J Sci Comput (2017) 72:1232–1268
DOI 10.1007/s10915-017-0399-6

Construction of a p-Adaptive Continuous Residual Distribution Scheme

R. Abgrall[1] · **Q. Viville**[2,3,4] · **H. Beaugendre**[2,3,4] · **C. Dobrzynski**[2,3,4]

Received: 6 November 2015 / Revised: 12 February 2017 / Accepted: 21 February 2017 /
Published online: 13 March 2017

Abstract A p-adaptive continuous residual distribution scheme is proposed in this paper. Under certain conditions, primarily the expression of the total residual on a given element K into residuals on the sub-elements of K and the use of a suitable combination of quadrature formulas, it is possible to change locally the degree of the polynomial approximation of the solution. The discrete solution can then be considered non continuous across the interface of elements of different orders, while the numerical scheme still verifies the hypothesis of the discrete Lax–Wendroff theorem which ensures its convergence to a correct weak solution. We detail the theoretical material and the construction of our p-adaptive method in the frame of a continuous residual distribution scheme. Different test cases for non-linear equations at different flow velocities demonstrate numerically the validity of the theoretical results.

Keywords p-Adaptation · Continuous and discontinuous finite elements · Residual distribution schemes · Non linear equations

1 Introduction

Because of their potential in delivering higher accuracy with lower cost than low order methods, high-order methods for computational fluid dynamics (CFD) have obtained considerable attention in the past two decades [28]. By high order, we mean third order or higher. Most industrial codes used today for CFD simulations are based upon second-order finite volume methods (FVM), finite difference methods (FDM), or finite element methods (FEM) [28]. As second-order methods are used in most CFD codes, some very complex flows simu-

✉ R. Abgrall
remi.abgrall@math.uzh.ch

[1] Institüt für Mathematik, Universität Zürich, Winterthurerstrasse 190, 8057 Zurich, Switzerland

[2] Institut de Mathématiques de Bordeaux, 33 405 Talence, France

[3] Inria Bordeaux Sud Ouest, 33 405 Talence, France

[4] Institut Polytechnique de Bordeaux, 33 405 Talence, France

lations might remain out of their reach. Indeed, in some cases, second-order methods are still too much dissipative and as a consequence they require much finer meshes and become too expensive even on modern supercomputer clusters. In order to deal with a large and diverse range of problems, lots of researches have been conducted with the aim of designing robust and stable high order methods, see [16] and [26]. High-order methods allow the use of coarser meshes [26–28], high-order boundary representation [20], and improve the accuracy of the solutions [11,26,28]. Because of their potential, we believe that the next generation of CFD solvers will have to be based upon high order methods. Besides the use of higher order methods, a very promising approach is the use of *hp*-adaptation to change locally the order of accuracy and the size of the mesh according to the solution [19]. Among high-order methods, the Discontinuous Galerkin (DG) method [15,18,23], the residual distribution (RD) method [1,4,5,12,22] and the hp finite element method (*hp*-FEM) [8,9,14,24] have attracted a lot of interest in the recent years. The DG method has a compact stencil regardless the order of the polynomials representing the solution [17]. This very local formulation leads to a great flexibility, especially for the parallelization of its implementation. However, DG methods suffer from the rapid increase of the number of nodes [10], and then, simulations in three-dimensional space may quickly become too expensive. A possible way to overcome this problem is to use *p*-adaptation [9] (which is conceptually easy with DG methods), where the local approximation order *p* (hence the term *p*-adaptation) is dictated by the flow field. The optimal solution is *hp*-adaptation (mesh and polynomial adaptations) to achieve the best accuracy with the minimum cost. In smooth regions, *p*-adaptation is preferred, whereas in discontinuous regions, *h*-adaptation is preferred.

Another possible approach is the class of residual distribution schemes. RD methods can be easily stabilized and show good shock capturing abilities [22]. They offer a very compact stencil like DG methods, but with a smaller number of nodes [5]. The drawback is that to achieve this low number of nodes, the continuity of the approximation is required and consequently, the use of *p*-adaptation with continuous finite elements in the frame of RD schemes is a priori not possible as this would violate the continuity requirement.

The aim of this work is to demonstrate that it is indeed possible to use *p*-adaptation with continuous finite elements in the frame of RD schemes. Such an approach, while offering the same advantages as classical residual distribution schemes (like among others the low number of nodes, the non-oscillatory behavior and the accuracy on smooth problems [5]), exhibits some interesting advantages thanks to *p*-adaptation, like an improved convergence and better shock capturing abilities. The practical implementation of *p*-adaptation for RD schemes results in a residual based solver that can use high order elements in smooth regions, and low order elements in discontinuous regions. In this sense, this approach is following the recommendations for the next generation of CFD solvers [28].

The hp finite element method can be traced back to the work of I. Babuska et al. [9]. The authors presented an evolution of the finite element method that showed super convergent properties thanks to a suitable combination of *h*-adaptation and *p*-adaptation. Of particular interest to our work, the hp finite element method applied to the simulation of turbulent and laminar flows presented in [8], is a method that couples dynamic adaptation techniques with a high order streamline/upwind Petrov–Galerkin (SUPG) finite element scheme. This method shares some traits with the work presented here, notably the authors try to achieve similar goals: the construction of a high order adaptive scheme using continuous finite elements for the simulation of fluid dynamics. But, as we will show, the construction presented by the authors is very different from the method we present here for residual distribution schemes.

The paper is organized as follows. In Sect. 2, the mathematical problem is defined and we recall briefly the general principles of the residual distribution schemes. In Sect. 3, we expose how it is theoretically possible to use p-adaptation in the frame of a continuous Residual Distribution scheme and we propose the detailed construction of the p-adaptive RD scheme. We present some numerical results in Sect. 4, along with some benefits brought by p-adaptation. In conclusion, we invoke some possible future extensions and developments to the work presented here.

2 Mathematical Problem and Residual Distribution Schemes

2.1 Basic Notions of Residual Distribution Schemes

In this section, the basic notions of RD schemes are summarized, more details can be found in [5]. We are interested in the numerical approximation of steady hyperbolic problems of the form

$$\operatorname{div} \mathbf{f}(\mathbf{u}) = 0\,, \tag{1}$$

where $\mathbf{u} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^{d+2}$ is the vector of conservative variables, Ω is an open set of $\mathbb{R}^d$, $d = 2, 3$ is the spatial dimension and $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_d)$ is the vector of flux functions, with $\mathbf{f}_i(\mathbf{u}) : \Omega \to \mathbb{R}^{d+2}$, $i = 1, \dots, d$. For convenience, the flux vector $\mathbf{f}$ can be written in column:

$$\mathbf{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_{d+2} \end{pmatrix} \tag{2}$$

with f_j=$(\mathbf{f}_{1_j}, \dots, \mathbf{f}_{d_j})$. Boundary conditions, that are problem dependent, are applied, we come back to this point in Sect. 2.4.

We study in the present work the numerical solution of the compressible Euler system of equations with the vector of unknowns

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathsf{v} \\ E \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} \rho \mathsf{v} \\ \rho \mathsf{v} \otimes \mathsf{v} + p\,\mathrm{Id}_{d\times d} \\ (E+p)\mathsf{v} \end{pmatrix}$$

where ρ is the density, v is the fluid velocity, p is the pressure and $E = \rho e^t$ is the total energy per unit volume where e^t is the specific total energy. We consider a calorically perfect gas, and the Euler equations are closed with the relations for energy

$$e^t = e + \frac{\mathsf{v}\cdot\mathsf{v}}{2} \quad \text{and} \quad e = \frac{1}{\gamma - 1} RT, \tag{3}$$

with the thermodynamic relation of state for a perfect gas

$$p = \rho R T \tag{4}$$

with the specific heat ratio $\gamma = 1.4$, T the gas temperature and R the gas constant which is 287 N m/kg for sea-level air.

In this setting, the vector of unknowns $\mathbf{u} \in \mathbb{R}^{d+2}$ is such that $\rho > 0$ and $e > 0$. On solid boundaries, we impose weakly no slip boundary conditions.

In this section, the discussion will stay rather general, and we will not focus particularly on the Euler system, but we are going to work on a generic hyperbolic system. We want to find an approximate solution to Eq. (1) with boundary conditions. We assume that Ω is a polyhedric open set, more precisely we suppose that $\bar{\Omega}$ is constituted of simplices (triangles in dimension two and tetrahedrons in dimension three). We can then consider a finite decomposition of the domain

$$\bar{\Omega} = \underset{K\in\mathscr{T}_h}{\cup} K \tag{5}$$

where K is a simplex and $\mathscr{T}_h$ is a conforming triangulation of $\bar{\Omega}$. We associate to each simplex $K \in \mathscr{T}_h$ a finite element $(K, \mathbb{P}^k(K)^{d+2}, \Sigma_K)$ where Σ_K denotes the set of nodes of K defined by

$$\Sigma_K = \left\{x \in \mathbb{R}^d, \lambda_j(x) \in \{0, \frac{1}{k}, \frac{2}{k}, \ldots, \frac{k-1}{k}, 1\}, 1 \leq j \leq d+1\right\} \tag{6}$$

where λ_j, $j = 1, \ldots, d+1$ are the barycentric coordinates with respect to the vertices of K (the finite elements thus defined are affine equivalent to each other).

For practical applications, it is important to define the basis of the vector space $\mathbb{P}^k(K)$. A natural choice is made by the Lagrange basis. Another choice that will be useful in this work is the Bézier basis. From the definition (6), any node $a_\mu \in \Sigma_K$ can be written as

$$a_\mu = \frac{1}{k}\sum_{j=1}^{d+1} \mu_j a_j,\ \ \mu = (\mu_1, \ldots, \mu_{d+1}) \tag{7}$$

where $(a_j)_{j=1,\ldots,d+1}$ are the vertices of K and $\frac{1}{k}(\mu_1, \ldots, \mu_{d+1})$ are the barycentric coordinates of a_μ with respect to the vertices of K.

The Lagrange basis function φ_μ associated to a_μ can then be written as

$$\varphi_\mu = \left(\prod_{j=1}^{d+1}(\mu_j!)\right)^{-1} \prod_{j=1,\ \mu_j\geq 1}^{d+1} \ \prod_{i=0}^{\mu_j-1} \left(k\lambda_j - i\right) \tag{8}$$

and the Bézier basis function B_μ associated to a_μ can be written as

$$B_\mu = \frac{k!}{\prod\limits_{j=1}^{d+1} \mu_j!} \prod_{j=1}^{d+1} \lambda_j^{\mu_j}\ . \tag{9}$$

We remark that the Bézier and Lagrange basis functions sum up to unity, and that the Bézier basis functions are positive on K.

We introduce now the global finite element space

$$X_h = \left\{\mathbf{u}_h \in C^0(\bar{\Omega})^{d+2};\ \forall K \in \mathscr{T}_h,\ \mathbf{u}_{h|K} \in \mathbb{P}^k(K)^{d+2}, k \in \mathbb{N}^*\right\} \tag{10}$$

where we look for the discrete solution to equations (1). We denote by Σ_h the set of N_h nodes of the finite elements

$$\Sigma_h = \underset{K\in\mathscr{T}_h}{\cup} \Sigma_K \tag{11}$$

and from the conformity assumption of the triangulation, a Lagrange basis of X_h is constituted of the Lagrange basis functions $\psi_i, 1 \leq i \leq N_h$ of the finite elements $(K, \mathbb{P}^k(K)^{d+2}, \Sigma_K)$, $K \in \mathscr{T}_h$ and likewise, a Bézier basis of X_h, denoted again $\psi_i, 1 \leq$

$i \leq N_h$, is constituted of the Bézier basis functions of the finite elements. In either basis, we can now write the approximate solution $\mathbf{u}_h \in X_h$ as

$$\mathbf{u}_h = \sum_{i \in \Sigma_h} \mathbf{u}_i \psi_i \ . \tag{12}$$

In the Lagrange case, we have $\mathbf{u}_j = \mathbf{u}_h(a_j)$, where $a_j \in \Sigma_h$ is the node associated with the global shape function ψ_j as described above.

In this paper, all the numerical applications will be made with simplices, but the work presented here can be extended to other kinds of geometric elements like hexahedrons.

We can now write the discrete equations of the residual distribution scheme. We first introduce the total residual of the element K

$$\Phi^K(\mathbf{u}_h) = \int_K \nabla \cdot \mathbf{f}(\mathbf{u}_h) dx. \tag{13}$$

As X_h is a subspace of $H^1(\Omega)^{d+2}$, we have

$$\mathbf{f}(\mathbf{u}_h) \in H^1(\Omega)^{(d+2)\times d}$$

and so we can transform the volume integral (13) into a surface integral with the divergence formula

$$\Phi^K(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \, dx \tag{14}$$

where $\mathbf{n}$ is the exterior normal to ∂K. This is the form that is used in practice. We define as well the boundary total residual for a boundary element (edge in dimension two and face in dimension three) $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$

$$\Phi^\Gamma = \int_\Gamma \big(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}^h)\big) \cdot \mathbf{n} \, dx \ . \tag{15}$$

Once the total residual Φ^K is computed, the next step is to compute the ”nodal residuals” Φ^K_σ (also called ”residuals” or ”sub-residuals” in the literature) that write in generic form

$$\Phi^K_\sigma = \beta^K_\sigma(\mathbf{u}_h)\Phi^K(\mathbf{u}_h), \sigma \in \Sigma_K. \tag{16}$$

Similarly, the boundary nodal residuals Φ^K_σ are defined by

$$\Phi^\Gamma_\sigma = \beta^\Gamma_\sigma(\mathbf{u}_h)\Phi^\Gamma(\mathbf{u}_h), \sigma \in \Sigma_\Gamma \tag{17}$$

where Σ_Γ is the set of nodes of the face Γ lying on the boundary of the domain. The coefficients β^K_σ et β^Γ_σ are the distribution coefficients and in general they depend on the solution $\mathbf{u}_h$. They are real numbers in the scalar case and are matrices in the case of a system of equations. The nodal residuals must satisfy the conservation constraints

$$\sum_{\sigma \in \Sigma_h} \Phi^K_\sigma = \Phi_K, \forall\, K \in \mathscr{T}_h \tag{18}$$

and

$$\sum_{\sigma \in \Sigma_\Gamma} \Phi^\Gamma_\sigma = \Phi_\Gamma, \forall\, \Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega} \ . \tag{19}$$

The total residual is written with a slight abuse of notation as the integral of $\mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n}$ *which is a vector valued function of size* $d+2$. *With the notation of Sect.* 2.1, *the total residual is then a vector of size* $d+2$ *where the component* j, $1\leq j\leq d+2$, *is the integral of the scalar product* $f_j\cdot\mathbf{n}$. *Like the total residual, the nodal residuals are vectors of dimension* $d+2$. *The boundary total and nodal residuals are of the same dimension.*

Finally, the residual distribution scheme reads

$$\forall\sigma\in\Sigma_h,\ \sum_{K,\,\sigma\in\Sigma_K}\Phi_\sigma^K+\sum_{\Gamma,\,\sigma\in\Sigma_\Gamma}\Phi_\sigma^\Gamma=0. \tag{20}$$

Before going further, let us give some examples of residual distribution schemes.

2.2 Some Particular Residual Distribution Schemes

2.2.1 The Lax–Wendroff Scheme

The Lax–Wendroff scheme is a central linear scheme, which name comes from the fact that in its scalar version, in the case of $\mathbb{P}^1$ interpolation and for a constant advection speed, the scheme coincides with the scheme named Lax–Wendroff presented in [21]. It is defined by

$$\Phi_\sigma^{K,LW}=\frac{\Phi^K}{N_{dof}^K}+\int_{\Omega_K}\big(A\cdot\nabla\varphi_\sigma\big)\,\Xi\,\big(A\cdot\nabla\mathbf{u}_h\big)dx\ . \tag{21}$$

The scalar N_{dof}^K is the cardinal of Σ_K, the term A represents $A=(A_1,\ldots,A_d)$ where

$$A_i(\mathbf{u})=\frac{\partial\mathbf{f}_i(\mathbf{u})}{\partial\mathbf{u}} \tag{22}$$

is the Jacobian matrix of the i-th component of the flux. The product $A\cdot\nabla\varphi_\sigma$ is the matrix defined by

$$A\cdot\nabla\varphi_\sigma=\sum_{i=1}^d A_i\frac{\partial\varphi_\sigma}{\partial x_i} \tag{23}$$

where φ_σ is the basis function (Lagrange or Bézier, see Sect. 2.1) associated to the node σ. The matrix Ξ is a scaling matrix defined by

$$\Xi=\frac{1}{d}|K|\bigg(\sum_{\sigma\in\Sigma_K}R_{\mathbf{n}_\sigma}(\bar{\mathbf{u}})\Lambda_{\mathbf{n}_\sigma}^+(\bar{\mathbf{u}})L_{\mathbf{n}_\sigma}(\bar{\mathbf{u}})\bigg)^{-1}. \tag{24}$$

In the definition above $|K|$ is the volume of the element K, $\bar{\mathbf{u}}$ is the average value of the vector of conservative variables on the element K:

$$\bar{\mathbf{u}}=\frac{1}{N_{dof}^K}\sum_{\sigma\in\Sigma_K}\mathbf{u}_\sigma\ , \tag{25}$$

$R_{\mathbf{n}_\sigma}$ and $L_{\mathbf{n}_\sigma}$ are respectively the matrices of the right and left eigenvectors of the matrix $A\cdot\mathbf{n}_\sigma$ defined as in (23), $\wedge_{\mathbf{n}_\sigma}^+$ is the diagonal matrix with $\max(\lambda_{\mathbf{n}_\sigma},0)$ on the diagonal, where the $\lambda_{\mathbf{n}_\sigma}$ are the eigenvalues of the matrix $A\cdot\mathbf{n}_\sigma$. The vector $\mathbf{n}_\sigma$ is defined by

$$\mathbf{n}_\sigma=\frac{1}{d}\int_K\nabla\varphi_\sigma\,dx. \tag{26}$$

The nodal residuals defined in (21) satisfy the conservation relation (18) as we have

$$\sum_{\sigma \in \Sigma_K} \nabla \varphi_\sigma = 0 \tag{27}$$

for the Lagrange (8) and Bézier basis functions (9).

2.2.2 The Rusanov Scheme

The Rusanov scheme is a generalization of the one dimensional Rusanov scheme. This scheme is obtained from a central distribution of the total residual with a dissipation term added. In the case of a system of equations it is defined by

$$\Phi_\sigma^{K,Rsv} = \frac{1}{N_{dof}^K} \left(\Phi^K(\mathbf{u}_h) + \alpha^K \sum_{\sigma_j \in \Sigma_K} (\mathbf{u}_\sigma - \mathbf{u}_{\sigma_j}) \right). \tag{28}$$

The parameter α^K is chosen as to be larger than the maximum of the spectral radius of the matrix $A \cdot \nabla \varphi_\sigma$. In the scalar case, this choice guaranties that the scheme satisfies a local maximum principle. In the system case, it is proved to be non oscillatory and known to be very dissipative [2].

2.3 Construction of a High Order Monotonicity Preserving Residual Distribution Scheme

If the flow is smooth, we can use the Lax–Wendroff scheme presented above. However, when the solution presents discontinuities, special care has to be taken to handle them. We recall now the method we follow.

2.3.1 Limitation

From the Godunov theorem [13,25], there is no monotone and linear scheme that is more than first order of accuracy. What we call limitation is a procedure that guaranties, at least for scalar problem, that we can get at the same time a local maximum preserving scheme that is formally of maximum accuracy; the accuracy being driven by the polynomial degree plus 1. In the system case, the procedure we describe in this section is high order accurate and provides non oscillatory solutions. The wording ”limitation” is not very adequate but it is the traditional wording in this field.

To construct such procedure, in [2], the idea is to start from a first order monotone scheme and to apply a limitation procedure to the distribution coefficients β_σ^K (presented in Sect. 2.1), such that the limited distribution coefficients are uniformly bounded. In doing so, we see from relation (16) that we obtain high order nodal residuals.

We choose to apply the limitation procedure for the Rusanov scheme (28), as the resulting scheme is well suited to handle the shocks occurring in high speed flows [2].

The limitation procedure is achieved through the following sequence of operations. First, we compute the matrices L_n and R_n which are respectively the matrices of the left and right eigenvectors of the matrix

$$A(\bar{\mathbf{u}}) \cdot \frac{\bar{\mathsf{v}}}{||\bar{\mathsf{v}}||} \tag{29}$$

where $A(\bar{\mathbf{u}})$ represents the Jacobian matrix (22) evaluated at the average state (25) and $\bar{\mathsf{v}}$ is the average speed vector computed the same way as $\bar{\mathbf{u}}$. The nodal residuals (28) are projected on the vector space generated by the left eigenvectors L_n, so the intermediate nodal residuals write

$$\Phi_\sigma^{K,*} = L_n \Phi_\sigma^{K,Rsv} \tag{30}$$

and we have the total residual

$$\Phi^{K,*} = \sum_{\sigma \in \Sigma_K} \Phi_\sigma^{K,*} \, . \tag{31}$$

Then, the high order distribution coefficients are computed from the original first order distribution coefficients by the non linear mapping:

$$\hat{\beta}_\sigma^{K,*} = \frac{\left(\dfrac{\Phi_\sigma^{K,*}}{\Phi^{K,*}}\right)^+}{\displaystyle\sum_{\sigma_j \in K} \left(\dfrac{\Phi_{\sigma_j}^{K,*}}{\Phi^{K,*}}\right)^+} \, . \tag{32}$$

We have built the intermediate high order nodal residuals in the characteristic space:

$$\hat{\Phi}_\sigma^{K,*} = \hat{\beta}_\sigma^{K,*} \Phi^{K,*} \tag{33}$$

and finally we project them from the characteristic space to the physical space:

$$\hat{\Phi}_\sigma^{K} = R_n \hat{\Phi}_\sigma^{K,*} \tag{34}$$

where R_n are the right eigenvectors defined above.

In the particular case where $\bar{\mathsf{v}}$ is close to the null vector:

$$||\bar{\mathsf{v}}|| < tol$$

where tol is a tolerance parameter (for example we used $tol = 10e^{-8}$ for the simulations of Sect. 4), the limited distribution coefficients are directly obtained by the same procedure with $\frac{\bar{\mathsf{v}}}{||\bar{\mathsf{v}}||}$ in (29) replaced by an arbitrary vector of norm unity. In this paper we choose the first element of the canonical basis, we have shown in [2] that this choice has no influence on the stability of the method.

2.3.2 Stabilization

The solution obtained with a limited Rusanov scheme can exhibit spurious modes and can show poor iterative convergence [5]. These problems are solved via the addition of the filtering term presented above for the scheme (21) and so the high order filtered Rusanov scheme for the Euler equations reads

$$\widehat{\Phi}_\sigma^{K,Rus}(\mathbf{u}_h) = \widehat{\Phi}_\sigma^{K}(\mathbf{u}_h) + \theta \int_{\Omega_K} \left(A \cdot \nabla \varphi_\sigma\right) \Xi \left(A \cdot \nabla \mathbf{u}_h\right) dx \tag{35}$$

where θ is a shock capturing term. In most applications, we take $\theta = 1$, and in some cases a more elaborated version must be chosen. In this paper, we take $\theta = 1$.

2.4 Boundary Conditions

For the nodes lying on the boundary of the domain, we use the following formula for the nodal residuals:

$$\forall\, \Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega},\ \forall \sigma \in \Sigma_\Gamma,\ \Phi_\sigma^\Gamma(\mathbf{u}_h) = \int_\Gamma \varphi_\sigma \big(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n}\big)\, dx \tag{36}$$

where $\mathbf{u}_-$ is the state that is imposed by the Dirichlet conditions and $\mathcal{F}$ is a numerical flux that depends on $\mathbf{u}_-$, the outward normal $\mathbf{n}$ and the local state $\mathbf{u}_h$. In the case of a no-slip condition, the boundary condition $\mathbf{u}_-$ satisfies this condition and the numerical flux is only a pressure flux. More details can be found in [5].

2.5 A Lax–Wendroff Like Theorem and Its Consequences

The following theorem has been proved in [6]:

Theorem 2.1 *Assume the family of meshes $\mathscr{T} = (\mathscr{T}_h)_h$ is regular. For K an element or a boundary element of $\mathscr{T}_h$, we assume that the residuals $\{\Phi_\sigma^K\}_{\sigma\in K}$ satisfy:*

- *For any $M \in \mathbb{R}^+$, there exists a constant C which depends only on the family of meshes $\mathcal{T}_h$ and M such that for any $\mathbf{u}_h \in X_h$ with $||\mathbf{u}_h||_\infty \leq M$, then*

$$\|\Phi_\sigma^K(u_{h|K})\| \leq C \sum_{\sigma,\sigma' \in \Sigma_K} |\mathbf{u}_\sigma - \mathbf{u}_{\sigma'}| \tag{37}$$

- *They satisfy the conservation property (18)–(19).*

Then if there exists a constant C_{max} such that the solutions of the scheme (20) *satisfy $||\mathbf{u}_h||_\infty \leq C_{max}$ and a function $\mathbf{v} \in L^2(\Omega)^p$ such that $(\mathbf{u}_h)_h$ (or at least a sub-sequence) verifies $lim_h ||\mathbf{u}_h - \mathbf{v}||_{L^2_{loc}(\Omega)^p} = 0$, then $\mathbf{v}$ is a weak solution of* (1).

One of the interests of this result, besides indicating automatic consistency constraints for a scheme (20), is that the important constraint is to have the conservation relations (18)–(19) *at the element level*. This is a weaker statement than the usual one: usually, we ask conservation at the element interface level, and not *globally* on the element. Note that (18)–(19) enables however to explicitly construct the numerical flux, so that the scheme (18)–(19) can be reinterpreted as an almost standard finite volume scheme, see [3] for more details. Of course there is something to pay: the flux needs to be genuinely multidimensional.

Let us take a closer look at the relation (18)–(19). We need to see how they are actually implemented. Indeed, in the numerical implementation, we do not ask for (18)–(19), but for a discrete version of these relations, namely:

For any element $K \in \mathscr{T}_h$,

$$\sum_{\sigma\in\Sigma_K} \Phi_\sigma^K(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} \tag{38a}$$

and any boundary element $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$,

$$\sum_{\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma = \int_\Gamma \left(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}\,(\mathbf{u}_h)\cdot\mathbf{n}\,\right)\, dx \tag{38b}$$

where here $\int$ signifies that the integral is computed with a quadrature formula, which reads

1. On elements:

$$\int_{\partial K} \mathbf{f}(\mathbf{u}^h)\cdot \mathbf{n} = \sum_{e \text{ edge/face } \subset \partial K} \left(|e| \sum_q \omega_q \mathbf{f}(\mathbf{u}_q)\cdot \mathbf{n}\right)$$

2. On boundary elements:

$$\int_\Gamma (\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n})\, dx = |\Gamma| \sum_q \omega_q \left(\mathcal{F}(\mathbf{u}_q, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_q)\cdot\ \mathbf{n}\right)$$

If we have a close look at the proof of Theorem 2.1, we see that besides the boundedness of the sequence of solutions in suitable norms that enable to use compactness argument, what really matters at the algebraic level is that we have the following property on any edge: *If $\Gamma = \Gamma'$ is the same face shared by respectively the two adjacent elements K and K', then we have $\mathbf{n}_{|\Gamma} = -\mathbf{n}_{\Gamma'}$ and consequently*

$$\int_\Gamma \varphi_\sigma \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n} \quad + \int_{\Gamma'} \varphi_\sigma \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n} = 0, \tag{39}$$

where $\varphi_\sigma \in \mathbb{P}^k(K)$ is the basis function associated to the node σ. This is clearly true because $\mathbf{u}_h$ is continuous. Now, in the numerical implementation, the easiest way to do so is that the quadrature points on Γ seen from K are the same as the ones on Γ seen from K'. These two remarks are at the core of the present development, as we see now.

3 Residual Distribution Schemes and *p*-Adaptation

As we have seen in the previous section, what matters for conservation is that, for any element K, the sum of the residuals is equal to the total residual, or more precisely

$$\sum_{\sigma \in K} \Phi_\sigma^K = \int_{\partial K} \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n}.$$

The total residual $\Phi^K = \int_{\partial K} \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n}$ is obtained thanks to quadrature formulas. We show in this section that in many cases this quantity can be rewritten as a weighted sum of total residuals on sub-elements. To make this more precise, we first look at the quadratic case on a triangle, where the Simpson formula is used on each edge. The main idea of this paper comes from this observation.

We then present the quadratic case on a tetrahedron. From these particular cases, we present two formulas that generalize to arbitrary polynomial orders the formulas given for quadratic polynomials. These general formulas make it possible to construct a *p*-adaptive residual distribution scheme of arbitrary orders, with the definition of a modified nodal residual that we present at the end of the section. We choose to present the particular cases of quadratic approximation along with their generalization, instead of giving only the general formulas, as we think that it is necessary for the comprehension of the reader.

As the formulas presented in this section allow to combine in the same mesh different polynomial orders for the approximation of the solution, we use the term p-adaptation. Our method of p-adaptation is proposed within the frame of a continuous residual distribution scheme and is not as general as the method described for example in [8] *which is proposed in the frame of the Finite Element method (see "Appendix 2").*

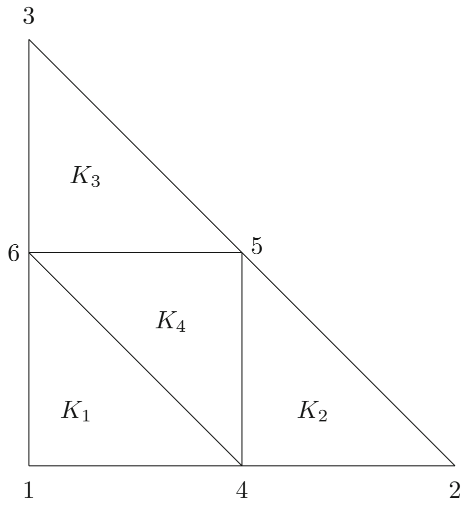

Fig. 1 Subdivided triangle K

3.1 Quadratic Interpolation on Triangular Elements

Let us now consider the case of a triangular element and a quadratic approximation.

We consider a triangle and for simplicity its vertices are denoted by 1,2,3 and the midpoints of the edges are denoted by 4,5,6 (see Fig. 1). We subdivide the triangle $K = (1, 2, 3)$ into four sub-triangles: $K_1 = (1, 4, 6)$, $K_2 = (4, 2, 5)$, $K_3 = (5, 3, 6)$ and $K_4 = (4, 5, 6)$, as shown in Fig. 1. If we denote by λ_i, $i = 1, 2, 3$ the barycentric coordinates corresponding respectively to the vertices $i = 1, 2, 3$ then the linear interpolant of the flux $\mathbf{f}$ reads

$$\mathbf{f}^{(1)} = \sum_{i=1}^{3} \mathbf{f}(\mathbf{u}_i)\lambda_i$$

and the quadratic interpolant of the same flux reads

$$\mathbf{f}^{(2)} = \sum_{i=1}^{6} \mathbf{f}(\mathbf{u}_i)\varphi_i$$

with, from (8):

$$\begin{aligned} &\varphi_i = (2\lambda_i - 1)\lambda_i, \text{ for } i = 1, 2, 3 \\ &\varphi_4 = 4\lambda_2\lambda_1, \qquad \varphi_5 = 4\lambda_3\lambda_2, \quad \varphi_6 = 4\lambda_1\lambda_3\,. \end{aligned}$$

Let us evaluate the total residual for the quadratic interpolant, we denote by $\mathbf{n}_i$ the integral $\int_K \nabla\lambda_i dx$, $i = 1, 2, 3$ and we have:

$$\begin{aligned} \int_K \operatorname{div} \mathbf{f}^{(2)} dx &= \sum_{i=1}^{6} \int_K \mathbf{f}_i \cdot \nabla\varphi_i dx \\ &= \sum_{i=1}^{6} \mathbf{f}_i \cdot \int_K \nabla\varphi_i dx \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{3} \mathbf{f}_i \frac{\mathbf{n}_i}{3} + \frac{4}{3}\big((\mathbf{f}_4(\mathbf{n}_1+\mathbf{n}_2) + \mathbf{f}_5(\mathbf{n}_2+\mathbf{n}_3) + \mathbf{f}_6(\mathbf{n}_1+\mathbf{n}_3) \big)\\
&= \frac{2}{3}\big(\mathbf{f}_1\frac{\mathbf{n}_1}{2} + \mathbf{f}_4\frac{\mathbf{n}_2}{2} + \mathbf{f}_6\frac{\mathbf{n}_3}{2}\big)\\
&\quad + \frac{2}{3}\big(\mathbf{f}_4\frac{\mathbf{n}_1}{2} + \mathbf{f}_2\frac{\mathbf{n}_2}{2} + \mathbf{f}_5\frac{\mathbf{n}_3}{2}\big)\\
&\quad + \frac{2}{3}\big(\mathbf{f}_6\frac{\mathbf{n}_1}{2} + \mathbf{f}_5\frac{\mathbf{n}_2}{2} + \mathbf{f}_3\frac{\mathbf{n}_3}{2}\big)\\
&\quad - 2\big(\mathbf{f}_5\frac{\mathbf{n}_1}{2} + \mathbf{f}_6\frac{\mathbf{n}_2}{2} + \mathbf{f}_4\frac{\mathbf{n}_3}{2}\big)\\
&= \frac{2}{3}\Big(\int_{K_1} \operatorname{div} \mathbf{f}^{(1)} dx + \int_{K_2} \operatorname{div} \mathbf{f}^{(1)} dx\\
&\quad + \int_{K_3} \operatorname{div} \mathbf{f}^{(1)} dx \Big) + 2\int_{K_4} \operatorname{div} \mathbf{f}^{(1)} dx
\end{aligned} \tag{40}
$$

where $\mathbf{f}^{(1)}$ denotes the $\mathbb{P}^1$ interpolant of the flux in each of the sub-triangles of Fig. 1. The change in signs comes from the fact that the inward normals of the sub-triangle K_4, appearing in the expression of $\int_{K_4} \operatorname{div} \mathbf{f}^{(1)} dx$, are the opposite of the vectors $\mathbf{n}_i/2$.

The relation (40) demonstrates that a very simple relation, with positive weights, exists between the $\mathbb{P}^1$ residuals in the sub-triangles and the quadratic residual in K.

3.2 Quadratic Approximation on Tetrahedral Elements

The formula (40) has no equivalent in dimension three if we use Lagrange finite elements, but this problem can be solved by a change of basis as explained in the following. For better clarity, we detail the reference quadratic tetrahedron K in Fig. 2. The ten nodes of K are numbered 1 to 10 and their coordinates are given by:

$$
\begin{aligned}
&1 = (0,0,0);\quad 2 = (1,0,0);\quad 3 = (0,0,1);\quad 4 = (0,1,0);\\
&5 = (1/2,0,0);\quad 6 = (1/2,0,1/2);\quad 7 = (0,0,1/2);\\
&8 = (1/2,1/2,0);\quad 9 = (0,1/2,1/2);\quad 10 = (0,1/2,0).
\end{aligned}
$$

We subdivide the tetrahedron into eight sub-tetrahedrons as shown in Fig. 2. The central octahedron (5, 6, 7, 8, 9, 10) can be split with the diagonal (7, 8), (6, 10) or (5, 9), as shown in Fig. 3, and the sub-tetrahedrons obtained are then:

- Exterior tetrahedrons:
  $K_1 = (5,7,10,1)$; $K_2 = (2,6,5,8)$; $K_3 = (3,9,6,7)$; $K_4 = (8,9,10,4)$.
- With diagonal (7, 8):
  $K_5 = (6,9,8,7)$; $K_6 = (8,9,10,7)$; $K_7 = (8,10,5,7)$; $K_8 = (8,6,5,7)$.
- With diagonal (6, 10):
  $K_5 = (6,9,8,7)$; $K_6 = (8,9,10,7)$; $K_7 = (8,10,5,7)$; $K_8 = (8,6,5,7)$.
- With diagonal (5, 9):
  $K_5 = (5,9,6,7)$; $K_6 = (5,9,10,7)$; $K_7 = (5,9,6,8)$; $K_8 = (5,9,10,8)$.

Now if we try to compute the equivalent of formula (40) for the three dimensional case using a Lagrange interpolation, we find that the coefficients corresponding to the sub-tetrahedrons K_1, K_2, K_3, K_4 are equal to 0. This is due to a property of the $\mathbb{P}^2$ Lagrange basis functions in dimension three:

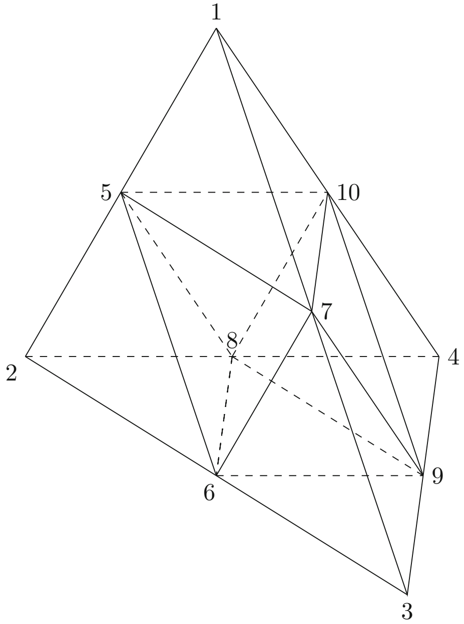

Fig. 2 Subdivided tetrahedron *K*

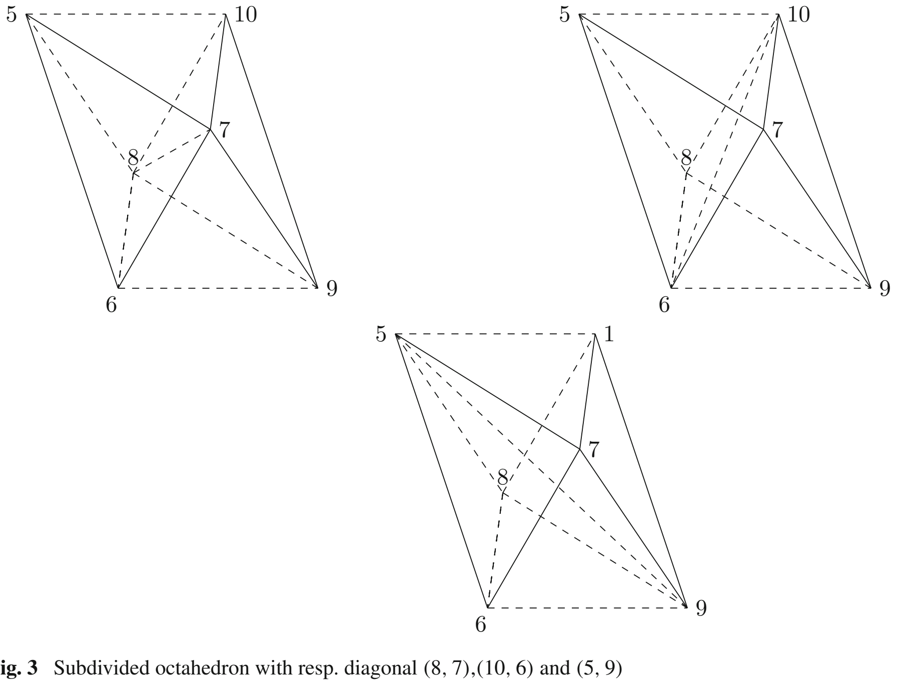

Fig. 3 Subdivided octahedron with resp. diagonal (8, 7),(10, 6) and (5, 9)

$$\int_K \frac{\partial \varphi_j}{\partial x_d} = 0, \quad j = 1, \ldots, 4, \quad d = 1, 2, 3.$$

This implies that the nodal residuals corresponding to the vertices of the tetrahedron K will not contribute to Eq. (20), which may give a problematic residual scheme. To avoid this problem, while still using quadratic elements, we use the Bézier basis functions (9). More precisely, instead of using a Lagrange interpolation of the flux, we approximate the flux with a Bézier expansion. From the linear algebra point of view, this is a change of basis, but with this basis we obtain positive weights.

If we denote by Φ a quadratic polynomial, with the notation of definition (9), the expansion of Φ in the Bézier basis reads

$$\Phi = \sum_{\sigma=1}^{10} \Phi_\sigma B_\sigma$$

and the difference with Lagrange interpolation is that $\Phi_\sigma = \Phi(\sigma)$ only for $\sigma = 1, 2, 3, 4$. For the other nodes, we have

$$\Phi(\sigma) = \frac{\Phi_{j_1} + \Phi_{j_2}}{4} + \frac{\Phi_\sigma}{2} \tag{41}$$

where j_1 and j_2 are the two vertices of the tetrahedron on the edge where σ lies. It is clear that $\Phi_\sigma = \Phi(\sigma) + O(h^2)$. We also notice that

$$\nabla B_\sigma = \sum_{j=1}^{4} P_j(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \nabla \lambda_j$$

where $P_j(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ is a polynomial in λ_1, λ_2, λ_3, and λ_4 with *positive* coefficients.

Now we need to find an equivalent formula of (40) for the three dimensional case with the Bézier basis functions. In the same spirit as for (40), we expand the flux in term of Bézier polynomials:

$$\mathbf{f}^B := \sum_{\sigma=1}^{10} \mathbf{f}_\sigma B_\sigma$$

where for $\sigma = 1, \ldots, 4$, $\mathbf{f}_\sigma$ is the value of the flux at the vertices of the tetrahedron and when $\sigma > 4$, $\mathbf{f}_\sigma$ is defined according to (41).

With the notation $N_i = \nabla \lambda_i$, the gradients of the Bézier basis functions write:

$$\begin{aligned}
&\nabla B_1 = 2\lambda_1 N_1; \quad \nabla B_2 = 2\lambda_2 N_2; \quad \nabla B_3 = 2\lambda_3 N_3; \quad \nabla B_4 = 2\lambda_4 N_4;\\
&\nabla B_5 = 2\big(\lambda_1 N_2 + \lambda_2 N_1\big); \quad \nabla B_6 = 2\big(\lambda_2 N_3 + \lambda_3 N_2\big);\\
&\nabla B_7 = 2\big(\lambda_1 N_3 + \lambda_3 N_1\big); \quad \nabla B_8 = 2\big(\lambda_2 N_4 + \lambda_4 N_2\big);\\
&\nabla B_9 = 2\big(\lambda_3 N_4 + \lambda_4 N_3\big); \quad \nabla B_{10} = 2\big(\lambda_1 N_4 + \lambda_4 N_1\big)\,.
\end{aligned} \tag{42}$$

With $\mathbf{n}_j = \int_K N_j$, we find the following formula for the three dimensional case:

$$\begin{aligned}
2\int_K \text{div}\, \mathbf{f}^B dx &= 2\sum_{\sigma=1}^{10} \Big(\int_K \mathbf{f}_\sigma \text{div}\, B_\sigma dx \Big)\\
&= (\mathbf{f}_1\mathbf{n}_1 + \mathbf{f}_2\mathbf{n}_2 + \mathbf{f}_3\mathbf{n}_3 + \mathbf{f}_4\mathbf{n}_4)\\
&\quad + (\mathbf{f}_5\mathbf{n}_2 + \mathbf{f}_5\mathbf{n}_1) + (\mathbf{f}_6\mathbf{n}_3 + \mathbf{f}_6\mathbf{n}_2)\\
&\quad + (\mathbf{f}_7\mathbf{n}_1 + \mathbf{f}_7\mathbf{n}_3) + (\mathbf{f}_8\mathbf{n}_4 + \mathbf{f}_8\mathbf{n}_2)
\end{aligned}$$

$$
\begin{aligned}
&+ (\mathbf{f}_9\mathbf{n}_3 + \mathbf{f}_9\mathbf{n}_4) + (\mathbf{f}_{10}\mathbf{n}_4 + \mathbf{f}_{10}\mathbf{n}_1)\\
=\ &(\mathbf{f}_1\mathbf{n}_1 + \mathbf{f}_5\mathbf{n}_2 + \mathbf{f}_{10}\mathbf{n}_4 + \mathbf{f}_7\mathbf{n}_3) \quad (I)\\
&+ (\mathbf{f}_2\mathbf{n}_2 + \mathbf{f}_6\mathbf{n}_3 + \mathbf{f}_5\mathbf{n}_1 + \mathbf{f}_8\mathbf{n}_4) \quad (II)\\
&+ (\mathbf{f}_3\mathbf{n}_3 + \mathbf{f}_9\mathbf{n}_4 + \mathbf{f}_6\mathbf{n}_2 + \mathbf{f}_7\mathbf{n}_1) \quad (III)\\
&+ (\mathbf{f}_8\mathbf{n}_2 + \mathbf{f}_9\mathbf{n}_3 + \mathbf{f}_{10}\mathbf{n}_1 + \mathbf{f}_4\mathbf{n}_4) \quad (IV)
\end{aligned}
\tag{43}
$$

The quantities (I), (II), (III), (IV) are interpreted as the integrals of the divergence of the following one degree polynomial functions:

$$
\begin{aligned}
(I) &= \int_{K_1} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx\\
(II) &= \int_{K_2} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx\\
(III) &= \int_{K_3} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx\\
(IV) &= \int_{K_4} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx
\end{aligned}
$$

where

- On K_1, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_1\lambda_1 + \mathbf{f}_5\lambda_2 + \mathbf{f}_{10}\lambda_4 + \mathbf{f}_7\lambda_3$,
- on K_2, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_2\lambda_2 + \mathbf{f}_6\lambda_3 + \mathbf{f}_5\lambda_1 + \mathbf{f}_8\lambda_4$,
- on K_3, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_3\lambda_3 + \mathbf{f}_9\lambda_4 + \mathbf{f}_6\lambda_2 + \mathbf{f}_7\lambda_1$,
- on K_4, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_8\lambda_2 + \mathbf{f}_9\lambda_3 + \mathbf{f}_{10}\lambda_1 + \mathbf{f}_4\lambda_4$.

Strictly speaking, $\mathbf{f}_\sigma$ is *not* the value of the flux at the vertices of K_j, but this is not a problem, following [7].

The equality (43) *shows like equality* (40) *a relation between the quadratic residual in K and the affine residuals in the sub-tetrahedrons, with positive weights independent of the splitting of the central octahedron.*

3.3 General Subdivision Formulas for all Orders of Approximation

The formulas presented above for the specific case of quadratic approximation can be generalized to any degree of approximation as we show now. For dimension two and dimension three, we use a Bézier basis to obtain consistent formulas for both cases. The derivation of these formulas is quite technical, so in order to eliminate any ambiguity, we detail exactly the construction of the formulas with figures, for dimension two and for dimension three, at the risk of being repetitive. A lot of indices are required for the presentation of the formulas and we notify the reader that the notation used here is local to this subsection. With a slight abuse of notation, in this section the term $\mathbb{P}^k$ triangle (respectively $\mathbb{P}^k$ tetrahedron) represents the finite element $(K, \mathbb{P}^k, \Sigma_K)$ where K is a triangle (respectively a tetrahedron). We denote $\mathbf{f}^B$ the k^{th}-order Bézier expansion of the flux $\mathbf{f}$:

$$
\mathbf{f}^B = \sum_{\sigma\in\Sigma_K} \mathbf{f}_\sigma B_\sigma. \tag{44}
$$

3.3.1 The Two Dimensional Case

For the $\mathbb{P}^k$ triangle, the sub-triangles are numbered from the top to the bottom of the triangle as shown in Fig. 1. This numbering is consistent with the iterative construction of the $\mathbb{P}^k$

triangle from the $\mathbb{P}^{k-1}$ triangle: a set of $\mathbb{P}^1$ sub-triangles is added at the bottom of the $\mathbb{P}^{k-1}$ triangle, and the whole set constitutes the $\mathbb{P}^k$ triangle. With this construction, the $\mathbb{P}^k$ triangle is constituted of k layers of $\mathbb{P}^1$ sub-triangles: the $\mathbb{P}^1$ triangle is constituted of the layer 1, the $\mathbb{P}^2$ triangle is constituted of layers 1 and 2, and so on until the $\mathbb{P}^k$ triangle. Each layer $i, i = 1, \dots, k$ is of length $2i - 1$.

Thanks to this numbering, we can now derive the general subdivision formula in the two dimensional case. We first need the following preliminary results:

$$\begin{aligned}
\int_K \nabla B_i =& \frac{2}{k+1} \int_K \nabla\lambda_i,\ i = 1, 2, 3\\
\int_K \nabla B_i =& \frac{2}{k+1} \big(\int_K \nabla\lambda_{i_1} + \int_K \nabla\lambda_{i_2} \big),\ i = 4, \dots, 3k\\
\int_K \nabla B_i =& \frac{2}{k+1} \big(\int_K \nabla\lambda_1 + \int_K \nabla\lambda_2 + \int_K \nabla\lambda_3 \big) = 0,\ i = 3k+1, \dots, \frac{(k+1)(k+2)}{2}
\end{aligned}$$

where i_1 and i_2 are the vertices of the segment that contains the node i. The results above are proved with the following formula:

$$\int_K \lambda_1^{i_1}\, \lambda_2^{i_2}\, \lambda_3^{i_3} = 2\,\mathrm{mes}(K) \frac{i_1!\, i_2!\, i_3!}{(i_1 + i_2 + i_3 + 2)!}. \tag{45}$$

We can now present the subdivision formula for the k-th order in dimension two.

Proposition 3.1 *In dimension two, we have the following formula, for $k \geq 1$:*

$$\begin{aligned}
\int_K div\ \mathbf{f}^B dx = \frac{2}{k+1} \sum_{i=1}^{k} \Big(& \int_{K_{(i-1)^2+1}} div \tilde{\boldsymbol{f}}^{(1)} dx + \int_{K_{(i-1)^2+3}} div \tilde{\boldsymbol{f}}^{(1)} dx + \cdots \\
& + \int_{K_{i^2-2}} div \tilde{\boldsymbol{f}}^{(1)} dx + \int_{K_{i^2}} div \tilde{\boldsymbol{f}}^{(1)} dx \Big).
\end{aligned} \tag{46}$$

Proof First, let us precise the notation used in the proof. The indices in $\mathbf{n}_{j_1}$ and $\mathbf{n}_{j_2}$ represent the vertices of the segment containing the node j. The triangles $K_l,\ l = 1, \dots, k^2$ are the $\mathbb{P}^1$ sub-triangles of the triangle K (see Fig. 1). For a given l, the vertices of K_l are denoted by l_1, l_2, l_3 and the function $\tilde{\mathbf{f}}^{(1)}$ is defined by:

$$\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_{l_1}\lambda_1 + \mathbf{f}_{l_2}\lambda_2 + \mathbf{f}_{l_3}\lambda_3\ . \tag{47}$$

This definition depends on the ordering of the vertices of K_l. They are ordered in the reference triangle as in Fig. 4. The quantities $\mathbf{f}_j$ are not the values of $\mathbf{f}$ at the nodes j, they are considered as the values of $\mathbf{f}$ at the Bézier control points. Practically, they are computed from the definition of the Bézier basis functions (9) and by solving a linear system (Fig. 5).

We denote by $\mathbf{n}_i$ the integral $\int_K \nabla\lambda_i\, dx$, $i = 1, 2, 3$ and we have:

$$\begin{aligned}
\int_K \mathrm{div}\, \mathbf{f}^B dx =& \sum_{j=1}^{\frac{(k+1)(k+2)}{2}} \mathbf{f}_j \cdot \int_K \nabla B_j \\
=& \frac{2}{k+1} \Big(\sum_{j=1}^{3} \mathbf{f}_j \cdot \int_K \nabla\lambda_j \\
& + \sum_{j=4}^{3k} \mathbf{f}_j \cdot \big(\int_K \nabla\lambda_{j_1} + \int_K \nabla\lambda_{j_2} \big)
\end{aligned}$$

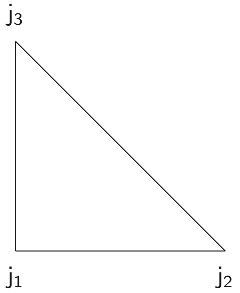

Fig. 4 Ordering of vertices for relation (47)

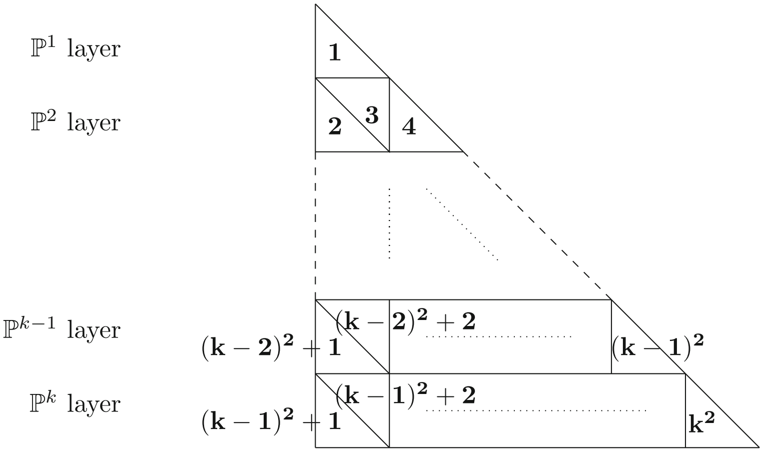
$\mathbb{P}^1$ layer
$\mathbb{P}^2$ layer
$\mathbb{P}^{k-1}$ layer
$\mathbb{P}^k$ layer
1
2
3
4
$(\mathbf{k}-\mathbf{2})^\mathbf{2}+\mathbf{1}$
$(\mathbf{k}-\mathbf{2})^\mathbf{2}+\mathbf{2}$
$(\mathbf{k}-\mathbf{1})^\mathbf{2}$
$(\mathbf{k}-\mathbf{1})^\mathbf{2}+\mathbf{1}$
$(\mathbf{k}-\mathbf{1})^\mathbf{2}+\mathbf{2}$
$\mathbf{k}^\mathbf{2}$

Fig. 5 Numbering of $\mathbb{P}^1$ sub-triangles K_i in $\mathbb{P}^k$ triangle K

$$
\begin{aligned}
&+\sum_{j=3k+1}^{\frac{(k+1)(k+2)}{2}} \mathbf{f}_j\cdot\big(\int_K\nabla\lambda_1+\int_K\nabla\lambda_2+\int_K\nabla\lambda_3)\Big)\\
&=\frac{2}{k+1}\Big(\mathbf{f}_1\cdot\mathbf{n}_1+\mathbf{f}_2\cdot\mathbf{n}_2+\mathbf{f}_3\cdot\mathbf{n}_3\\
&\quad+\mathbf{f}_4\cdot\big(\mathbf{n}_{4_1}+\mathbf{n}_{4_2}\big)+\cdots+\mathbf{f}_{3k}\cdot\big(\mathbf{n}_{3k_1}+\mathbf{n}_{3k_2}\big)\\
&\quad+\mathbf{f}_{3k+1}\cdot\big(\mathbf{n}_1+\mathbf{n}_2+\mathbf{n}_3\big)+\cdots+\mathbf{f}_{\frac{(k+1)(k+2)}{2}}\cdot\big(\mathbf{n}_1+\mathbf{n}_2+\mathbf{n}_3\big)\Big)\\
&=\frac{2}{k+1}\sum_{i=1}^{k}\Big(\int_{K_{(i-1)^2+1}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx+\int_{K_{(i-1)^2+3}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx+\cdots\\
&\quad+\int_{K_{i^2-2}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx+\int_{K_{i^2}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx\Big). \qquad (48)
\end{aligned}
$$

□

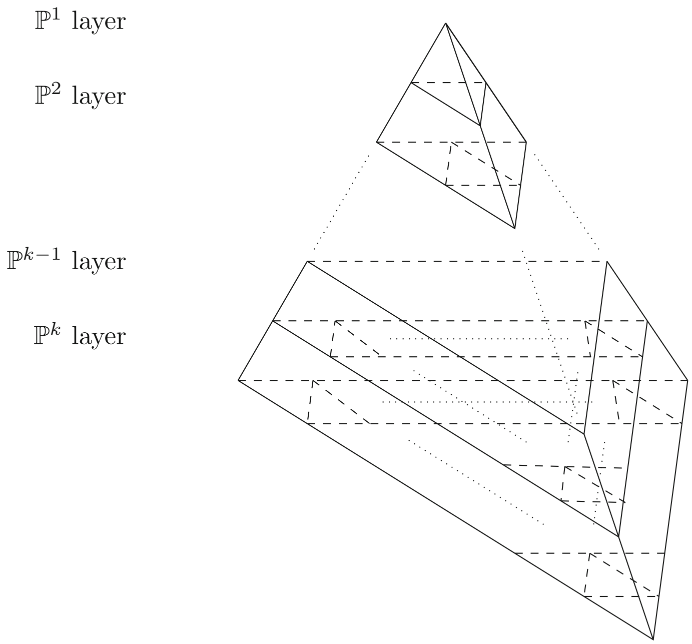

Fig. 6 Subdivided $\mathbb{P}^k$ tetrahedron K: layers and their bottom faces

3.3.2 The Three Dimensional Case

The construction is similar to the two dimensional case, but slightly more complicated. The $\mathbb{P}^k$ tetrahedron is iteratively constructed by layers constituted of $\mathbb{P}^1$ tetrahedrons. Each i-layer, $i = 1, \ldots, k$ has a bottom face constituted of i^2 $\mathbb{P}^1$ triangles. The iterative layers and their bottom faces are described in Fig. 6. Let us now consider the bottom face of the layer i. Its triangles are numbered from the top to the bottom, like in the two dimensional case, as shown in Fig. 7. Each triangle is a face of a $\mathbb{P}^1$ tetrahedron of the layer i, and we number this tetrahedron with the number of such triangles. With this numbering, we can obtain a general three dimensional subdivision formula. We use again the Bézier basis functions (9) like in the two dimensional case, and we have the following preliminary results:

$$\begin{aligned}
\int_K \nabla B_i =& \frac{6}{(k+1)(k+2)} \int_K \nabla \lambda_i,\ i \in S_v\\
\int_K \nabla B_i =& \frac{6}{(k+1)(k+2)} \Big(\int_K \nabla \lambda_{i_1} + \int_K \nabla \lambda_{i_2} \Big),\ i \in S_e\\
\int_K \nabla B_i =& \frac{6}{(k+1)(k+2)} \Big(\int_K \nabla \lambda_{i_1} + \int_K \nabla \lambda_{i_2} + \int_K \nabla \lambda_{i_3} \Big),\ i \in S_f\\
\int_K \nabla B_i =& \frac{6}{(k+1)(k+2)} \Big(\int_K \nabla \lambda_1 + \int_K \nabla \lambda_2 + \int_K \nabla \lambda_3 + \int_K \nabla \lambda_4 \Big) = 0,\ i \in S_t
\end{aligned}$$

where

- S_v is the set of vertices of K,
- S_e is the set nodes situated on the edges of K, excepted the vertices (i_1 and i_2 are the vertices of the edge that contains the node i),

- S_f is the set nodes situated on the faces of K, excepted those on the edges (i_1, i_2 and i_3 are the vertices of the face containing the node i),
- S_t is the set of nodes situated strictly inside K,
- B_i is the k^{th}-order Bézier basis function associated to the node i.

The results above are proved with the 3D version of (45):

$$\int_K \lambda_1^{i_1}\,\lambda_2^{i_2}\,\lambda_3^{i_3}\,\lambda_4^{i_4} = 6\,\mathrm{mes}(K)\frac{i_1!\,i_2!\,i_3!\,i_4!}{(i_1+i_2+i_3+i_4+3)!}. \tag{49}$$

We can now present the subdivision formula for the k-th order in dimension three.

Proposition 3.2 *In dimension three, we have the following formula, for $k \geq 1$:*

$$\begin{aligned}\int_K div\,\mathbf{f}^B dx = \frac{6}{(k+1)(k+2)}\sum_{i=1}^{k}\sum_{i'=1}^{i}\Bigg(&\int_{K_{(i'-1)^2+1}} div\,\tilde{\boldsymbol{f}}^{(1)}dx + \int_{K_{(i'-1)^2+3}} div\,\tilde{\boldsymbol{f}}^{(1)}dx + \cdots\\ &+\int_{K_{i'^2-2}} div\,\tilde{\boldsymbol{f}}^{(1)}dx + \int_{K_{i'^2}} div\,\tilde{\boldsymbol{f}}^{(1)}dx\Bigg).\end{aligned} \tag{50}$$

Proof Let us precise the notation used in the proof. We denote by $\mathbf{f}_j$ the value of the flux at the global node j, and by $\mathbf{f}_{i_{\alpha_\beta}}$ the flux at the node β of the tetrahedron α of the layer i. We precise the sublayer, it indicates in which layer of the bottom face of triangles the tetrahedron is taken (see Fig. 7). It corresponds to the index i'. The indices in $\mathbf{n}_{j_1}$, $\mathbf{n}_{j_2}$ (for $j \in S_e$) represent the vertices of the edge containing the node j and in $\mathbf{n}_{j_1}, \mathbf{n}_{j_2}, \mathbf{n}_{j_3}$ (for j $\in S_f$) they represent the vertices of the face containing the node j. For a given layer i, $i = 1, \ldots, k$, the tetrahedrons K_l are $\mathbb{P}^1$ sub-tetrahedrons of the tetrahedron K in the layer i. The number l, with $l = (i-1)^2, \ldots, i^2$, is local to the layer i and coincides with the number of the triangle l of the bottom face of the layer i which is a face of K_l (see Figs. 6, 7). With this numbering, we have a general formula in the three dimensional case that is similar to the two dimensional case. For a given l, the vertices of K_l are denoted by l_1, l_2, l_3, l_4 and the function $\tilde{\mathbf{f}}^{(1)}$ is defined by:

$$\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_{l_1}\lambda_1 + \mathbf{f}_{l_2}\lambda_2 + \mathbf{f}_{l_3}\lambda_3 + \mathbf{f}_{l_4}\lambda_4\ . \tag{51}$$

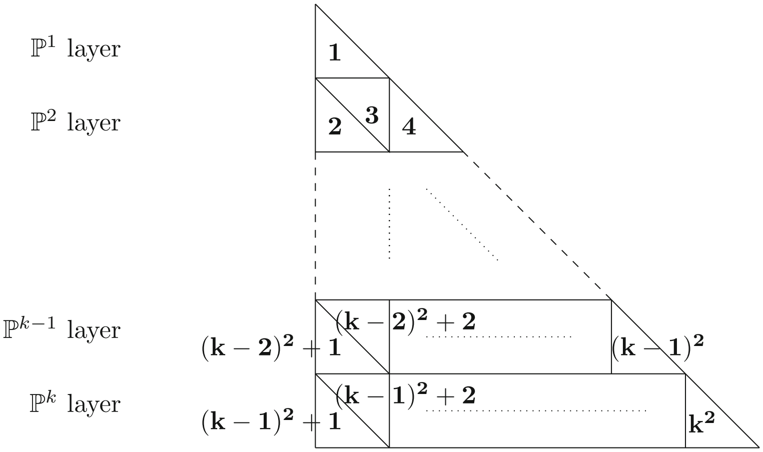
$\mathbb{P}^1$ layer
1
$\mathbb{P}^2$ layer
2
3
4
$\mathbb{P}^{k-1}$ layer
$(\mathbf{k}-\mathbf{2})^2+\mathbf{1}$
$(\mathbf{k}-\mathbf{2})^2+\mathbf{2}$
$(\mathbf{k}-\mathbf{1})^2$
$\mathbb{P}^{k}$ layer
$(\mathbf{k}-\mathbf{1})^2+\mathbf{1}$
$(\mathbf{k}-\mathbf{1})^2+\mathbf{2}$
$\mathbf{k}^2$

Fig. 7 Numbering of $\mathbb{P}^1$ sub-triangles constituting the bottom face of the $\mathbb{P}^k$ layer of tetrahedron K

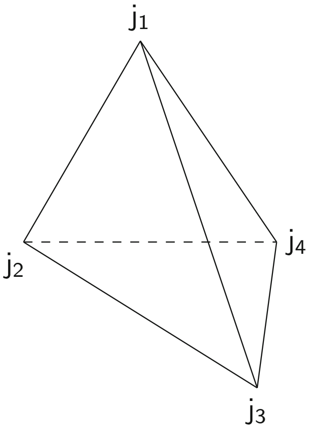

Fig. 8 Ordering of vertices

This definition depends on the ordering of the vertices of K_l. They are ordered in the reference tetrahedron as in Fig. 8. Like in the two dimensional case, the quantities $\mathbf{f}_j$ are not the values of $\mathbf{f}$ at the nodes j, they are considered as the values of $\mathbf{f}$ at the Bézier control points and in practice are computed from the definition of the Bézier basis functions (9) and by solving a linear system.

We denote by $\mathbf{n}_i$ the integral $\int_K \nabla\lambda_i \, dx$, $i = 1, 2, 3$ and we have:

$$\begin{aligned}
\int_K \operatorname{div} \mathbf{f}^B dx &= \sum_{j=1}^{\frac{(k+1)(k+2)(k+3)}{6}} \mathbf{f}_j \cdot \int_K \nabla B_j \\
&= \frac{6}{(k+1)(k+2)} \Bigg(\sum_{j\in S_v} \mathbf{f}_j \cdot \int_K \nabla\lambda_j \\
&\quad + \sum_{j\in S_e} \mathbf{f}_j \cdot \Big(\int_K \nabla\lambda_{j_1} + \int_K \nabla\lambda_{j_2} \Big) \\
&\quad + \sum_{j\in S_f} \mathbf{f}_j \cdot \Big(\int_K \nabla\lambda_{j_1} + \int_K \nabla\lambda_{j_2} + \int_K \nabla\lambda_{j_3} \Big) \\
&\quad + \sum_{j\in S_t} \mathbf{f}_j \cdot \Big(\int_K \nabla\lambda_1 + \int_K \nabla\lambda_2 + \int_K \nabla\lambda_3 + \int_K \nabla\lambda_4 \Big) \Bigg) \\
&= \frac{6}{(k+1)(k+2)} \Bigg(\sum_{j\in S_v} \mathbf{f}_j \cdot \mathbf{n}_j \\
&\quad + \sum_{j\in S_e} \mathbf{f}_j \cdot \big(\mathbf{n}_{j_1} + \mathbf{n}_{j_2} \big) \\
&\quad + \sum_{j\in S_f} \mathbf{f}_j \cdot \big(\mathbf{n}_{j_1} + \mathbf{n}_{j_2} + \mathbf{n}_{j_3} \big) \\
&\quad + \sum_{j\in S_t} \mathbf{f}_j \cdot \big(\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4 \big) \Bigg)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{6}{(k+1)(k+2)}\Big(\mathbf{f}_{1_{1_1}}\cdot\mathbf{n}_1+\mathbf{f}_{1_{1_2}}\cdot\mathbf{n}_2+\mathbf{f}_{1_{1_3}}\cdot\mathbf{n}_3+\mathbf{f}_{1_{1_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. 1)}\\
&\quad+\mathbf{f}_{2_{1_1}}\cdot\mathbf{n}_1+\mathbf{f}_{2_{1_2}}\cdot\mathbf{n}_2+\mathbf{f}_{2_{1_3}}\cdot\mathbf{n}_3+\mathbf{f}_{2_{1_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. 2, tet. 1, sublay. 1)}\\
&\quad+\mathbf{f}_{2_{2_1}}\cdot\mathbf{n}_1+\mathbf{f}_{2_{2_2}}\cdot\mathbf{n}_2+\mathbf{f}_{2_{2_3}}\cdot\mathbf{n}_3+\mathbf{f}_{2_{2_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. 2, tet. 2, sublay. 2)}\\
&\quad+\mathbf{f}_{2_{4_1}}\cdot\mathbf{n}_1+\mathbf{f}_{2_{4_2}}\cdot\mathbf{n}_2+\mathbf{f}_{2_{4_3}}\cdot\mathbf{n}_3+\mathbf{f}_{2_{4_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. 2, tet. 4, sublay. 2)}\\
&\quad+\cdots\\
&\quad+\mathbf{f}_{k_{1_1}}\cdot\mathbf{n}_1+\mathbf{f}_{k_{1_2}}\cdot\mathbf{n}_2+\mathbf{f}_{k_{1_3}}\cdot\mathbf{n}_3+\mathbf{f}_{k_{1_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. } k\text{, tet. 1, sublay. 1)}\\
&\quad+\mathbf{f}_{k_{2_1}}\cdot\mathbf{n}_1+\mathbf{f}_{k_{2_2}}\cdot\mathbf{n}_2+\mathbf{f}_{k_{2_3}}\cdot\mathbf{n}_3+\mathbf{f}_{k_{2_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. } k\text{, tet. 2, sublay. 2)}\\
&\quad+\mathbf{f}_{k_{4_1}}\cdot\mathbf{n}_1+\mathbf{f}_{k_{4_2}}\cdot\mathbf{n}_2+\mathbf{f}_{k_{4_3}}\cdot\mathbf{n}_3+\mathbf{f}_{k_{4_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. } k\text{, tet. 4, sublay. 2)}\\
&\quad+\cdots\\
&\quad+\mathbf{f}_{k_{(k-1)^2+1_1}}\cdot\mathbf{n}_1+\mathbf{f}_{k_{(k-1)^2+1_2}}\cdot\mathbf{n}_2\\
&\quad+\mathbf{f}_{k_{(k-1)^2+1_3}}\cdot\mathbf{n}_3+\mathbf{f}_{k_{(k-1)^2+1_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. } k\text{, tet. } (k-1)^2+1\text{, sublay. } k\text{)}\\
&\quad+\cdots\\
&\quad+\mathbf{f}_{k_{k^2{}_1}}\cdot\mathbf{n}_1+\mathbf{f}_{k_{k^2{}_2}}\cdot\mathbf{n}_2+\mathbf{f}_{k_{k^2{}_3}}\cdot\mathbf{n}_3+\mathbf{f}_{k_{k^2{}_4}}\cdot\mathbf{n}_4 \;\; \text{(lay. } k\text{, tet. } k^2\text{, sublay. } k\text{)}\Big)\\
&= \frac{6}{(k+1)(k+2)}\sum_{i=1}^{k}\sum_{i'=1}^{i}\Bigg(\int_{K_{(i'-1)^2+1}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx+\int_{K_{(i'-1)^2+3}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx+\cdots\\
&\quad+\int_{K_{i'^2-2}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx+\int_{K_{i'^2}}\operatorname{div}\tilde{\mathbf{f}}^{(1)}dx\Bigg).
\end{aligned}
\tag{52}
$$

□

3.4 Definition of the Nodal Residuals in a Divided Element

We denote here by S the set of $\mathbb{P}^1$ sub-elements of a given element K. The term element represents a triangle in dimension two or a tetrahedron in dimension three as above. For a given element K subdivided into smaller elements indexed by $\xi, \xi \in S$ (for example $\xi = K_1, \dots, K_4$ in the case of quadratic interpolation in dimension two), we have, as we have proved, a relation between the residual Φ^K and the residuals Φ^ξ of the sub-elements of K. Once the residuals Φ^ξ are computed, the next step is to compute the nodal residuals Φ^ξ_σ, as described in Sect. 2.2. Then, the nodal residuals Φ^ξ_σ need to be modified in order to satisfy the conservation relation (18), as we show now. We denote by $\gamma_\xi, \xi \in S$ the coefficients given by the formulas of Sect. 3.3.

Proposition 3.3 *If the nodal residuals $\Phi^\xi_\sigma, \sigma \in \Sigma_\xi, \xi \in S$ satisfy the conservation relation (18), then the nodal residuals defined by*

$$\Phi^K_\sigma := \sum_{\xi\in S, \sigma\in\Sigma_\xi} \gamma_\xi \Phi^\xi_\sigma \tag{53}$$

satisfy the conservation relation (18).

Proof With the notation of Sect. 3, $\mathbf{f}^{(k)}$ represents the k-th order interpolant of the flux $\mathbf{f}$. We assume that

$$\forall \xi \in S, \sum_{\sigma\in\Sigma_\xi} \Phi^\xi_\sigma = \int_{\partial\xi} \mathbf{f}^{(1)} \cdot \mathbf{n}$$

and that

$$\Phi_\sigma^K = \sum_{\xi \in S, \sigma \in \Sigma_\xi} \gamma_\xi \Phi_\sigma^\xi .$$

We then have

$$\begin{aligned}\sum_{\sigma \in \Sigma_K} \Phi_\sigma^K &= \sum_{\sigma \in \Sigma_K} \left(\sum_{\xi \in S, \sigma \in \Sigma_\xi} \gamma_\xi \Phi_\sigma^\xi \right) \\ &= \sum_{\xi \in S} \gamma_\xi \left(\sum_{\sigma \in \Sigma_\xi} \Phi_\sigma^\xi \right) \\ &= \sum_{\xi \in S} \gamma_\xi \int_{\partial \xi} \mathbf{f}^{(1)} \cdot \mathbf{n} \\ &= \int_{\partial K} \mathbf{f}^{(k)} \cdot \mathbf{n} \, dx .\end{aligned}$$

□

We remark that it is possible to use any scheme inside the elements $\xi \in S$ as long as they satisfy the conservation relation (18) and the nodal residuals in K are defined by relation (53). Under the assumption of the Lax–Wendroff theorem, the new scheme with arbitrary mixed $\mathbb{P}^1$ and $\mathbb{P}^k$ elements will also be convergent to a weak solution of the problem because the conservation property (18) is still satisfied (see [4]). We follow the same reasoning for the boundary residuals.

3.5 Practical Implementation

The nodal residuals are modified according to the definition given in Sect. 3.4. We describe now the consequences on the practical implementation of the residual distribution scheme.

3.5.1 Computation of the Nodal Residuals

As the computation of the nodal residuals is made inside each sub-element ξ of a divided element K, it is convenient to introduce $\Phi_\sigma^{K,\xi}$ the contribution of the node σ in triangle K brought by the sub-element ξ, as it is the quantity that is actually computed. It is defined by

$$\Phi_\sigma^{K,\xi} := \gamma_\xi \, \Phi_\sigma^\xi . \tag{54}$$

3.5.2 Computation of the Jacobian Matrix of the Implicit Scheme

The system of equations (20) is solved with an implicit scheme detailed in Appendix 1. To implement the subdivision formulas, the implicit scheme requires only a slight modification in the definition of the Jacobian matrices. Indeed, to compute the Jacobian matrix, for example in the case of the Lax–Wendroff scheme (21), we need to differentiate the two terms of (21):

$$\frac{\partial \Phi_\sigma^\xi(\mathbf{u}_h)}{\partial \mathbf{u}_j} = \frac{1}{N_{dof}^\xi} \frac{\partial}{\partial \mathbf{u}_j} \Big(\Phi^\xi(\mathbf{u}_h) + \int_\xi A \cdot \nabla \varphi_\sigma \; \Xi \; A \cdot \nabla \mathbf{u}_h dx \Big) \tag{55}$$

and so, according to (53), we have, with the same notation as in Sect. 3.5.1:

$$\frac{\partial \Phi_\sigma^{K,\xi}(\mathbf{u}_h)}{\partial \mathbf{u}_j} := \gamma_\xi \, \frac{\partial \Phi_\sigma^{\xi}(\mathbf{u}_h)}{\partial \mathbf{u}_j}. \tag{56}$$

3.5.3 Boundary Conditions

The nodal residuals of boundary faces (here we use the term face for either the edge of a triangle in dimension two or the face of a tetrahedron in dimension three) are computed by using the formula (36). If we denote by Γ the face of a subdivided element K lying on the boundary of Ω, Γ is therefore subdivided into sub-faces. Let ς be such a sub-face. From the relation (53), we see that in order to be consistent with relation (36) in the case of subdivision, we need to multiply the nodal residuals of the subdivided boundary by the sub-division coefficient of the element containing this sub-divided boundary face, and so the contribution of the node σ in the face Γ brought by the sub-face ξ is

$$\Phi_\sigma^{\Gamma,\varsigma} \; := \gamma_\xi \; \Phi_\sigma^{\varsigma} \tag{57}$$

and thus the contribution to the global Jacobian is

$$\frac{\partial \Phi_\sigma^{\Gamma,\varsigma} \; (\mathbf{u}_h)}{\partial \mathbf{u}_j} := \gamma_\xi \; \frac{\partial \Phi_\sigma^{\varsigma} \; (\mathbf{u}_h)}{\partial \mathbf{u}_j} \; . \tag{58}$$

3.5.4 Choice of Quadrature Formulas

As explained in Sect. 2.5, the relation (39) is verified if we use the same quadrature points at the interfaces between elements. In the case of an interface between two elements of the same degree, we simply use the same quadrature formula, and so the requirements of relation (39) are automatically satisfied. For an interface between a subdivided element and a non-subdivided one, we use quadrature formulas such that all the quadrature points at the interface physically coincide.

4 Numerical Results

We present now some numerical results for different speed flows in dimension two and three that illustrate the theoretical results exposed above. Even if the formulas presented in Sect. 3.3 make it possible to use any polynomial order, for simplicity the test cases presented in this paper use $\mathbb{P}^1$ and $\mathbb{P}^2$ elements. As explained in Sect. 2.3, for subsonic flows we use the Lax–Wendroff scheme (21) and for transonic and faster flows we use the Rusanov scheme (35).

Our approach can be described as follows. In all test cases except the subsonic test case, the mesh is mostly made of $\mathbb{P}^2$ elements except in the shock zone where $\mathbb{P}^1$ elements are used. The shock zone is located by a shock detector (see [5]) based here on the variation of pressure inside each element of the mesh:

$$\varphi_K = \max_{\sigma\in\Sigma_K} \left(\max_{K',\sigma\in\Sigma_{K'}} \frac{|\max\limits_{\nu\in\Sigma_{K'}} p_\nu - \min\limits_{\nu\in\Sigma_{K'}} p_\nu|}{|\max\limits_{\nu\in\Sigma_{K'}} p_\nu \; | + | \min\limits_{\nu\in\Sigma_{K'}} p_\nu \; | + \epsilon} \right) \tag{59}$$

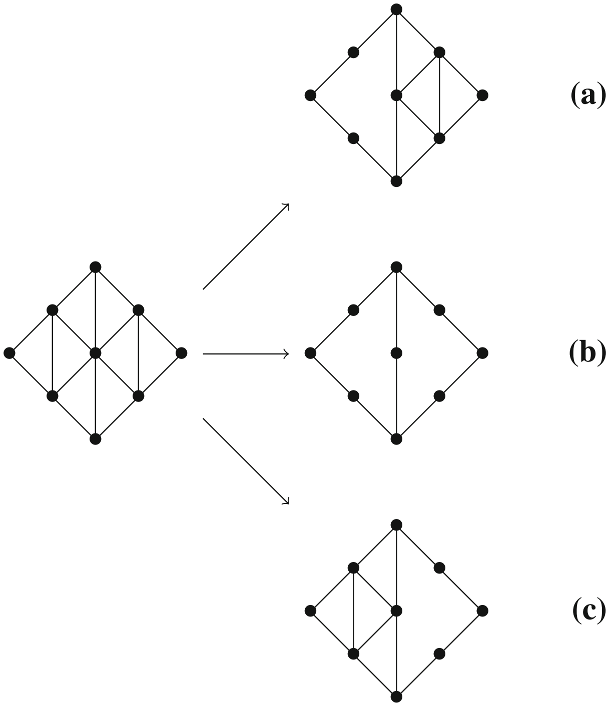

Fig. 9 Possible transformations of two adjacent (initially subdivided) triangles. Notice that no hanging node is generated

where p_ν is the pressure at the node ν, K, K' are elements of the mesh $\mathscr{T}_h$ and ϵ is the machine epsilon (in our implementation, $\epsilon = 1e - 16$). This shock detector is used as follows. We set a threshold φ which depends on the test case, we start with a mesh that contains only subdivided $\mathbb{P}^1$ elements, then after a short number of iterations (which depends on the test case too), the values φ_K are computed for each element K. If the value φ_K is above the threshold φ, then the element remains subdivided with $\mathbb{P}^1$ elements, otherwise, the element becomes a $\mathbb{P}^2$ element and is not subdivided. As a result, we use $\mathbb{P}^1$ elements in the shock zone, and $\mathbb{P}^2$ elements everywhere else. The threshold φ is determined empirically for each test case by initially running the same simulation with different values of φ for a short number of iterations. The optimal value of φ is the one for which the shock zone seems to be visually the closest to the zone where there are strong variations of the isolines of pressure.

We precise now what happens at the interface between two elements. To illustrate this, we have represented in Fig. 9 the possible transformations of two adjacent elements in the quadratic two-dimensional case. As described above, the two adjacent elements are initially subdivided into $\mathbb{P}^1$ elements. Then, they can be transformed into either one element subdivided into $\mathbb{P}^1$ elements and one $\mathbb{P}^2$ element (case a and c), or they can be transformed into two $\mathbb{P}^2$ elements (case b). It is important to remark that the method does not generate hanging nodes, in dimension two or three and for any degree k of approximation.

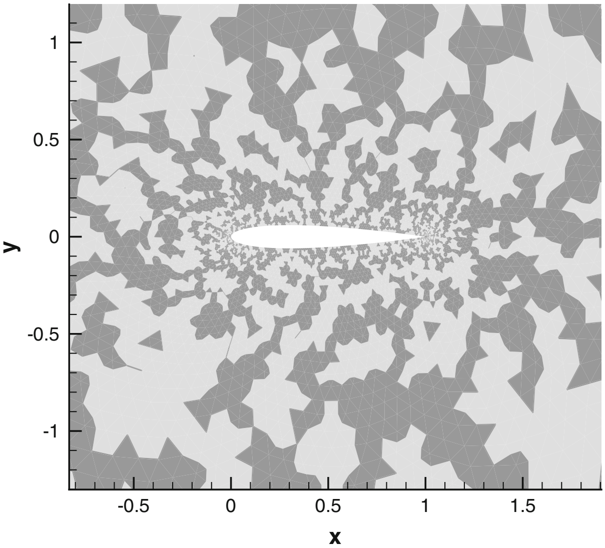

Fig. 10 Mesh for test case 4.1 with elements randomly subdivided: elements in dark zones are subdivided, the others are not subdivided

4.1 Subsonic Flow

We show here, just for theoretical purposes, how our method behaves with the mesh of a NACA0012 wing profile constituted of randomly subdivided elements. The inflow condition is Mach=0.5, the pressure is $P_{inlet} = 0.7$ and half of the elements of the mesh are randomly selected and subdivided into $\mathbb{P}^1$ elements. As the flow is subsonic, we use the Lax–Wendroff scheme (21). The mesh is shown in Fig. 10.

We can see, in Fig. 11, a comparison of the convergence curves when $\mathbb{P}^1$, $\mathbb{P}^2$ and mixed elements ($\mathbb{P}^2$ elements and subdivided $\mathbb{P}^1$ elements) are used. We remark that with the same level of convergence for the three types of finite elements, the convergence speed with subdivided elements is as expected between those obtained with $\mathbb{P}^1$ and $\mathbb{P}^2$ elements.

As stated above, for this test case the elements are arbitrarily subdivided and we do not take advantage of p-adaptation to improve the accuracy of the solution. In the next test cases, the elements will be subdivided according to the properties of the solution.

4.2 Transonic Flow

We test our method on a Naca0012 wing profile with Mach=0.8, a pressure of 0.71 and an angle of attack of 1.25 degrees. For this problem, we use the limited Rusanov scheme (35), more adapted for transonic flows than the Lax–Wendroff scheme, as explained in Sect. 2.3. As small size, low-order elements capture irregular solutions better than high-order elements, which to the contrary approximate smooth solutions better than low-order elements [9], we use the shock detector (59) that allows to use $\mathbb{P}^1$ elements only where the shock is detected and $\mathbb{P}^2$ elements otherwise. The mesh is shown in Fig. 12, the subdivided elements are dark-colored and correspond to the elements containing a strong variation of pressure. The convergence curve obtained with p-adaptation, Fig. 13, shows a jump of the residual due to the switch from $\mathbb{P}^1$ elements to $\mathbb{P}^2$ elements everywhere except in the shock zone. After the

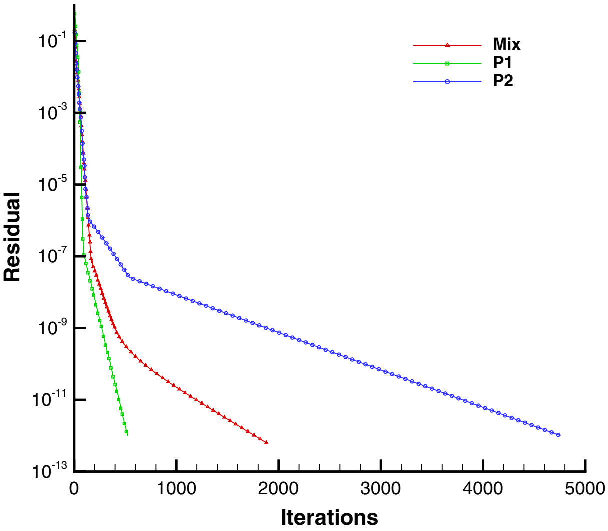

Fig. 11 Convergence of residuals for test case 4.1

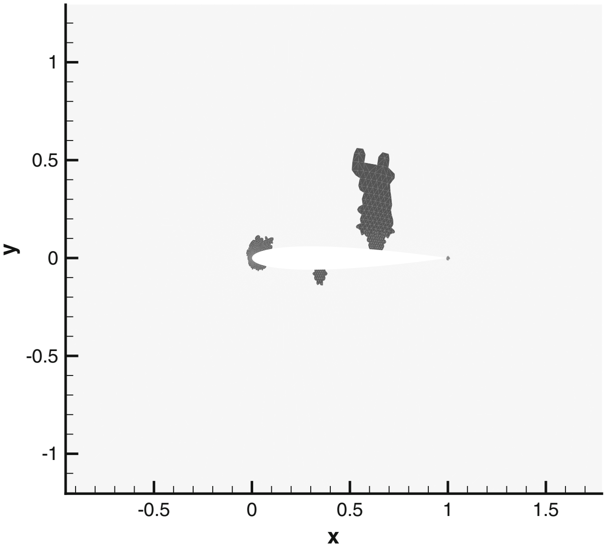

Fig. 12 Shock zone for test case 4.2: subdivided elements are *dark-colored*

jump, we observe the convergence of the residual. For this test case φ was set to 1.5, and the switch was set after 300 iterations. We remark from Fig. 14, that the position of the shock obtained with mixed elements is in very good agreement with the position predicted using classical $\mathbb{P}^2$ or $\mathbb{P}^1$ elements. This is important for the validation of our method and proves that we are here consistent with the results obtained with classical $\mathbb{P}^1$ and $\mathbb{P}^2$ elements.
In addition, we make the following interesting observation. As we use smaller $\mathbb{P}^1$ elements in the discontinuous zone where they are better suited to capture a discontinuity than coarser $\mathbb{P}^2$ elements, we obtain a better representation of the shock, as with the $\mathbb{P}^1$ mesh. This

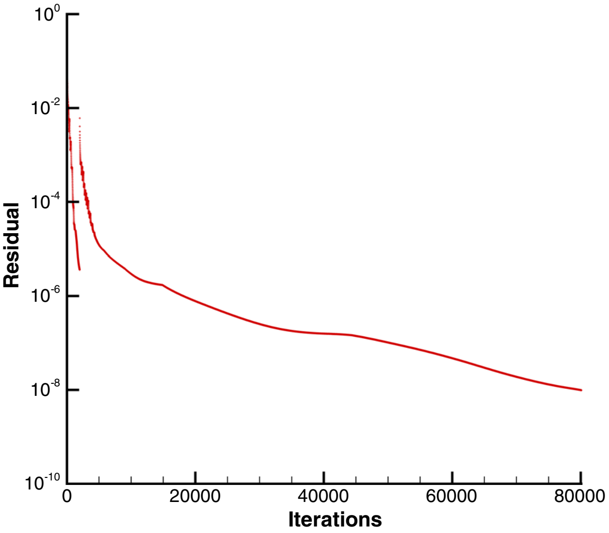

Fig. 13 Convergence of residual for test case 4.2 with mixed elements

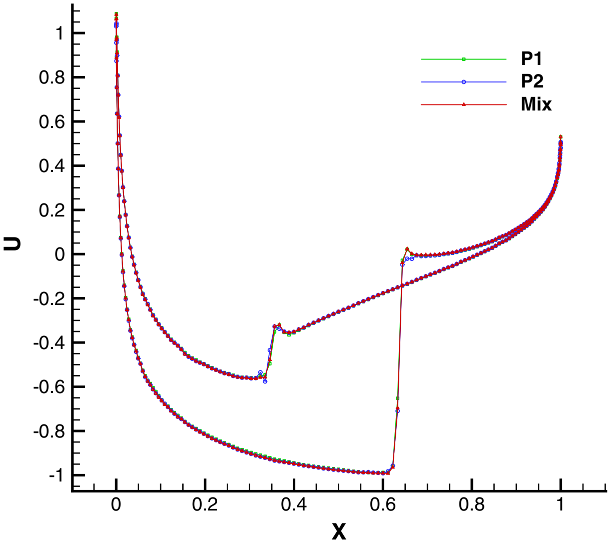

Fig. 14 Comparison of pressure coefficients obtained with $\mathbb{P}^1$, $\mathbb{P}^2$ and mixed elements

observation is confirmed if we compare, in Fig. 15, the solutions obtained with *p*-adaptation to the solution obtained with a $\mathbb{P}^2$ mesh. We can see that the shock is better represented with the *p*-adaptive solution.

4.3 Supersonic Flow

Now we present a numerical test with a higher speed, Mach=3. Because of the higher speed used, such test cases can be more difficult to run. The application of *p*-adaptation, even to

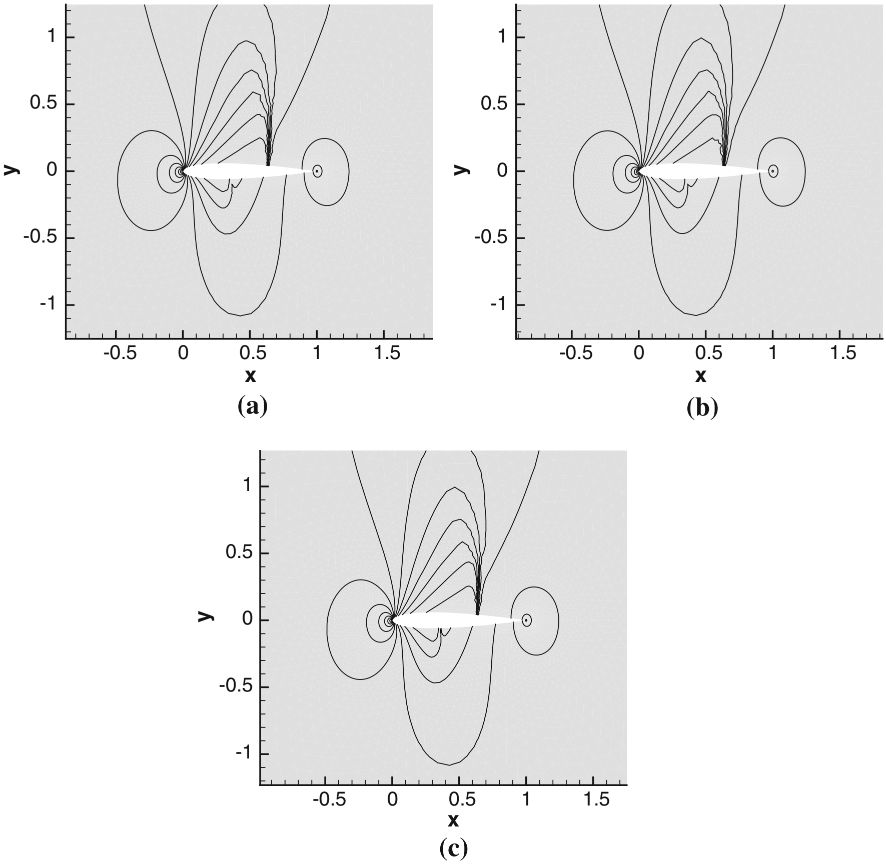

Fig. 15 Comparison of solutions for test case 4.2 between $\mathbb{P}^2$ and mixed elements: **a** pressure ($\mathbb{P}^1$ elements), **b** pressure ($\mathbb{P}^2$ elements), **c** pressure (mixed elements)

a little number of elements (only in the shock zone), allows convergence of the residual and proves to be an efficient approach to simulate such phenomenon. Since the number of sub-divided elements is small the method remains mostly $\mathbb{P}^2$ based.

The following parameters are set: the mesh is made of 3749 vertices and contains a sphere of diameter 1, centered in 0, which is moving at Mach=3.0.

We divide the boundary conditions into four sub-boundaries as shown in Fig. 16, and we detail in the following the conditions applied to each of these boundaries:

- on the sphere (boundary 1) inside the domain, we impose a slipping wall boundary condition,
- in front of the sphere (boundary 2, the half circle on the left of the domain), we impose Dirichlet boundary conditions with $(\rho, u, v, p) = (1.4,\ 3,\ 0,\ 1)$,
- on the upper and lower horizontal lines (boundary 3), we impose a slipping wall boundary condition,
- behind the sphere (boundary 4, the vertical line at the right of the domain), we use a Steger-Warming exit boundary condition [5] with $(\rho, u, v, p) = (1,\ 0.8,\ 0,\ 0.3)$.

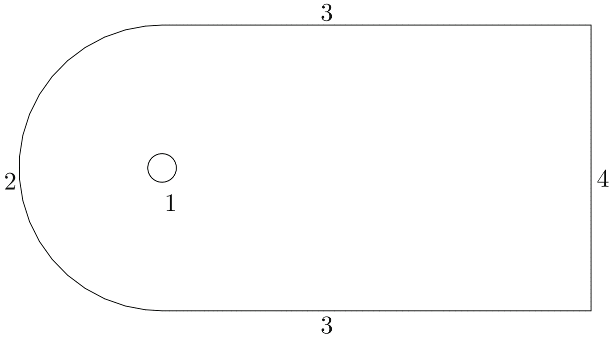

Fig. 16 Boundaries for test case 4.3

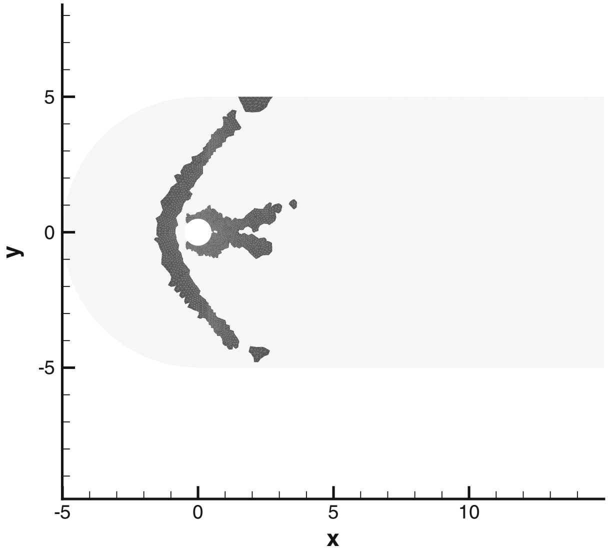

Fig. 17 Shock zone for test case 4.3: subdivided elements are *dark-colored*

As initial conditions we set a discontinuity line at $x = 0.435$, with $(\rho, u, v, p) = (1.4,\ 0,\ 0,\ 1)$ on the left of the discontinuity and $(\rho, u, v, p) = (1.4,\ 3,\ 0,\ 1.4)$ on the right.

We use the same method as before with the shock detector (59) and obtain the mesh shown in Fig. 17. The threshold is set at $\varphi = 3$ and the switch is set at 300 iterations. Again, we notice in Fig. 18 a jump in the residual due to the change from a $\mathbb{P}^1$-only scheme to a mixed $\mathbb{P}^1$-$\mathbb{P}^2$ scheme after the activation of the shock detector.

The isolines of the Mach number, the pressure and the density are shown in respectively Figs. 19, 20, and 21. As of today, we have not been able to make the classical $\mathbb{P}^2$ scheme converge for this test case. With p-adaptation we obtain a high-order solution that physically agrees with the solution obtained with a classical $\mathbb{P}^1$ scheme.

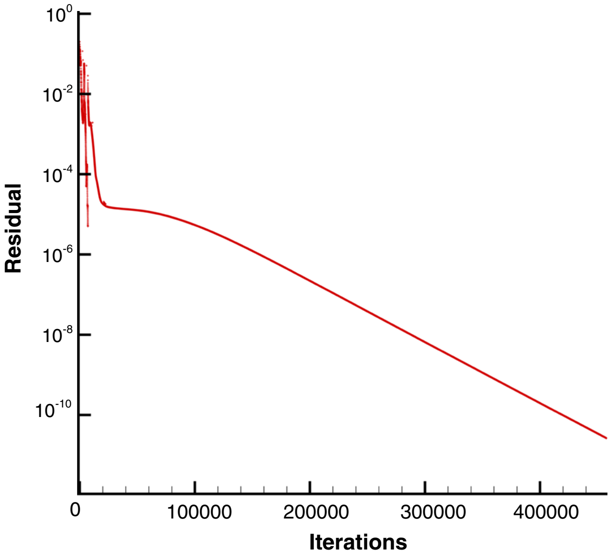

Fig. 18 Convergence of residual in test case 4.3

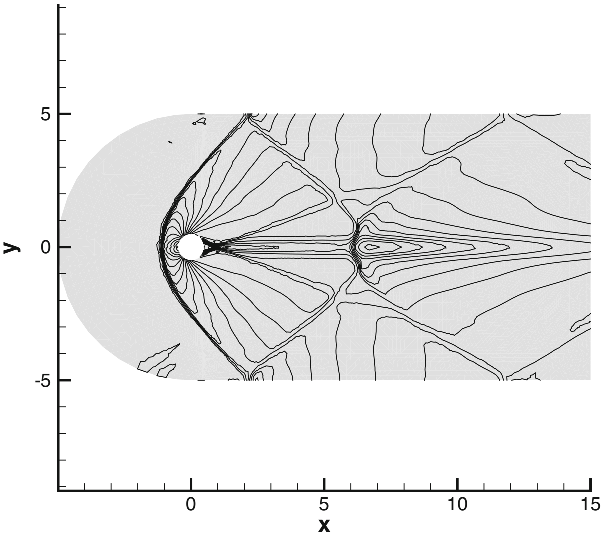

Fig. 19 Mach number isolines for test case 4.3

4.4 Hypersonic Three Dimensional Flow

We present now a numerical test case with a hypersonic speed (Mach=8) in dimension three. Like in dimension two, the application of *p*-adaptation to a little number of elements in the shock zone (the method remaining mostly $\mathbb{P}^2$ based), proves practically to be efficient as it allows the convergence of the residual and gives a solution that looks physically admissible.

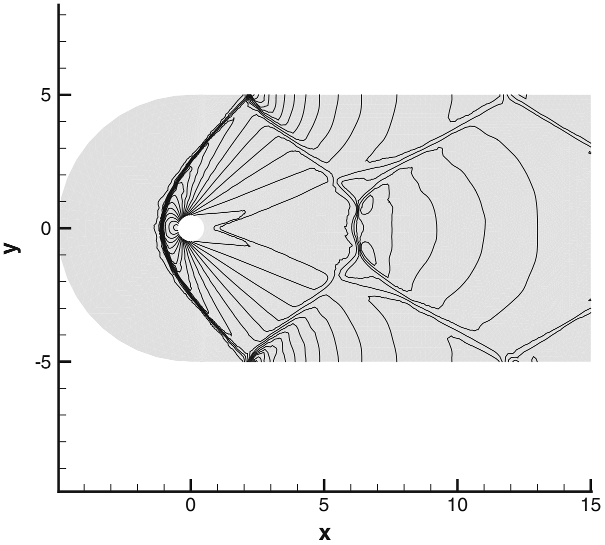

Fig. 20 Pressure isolines for test case 4.3

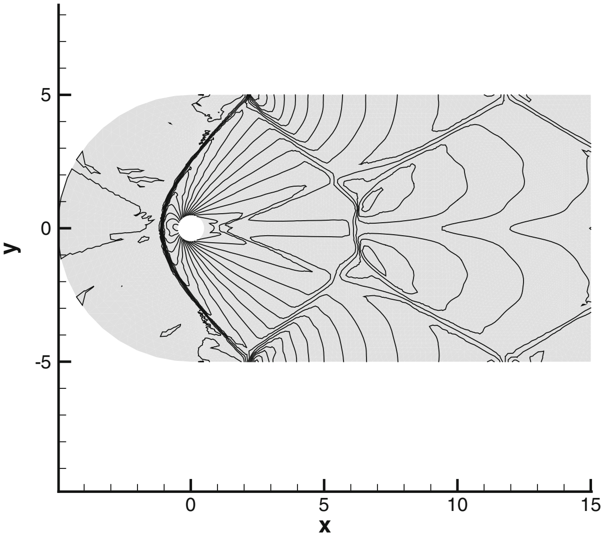

Fig. 21 Density isolines for test case 4.3

The following parameters are set: a sphere of diameter two is centered in 0 and is moving at the speed of Mach=8. The boundary conditions are divided into four sub-boundaries, as shown in Fig. 22. On the sphere inside the domain (boundary 1), we impose a slipping wall boundary condition, in the left face of the domain (boundary 2), we impose a Steger-Warming entry boundary condition, with $(\rho, u, v, w, p) = (8.0\ ,\ 8.25\ ,\ 0.0\ ,\ 0.0\ ,\ 116.5)$, in the right face of the domain (boundary 3), we impose a Steger-Warming exit boundary condition, with $(\rho, u, v, w, p) = (1.4\ ,\ 0.0\ ,\ 0.0\ ,\ 0.0\ ,\ 1.0)$, and on the other faces of the boundary (boundary 4), we impose a slipping wall boundary condition. As an initial condition we set a vertical

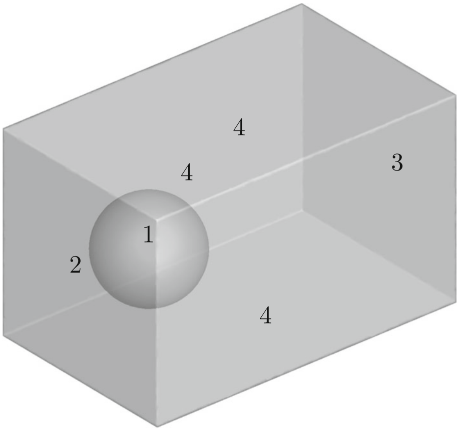

Fig. 22 Boundaries for test case 4.4

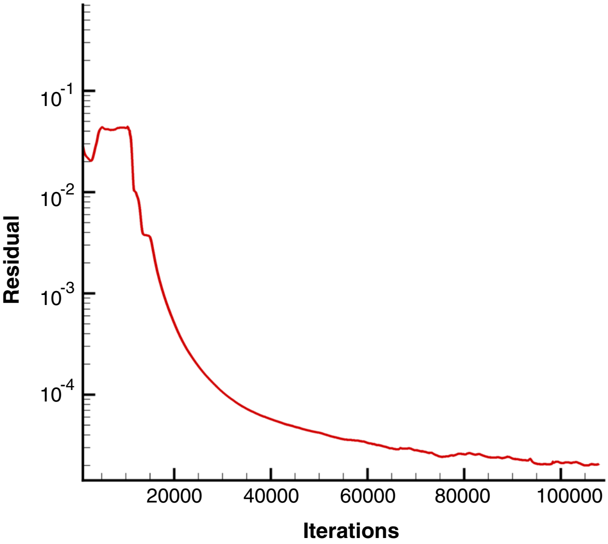

Fig. 23 Convergence of residual for test case 4.4

plan of discontinuity at $x = 0.09$, with $(\rho, u, v, w, p) = (8.0\ ,\ 8.25\ ,\ 0.0\ ,\ 0.0\ ,\ 116.5)$ at the left of the discontinuity, and $(\rho, u, v, w, p) = (1.4\ ,\ 0.0\ ,\ 0.0\ ,\ 0.0\ ,\ 1.0)$ at the right of the discontinuity.

As shown in Fig. 23, the residual converges well, even with only a very small number of $\mathbb{P}^1$ elements in the shock zone (see Fig. 24). Like in test case 4.3, with here a threshold $\varphi = 3$, we start with subdivided $\mathbb{P}^1$ elements everywhere. After 300 iterations, only elements in the shock zone remain $\mathbb{P}^1$, all the others become $\mathbb{P}^2$ elements. The converged solution (Mach number, pressure and density), is shown in respectively Figs. 25, 26, and 27.

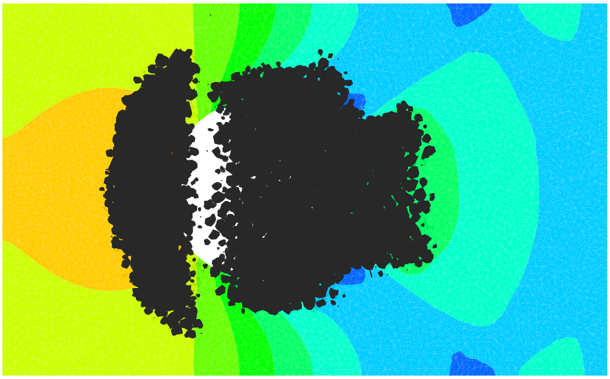

Fig. 24 Shock zone for test case 4.4 (with two dimensional slice cut at $y = 0$ of the pressure)

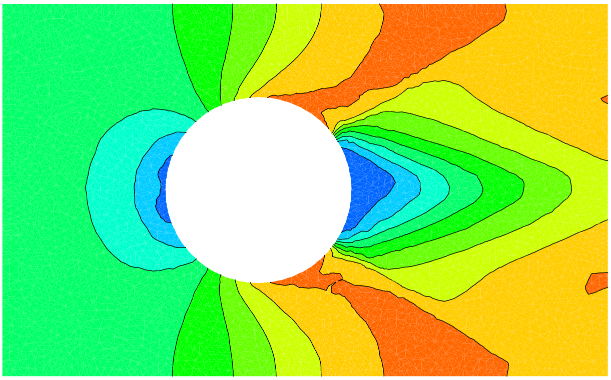

Fig. 25 Mach number isolines for test case 4.4 (two dimensional slice cut at $y = 0$)

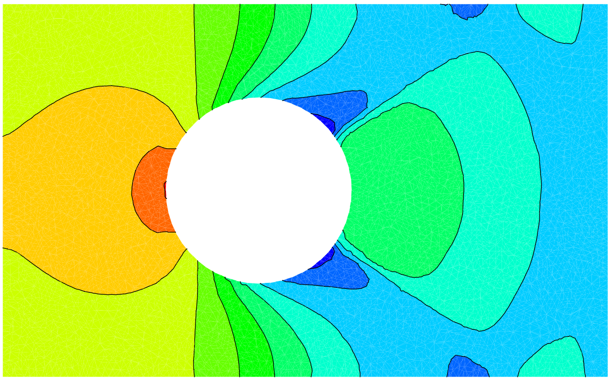

Fig. 26 Pressure isolines for test case 4.4 (two dimensional slice cut at $y = 0$)

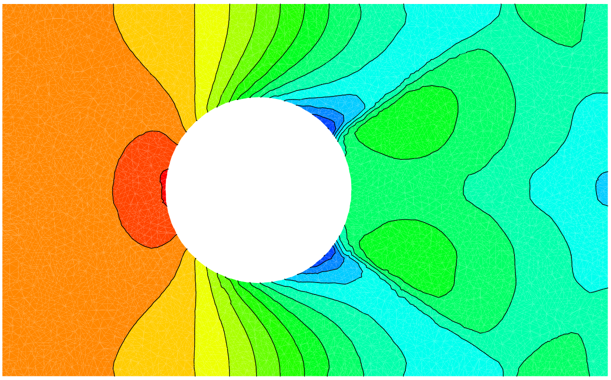

Fig. 27 Density isolines for test case 4.4 (two dimensional slice cut at $y = 0$)

5 Conclusion

We have described a way to use p-adaptation with continuous finite elements within the frame of residual distribution schemes. We have showed with general formulas that the method can be theoretically extended to arbitrary polynomial orders, and we have shown for complex problems modeled by the Euler equations that in practice, in the case of quadratic approximation in dimension two and three, the method is robust for subsonic, transonic, supersonic and hypersonic flows. Other applications can be envisaged and in particular, the extension of our method to the equations of Navier–Stokes coupled with the Penalization method will be the subject of a forthcoming paper.

Acknowledgements Q. V. has been funded by the French State via the French National Research Agency (ANR) in the frame of the “Investments for the future” Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02). All the numerical experiments presented in this paper were carried out using the cluster Avakas and the PLAFRIM experimental testbed. PLAFRIM is developed under the Inria PlaFRIM development action with support from LABRI, IMB and other entities: Conseil Régional d’Aquitaine, FeDER, Université de Bordeaux and CNRS. R. A. has been partialy funded with the SNF Grant # 200021_153604.

Appendix 1: Implicit Numerical Solver

We need to solve the system of equations (20), written in a compact way as:

$$R(\mathbf{u}_h) = 0. \tag{60}$$

This problem is first relaxed as:

$$\frac{d\mathbf{u}_h}{dt} = -R(\mathbf{u}_h). \tag{61}$$

To approximate this time derivative, we use the backward Euler formula, and the problem becomes:

$$\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t^n} = -R(\mathbf{u}_h^{n+1}),\ n = 0, 1, 2, \ldots \tag{62}$$

Thus, we have to solve a non linear problem at each time step n. We use for that a Newton method, that when applied to (60) reads:

$$\mathbf{u}_h^{k+1} = \mathbf{u}_h^k - J^{-1} R(\mathbf{u}_h^k), \quad k = 0, 1, 2, \tag{63}$$

where

$$J = \frac{\partial R(\mathbf{u}_h^k)}{\partial \mathbf{u}} \tag{64}$$

is the Jacobian of R. In practice, the Jacobian J is computed from a first order scheme with the stabilization term. And so, with our Newton method applied to (62) , the linear system we have to solve at each time step n reads:

$$[\frac{1}{\Delta t^n} + J(\mathbf{u}_{h_k}^n)]\Delta \mathbf{u}_{h_k}^n = -R(\mathbf{u}_{h_k}^n) \tag{65}$$

$$\mathbf{u}_{h_{k+1}}^n = \mathbf{u}_{h_k}^n + \Delta \mathbf{u}_{h_k}^n, \quad k = 0, 1, 2 \ldots \tag{66}$$

In practice, for each time step n, we use only one iteration on k, which seems enough to reach convergence of the nodal residuals to the zero machine.

Appendix 2: Comparison with Constrained Approximation for the Finite Element Method

In [8] the authors present a mesh adaptation method called “constrained approximation“ in the frame of a Petrov–Galerkin finite-element method. The idea behind the method is to use a non regular mesh (non conforming and of various order) and to constrain the value of the discrete solution at the interface shared by elements of different size (h-adaptation) and different order (p-adaptation) to keep the solution continuous across the interface.

In Fig. 28 inspired from [8], the authors give an example of h-refinement and p-refinement. In the h-refinement case, the two triangles share a common edge with two different sizes, which generates a hanging node in the middle of the edge. In the p-refinement case, the triangles are of two different polynomial orders ($\mathbb{P}1$ and $\mathbb{P}2$) and so a hanging node is generated in the middle of the edge. In both cases, the solution is constrained at the hanging node so that it remains continuous.

The authors use a hierarchical basis of shape functions [24]. For example, in dimension two this basis is constituted of functions associated with the vertices, the edges and the element and are called accordingly vertex, edge and bubble functions. They are built incrementally with respect to the desired order of the approximation, by using kernel functions in the definition of the edge and bubble functions. The main advantages of this basis over the classical Lagrange basis functions, is that the polynomial order of the approximation can be changed easily (by adding or removing kernel functions) and in the case of p-adaptation, like in Fig. 28, the hanging node is naturally suppressed, by changing to one the polynomial order of the shape function associated with the edge at the interface in the $\mathbb{P}_2$ element.

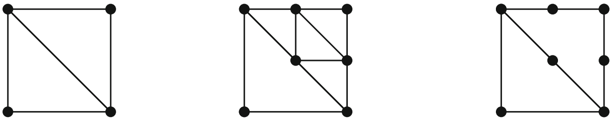

Fig. 28 From *left* to *right*: initial mesh, h-adaptation and p-adaptation

Once the constrains are established, the next step is to apply these constrains. The authors present a complete set of rules to impose the continuity of the solution in the case of h-adaptation and *p*-adaptation. The authors suggest to follow a rule called "1-irregularity" rule (only one hanging node between two elements) in order to avoid too complex situations, at least for h-adaptation.

We believe that this method is very different from the method we have proposed in this paper. The first difference is that the generation of hanging nodes makes the mesh non conforming. In a RD scheme, the question of how this hanging node should be treated does not seem evident. The distribution schemes (like for example (21) and (35)) would probably have to be redefined to take into account the hanging nodes, so that the scheme remains conservative (which is an essential ingredient for the convergence) and still exhibit the same behavior (for example (non) diffusive or (non) oscillatory). Even if it is still possible to use a classical Lagrange basis, it is advised in [8] to use a hierarchical basis of functions, as such a basis is more suited for *p*-adaptation. So, the compatibility of a hierarchical basis function with RD schemes should be analyzed. Finally, there is the question as to whether the hanging nodes should be taken into account in the system of equations (20), and for the implementation of an implicit RD scheme, what the consequences would be for the computation of the Jacobian matrix. We are not saying that the constrained approximation method is not compatible with RD schemes (indeed, we have obtained some preliminary numerical results with a prototype of a RD scheme based upon the constrained approximation method, in the case of scalar equations with Lagrange elements and an explicit scheme). But the questions we have raised should first be carefully assessed before trying to make an RD scheme compatible with the adaptive method of [8]. To the contrary, the *p*-adaptive method for RD schemes we have proposed here is designed for RD schemes by construction and as such avoids all the problems evoked above.

References

1. Abgrall, R.: Toward the ultimate conservative scheme: following the quest. J. Comput. Phys. **167**(2), 277–315 (2001)
2. Abgrall, R.: Essentially non-oscillatory residual distribution schemes for hyperbolic problems. J. Comput. Phys. **214**(2), 773–808 (2006)
3. Abgrall, R.: Two remarks on the approximation of hyperbolic problems by finite element methods. In: Jang S.Y., Kim Y.R., Lee D.-W., and Yie I. (eds.) Proceeding of the International Congress of Mathematicians, vol. IV, pp. 699–726. KM Kyong Moon SA (2014)
4. Abgrall, R., Barth, T.: Residual Distribution schemes for conservation laws via adaptive quadrature. SIAM J. Sci. Comput. **24**(3), 732–769 (2003)
5. Abgrall, R., Larat, A., Ricchiuto, M.: Construction of very high order Residual Distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. J. Comput. Phys. **230**(11), 4103–4136 (2011)
6. Abgrall, R., Roe, P.L.: High order fluctuation schemes on triangular meshes. J. Sci. Comput. **19**(1–3), 3–36 (2003)
7. Abgrall, R., Trefilik, J.: An example of high order residual distribution scheme using non-lagrange elements. J. Sci. Comput. **45**(1–3), 3–25 (2010)
8. Ahrabi, B.R., Anderson, W.K., Newman, J.C.: High-order finite-element method and dynamic adaptation for two-dimensional laminar and turbulent Navier–Stokes. In: AIAA paper 2014-2983. 32nd AIAA Applied Aerodynamics Conference (2014)
9. Babuška, I., Guo, B.Q.: The h, p and $h-p$ version of the finite element method; basis theory and applications. Adv. Eng. Softw. **15**(3–4), 159–174 (1992)
10. Cangiani, A., Chapman, J., Georgoulis, E.H., Jensen, M.: Implementation of the continuous-discontinuous Galerkin finite element method. In: Numerical mathematics and advanced applications 2011, Proceedings

of ENUMATH 2011, the 9th European conference on numerical mathematics and advanced applications, Leicester, UK, September 5–9, 2011. pp. 315–322. Springer, Berlin (2013)

11. Cheng, J., Shu, C.W.: High order schemes for cfd: a review. Chin. J. Comput. Phys. **5**, 002 (2009)
12. Deconinck, H., Paillere, H., Struijs, R., Roe, P.L.: Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. Comput. Mech. **11**(5–6), 323–340 (1993)
13. Deconinck, H., Struijs, R., Bourgois, G., Roe, P.L.: Compact advection schemes on unstructured grids. In: Computational Fluid Dynamics, VKI LS 1993-04. SEE N94-18557 04-34 (1993)
14. Demkowicz, L.: Computing with hp-ADAPTIVE FINITE ELEMENTS: Volume 1 One and Two Dimensional Elliptic and Maxwell Problems. CRC Press, Boca Raton (2006)
15. Di Pietro, D.A., Ern, A.: Mathematical Aspects of Discontinuous Galerkin Methods, vol. 69. Springer Science & Business Media, New York (2011)
16. Ekaterinaris, J.A.: High-order accurate, low numerical diffusion methods for aerodynamics. Prog. Aerosp. Sci. **41**(3), 192–300 (2005)
17. Hartmann, R.: Discontinuous Galerkin methods for compressible flows: higher order accuracy, error estimation and adaptivity. Lecture Series—Von Karman Institute for Fluid Dynamics **1**, 5 (2006)
18. Karniadakis, G.E., Shu, C.W., Cockburn, B.: Discontinuous Galerkin Methods: Theory, Computation and Applications. Springer, New York (2000)
19. Mitchell, W.F., McClain, M.A.: A comparison of hp-adaptive strategies for elliptic partial differential equations. ACM Trans. Math. Softw. (TOMS) **41**(1), 2 (2014)
20. Moxey, D., Green, M.D., Sherwin, S.J., Peiró, J.: An isoparametric approach to high-order curvilinear boundary-layer meshing. Comput. Methods Appl. Mech. Eng. **283**, 636–650 (2015)
21. Ni, Ron-Ho: A multiple-grid scheme for solving the euler equations. AIAA J. **20**(11), 1565–1571 (1982)
22. Ricchiuto, M.: Contributions to the development of residual discretizations for hyperbolic conservation laws with application to shallow water flows. Université Sciences et Technologies - Bordeaux I, Habilitation à diriger des recherches (2011)
23. Shahbazi, K., Fischer, P.F., Ethier, C.R.: A high-order discontinuous Galerkin method for the unsteady incompressible Navier–Stokes equations. J. Comput. Phys. **222**(1), 391–407 (2007)
24. Solin, P., Segeth, K., Dolezel, I.: Higher-Order Finite Element Methods. CRC Press, Boca Raton (2003)
25. Toro., E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction. 2nd ed., 2nd edn. Springer, Berlin (1999)
26. Wang, Z.J.: High-order methods for the Euler and Navier–Stokes equations on unstructured grids. Prog. Aerosp. Sci. **43**(1), 1–41 (2007)
27. Wang, Z.J.: High-order computational fluid dynamics tools for aircraft design. Phil. Trans. R. Soc. A. **372**, 20130318 (2014). doi:10.1098/rsta.2013.0318
28. Wang, Z.J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., et al.: High-order cfd methods: current status and perspective. Int. J. Numer. Methods Fluids **72**(8), 811–845 (2013)