vincent [61] cubature rules. This underlines once again the plurality of the possible stable SDRT formulations. The matrix \mathbf{M} spectrum of stable formulations from the cubatures rules and the parametrization process are given on Fig. 5. For the SDRT₂ and SDRT₃ schemes, considering the important number of stable cubatures, results are only presented for set of interior FP taken as the Williams-Shunn-Jameson cubature rule [60]. For each SDRT _{p} scheme, a closer view shows that all the real part of \mathbf{M} eigenvalues are negatives, thus ensuring the spatial discretization stability.

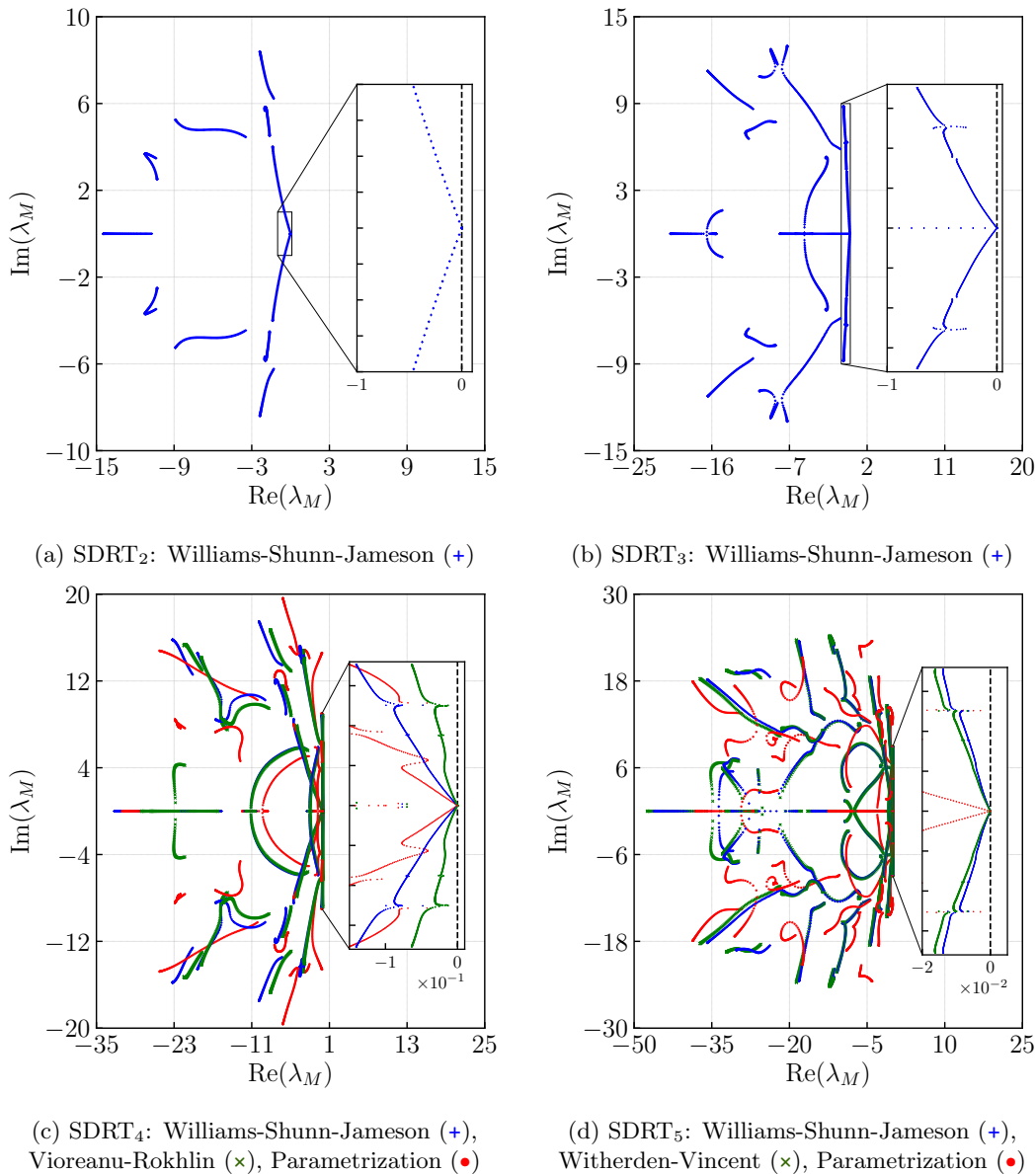


Figure 5: Spectrum of matrix \mathbf{M} for stable sets of FP for 3rd to 6th order SDRT schemes

In this paper, we chose to define the interior FP location as following the Williams-Shunn-Jameson cubatures rule. This choice is motivated by the fact that compared to other cubature rules, this one led to stable schemes for each scheme from $p = 2$ to $p = 5$. The Williams-Shunn-Jameson cubature rules is preferred to the location given by the parametrization process because of their better distribution in the triangle.

For a polynomial degree equal to 6 or higher, none of the tested cubature rules led to a stable SDRT scheme. When using the optimization process for $p = 6$, given the large number of interior FP, the parametrization becomes too expensive in time and did not manage to detect a stable formulation before the final time of the computation. This does not state the fact that stable formulations could not be obtained in the future.

3.4 CFL condition

Now that a stable SDRT spatial discretization has been established through the definition of a set of FP, the system of linear differential equations given by Eq. (58) can be time-integrated. Two temporal schemes are studied: the low-storage second-order six-stage Runge-Kutta scheme of Bogey and Bailly (RK6-ldld) [63] and the Total Variation Diminishing Runge-Kutta scheme of Gottlieb and Shu (RK3-GS) [64]. Using a RK6-ldld scheme, Eq. (58) becomes:

$$\hat{\mathbf{U}}^{n+1} = \left(\mathbf{I} + \sum_{l=1}^6 \gamma_l (\Delta t)^l \mathbf{M}^l \right) \hat{\mathbf{U}}^n \quad (59)$$

$$= \mathbf{G} \hat{\mathbf{U}}^n. \quad (60)$$

With the RK3-GS scheme, the temporal discretization reads:

$$\hat{\mathbf{U}}^1 = (\mathbf{I} + \Delta t \mathbf{M}) \hat{\mathbf{U}}^n \quad (61)$$

$$\hat{\mathbf{U}}^2 = \frac{3}{4} \hat{\mathbf{U}}^n + \frac{1}{4} (\mathbf{I} + \Delta t \mathbf{M}) \hat{\mathbf{U}}^1 \quad (62)$$

$$\hat{\mathbf{U}}^{n+1} = \frac{1}{3} \hat{\mathbf{U}}^n + \frac{2}{3} (\mathbf{I} + \Delta t \mathbf{M}) \hat{\mathbf{U}}^2, \quad (63)$$

and injecting Eq. 61 in Eq. 62 and then in Eq. 63, one obtains easily $\hat{\mathbf{U}}^{n+1} = \mathbf{G} \hat{\mathbf{U}}^n$. So, the transfer matrix between time steps n and $n + 1$, denoted \mathbf{G} , accounts for both space and time integration and depends on the CFL number ν since $\nu = \|\mathbf{c}\| \Delta t / \Delta x$ as well as on the spatial discretization matrix \mathbf{M} . To ensure a stable discretization, the spectral radius of the matrix \mathbf{G} should be lower than 1. The stability limits are summarized in Tab. 2 for several advection angles. Note that in order to compare the CFL numbers given in Tab. 2 with those from the Finite Volume or Finite Difference methods, the given values should be multiplied by an appropriated constant. Indeed, the maximum CFL values strongly depends on the reference length for a triangle and many options are possible (radius of the inscribed circle, minimum distance between triangle center and edge centers...).

	RK6-ldld			RK3-GS		
	$\theta = 0$	$\theta = \pi/8$	$\theta = \pi/4$	$\theta = 0$	$\theta = \pi/8$	$\theta = \pi/4$
SDRT _{<i>p</i>}						
SDRT ₂	0.380	0.308	0.286	0.221	0.186	0.173
SDRT ₃	0.280	0.220	0.200	0.165	0.127	0.116
SDRT ₄	0.162	0.137	0.128	0.098	0.082	0.077
SDRT ₅	0.132	0.101	0.089	0.076	0.061	0.054

Table 2: Maximum CFL number ν for SDRT_{*p*} schemes ($p \in [2, 5]$) with a reference scale $\Delta x = 1$.

Several spectrum of \mathbf{G} obtained using the RK6-ldld scheme are presented on Fig. 6 for the advection angle $\theta \in [0, \pi/4]$. Fig. 6a and Fig. 6c show the spectrum using the stable CFL number numerically determined for SDRT₄ and SDRT₅ schemes (respectively). A closer view shows that all the real part of \mathbf{G} eigenvalues are negatives, thus ensuring the scheme stability. Unstable spectrum can be obtained by slightly raising the CFL number and are presented on Fig. 6b and Fig. 6d. On the closer view, it is clear that positive real part of \mathbf{G} eigenvalues appear, showing the scheme instability. The same analysis can be found in C for SDRT₂ and SDRT₃ schemes.

4 Numerical Experiments

4.1 Linear Advection

Experiments on the 2D linear advection equation given by Eq. (45) are performed to assess numerically the order of accuracy of the SDRT scheme up to $p = 5$. The advection equation is solved on a square domain $\Omega = [-1, 1] \times [-1, 1]$ and periodic boundary conditions are used in both x and y directions. The initial solution is $u(x, y, 0) = \sin(\pi(x + y))$ and the advection velocity is $\mathbf{c} = (\cos(\frac{\pi}{8}), \sin(\frac{\pi}{8}))$. At interfaces, an upwind flux is used as the numerical flux. Eq. (45) is integrated in time using the low-storage second-order six-stage Runge-Kutta scheme previously introduced for the linear stability analysis and the simulation is carried out until $t = 1$ sec. 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^{-3} . The convergence study was conducted on three types of grids: two different triangular meshes (regular and irregular) and a regular hybrid one (Fig 7). To perform the convergence study, each mesh is refined several times. In order to verify the order of accuracy of the SDRT scheme,

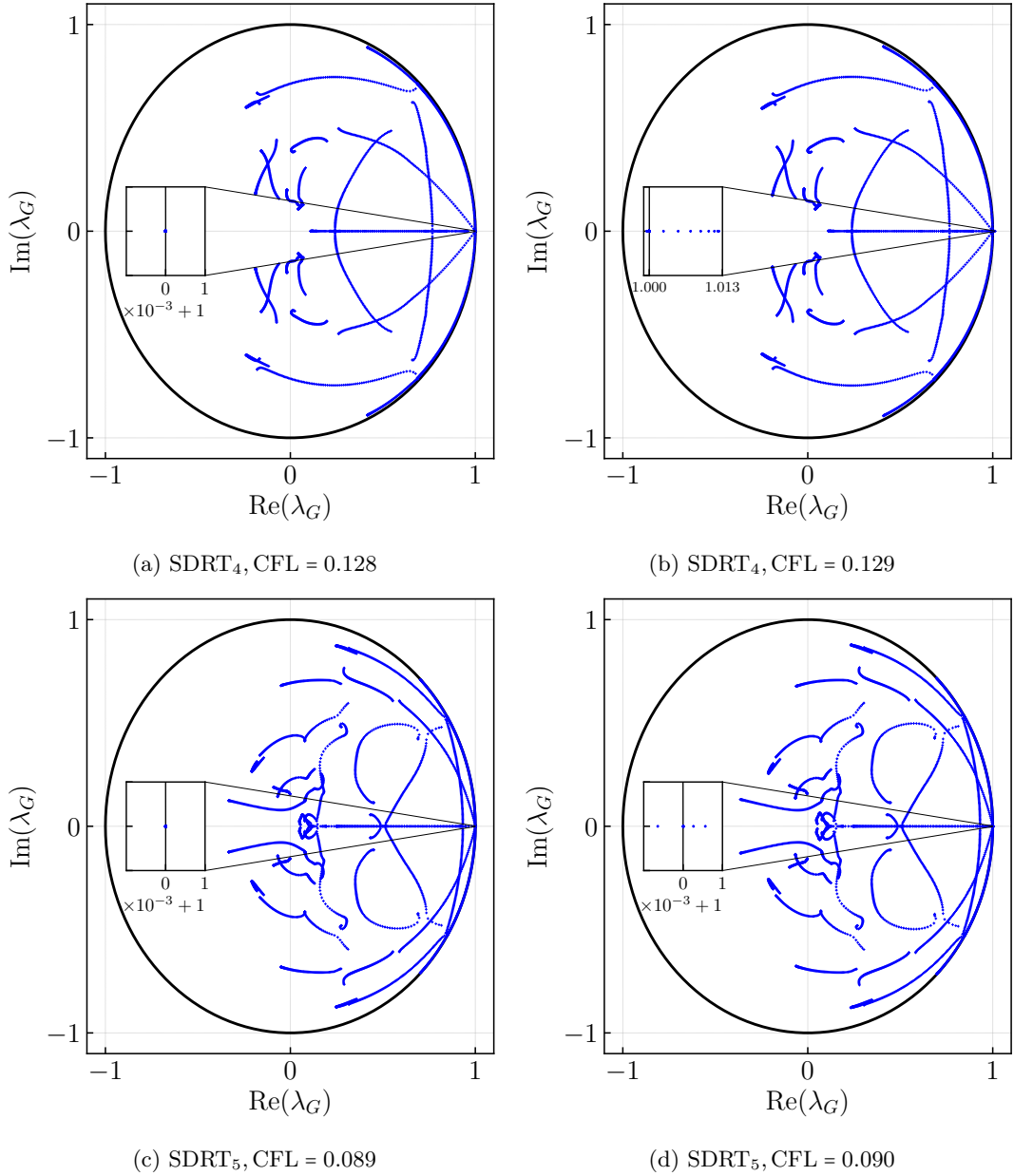


Figure 6: Spectrum of matrix \mathbf{G} using RK6-ldld scheme for 5^{rd} and 6^{th} order SDRT schemes

the L_2 error is computed on the domain as:

$$L_2 = \sqrt{\frac{\int_{\Omega} (u_a - u_{num})^2 dx}{\int_{\Omega} dx}} \quad (64)$$

In Eq. (64), the integral on the top can be expressed as the following sum on each cell:

$$\int_{\Omega} (u_a - u_{num})^2 dx = \sum_{i=1}^N \int_i (u_a^{(i)} - u_{num}^{(i)})^2 dx, \quad (65)$$

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:

$$\sum_{i=1}^N \int_i (u_a^{(i)} - u_{num}^{(i)})^2 dx = \sum_{i=1}^N \sum_{j=1}^{N_q} A |J^{(i,j)}| \omega_j (u_a^{(i)}(\boldsymbol{\xi}_j) - u_{num}^{(i)}(\boldsymbol{\xi}_j))^2, \quad (66)$$

where A is the reference element area ($A = 1$ and $A = 1/2$ for a quadrilateral and a triangular element, respectively), $|J^{(i,j)}|$ 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 to the weight ω_j . Since u_a and u_{num} are polynomials of degree p , the term $(u_a^{(i)} - u_{num}^{(i)})^2$ should be approximated

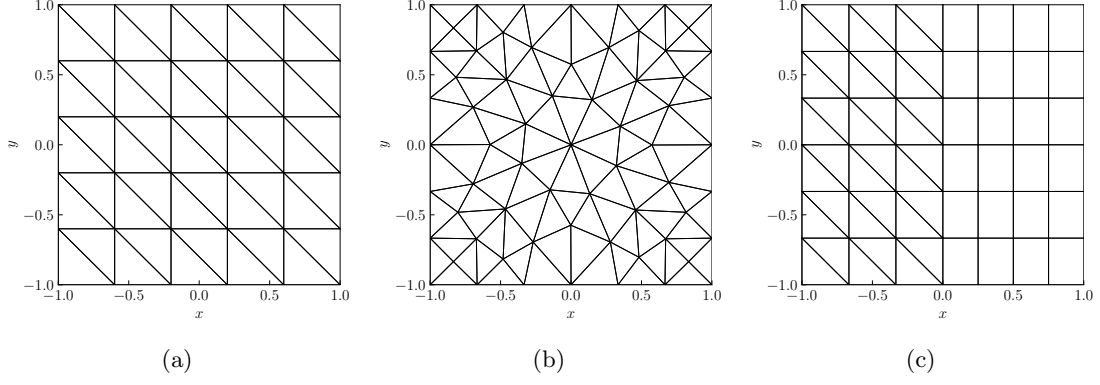


Figure 7: Regular (a), irregular (b) and hybrid (c) grids used for the convergence study of SDRT schemes

using a quadrature of order $2p$. On triangles, the integration is carried out using the 175-points symmetric quadrature given in [65], which can be used up to order 30. On quadrilaterals, the integration is performed using the tensor product of two 1D integration at SP, with the appropriate Gauss-Chebyshev weights. For SDRT $_p$ schemes ($p \in \llbracket 2, 5 \rrbracket$), Table 3 and Figure 8 show the L_2 errors and orders of accuracy for the three different type of grids. For any SDRT $_p$ scheme, the order of accuracy tends to $p+1$ for each family of meshes.

p	Regular mesh			Irregular mesh			Hybrif mesh		
	DoF number	L_2 error	Order of accuracy	DoF number	L_2 error	Order of accuracy	DoF number	L_2 error	Order of accuracy
2	300	7.725E-03	-	624	1.029E-02	-	432	6.253E-03	-
	1200	9.735E-04	2.99	2256	1.492E-03	3.01	1512	1.023E-03	2.89
	2700	2.874E-04	3.00	5232	4.364E-04	2.97	3564	2.695E-04	2.98
	4800	1.210E-04	3.00	9348	1.933E-04	2.94	6048	1.273E-04	2.95
3	500	5.762E-04	-	1040	9.979E-04	-	744	4.032E-04	-
	2000	3.788E-05	3.93	3760	7.155E-05	4.10	2592	3.383E-05	3.97
	4500	7.502E-06	3.95	8720	1.270E-05	4.10	6120	5.905E-06	4.01
	8000	2.372E-06	3.96	15580	4.399E-06	4.01	10368	2.148E-06	3.97
4	750	3.975E-05	-	1560	7.835E-05	-	1140	2.183E-05	-
	3000	1.339E-06	4.89	5640	2.822E-06	5.17	3960	9.165E-07	5.09
	6750	1.767E-07	4.93	13080	3.264E-07	5.16	9360	1.050E-07	5.07
	12000	4.087E-08	4.96	23370	9.315E-08	4.98	15840	2.933E-08	5.03
5	1050	2.296E-06	-	2184	6.057E-06	-	1620	9.961E-07	-
	4200	3.675E-08	5.97	7896	1.033E-07	6.34	5616	2.047E-08	6.25
	9450	3.221E-09	5.98	18312	7.297E-09	6.32	13284	1.517E-09	6.17
	16800	5.677E-10	5.99	32718	1.640E-09	6.07	22464	3.213E-10	6.11

Table 3: L_2 error and order of accuracy values for regular, irregular and hybrid grids

4.2 Convection of an isentropic vortex

In order to assess the SDRT scheme capability to preserve vorticity in an unsteady inviscid flow, the isentropic vortex test case proposed by Shu [66] is studied. An isentropic vortex is transported by an inviscid uniform flow defined by normalized quantities $(\rho_\infty, u_\infty, v_\infty, P_\infty) = (1, 1, 1, 1)$. The fluid is assumed to be a perfect gas, with a specific heat $\gamma = 1.4$. The isentropic vortex is added to this mean flow through perturbation in u , v and the temperature T given as:

$$\Delta u = -\frac{\varepsilon(y-5)}{2\pi} \exp(0.5(1-r^2)) \quad (67)$$

$$\Delta v = \frac{\varepsilon(x-5)}{2\pi} \exp(0.5(1-r^2)) \quad (68)$$

$$\Delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} \exp(1-r^2) \quad (69)$$

where $r = \sqrt{(x-5)^2 + (y-5)^2}$ and the vortex strength is $\varepsilon = 5$. Euler equations are solved on the computational domain $\Omega = [0, 10] \times [0, 10]$. Translational periodic boundary conditions are imposed for

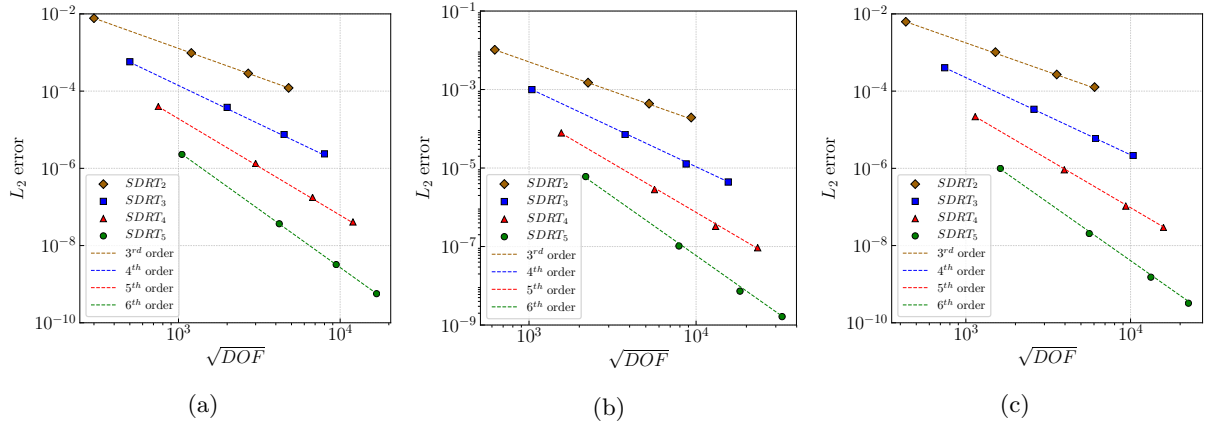


Figure 8: L_2 error and theoretical order of accuracy slopes for regular (a), irregular (b) and hybrid (c) grids used for the convergence study of SDRT schemes

the left/right and top/bottom boundaries respectively. The computation is initialized by the local velocity components u and v as well as temperature T :

$$u = u_\infty + \Delta u, \quad (70)$$

$$v = v_\infty + \Delta v, \quad (71)$$

$$T = T_\infty + \Delta T, \quad (72)$$

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

$$\rho = \rho_\infty \left(\frac{T}{T_\infty} \right)^{\frac{1}{\gamma-1}}, \quad (73)$$

where $\rho_\infty = 1$. The RK6-ldld scheme introduced for the space/time analysis is considered for the simulations. The CFL number is $0.1/(p+1)$ and the final time t_f corresponds to 10 periods. At interfaces, Roe's Riemann solver [67] was used to compute the numerical flux. The simulation is initialized and computed on a fully regular symmetric mesh. In order to make a fair comparison between the different SDRT $_p$ schemes, computations are performed on different mesh refinements so that the number of degrees of freedom (DoF) between orders is as equivalent as possible. Two set of grids are studied: a coarse one, to highlight the improvement in accuracy when the order increases, and a refined one, to show mesh convergence towards the analytical solution. The associated number of DoF are given on Table 4 for each SDRT $_p$ scheme. Fig. 9 gives an example of the initial solution on the coarse mesh used for the SDRT $_5$ scheme.

SDRT $_p$	Coarse Mesh	Refined Mesh
SDRT $_2$	4800	9408
SDRT $_3$	3920	9680
SDRT $_4$	3000	9720
SDRT $_5$	2688	8232

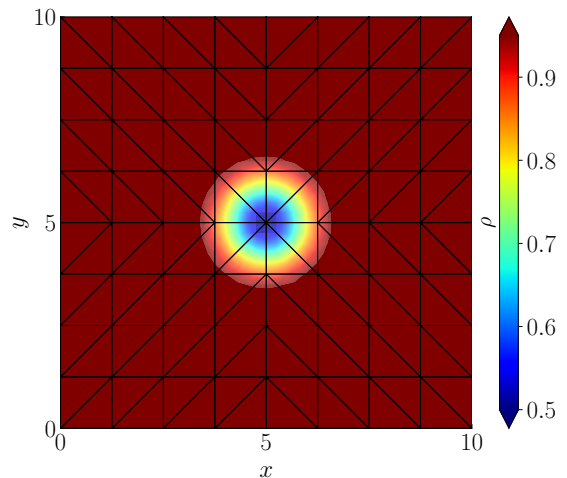


Table 4: Number of DoF for the coarse and the refined set of grids for 3rd to 6th order SDRT schemes

Figure 9: Initialization of the density on the coarse mesh used for SDRT $_5$

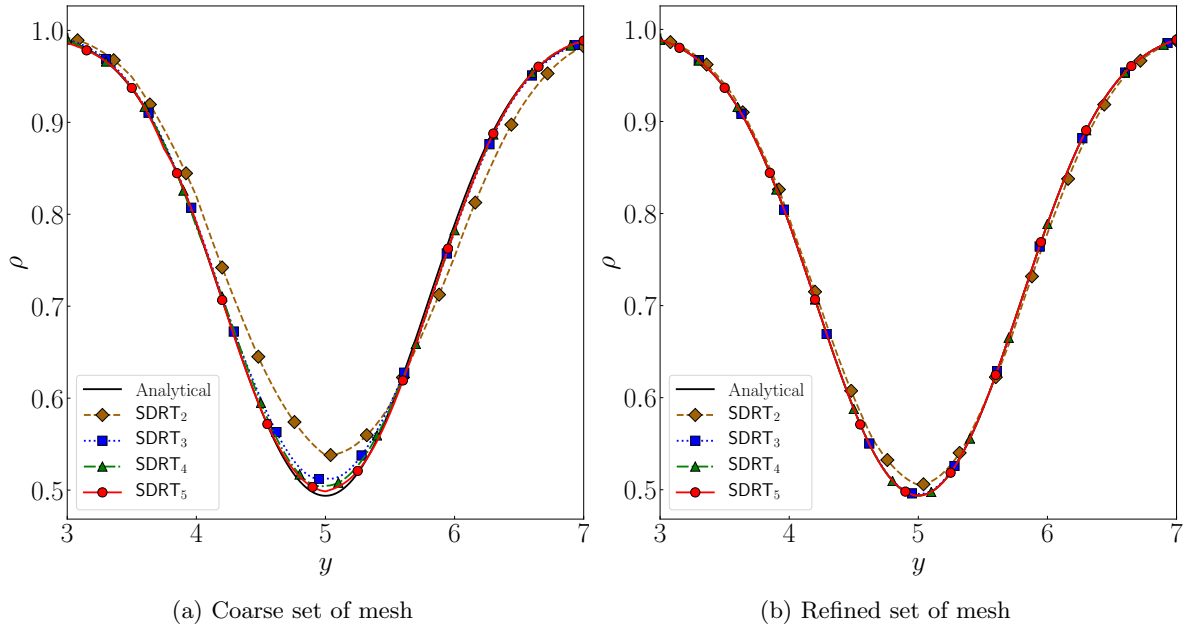


Figure 10: Density ρ at cut $x = 5$ after 10 periods for 3^{rd} to 6^{th} order SDRT schemes

The computation is run on this first set of grids until a final time $t_f = 100\text{s}$, corresponding to 10 periods. Fig. 10a shows a cut on the density at $x = 5$ after 10 periods on the interval $x = [3, 7]$ in order to have a clear view of the impact of the SDRT scheme order on the accuracy of the solution. For the same (or smaller) number of DoF, the solution improves when the degree p increases. The exact same test case is run on a second set of refined grids (~ 9000 DoF) and presented on Fig. 10b. As expected, the SDRT formulation can recover the analytic solution provided that the mesh is refined enough or that the polynomial degree p is sufficiently high. These solutions enable the validation of the proposed SDRT formulation for the non linear Euler equations.

4.3 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. We solve the compressible Navier-Stokes equations and consider a laminar viscous flow over the NACA0012 airfoil. The computational setup is defined by the angle of attack α , the Mach number M_∞ and the Reynolds number $Re = \rho_\infty U_\infty C / \mu_\infty$, where C is the airfoil chord. Three different laminar flow conditions chosen from the NASA technical report [68] are considered:

- Case A: Subsonic flow, $M_\infty = 0.5$, $\alpha = 0^\circ$, $Re = 5000$
- Case B: Subsonic flow, $M_\infty = 0.5$, $\alpha = 2^\circ$, $Re = 5000$
- Case C: Transonic flow, $M_\infty = 0.8$, $\alpha = 10^\circ$, $Re = 500$

The NACA 0012 airfoil equation used is:

$$y = \pm 0.6 \left(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4 \right) \quad (74)$$

so the trailing edge has a zero thickness. At the airfoil, a no-slip adiabatic wall condition is imposed. In order 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 second order triangular elements (with 62 cells on the airfoil). The resulting mesh is shown on Fig. 11. Solutions are time-integrated using the RK3-GS Runge-Kutta method and as before the convection flux is Roe's scheme. The CFL number is set to $0.3/(p+1)$. Results are presented for SDRT schemes from the third to the sixth order at the time $t = 10$ sec. Fig. 12 shows the convergence histories for the three cases. The L_2 error between the initial and the current solution versus time is plotted. Since the reference state is the initial density field the L_2 error first increases before reaching a bound values, indicating that the computation has converged. Solutions can be considered as converged at a time $t \simeq 3$ sec. Steady solutions are obtained for each case and each order of accuracy.

In the following, results obtained for the Case C will be presented in details since the transonic flow can

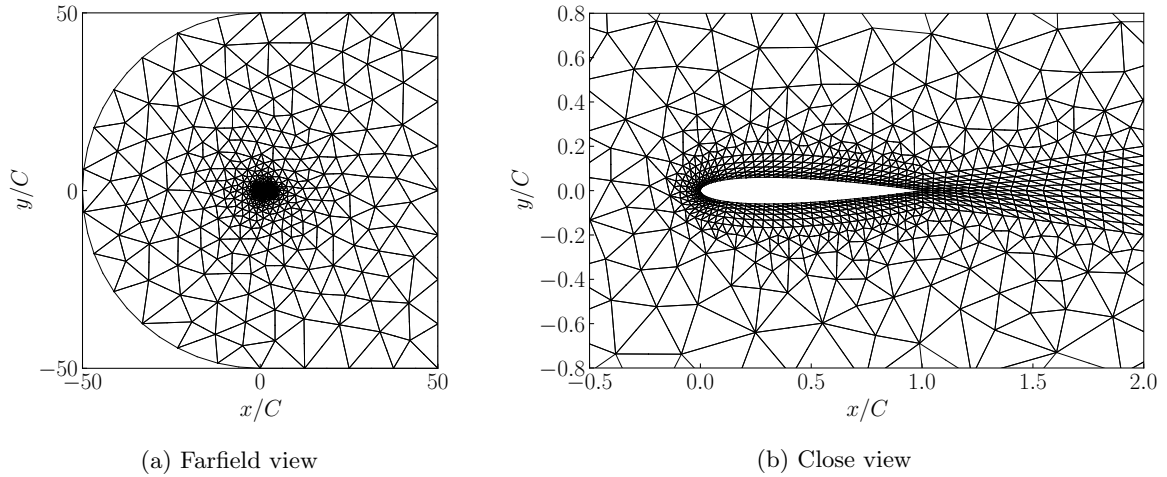


Figure 11: Unstructured mesh around a NACA 0012 airfoil - 2407 second order triangular elements

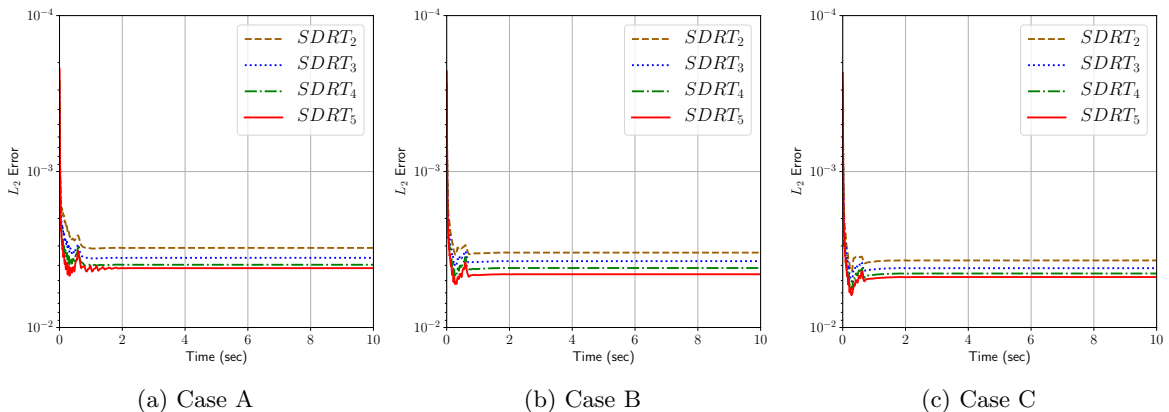


Figure 12: Convergence histories

be considered as the 'most critical' test case. For the subsonic flows (Case A and B), briefer results will be given.

Fig. 13 shows the Mach contours obtained with SDRT schemes of 3rd to 6th order of accuracy for the 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 SDRT₂ and SDRT₃ schemes (Fig. 13a, 13b), 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 SDRT₄ and SDRT₅ schemes (Fig. 13c, 13d) 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 $\alpha = 0^\circ$. Apart from this region, the Mach contours obtained show that the 5th and 6th order SDRT schemes converge to the same solution.

The surface skin-friction coefficient C_f and the surface pressure coefficient C_p distributions are plotted on Fig. 14 and Fig. 15 respectively and compared to results from [68] (finite volume solver RK/Implicit scheme with matrix dissipation on structured grid) on the most refined mesh (1024×512 elements). For both coefficients, there is good agreement between the results obtained from SDRT_p schemes and the NASA data. On Fig. 14, a closer view shows that all of the SDRT schemes were able to capture the maximum value of the surface skin-friction coefficient at the leading edge. As for the Mach contours, discontinuities between cells are observed for SDRT₂ and SDRT₃, resulting from the interpolation post-processing step performed independently on each cell. A difference with the NASA data can be noticed at the trailing edge, where the SDRT_p did not manage to capture the maximum values due to low mesh refinement. The surface pressure coefficient plot (Fig. 15) 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. In addition to Case C, the surface skin-friction coefficient C_f distribution is computed for Case A and Case B. The values of the skin-friction drag coefficients (viscous component of the drag coefficient) $(C_D)_f$ are obtained by a surface integration using the trapezoidal rule with more than 100 control points per edge. For each case, the $(C_D)_f$ values are presented for SDRT_p schemes and compared to the NASA values using structured grid RK/Implicit

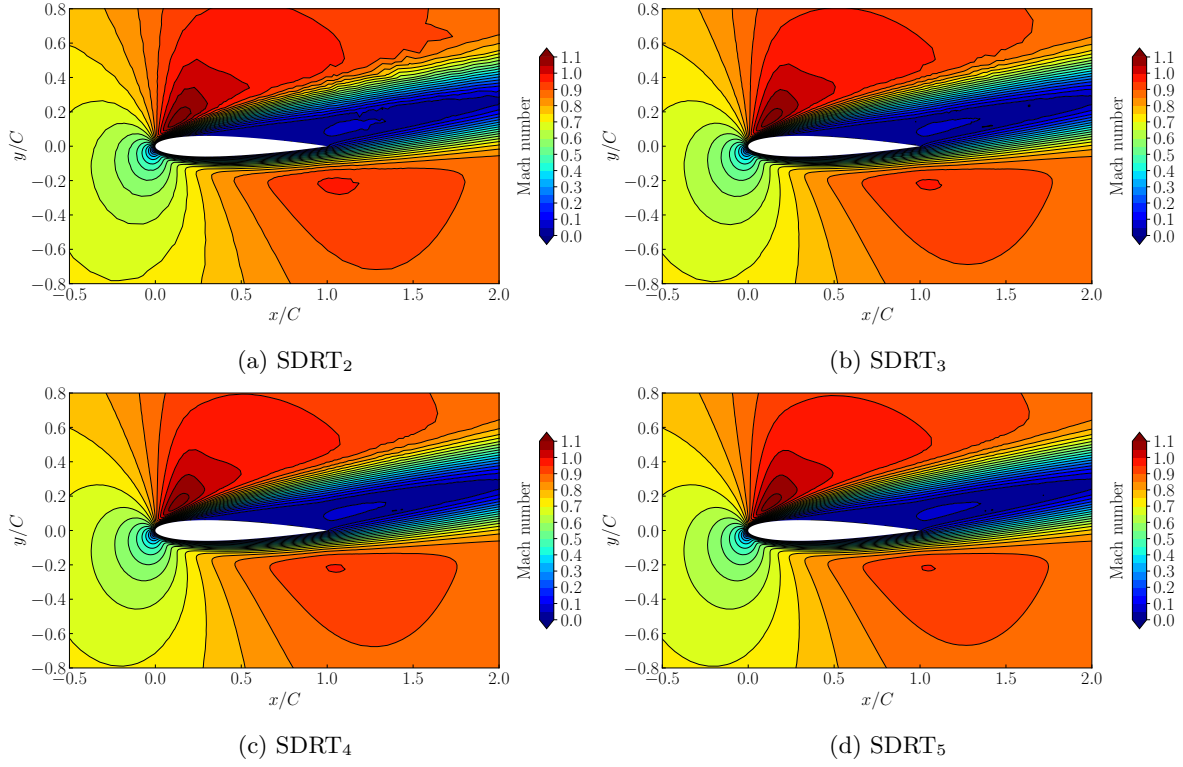


Figure 13: Mach number contours using 3rd to 6th order SDRT schemes for Case C

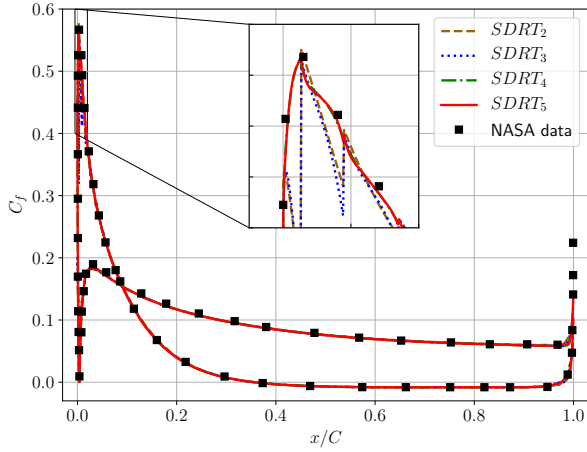


Figure 14: Surface skin-friction coefficient C_f

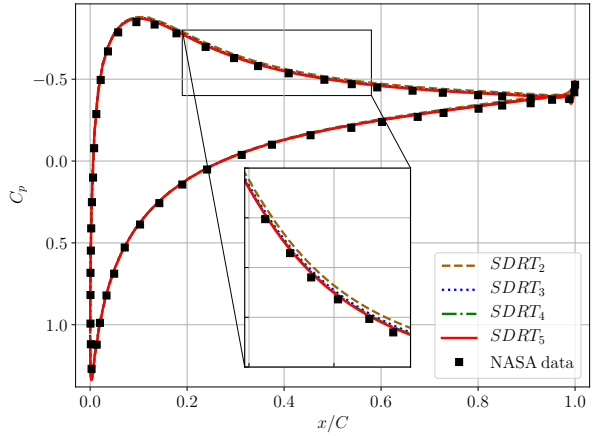


Figure 15: Surface pressure coefficient C_p

scheme with matrix dissipation on the coarser and the most refined mesh (Table 5). For Case B and Case C, all of the $(C_D)_f$ values obtained with the $SDRT_p$ schemes lies in the interval given by the coarse and the refined grid from NASA data. For Case A, the $SDRT_5$ is the only one to lie in the interval. However, other SDRT schemes lead to very close values: the $SDRT_2$, which led to the furthest value, only show a 2.2% difference with the NASA value obtained on the refined mesh.

	DoF number	Case A	Case B	Case C
$SDRT_2$	2407×6 (14, 442)	0.0320594	0.0323926	0.123839
$SDRT_3$	2407×10 (24, 070)	0.0322138	0.0325189	0.123106
$SDRT_4$	2407×15 (36, 105)	0.0323401	0.0324839	0.123628
$SDRT_5$	2407×21 (50, 547)	0.0324200	0.0324823	0.123478
NASA [68]	128×64 (8, 192)	0.0323651	0.0321277	0.121894
NASA [68]	1024×512 (524, 288)	0.0327855	0.0325741	0.127611

Table 5: Comparison of the surface skin-friction drag coefficient $(C_D)_f$

4.4 Viscous flow around a circular cylinder

The last test case aims to validate the method for the computation of viscous flow using hybrid mesh. We consider a steady laminar viscous flow at $Re = 20$ around a cylinder. The Mach number is $M_\infty = 0.1$ and the Reynolds number is defined by $Re = \rho_\infty U_\infty d / \mu_\infty$, where the dynamic viscosity is $\mu_\infty = 1.853 \cdot 10^{-3} Pa \cdot s$ and the cylinder diameter is $d = 1m$. The density ρ_∞ and the velocity U_∞ can be deduced from the temperature $T = 300K$ 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. 16. On the farfield boundary, the pressure, temperature and velocity are settled. At the cylinder surface, a no-slip isothermal wall condition is imposed. Solutions are time-integrated using the RK3-GS scheme and Roe's Riemann solver is used to compute flux at interface flux points. The CFL number is set to $0.3/(p+1)$. The computation is performed using a SDRT₄ scheme until a time $t = 3$ sec. Fig. 17 shows the normalized x -velocity contours and streamlines around the cylinder. The streamlines show a recirculation zone within the wake of the cylinder where two vortices were generated. The computed total drag coefficient (non-viscous and viscous effects) is 2.018 and the reattachment length L/D is equal to 0.916. These values are in good agreement with those from [69] obtained using a body-fitted grid (total drag coefficient of 1.98 and normalized reattachment length of 0.92).

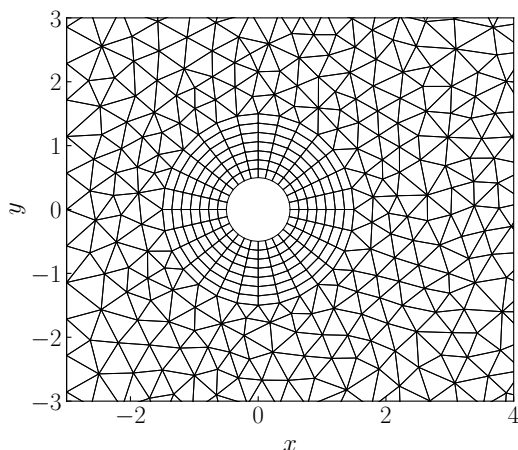


Figure 16: Close view of the hybrid mesh

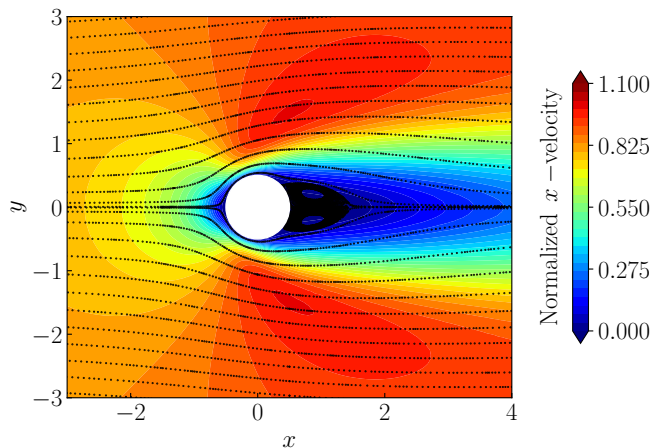


Figure 17: Normalized x -velocity contours and streamlines around the cylinder using a SDRT₄ scheme

5 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. It is today applied routinely to segments, quadrangles and hexahedrons following a tensor approach. The weakness of the method lies in its inability to deal with hybrid grids.

In this context, the goal of the paper is to propose an extension of the SD method to 2D hybrid grids composed of triangles and quadrangles. Following the pioneering work of Balan *et al.* [34, 35], accounting for triangles is possible by introducing the RT space to compute the flux. Compared to the literature, the proposed work gives new results. First, the linear stability analysis of the SDRT method is performed in a framework that differs with the one of the original paper, based now on an eigenvalues analysis, as in [42]. Moreover, the definition of the flux point positions is extended from the papers of Balan *et al.* in two directions. First of all, SDRT schemes were proved stable for a degree of accuracy greater than four, which outperforms results published in the literature. Secondly, the uniqueness of the set of flux points is never obtained and several sets of points lead to a stable formulation. Among them, we also considered several sets of point from recent papers. For all computations at polynomial degree p with $2 \leq p \leq 5$, the internal flux points are located at the position of the control points for Williams-Shunn-Jameson [60] cubature rule and the ones on the edges are the Gauss-Chebyshev points. A theoretical analysis

shows that the present formulation differs with the FR/CPR approaches. There are now two alternative approaches to the DG formulation working on both simplices and quadrangles.

The last effort deals with validation and it is shown that the formulation proposed in the paper is able to perform simulations for linear advection, Euler and Navier-Stokes equations. Order of accuracy is recovered.

The present work is our first effort towards the definition of the SD method for 3D simulations on hybrid grids and many questions are still open. Our next papers will deal with the definition of a stable SD formulation for tetrahedrons 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.

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.

A Definition of the Raviart-Thomas (RT) space

The RT space was originally introduced by [70] to approximate the Sobolev space $H(\text{div})$. Considering the reference triangle \mathcal{T} , the RT space of order k is defined in 2D by:

$$RT_k = (\mathbb{P}_k)^2 + \begin{pmatrix} x \\ y \end{pmatrix} \bar{\mathbb{P}}_k \quad (75)$$

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

$$\mathbb{P}_k(x, y) = \text{Span}\{x^i y^j, i, j \geq 0, i + j \leq k\}, \quad (76)$$

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

$$\bar{\mathbb{P}}_k(x, y) = \text{Span}\{x^i y^j, i, j \geq 0, i + j = k\} \quad (77)$$

and $(\mathbb{P}_k)^2 = (\mathbb{P}_k, \mathbb{P}_k)$ 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 $\dim RT_k = (k+1)(k+3)$. We denote $\vec{\phi}_n, n = 1, \dots, N_{FP}$ the monomials which form a basis in the RT_k space where

$$N_{FP} = (k+1)(k+3) \quad (78)$$

Determination of $\vec{\phi}_n$ for RT_1 , $N_{FP} = 8$

$$\mathbb{P}_1(x, y) = \text{Span}\{1, x, y\} \quad (79)$$

$$\mathbb{P}_1^2(x, y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix} \quad (80)$$

$$\bar{\mathbb{P}}_1(x, y) = \text{Span}\{x, y\} \quad (81)$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \bar{\mathbb{P}}_1 = \left(\begin{pmatrix} x \\ y \end{pmatrix} x, \begin{pmatrix} x \\ y \end{pmatrix} y \right) = \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix} \quad (82)$$

$$RT_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} x \\ 0 \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} 0 \\ y \end{pmatrix}, \begin{pmatrix} x^2 \\ xy \end{pmatrix}, \begin{pmatrix} xy \\ y^2 \end{pmatrix} \quad (83)$$

B Proriot-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 quadrangle 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 [48].

B.1 Collapsed coordinated system

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

$$\begin{cases} \xi = 2\frac{(1+x)}{(1-y)} - 1, \\ \eta = y, \end{cases} \quad \text{or} \quad \begin{cases} x = \frac{(1+\xi)(1-\eta)}{2} - 1, \\ y = \eta. \end{cases} \quad (84)$$

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

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:

$$\Phi_{i,j} = P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p \quad (85)$$

In Eq. (85), $P_i^{\alpha,\beta}$ 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.,

$$\int_{-1}^1 (1-x)^\alpha (1+x)^\beta P_i^{\alpha,\beta}(x) P_j^{\alpha,\beta}(x) dx = \frac{2^{\alpha+\beta+1}}{2i+\alpha+\beta+1} \frac{\Gamma(i+\alpha+1)\Gamma(i+\beta+1)}{i!\Gamma(i+\alpha+\beta+1)} \delta_{ij} \quad (86)$$

where Γ is the Gamma function

$$\Gamma(n) = (n-1)! \quad (87)$$

In the case of $\beta = 0$, Eq. (86) can be simplify as

$$\int_{-1}^1 (1-x)^\alpha P_i^{\alpha,0}(x) P_j^{\alpha,0}(x) dx = \frac{2^{\alpha+1}}{2i+\alpha+1} \delta_{ij} \quad (88)$$

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 .

B.3 Demonstration of L^2 orthogonality

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

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \int_{-1}^1 \int_{-1}^{-y} \Phi_{i,j} \Phi_{k,l} dx dy \quad (89)$$

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

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \int_{-1}^1 \int_{-1}^1 P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta) P_k^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^k P_l^{2k+1,0}(\eta) |J| d\xi d\eta \quad (90)$$

where the Jacobian determinant is

$$|J| = \left| \frac{\partial(x, y)}{\partial(\xi, \eta)} \right| = \frac{1-\eta}{2} \quad (91)$$

Eq. (90) can be written:

$$\begin{aligned} \langle \Phi_{i,j} | \Phi_{k,l} \rangle &= \int_{-1}^1 \int_{-1}^1 \left(\frac{1-\eta}{2}\right)^{i+k+1} P_i^{0,0}(\xi) P_j^{2i+1,0}(\eta) P_k^{0,0}(\xi) P_l^{2k+1,0}(\eta) d\xi d\eta \\ &= \frac{1}{2^{i+k+1}} \int_{-1}^1 P_i^{0,0}(\xi) P_k^{0,0}(\xi) d\xi \int_{-1}^1 (1-\eta)^{i+k+1} P_j^{2i+1,0}(\eta) P_l^{2k+1,0}(\eta) d\eta \end{aligned} \quad (92)$$

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

$$\int_{-1}^1 P_i^{0,0}(\xi) P_k^{0,0}(\xi) d\xi = \frac{2\delta_{ik}}{2i+1} \quad (93)$$

The first integral is equal to zero if $i \neq k$. When $i = k$, using Eq. (88), the second integral becomes:

$$\int_{-1}^1 (1-\eta)^{i+k+1} P_j^{2i+1,0}(\eta) P_l^{2k+1,0}(\eta) d\eta = \int_{-1}^1 (1-\eta)^{2i+1} P_j^{2i+1,0}(\eta) P_l^{2i+1,0}(\eta) d\eta = \frac{2^{2i+2} \delta_{jl}}{2j+2i+2} \quad (94)$$

Then Eq. 92 becomes:

$$\langle \Phi_{i,j} | \Phi_{k,l} \rangle = \frac{\delta_{ik} \delta_{jl}}{(i+1/2)(i+j+1)} \quad (95)$$

From Eq. (95), it can be seen that if $i \neq k$ and $j \neq l$, $\langle \Phi_{i,j} | \Phi_{k,l} \rangle$ is necessarily equal to zero, which shows the orthogonality of the PKD basis in \mathcal{T} .

B.4 Normalized PKD basis

We use the L^2 normalized PKD basis. From Eq. (95), it comes:

$$\|\Phi_{i,j}\|_{\mathcal{T}}^2 = \langle \Phi_{i,j} | \Phi_{i,j} \rangle = \frac{1}{(i+1/2)(i+j+1)} \quad (96)$$

Which leads to the expression of the normalized 2D PKD basis:

$$\Phi_{i,j} = \sqrt{(i+1/2)(i+j+1)} P_i^{0,0}(\xi) \left(\frac{1-\eta}{2}\right)^i P_j^{2i+1,0}(\eta), \quad i+j \leq p \quad (97)$$

C G spectrum using RK6-ldld scheme for SDRT₂ and SDRT₃ schemes

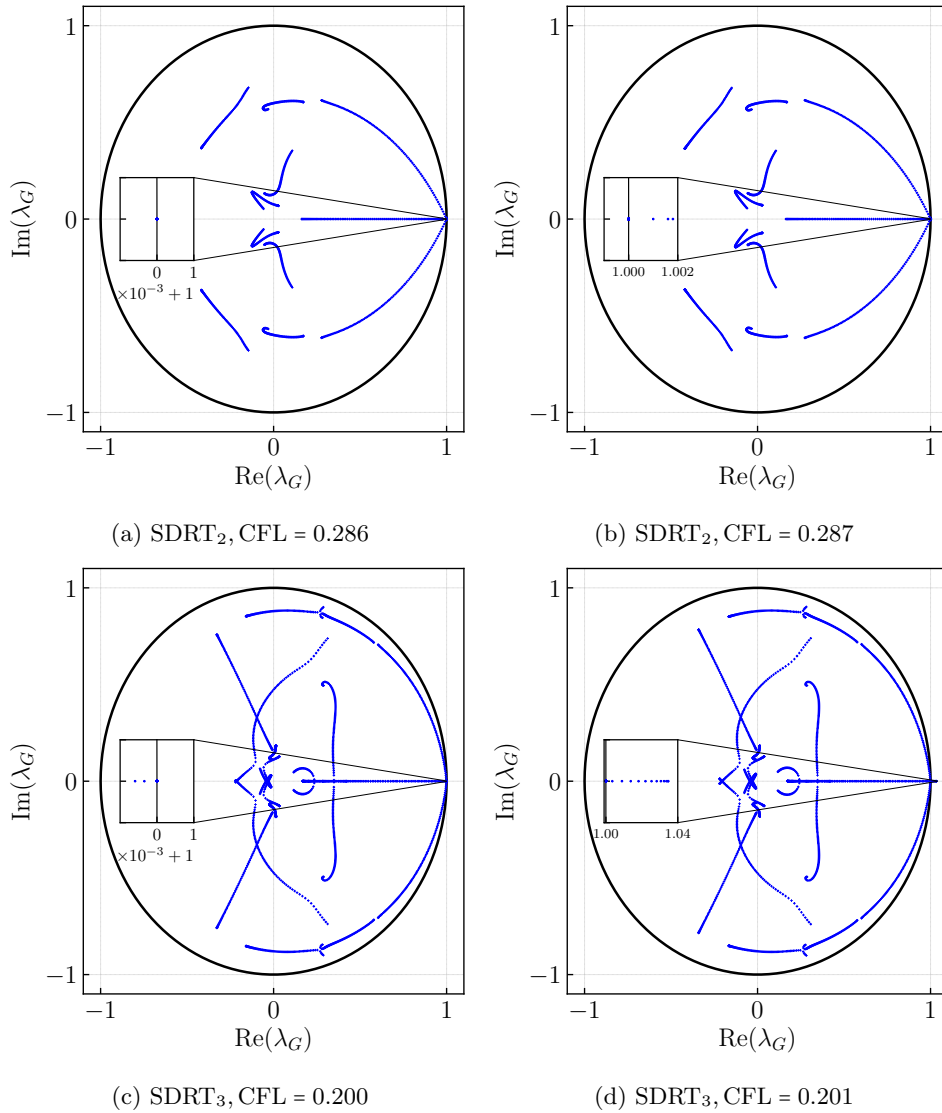


Figure 18: Spectrum of matrix \mathbf{G} using RK6-ldld scheme for 3rd and 4th order SDRT schemes

References

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. Poinot, 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 comporesso, in: *ASME Turbo Expo 2018: Turbomachinery 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:https://doi.org/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] 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.
- [18] 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.
- [19] H. Huynh, Z. Wang, P. Vincent, High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids, *Computers and Fluids* 98 (2014) 209–220. doi:10.1016/j.compfluid.2013.12.007.
- [20] D. A. Kopriva, J. H. Kolas, A Conservative and Staggered-Grid Chebyshev and Multidomain Method and for Compressible and Flows, *Journal of Computational Physics* 125 (1996) 244–261. doi:https://doi.org/10.1006/jcph.1996.0091.
- [21] 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:https://doi.org/10.1006/jcph.1996.0225.
- [22] 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.
- [23] 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.
- [24] 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:https://doi.org/10.2514/6.2006-304.
- [25] 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.
- [26] W. Chen, Y. Ju, C. Zhang, A collocated-grid spectral difference method for compressible flows, *Computers and Fluids* 196 (2020) 104341. doi:10.1016/j.compfluid.2019.104341.
- [27] 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.
- [28] K. Van den Abeele, C. Lacor, Z.J. Wang, On the stability and accuracy of the spectral difference method, *Journal of Scientific Computing* 37 (2008) 162–188.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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:https://doi.org/10.2514/6.2011-47.

- [35] 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.
- [36] 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).
- [37] 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 and Fluids* 184 (2019) 187–198. doi:10.1016/j.compfluid.2019.03.010.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] A. Cassagne, J.-F. Boussuge, N. Villedieu, G. Puigt, I. DAs, 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.
- [42] 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.
- [43] 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).
- [44] T. Poinso, 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.
- [45] 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.
- [46] R. Fiévet, H. Deniau, E. Piot, Strong compact formalism for characteristic boundary conditions with discontinuous spectral methods, accepted for publication in *Journal of Computational Physics* 408 (2020) 109276. doi:10.1016/j.jcp.2020.109276.
- [47] 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.
- [48] G. E. Karniadakis, S. J. Sherwin, *Spectral/HP Element Methods for Computational Fluid Dynamics*, Oxford University Press, Oxford, 2005.
- [49] J. Proriol, Sur une famille de polynômes à deux variables orthogonaux dans un triangle, *Sci. Paris* 257 (1957) 24592461.
- [50] T. Koornwinder, Two-variable analogues of the classical orthogonal polynomials, in: R. Askey (Ed.), *Theory and Applications of Special Functions*, San Diego, 1975.
- [51] M. Dubiner, Spectral methods on triangles and other domains, *Journal of Scientific Computing* 6 (4) (1991) 345–390. doi:10.1007/BF01060030.
- [52] G. Pena, Spectral element approximation of the incompressible Navier-Stokes equations in a moving domain and applications, Ph.D. thesis, Ecole Polytechnique Fdrale de Lausanne (2009).
- [53] P. Castonguay, High-order energy stable flux reconstruction schemes for fluid flow simulations on unstructured grids, Ph.D. thesis, Stanford University (2012).
- [54] S. Sherwin, Dispersion analysis of the continuous and discontinuous Galerkin formulations, in: *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, 2000, pp. 425–431. doi:10.1007/978-3-642-59721-3_43.

- [55] 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.
- [56] G. Cowper, Gaussian quadrature formulas for triangles, *Numerical Methods in Engineering* 7(3) (1973) 405–408. doi:10.1002/nme.1620070316.
- [57] 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.
- [58] 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.
- [59] B. Vioreanu, V. Rokhlin, Spectra of multiplication operators as a numerical tool, *Methods and Algorithms for Scientific Computing, SIAM J. Sci. Comput.* 36 (2014) 267–288. doi:10.1137/110860082.
- [60] 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.
- [61] 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.
- [62] L. Zhang, T. Cui, H. Liu, A set of symmetric quadrature rules on triangles and tetrahedra, *J. Comput. Math.* 27 (2009) 89–96.
- [63] C. Bogey, C. Bailly, A family of low dispersive and low dissipative explicit schemes for flow and noise computations, *Journal of Computational Physics* 194 (2004) 194–214. doi:https://doi.org/10.1016/j.jcp.2003.09.003.
- [64] 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.
- [65] 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.
- [66] C. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, *Lecture Notes in Mathematics* 1697 (1998) 325–432. doi:10.1007/BFb0096355.
- [67] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *Journal of Computational Physics* 43 (2) (1981) 357–372. doi:https://doi.org/10.1016/0021-9991(81)90128-5.
- [68] 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.
- [69] J.-I. Choi, R. C. Oberoi, J. R. Edwards, J. A. Rosati, An immersed boundary method for complex incompressible flows, *Journal of Computational Physics* 224 (2) (2007) 757 – 784. doi:10.1016/j.jcp.2006.10.032.
- [70] P. Raviart, J. Thomas, A mixed finite element method for 2nd order elliptic problem, in: *Lecture Notes in Mathematics* 606, 1977, p. 292315.