

A NURBS-based Discontinuous Galerkin Framework for Compressible Aerodynamics

Stefano Pezzano, Régis Duvigneau

▶ **To cite this version:**

Stefano Pezzano, Régis Duvigneau. A NURBS-based Discontinuous Galerkin Framework for Compressible Aerodynamics. AIAA Aviation 2020 Forum, Jun 2020, Reno, United States. hal-02887354

HAL Id: hal-02887354
https://inria.hal.science/hal-02887354

Submitted on 2 Jul 2020

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

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

A NURBS-based Discontinuous Galerkin Framework for Compressible Aerodynamics

Stefano Pezzano* and Régis Duvigneau†
Université Côte d'Azur, INRIA, CNRS, LJAD, 2004 route des Lucioles - B.P. 93, 06902 Sophia-Antipolis, France

We propose a Discontinuous Galerkin (DG) method for the numerical solution of the compressible Navier-Stokes equations with a functional representation derived from Computer Aided Design (CAD). The extraction of a set of DG-compliant basis functions from Non-Uniform Rational B-Splines (NURBS) is briefly discussed and the discretization of the equations is presented in this new framework. The benefits of using high-order geometries are assessed in a first numerical example. The proposed scheme is then extended to deformable domains through an Arbitrary Lagrangian-Eulerian (ALE) formulation and a NURBS-based mesh movement, allowing smooth high-order deformations. A verification test case is carried out to evaluate the impact of mesh movement on numerical solutions. Finally, we propose a dynamic mesh deformation-adaptation algorithm to couple the ALE formulation with Adaptive Mesh Refinement (AMR), based on hierarchical properties of NURBS. The potential of the developed approach is investigated in an application dealing with a pitching aifoil.

I. Introduction

In the last 20 years, Computational Fluid Dynamics (CFD) has become a standard tool for analysis and design in the aerospace industry. However, due to the lack of integration between Computer Aided Design (CAD) and simulation software, the grid generation process is not fully automatized and it represenents a major bottleneck in the analysis chain. Moreover, industrial solvers rely on piecewise linear geometry descriptions and it has been shown [1, 2] that this approximation may lead to non-physical phenomena in numerical solutions.

In order to facilitate the integration of design and simulation, Hughes et al. [3] introduced the concept of Isogeometric Analysis, which consists in using NURBS, the mathematical representation adopted in CAD, as a basis for analysis. Thanks to this approach, high-order CAD-consistent geometries can be directly employed in simulation, simplifying mesh optimization and refinement procedures. Due to the high regularity of NURBS functions, Isogeometric Analysis is well suited to elliptic and parabolic problems, such as structural mechanics, whereas the number of applications in CFD has been so far very limited [4, 5].

At the same time, the CFD community has developed a growing interest in high-order schemes [6]. In particular, promising results have been obtained with Discontinuous Galerkin (DG) methods in a number of applications [1, 7, 8]. In an effort to combine the benefits of Isogeometric Analysis with those of high-order methods for CFD, we investigated in [9] a NURBS-based DG formulation for Euler equations and tested it on some simple validation test cases. In the present work we extend the NURBS-based DG method to more complex two-dimensional problems, with the aim of showing the accuracy and flexibility of the developed framework. A particular attention is given to problems where the computational domain is deforming.

The first section is dedicated to the mathematical formulation. We explain how to obtain a set of DG-compliant basis functions starting from CAD geometries and we discretize the compressible Navier-Stokes equations. An Arbitrary Lagrangian-Eulerian (ALE) formulation is employed to deal with deformable domains. Flows around moving obstacles present highly unstationary phenomena, therefore dynamic mesh adaptation may lead to significant gain in computational time. For this reason, in the second section, we show how the NURBS representation can be exploited to develop a coupled dynamic mesh deformation-adaptation algorithm. In the last section we present the results obtained in three numerical test cases.

*Ph.D. Fellow, ACUMES team, stefano.pezzano@inria.fr
†Permanent Researcher, ACUMES team, regis.duvigneau@inria.fr

II. Numerical scheme

A. Bézier and NURBS surfaces

The first approach to interactive geometric design consisted historically in the use of Bézier surfaces, defined as:

$$\mathbf{S}(\xi,\eta) = \sum_{i_1=1}^{p+1}\sum_{i_2=1}^{p+1} B_{i_1}^{p}(\xi)\, B_{i_2}^{p}(\eta)\, \mathbf{X}_{i_1 i_2}, \tag{1}$$

where $\mathbf{X}_{i_1 i_2}$ are the control points, and B_i^p are the Bernstein polynomials of degree p:

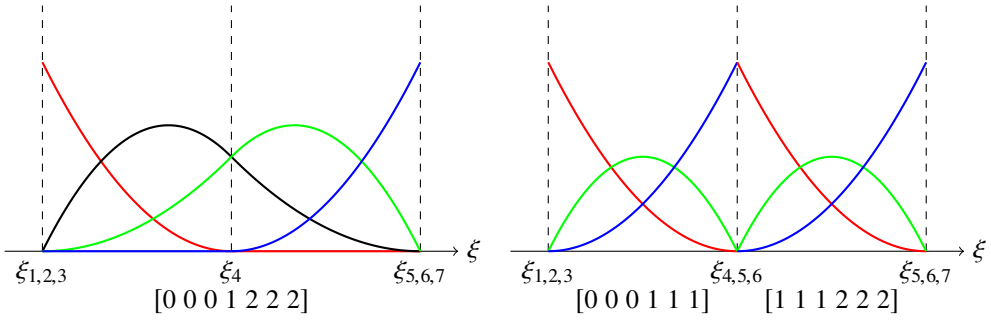
$$B_i^p(\varphi) = \binom{p}{i-1} \varphi^{i-1}(1-\varphi)^{p-i-1}, \tag{2}$$

with $\varphi \in [0,1]$. Polynomial curves are not able to exactly represent conic sections, therefore, in order to overcome this limitation, rational Bernstein functions were introduced:

$$R_{i_1 i_2}^{p}(\xi,\eta) = \frac{B_{i_1}^{p}(\xi)\, B_{i_2}^{p}(\eta)\, \omega_{i_1 i_2}}{\sum_{j_1=1}^{p+1}\sum_{j_2=1}^{p+1} B_{j_1}^{p}(\xi)\, B_{j_2}^{p}(\eta)\, \omega_{j_1 j_2}}, \tag{3}$$

with the coefficients $\omega_{i_1 i_2}$ being positive real numbers called weights. Since Bernstein polynomials are a partition of unity, when all the weights are unitary, $R_i^p(\xi) = B_i^p(\xi)$. Rational Bézier surfaces are defined as:

$$\mathbf{S}(\xi,\eta) = \sum_{i_1=1}^{p+1}\sum_{i_2=1}^{p+1} R_{i_1 i_2}^{p}(\xi,\eta)\, \mathbf{X}_{i_1 i_2} \tag{4}$$



(a) Set of quadratic B-Spline functions **(b) Split set of quadratic Bernstein polynomials**

Fig. 1 Bézier extraction procedure: continuous B-Splines over [0,2] are split into two sets of DG-compliant quadratic Bernstein polynomials over [0,1] and [1,2]

Complex geometries require a high-degree basis when represented using a single polynomial patch, therefore, as explained in [3], CAD software representations are rather based on B-Splines, that are piecewise extension of Bernstein polynomials. We consider the parametric domain $\hat{\Omega} = [\xi_1, \xi_l]$, discretized by the knot vector $\Xi = (\xi_1, ..., \xi_i, ..., \xi_l)$. B-Spline functions of degree p are evaluated recursively:

$$N_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \le \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi). \tag{6}$$

NURBS functions are the rational extension of B-Splines:

$$R_{i_1 i_2}^{p}(\xi,\eta) = \frac{N_{i_1}^{p}(\xi)\, N_{i_2}^{p}(\eta)\, \omega_{i_1 i_2}}{\sum_{j_1=1}^{n_1}\sum_{j_2=1}^{n_2} N_{j_1}^{p}(\xi)\, N_{j_2}^{p}(\eta)\, \omega_{j_1 j_2}}. \tag{7}$$

Thanks to the possible presence of solution discontinuities at element interfaces, rational Bernstein functions (3) are a suitable basis for a DG method, unlike NURBS, that can be used for a Continuous Galerkin (CG) approximation as in classical Isogeometric Analysis [3]. As described in [9, 10], it is therefore possible to generate a DG-compliant representation, starting from CAD geometries, by a procedure called Bézier extraction. It consists in a multiple knot insertion at the internal knots of Ξ, until all knots have $(p+1)$-multiplicity, as illustrated in Fig. 1. This operation extracts a set of rational Bézier patches from a single NURBS patch, without altering the geometric representation.

B. Discontinuous Galerkin formulation

We consider the conservative form of the two-dimensional Navier-Stokes equations:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \left(\mathbf{F}_c - \mathbf{F}_v\right) = 0, \tag{8}$$

with $\mathbf{W}$ being the vector of conservative variables, $\mathbf{F}_c$ the convective flux, and $\mathbf{F}_v$ the viscous flux:

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho e \end{pmatrix}, \quad \mathbf{F}_{c,i} = \begin{pmatrix} \rho u_i \\ \rho u_1 u_i + p\delta_{1i} \\ \rho u_2 u_i + p\delta_{2i} \\ \rho u_i \left(e + \frac{p}{\rho}\right) \end{pmatrix}, \quad \mathbf{F}_{v,i} = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ u_k \tau_{ki} - q_i \end{pmatrix}, \tag{9}$$

where τ_{ij} is the viscous stress tensor and q_i is the thermal conduction flux, defined as:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k}\delta_{ij}, \tag{10}$$

$$q_i = -\gamma \frac{\mu}{Pr}\frac{\partial e}{\partial x_i}. \tag{11}$$

The treatment of second order derivatives follows the Local Discontinuous Galerkin (LDG) method [11], in which an auxiliary variable is introduced to rewrite eqs. (8) as a system of first order equations:

$$\begin{cases} \dfrac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{W}) - \nabla \cdot \mathbf{F}_v(\mathbf{W}, \mathbf{G}) = 0, \\ \mathbf{G} - \nabla \mathbf{W} = 0. \end{cases} \tag{12}$$

The isogeometric paradigm is used to represent the geometry $\mathbf{x}$ and the discrete solution fields $\mathbf{w}_h$ and $\mathbf{g}_h$ in each element:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{w}_h \\ \mathbf{g}_h \end{pmatrix} = \sum_{i=1}^{(p+1)^2} R_i(\xi, \eta) \begin{pmatrix} \mathbf{x}_i \\ \mathbf{w}_i \\ \mathbf{g}_i \end{pmatrix}. \tag{13}$$

In fact, each element is a rational Bézier patch, described by its control points $\mathbf{x}_i$ and the degrees of freedom (DOF) for the conservative variables $\mathbf{w}_i$ and their gradients $\mathbf{g}_i$. The NURBS-based DG discretization of eqs. (12) is:

$$\begin{cases} \dfrac{d\mathbf{w}_i}{dt} \displaystyle\int_{\hat{\Omega}_j} R_k R_i |J_\Omega| \, \mathrm{d}\hat{\Omega} = \int_{\hat{\Omega}_j} \nabla R_k \cdot \left(\mathbf{F}_c - \mathbf{F}_v\right) |J_\Omega| \, \mathrm{d}\hat{\Omega} - \oint_{\partial\hat{\Omega}_j} R_k \left(\mathbf{F}_c^* - \mathbf{F}_v^*\right) |J_\Gamma| \, \mathrm{d}\hat{\Gamma}, & \text{(14a)} \\ \mathbf{g}_i \displaystyle\int_{\hat{\Omega}_j} R_k R_i |J_\Omega| \, \mathrm{d}\hat{\Omega} = \int_{\hat{\Omega}_j} \nabla R_k \mathbf{w}_h |J_\Omega| \, \mathrm{d}\hat{\Omega} - \oint_{\partial\hat{\Omega}_j} R_k \mathbf{W}^* |J_\Gamma| \, \mathrm{d}\hat{\Gamma}. & \text{(14b)} \end{cases}$$

The convective numerical flux $\mathbf{F}_c^*$ is computed through the HLL Riemann solver [12], whereas the viscous fluxes $\mathbf{F}_v^*$ and $\mathbf{W}^*$ are evaluated using the LDG approach [11]. Note that, in the case of shock capturing, the viscous flux includes additional terms related to artifical viscosity, see [9] for more details. The Gauss-Legendre quadrature rule is adopted for space integration, which is performed on a reference square element $\hat{\Omega}$, thus the determinant of the Jacobian matrix J of the coordinate transformation appears within the integrals. Equation (14b) does not contain a time derivative, hence it can be solved within each time iteration in a decoupled manner. It is thus possible to rewrite eqs. (14) in a compact form:

$$\mathcal{M}\frac{d\mathbf{w}}{dt} = \mathcal{R}(\mathbf{w}_h), \tag{15}$$

where $\mathcal{M}$ is the mass matrix and the residual $\mathcal{R}$ is the right-hand side of eq. (14a). The system of ordinary differential equations (ODE) (15) is integrated in time either with the standard 4th-order four-stage explicit Runge-Kutta (RK) scheme or with the Strong Stability Preserving (SSP) 3rd-order three-stage RK method proposed by Gottlieb et al. [13].

C. Extension to deformable domains

The Arbitrary Lagrangian-Eulerian (ALE) method consists in solving the equations of motion in a domain that moves with an arbitrary velocity field $\mathbf{V}_g(\mathbf{x}, t)$. To this end, we rewrite Navier-Stokes equations in ALE form:

$$\begin{cases} \dfrac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{W}) - \nabla \cdot \mathbf{F}_v(\mathbf{W}, \mathbf{G}) - \mathbf{V}_g \cdot \nabla \mathbf{W} = 0, \\ \mathbf{G} - \nabla \mathbf{W} = 0. \end{cases} \tag{16}$$

In order to obtain the weak formulation of eqs. (16), we adopt the approach proposed by Nguyen [14], leading to the following DG scheme:

$$\begin{cases} \dfrac{d}{dt}\left(\mathbf{w}_i \displaystyle\int_{\hat{\Omega}_j} R_k R_i |J_\Omega| \, \mathrm{d}\hat{\Omega}\right) = \int_{\hat{\Omega}_j} \nabla R_k \cdot \left(\mathbf{F}_c - \mathbf{F}_v - \mathbf{V}_g \mathbf{w}_h\right) |J_\Omega| \, \mathrm{d}\hat{\Omega} - \oint_{\partial\hat{\Omega}_j} R_k \left(\mathbf{F}^*_{ale} - \mathbf{F}^*_v\right) |J_\Gamma| \, \mathrm{d}\hat{\Gamma}, & \text{(17a)} \\ \mathbf{g}_i \displaystyle\int_{\hat{\Omega}_j} R_k R_i |J_\Omega| \, \mathrm{d}\hat{\Omega} = \int_{\hat{\Omega}_j} \nabla R_k \mathbf{w}_h |J_\Omega| \, \mathrm{d}\hat{\Omega} - \oint_{\partial\hat{\Omega}_j} R_k \mathbf{W}^* |J_\Gamma| \, \mathrm{d}\hat{\Gamma}. & \text{(17b)} \end{cases}$$

Equation (17b) is the same as its fixed grid counterpart, whereas eq. (17a) presents some additional flux terms due to mesh movement. The convective numerical flux $\mathbf{F}^*_{ale} = \mathbf{F}^*_{ale}(w_h^+, w_h^-, \mathbf{V}_g, \mathbf{n})$ is a modified HLL Riemann solver that complies with the ALE consistency condition:

$$\mathbf{F}^*_{ale}\left(\mathbf{w}_0, \mathbf{w}_0, \mathbf{V}_g, \mathbf{n}\right) = \mathbf{F}_c\left(\mathbf{w}_0\right) \cdot \mathbf{n} - \left(\mathbf{V}_g \cdot \mathbf{n}\right)\mathbf{w}_0 \tag{18}$$

Furthermore, the mass matrix is time dependent for general grid velocity fields, which means that the system of ODEs to solve becomes:

$$\frac{d}{dt}\left(\mathcal{M}\mathbf{w}\right) = \mathcal{R}(\mathbf{w}_h, \mathbf{V}_g) \tag{19}$$

It is worth noting that the proposed formulation does not satisfy the Discrete Geometric Conservation Law (DGCL), hence, constant solutions are not exactly preserved.

In a isogeometric fashion, the grid velocity field is expressed by means of the same set of basis functions used for the geometry and the solution:

$$\mathbf{V}_g = \sum_{i=1}^{(p+1)^2} R_i(\xi, \eta)\, \mathbf{v}_i, \tag{20}$$

allowing us to easily treat arbitrarily high-order movement laws and, at the same time, exploit the regularity properties of rational Bézier functions to obtain smoothly deformed meshes. For the present work, grid displacement is known *a priori*, therefore, the distribution of control point velocities $\mathbf{v}_i$ is imposed at each time step and integrated together with eqs. (17).

III. Adaptive refinement for moving meshes

Adaptive Mesh Refinement (AMR) techniques allow to optimize the grid at runtime by choosing the appropriate resolution for each region of the computational domain. It is thus interesting to use adaptive meshes for flows around moving and deforming obstacles, where highly unsteady physical phenomena occurs, such as moving shocks or complex vortical wakes.

A quadtree-like approach is adopted to refine rational Bézier patches. Whenever an element of Level-k is marked for refinement, it is split into 4 child elements of Level-(k+1) by inserting multiple knots at $\xi = 0.5$ and $\eta = 0.5$, as in the Bézier extraction procedure. The father element is stored and it can be recovered if its child elements are selected for coarsening. Thanks to the isogeometric paradigm, the same approach applies for both the geometry and the solution.

An error estimator is introduced to decides which elements should be refined, based on the measure of interface jumps. As will be shown in application section, this indicator has the ability to identify regions with under-resolved

elements. In practice, for each element Ω_j, one evaluates the solution jump at the interface between Ω_j and the neighbouring elements:

$$\epsilon_j = \sum_{k \in \mathcal{N}_j} \int_{\Gamma_{jk}} \| \mathbf{W}|^j - \mathbf{W}|^k \| \, d\Gamma, \tag{21}$$

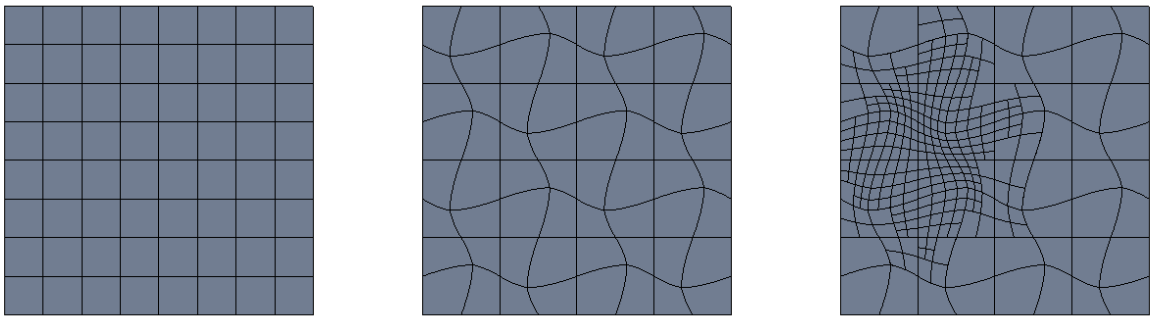
where $\mathcal{N}_j$ represents the set of elements around Ω_j and Γ_{jk} the interface between Ω_j and Ω_k. Then, the element Ω_j is flagged for refinement if the indicator ϵ_j exceeds the indicator average $\bar{\epsilon}$ by a user-defined ratio ϵ_{ref}:

$$\frac{\epsilon_j}{\bar{\epsilon}} > \epsilon_{ref}. \tag{22}$$

Similarly, the element is flagged for coarsening if ϵ_j is lower than $\bar{\epsilon}$, with a user-defined ratio ϵ_{coa}:

$$\frac{\epsilon_j}{\bar{\epsilon}} < \epsilon_{coa}. \tag{23}$$

In the case of coarsening, the solution should be approximated in the father element, from data located in the four sons elements. This is achieved by using a least-squares projection. The whole adaptation procedure is detailed in [15].



(a) Level-0 mesh **(b) Deformed Level-0 mesh** **(c) Refined deformed mesh**

Fig. 2 Example of adaptive refinement with mesh deformation

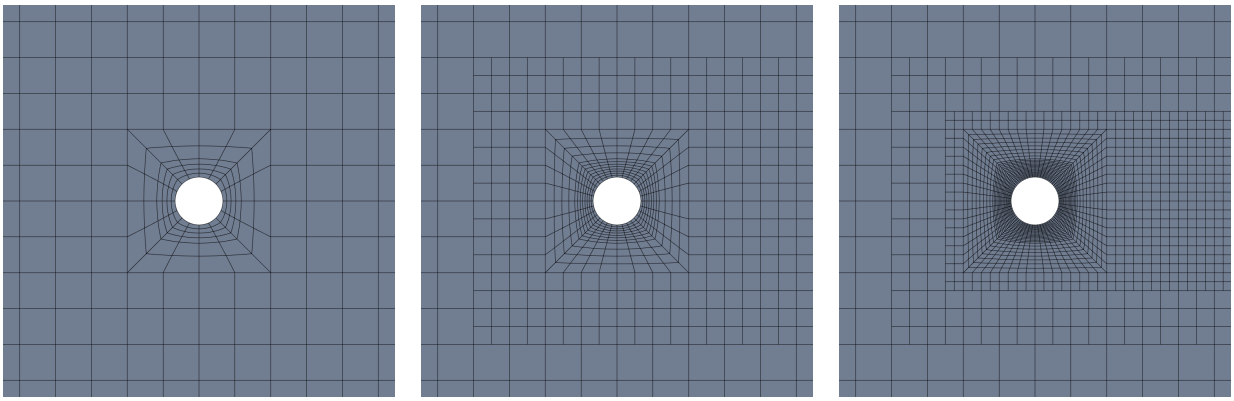
When an adapted mesh is subject to non-linear deformations, the refinement operation might become irreversible. In fact, father elements are not capable of exactly representing their deformed child elements, yielding possible holes within the mesh. To remedy this problem, one just has to apply the mesh movement to the Level-0 elements and then propagate the velocity field and the deformation to the higher levels of the tree by means of the knot insertion procedure, as presented in Fig. 2.

IV. Results

A. Cylinder flow

As a first test case, we investigate the laminar flow around a circular cylinder. Thanks to the proposed formulation, it is possible to exactly represent the computational domain, in fact, the circular boundary can be perfectly described by means of 4 rational Bézier arcs, from which a very coarse baseline grid is generated. Local refinement is then applied to the initial patches to obtain the computational mesh without the necessity of refitting the curved boundary. The cylinder of diameter D is centered at the origin (0,0) and the domain is delimited by the rectangle $[-25D, 100D] \times [-25D, 25D]$, the freestream Mach number is 0.2 and the Reynolds number is equal to 500. Rational shape functions based on polynomials of degrees from 3 to 5 are tested, and, for each degree, the simulations are carried out with 3 different refinement levels:

- coarse: 1065 elements, with 16 Bézier arcs on the boundary,
- intermediate: 2145 elements, with 32 Bézier arcs on the boundary,
- fine: 4455 elements, with 64 Bézier arcs on the boundary.

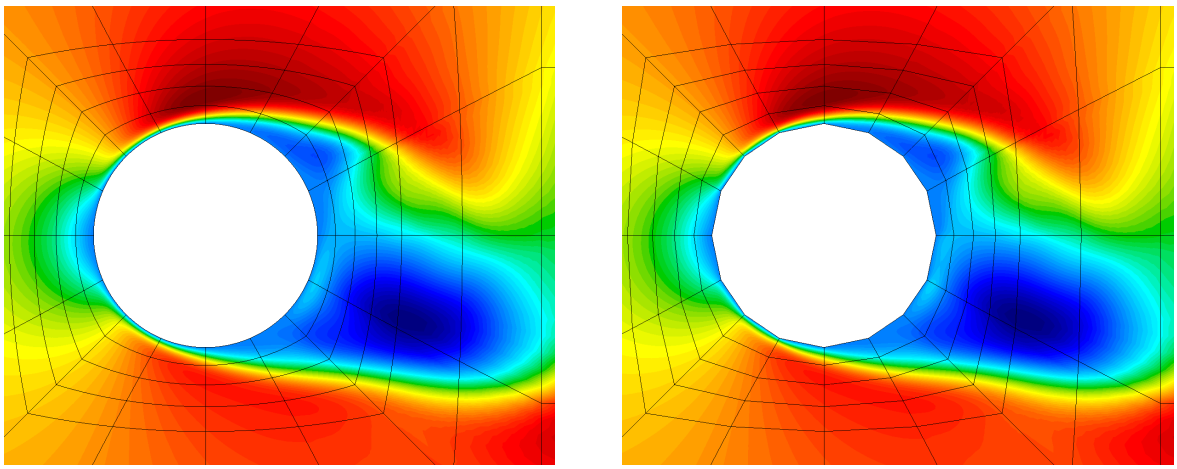


(a) Coarse mesh **(b) Intermediate mesh** **(c) Fine mesh**

Fig. 3 Different refinement levels for the cylinder simulation

The different mesh levels are represented in Fig. 3. For each combination of degree and grid refinement, we compare the results obtained with curved and linear boundaries, as it can be observed in Fig. 4, in order to assess the gain given by the NURBS-based representation.

The average drag coefficient $\bar{C}_D$, and the Strouhal number St are calculated, and, for each basis degree, we study their evolution over the different refinement levels. The converged values that we obtain are in close agreement to those provided by Blackburn et al. [16]. In Fig. 5, it is possible to observe that, for all the tested degrees, the same converged solution is found, and, as expected, the convergence is faster for higher order basis functions, but only when curved boundaries are used, indeed, the approximated geometry acts as a source of error, lowering the convergence rate even if the solution field is represented with a high-order basis.

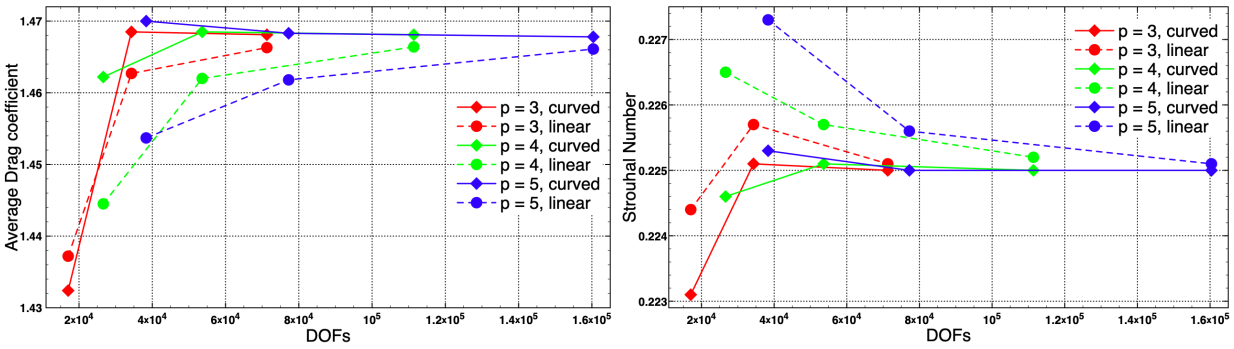


(a) Curved mesh **(b) Linear mesh**

Fig. 4 Close-up view of streamwise momentum field, coarse mesh, p = 5

B. Isentropic vortex with fixed and deforming mesh

An extensive validation of the NURBS-based DG method for the Euler equations has been carried out in [9]. As a second test case, we perform a validation of the ALE formulation and we quantify the error introduced by the mesh



(a) Drag coefficient convergence **(b) Strouhal number convergence**

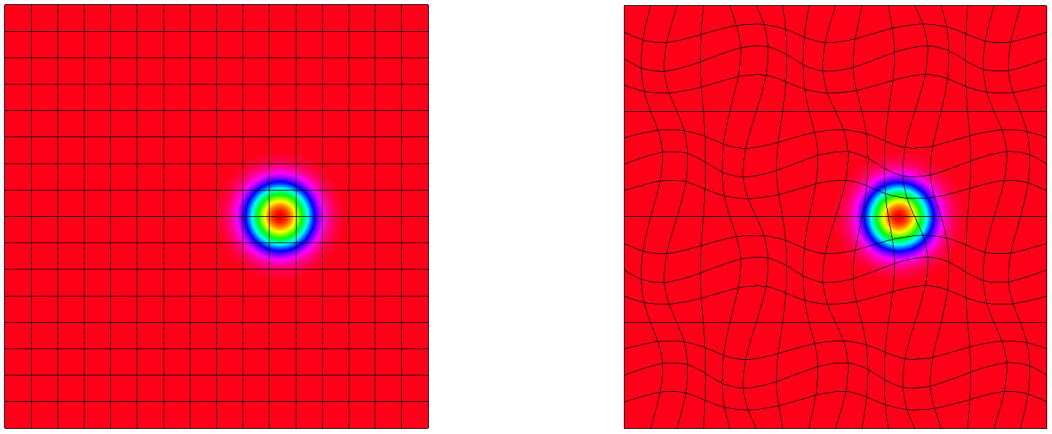
Fig. 5 Cylinder flow, numerical results. The continuous lines represent results obtained with curved boundaries, whereas dashed lines are used for those obtained with linear boundaries.

movement. The advection of an isentropic vortex [17] is considered, with the following analytic solution:

$$\begin{cases} \rho = \left(1 - \dfrac{\gamma - 1}{16\gamma\pi^2}\beta^2 e^{2(1-r^2)}\right)^{\frac{1}{\gamma-1}} \\ u_1 = 1 - \beta\dfrac{y - y_0}{2\pi}e^{1-r^2} \\ u_2 = \beta\dfrac{x - t - x_0}{2\pi}e^{1-r^2} \\ p = \rho^\gamma \end{cases} \tag{24}$$

where $r = \sqrt{(x - t - x_0)^2 + (y - y_0)^2}$, with $x_0 = 5$, $y_0 = 0$ and $\beta = 5$. The computational domain is $[0, 10] \times [-5, 5]$ and the boundary fluxes are evaluated by means of the analytic solution. The simulation is run on a fixed cartesian grid and on a deforming mesh, whose control points move following a sinusoidal law:

$$\mathbf{v}_i(\mathbf{x}_i, t) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \sin\left(N_1 \frac{\pi}{4} x_{i,1}\right) \sin\left(N_2 \frac{\pi}{4} x_{i,2}\right) \sin(2\pi t) \tag{25}$$

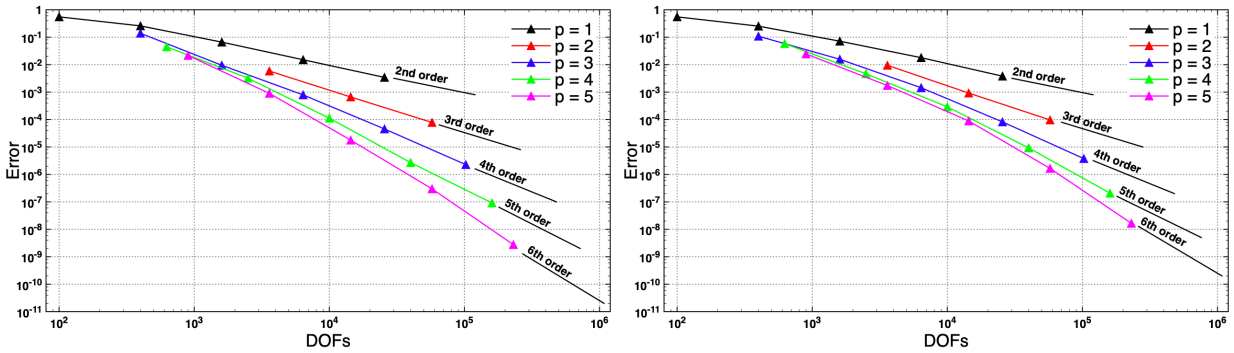


(a) Fixed mesh **(b) Deforming mesh**

Fig. 6 Numerical solution and computational domain, density, t=0.25

We compute the L2 norm of the error of the internal energy at time $t = 2$. Polynomials up to fifth degree are tested, optimal convergence rates are verified in both cases for each degree as it is shown in Fig. 7, where it is also possible to

compare the error between the two setups. As expected, when the mesh deforms over time, we experience a loss of accuracy, which depends on the polynomial degree. Up to cubic polynomials the ratio between the two errors is less than 1.8, whereas for quartic basis a factor 3.5 is found in the worst case and for quintic functions a value of 6 is reached. This sudden increase can be explained as a loss of accuracy of the Gauss quadrature rule when the mesh is subject to high-order deformations, mainly due to the highly non-linear metric terms introduced by the isoparametric map. In our investigation, we did not find the lack of exact freestream preservation to be a major source of error.

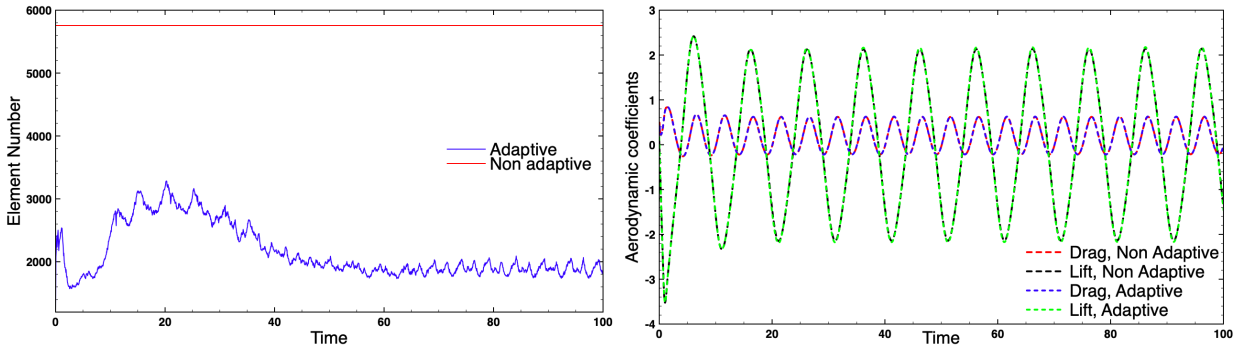


(a) Fixed mesh **(b) Deforming mesh**

Fig. 7 Convergence analysis, L2 error of internal energy

C. Pitching NACA airfoil

In order to show the capability of the proposed adaptive mesh refinement technique, we study the flow around a pitching NACA 0012 airfoil. Two configurations are investigated: a subsonic laminar flow with high frequency and high amplitude pitching motion, and a transonic inviscid flow with very low frequency and amplitude pitching. In order to fairly compare the results, the smallest elements of the adaptively refined mesh must have the same size of those of the non-adaptive grid. Therefore, a common baseline mesh is used. The non-adaptive grid is then obtained by refining N_r times using rectangular boxes, whereas the adaptive mesh is the result of the proposed AMR procedure with N_r refinement levels. The geometry of the airfoil cannot be exactly represented using NURBS, therefore the boundary is obtained by polynomial curve fitting. Cubic Bernstein polynomials are employed for both the laminar and transonic test cases.



(a) Evolution of the number of elements **(b) Evolution of the aerodynamic coefficients**

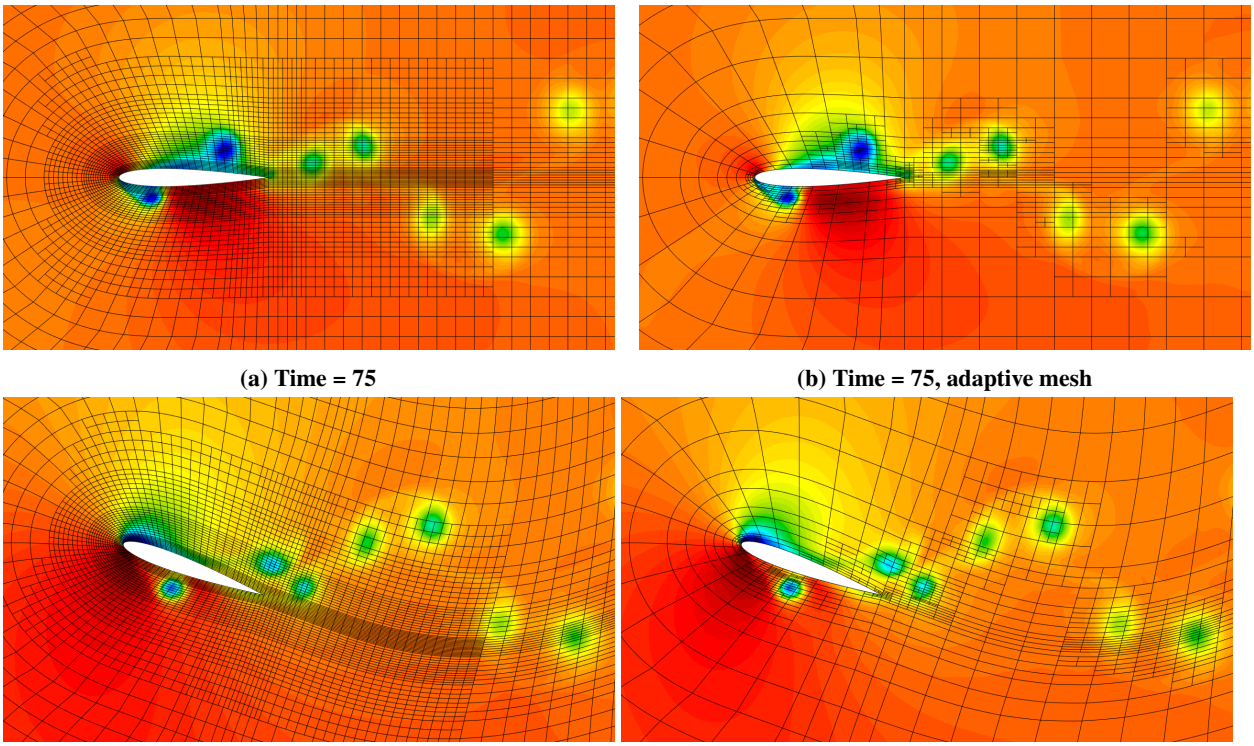
Fig. 8 Laminar pitching NACA airfoil, results comparison

1. Laminar flow

For the laminar test case, the airfoil is pitching about mid-chord and the angle of attack follows the law:

$$\alpha(t) = -A \sin(2\pi f t), \tag{26}$$

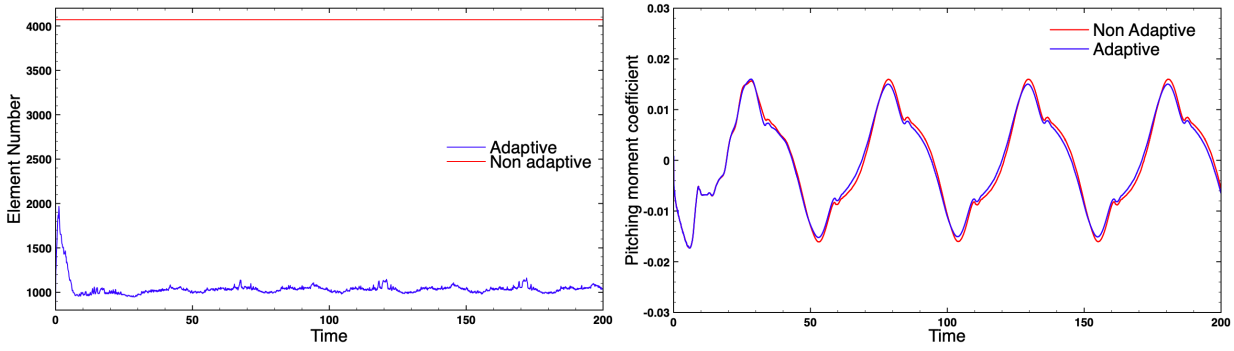
with amplitude $A = 20^\circ$ and reduced frequency $k = \pi f c/U_\infty = 0.25$, with c being the chord length, and U_∞ the undisturbed velocity. The freestream Mach number is equal to 0.2 and the Reynolds number is 1000. This configuration generates a massively separated flow, with pairs of vortices being shed from the airfoil. The dynamic mesh adaptation procedure is capable of capturing the formation of the vortices at the leading edge of the airfoil and following the complex wake pattern induced by the pitching motion, as shown in Fig. 9. The advantage in terms of the number of elements is represented in Fig. 8a, one can observe that after an initial peak, the adaptive mesh has nearly three times less elements than the non-adaptive grid. In Fig. 8b we report the time evolution of the aerodynamic force coefficients C_D and C_L. The curves obtained with the two meshes are superposed, meaning that the reduction in the number of elements has very little impact on the accuracy of the simulation. As shown in Fig. 9, the solution fields are in close agreement as well.



(a) Time = 75 **(b) Time = 75, adaptive mesh**

(c) Time = 87.5 **(d) Time = 87.5, adaptive mesh**

Fig. 9 Laminar pitching NACA airfoil, comparison of the density field

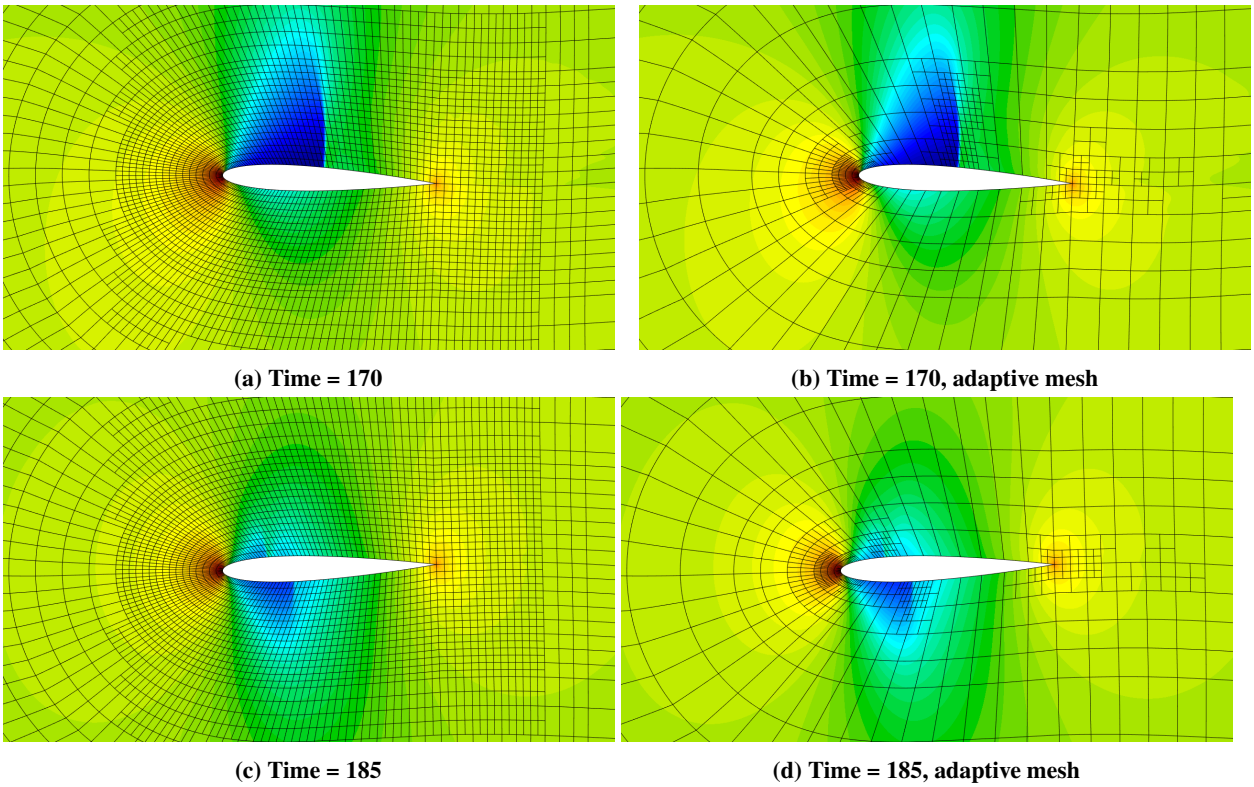


(a) Evolution of the number of elements **(b) Evolution of the pitching moment coefficient**

Fig. 10 Transonic pitching NACA airfoil, results comparison

2. Transonic flow

For this test case, the airfoil is pitching about quarter-chord, the initial angle of attack is 0.016°, the pitch amplitude is 2.51°, the reduced frequency is equal to 0.0814 and the freestream Mach number is 0.755. Simulating unsteady transonic flows with high-order solvers can be challenging, indeed, the presence of moving shocks requires a very fine grid around the airfoil and a suitable shock capturing algorithm. In this context, the use of AMR allows to have the necessary mesh resolution around the shock while keeping a very coarse grid in the other regions of the airfoil, as shown in Fig. 11. This is obviously possible thanks to the high order representation of the boundary. The time evolution of the pitching moment coefficient C_m is reported in Fig. 10b for both meshes, the two curves are comparable, but a slight discrepancy can be observed. Indeed, the mesh adaptation interacts with the shock sensor, thus altering the quantity of artificial viscosity introduced to capture the discontinuity. The density fields obtained with the two approaches are comparable as well, as shown in Fig. 11, and the gain in terms of number of elements is considerable, as presented in Fig. 10a.



(a) Time = 170 **(b) Time = 170, adaptive mesh**

(c) Time = 185 **(d) Time = 185, adaptive mesh**

Fig. 11 Transonic pitching NACA airfoil, comparison of the density field

V. Conclusion

We presented, in this work, a CAD-consistent DG formulation for the simulation of compressible flows. The generation of a suitable curved mesh starting from NURBS representations has been discussed. The convergence analysis performed on the cylinder test case shows that the use of curvilinear boundary representations is required in the context of high-order solvers for CFD. The developed scheme has then been extended to deformable domains by means of an ALE formulation, which has been validated using an analytic test case. Optimal convergence rates were observed and only a minimal increase of the error was found on the deforming mesh, compared to a fixed cartesian grid.

We then proposed a simple yet effective strategy to couple AMR and ALE using the hierarchical properties of NURBS. The dynamic mesh deformation-adaptation was tested on the pitching airfoil benchmark in two different flow configurations. The obtained results show the interest of the ALE-AMR coupling. A significant reduction in the number of elements can be obtained without losing accuracy, thanks to the dynamic mesh adaptation, which is capable of capturing the most prominent features of unsteady flows, such as shocks and vortices.

References

[1] Bassi, F., and Rebay, S., "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 131, No. 2, 1997, pp. 267 – 279.

[2] Silveira, A. S., Moura, R. C., Silva, A. F. C., and Ortega, M. A., "Higher-order surface treatment for discontinuous Galerkin methods with applications to aerodynamics," *International Journal for Numerical Methods in Fluids*, Vol. 79, No. 7, 2015, pp. 323–342.

[3] Hughes, T., Cottrell, J., and Bazilevs, Y., "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 39, 2005, pp. 4135 – 4195.

[4] Nielsen, P. N., Gersborg, A. R., Gravesen, J., and Pedersen, N. L., "Discretizations in isogeometric analysis of Navier–Stokes flow," *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, No. 45, 2011, pp. 3242 – 3253.

[5] Bazilevs, Y., Michler, C., Calo, V., and Hughes, T., "Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly-enforced boundary conditions on unstretched meshes," *Computer Methods in Applied Mechanics and Engineering*, , No. 199, 2010, pp. 780–790.

[6] Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.

[7] Lomtev, I., Kirby, R., and Karniadakis, G., "A Discontinuous Galerkin ALE Method for Compressible Viscous Flows in Moving Domains," *Journal of Computational Physics*, Vol. 155, No. 1, 1999, pp. 128 – 159.

[8] Chapelier, J.-B., de la Llave Plata, M., Renac, F., and Lamballais, E., "Evaluation of a high-order discontinuous Galerkin method for the DNS of turbulent flows," *Computers & Fluids*, Vol. 95, 2014, pp. 210 – 226.

[9] Duvigneau, R., "Isogeometric analysis for compressible flows using a Discontinuous Galerkin method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 333, 2018, pp. 443 – 461.

[10] Piegl, L., and Tiller, W., *The NURBS Book*, 2nd ed., Springer-Verlag, New York, NY, USA, 1996.

[11] Cockburn, B., and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 2440–2463.

[12] Harten, A., Lax, P. D., and van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," *SIAM Review*, Vol. 25, No. 1, 1983, pp. 35–61.

[13] Gottlieb, S., Shu, C.-W., and Tadmor, E., "Strong Stability-Preserving High-Order Time Discretization Methods," *SIAM Review*, Vol. 43, No. 1, 2001, pp. 89, 112.

[14] Nguyen, V.-T., "An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for simulations of flows over variable geometries," *Journal of Fluids and Structures*, Vol. 26, No. 2, 2010, pp. 312, 329.

[15] Duvigneau, R., "CAD-consistent adaptive refinement using a NURBS-based discontinuous Galerkin method," *Int. J. for Numerical Methods in Fluids*, 2020.

[16] Blackburn, H. M., and Henderson, R. D., "A study of two-dimensional flow past an oscillating cylinder," *Journal of Fluid Mechanics*, Vol. 385, 1999, p. 255–286.

[17] Hesthaven, J. S., and Warburton, T., *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, 1st ed., Springer, 2007.