Isogeometric analysis for compressible flows using a Discontinuous Galerkin method

Régis Duvigneau

To cite this version:

Régis Duvigneau. Isogeometric analysis for compressible flows using a Discontinuous Galerkin method. Computer Methods in Applied Mechanics and Engineering, Elsevier, 2018, 333, 10.1016/j.cma.2018.01.039 . hal-01589344v3

HAL Id: hal-01589344
https://hal.inria.fr/hal-01589344v3

Submitted on 1 Feb 2018

Isogeometric analysis for compressible flows using a Discontinuous Galerkin method

R. Duvigneau

Université Côte d'Azur, INRIA, CNRS, LJAD
INRIA Sophia-Antipolis, 2004 route des Lucioles - B.P. 93, 06902 Sophia Antipolis, France

Abstract

The objective of this work is to investigate a Discontinuous Galerkin (DG) method for compressible Euler equations, based on an isogeometric formulation: the partial differential equations governing the flow are solved on rational parametric elements, that preserve exactly the geometry of boundaries defined by Non-Uniform Rational B-Splines (NURBS), while the same rational approximation space is adopted for the solution. We propose a new approach to construct a DG-compliant computational domain based on NURBS boundaries and examine the resulting modifications that occur in the DG method. Some two-dimensional test-cases with analytical solutions are considered to assess the accuracy and illustrate the capabilities of the proposed approach. The critical role of boundary curvature is especially investigated. Finally, a shock capturing strategy based on artificial viscosity and local refinement is adapted to this isogeometric context and is demonstrated for a transonic flow.

Keywords: isogeometric analysis; compressible flows; Discontinuous Galerkin; high-order scheme

Email address: Regis.Duvigneau@inria.fr (R. Duvigneau)

1. Introduction

Computation Fluid Dynamics (CFD) has now a significant impact in several domains, from transportation industry, chemical process engineering to biofluidics. The increase of computational facilities has significantly reduced the computational costs related to CFD, and thus promoted industrial applications. Nevertheless, the time necessary to set up CFD simulations is still significant in several cases because pre-processing tasks, like mesh generation from Computer-Aided Design (CAD) data, still rely on time-consuming human actions.

This verdict motivated the emergence of the so-called *isogeometric analysis* approach ten years ago, which targeted the seamless integration of Finite-Element (FE) methods and CAD representations [1]. It advocates the use of CAD bases like Non-Uniform Rational B-Splines (NURBS) in variational formulations, not only to describe the geometry, but also to define the approximation space. This approach is conceptually appealing and has several interesting properties from technical point of view: it allows to achieve simulations based on exact geometries [2] (i.e. identical to the geometry defined in the CAD framework); it facilitates the development of fully automated tools from CAD design to performance analysis for optimization without geometry approximate [3, 4]; it permits to include geometry-preserving mesh refinement procedures [5], etc. This approach has been a significant success, as reflected in the growing number of publications related to the topic. However, one can observe that a very large majority of problems addressed by the isogeometric analysis method concerns elliptic or parabolic partial differential equations, mainly with applications in structural mechanics. Only a few investigations can be found in CFD [6, 7, 8], mainly regarding incompressible viscous flows in laminar regime. Indeed, most CFD methods for convection-dominant problems rely on Finite-Volume (FV) or Discontinuous Galerkin (DG) formulations, which are more suited to hyperbolic conservation laws.

Besides, in the CFD community, several recent studies pointed out the fact that a piecewise linear representation of the geometry may yield convergence and accuracy issues [9, 10], in the context of high-order DG or FV schemes, which are more and more employed in the context of complex problems (e.g. for Large Eddy Simulations). As shown by some authors, the use of a high-order geometry description is beneficial, not only in terms of solution accuracy, but also in terms of computational efficiency, because it avoids performing excessive mesh refinement to describe curved boundaries [11].

Accounting for curvilinear boundaries requires (*i*) to construct a curvilinear mesh ; (*ii*) to adapt discretization methods. Regarding mesh generation, some works are currently in progress to extend classical meshing techniques to produce curvilinear elements in the vicinity of boundaries [12, 13, 14, 15]. However, they are often restricted to polynomial representations, thus CAD geometry is not exactly preserved. Alternatively, the CAD community has been very active in the last years in proposing new algorithms to construct high-order computational domains from CAD boundaries, either in a structured environment [16, 17, 18, 19, 20, 21], or in an unstructured triangular or tetrahedral context [22, 23, 24, 25, 26]. Concerning the adaption of PDE solvers to curvilinear geometries, the isoparametric approach has existed for a long time [27], but its basic application yields only an approximate of the CAD geometry, based on polynomials. As results, points at the boundary of the grid do not lie exactly on the CAD geometry, normals and derivatives are not preserved. More advanced approaches have been proposed recently, to fully integrate NURBS geometries in DG solvers, independently from the approximation space [11, 10, 28]. The demonstrated results are convincing, in terms of accuracy and convergence, but the proposed methodology suffers from the complexity of the underlying mappings used for spatial integration.

Therefore, this work aims at proposing a DG method for CFD computations, that relies on a fully isogeometric formulation, i.e. the PDEs governing the flow are solved on rational parametric elements, that preserve exactly the CAD geometry at the boundaries, while the approximation space is defined thanks to the same rational bases originating from CAD. Obviously, NURBS basis functions classically used in CAD cannot be employed in a straightforward way as test functions for a DG formulation, because they do not naturally exhibit a common support enabling discontinuities in the solution. Therefore, the first part of the current work will explain how to generate a set of rational elements that both preserve the CAD geometry at the boundary and are defined on basis functions suitable to DG methods. We also envisage the case where the geometry of the problem is not defined using a CAD environment. Then, we will describe how to modify a classical DG formulation to account for the new integration supports, with the concern of introducing as few modifications as possible. Indeed, an objective of this work is to provide a computational framework as simple as possible, that permits to account for CAD boundaries without approximation, and maintains the implementation complexity reasonable. Then, a set of two-dimensional test-cases commonly used to validate CFD methods will be studied, in order to assess the numerical properties of the proposed approach. In particular the critical role of the boundary curvature is investigated. Finally, we will consider the issue of shock capturing in this isogeometric context. A classical approach for high-order DG, based on artificial viscosity [29], is adapted to the rational representation of the solution. The proposed method is associated with a strategy to refine locally the rational elements in the vicinity of the shock. The efficiency of the resulting procedure will be demonstrated for a transonic flow.

Advantages and limitations of the proposed methodology will be discussed in conclusion.

2. NURBS representations

NURBS curves (and surfaces) are now considered as standard to define geometries in CAD [30]. In particular, they allow to represent exactly a broad class of geometric curves like conic sections. Therefore, the use of such a representation for the construction of the computational grid would be of great interest in CFD, at least to avoid introducing geometrical errors in the solving procedure, and more generally to improve the integration of CFD in design procedures.

In the present study, we aim at using a unique representation for both the geometry and the approximation space, in the spirit of the isogeometric analysis methods. In this perspective, a necessary task consists in generating a curvilinear grid based on NURBS boundaries, as well as an approximation space suitable to DG method. In this section, the definition and relevant properties of NURBS functions are first provided, and different approaches to construct the curvilinear grid and approximation space are then proposed.

2.1. NURBS functions

The one-dimensional case is presented first, the extension to multi-dimensional problems being exposed further. A NURBS basis is defined using the so-called *knot* vector $\Xi = (\xi_1, \ldots, \xi_l) \in \mathbb{R}^l$, which consists of nondecreasing l real numbers. This knot vector defines a discretization of the *parametric domain* $\widehat{\Omega} = [\xi_1, \xi_l]$. Open knot vectors, i.e., knot vectors with first and last knots of multiplicity $p+1$, are usually used for a curve of degree p to impose interpolation and tangency conditions at both extremities [31]. Therefore $\xi_1 = \ldots = \xi_{p+1}$ and $\xi_{n+1} = \ldots = \xi_{n+p+1}$. NURBS functions [32] are rational extensions of B-Spline functions [31] $(N_i^p)_{i=1,\cdots,n}$ that are defined recursively as:

$$N_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi). \tag{2}$$

Note that the quotient $0/0$ is assumed to be zero. According to this definition, B-Spline functions of degree 0 and 1 coincide with classical piecewise constant and linear FE basis functions. However, B-Spline functions of higher degree differ from classical Lagrange polynomials. We underline that this hierarchical construction is practically convenient and promotes the use of p-adaptive strategies. The degree of the functions p, the number of knots l and functions n are related by the relation $l = n + p + 1$. NURBS functions of degree p are then defined by:

$$R_i^p(\xi) = \frac{w_i N_i^p(\xi)}{\sum_{j=1}^n w_j N_j^p(\xi)}, \tag{3}$$

where $w_i \in \mathbb{R}$ is the weight associated to the ith function. Therefore, NURBS functions with uniform weights are actually B-Spline (and polynomial) functions. The use of non-uniform weights allows to generate rational functions to represent conic sections.

The transformation of the parametric domain $\widehat{\Omega}$ to the *physical domain* Ω can now be introduced:

$$\mathcal{F} : \widehat{\Omega} \to \Omega, \quad \xi \mapsto x(\xi) = \sum_{i=1}^n R_i^p(\xi) X_i. \tag{4}$$

The transformation associates a *control point* X_i to each basis function and defines a NURBS curve, that lies in a space of dimension N. For general geometric modeling purpose, N is usually higher than one (for instance, the curve may lie in $\mathbb{R}^2$ or $\mathbb{R}^3$). But in the perspective of using NURBS domains to describe the computational domain, we will consider the simpler case where the NURBS domain lies in a space of dimension equal to the number of parameters. Note that NURBS curves can be interpreted as the projection of a B-Spline curve that lies in a space of dimension $N+1$, the additional coordinate being defined by the weights associated to the control points [32]. To summarize, this representation allows to describe the domain Ω as a NURBS *patch*, defined by its control points, the associated weights, the knot vector and the degree of the basis.

An important property of the NURBS representation is the capability to insert a new knot, and thus a new basis function, without altering the geometry [31]. It can be considered as a local h-refinement procedure [2]. Any NURBS curve defined by the knot vector $\Xi = (\xi_1, \dots, \xi_l) \in \mathbb{R}^l$, n basis functions and control points, as described above, can be identically represented using the knot vector $(\xi_1, \dots, \xi_q, \bar{\xi}, \xi_{q+1} \dots, \xi_l) \in \mathbb{R}^{l+1}$, that includes an additional knot $\bar{\xi}$ inserted between ξ_q and ξ_{q+1}. If we consider first the case of a B-Spline curve, the new representation can be written as:

$$x(\xi) = \sum_{i=1}^{n} N_i^p(\xi) X_i = \sum_{i=1}^{n+1} N_i^p(\xi) \bar{X}_i, \tag{5}$$

with a new set of control points defined as:

$$\bar{X}_i = (1-\alpha_i) X_{i-1} + \alpha_i X_i \quad \alpha_i = \begin{cases} 1 & \text{if } i \leq q-p \\ \frac{\bar{\xi}-\xi_i}{\xi_{i+p}-\xi_i} & \text{if } q-p+1 \leq i \leq q \\ 0 & \text{if } i \geq q+1 \end{cases} \tag{6}$$

The extension to NURBS curves relies on the projection approach mentioned above [32]. It is important to precise a particular case: if a new knot is inserted at an existing knot, the regularity of the curve is decreased. More generally, the curve at the knot ξ_q has the regularity C^{p-r}, where r is the multiplicity of the knot q [31]. Therefore, if one inserts p knots at an existing knot, the curve is preserved but divided in two independent parts. This property will be used in a forthcoming section to define an approximation space suitable to DG formulation.

2.2. Multi-dimensional case

NURBS surfaces and volumes can be generated from the one-dimensional case by using a tensorial representation. For the sake of clarity, we consider a two-dimensional example with identical degrees in both directions:

$$\mathbf{x}(\xi,\eta) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} R^p_{i_1\, i_2}(\xi,\eta) \mathbf{X}_{i_1 i_2} = \sum_{i=1}^{n_1 \times n_2} R_i(\xi,\eta) \mathbf{X}_i, \tag{7}$$

$$R^p_{i_1\,i_2}(\xi,\eta) = \frac{w_{i_1\,i_2}\;N^p_{i_1}(\xi)\;N^p_{i_2}(\eta)}{\sum_{j_1=1}^{n_1}\sum_{j_2=1}^{n_2} w_{j_1\,j_2} N^p_{j_1}(\xi) N^p_{j_2}(\eta)}, \tag{8}$$

where $\mathbf{x} = (x, y)$ are the cartesian coordinates and $(\mathbf{X}_{i_1 i_2})_{i_1=1,\cdots,n_1\;i_2=1,\cdots,n_2}$ is the lattice of control points. $R_i(\xi,\eta) = R^p_{i_1\,i_2}(\xi,\eta)$ represents the two-parameter basis function of index $i = i_1 \times i_2$. This construction is illustrated in Fig. (1), for a cubic patch ($p = 3$) with a 4×4 control point lattice ($n_1 = n_2 = 4$), in the particular case $\xi_1 = \xi_2 = \xi_3 = \xi_4 = 0$ and $\xi_5 = \xi_6 = \xi_7 = \xi_8 = 1$ (same knot vector for η). Note that this study is restricted to two-dimensional problems, but the proposed methodology can be extended to three dimensional cases. In particular, the procedure described above concerning knot insertion for curves can be easily extended to surfaces and volumes, by applying the procedure to the different parameters independently. This tensorial extension is certainly the easiest way to generate surfaces and volumes, but other techniques could be envisaged, for instance based on rational triangles by using barycentric coordinates. Therefore, the proposed methodology is not restricted to "structured" representations based on tensorial products, but "unstructured" representations could be envisaged as well, as recently exposed in [28].

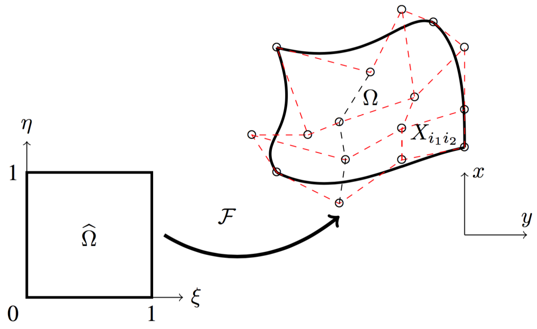


Figure 1: NURBS representation

2.3. Construction of the grid and DG-compliant approximation space

We consider an arbitrary physical domain $\Omega \subset \mathbb{R}^2$ with the boundary $\partial\Omega$. According to the isogeometric paradigm, we adopt a NURBS representation for the whole physical domain. For the sake of simplicity, we assume in this preliminary work that the domain can be represented by only one patch, as illustrated in Fig. (1). For problems involving a more complex geometry, a multi-patch representation should be adopted. This point is discussed more in depth in concluding section.

Two different cases should now be addressed regarding the construction of the curvilinear grid: firstly, we will make the assumption that the boundary is defined as a set of NURBS curves. In a second time, for the sake of completeness, we will consider the alternate case, for which boundaries are not defined in a CAD environment.

2.3.1. CAD-based boundary

We suppose here that the boundary of the physical domain is defined as a set of four compatible NURBS curves, i.e. opposite curves have the same degree and the same number of control points. If not, basic CAD procedures like *degree elevation* and *knot insertion* should be applied first [32]. Thus, one only has to determine the inner control point coordinates to define a NURBS patch that matches the physical domain exactly. This task can be achieved by using the so-called *discrete Coons patch* construction [33]. It consists in defining inner control point lattice $(\mathbf{X}_{i_1 i_2})_{i_1=2,\cdots,n_1-1\ i_2=2,\cdots,n_2-1}$, and associated weights, from boundary control points according to the following expression:

$$\begin{aligned}\mathbf{X}_{i_1 i_2} =&(1-\frac{i_1-1}{n_1-1})\mathbf{X}_{1 i_2}+\frac{i_1-1}{n_1-1}\mathbf{X}_{n_1 i_2}\\ &+(1-\frac{i_2-1}{n_2-1})\mathbf{X}_{i_1 1}+\frac{i_2-1}{n_2-1}\mathbf{X}_{i_1 n_1}\\ &-\begin{pmatrix}1-\frac{i_1-1}{n_1-1} & \frac{i_1-1}{n_1-1}\end{pmatrix}\begin{pmatrix}\mathbf{X}_{11} & \mathbf{X}_{1n_2}\\ \mathbf{X}_{n_1 1} & \mathbf{X}_{n_1 n_2}\end{pmatrix}\begin{pmatrix}1-\frac{i_2-1}{n_2-1}\\ \frac{i_2-1}{n_2-1}\end{pmatrix}.\end{aligned} \tag{9}$$

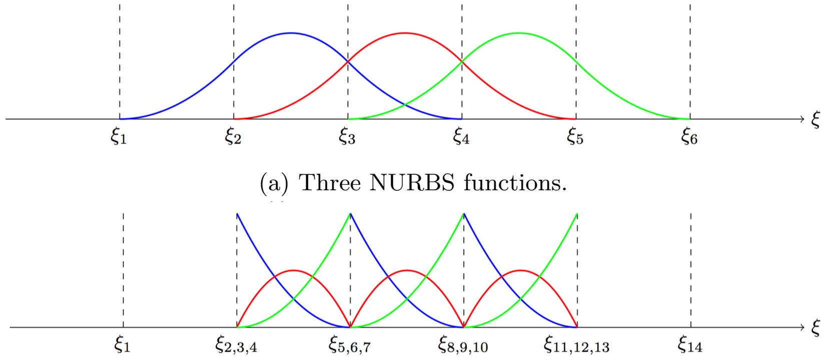
As shown in [17], this construction does not ensure that the transformation $\mathcal{F}$ is injective. However, satisfactory results are obtained in practice, provided that the boundary of the domain does not exhibit too high curvature areas. Note that more sophisticated construction methods can be employed [18], for which a sufficient condition exists, that prevents overlap.

The construction proposed above allows to define a NURBS patch, that exactly matches the boundary of physical domain. However, the underlying NURBS functions are not suitable to DG method, since it relies on piecewise rational functions defined over the whole patch. Therefore, the patch has to be transformed into a set of elements, and associated basis functions, which can exhibit discontinuities at the interfaces between elements, without altering the geometry. This task can be achieved thanks to the *knot insertion procedure*, described in section 2.1. Indeed, as explained above, if p knots are inserted at existing inner knots, the original NURBS patch is divided to a set of $(n_1 - p) \times (n_2 - p)$ patches, each of them being actually a *rational Bézier patch* of degree p. The geometry of the physical domain is unchanged, but each patch is now defined according to its own basis of size $(p + 1) \times (p + 1)$, which enables the generation of discontinuities at the interfaces. Therefore, we will consider these rational Bézier patches as elements in a DG method. To summarize, the approach to generate a curvilinear grid and the associated approximation space, from a set of four NURBS curves is the following:

1. Insert new knots and elevate basis degree (if necessary) to obtain a set of compatible boundary curves ;

2. Apply discrete Coons patch construction procedure described by Eq. (9) to define the physical domain as a single NURBS patch ;
3. Insert p knots at each existing inner knot to generate $(n_1 - p) \times (n_2 - p)$ independent Bézier patches.

The proposed procedure is illustrated in Fig. (2) for a one-dimensional case. Three NURBS functions of degree $p = 2$ are plotted in Fig. (2a), each of them relying on a support composed of $p+1 = 3$ knot intervals. The knot insertion procedure applied twice for each existing inner knot yields nine quadratic functions, each of them with a support restricted to one knot interval. Two-dimensional examples will be provided further, in the section 4 devoted to applications.



(a) Three NURBS functions.

(b) Nine rational Bézier functions obtained by multiple knot insertions.

Figure 2: Procedure to transform a single NURBS patch to a set of DG-compliant rational Bézier elements.

2.3.2. General case

We examine now the alternate case: if boundary curves cannot be exactly represented as NURBS curves, approximate representations have to be used. In that case, the exact boundary representation property is lost, but the benefit of using a high-order boundary description remains. In the proposed approach, the approximation of the boundary is achieved locally, after an initial discretization of the boundary curves. Thus, it can be considered as an extension of classical piecewise linear grid generation methods. More precisely, the proposed approach is composed of the following steps:

1. Sample opposite curves with N_1 and N_2 points respectively ($\mathbf{x}_q$), located on the exact boundary ;
2. Approximate each curve delimited by two points $\mathbf{x}_q$ and $\mathbf{x}_{q+1}$ by a rational Bézier curve $\mathbf{x}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{X}_i = \mathbf{R}(\xi)^\top \mathbf{X}$;
3. Construct interior domain by the discrete Coons patch method.

The second step is achieved by solving a set of $2 \times (N_1 + N_2 - 2)$ local linear least-squares problems:

$$\mathbf{X} = \text{argmin}\ \frac{1}{2} \int_{\mathbf{x}_q}^{\mathbf{x}_{q+1}} \|\mathbf{R}(\xi)^\top \mathbf{X} - \mathbf{x}^\star(\xi)\|^2 d\xi, \tag{10}$$

where $\mathbf{x}^\star$ denotes the coordinates of a point on the exact boundary, between x_q and x_{q+1}. As result of the third step, one obtains a set of $(N_1 - 1) \times (N_2 - 1)$ rational Bézier elements, which are similar to those obtained by the previous approach but approximate of the exact boundary. Again, a two-dimensional example is provided in the section 4 devoted to applications.

2.4. Local refinement

The ability to refine locally the computational domain (and the solution) is critical to accurately capture complex phenomena, like shocks in aerodynamics. As explained above, any NURBS patch can be easily refined by knot insertion, but the tensorial nature of NURBS representations prevents such a refinement to be really local. This drawback has motivated the development of more sophisticated representations, like T-Splines [16] that allow T-junctions between elements.

In the present context, the computational domain is composed of independent rational elements. Thus, it is straightforward to use again the knot insertion procedure described by (5 - 6) to split an element in several elements of the same degree, independently from its neighbors. In this way, the refinement is really local. This can be achieved in an isotropic way (i.e. in both parametric directions) or in an anisotropic manner (in a single direction), and preserves exactly the geometry. A generic refinement procedure based on error estimation is out of the scope of the present work. Nevertheless, refinement will be employed for shock capturing purpose (see section 4.3).

3. Discontinuous Galerkin method

3.1. Discretization

We consider the two-dimensional compressible Euler equations, that can be written in the conservative form as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \vec{F} = 0, \tag{11}$$

where $\mathbf{W}$ are the conservative flow variables $(\rho, \rho u, \rho v, \rho e)$, with ρ the density, $\vec{U} = (u, v)$ the velocity vector and e the total energy per unit of mass. $\vec{F} = (\mathbf{F}_x(\mathbf{W}), \mathbf{F}_y(\mathbf{W}))$ is the vector of the inviscid fluxes. The pressure p is obtained from the perfect gas state equation:

$$p = \rho(\gamma - 1)(e - \frac{1}{2}\|\overrightarrow{U}\|^2) = \rho(\gamma - 1)e_i \tag{12}$$

where γ is the ratio of the specific heat coefficients and e_i the internal energy. We denote $\mathbf{U}$ the primitive flow variables (ρ, u, v, p). The inviscid fluxes are given by:

$$\mathbf{F}_x(\mathbf{W}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u (e + \frac{p}{\rho}) \end{pmatrix} \quad \mathbf{F}_y(\mathbf{W}) = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v (e + \frac{p}{\rho}) \end{pmatrix}. \tag{13}$$

The DG method [34] is derived from a weak formulation of the problem, in the proposed isogeometric context, obtained by multiplying Eq. (11) by an arbitrary rational Bézier function $R_k(\xi, \eta)$ and integrating over the rational Bézier patch Ω_j:

$$\int_{\Omega_j} \frac{\partial \mathbf{W}}{\partial t} R_k \, d\Omega + \int_{\Omega_j} \nabla \cdot \vec{F}(\mathbf{W}) \, R_k \, d\Omega = 0. \tag{14}$$

After integration by parts, one obtains classically:

$$\int_{\Omega_j} \frac{\partial \mathbf{W}}{\partial t} R_k \, d\Omega - \int_{\Omega_j} \vec{F}(\mathbf{W}) \cdot \nabla \vec{R}_k \, d\Omega + \int_{\partial\Omega_j} \vec{F}(\mathbf{W}) \cdot \vec{n} \, R_k \, d\Gamma = 0. \tag{15}$$

Since the solution is *a priori* discontinuous at the interfaces, the normal flux is evaluated by a numerical flux function $F^\star(\mathbf{W}^+, \mathbf{W}^-, \vec{n})$, defined according to the values of the solution that prevail at each side of the interface and the local unit vector $\vec{n}$ directed outwards. Several flux functions are classically used in DG methods [35, 36]. In this work, the HLLC flux is employed [37, 38].

Finally, the solution on each element Ω_j, denoted $\mathbf{W}|_j$, is expressed in the rational Bézier basis of degree p chosen to describe the geometry of the element:

$$\mathbf{W}|_j(\xi, \eta) = \sum_{i_1=1}^{p+1} \sum_{i_2=1}^{p+1} R^p_{i_1\, i_2}(\xi, \eta) \mathbf{W}_{i_1 i_2} = \sum_{i=1}^{(p+1)^2} R_i(\xi, \eta) \mathbf{W}_i. \tag{16}$$

By substitution and expressing integrals in the parametric domain, one obtains:

$$\sum_{i=1}^{(p+1)^2} \frac{\partial \mathbf{W}_i}{\partial t} \int_{\widehat{\Omega}_j} R_i \, R_k \, |J_\Omega| \, d\widehat{\Omega} = \int_{\widehat{\Omega}_j} \vec{F}(\mathbf{W}) \cdot \nabla \vec{R}_k \, |J_\Omega| \, d\widehat{\Omega} - \int_{\partial\widehat{\Omega}_j} F^\star(\mathbf{W}^+, \mathbf{W}^-, \vec{n}) \, R_k \, |J_\Gamma| \, d\widehat{\Gamma}. \tag{17}$$

$|J_\Omega|$ represents the determinant of the Jacobian matrix of the geometric transformation $\mathcal{F}$, defined according to Eq. (7):

$$J_\Omega = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix} = \sum_{i=1}^{(p+1)^2} \begin{pmatrix} \frac{\partial R_i}{\partial \xi}(\xi, \eta) X_i & \frac{\partial R_i}{\partial \eta}(\xi, \eta) X_i \\ \frac{\partial R_i}{\partial \xi}(\xi, \eta) Y_i & \frac{\partial R_i}{\partial \eta}(\xi, \eta) Y_i \end{pmatrix}, \tag{18}$$

and $|J_\Gamma|$ is the lineal counterpart, defined thanks to Eq. (4):

$$|J_\Gamma| = \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2} = \sqrt{\left(\sum_{i=1}^{p+1} \frac{\partial R_i}{\partial \xi}(\xi) X_i\right)^2 + \left(\sum_{i=1}^{p+1} \frac{\partial R_i}{\partial \xi}(\xi) Y_i\right)^2} \tag{19}$$

The gradient of the basis functions in the cartesian frame, that appears in Eq. (17), are evaluated by using the transposed of the inverse of the Jacobian matrix:

$$\nabla \vec{R}_k = J_\Omega^{-\top} \begin{pmatrix} \frac{\partial R_k}{\partial \xi}(\xi, \eta) \\ \frac{\partial R_k}{\partial \eta}(\xi, \eta) \end{pmatrix} \tag{20}$$

Obviously, one identifies easily a mass matrix on the left-hand side of Eq. (17), volumic residuals and interface fluxes on the right-hand side. The local mass matrix is inverted once and its inverse is stored, an explicit four-step Runge-Kutta method being used for time integration.

Remark 1. One should underline that the spatial integrals in Eq. (17) have to be evaluated carefully. Indeed, the non-linearities in the flow variables and in the geometrical transformations necessitate high-order quadratures, depending on the basis degree. For the present work, classical Gauss-Legendre quadrature rules have been applied, but involving a large number of evaluation points when required.

Remark 2. For unsteady problems, the initial flow solution has to be expressed accordingly to the approximation space selected, i.e. in terms of discontinuous rational Bézier functions, as in Eq. (16). In practice, this task is achieved by solving a set of local least-squares fitting problems.

Remark 3. Boundary conditions are imposed via the normal flux at boundary, which is computed using the numerical flux function. For inlet and outlet boundaries, the exterior state is imposed, whereas for wall boundaries a reflective condition is imposed by choosing the following exterior state:

$$\rho_{ext} = \rho_{int} \quad u_{ext} = u_{int} - 2(\vec{U} \cdot \vec{n})\, n_x \quad v_{ext} = v_{int} - 2(\vec{U} \cdot \vec{n})\, n_y \quad e_{ext} = e_{int}. \tag{21}$$

Remark 4. The visualization of the solution and the curvilinear grid is not straightforward, because most visualization software are limited to piecewise linear or quadratic solution. In the current work, we use the GLVis tool (`http://glvis.org`), which permits to visualize NURBS elements and solution fields.

3.2. Shock capturing

The ability to capture shocks is critical for aerodynamic applications. We describe in this section how to adapt to rational representations the sub-cell capturing approach proposed in [29] for classical DG methods. It relies on the classical concept of explicit artificial viscosity, introduced to eliminate the oscillations in the solution.

More precisely, physical diffusive terms are added to the Euler equations, with a viscosity value ϵ taken to be constant over each element and determined by the following smooth function:

$$\epsilon = \begin{cases} 0 & \text{if } s_e < s_0 - \kappa \\ \frac{\epsilon_0}{2}\left(1 + \sin\frac{\pi(s_e - s_0)}{2\kappa}\right) & \text{if } s_0 - \kappa < s_e < s_0 + \kappa \\ \epsilon_0 & \text{if } s_e > s_0 + \kappa \end{cases} \tag{22}$$

where ϵ_0 is the maximum viscosity value, s_e a sensor of the oscillations in the solution and (s_0, κ) numerical parameters that control the viscosity distribution. As analyzed in [29], the structure of the shock imposes that ϵ_0 scales like the shock thickness, i.e. $\mathcal{O}(h/k)$. In the present study, we simply set $\epsilon_0 = d/(p+1)$, where d is the diameter of the current element. According to a Fourrier analysis, the amplitude of high-frequencies should decay like $\mathcal{O}(1/k^2)$. Therefore, s_0 is set to the value $1/(p+1)^2$. The parameter κ is chosen equal to s_0 in order to add diffusive effects as soon as oscillations are detected. It remains to define the sensor s_e. In the original method [29], it is based on the $\mathcal{L}^2$ norm of the highest term in the polynomial expansion of the solution, in each element. In the isogeometric context, we propose to define the sensor by quantifying the oscillations in the control point lattice that represents the solution in each element. More precisely, one evaluates the following quantities, that compare the total variation of control points in each direction to the variations observed between patch extremities (we suppose here that $p > 1$):

$$s_\xi^{i_2} = \left(\sum_{i_1=1}^{p} |\mathbf{W}_{i_1+1\,i_2} - \mathbf{W}_{i_1 i_2}|\right) - |\mathbf{W}_{p+1\,i_2} - \mathbf{W}_{1 i_2}| \tag{23}$$

$$s_\eta^{i_1} = \left(\sum_{i_2=1}^{p} |\mathbf{W}_{i_1 i_2+1} - \mathbf{W}_{i_1 i_2}|\right) - |\mathbf{W}_{i_1\,p+1} - \mathbf{W}_{i_1 1}| \tag{24}$$

Note that these quantities vanish if the solution is monotonic over the element. If not, it is a measure of the oscillatory nature of the solution. The sensor for a given element is finally defined as:

$$s_e = \frac{1}{2(p+1)}\left(\sum_{i_2=1}^{p+1} s_\xi^{i_2} + \sum_{i_1=1}^{p+1} s_\eta^{i_1}\right)\frac{1}{\overline{\overline{\mathbf{W}}}} \tag{25}$$

where $\overline{\overline{\mathbf{W}}}$ is the mean solution over the element. In the illustration presented below, we choose to apply this sensor to the density field. Once the viscosity is known for each element, the diffusive terms are discretized using the original formulation from Bassy and Rebay [9], which is based on auxiliary variables for the flow derivatives and centered diffusive fluxes.

The last methodological ingredient consists in refining the computational domain in the vicinity of the shock location, to obtain a sharp shock capture.

As explained above, any element can easily be split in two elements by multiple knot insertion in one direction. Therefore, we employ again Eqs. (23 - 24) as anisotropic refinement criteria: if one of these quantities exceeds a threshold value, the element is split in the direction which exhibits the major oscillations. Regarding the resolution of the PDE system, refinement only affects the computation of inter-element fluxes, which should account for the new configurations.

4. Applications

4.1. Isentropic vortex

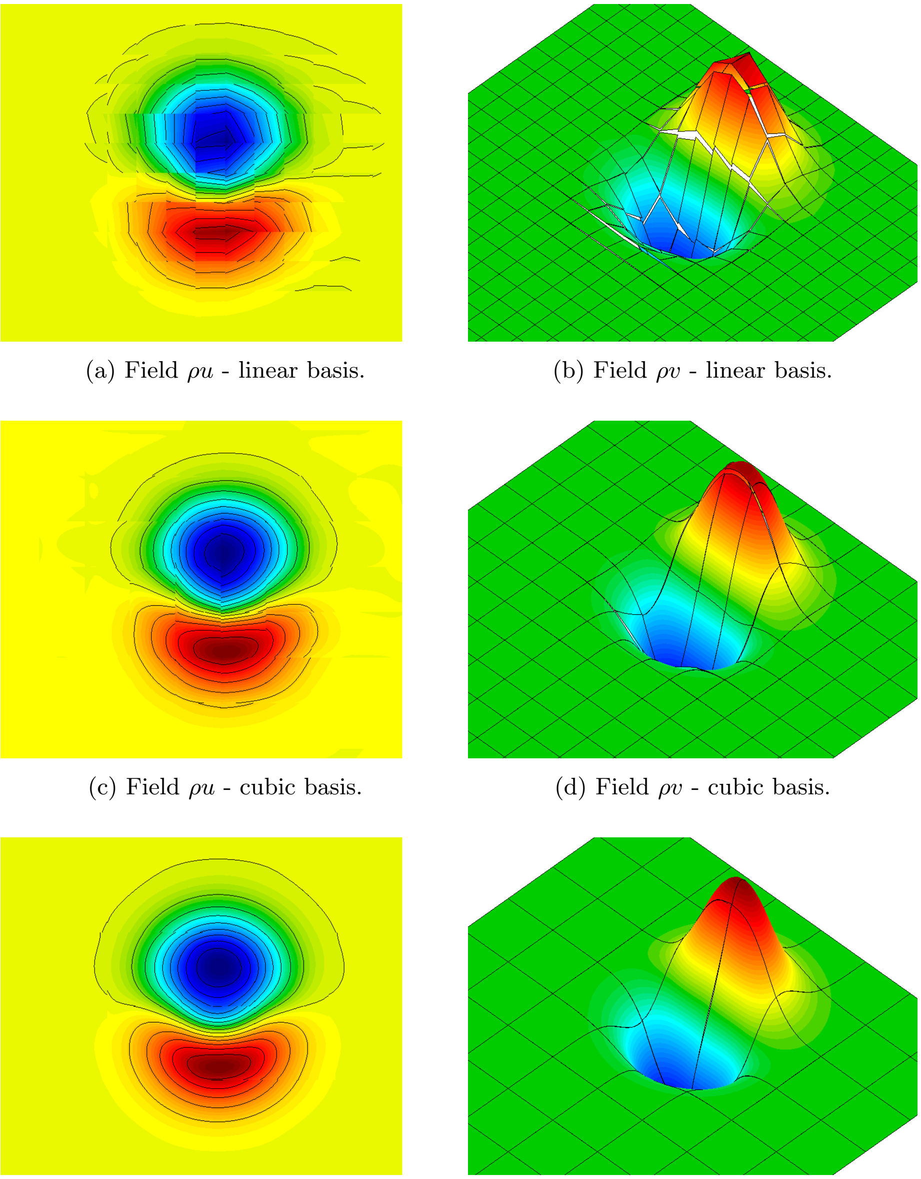
As first problem, we consider the transport of an isentropic vortex [34], whose analytical solution is provided by:

$$\begin{aligned}
\rho &= \left(1 - \frac{\gamma - 1}{16\gamma\pi^2}\beta^2 e^{2(1-r^2)}\right)^{\frac{1}{\gamma-1}}, \\
u &= 1 - \beta e^{1-r^2}\ \frac{y - y_0}{2\pi}, \\
v &= \beta e^{1-r^2}\ \frac{x - t - x_0}{2\pi}, \\
p &= \rho^\gamma,
\end{aligned} \tag{26}$$

with $r = \sqrt{(x-t-x_0)^2 + (y-y_0)^2}$, $x_0 = 5$, $y_0 = 0$ and $\beta = 5$. The physical domain Ω is the square $[0, 10] \times [0, 10]$. The analytical solution is used to define the initial condition and the exterior state for boundary fluxes. Error is computed at final time $T = 2$. This problem is used to assess the intrinsic properties of the proposed scheme, independently from the boundary representation.

A first accuracy study is conducted on the basis of cartesian nested grids. In this particular case, uniform NURBS weights are employed, yielding actually Bézier elements. Some solutions obtained by using the same number of degrees of freedom, but different bases, are shown in Fig. (3). More precisely, one can compare the fields ρu and ρv computed with a linear basis (20×20 elements of degree 1), a cubic basis (10×10 elements of degree 3) and a quartic basis (8×8 elements of degree 4), corresponding to a total of 1600 degrees of freedom in all cases. The benefit of using high-order representations is obvious, when observing the smoothness of the numerical solution. The $\mathcal{L}^2(\Omega)$ norm of the error for the energy field is depicted in Fig. (4a), for sequences of nested grids corresponding to bases ranging from linear to quintic (degree 5). An optimal convergence rate is observed.

Then, some tests are carried out to quantify the impact of changes in the grid characteristics. More specifically, the following configurations are considered: *(i)* grid with stretching (aspect ratio 10) ; *(ii)* grid with skewness (angle 45°) ; *(iii)* grid with random perturbations of vertices (uniform distribution of amplitude $h/5$) ; *(iv)* grid with smooth deformation (60° solid rotation with exponential damping). Note that, for the later case, the geometry of the elements is no longer linear and only a C^0 continuity is observed at the interfaces.



(a) Field ρu - linear basis.

(b) Field ρv - linear basis.

(c) Field ρu - cubic basis.

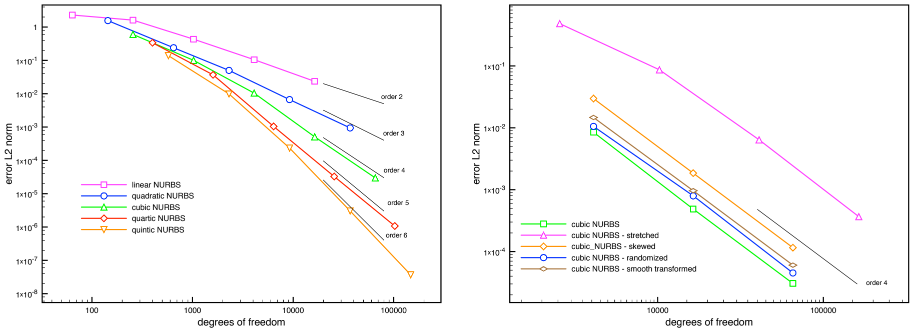
(d) Field ρv - cubic basis.

(e) Field ρu - quartic basis.

(f) Field ρv - quartic basis.

Figure 3: Solution fields for different bases but the same number of degrees of freedom, for vortex case.

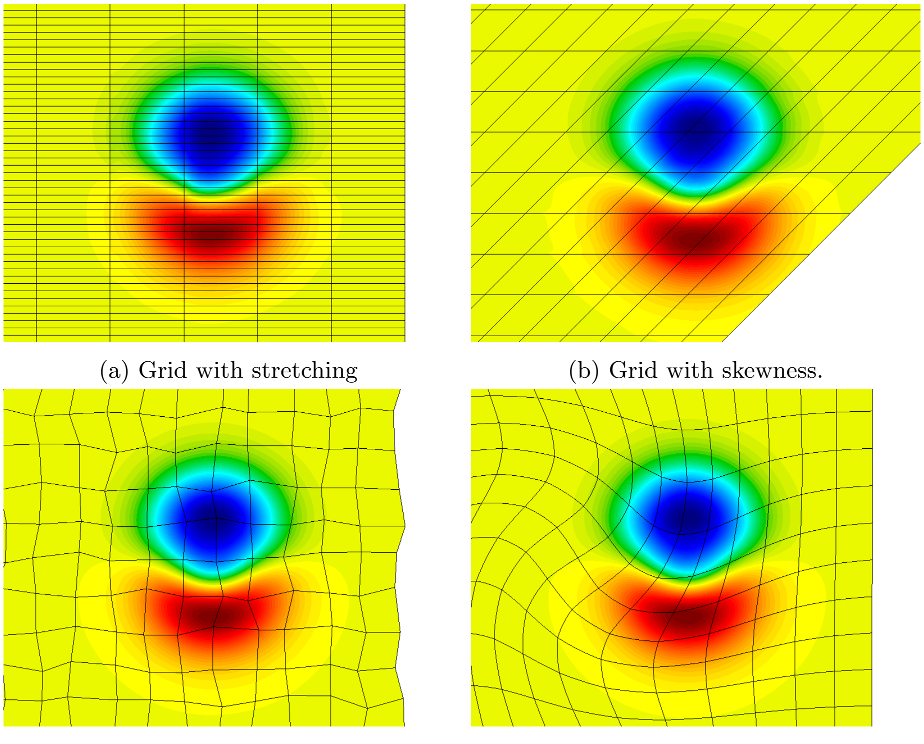
The resulting grids and solutions can be seen in Fig. (5). The evaluation of the error, for cubic bases, yields again an optimal convergence rate whatever the distortions applied, as can be seen in Fig. (4b).



(a) Cartesian grids.

(b) Grids with different distorsions.

Figure 4: Accuracy study for vortex case (energy field).



(a) Grid with stretching.

(b) Grid with skewness.

(c) Grid with random perturbations.

(d) Grid with smooth deformation.

Figure 5: Field ρu for vortex case for different grid distorsions (cubic bases).

4.2. Flow between cylinders

We now investigate the flow between two quarters of cylinders of respective radius one and four, to quantify the influence of curvilinear wall boundaries [34]. The targeted analytical solution writes:

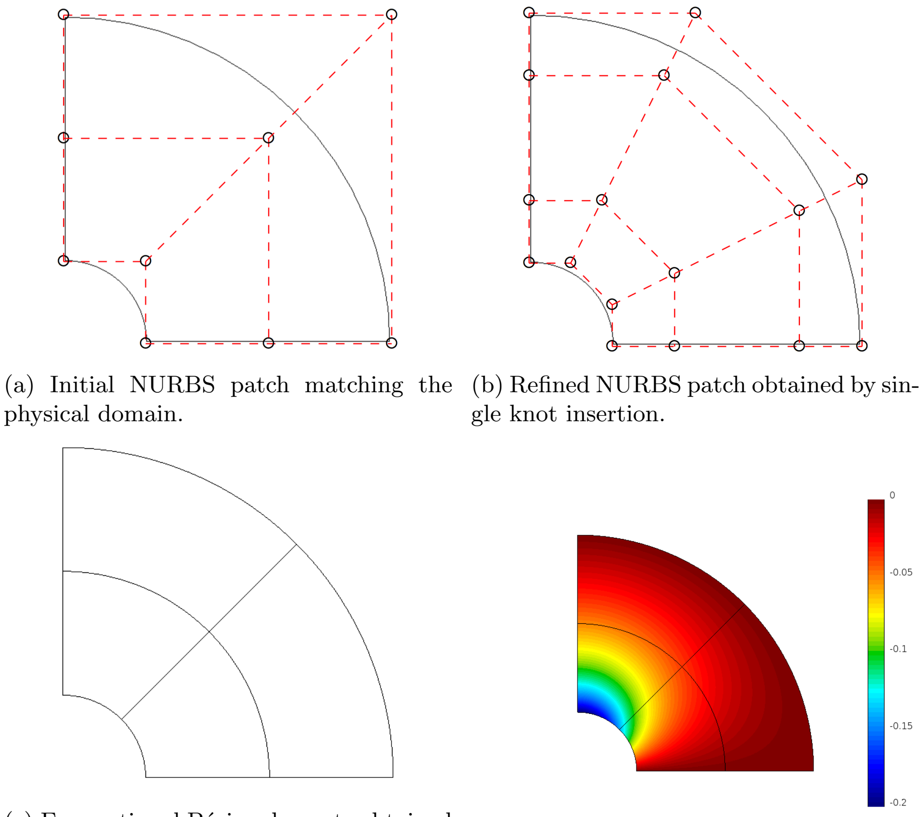
$$
\begin{aligned}
&\rho = 1,\\
&u = -\sin(\theta)\; u_\theta,\\
&v = \cos(\theta)\; u_\theta,\\
&p = 1 + \frac{1}{75^2}\left(\frac{r^2}{2} - 32\, ln(r) - \frac{128}{r^2}\right),
\end{aligned}
\tag{27}
$$

where $u_\theta = \frac{1}{75}(-r + \frac{16}{r})$, $r = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}(\frac{y}{x})$. The analytical solution is provided as initial condition and as exterior state for inlet and outlet fluxes. Reflective wall conditions are prescribed on the two cylinders. Time integration is achieved until convergence, based on the monitoring of the $\mathcal{L}^2(\Omega)$ norm of the error for all solution components.

The different steps of the construction of the grid are illustrated in Fig. (6). The physical domain can be defined exactly as a single quadratic NURBS patch, whose boundaries and control point lattice can be seen in Fig. (6a). Therefore, the procedure described in section 2.3.1 is used to construct the grid. The initial quadratic NURBS patch counts 3×3 control points, the knot vectors for both parameters being simply $\Xi = (0, 0, 0, 1, 1, 1)$. The insertion of a new knot at $\xi = 0.5$ for both parameters allows to refine the parameterization by adding some control points in the patch, as illustrated in Fig. (6b). Two additional knots are then inserted at $\xi = 0.5$, yielding a set of 2×2 quadratic rational Bézier elements, as shown in Fig. (6c). Each element counts 3×3 control points. Finally, the solution is computed using a quintic basis (thanks to the degree elevation property [2], the quadratic elements can be represented exactly using bases of higher degree). The first momentum field is depicted in Fig. (6d).

The procedure described above is used to generate some sequences of grids, for bases ranging from linear to quintic. As for the first problem, we examine the evolution of the $\mathcal{L}^2(\Omega)$ norm of the error for the energy with respect to the number of degrees of freedom, to assess the scheme accuracy. Results are plotted in Fig. (7a). Optimal convergence rates are obtained again. One can underline the significant gap between the errors obtained for linear and quadratic bases, which is certainly due to the exact boundary definition used in the latter case. In the linear case, the geometry of cylinders is only approximated.

Then, these computations are carried out again, but using straight boundaries instead of curvilinear ones, for bases ranging from quadratic to quintic. The corresponding grids have been constructed by setting element vertices first, and imposing linear distributions of control points in each element then, with uniform weights. The evolution of the error for the energy field with respect to the number of degrees of freedom can be seen in Fig. (7b). Surprisingly, all solutions exhibit a convergence rate close to the value two. One can even observe that bases of degree four and five yield a worse convergence rate than a linear



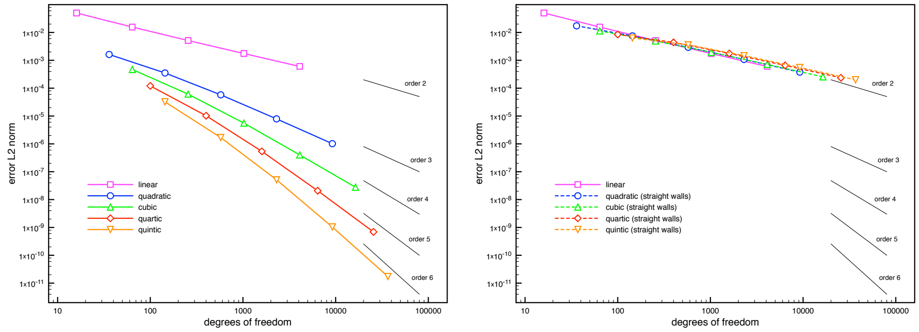
(a) Initial NURBS patch matching the physical domain.

(b) Refined NURBS patch obtained by single knot insertion.

(c) Four rational Bézier elements obtained by multiple knot insertion.

(d) ρu field (quintic basis).

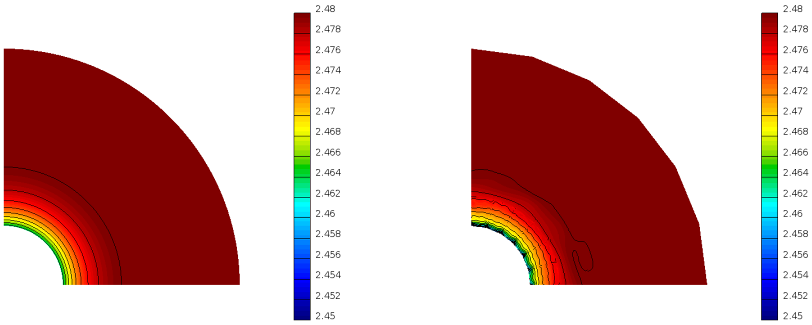
Figure 6: Illustration of grid construction for cylinder case.



(a) Curvilinear boundaries.

(b) Straight boundaries.

Figure 7: Accuracy study for cylinder case (energy field).



(a) Curvilinear boundaries.

(b) Straight boundaries.

Figure 8: Energy field for cylinder case (quintic bases, 6 × 6 elements).

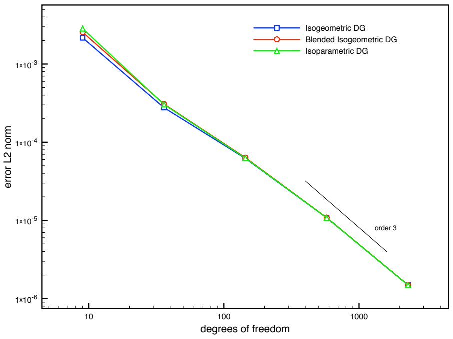


Figure 9: Comparison between different methods with curved domains (quadratic bases).

basis. The reason of the phenomenon can be easily understood by comparing the solution fields obtained for a quintic basis, depicted in Fig. (8), using grids composed of 6×6 curvilinear and straight elements. In the former case, circular iso-value contours are obtained as expected. In contrast, the use of straight elements leads to an erroneous solution in the elements in the vicinity of the cylinder, in which a rarefaction wave appears at each vertex. As the degree of the representation increases in each element, the effects of these waves are amplified. This phenomenon has been also observed in other works [10]. These results demonstrate the critical role of geometry curvature when high-order schemes are used.

Finally, we compare the proposed method with two other approaches, that can deal with curved computational domains: the classical isoparametric DG approach [27], which uses polynomial representations for both the geometry and the solution, and the Blended Isogeometric Discontinuous Galerkin method (BIDG) [28], which is based on rational elements and mapped polynomial solutions. The comparison is restricted to the quadratic case. For the isoparametric method, the geometry is not exactly represented, but approximated by piecewise quadratic curves. On the contrary, the proposed method and the BIDG approach rely on the same exact geometry, but different representations of the solution fields. Fig. (9) shows the evolution of the error norm of the density field as the computational domain is refined. As seen, all methods exhibit a third order accuracy asymptotically. The discrepancy between the three methods is not really significant here and the gain of accounting for the exact geometry can be observed only for extremely coarse grids. These results underline the fact that the use of CAD-based computational domains should not be motivated by the accuracy but more by the ease of interaction between CAD and simulation tools.

4.3. Ringleb problem

The third test-case considered in this work concerns the Ringleb flow problem [39]. It deals with a transonic flow in a nozzle, that evolves from subsonic to supersonic conditions, and then back to subsonic state without shock. It is well known in the community that accounting for curved boundaries is critical for this test-case, to achieve both a full convergence to steady state and a shock-free flow solution [39].

The analytical solution is defined parametrically using two parameters. The parameter k is constant on each streamline, the inner and outer wall boundaries corresponding respectively to $k_{max} = 1.5$ and $k_{min} = 0.7$. The second parameter q represents the velocity magnitude. On each streamline, q varies between $q_{min} = 0.5$ at inlet and outlet boundaries and k. For each pair (k, q), one defines the sound speed a, the density ρ, the pressure p and an intermediate quantity J by:

$$\begin{aligned}
a &= \sqrt{1 - \frac{\gamma - 1}{2} q^2}, \\
\rho &= a^{\frac{2}{\gamma-1}}, \\
p &= \frac{1}{\gamma} a^{\frac{2\gamma}{\gamma-1}}, \\
J &= \frac{1}{a} + \frac{1}{3a^3} + \frac{1}{5a^5} - \frac{1}{2} \log\left(\frac{1+a}{1-a}\right).
\end{aligned} \tag{28}$$

The geometry is also defined in terms of (k, q) parameters:

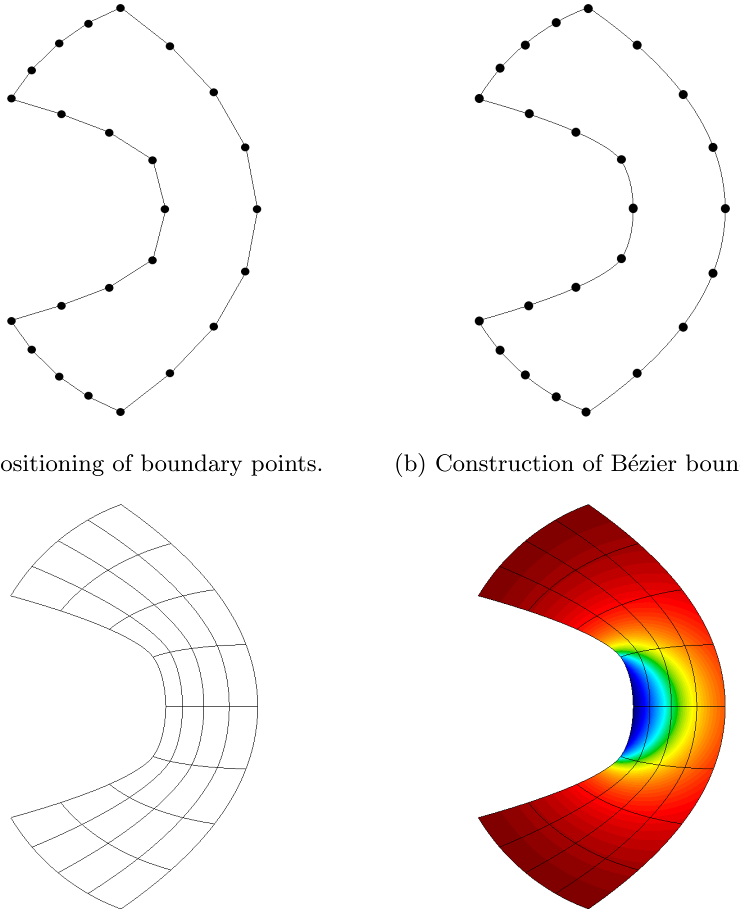
$$\begin{aligned}
x &= \frac{1}{2\rho}\left(\frac{2}{k^2} - \frac{1}{q^2}\right) - \frac{J}{2}, \\
y &= \pm \frac{1}{k\rho q}\sqrt{1 - \left(\frac{q}{k}\right)^2}.
\end{aligned} \tag{29}$$

The flow is isentropic and irrotational, and reaches a supersonic velocity of Mach number 1.5 at location $y = 0$ of the inner wall. Contrary to previous problems, the solution is not known explicitly: for each point (x, y) of the physical domain, one should first determine (by a numerical procedure) the corresponding parameters (k,q) thanks to Eq. (29), and then evaluate the solution fields using Eq. (28).

The geometry of the problem cannot be defined in a NURBS-based environment. Thus, the general procedure described in section 2.3.2 is employed to construct the computational grid, as illustrated in Fig. (10). Thanks to Eq. (29), some points are set on the boundary, with a regular distribution, as shown in Fig. (10a). A set of least-squares approximation problems are then solved to fit the boundary using Bézier curves (uniform weights are used). This is illustrated by Fig. (10b), in the case of quintic bases. At this step, control points are known for all boundaries, which allows to define interior control points by discrete Coons patch construction. The resulting grid and flow solution are shown in Fig. (10c) and (10d) respectively.

As for the previous problem, the exact solution is used as initial condition and as exterior state for boundary fluxes. Time integration is achieved until convergence of all error components. The evolution of the error for the energy and the entropy with respect to the number of degrees of freedom, in $\mathcal{L}^2(\Omega)$ norm, are plotted in Fig. (11), for bases ranging from linear to quintic and for two different boundary configurations. Indeed, very different results are obtained depending on the choice of boundary conditions. If one specifies the exact solution as exterior state for all boundary fluxes, the convergence to a steady state is obtained for all grids and bases tested. However, if reflective conditions are prescribed at inner and outer walls, convergence difficulties have been encountered. It has been found that only cases relying on quartic bases at least, and a sufficiently refined grid can converge to a steady state. Only such converged cases are reported in Fig. (11). As can be seen, accounting for walls yields a small increase of entropy error, but a more significant increase of energy error, mainly through kinetic energy term. These results are actually in agreement with those obtained by Yano & Darmofal [40], who established that entropy error should remain below a threshold value of about 10^{-5} to obtain a satisfactory convergence, when wall conditions are employed. On Fig. (12), the density fields computed using wall conditions, for linear and quartic bases, are compared. The linear grid counts 40×80 elements whereas the quartic grid is based on 16×32 elements, yielding the same number of degrees of freedom (12800). For the quartic case, a smooth and fully converged solution is found, but the linear case exhibits an artificial shock in formation and flow oscillations in the vicinity of the inner wall in compression region. In the latter case, a screenshot at an arbitrary time is shown in Fig. (12a).

Regarding the convergence rates, a close observation of Fig. (11) shows that convergence rates close to optimal values are obtained for the entropy error, whereas lower values are observed for the energy error. A possible explanation coud be related to the error computation: for energy, an additional error is committed because the analytical solution at given (x,y) coordinates is computed via a numerical search for the corresponding (k,q) parameters. This could be dommageable for the finest cases, for which the local error is found very close to machine accuracy.



(a) Positioning of boundary points.

(b) Construction of Bézier boundaries.

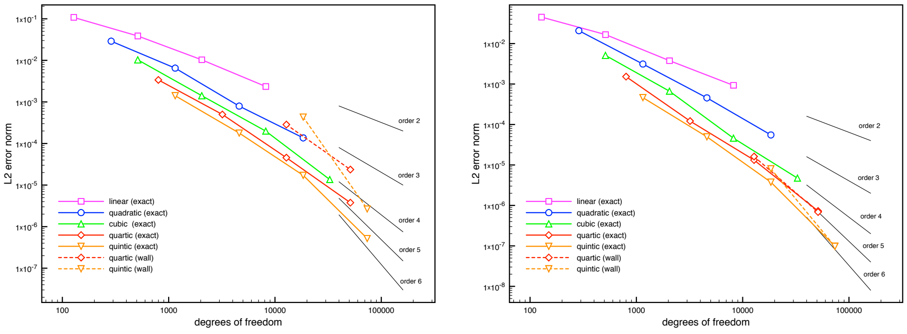
(c) Construction of Bézier patches.

(d) Energy solution field.

Figure 10: Illustration of grid construction for Ringleb problem.

4.4. Transonic flow over a bump

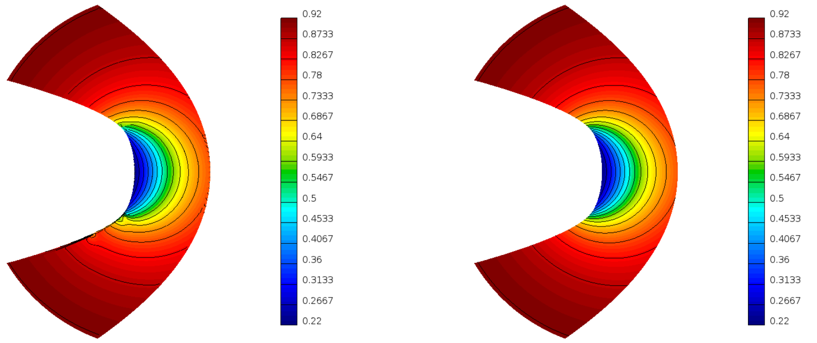
To illustrate the shock capturing capabilities of the proposed approach, we consider now the simulation of the transonic flow over a bump. The computational domain is defined by a cubic NURBS patch, of length $l = 8$ and height $h = 1$. The symmetric bump of amplitude $20\%h$ is located between -1 and 1. Far field conditions are prescribed at all boundaries, except the bottom one for which a slip wall condition is imposed. The inlet flow corresponds to a Mach number of value 0.7, yielding the generation of a steady shock wave on the bump.



(a) Energy error.

(b) Entropy error.

Figure 11: Accuracy study for Ringleb case.



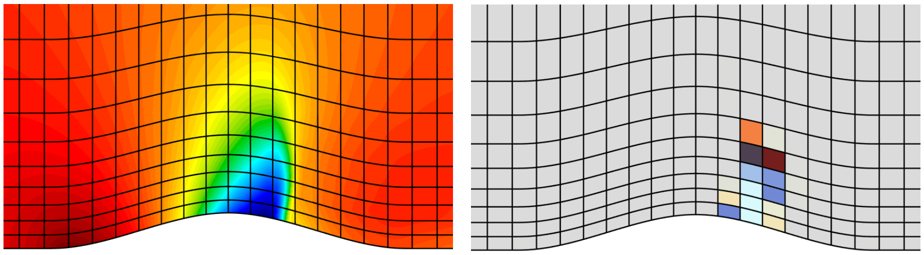
(a) Linear boundaries (not converged).

(b) Quartic boundaries.

Figure 12: Density field for Ringleb case.

A first cubic grid of size 32 × 24, without local refinement, is constructed by splitting of the initial NURBS patch. This computational domain is obviously too coarse for an accurate capture of the shock. The density and viscosity fields can be seen in Fig. (13). One can see that the artificial viscosity is non-zero only in a few elements in the vicinity of the shock and no viscosity is added to the elements further away. Moreover, the shock thickness is lower than the element size, as expected by using this sub-cell strategy, yielding a shock capture in a single element.

An automated refinement procedure is then implemented, as described in section 3.2, starting from an even coarser grid of size 24 × 16. We underline again that this refinement is achieved without any geometrical modification. The density fields and the refined grids can be seen in Fig. (14), at refinement cycles two, four and six. The grid is refined only in the shock neighborhood, yielding an accurate capture of the solution discontinuity. These results highlight the flexibility of the proposed DG method to handle shocks.



(a) Density field.

(b) Viscosity field.

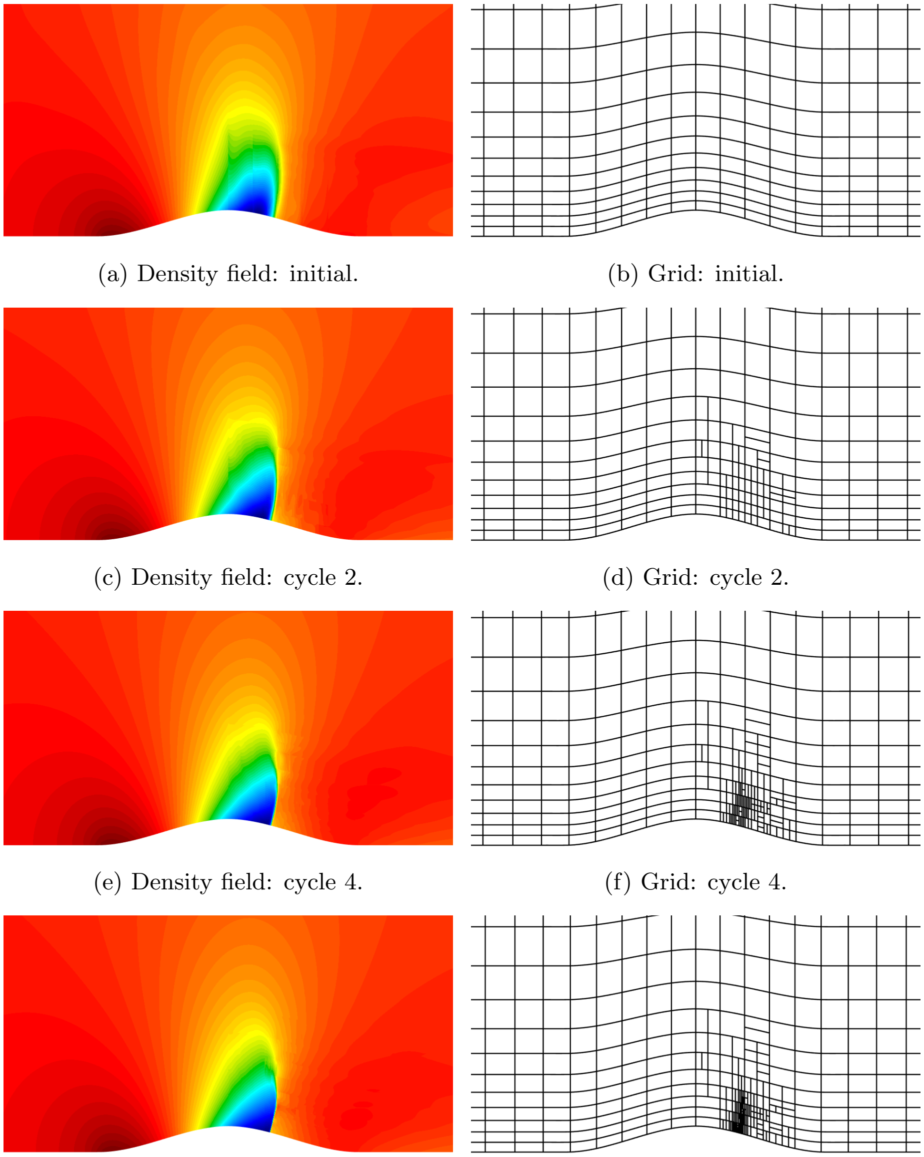
Figure 13: Transonic flow over a bump: results without refinement.

5. Discussion and conclusion

In this work, we have explained how to construct a DG-compliant curvilinear grid based on a set of rational Bézier elements, from boundaries defined by NURBS curves. The proposed method allows both to preserve the CAD geometry and solve governing equations with a DG method including only a few modifications. As underlined in the results obtained for different test-cases, the resolution scheme exhibits a high-order accuracy with quasi-optimal convergence rate, and seems to be robust with respect to mesh distorsions. The benefit of accounting for boundary curvature has been demonstrated. Finally, a classical shock capturing strategy based on artificial viscosity has been adapted to the isogeometric context. The oscillations of the control point lattice in the solution are employed to define the local viscosity. In association with a local refinement strategy, the method has proved to provide a sharp shock capture.

Obviously, as shown by several authors, other approaches could be employed successfully to enable high-order definitions of the geometry in CFD codes. Nevertheless, we consider that the use of a unique representation for both CAD and CFD environments is a true advantage, in particular when more complex problems are considered, for which geometry and analysis are strongly coupled, as aerodynamic shape optimization or fluid-structure interactions.

The main weakness of the proposed approach concerns the lack of efficient and flexible tools to construct a suitable curvilinear grid, that would allow the study of more complex problems. In particular, when it is not possible to represent the physical domain as a single NURBS patch, the use of a multipatch construction is targeted (before the generation of elements in each patch as proposed by the current work). However, the splitting of the physical domain into a set of compatible NURBS patch is not straightforward and cannot be performed easily with existing CAD software. A possible alternative is to extend the proposed approach to rational triangular Bézier elements, as used in the BIDG method [28]. Indeed, several works have been recently reported on the construction of grids based on such elements [22, 23, 24, 25, 26]. An other way would be to mix high-order parametric elements near boundaries with linear



(a) Density field: initial.

(b) Grid: initial.

(c) Density field: cycle 2.

(d) Grid: cycle 2.

(e) Density field: cycle 4.

(f) Grid: cycle 4.

(g) Density field: cycle 6.

(h) Grid: cycle 6.

Figure 14: Transonic flow over a bump: results with local refinement.

elements (possibly triangular) elsewhere. Note that this hybrid representation should be beneficial from computational point of view also, but at the cost of a complexity overhead in the implementation. Finally, the refinement properties of NURBS patches, illustrated in the last application, could also be employed to construct iteratively the computational domain in a quad-tree framework.

[1] J. Cottrell, T. Hughes, Y. Bazilevs, Isogeometric analysis : towards integration of CAD and FEA, John Wiley & sons, 2009.

[2] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement, Computer Methods in Applied Mechanics and Engineering (194) (2005) 4135–4195.

[3] W. A. Wall, M. A. Frenzel, C. Cyron, Isogeometric structural shape optimization, Computer Methods in Applied Mechanics and Engineering 197 (33-40) (2008) 2976–2988.

[4] L. Blanchard, R. Duvigneau, A.-V. Vuong, B. Simeon, Shape gradient for isogeometric structural design, J. Optimization Theory and Applications 161 (2).

[5] J. Cottrell, T. Hughes, A. Reali, Studies of refinement and continuity in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering (196) (2007) 4160–4183.

[6] Y. Bazilevs, C. Michler, V. Calo, T. Hughes, Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly-enforced boundary conditions on unstretched meshes, Computer Methods in Applied Mechanics and Engineering (199) (2010) 780–790.

[7] J. Evans, T. Hughes, Isogeometric divergence-conforming B-Splines for the steady Navier–Stokes equations, Mathematical Models and Methods in Applied Sciences 23 (8).

[8] P. Nortoft, T. Dokken, Isogeometric analysis of Navier-Stokes flow using locally refinable B-Splines, in: Springer (Ed.), SAGA – Advances in ShApes, Geometry, and Algebra, Vol. 10, 2014, pp. 299–318.

[9] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations, J. Comput. Physics 131 (2) (1997) 267–279.

[10] A. Silveira, R. Moura, A. Silva, M. Ortega, Higher-order surface treatment for discontinuous Galerkin methods with applications to aerodynamics, Int. J. for Numerical Methods in Fluids (79) (2015) 323–342.

[11] R. Sevilla, S. Fernandez-Mendez, A. Huerta, NURBS-enhanced finite element method for euler equations, Int. J. for Numerical Methods in Fluids 57 (9).

[12] P. Persson, J. Peraire, Curved mesh generation and mesh refinement using lagrangian solid mechanics, in: 47th AIAA Aerospace Sciences Meeting, no. 2009-949, 2009.

[13] P. George, H. Borouchaki, Construction of tetrahedral meshes of degree two, Int. J. for Numerical Methods in Engineering 90 (9) (2012) 1156–1182.

[14] R. Abgrall, C. Dobrzynski, A. Froehly, A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems, Int. J. for Numerical Methods in Fluids 76 (2014) 246–266.

[15] C. Geuzaine, A. Johnen, J. Lambrechts, J.-F. Remacle, T. Toulorge, The generation of valid curvilinear meshes, Notes on Numerical Fluid Mechanics and Multidisciplinary Design Volume 128 (15–39).

[16] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines, Computer Methods in Applied Mechanics and Engineering 199 (5–8) (2010) 229–263.

[17] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parametrization of computational domain in isogeometric analysis: methods and comparison, Computer Methods in Applied Mechanics and Engineering 200 (23-24).

[18] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Analysis-suitable volume parameterization of multi-block computational domain in isogeometric analysis, Computer Aided Design 45 (2).

[19] H. Speleers, C. Manni, Optimizing domain parameterization in isogeometric analysis based on Powell–Sabin splines, Journal of Computational and Applied Mathematics 289 (2015) 68–86.

[20] K. A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-Splines, Computer Methods in Applied Mechanics and Engineering 269 (2014) 471–514.

[21] M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, A. V. Vuong, Swept volume parameterization for isogeometric analysis, in: 13th IMA International Conference on Mathematics of Surfaces, 2009, pp. 19–44.

[22] L. Engvall, J. Evans, Isogeometric triangular bernstein-bézier discretizations: automatic mesh generation and geometrically exact finite-element analysis, Computer Methods in Applied Mechanics and Engineering (304) (2016) 378–407.

[23] L. Engvall, J. Evans, Isogeometric unstructured tetrahedral and mixed-element bernstein-bézier discretizations., Computer Methods in Applied Mechanics and Engineering (319) (2017) 83–123.

[24] N. Jaxon, X. Qian, isogeometric analysis on triangulations, Computer Aided Design (46) (2014) 45–57.

[25] S. Xia, X. Wang, X. Qian, Continuity and convergence in rational triangular bézier spline based isogeometric analysis, Computer Methods in Applied Mechanics and Engineering (297) (2015) 292–324.

[26] S. Xia, X. Qian, Isogeometric analysis with bézier tetrahedra, Computer Methods in Applied Mechanics and Engineering (316) (2017) 782–816.

[27] O. C. Zienkiewicz, P. Morice, The finite element method in engineering science, McGraw-Hill, 1971.

[28] C. Michoski, J. Chan, L. Engvall, J. Evans, Foundations of the blended isogeometric discontinuous Galerkin (BIDG) method, Computer Methods in Applied Mechanics and Engineering 305 (2016) 658–681.

[29] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous galerkin methods, in: AIAA Paper 2006-112, 2006.

[30] G. Farin, Curves and Surfaces for Computer-Aided Geometric Design, Academic Press, 1989.

[31] C. De Boor, A Practical Guide to Splines, Springer Verlag, 1978.

[32] L. Piegl, W. Tiller, The NURBS book, Springer-Verlag, 1995.

[33] G. Farin, D. Hansford, Discrete coons patches, Computer Aided Design 16 (7).

[34] J. S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods, Springer, 2008.

[35] B. Cockburn, C.-W. Shu, The local discontinuous galerkin method for time-dependent convection-diffusion systems, SIAM Journal of Num. An. 35 (6) (1998) 2440–2463.

[36] B. Cockburn, High-Order Methods for Computational Physics, Springer, 1999, Ch. Discontinuous Galerkin Methods for Convection-Dominated Problems.

[37] P. Batten, N. Clarke, C. Lambert, M. Causon, On the choice of wavespeeds for the hllc Riemann solver, SIAM J. Sci. Comput. 18 (6) (1997) 1553–1570.

[38] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer Berlin Heidelberg, 1997.

[39] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order cfd methods: current status and perspective, Int. J. for Numerical Methods in Fluids 72 (8).

[40] M. Yano, D. L. Darmofal, Case c1.2: Transonic ringleb flow, Tech. rep., MIT (2012).