A NURBS-based Discontinuous Galerkin method for conservation laws with high-order moving meshes

Stefano Pezzano, Régis Duvigneau

A NURBS-based Discontinuous Galerkin method for conservation laws with high-order moving meshes

Stefano Pezzano, Régis 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 the present work is to develop a new numerical framework for simulations including moving bodies, in the specific context of high-order meshes consistent with Computer-Aided Design (CAD) representations. Thus, the proposed approach combines ideas from isogeometric analysis, able to handle exactly CAD-based geometries, and Discontinuous Galerkin (DG) methods with an Arbitrary Lagrangian-Eulerian (ALE) formulation, able to solve complex problems with moving grids. The resulting approach is a DG method based on rational Bézier elements, that can be easily constructed from Non-Uniform Rational B-Splines (NURBS), formulated in a general ALE setting. We focus here on applications in compressible aerodynamics, but the method could be applied to other models. Two verification exercises are conducted, to assess rigorously the properties of the method and the convergence rates for representations up to sixth order. Finally, two problems are analysed in depth, involving compressible Euler and Navier-Stokes equations, for an oscillating cylinder and a pitching airfoil. In particular, the convergence of flow characteristics is investigated, as well as the impact of using curved boundaries in the context of moving bodies.

1 Introduction

In the last 20 years, Computational Fluid Dynamics (CFD) has become a standard tool for analysis and design of fluid systems in industry. In this context, the issue of a better integration of CFD solvers in complex multidisciplinary analysis chains has emerged. In particular, the lack of integration between Computer Aided Design (CAD) and CFD software regarding the treatment of the geometry has been reported. This difficulty is mainly due to the fact that, in most industrial cases, the grid generation process is not fully automatized yet and requires several geometrical transformations that are still carried out by hand. In order to facilitate the integration of design and simulation, Hughes et al. (1) introduced the concept of Iso-Geometric Analysis (IGA), whose formulation relies on Non-Uniform Rational B-Splines (NURBS), the standard mathematical representation of modern CAD (2). Essentially, IGA consists in a finite element method in which both the geometry and the solution are described by NURBS functions.

At the same time, the CFD community has developed a growing interest in high-order schemes for conservation laws (3). In particular, promising results have been obtained with Discontinuous Galerkin (DG) methods in a number of applications (4; 5; 6). However, as

shown by several authors, the use of piecewise-linear geometry descriptions in such very high-order CFD solvers may lead to non-physical phenomena in numerical solutions (4; 7; 8). This observation justified the development of a new generation of high-order solvers, capable of handling CAD geometries natively. One can refer for instance to the NURBS-Enhanced Finite Element Method (NEFEM) (9) and its Discontinuous Galerkin counterpart (7), in which the NURBS representations are only employed for the geometry of the boundaries. Similarly, an extension of the residual distribution scheme for NURBS geometries has been proposed (10). More recently, two DG schemes inspired by IGA have been developed: the Blended-Isogeometric Discontinuous Galerkin (BIDG) method (11) which relies on nodal elements whose geometry is defined by NURBS in the whole computational domain, and its fully isogeometric counterpart which uses NURBS for both the geometry and the solution (12). All these approaches have demonstrated the interest of using high-order boundary representations in terms of accuracy for some selected test-cases governed by compressible or incompressible flow models.

Nevertheless, none of the aforementioned studies presented applications with moving grids. Improving the description of the geometry is certainly beneficial in such applications, in particular because the displacement of the boundary may amplify the spurious phenomena generated by a low-order geometry representation. Therefore, simulating flows while accounting exactly for CAD geometries should be an advantage over traditional approaches where the moving boundaries are piecewise linear. However, several issues arise when one attempts to combine Arbitrary Lagrangian-Eulerian (ALE) formulations and NURBS representations, such as the development of a robust technique to deform high-order grids or the preservation of constant solutions on moving meshes. As a consequence, the present study aims at proposing a new framework for simulating compressible flows with arbitrary displacement of boundaries defined by NURBS. The proposed approach relies on a DG formulation based on rational Bézier elements (12), that can be easily constructed from a set of NURBS boundaries using local refinement (13), preserving the CAD geometry exactly.

Several ALE formulations have been proposed in the context of DG methods. A review of the possible ALE technique is therefore carried out in section 2 and two categories of approaches are identified. In the first family of methods, the conservation laws are effectively solved in the moving domain (5; 14), whereas, in the second class of schemes, the equations are solved in a fixed reference domain thanks to the introduction of a map (15; 16). The main characteristics of the various approaches are discussed, with a particular emphasis on the role of the Discrete Geometric Conservation Law (DGCL). For the sake of simplicity, the proposed analysis is limited to method of lines DG discretizations. It is however worth mentioning that interesting results have also been obtained with space-time ALE-DG schemes, either in fully implicit form (17) or, more recently, in the ADER framework (Arbitrary high-order schemes using DERivatives) (18; 19).

The most suitable ALE formulation for NURBS representation is selected, and its extension to rational Bézier elements is detailed in section 3. Moreover, we explain how the regularity and hierarchy properties of NURBS can be exploited to define a very smooth mesh deformation algorithm, even in presence of large boundary displacements. In section 4, a rigorous verification of the methods is presented using two problems with analytic solutions governed by linear advection and compressible Euler equations. Then, in section 5, two more complex test-cases are considered to assess the capabilities of the proposed methodologies. The compressible Navier-Stokes are first solved for an oscillating cylinder, for different representations ranging from degree 3 to degree 6. A systematic grid refinement study is achieved

to assess the accuracy of the computations and compare the results obtained with reference data from the literature. As a second problem, we consider the compressible flow around a pitching airfoil, governed by Euler and compressible Navier-Stokes equations. Here, we focus on quantifying the impact of using curved grids, in comparison with classical piecewise-linear meshes, in order to highlight the gain obtained by improving the geometry description in the context of moving bodies. Finally, the main points learned in this study are summarized in the conclusion.

2 Discontinuous Galerkin with deformable domains

2.1 Discontinuous Galerkin discretization

We consider, for the sake of generality, a system of conservation laws of the following form:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{1}$$

with $\mathbf{W}$ being the vector of conservative variables and $\mathbf{F}$ the physical flux. The computational domain Ω is discretized using a set of elements Ω_j, $j = \{1, 2, ..., N_{el}\}$. In order to discretize eq. (1) using a DG scheme, the numerical solution $\mathbf{w}_h$ is expressed in each element of index j as:

$$\mathbf{w}_h^j = \sum_{i=1}^{N_p} \varphi_i^j(\mathbf{x})\, \mathbf{w}_i^j\,, \tag{2}$$

with φ_i^j being a generic set of basis functions and $\mathbf{w}_i^j$ the N_p degrees of freedom, defined on the j-th element. The weak formulation is obtained multiplying eq. (1) by a basis function φ_k and integrating over the elemental domain Ω_j:

$$\int_{\Omega_j} \varphi_k \frac{\partial \mathbf{w}_h}{\partial t}\, \mathrm{d}\Omega + \int_{\Omega_j} \varphi_k \nabla \cdot \mathbf{F}\, \mathrm{d}\Omega = 0. \tag{3}$$

Note that the index j on the basis functions and the discrete solution has been dropped to ease the notation. Then, integrating by parts, the classic DG discretization is found:

$$\int_{\Omega_j} \varphi_k \frac{\partial \mathbf{w}_h}{\partial t}\, \mathrm{d}\Omega = \int_{\Omega_j} \nabla \varphi_k \cdot \mathbf{F}\, \mathrm{d}\Omega - \oint_{\partial\Omega_j} \varphi_k \mathbf{F}^*\, \mathrm{d}\Gamma. \tag{4}$$

Since the discrete solution admits multiple values across elements boundaries, the physical flux in the surface integral is replaced by a numerical flux $\mathbf{F}^*(\mathbf{w}_h^+, \mathbf{w}_h^-, \mathbf{n})$, which is computed according to the values of the solution $\mathbf{w}_h^+$ and $\mathbf{w}_h^-$ that prevail at each side of the interface and a unitary normal vector $\mathbf{n}$, as in finite volume methods, with a consistent Riemann solver. We present in the next section how this formulation can be modified to account for moving meshes.

2.2 Comparison of Arbitrary Lagrangian-Eulerian formulations

The Arbitrary Lagrangian-Eulerian (ALE) approach represents a generalization of the Eulerian and Lagrangian viewpoints. Using the ALE framework, conservation laws can be solved in a referential domain, which is moving with an arbitrary velocity $\mathbf{V}_g$. As explained by

Donea et al. (20) and Venkatasubban (21), it is possible to write the conservation law 1 in its respective ALE form:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F} - \mathbf{V}_g \cdot \nabla \mathbf{W} = 0, \tag{5}$$

where $\partial/\partial t$ is the time derivative in the referential domain. In a first family of ALE-DG approaches, the weak form of the conservation law is solved in the referential domain. In order to derive a weak formulation, eq. (5) is multiplied by a test function φ_k and then integrated over the elemental domain Ω_j:

$$\int_{\Omega_j} \varphi_k \frac{\partial \mathbf{w}_h}{\partial t} \,\mathrm{d}\Omega + \int_{\Omega_j} \varphi_k \nabla \cdot \mathbf{F} \,\mathrm{d}\Omega - \int_{\Omega_j} \varphi_k \mathbf{V}_g \cdot \nabla \mathbf{w}_h \,\mathrm{d}\Omega = 0. \tag{6}$$

After integration by parts, one obtains:

$$\int_{\Omega_j} \varphi_k \frac{\partial \mathbf{w}_h}{\partial t} \,\mathrm{d}\Omega + \int_{\Omega_j} \nabla \varphi_k \cdot \mathbf{F} \,\mathrm{d}\Omega - \int_{\Omega_j} \nabla \cdot (\varphi_k \mathbf{V}_g) \mathbf{w}_h \,\mathrm{d}\Omega + \oint_{\partial \Omega_j} \varphi_k (\mathbf{F}^* - \mathbf{V}_g \mathbf{w}^*) \cdot \mathbf{n} \,\mathrm{d}\Gamma = 0, \tag{7}$$

where $\mathbf{w}^*(\mathbf{w}_h^+, \mathbf{w}_h^-)$ is the solution of the associated Riemann problem at the element interface. Manipulating equation (7) it is possible to derive several ALE-DG schemes, that are equivalent from a continuous point of view, but not at a discrete level. Indeed, constant solutions may not be preserved by the discrete scheme due to the mesh movement, as discussed in (5). In order to ensure an exact preservation of constant solutions, an ALE method has to satisfy the so-called Discrete Geometric Conservation Law (DGCL), which is specific to each numerical scheme, as explained by Guillard et al. (22).

Applying again integration by parts to the second and third terms of eq. (7), the ALE-DG scheme of Lomtev et al. (5) is found:

$$\int_{\Omega_j} \varphi_k \frac{\partial \mathbf{w}_h}{\partial t} \,\mathrm{d}\Omega + \int_{\Omega_j} \varphi_k (\nabla \cdot \mathbf{F} - \mathbf{V}_g \cdot \nabla \mathbf{w}_h) \,\mathrm{d}\Omega + \oint_{\partial \Omega_j} \varphi_k [\mathbf{F}^* - \mathbf{F} - \mathbf{V}_g (\mathbf{w}^* - \mathbf{w}_h)] \cdot \mathbf{n} \,\mathrm{d}\Gamma = 0. \tag{8}$$

This formulation can be interpreted as the ALE extension of the strong form DG, presented in (23). Provided that the numerical fluxes $\mathbf{F}^*$ and $\mathbf{w}^*$ are consistent, eq. (8) always preserves constant solutions exactly. However, the implementation is substantially different with respect to the DG method presented in eq. (4). In order to find an alternative formulation, we rewrite eq. (7) as:

$$\int_{\Omega_j} \varphi_k \frac{\partial \mathbf{w}_h}{\partial t} \,\mathrm{d}\Omega + \int_{\Omega_j} \varphi_k \mathbf{w}_h \nabla \cdot \mathbf{V}_g \,\mathrm{d}\Omega - \int_{\Omega_j} \nabla \varphi_k \cdot (\mathbf{F} - \mathbf{V}_g \mathbf{w}_h) \,\mathrm{d}\Omega + \oint_{\partial \Omega_j} \varphi_k (\mathbf{F}^* - \mathbf{V}_g \mathbf{w}^*) \cdot \mathbf{n} \,\mathrm{d}\Gamma = 0. \tag{9}$$

Then, applying the Reynolds transport theorem for moving control volumes, the following ALE-DG scheme is found:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_j} \varphi_k \mathbf{w}_h \,\mathrm{d}\Omega - \int_{\Omega_j} \nabla \varphi_k \cdot (\mathbf{F} - \mathbf{V}_g \mathbf{w}_h) \,\mathrm{d}\Omega + \oint_{\partial \Omega_j} \varphi_k \mathbf{F}^*_{ale} \,\mathrm{d}\Gamma = 0, \tag{10}$$

where $\mathbf{F}^*_{ale} = \mathbf{F}^*_{ale}(w_h^+, w_h^-, \mathbf{V}_g, \mathbf{n})$ is a numerical flux function that satisfies the modified consistency property:

$$\mathbf{F}^*_{ale}(\mathbf{w}_0, \mathbf{w}_0, \mathbf{V}_g, \mathbf{n}) = \mathbf{F}(\mathbf{w}_0) \cdot \mathbf{n} \; - (\mathbf{V}_g \cdot \mathbf{n})\mathbf{w}_0. \tag{11}$$

The formulation (10) was first introduced by Nguyen (14) and represents an elegant extension of the classic DG scheme (4) to moving domains. One can show that this scheme exactly preserves constant solutions if the following identity is verified at the discrete level:

$$\oint_{\partial\Omega_j} \varphi_k \mathbf{V}_g \, \mathrm{d}\Gamma - \int_{\Omega_j} \nabla\varphi_k \cdot \mathbf{V}_g \, \mathrm{d}\Omega = 0, \tag{12}$$

which means that the numerical quadrature has to be sufficiently accurate to exactly compute the terms of the identity. As a consequence, constant solutions are not exactly preserved for elements of general shape, such as curvilinear elements, and for arbitrary grid velocity fields.

In a second family of ALE approaches the conservation law (1) is mapped in a fixed reference domain by means of a function $\mathcal{G}$ that transforms a point $\hat{\mathbf{x}}$ in the reference domain into the corresponding point $\mathbf{x}$ in the physical space: $\mathbf{x}(t) = \mathcal{G}(\hat{\mathbf{x}}, t)$. It has been shown in (24; 15) that the equivalent of equations (1) in the reference domain can be written as:

$$\frac{\partial \widehat{\mathbf{W}}}{\partial t} + \widehat{\nabla} \cdot \widehat{\mathbf{F}} = 0, \tag{13}$$

where $\widehat{\nabla}$ is the vector of partial derivatives with respect to the reference coordinate frame, and the transformed conservative variables $\widehat{\mathbf{W}}$ and corresponding flux vector $\widehat{\mathbf{F}}$ are defined as:

$$\widehat{\mathbf{W}} = |\mathrm{J}_G| \, \mathbf{W}, \quad \widehat{\mathbf{F}} = |\mathrm{J}_G| \, \mathrm{J}_G^{-1}(\mathbf{F} - \mathbf{V}_M \mathbf{W}), \tag{14}$$

where J_G is the Jacobian matrix of the map and $\mathbf{V}_M$ is the map velocity:

$$\mathbf{V}_M = \frac{\partial \mathcal{G}}{\partial t}. \tag{15}$$

The mapped ALE technique consists in solving the transformed system of conservation laws (13) in the reference domain. The mapped ALE-DG method has been introduced by Persson et al. (15) in the context of compressible fluid mechanics. The reference domain is discretized and the basis functions are defined in the reference space on each element $\widehat{\Omega}_j$. Then, a DG discretization is applied to (13), obtaining:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int_{\widehat{\Omega}_j} \varphi_k \mathbf{w}_h \, |\mathrm{J}_G| \, \mathrm{d}\widehat{\Omega} - \int_{\widehat{\Omega}_j} \widehat{\nabla}\varphi_k \cdot \mathrm{J}_G^{-1}(\mathbf{F} - \mathbf{V}_M \mathbf{w}_h) \, |\mathrm{J}_G| \, \mathrm{d}\widehat{\Omega} \\ + \oint_{\partial\widehat{\Omega}_j} \varphi_k \mathrm{J}_G^{-1} \mathbf{F}^*_{ale} \, |\mathrm{J}_G| \, \mathrm{d}\widehat{\Gamma} = 0, \end{aligned} \tag{16}$$

where the map $\mathcal{G}$ is assumed to be $\mathcal{C}^1$. It is worthwhile noting that, even though the mapped ALE approach is significantly different, many similarities can be observed comparing equations (16) and (10). Due to the presence of the metric terms of the map within the discrete equations, this method violates the DGCL in general. However, as explained in (15), the preservation of constant solutions can be enforced by slightly modifying eq. (13) and adding a scalar conservation equation that has to be solved with the same DG scheme, see (15) for

more details. Since arbitrary maps can be employed, the mapped ALE approach yields an extremely flexible method and allows high-order domain deformations. On the other hand, computing the map function may not be straightforward when the movement law is not known *a priori*. After this synthesis on ALE methods for DG, we describe in the following section how to adapt one of the analyzed schemes to CAD-consistent representations.

3 NURBS-based ALE-DG

3.1 Basis functions

The construction of a CAD-consistent scheme necessarily relies on bases used in CAD. The first approach to interactive geometric design consisted in the use of Bézier surfaces, defined as (2):

$$\mathbf{S}(\xi,\eta) = \sum_{i_1=1}^{p+1}\sum_{i_2=1}^{p+1} B_{i_1}^p(\xi)\, B_{i_2}^p(\eta)\, \mathbf{x}_{i_1 i_2}, \tag{17}$$

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

$$B_i^p(\varphi) = \binom{p}{i-1} \varphi^{i-1}(1-\varphi)^{p-i-1}, \tag{18}$$

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 (25):

$$R_{i_1 i_2}^p(\xi,\eta) = \frac{B_{i_1}^p(\xi)\, B_{i_2}^p(\eta)\, \omega_{i_1 i_2}}{\sum_{j_1=1}^{p+1}\sum_{j_2=1}^{p+1} B_{j_1}^p(\xi)\, B_{j_2}^p(\eta)\, \omega_{j_1 j_2}}, \tag{19}$$

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

$$\mathbf{S}(\xi,\eta) = \sum_{i_1=1}^{p+1}\sum_{i_2=1}^{p+1} R_{i_1 i_2}^p(\xi,\eta)\, \mathbf{x}_{i_1 i_2} \tag{20}$$

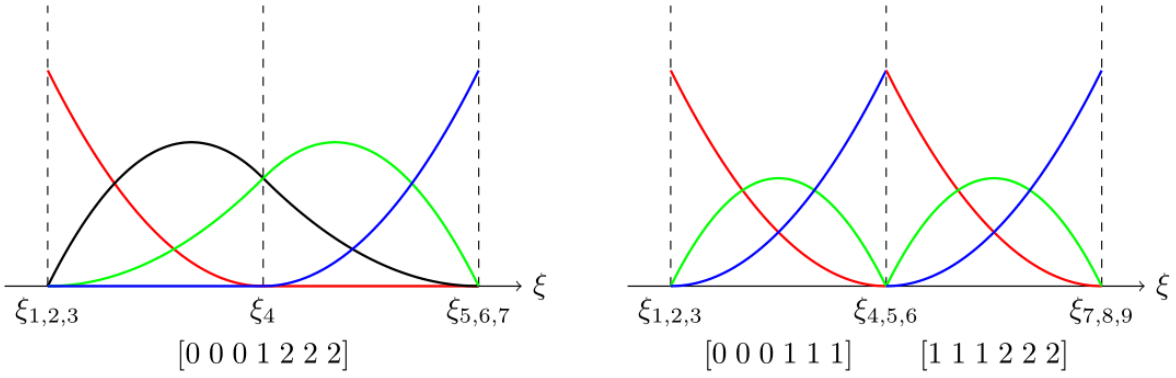
However, complex geometries require a high-degree basis when represented using a single polynomial patch, which has several drawbacks from numerical point of view. Therefore, as explained in (1), CAD software representations are commonly based on B-Splines, rather than Bézier curves. To introduce B-Splines, 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:

$$N_i^0(\xi) = \begin{cases} 1 & \text{if } \xi_i \le \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

$$N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi). \tag{22}$$

NURBS functions are the rational extension of B-Splines:

$$R_{i_1 i_2}^p(\xi,\eta) = \frac{N_{i_1}^p(\xi)\, N_{i_2}^p(\eta)\, \omega_{i_1 i_2}}{\sum_{j_1=1}^{n_1}\sum_{j_2=1}^{n_2} N_{j_1}^p(\xi)\, N_{j_2}^p(\eta)\, \omega_{j_1 j_2}}. \tag{23}$$



(a) Quadratic B-Spline functions

(b) Split Bernstein polynomials

Figure 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]

Since Rational Bernstein functions (19) allow to generate discontinuous solutions at element interfaces, they are a suitable basis for a DG discretization. This is not true for NURBS, that can be used for a Continuous Galerkin (CG) approximation as in classical Isogeometric Analysis (1). As described in (12; 25), it is possible to generate a DG-compliant representation, starting from CAD geometries, by a procedure called Bézier extraction, which 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.

3.2 Formulation

The rational Bernstein functions (19) are defined on a parametric domain, therefore, as in Isogeometric Analysis, the elements are mapped from the physical space to the parametric domain. In this context, a mapped ALE approach would require an additional map function to simulate the moving domain, resulting in a complex and cumbersome implementation. For this reason we opt to solve the equations in the moving referential domain. In particular, the formulation (10) is chosen, as the implementation of the strong form ALE-DG scheme (8) would represent a major redesign of the framework we proposed in (12).

We assume that, in general, the physical flux can be expressed as:

$$\mathbf{F} = \mathbf{F}_c(\mathbf{W}) - \mathbf{F}_v(\mathbf{W}, \nabla\mathbf{W}) \tag{24}$$

where $\mathbf{F}_c$ is the convective flux and $\mathbf{F}_v$ is the viscous flux. $\mathbf{G} = \nabla\mathbf{W}$ is the gradient of the conservative variables. The second order derivatives are discretized with the Local Discontinuous Galerkin (LDG) approach (26). We thus rewrite eq. (5) as a system of first order equations:

$$\begin{cases} \dfrac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{W}) - \nabla \cdot \mathbf{F}_v(\mathbf{W}, \mathbf{G}) - \mathbf{V}_g \cdot \nabla\mathbf{W} = 0, \\ \mathbf{G} - \nabla\mathbf{W} = 0. \end{cases} \tag{25}$$

In the NURBS-based DG framework, each element is a rational Bézier patch, as represented in figure 2. The geometry $\mathbf{x}$ and the local solution fields $\mathbf{w}_h$ and $\mathbf{g}_h$ are expressed

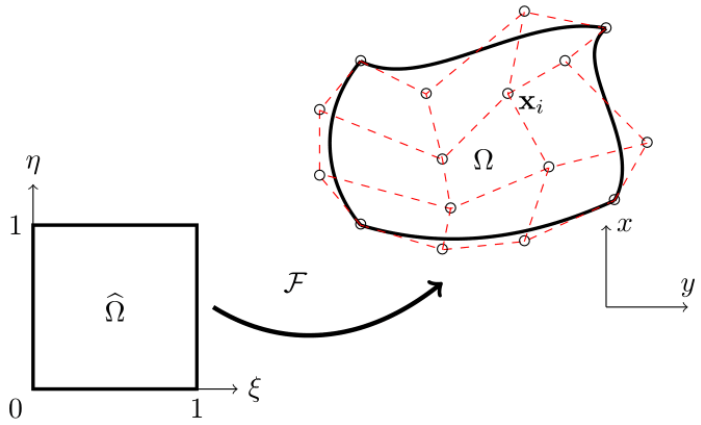


Figure 2: example of Bézier patch

using the same basis functions:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{w}_h \\ \mathbf{g}_h \end{pmatrix} = \sum_{i=1}^{(p+1)^2} R_i(\xi,\eta) \begin{pmatrix} \mathbf{x}_i \\ \mathbf{w}_i \\ \mathbf{g}_i \end{pmatrix}, \tag{26}$$

where $R_i(\xi,\eta)$ are the rational Bernstein functions $R^p_{i_1,i_2}(\xi,\eta)$ with a trivial index change and omission of degree p. The isogeometric paradigm is adopted to extend the scheme (10) and discretize eq. (25). Using the map defined by the rational Bézier functions on each element Ω_j, the integrals are transposed from the physical space to the parametric unit square $\widehat{\Omega} = [0,1]^2$, obtaining the NURBS-based ALE-DG formulation:

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{w}_i \displaystyle\int_{\widehat{\Omega}} R_k R_i \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega}\right) = \displaystyle\int_{\widehat{\Omega}} \nabla R_k \cdot \left(\mathbf{F}_c - \mathbf{F}_v - \mathbf{V}_g \mathbf{w}_h\right) \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} \\ \qquad\qquad\qquad\qquad - \displaystyle\oint_{\partial\widehat{\Omega}} R_k \left(\mathbf{F}^*_{ale} - \mathbf{F}^*_v\right) \left|\mathrm{J}_{\Gamma_j}\right| \mathrm{d}\widehat{\Gamma}, & \text{(27a)} \\ \mathbf{g}_i \displaystyle\int_{\widehat{\Omega}} R_k R_i \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} = \displaystyle\int_{\widehat{\Omega}} \nabla R_k \mathbf{w}_h \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} - \displaystyle\oint_{\partial\widehat{\Omega}} R_k \mathbf{W}^* \left|\mathrm{J}_{\Gamma_j}\right| \mathrm{d}\widehat{\Gamma}. & \text{(27b)} \end{cases}$$

With respect to the original method, the integrals in (27) contain the additional metric terms $|\mathrm{J}_{\Omega_j}|$ and $|\mathrm{J}_{\Gamma_j}|$, which are the Jacobians of the coordinate transformation defined by equation (26). Furthermore, the gradient of the basis functions on each element Ω_j, in the physical space, is computed by means of the Jacobian matrix of the isogeometric map J_{Ω_j}:

$$\nabla R_k = \mathrm{J}^{-T}_{\Omega_j} \widehat{\nabla} R_k, \tag{28}$$

where $\widehat{\nabla} R_k$ is the gradient of the basis function in the parametric domain. The Gauss-Legendre quadrature rule is employed to approximate the integrals. The viscous numerical fluxes $\mathbf{F}^*_v$ and $\mathbf{W}^*$ are computed with the LDG approach (26). For the numerical flux function $\mathbf{F}^*_{ale}$, any existing Riemann solver can be adapted. It is indeed possible to write the Jacobian matrix of the convective physical flux in the direction $\boldsymbol{\tau}$ as:

$$\mathbf{J}_{ale} \cdot \boldsymbol{\tau} = \mathbf{J}_F \cdot \boldsymbol{\tau} - (\mathbf{V}_g \cdot \boldsymbol{\tau})\mathbf{I}, \tag{29}$$

with $\mathbf{J}_F$ the Jacobian of the convective flux on the fixed mesh and $\mathbf{I}$ the identity matrix. It is thus trivial to show that the eigenvectors of $\mathbf{J}_{ale}$ are invariant with respect to the grid velocity, whereas for the eigenvalues the following relation holds:

$$S_{ale} = S_0 - \mathbf{V}_g \cdot \boldsymbol{\tau}, \tag{30}$$

where S_0 are the eigenvalues computed without mesh movement. It is thus possible to obtain a consistent numerical flux for moving meshes using the modified wavespeeds computed with eq. (30) and adding the flux contribution generated by mesh movement to the fixed grid Riemann solver. Two examples will be provided in section 4.

Equation (27b) does not contain a time derivative, therefore it can be solved separately from eq. (27a) within each time iteration. The system of equations (27) can be thus rewritten in a compact form:

$$\frac{d}{dt}(\mathcal{M}\mathbf{w}) = \mathcal{R}(\mathbf{w}_h, \mathbf{V}_g), \tag{31}$$

where the residual $\mathcal{R}$ is the right-hand side of eq. (27a) and $\mathcal{M}$ is the block-diagonal mass matrix, which is generally not constant in time, except for rigid mesh motion. In this paper we consider explicit Runge-Kutta (RK) time integration to solve the system of ordinary differential equations (31). In particular, we either employ the classic four-stage 4th-order Runge-Kutta (RK) scheme or the Strong Stability Preserving (SSP) 3rd-order RK method of Gottlieb et al. (27).

One can observe that, due to the introduction of the isogeometric map, the mathematical structure of the proposed formulation (27) is nearly identical to that of the mapped ALE-DG scheme (16). However, in our approach the map is local to each element and it is only used to transform the integrals of the weak formulation. On the other hand, in the technique proposed by Persson et al. (15), the map is global and it is applied to modify the conservation laws at the continuous level. Similarly to eq. (16), the proposed scheme (27) violates the DGCL because of the use of rational basis functions and the presence within the integrals of the metric terms of the coordinate transformation. Indeed, the equivalent of the identity (12) for equation (27a) is:

$$\oint_{\partial\widehat{\Omega}} R_k \mathbf{V}_g \left|\mathrm{J}_{\Gamma_j}\right| \mathrm{d}\widehat{\Gamma} - \int_{\widehat{\Omega}} \mathrm{J}_{\Omega_j}^{-T} \widehat{\nabla} R_k \cdot \mathbf{V}_g \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} = 0. \tag{32}$$

In general the basis functions R_k are rational, therefore exact integration is not achievable by means of Gauss-Legendre quadrature. In the case of uniform weights, R_k is a polynomial, but the inverse of the Jacobian matrix is not. As a consequence, numerical quadrature cannot be exact, as remarked by Sevilla et al. (28) as well. We decide however to not enforce the DGCL, as the preservation of constant solutions is not a necessary condition for stability and high-order accuracy, as explained in (15). We show in sections 4 and 5 that this choice does not impact the accuracy of the proposed approach.

3.3 Mesh movement technique

The second key ingredient of an ALE method is the mesh movement algorithm. The majority of the ALE approaches presented so far in the literature are based on rectilinear grids. Therefore, the grid velocity in each element its defined by the velocities of its vertices, usually computed via elastic analogy (29) or graph theory based methods (5). On the contrary, our

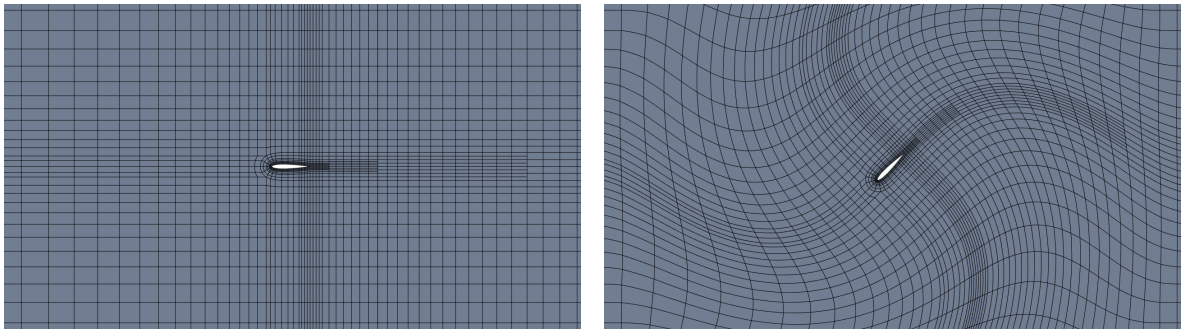
formulation is based on a high-order geometric representation, enabling more control over the shape of the elements. Therefore, a wider range of mesh velocities can be employed with respect to traditional approaches. We decide to adopt the isogeometric paradigm to define the grid velocity field $\mathbf{V}_g$ in each element:

$$\mathbf{V}_g = \sum_{i=1}^{(p+1)^2} R_i(\xi, \eta)\mathbf{v}_{g,i}\,, \tag{33}$$

with $\mathbf{v}_{g,i}$ being the velocity of the control point $\mathbf{x}_i$. Obviously, we impose the additional constraint that the velocities match exactly at the interface between two elements. This choice leads to a completely unified description of all the variables appearing in the ALE formulation: the solution fields, the geometry and the mesh velocity. Thus, the time evolution of the control point net is determined by:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_{g,i}. \tag{34}$$

Given the distribution of control point velocities, the eq. (34) is integrated with the same Runge-Kutta method employed for the DG formulation. In the present work, we analyse cases where the mesh movement is imposed in the whole computational domain. The control point velocities are thus explicitly assigned in each time step. Nevertheless, the algorithm can be easily extended to problems where the grid velocity is the result of a multi-physics coupling, such as fluid-structure interaction.



(a) Initial (b) Deformed

Figure 3: Mesh movement example

The proposed technique is capable of generating arbitrarily high-order mesh deformations. Moreover, we take advantage of the use of Bézier surfaces, which possess important regularity properties, such as the convex hull property (25). Consequently, the mesh quality can be preserved over time, even in presence of large deformations, as illustrated in fig. 3. It is also possible to employ piecewise defined distributions of $\mathbf{v}_{g,i}$ without loss of smoothness, thanks to the non-interpolatory nature of control points.

The velocity field $\mathbf{V}_g$ inside each element is always $\mathcal{C}^\infty$, whereas, across the element boundaries, the derivatives can be discontinuous. Nevertheless, it is possible to ensure a higher global regularity by exploiting the hierarchical properties of NURBS: when the rational Bézier elements are generated by recursive splitting of a larger NURBS patch of degree p, the geometric regularity at element interfaces is $\mathcal{C}^{p-1}$ (when knots are distinct). Therefore,

a similar splitting strategy can be employed for the velocity field to obtain a highly smooth deformation field of regularity $\mathcal{C}^{p-1}$ too. This hierarchical splitting of the geometrical elements is also an important tool for moving non-conformal meshes. Indeed, a naive application of any non-linear deformations would generate holes between elements in the proximity of hanging nodes. This phenomenon occurs because only linear transformations applied to control points are identical to transforming the curve itself (25). Anyway, this difficulty can be easily circumvented by managing local refinement in a quad-tree framework and splitting hierarchically the elements (13), as in fig. 3. The potential and the flexibility of the proposed mesh movement algorithm are demonstrated in sections 4 and 5.

4 Verification

4.1 Advection equation

The proposed approach is firstly tested on a scalar advection problem. As numerical flux function we adopt a modified Local Lax-Friedrichs flux that suits the Arbitrary Lagrangian-Eulerian formulation:

$$\begin{aligned}\mathbf{F}^*_{ale}(w^+, w^-, \mathbf{V}_g, \mathbf{n}) = \frac{1}{2}(w^+ + w^-)(\mathbf{V} - \mathbf{V}_g)\cdot\mathbf{n} \\ + \frac{1}{2}|(\mathbf{V} - \mathbf{V}_g)\cdot\mathbf{n}|(w^+ - w^-),\end{aligned} \tag{35}$$

where w is the scalar variable and $\mathbf{V}$ is the advection velocity, that we assume constant in space. It is straightforward to verify that the proposed numerical flux complies with the consistency condition (11). A convergence analysis is performed on the following test case:

$$\begin{cases} w_0(\mathbf{x}) = e^{-(x-2)^2 - y^2} \\ u(t) = -4\pi\sin(2\pi t) \\ v(t) = 4\pi\cos(2\pi t) \end{cases} \tag{36}$$

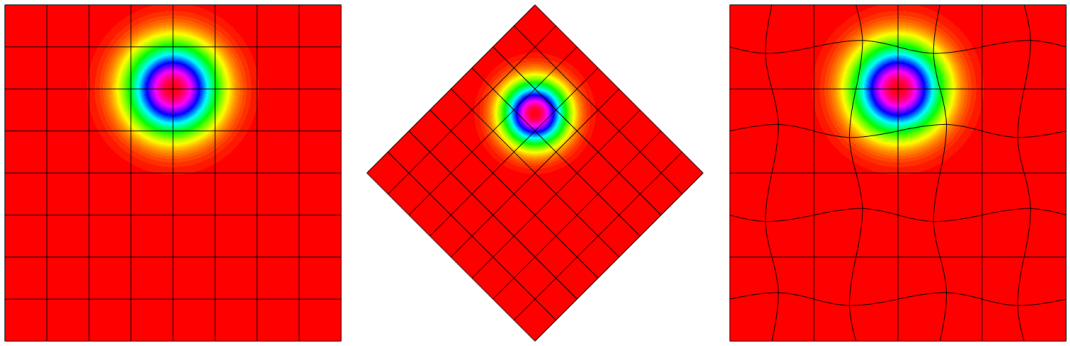
The initial solution is advected along an anticlockwise circle of radius 2, thus, the analytic solution is:

$$w(\mathbf{x}, t) = e^{-[x - 2\cos(2\pi t)]^2 - [y - 2\sin(2\pi t)]^2} \tag{37}$$

The computational domain is $[-4, 4] \times [-4, 4]$, discretized with a cartesian grid, and the boundary numerical fluxes are computed by means of the exact solution. The error in L2-norm is evaluated using numerical quadrature after one rotation period. In order to assess the accuracy of the numerical scheme, two mesh velocity laws are tested, as illustrated in Fig. 4, a rigid clockwise rotation about the origin and a sinusoidal deformation where the control points move with the following velocity:

$$u_g(\mathbf{x}, t) = v_g(\mathbf{x}, t) = \sin\left(\frac{N_x\pi}{L_x}x\right)\sin\left(\frac{N_y\pi}{L_y}y\right)\sin(2\pi t). \tag{38}$$

We compare the results obtained with the 3 configurations. Polynomials up to fifth degree are tested and optimal convergence rates are verified for each degree, as shown in figure 5. It is also possible to observe that results on rigidly moving mesh do not substantially differ from the fixed domain simulation, whereas the error increases when the mesh deforms over time,



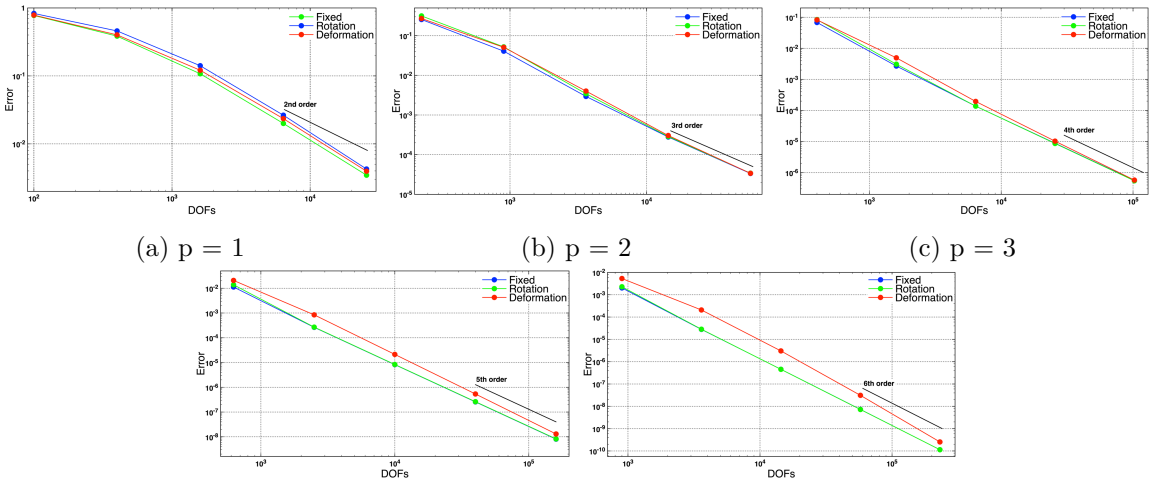
(a) Fixed mesh (b) Rotating mesh (c) Deforming mesh

Figure 4: Advection test case, solution at t=0.25, p=5

especially for high polynomial degrees. Defining r_d as the ratio between the errors computed respectively on the deforming and the fixed mesh:

$$r_d = \frac{\epsilon_{deforming}}{\epsilon_{fixed}}, \tag{39}$$

the maximum value of r_d for cubic functions is around 1.8, whereas for quartic polynomials a factor 3.2 is found in the worst case and for the quintic basis a value of 7.4 is reached. This steep increase is caused by the isogeometric map. Indeed, the Jacobian matrix becomes highly non-linear when the mesh is subject to high-order deformation, resulting in a loss of accuracy of the Gauss quadrature rule with respect to the cartesian grid.



(a) p = 1 (b) p = 2 (c) p = 3

(d) p = 4 (e) p = 5

Figure 5: Convergence analysis, 2D advection equation

4.2 Euler equations

In order to validate the proposed approach for nonlinear problems, we consider the compressible Euler equations for the second test case. The numerical flux is computed using the HLL

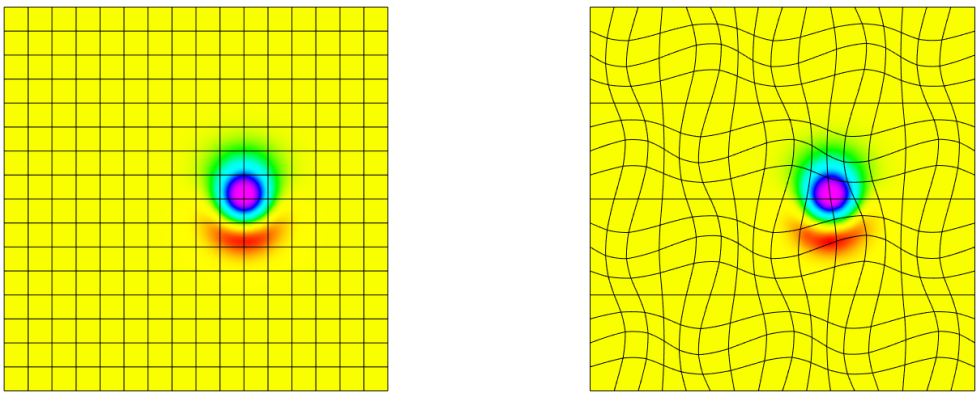
Riemann solver (30), which has been adapted to take into account the mesh movement flux:

$$\mathbf{F}^*_{ale} = \begin{cases} \mathbf{F}^-_{ale} = \mathbf{F}(\mathbf{w}^-_h) - (\mathbf{V}_g \cdot \mathbf{n})\mathbf{w}^-_h & \text{if } S^-_{ale} \geq 0 \\ \dfrac{S^+_{ale}\mathbf{F}^-_{ale} - S^-_{ale}\mathbf{F}^+_{ale} + S^+_{ale}S^-_{ale}(\mathbf{w}^+_h - \mathbf{w}^-_h)}{S^+_{ale} - S^-_{ale}} & \text{if } S^-_{ale} \leq 0 \leq S^+_{ale} \\ \mathbf{F}^+_{ale} = \mathbf{F}(\mathbf{w}^+_h) - (\mathbf{V}_g \cdot \mathbf{n})\mathbf{w}^+_h & \text{if } S^+_{ale} \leq 0 \end{cases} \tag{40}$$

where S^-_{ale} and S^+_{ale} are respectively the minimum and the maximum wave speeds computed with the two states $\mathbf{w}^-_h$ and $\mathbf{w}^+_h$. The proposed numerical flux function (40) respects the consistency condition (11). The advection of an isentropic vortex (23) is considered, with the following analytic solution:

$$\begin{cases} \rho = \left(1 - \dfrac{\gamma - 1}{16\gamma\pi^2}\beta^2 e^{2(1-r^2)}\right)^{\frac{1}{\gamma-1}} \\ u = 1 - \beta\dfrac{y - y_0}{2\pi}e^{1-r^2} \\ v = \beta\dfrac{x - t - x_0}{2\pi}e^{1-r^2} \\ p = \rho^\gamma \end{cases} \tag{41}$$

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 conditions are weakly imposed using the analytic solution. The simulation is run on a fixed cartesian grid and on a deforming mesh, whose control points follow the sinusoidal law (38) used for the advection test case.

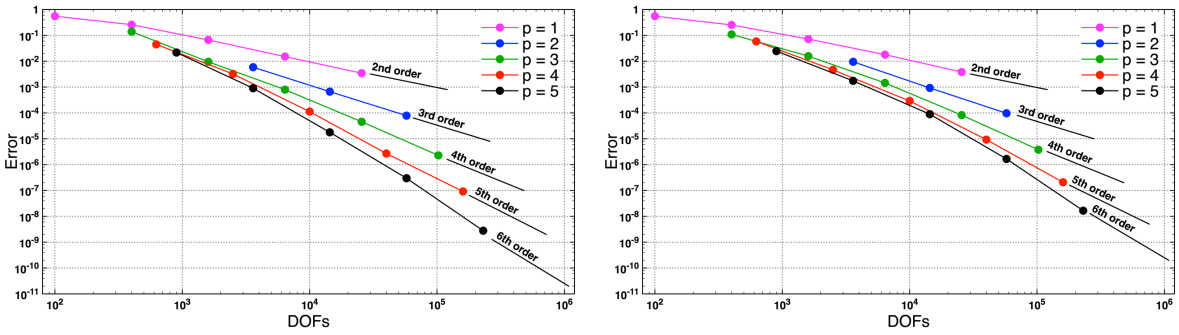


(a) Fixed mesh (b) Deforming mesh

Figure 6: Isentropic vortex, energy, t=1.25, p=3

We compute the L2 norm of the error of the total 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 levels between the two setups. The evolution of the ratio r_d (39) confirms the results obtained for the advection equation: for cubic polynomials the maximum ratio is around 1.8, whereas for the quartic basis a factor 3.5 is found in the worst case and for quintic functions a value of 6 is reached. The sensitivity with respect to mesh deformation is a natural consequence of the

quadrature error, as already evidenced for the advection problem. The proposed numerical scheme is hence capable of accurately solving non-linear conservation equations on moving meshes, preserving the high-order convergence rate of the DG discretization.



(a) Fixed mesh

(b) Deforming mesh

Figure 7: Convergence analysis, Euler equations, L2 error of total energy

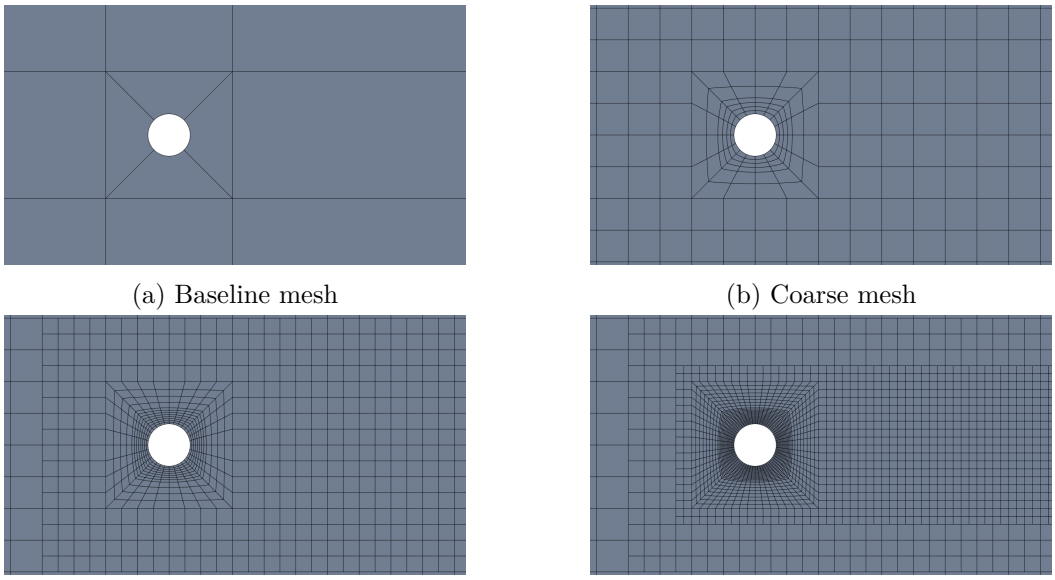
5 Case study

5.1 Oscillating cylinder

In the next example we exploit the capability of rational Bézier curves of exactly representing conic shapes. We simulate the bidimensional viscous flow around a circular cylinder oscillating in the crossflow direction. Indeed, using 4 rational Bézier curves, the circular boundary can be exactly described and a very coarse baseline mesh can be generated. The external boundary of the computational domain is the rectangle $[-25D, 100D] \times [-25D, 25D]$, where D is the diameter, and the center of cylinder is the origin (0,0). The actual computational mesh is obtained by locally refining the initial patches, without the need of refitting, since the coarse grid already provides an exact representation of the domain. In order to perform a mesh convergence study, 3 different refinement levels, shown in Fig. 8, are used for the simulations:

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

The oscillating cylinder flow has been widely investigated in the literature, using both experimental and numerical techniques (14; 31; 32). It is characterised by complex nonlinear physics, such as the lock-in phenomenon, that consists in the synchronisation of the vortex shedding with the oscillation of the body. Due to its nonlinear nature, the problem is extremely sensitive to acoustic perturbations, therefore, it is important to avoid the interference between the acoustic waves possibly reflected at the boundaries and the flow in the vicinity of the cylinder. This is achieved by choosing a sufficiently large computational domain, combined with far-field boundary conditions based on Riemann invariants and a layer of coarse cells in the proximity of the external boundaries, in order to dissipate the remaining reflected acoustic noise.



(a) Baseline mesh (b) Coarse mesh

(c) Intermediate mesh (d) Fine mesh

Figure 8: Different refinement levels for the cylinder simulation

The convergence study is conducted using rational functions built from polynomials of degrees from 3 up to 6. We consider a freestream Mach number equal to 0.2, in order to avoid strong compressibility effects, and a Reynolds number of 500 with respect to the diameter. The cylinder motion is given by:

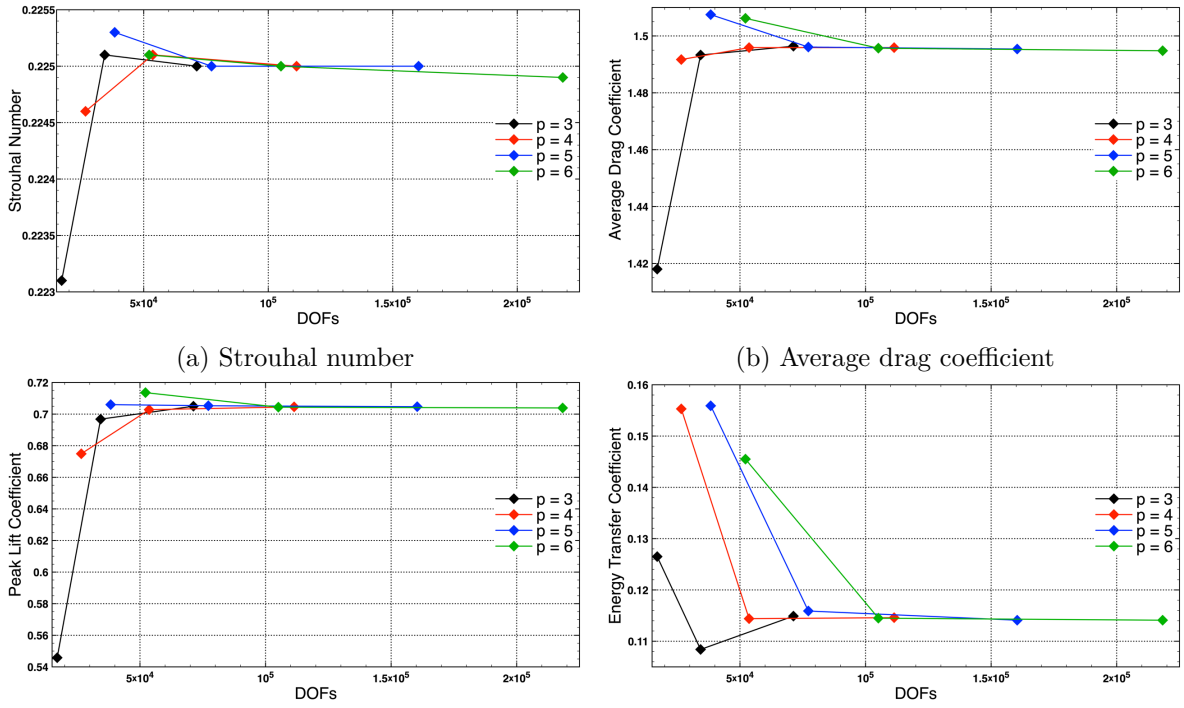
$$y = A\sin(2\pi f t), \tag{42}$$

with $A = 0.25D$ and $f = 0.875 f_{sh}$, with f_{sh} being the vortex shedding frequency of the fixed cylinder at equal Mach and Reynolds numbers. The chosen configuration of A and f lies in the lock-in range. For each combination of degree and refinement level the oscillation frequency f is determined from the value of f_{sh} obtained with the corresponding fixed mesh simulation.

A first set of results, shown in Fig. 9, is obtained by running the test case with a rigid mesh movement law. Once a periodic solution is established, we examine more precisely the peak lift coefficient $\hat{C}_l$, the average drag coefficient $\bar{C}_d$, and the energy transfer coefficient E, which quantifies the mechanical work done by the fluid on the cylinder over a lock-in cycle:

$$E = \frac{1}{\frac{1}{2}\rho_\infty u_\infty^2 D^2} \oint \mathbf{f} \cdot \mathrm{d}\mathbf{x} = \oint \frac{L}{\frac{1}{2}\rho_\infty u_\infty^2 D} \,\mathrm{d}\Big(\frac{y}{D}\Big) = \oint C_l \,\mathrm{d}\alpha, \tag{43}$$

where $\mathbf{f}$ is the force vector, $\mathbf{x}$ the displacement, L the lift force and α the dimensionless displacement in the y direction. The convergence of the Strouhal number of the fixed cylinder is illustrated for the sake of completeness. The lock-in limit cycle is well reproduced with all the basis degrees and a faster convergence is observed with functions of degree 4,5 and 6. Note that using high-order bases on the coarsest mesh does not lead to significant improvements. In order to estimate E with sufficient precision, it is necessary to use at least the intermediate refinement level. Hence, increasing the basis degree on coarse grids becomes inefficient, indeed, the severe stability restrictions due to explicit time stepping significantly raises the computational costs without benefits in terms of accuracy. Therefore, a combination of h and



(a) Strouhal number

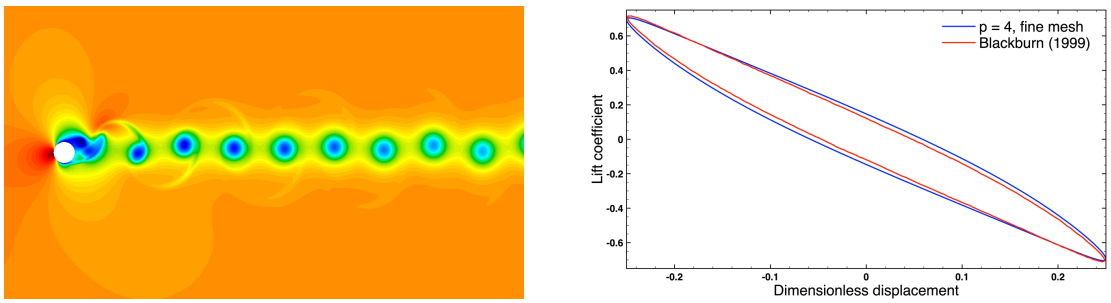
(b) Average drag coefficient

(c) Peak lift coefficient

(d) Energy transfer coefficient

Figure 9: Oscillating cylinder flow, convergence study

p refinements has to be used in order to optimise the computational cost, for a given level of accuracy. The numerical solution of the lock-in limit cycle is compared to the results provided by Blackburn et al. (32). The comparison is presented in Fig. 10b, the lift-displacement curve obtained with the proposed scheme is in close comparison with the reference cycle.



(a) Density, p=4, fine mesh

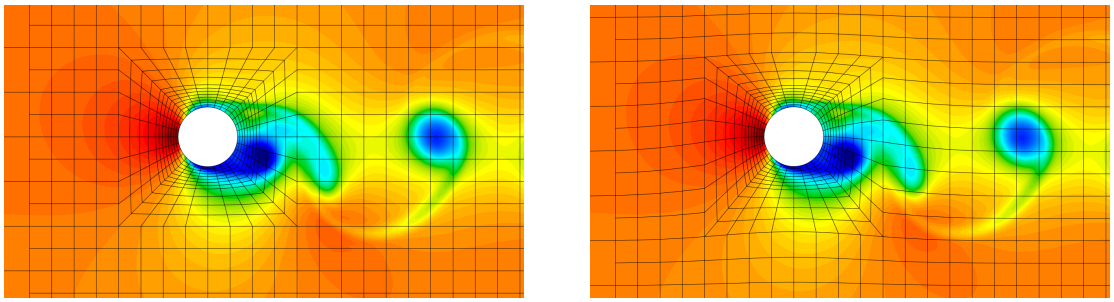
(b) Lock-in limit cycle

Figure 10: Numerical solution of the oscillating cylinder test case

The proposed test case is also used to evaluate the influence of the grid movement law in the presence of moving walls. Therefore, the results obtained with rigid mesh displacements are compared to the simulations performed with a smooth deformation velocity field, where the control point net moves with the following law:

$$v_g(\mathbf{x},t) = v_c(t)\, e^{-\frac{\psi^2(\mathbf{x},t)}{d^2}} \tag{44}$$

where $v_c(t)$ is the velocity of the cylinder, $\psi(\mathbf{x}, t)$ is the distance of the point $\mathbf{x}$ from the cylinder surface at time t and d is a characteristic length, that controls the propagation of the boundary displacement within the domain. As shown in Fig. 11b, this profile allows the elements close to the wall to move almost rigidly, preserving the initial mesh quality. At the same time, thanks to the exponential decay far from the cylinder, only a small portion of the domain is deformed.



(a) Rigid mesh (b) Deforming mesh

Figure 11: Comparison of movement laws, density fields, p=5, intermediate mesh

In Fig. 11a and 11b the comparison of the density field in the periodic regime for the rigid and deforming mesh movement laws is presented for the intermediate mesh level and quintic basis. The two solutions are nearly identical and, for the deforming mesh, no spurious effects generated by the lack of freestream preservation can be observed, even if the numerical quadrature of rational functions is less accurate. The same conclusion is obtained by comparing the evolution of the aerodynamic coefficients, reported in Fig. 12 for the same mesh and basis configuration.

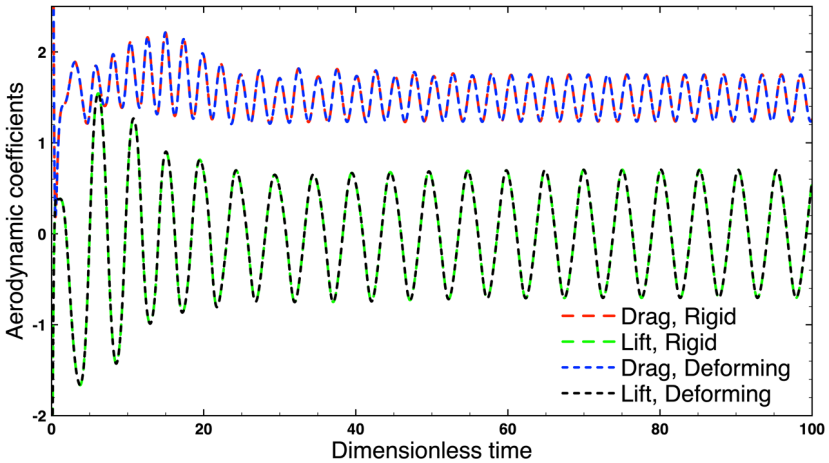


Figure 12: Comparison of movement laws, force coefficients, p=5, intermediate mesh

An extensive comparison of the two mesh movement techniques has been carried out, for all the combination of meshes and basis degrees. We report in table 1 the energy transfer coefficient, as this quantity is more sensitive than the force coefficients. For a given basis degree, a discrepancy in the third significant figure can be observed for coarse meshes. However, both values are significantly far from the converged result. Refining the grid, the difference

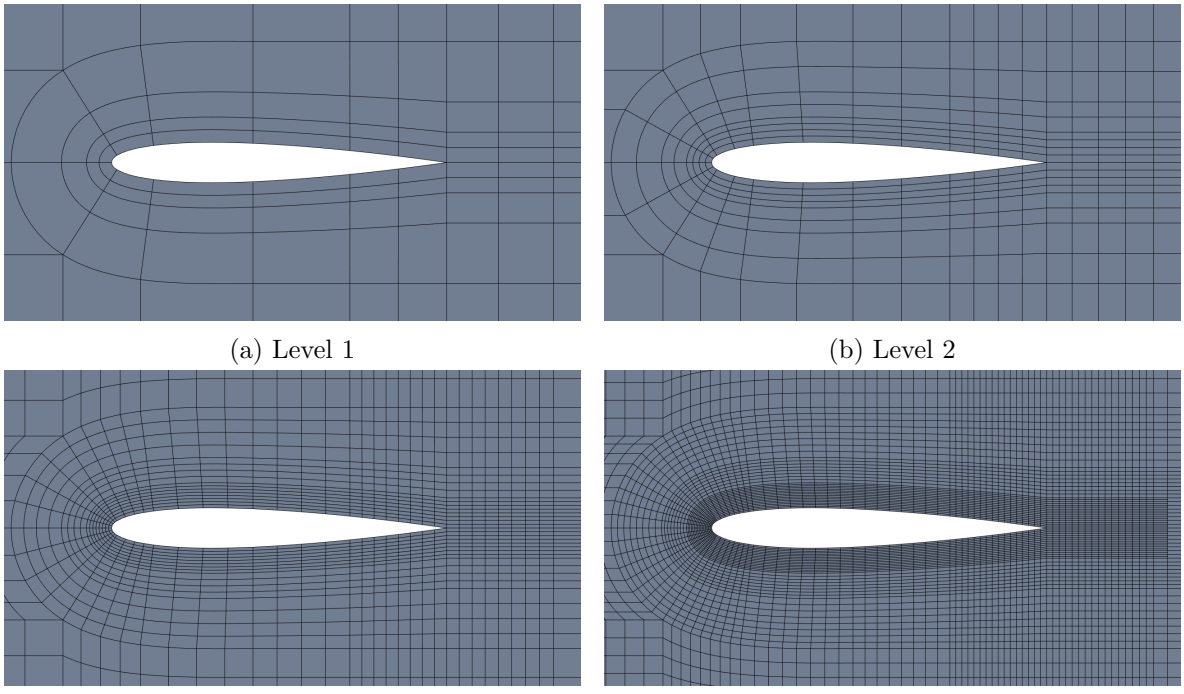
between the two movement techniques becomes smaller and the predicted values are equally accurate.

Table 1: Energy transfer coefficient for different mesh movement laws

	Rigid			Deformation		
degree	coarse	intermediate	fine	coarse	intermediate	fine
3	0.1265	0.1084	0.1149	0.1298	0.1086	0.1151
4	0.1553	0.1144	0.1146	0.1566	0.1150	0.1148
5	0.1559	0.1159	0.1141	0.1590	0.1154	0.1141
6	0.1455	0.1145	0.1141	0.1422	0.1145	0.1141

5.2 Pitching airfoil

In the second case study, the compressible flow around a pitching NACA 0012 airfoil is investigated. Contrarily to the cylinder case, the geometry cannot be exactly represented with NURBS functions, therefore, the boundary of the computational domain is approximated using polynomial curve fitting. We employ a cubic Bernstein basis for both the geometry and the solution fields. The goal of the proposed test case is to assess the impact of using curved grids. This is achieved via a double convergence study with 4 meshes, shown in Fig. 13. The first convergence analysis is performed using the high-order boundary representation, whereas a piecewise linear approximation is used for the second. For each refinement level, the low-order grid is obtained by linearizing the respective curved mesh.



(a) Level 1 (b) Level 2

(c) Level 3 (d) Level 4

Figure 13: Meshes for the pitching airfoil convergence study

The airfoil is subject to a pure pitching motion about mid-chord. The angle of attack oscillates in time with a sinusoidal law:

$$\alpha(t) = -A\sin(2\pi f t), \tag{45}$$

with A and f being respectively the pitch amplitude and frequency. The rotation of the airfoil is transferred to the mesh through a piecewise defined velocity field:

$$\begin{cases} u_g(\mathbf{x},t) = -\tilde{\omega}\,(y - y_c) \\ v_g(\mathbf{x},t) = \tilde{\omega}\,(x - x_c) \end{cases} \tag{46}$$

where (x_c, y_c) is the mid-chord position and $\tilde{\omega} = \dot{\alpha}\,\sigma(\mathbf{x},t)$, with σ being a blending function of the following form:

$$\sigma(\mathbf{x},t) = \begin{cases} 1, & \text{if } R \leq R_{int} \\ \dfrac{1}{2}\left[1 + \cos\left(\pi \dfrac{R - R_{int}}{R_{ext} - R_{int}}\right)\right], & \text{if } R_{int} < R < R_{ext} \\ 0, & \text{if } R \geq R_{ext} \end{cases} \tag{47}$$

with R the distance with respect to the mid-chord position. The blending function divides the domain into 3 regions, the internal one, delimited by a circle of radius R_{int}, moves rigidly with the airfoil, while the area outside the circle of radius R_{ext} is fixed. The 2 regions are connected by a deforming transition ring. The proposed function (47) is C^1, in order to have a smoother deformation profile, however a C^0 class function could be employed as well.

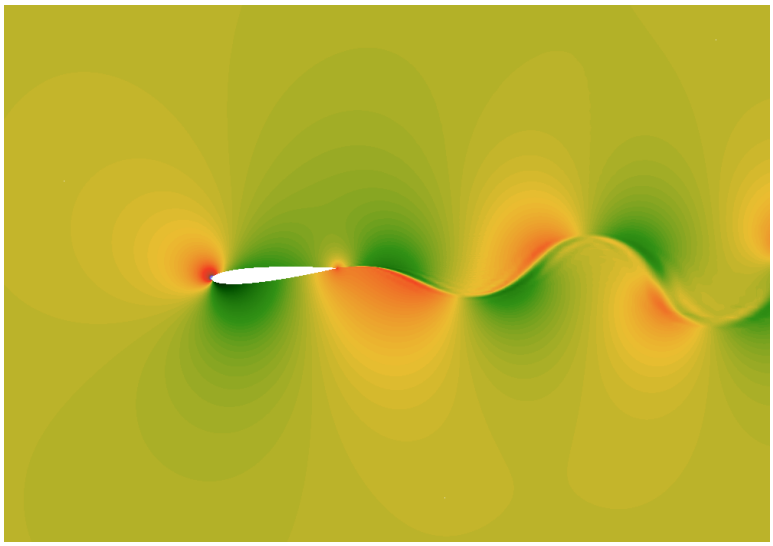
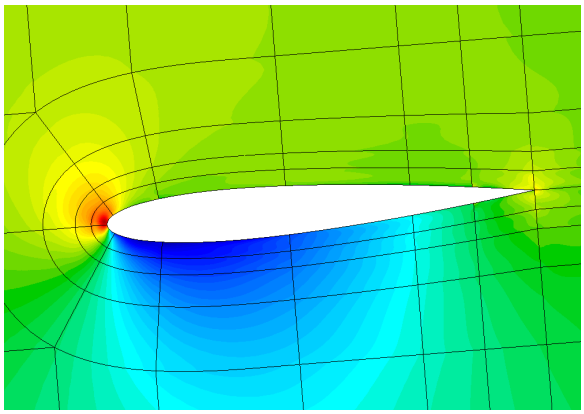


Figure 14: Pitching airfoil, inviscid flow, streamwise momentum field

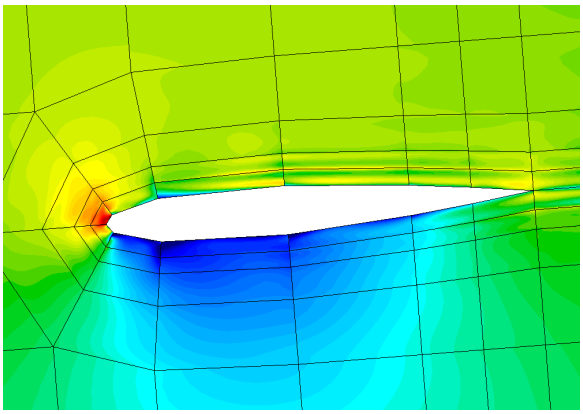
5.2.1 Inviscid flow

We first consider an inviscid flow configuration, the freestream Mach number M_∞ is set to 0.2, a pitch amplitude of 5° is considered, with a reduced frequency $k = \pi f c/U_\infty$ of 0.25, where c is the chord length and U_∞ the freestream flow velocity. The streamwise momentum

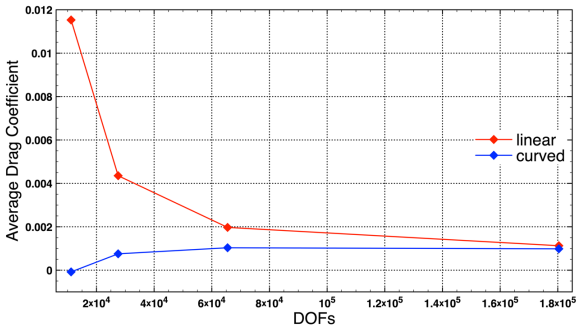
field ρu is illustrated in Fig. 14. The main physical feature is the slip line that develops from the trailing edge of the airfoil due to the pitching motion. Once the transitory effects have vanished, the solution converges to a periodic flow regime, it is hence possible to compute an average drag coefficient $\bar{C}_d$ and a peak lift coefficient $\hat{C}_l$.



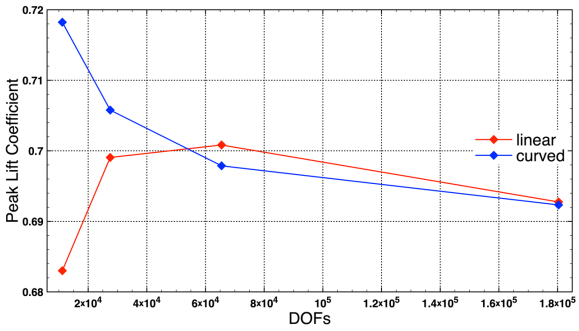
(a) Density field, curved mesh



(b) Density field, linear mesh



(c) Convergence of average drag coefficient



(d) Convergence of peak lift coefficient

Figure 15: Influence of the geometry, inviscid flow

In Fig. 15 we compare the results obtained using the high-order and piecewise linear grids. In particular, the density fields are presented in Fig. 15a and 15b for the coarsest mesh. The spurious effects introduced by the piecewise linear mesh can be easily noticed, non-physical expansion fans develop at each boundary vertex, generating further numerical oscillations. On the contrary, the high-order mesh guarantees a physically coherent solution even when the mesh is very coarse. The strong influence of the geometry description is caused by the flow-tangency boundary condition, which depends on the normal vector. For piecewise linear approximations, the normal vector is piecewise constant, therefore, across each element the boundary condition presents a discontinuity that creates the non-physical oscillations. This behaviour has already been evidenced in (7; 12). The observations are confirmed by the convergence study, illustrated in Fig. 15c and 15d for $\bar{C}_d$ and $\hat{C}_l$. The aerodynamic coefficients converge considerably faster when the curved boundary representation is adopted. Moreover, the error decreases monotonically for the high-order mesh.

5.2.2 Laminar flow

In the second part of the case study, we repeat the convergence analysis for a viscous fluid. The Reynolds number with respect to the chord length is equal to 1000. The freestream Mach number and the pitching frequency are unchanged, whereas the pitch amplitude is increased to 20^o, in order to generate a massively separated flow. Indeed, the considered setup leads to a dynamic stall of the airfoil with a periodic shedding of vortex pairs, as shown in Fig. 16, where the density field is represented. The convergence study is conducted using the same 4 meshes adopted for the inviscid test case.

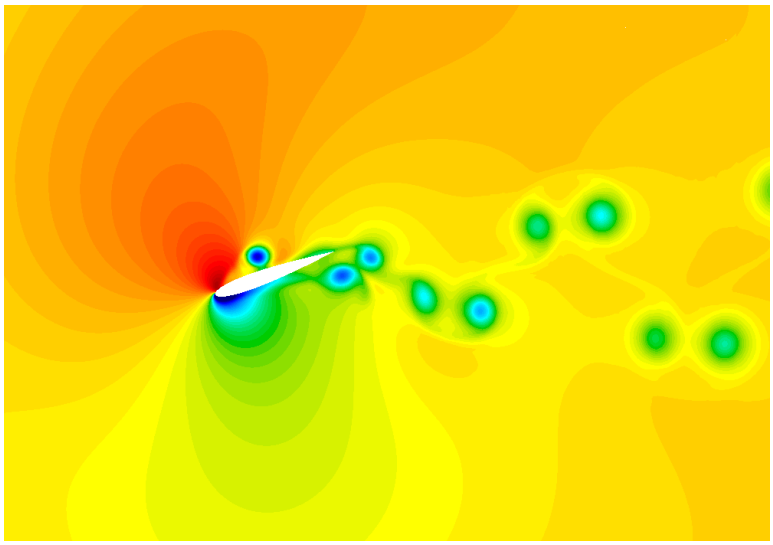
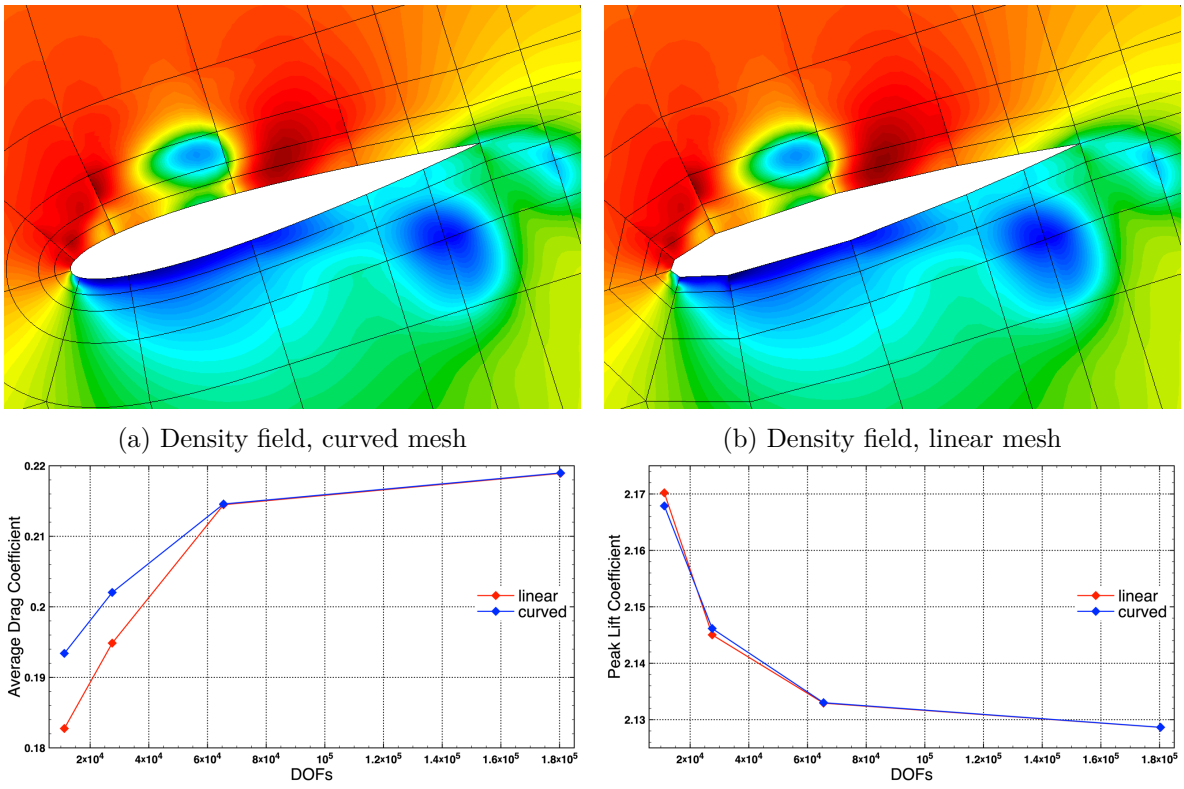


Figure 16: Pitching airfoil, laminar flow, density field

In Fig. 17a and 17b it is possible to compare the density fields computed with the high-order and the linearized mesh. Despite the low grid resolution, only very small discrepancies between the two solutions can be spotted, on the suction side of the airfoil close to the leading edge, but, overall, the two fields are very similar. With respect to the inviscid configuration, the high-order boundary representation seems to play a smaller role. This effect is caused by the different nature of the no-slip boundary condition, which is solely a function of the position. In contrast to the flow tangency condition used for Euler equations, the no-slip boundary condition is continuous across each element, therefore it introduces less numerical error. Furthermore, the physical viscosity regularizes the solution field, damping the artificial oscillations. It is thus possible that the influence of the boundary representation increases for higher Reynolds numbers, also due to the development of thin boundary layers and turbulent phenomena. However, such an investigastion is beyond the scope of the present paper.

The aerodynamic forces obtained with the linearized geometry are comparable to those computed using the curved boundary. Even for the coarsest mesh, the values of the peak lift coefficient are nearly identical, as reported in Fig. 17d. Regarding the convergence of the average drag coefficient, presented in Fig. 17c, the curved boundary leads to a slightly better results than its linearized counterpart for the first two meshes, but the advantage is less important with respect to the inviscid flow simulation.



(a) Density field, curved mesh

(b) Density field, linear mesh

(c) Convergence of average drag coefficient

(d) Convergence of peak lift coefficient

Figure 17: Influence of the geometry, laminar flow

6 Conclusion

In this work, we investigated the use of NURBS-based grids in the context of moving computational domains. Different possible approaches to account for moving meshes in a DG formulation were analysed, in the perspective of implementing a scheme for moving rational Bézier elements. The proposed formulation combines an isogeometric approach for DG with the ALE description, leading to a unified representation of the geometry, the grid velocity and the solution variables.

The accuracy of the method was verified on two problems with analytical solutions, exhibiting optimal convergence rates for rigidly moving or deforming grids. Although the discrete geometric conservation law cannot be preserved using rational bases, no significant impact on the results was reported. As expected, large grid deformations cause an increase of the error, due to the use of inexact numerical quadratures.

Finally, we considered two more demanding test-cases, involving inviscid and viscous flows around an oscillating cylinder and a pitching airfoil. The results confirmed the robustness of the proposed scheme in presence of high-order mesh deformations. Moreover, a faster convergence of the flow characteristics thanks to the use of curved grids was clearly established, in the context of moving bodies. It was also noted that inviscid flows are particularly sensitive to the lack of regularity in the geometry, due to the slip boundary condition at wall.

Several prospects can be drawn, starting from these results. In particular, one could envisage the extension to physics-based movements, like in fluid-structure interaction problems.

The possibility to define a unique high-order interface with different discretizations on each side would be appealing to account for the different characteristics of the fluid and structural models. Because of the capability to exactly represent circular shapes, the proposed approach would also be interesting to accurately compute fluxes at sliding interfaces, allowing the development of a fully conservative scheme for sliding grids.

Acknowledgements

The authors are grateful to the OPAL infrastructure from Université Côte d'Azur and Inria Sophia Antipolis - Méditerranée "NEF" computation platform for providing resources and support.

References

[1] 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.

[2] G. Farin, Curves and Surfaces for CAGD, 5th ed., Morgan Kaufmann, 2002.

[3] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order cfd methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (2013) 811–845.

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

[5] I. Lomtev, R. Kirby, G. Karniadakis, A discontinuous galerkin ale method for compressible viscous flows in moving domains, Journal of Computational Physics 155 (1999) 128 – 159.

[6] J.-B. Chapelier, M. de la Llave Plata, F. Renac, E. Lamballais, Evaluation of a high-order discontinuous galerkin method for the dns of turbulent flows, Computers & Fluids 95 (2014) 210 – 226.

[7] A. S. Silveira, R. C. Moura, A. F. C. Silva, M. A. Ortega, Higher-order surface treatment for discontinuous galerkin methods with applications to aerodynamics, International Journal for Numerical Methods in Fluids 79 (2015) 323–342.

[8] R. Costa, S. Clain, R. Loubère, G. J. Machado, High-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection-diffusion equation with dirichlet condition, Applied Mathematical Modelling 54 (2018).

[9] R. Sevilla, S. Fernandez-Mendez, A. Huerta, NURBS-enhanced finite element method for euler equations, Int. J. for Numerical Methods in Fluids 57 (2008).

[10] R. Abgrall, C. Dobrzynski, A. Froehly, An example of high order residual distribution scheme using non Lagrange elements : example of Bézier and NURBS., in: WCCM 2010 - the 9th World Congress on Computational Mechanics and 4th Asian Pacific Congress on Computational Mechanics, Sydney, Australia, 2010.

[11] 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.

[12] R. Duvigneau, Isogeometric analysis for compressible flows using a discontinuous galerkin method, Computer Methods in Applied Mechanics and Engineering 333 (2018) 443 – 461.

[13] R. Duvigneau, Cad-consistent adaptive refinement using a nurbs-based discontinuous galerkin method, Int. J. for Numerical Methods in Fluids (2020).

[14] V.-T. Nguyen, An arbitrary lagrangian-eulerian discontinuous galerkin method for simulations of flows over variable geometries, Journal of Fluids and Structures 26 (2010) 312, 329.

[15] P.-O. Persson, J. Bonet, J. Peraire, Discontinuous galerkin solution of the navier–stokes equations on deformable domains, Computer Methods in Applied Mechanics and Engineering 198 (2009) 1585 – 1595.

[16] F. Vilar, P.-H. Maire, R. Abgrall, A discontinuous galerkin discretization for solving the two-dimensional gas dynamics equations written under total lagrangian formulation on general unstructured grids, Journal of Computational Physics 276 (2014) 188 – 234.

[17] J. van der Vegt, H. van der Ven, Space–time discontinuous galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. general formulation, Journal of Computational Physics 182 (2002) 546 – 585.

[18] W. Boscheri, M. Dumbser, Arbitrary-lagrangian–eulerian discontinuous galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes, Journal of Computational Physics 346 (2017) 449 – 479.

[19] E. Gaburro, W. Boscheri, S. Chiocchetti, C. Klingenberg, V. Springel, M. Dumbser, High order direct arbitrary-lagrangian-eulerian schemes on moving voronoi meshes with topology changes, Journal of Computational Physics 407 (2020) 109 – 167.

[20] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian Methods, 2004.

[21] C. S. Venkatasubban, A new finite element formulation for ale (arbitrary lagrangian eulerian) compressible fluid mechanics, International Journal of Engineering Science 33 (1995) 1743 – 1762.

[22] H. Guillard, C. Farhat, On the significance of the geometric conservation law for flow computations on moving meshes, Computer Methods in Applied Mechanics and Engineering 190 (2000) 1467 – 1482.

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

[24] M. R. Visbal, D. V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, Journal of Computational Physics 181 (2002) 155 – 185.

[25] L. Piegl, W. Tiller, The NURBS Book, second ed., Springer-Verlag, New York, NY, USA, 1996.

[26] B. Cockburn, C.-W. Shu, The local discontinuous galerkin method for time-dependent convection-diffusion systems, SIAM Journal on Numerical Analysis 35 (1998) 2440–2463.

[27] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Review 43 (2001) 89, 112.

[28] R. Sevilla, S. Fernández-Méndez, A. Huerta, Comparison of high-order curved finite elements, International Journal for Numerical Methods in Engineering 87 (2011) 719–734.

[29] R. Löhner, C. Yang, Improved ale mesh velocities for moving bodies, Communications in Numerical Methods in Engineering 12 (1996) 599–608.

[30] A. Harten, P. D. Lax, B. van Leer, On upstream differencing and godunov-type schemes for hyperbolic conservation laws, SIAM Review 25 (1983) 35–61.

[31] C. Williamson, A. Roshko, Vortex formation in the wake of an oscillating cylinder, Journal of Fluids and Structures 2 (1988) 355 – 381.

[32] H. M. Blackburn, R. D. Henderson, A study of two-dimensional flow past an oscillating cylinder, Journal of Fluid Mechanics 385 (1999) 255–286.