Geometrically Consistent Aerodynamic Optimization using an Isogeometric Discontinuous Galerkin Method

Stefano Pezzano, Régis Duvigneau, Mickael Binois

Geometrically Consistent Aerodynamic Optimization using an Isogeometric Discontinuous Galerkin Method

Stefano Pezzano, Régis Duvigneau & Mickaël Binois

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

Abstract

The objective of the current work is to define a design optimization methodology in aerodynamics, in which all numerical components are based on a unique geometrical representation, consistent with Computer-Aided Design (CAD) standards. In particular, the design is parameterized by Non-Uniform Rational B-Splines (NURBS), the computational domain is automatically constructed using rational Bézier elements extracted from NURBS boundaries without any approximation and the resolution of the flow equations relies on an adaptive Discontinuous Galerkin (DG) method based on rational representations. A Bayesian framework is used to optimize NURBS control points, in a single- or multi-objective, constrained, global optimization framework. The resulting methodology is therefore fully CAD-consistent, high-order in space and time, includes local adaption and shock capturing capabilities, and exhibits high parallelization performance. The proposed methods are described in details and their properties are established. Finally, two design optimization problems are provided as illustrations: the shape optimization of an airfoil in transonic regime, for drag reduction with lift constraint, and the multi-objective optimization of the control law of a morphing airfoil in subsonic regime, regarding the time-averaged lift, the minimum instantaneous lift and the energy consumption.

1 Introduction

Modeling complex phenomena by solving systems of Partial Differential Equations (PDEs) has become a classical approach in several scientific disciplines, such as fluid and structural mechanics, electromagnetics, etc. This results from both the increase of the computational facilities and the maturity of numerical methods. As a consequence, it is now possible to simulate industrial products before manufacturing, at early design phase, to analyze in detail the physical behaviors. Simultaneously, the need for optimization algorithms has grown, in order to maximize the product performance, beyond engineers intuition, and minimize the environmental impact.

However, the use of fully automated design optimization loops is still difficult, in particular due to the complexity of the simulation pipeline. The simulation of realistic systems indeed requires mastering a large number of techniques and specific software: geometric modeling using Computer-Aided Design (CAD) tools, automated grid generation, PDE solvers for the different physical problems and couplings, post-processing and visualization, design optimization. Not only a deep knowledge is necessary to efficiently use each of these components, but also the coordination of the related software is a source of difficulties, due to a lack of

interoperability. The geometry plays a central role in this complexity overhead, because it is described by different representations, depending on the disciplinary context: CAD tools use high-order representations like Non-Uniform Rational B-Splines (NURBS) (1; 2; 3), whereas most PDE solvers are using grids, which are usually based on piecewise linear representations (triangles, tetrahedra, etc.). The coexistence of the two geometrical representations in the design loop yields a loss of accuracy, in particular in aerodynamics (4; 5; 6; 7), and an overhead complexity due to numerous geometrical transformations. This situation is even more tedious in a multidisciplinary context, where different solvers are coupled, and in design optimization because all the steps have to be automated (see (8) for a complete discussion on these issues).

A first breakthrough has been proposed in 2005 by T. Hughes and coauthors, namely IsoGeometric Analysis (IGA) (9), which consists in solving PDEs directly using parametric surfaces or volumes from CAD as computational domain, in a Finite-Element formulation. Hence, the proposed approach bypasses the approximation of the geometry by grids and unifies the representations used in CAD and analysis, yielding a significant simplification of the design pipeline and a gain of accuracy (10; 11; 12). However, several difficulties and limitations have arisen: i) Despite the developments achieved by the CAD community for 15 years, the capability of CAD software to generate ready-to-use computational domains is still very limited. ii) Multidimensional NURBS representations commonly used in CAD rely on tensorial products which make local refinement tedious, due to uncontrolled propagation. This has motivated the development of more sophisticated representations, like T-Splines (13), LR-Splines (14) or THB-Splines (15), which are however far more complex to handle. iii) The use of IGA for hyperbolic systems suffers from the need for stabilization, which is not straightforward in this context. As a consequence, applications in fluid mechanics for instance are rare (16; 17; 18; 19), especially for compressible flows (20).

To overcome the limitations of the IGA method, an extension of the isogeometric analysis paradigm to Discontinuous Galerkin (DG) methods has been proposed (21; 22), with application to non-linear hyperbolic systems. This requires a change of basis to generate discontinuities in the solution at element interfaces without altering the CAD geometry (for this, a CAD technique named Bézier extraction is used). The resulting approach demonstrates several interesting properties to improve the original formulation: conservativity, straightforward local refinement (23), CAD-consistency, easy extension to moving or deforming geometries (24).

The present work describes how this methodology can be employed to construct a design optimization loop in which the geometry is uniquely defined in terms of CAD representations, yielding gains in both accuracy and ease to deploy. We detail in section 2 the geometric modeling and in section 3 its extension to the construction of a CAD-consistent computational domain. The resolution of compressible Navier-Stokes equations is described in this context in section 4, including adaptive refinement of the mesh. Sections 5 and 6 are devoted to a brief description of the Bayesian optimization framework and the design loop. Finally two illustrations are presented, concerning the shape optimization of a transonic airfoil and the optimization of the control law of a morphing airfoil, in sections 7 and 8.

2 Geometry parameterization

Design procedures of most modern engineering systems are based on CAD kernels, which allow a straightforward communication with the manufacturing process. Several representations can be adopted, but the most commonly used approach relies on the description of the object

using parameterized boundaries (B-Rep approach). We will therefore adopt this point of view in the current work. However, for the sake of simplicity, we restrict here on two-dimensional problems in order to focus on the methodology. The extension to three-dimensional problems may not be straightforward, depending on the complexity of the selected case, and is out of the scope of this work. Nevertheless, we pay attention to propose methods that are not essentially restricted to two-dimensional cases.

2.1 NURBS Basis

We make the assumption that the geometry of the system of interest is described by a set of parameterized curves. Non-Uniform Rational B-Splines (NURBS) curves are now considered as standard in CAD (1; 2), since they allow to represent exactly a broad class of curves commonly encountered in engineering, like conic sections. Moreover, they permit a very intuitive definition and modification of geometrical objects through the handling of *control points*. Hence NURBS curves are chosen to describe the geometry. They are defined using a so-called *knot* vector $\Xi = (\xi_1, \ldots, \xi_l) \in \mathbb{R}^l$, which consists of l nondecreasing real numbers. This knot vector defines a discretization of the *parametric domain* $[\xi_1, \xi_l]$. NURBS basis functions are derived from B-Spline functions $(N_i^p)_{i=1,\cdots,n}$, which are polynomial and defined recursively as (2):

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

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

Note that the quotient $0/0$ is assumed to be zero. The degree of the functions p, the number of knots l and functions n are related by $l = n + p + 1$ (2). Open knot vectors, i.e. knot vectors with first and last knots of multiplicity $p + 1$, are usually used for representations of degree p to impose interpolation and tangency conditions at both extremities (2). Therefore $\xi_1 = \ldots = \xi_{p+1}$ and $\xi_{n+1} = \ldots = \xi_{n+p+1}$. A set of B-Spline basis functions of degree two is illustrated in Fig. 1.

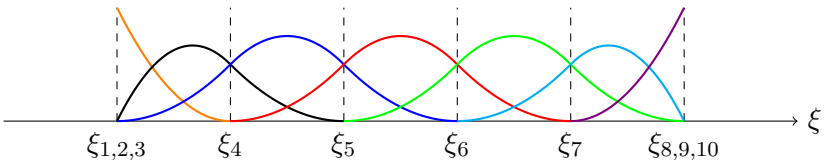


Figure 1: Seven quadratic B-Spline basis functions.

Then, the NURBS basis functions $(R_i^p)_{i=1,\cdots,n}$ are defined by associating a set of weights $(\omega_i)_{i=1,\cdots,n}$ to the B-Spline functions according to:

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

Finally, a NURBS curve of coordinates $\mathbf{x}(\xi) = (x(\xi), y(\xi))$ is obtained by positioning a

set of *control points* which are associated to the NURBS basis functions:

$$\mathbf{x}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{x}_i, \tag{4}$$

where $\mathbf{X} = (\mathbf{x}_i)_{i=1,\cdots,n}$ are the coordinates of the control points in the *physical domain*. Note that the previous equation defines a mapping $\mathcal{F}$ from the parametric domain $\widehat{\Omega}$ to the physical one Ω, as illustrated in Fig. 2

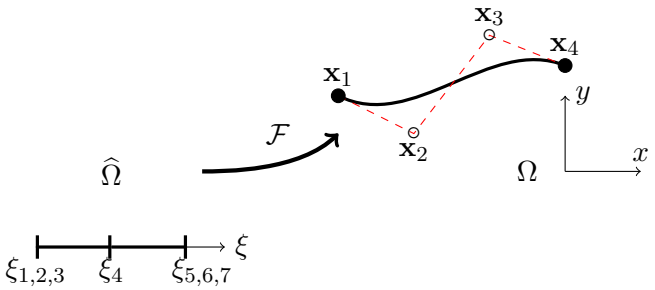


Figure 2: Example of quadratic NURBS curve with four control points.

2.2 Some properties

Projection A first important property concerns the relationship between B-Spline and NURBS curves: it can be shown (3) that any NURBS curve can be considered as the projection of a B-Spline curve defined in a space of higher dimension using its weights. For instance, a NURBS curve lying in the plane (x, y), characterized by a set of control points $\mathbf{X} = (\mathbf{x}_i)_{i=1,\cdots,n} = (x_i, y_i)_{i=1,\cdots,n}$ and weights $(\omega_i)_{i=1,\cdots,n}$, can be described as the projection of a B-Spline curve lying in a 3D space and defined by the control points $(x_i\omega_i, y_i\omega_i, \omega_i)_{i=1,\cdots,n}$. This representation of NURBS curves as B-Spline ones is useful for practical manipulations because one can apply transformations directly to B-Spline curves and project the result, instead of developing procedures specific to NURBS.

Knot insertion A second property of NURBS representations is the capability to insert a new knot, and thus a new basis function, without altering the geometrical object (3). This procedure can be considered as a local h-refinement procedure (12) and is referred as *knot insertion*.

Indeed, any NURBS curve with the associated knot vector $\Xi = (\xi_1, \ldots, \xi_l) \in \mathbb{R}^l$ can be identically represented using the knot vector $(\xi_1, \ldots, \xi_q, \bar{\xi}, \xi_{q+1} \ldots, \xi_l) \in \mathbb{R}^{l+1}$, that includes the additional knot $\bar{\xi}$ inserted between ξ_q and ξ_{q+1}. The new curve is then defined by $n+1$ control points. To compute their new locations, one can use the projection property described above and simply apply the procedure to the corresponding B-Spline curve defined in the space of higher dimension. With the additional knot, the B-Spline curve can be written as (3):

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

with the new set of $n+1$ control points defined as:

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

The figure 3 illustrates the knot insertion procedure applied to the curve of the previous figure.

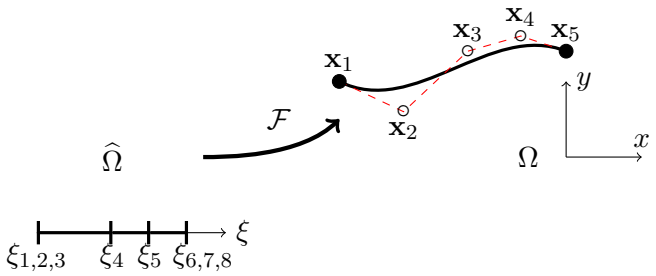


Figure 3: Example curve of figure 2 with a new inserted knot.

Bézier extraction It is important to underline now a particular case: if a new knot is inserted at an existing knot location, the regularity of the curve is decreased. More generally, the curve at the knot ξ_q has the regularity C^{p-r}, where r is the multiplicity of the knot q (3). Therefore, if one inserts p knots at an existing knot location, the geometry of the curve is preserved but the curve is now divided in two independent parts. If this multiple-knot insertion is achieved for all inner knots, the curve is decomposed into a set of independent *Bézier curves*, each of them composed of $p+1$ control points (3), as depicted in figure 4.

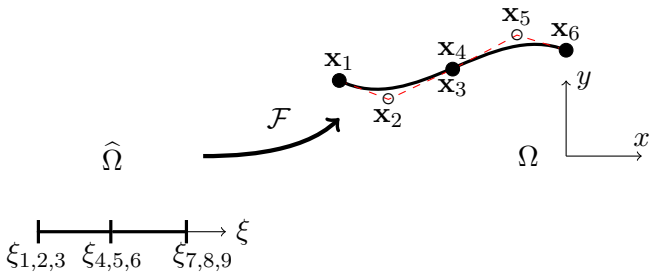


Figure 4: Example curve of figure 2 after Bézier extraction procedure.

2.3 Surfaces and volumes

All the previous concepts and properties can be extended to surfaces (respectively volumes) by using bivariate (respectively trivariate) tensor products. In particular, a NURBS surface of degree p is defined as:

$$\mathbf{x}(\xi,\eta) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} R_i^p(\xi) R_j^p(\eta)\mathbf{x}_{ij}, \tag{7}$$

where $(\mathbf{x}_{ij})_{i=1,\cdots,n}$ are the coordinates of the control point indexed ij in the lattice. Such a surface defined in the plane is shown in figure 5. This tensorial construction is especially important because it will be the baseline of the CAD-consistent computational domain.

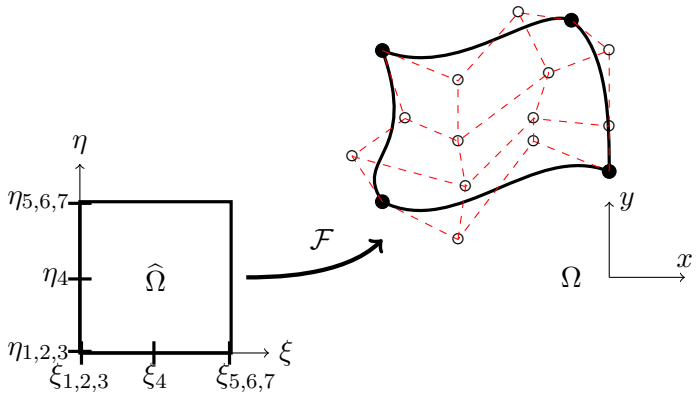


Figure 5: Example of quadratic NURBS surface with four control points.

3 Computational domain construction

3.1 Principles

For each geometry explored during the design optimization phase, a corresponding computational domain must be constructed to allow the physical analysis using a dedicated PDE solver. According to the proposed paradigm, we aim at defining a mesh which is consistent with the CAD geometry, yielding a geometrically exact analysis. In this perspective, we have to satisfy the following constraints:

1. The mesh should cope with the solver requirements;
2. The mesh boundary should exactly match the NURBS curves that define the geometry;
3. The mesh should remain valid for large variations of the geometry;
4. The mesh generation process should be completely automated.

Let us consider first the question of the solver requirements. Indeed, our objective of geometrically exact analyses yields a strong dependency between the mesh generation task and the PDE resolution, contrary to more classical approaches. As explained in previous works (22), we do not follow the original isogeometric paradigm proposed by T. Hughes and coauthors (12), which consists in defining the mesh as a set of NURBS patches and apply a Continuous Galerkin (CG) formulation, for two main reasons. Firstly, CG schemes are not so well suited to aerodynamic applications due to the necessity to adjust a set of parameters to stabilize the hyperbolic terms and avoid oscillations in the vicinity of shocks. Secondly, local mesh refinement requires the use of sophisticated and somehow cumbersome representations like T-Splines (13) to avoid a large propagation of the refinement, due to the tensorial nature of NURBS patches. To overcome these difficulties, a Discontinuous Galerkin (DG) formulation based on rational Bézier representations is preferred (22). Indeed, the DG formulation

is very well adapted to the resolution of hyperbolic systems of conservation laws and the use of a discontinuous representation makes refinement essentially local.

However, this choice makes the construction of the mesh a bit more complicated. Indeed, the proposed approach necessitates to generate a mesh composed of rational Bézier elements that coincide with the NURBS boundary. Fortunately, this can be achieved quite easily, thanks to the Bézier extraction property described in the previous section. This allows to split the boundary NURBS curves into a set of rational Bézier curves that will compose the boundary edges of the elements.

To make the procedure fully automated and allows large deformations of the boundary, we proceed in three steps for the construction of the grid: first, a small set of fixed rational Bézier elements are defined inside the computational domain, but far from its deforming boundary. Then, an extremely coarse grid is generated by connecting the boundary edges to surrounding fixed points, yielding a *primary mesh* composed of a few rational Bézier elements. The edges define the control points and weights at the border of the elements. The interior ones are initialized using a discrete Coons patch construction(1). Note that the fixed part of this mesh is far enough from the deforming boundary to permit large geometry modifications. Even if this mesh is not suitable for analysis due to its coarseness, it is anyway consistent with the CAD definition of the geometry. Finally, some local refinement steps are achieved by using multiple knot insertions to obtain a *secondary mesh*, which will be considered as the initial grid for an adaptive analysis. The properties of the refinement process maintain both the boundary geometry and the viability of the elements (e.g. positive Jacobian).

3.2 Ilustration

The proposed approach is illustrated for the specific case of an airfoil. This example is very simple on purpose, to put in light the methodological steps. The airfoil is parameterized by two NURBS curves of degree $p = 3$, one for the suction side and one for the pressure side, composed of $n = 10$ control points each. The knot vector of length $k = 14$ is defined with a uniform distribution for the two curves $\Xi = (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 7, 7, 7)$. Four repeated knots at start and end ensure that the curve interpolates the control points at the two extremities. The positions of the control points are provided in table 1. Note that the alignment of control points at the leading edge ensures a C^1 continuity. For the sake of simplicity, unitary weights are employed. This defines completely the two curves illustrated in figure 6.

index	1	2	3	4	5	6	7	8	9	10
x	0.	0.	0.125	0.25	0.375	0.5	0.626	0.75	0.875	1.
y^+	0.	0.031	0.068	0.074	0.065	0.048	0.034	0.039	0.025	0.
y^-	0.	-0.027	-0.031	-0.044	-0.069	-0.061	-0.039	-0.013	-0.010	0.

Table 1: Airfoil case: NURBS control point coordinates.

Then, we apply the Bézier extraction procedure which allows to split the curves into a set of rational Bézier edges without modifying the geometry, as illustrated in figure 7. The filled markers represent the extremities of the rational Bézier edges, where the control points are interpolated.

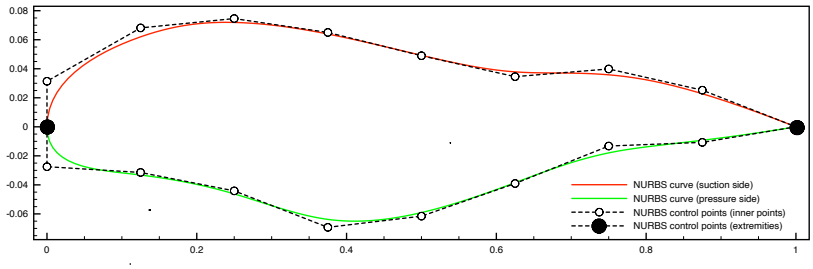


Figure 6: Airfoil geometry defined by two NURBS curves.

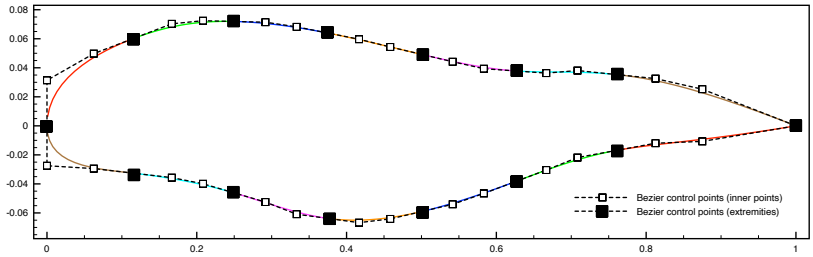


Figure 7: Airfoil geometry defined by a set of rational Bézier edges, after Bézier extraction procedure.

The first step of the mesh construction consists in defining a set of rational elements surrounding the airfoil, independently from its shape, as depicted in figure 8. This task is achieved manually, given the small number of elements. Note that the tiling is chosen to coincide with the number of edges on the airfoil boundary. The gap between the airfoil and these fixed elements is obviously large enough to define very different airfoil shapes.

During the second step, the extremities of the Bézier edges on the airfoil are connected to the fixed elements, filling the computational domain and yielding the so-called primary mesh, as shown in figure 9. Although extremely coarse, this mesh is CAD-consistent.

The third step consists in refining the grid in the vicinity of the airfoil to obtain a mesh suitable for analysis. This can be done automatically by applying the knot insertion algorithm. The resulting grid is denoted as secondary mesh shown in figure 10. Note that this mesh is still coarse because we intend to employ an Adaptive Mesh Refinement (AMR) strategy during the analysis and, therefore, it is only an initial grid. Again, this task is achieved without altering the geometry with respect to the baseline definition of the airfoil using two NURBS curves.

4 Resolution of flow equations

We present in this section the methods employed to solve flow equations using rational Bézier elements, yielding geometrically exact analyses. Most of these methods have already been described in previous works (22; 23; 24), therefore we focus here on their integration in the design optimization loop.

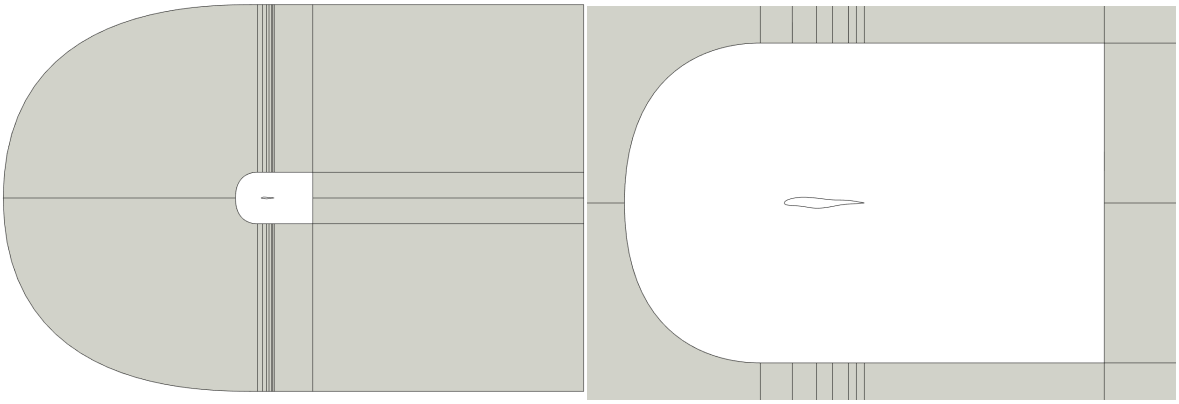


Figure 8: Set of fixed rational Bézier elements surrounding the airfoil (left: full domain, right: zoom).

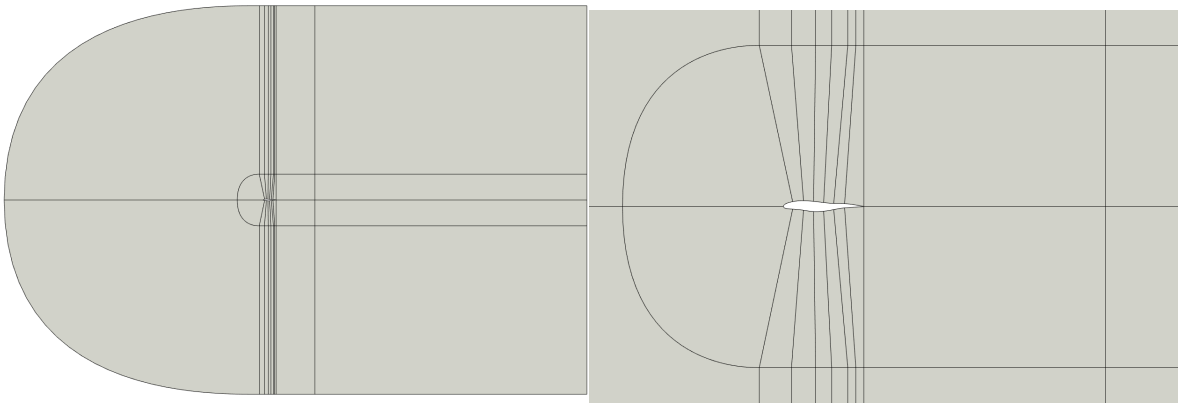


Figure 9: Primary mesh after connections (left: full domain, right: zoom).

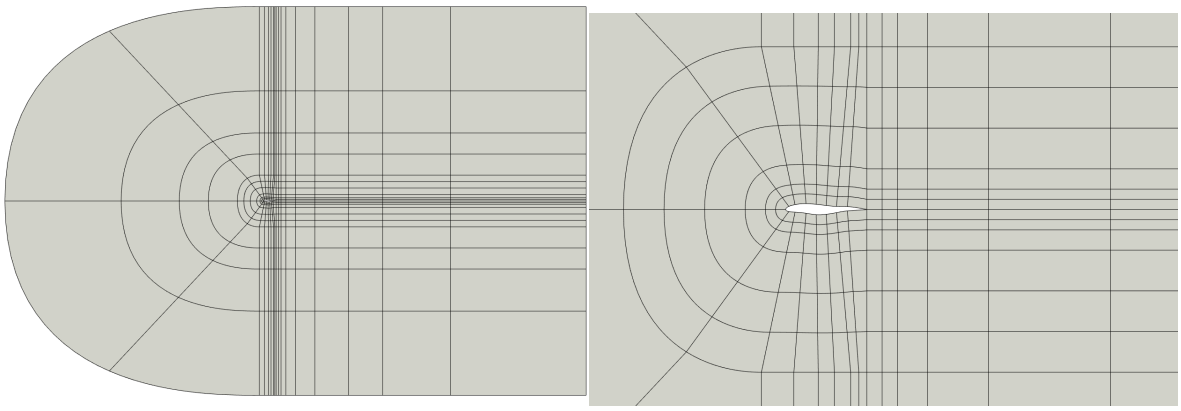


Figure 10: Secondary mesh after refinements (left: full domain, right: zoom).

4.1 Isogeometric Discontinous Galerkin method

As explained above, we adopt in this work a Discontinuous Galerkin formulation based on rational Bézier elements, which permits both to use CAD-consistent grids and offers a suitable framework for the resolution of hyperbolic systems. Due to the applications targeted, we consider as state equations the compressible Euler / Navier-Stokes equations. Moreover, an Arbitrary Lagrangian-Eulerian (ALE) form is adopted to be as general as possible and to allow to solve problems including time-dependent geometries. In this context, the physical flux can be expressed as:

$$\mathbf{F} = \mathbf{F}_c(\mathbf{W}) - \mathbf{F}_v(\mathbf{W}, \nabla\mathbf{W}), \tag{8}$$

where $\mathbf{F}_c$ is the convective flux, $\mathbf{F}_v$ the viscous flux, $\mathbf{W}$ the conservative variables:

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

where ρ denotes the density, p the pressure, e the total energy and u_i the velocity vector. τ_{ij} is the viscous stress tensor and q_i is the thermal conduction flux, defined as:

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

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

where $\gamma = 1.4$, $Pr = 0.72$ and μ is determined by the Reynolds number. When Euler equations are considered, the viscous flux $\mathbf{F}_v$ vanishes. The second order derivatives are discretized with the Local Discontinuous Galerkin (LDG) approach (25). We thus write the state equations in ALE form as a system of first order equations:

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

where $\mathbf{G}$ is the gradient of the conservative variables and $\mathbf{V}_g$ the velocity of the moving domain. According to the DG formulation, these equations are multiplied by a rational Bézier trial function R_k (for the sake of clarity we drop degree p) and integrated by part on each element Ω_j. Using the map defined by the rational Bézier functions, the integrals are transposed from the physical space to the parametric domain $\widehat{\Omega}$, yielding the following *isogeometric ALE-DG formulation* (24):

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{w}_i \displaystyle\int_{\widehat{\Omega}} R_k R_i \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega}\right) = \displaystyle\int_{\widehat{\Omega}} \nabla R_k \cdot \left(\mathbf{F}_c - \mathbf{F}_v - \mathbf{V}_g \mathbf{w}_h\right) \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} \\ \qquad\qquad - \displaystyle\oint_{\partial\widehat{\Omega}} R_k (\mathbf{F}^*_{ale} - \mathbf{F}^*_v) \left|\mathrm{J}_{\Gamma_j}\right| \mathrm{d}\widehat{\Gamma}, & \text{(13a)} \\ \mathbf{g}_i \displaystyle\int_{\widehat{\Omega}} R_k R_i \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} = \displaystyle\int_{\widehat{\Omega}} \nabla R_k \mathbf{w}_h \left|\mathrm{J}_{\Omega_j}\right| \mathrm{d}\widehat{\Omega} - \displaystyle\oint_{\partial\widehat{\Omega}} R_k \mathbf{W}^* \left|\mathrm{J}_{\Gamma_j}\right| \mathrm{d}\widehat{\Gamma}. & \text{(13b)} \end{cases}$$

This discretization is qualified as isogeometric because the same rational Bézier representation 7 is employed for the geometry $\mathbf{x}$, the local solution fields $\mathbf{w}_h$ and $\mathbf{g}_h$, as well as the domain velocity $\mathbf{V}_g$:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{w}_h \\ \mathbf{g}_h \\ \mathbf{V}_g \end{pmatrix} = \sum_{i=1}^{(p+1)^2} R_i(\xi,\eta) \begin{pmatrix} \mathbf{x}_i \\ \mathbf{w}_i \\ \mathbf{g}_i \\ \mathbf{v}_{g,i} \end{pmatrix}, \tag{14}$$

The convective flux function $\mathbf{F}^*_{ale} = \mathbf{F}^* - \mathbf{V}_g \mathbf{w}^*$ is evaluated using the HLL Riemann solver (26), modified to account for the time-dependent domain (24), whereas the diffusive flux functions $(\mathbf{F}^*_v, \mathbf{W}^*)$ are computed using the LDG approach (25). Moreover, integrals are evaluated by numerical quadrature based on Gauss-Legrendre rules and time integration is carried out using standard Runge-Kutta methods. The resulting method has been verified on a set of analytical problems, exhibiting optimal $p+1$ convergence rate (22; 23), and validated on classical benchmarks (24). To complete this description, we mention the use of the subcell shock capturing method (27) adapted to the rational Bézier representation (22; 23). As summary, we employ very classical techniques to solve the flow equations, except that they are based on rational Bézier representations intead of standard nodal Lagrange functions, to cope with CAD-consistent grids.

4.2 Adaptive mesh refinement

The use of Adaptive Mesh Refinement (AMR) techniques is a critical ingredient of the design procedure because it minimizes the computational time required for each analysis and makes the automated grid generation process reliable. Indeed, constructing *a priori* a mesh adapted to the flow features is far from being straightforward, especially when the process has to be fully automated. Therefore, it is far more efficient and robust to construct first a baseline grid, denoted as secondary mesh in section 3.1, and then use AMR techniques at runtime to obtain an appropriate resolution for each region of the computational domain. We underline that the proposed isogeometric DG approach is a perfect framework in this perspective because the geometry is essentially preserved during the isogeometric refinement and the DG formulation allows a natural handling of non conformities.

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 specific to DG discretization is introduced to decides which elements should be refined, based on the measure of interface jumps. 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:

$$\epsilon_j = \sum_{k \in \mathcal{N}_j} \int_{\Gamma_{jk}} \| \mathbf{W}|_{\Omega_j} - \mathbf{W}|_{\Omega_k} \| \, d\Gamma, \tag{15}$$

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, respectively coarsening, if the indicator ϵ_j

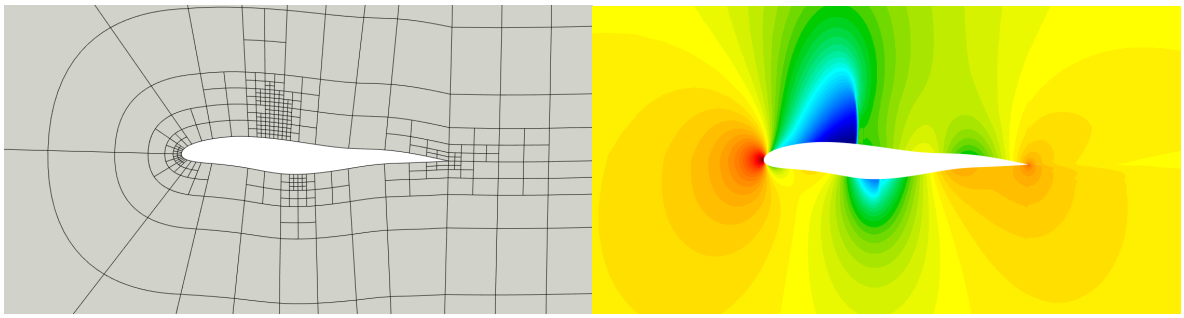


Figure 11: Adapted Mesh Refinement (left: adapted mesh, right: density field).

exceeds a used-defined criterion ϵ_{ref}, respectively ϵ_{coa}. 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 conservative least-squares projection. The whole adaptation procedure is detailed in (23).

We illustrate the proposed AMR approach for the case of an airfoil in transonic regime. The geometry and the corresponding secondary mesh are the ones described in section 3.1. The flow is governed by Euler equations, with a free-stream Mach number $M_\infty = 0.7$ and an incidence $\alpha = 1°$. Cubic elements are used and three refinement levels are allowed starting from the secondary mesh. As can be seen in figure 11, the adaption procedure targets areas in the vicinity of shocks as well as leading and trailing edges, while maintaining a coarse grid where refinement is not required. As initially targeted, the whole evaluation procedure, starting from the NURBS-based geometry construction to the adapted flow simulation, is fully automated and CAD-consistent.

5 Bayesian optimization

The last component of the design loop is the optimization algorithm, that has to provide a new tentative of design parameters at each optimization step according to the knowledge acquired. We aim at solving complex problems, based on different physical behaviors, possibly characterized by the presence of multiple local minima, active constraints and including several optimization criteria. As a consequence, we choose to use a Bayesian Optimization (BO) approach as a single- or multi-objective, constrained, global optimization framework (28).

The BO methods have been popularized by Jones (29) for single-objective problems, and then extended to multi-objective ones (30). They rely on the construction of Gaussian Processes (31) (GPs), also denoted Kriging models, as surrogates for the objective functions and constraints. These models are initialized on the basis of values obtained during a Design of Experiment (DoE) exploring phase (32), and then iteratively enriched with new values during the optimization phase. In this latter stage, the new points to be evaluated are determined via a merit function, that balances the optimization of the surrogate model and the improvement of its accuracy. The most commonly used merit function for single-objective problems is the *Expected Improvement* (33) (EI), that is computed from the expectation and variance of the GPs. For multi-objective problems, the *Expected Hypervolume Improvement* (34) (EHI) generalizes the approach to an arbitrary number of criteria. Constraints are taken into account by introducing the *Expected Feasible Improvement* (35) (EFI) criterion.

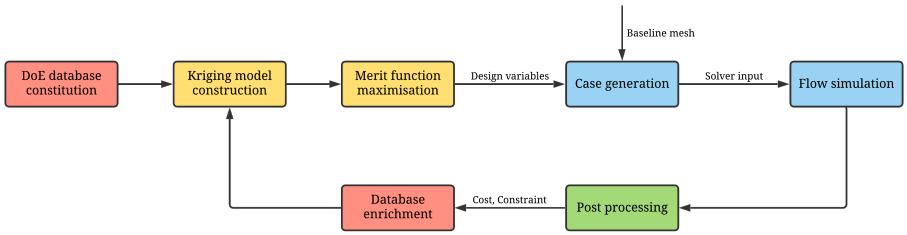


Figure 12: Design optimization loop.

The above-mentionned methods are not detailled, provided that this topic is not the core of the present work. However, interested readers are encouraged to consult the proposed references. Practically, the algorithms used are implemented in a set of open-source R packages (36; 37).

6 Overview of the design loop

The assembly of the different methods described in the previous sections yields the design optimization loop depicted in figure 12. From a mathematical point of view, we seek to solve the following problem:

$$\begin{aligned} \min_{\mathbf{z}\in\mathcal{D}} \quad & g(\mathbf{z}), \\ \text{subject to} \quad & h(\mathbf{z}) \leq 0, \end{aligned} \tag{16}$$

where g is the cost function and h is the constraint function. The design variables $\mathbf{z}$ constitute a d-dimensional vector contained in the design space $\mathcal{D} \subset \mathbb{R}^d$. Typically, the components of $\mathbf{z}$ represent the coordinates of the NURBS control points defining the geometry. Introducing the dependency of the cost function with respect to the discrete flow solution $\mathbf{w}_h$ and the parametric representation of the geometry γ, the cost function writes:

$$g(\mathbf{z}) = \mathcal{G}\left(\gamma(\mathbf{z}), \mathbf{w}_h(\gamma(\mathbf{z}))\right). \tag{17}$$

Obviously, the same formulation holds for the constraint function. This formula underlines the main advantage of the proposed approach: the geometry $\gamma(\mathbf{z})$ used during the resolution of the state equations and the evaluation of the cost and constraint functions is not discretized (and approximated), contrary to methods based on classical piecewise-linear grids for which the computations rely on an approximated geometry $\gamma_h(\mathbf{z})$. This results in an increased accuracy, as demonstrated in previous works (22; 24). Moreover, this simplifies the software coupling, thanks to the absence of a real mesh generation / deformation algorithm.

These advantages are highlighted in the following sections, which deal with the application of the proposed methodology to the optimization of an airfoil shape and the determination of the optimal deformation of a morphing airfoil.

7 Application to airfoil shape optimization

We consider the optimization of the shape of an airfoil in transonic regime. The shape is defined using two NURBS curves of degree $p = 3$ and $n = 10$ control points, whose

index	2	3	4	5	6	7	8	9
y^+_{max}	0.040	0.070	0.080	0.080	0.080	0.070	0.050	0.040
y^+_{min}	0.030	0.040	0.040	0.040	0.040	0.030	0.020	0.010
y^+_{best}	0.032	0.050	0.057	0.064	0.063	0.059	0.042	0.018
y^-_{max}	-0.020	-0.030	-0.040	-0.040	-0.040	-0.020	-0.000	-0.000
y^-_{min}	-0.030	-0.050	-0.060	-0.070	-0.070	-0.050	-0.030	-0.020
y^-_{best}	-0.020	-0.050	-0.052	-0.049	-0.060	-0.040	-0.015	-0.007

Table 2: Airfoil optimization: bound constraints and optimal configuration.

characteristics have been presented in section 3.2. The first and last control points are fixed during the optimization to maintain the leading and trailing edges. Inner control points are allowed to move vertically, yielding a vector of design parameters $\mathbf{z}$ of dimension $d = 16$. The automated construction of the grid according to each design parameters set follows the steps described in section 3.1 and illustrated in 3.2.

The flow is modeled by compressible Euler equations and is characterized by a free-stream Mach number $M_\infty = 0.7$ and an incidence $\alpha = 1°$. The adaptive DG method presented above is used to solve the state equations until a steady state is reached. The objective of the optimization problem is the minimization of the drag coefficient $g(\mathbf{z}) = C_d$ subject to a constraint on the lift coefficient $h(\mathbf{z}) = C_l^0 - C_l \leq 0$ with $C_l^0 = 0.3644$.

The assessment of the flow computation is more tedious in the context of design optimization than for a single flow analysis. Indeed, a mesh that is satisfactory for a certain configuration may not be suitable for others which have different flow characteristics. This difficulty is alleviated by the use of AMR techniques. Nevertheless, one should select the maximum number of refinement levels allowed by the automated procedure. To assess as much as possible this choice, the DoE phase is achieved using different values, ranging from one to four refinement levels, then the lift and drag values obtained with the different levels are compared. Since the DoE explores the whole design space, this gives a more robust assessment than just considering a single starting configuration. The size of the DoE is set to $N_{DoE} = 32$. Therefore, 32 airfoil geometries are generated randomly, according to a Latin Hypercube sampling (32), in the domain detailed in table 2. The figure 13 depicts the lift and drag values obtained with the different refinement levels. As can be observed, the results corresponding to one refinement level are clearly far from those obtained with more refined grids. The values obtained with three and four levels are almost indistinguishable, therefore the optimization phase is achieved using three refinement levels.

Some configurations are extracted from the DoE and shown in figure 14, for three refinement levels. As can be seen, the shape modifications strongly impact the flow characteristics, in particular the number, the intensity and the location of the shocks. Therefore, the use of AMR techniques is critical here to define a suitable grid whatever the airfoil shape, while maintaining a low computational cost.

The evolution of the cost (drag) and constraint (lift) functions during the whole design optimization procedure can be seen in figure 15. As observed, the lift and drag values are quite well learnt after the DoE phase, yielding a fast convergence towards interesting areas. One can notice that most trials in the optimization phase verify the lift constraint, whereas only minor improvements of the drag are reported after design 50. The design parameters

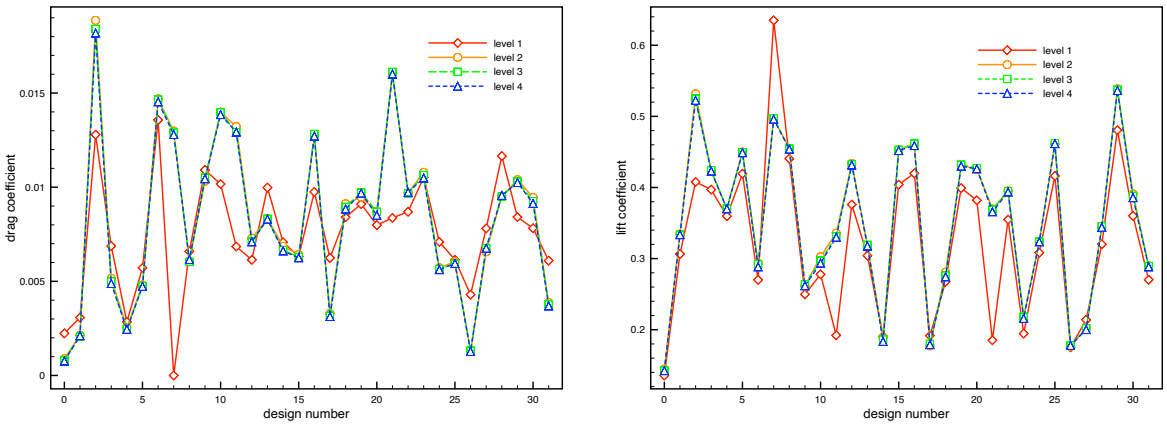


Figure 13: Drag and lift values in the DoE phase, for different refinement levels.

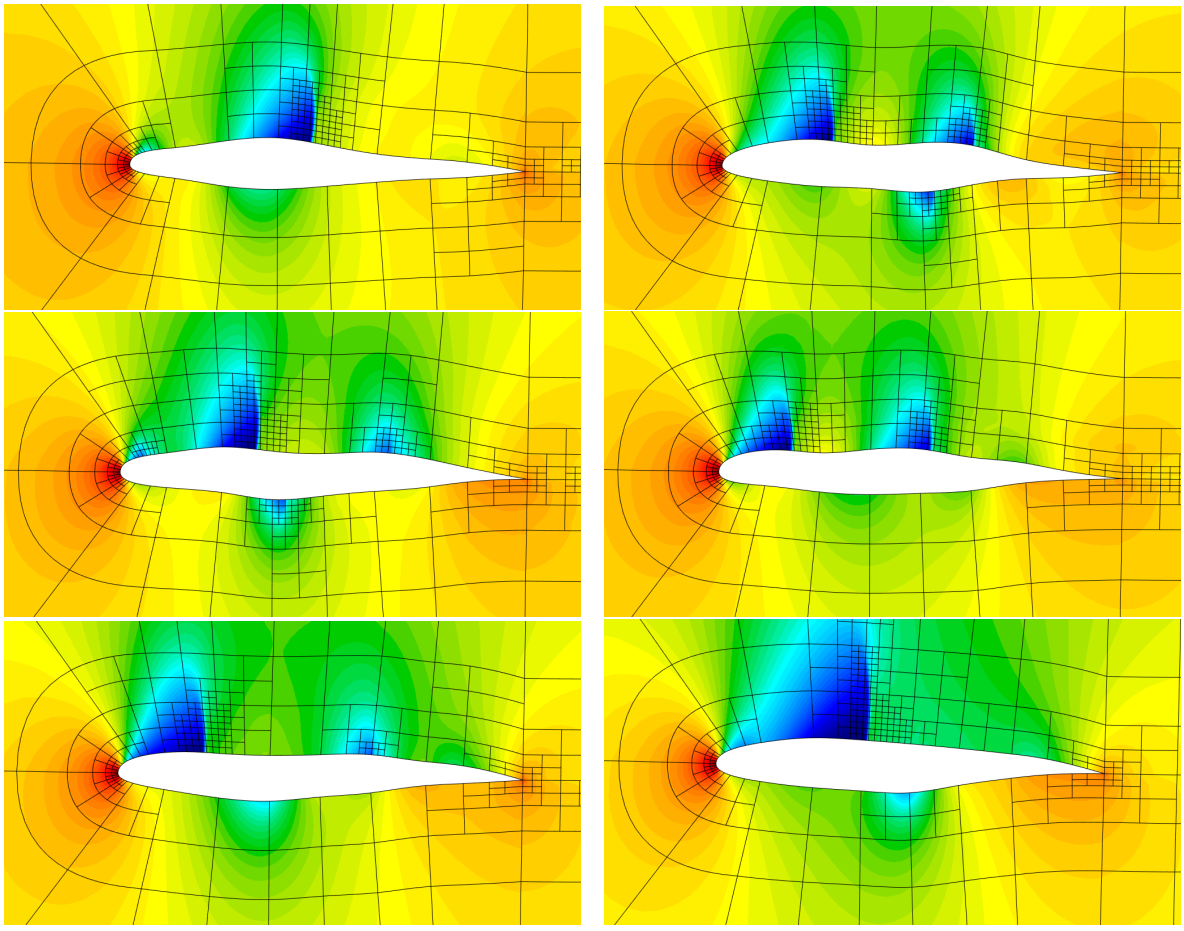


Figure 14: Some configurations from the DoE phase (adapted meshes and density fields).

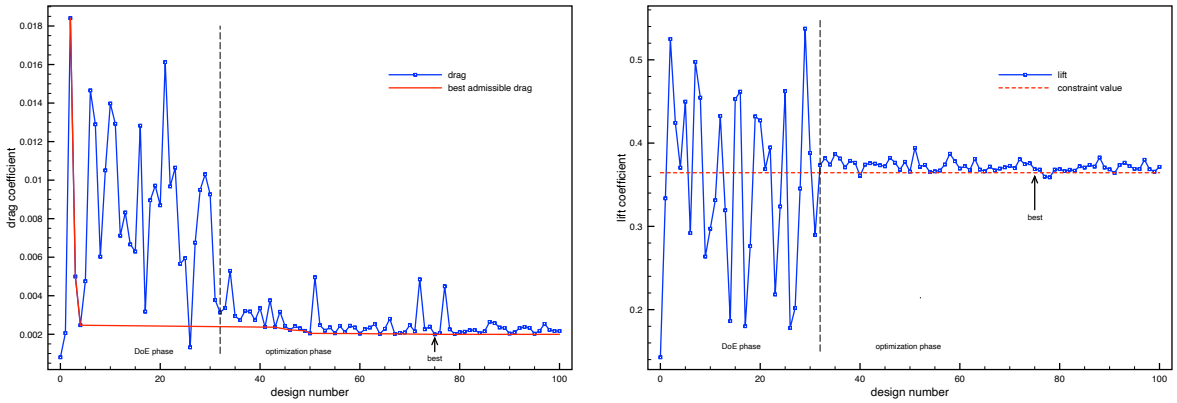


Figure 15: Drag and lift values during the design optimization procedure.

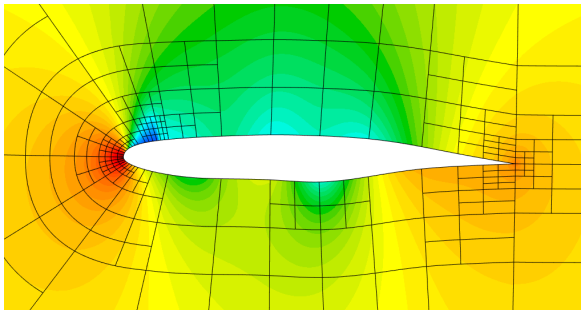


Figure 16: Optimal shape (adapted mesh and density field).

defining the best shape, corresponding to the design 75, are provided in table 2. As seen, two control points have reached their bounds, at the suction side. Figure 16 shows the density field (same colormap as figure 14), which is characterized by a quasi-absence of shock on the pressure side. As expected, the drag minimization results in a shock reduction.

8 Application to airfoil morphing optimization

As second application, we consider the optimization of a morphing airfoil, i.e. an airfoil with a periodically moving trailing edge. Therefore, contrary to the previous case, we do not aim at optimizing the shape but the shape movement. The morphing concept takes inspiration from the birds' feathers, thanks to which birds can control the geometry of their wings. Thanks to morphing, it is possible to obtain optimal aerodynamic performance for multiple flight conditions. For this reason, several morphing prototypes have been studied over the years (38). A multi-point optimisation study of a morphing airfoil has been carried out by Fincham et al. (39). We investigate here a trailing edge morphing technique to control the boundary layer separation in the laminar regime. The aim of the study is to demonstrate the potential of using the isogeometric framework for flow problems with deforming geometries.

8.1 Case definition

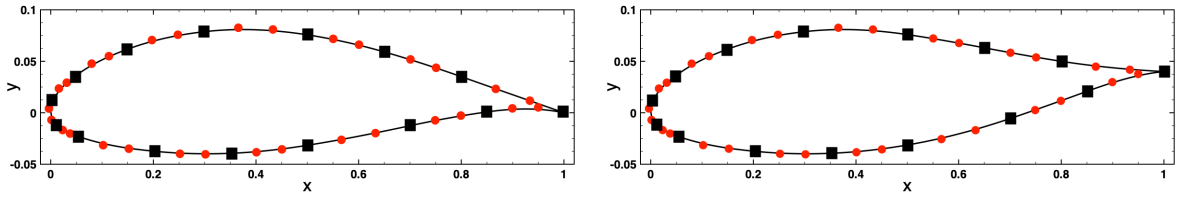
We consider a NACA $63_1 - 412$ airfoil, as baseline geometry. The NACA 6-series of airfoils is widely employed for aircraft wings and wind turbine blades and it was originally designed with the aim of maximizing the region of laminar flow around the airfoil. Using a least-square fitting, we obtain a description of the NACA $63_1 - 412$ airfoil as a single NURBS curve. We adopt a knot vector defined by 15 non-zero knot spans and a cubic representation. The multiplicity of each internal knot vector is equal to 2, meaning that the obtained NURBS curves are C^1. Similarly to the work of Simiriotis et al. (40), we adopt a quadratic morphing law in space and a sine law in time. Thus, the movement of each control point of index i of the NURBS boundaries is given by:

$$\begin{cases} x_i(t) = x_{i,0} + a_{x,i} \sin\left(2\pi f t\right), \\ x_i(t) = x_{i,0} + a_{y,i} \sin\left(2\pi f t\right), \end{cases} \tag{18}$$

where $\mathbf{x}_{i,0} = (x_{i,0}, y_{i,0})$ are the control points of the original NURBS curve, f is the frequency, $a_{x,i} = 0$ and $a_{y,i}$ is determined by:

$$a_{y,i} = \begin{cases} 0, & \text{if } x_{i,0} < x_0 \\ A\left(\dfrac{x_{i,0} - x_0}{1 - x_0}\right)^2, & \text{if } x_{i,0} \geq x_0 \end{cases} \tag{19}$$

where A is the amplitude of the oscillation of the trailing edge and $x_0 = 1 - L$, with L being the controlled fraction of the airfoil surface. Using the proposed deformation law, the morphing is controlled by just 3 parameters: L, A and f. The Bézier extraction is then employed to obtain a set of rational Bézier edges from the NURBS curve at any time. The Bézier airfoil computed with this approach is reported in figure 17a for the undeformed configuration and figure 17b for a trailing edge displacement equal to a third of the thickness, corresponding to the parameters $L = 50\%$, $A = 4\%$. The red rounded markers correspond to the inner control points of the Bézier edges, whereas the black squared ones correspond to the extremities control points. We remark that, since the extraction operator preserves the geometry, the curves described by the rational Bézier edges are $\mathcal{C}^1$ at all times, as the original NURBS curve.

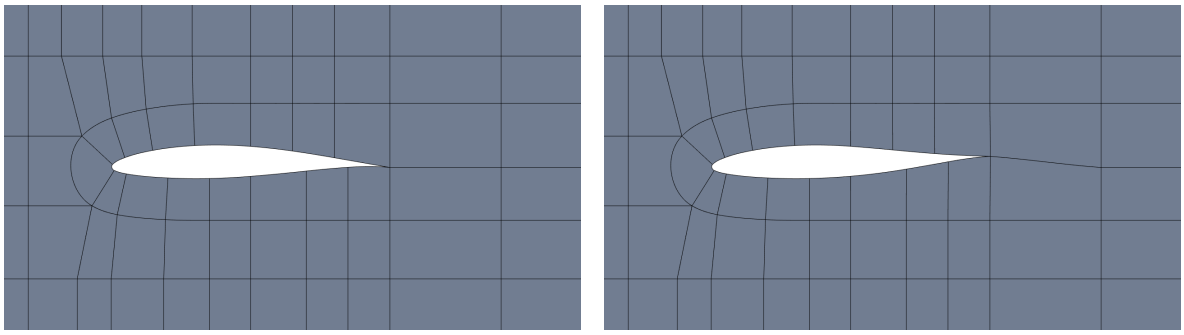


(a) undeformed (b) Morphed

Figure 17: Rational Bézier representation of the morphing airfoil

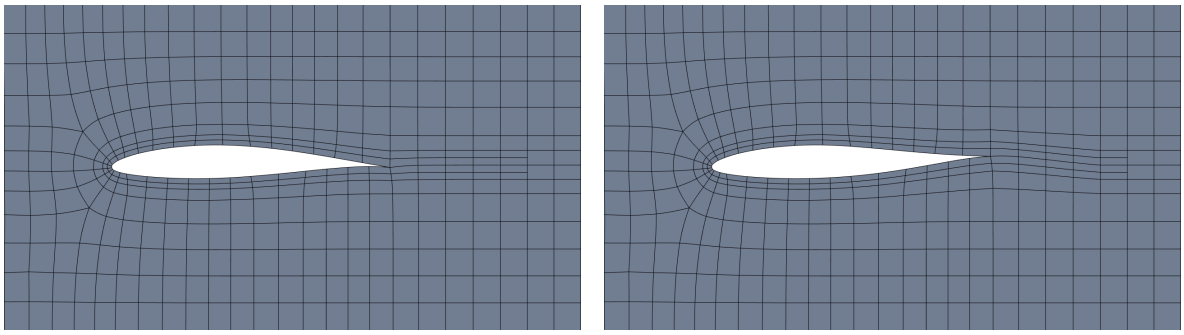
The computational domain is then constructed according to the approach described in section 3.1. A small set of fixed cubic elements are built around the airfoil, and then connected to the rational Bézier edges defining the boundary, yielding the primary mesh depicted in figure 18. Finally, some refinement steps are achieved to obtain the secondary mesh that will be employed for the computations. Contrary to the previous transonic airfoil case, we do not use AMR here because the boundary layer and the wake have to be refined for all

configurations. Therefore, we apply refinement in these areas *a priori* and avoid AMR techniques and related computational overhead. The secondary mesh is shown in figure 19, for the coarsest grid tested.



(a) Undeformed (b) Morphed

Figure 18: Primary mesh around the morphing airfoil



(a) Undeformed (b) Morphed

Figure 19: Secondary mesh around the morphing airfoil

The flow conditions for this problem correspond to a subsonic laminar flow with a freestream Mach number $M_\infty = 0.2$ and a Reynolds number with respect to the airfoil chord $Re_\infty = 5000$. The angle of attack selected for the study is $\alpha = 3°$, since it corresponds to the maximum aerodynamic performance of the rigid airfoil, in terms of lift-to-drag ratio.

A grid convergence study is conducted using a coarse (28960 d.o.f.), medium (56192 d.o.f.) and fine (125824 d.o.f.) grid, for a rigid airfoil and a morphing airfoil with the parameters $L = 50\%$, $A = 4\%$, $f = 0.4$. We also tested an intermediate grid (38624 d.o.f.), denoted as optimized, which is characterised by the same resolution level of the medium mesh around the airfoil with a much coarser discretization of the far wake area. We compute for each grid the time-average values of the lift, drag and pitching moment coefficients and the Strouhal number. The results are reported in table 3 for the rigid airfoil. The considered quantities are already well estimated using the coarse mesh. The results obtained with the optimised grid are in line with those predicted with the medium refinement level.

For the morphing airfoil, we also observe the convergence of the power coefficient C_P,

	N_{el}	d.o.f.	$\bar{C}_l$	$\bar{C}_d$	$\bar{C}_m$	St
coarse	1810	28960	0.2145	0.0641	-0.0261	1.9366
medium	3512	56192	0.2212	0.0644	-0.0266	1.9843
fine	7864	125824	0.2225	0.0645	-0.0267	2.0009
optimized	2414	38624	0.2215	0.0645	-0.0267	1.9858

Table 3: Rigid airfoil, convergence of the aerodynamic coefficients

which quantifies the amount of power necessary actuate the morphing:

$$C_P = \frac{1}{\frac{1}{2}\rho_\infty U_\infty^3 c} \oint_{\partial\Omega_a} (p\,\mathbf{n} - \tau \cdot \mathbf{n}) \cdot \mathbf{V}_g \,\mathrm{d}\Gamma, \tag{20}$$

where p is the pressure, τ is the viscous stress tensor, $\mathbf{V}_g$ the surface velocity and $\partial\Omega_a$ is airfoil surface. Results are reported in table 4 for each mesh. As for the rigid airfoil, the flow solver is already capable of predicting the correct flow physics using the coarse mesh and a sufficiently accurate estimate of the quantities of interest. The results computed with the optimised mesh are in line with the medium mesh, except for the average lift coefficient $\bar{C}_l$, which is slightly underestimated. This effect may be caused by the reduced resolution in the wake region.

	N_{el}	d.o.f.	$\bar{C}_l$	$\bar{C}_d$	$\bar{C}_m$	$\bar{C}_P$
coarse	1810	28960	0.7471	0.0564	-0.1225	0.0859
medium	3512	56192	0.7420	0.0572	-0.1204	0.0855
fine	7864	125824	0.7438	0.0577	-0.1205	0.0844
optimized	2414	38624	0.7393	0.0572	-0.1200	0.0855

Table 4: Morphing airfoil, convergence of the aerodynamic coefficients

Comparing the aerodynamic coefficients for the rigid and the morphing airfoil, we can notice that the average lift and pitching moment are significantly increased, whereas the drag is slightly reduced. From the results of table 4, we could assume that the coarse mesh provides a sufficiently reliable estimate of the aerodynamic coefficients. However, for some combinations of the morphing parameters (not presented here), the simulation performed with the coarse mesh does not converge to a periodic flow, due to a lack of resolution of the boundary layer close to the leading edge. The medium grid has the adequate resolution close to the wall, but the flow computation would be too expensive in the context of a complete optimization procedure. The optimised mesh has the same wall resolution of the medium refinement level with a 25% reduction of the computational time. Furthermore, the relative difference between the C_l computed with the two grids is less than 1%. We hence conclude that the optimised mesh provides a reasonable compromise between accuracy, reliability and computational cost.

8.2 Single-objective optimization

In this first optimization exercise, we aim at maximizing the time-averaged lift coefficient $\bar{C}_l$, whereas the morphing parameters are considered as optimization variables $\mathbf{z} = (L, A, f)$:

$$g(L, A, f) = -\bar{C}_l. \tag{21}$$

Since morphing can induce strong oscillations of lift, which are undesirable, we force the lift to remain positive during the whole morphing oscillation by employing the following constraint function:

$$h(L, A, f) = -\check{C}_l, \tag{22}$$

where $\check{C}_l$ is the minimum value of C_l over a flow period.

To perform the optimization in realistic conditions, we limit the upper bound of the amplitude to a third of the airfoil thickness and the maximum morphing length to 60% of the chord. The maximum oscillation frequency is twice the natural vortex shedding frequency $f_s = 0.4$. The design space is finally defined as:

$$\mathcal{D} = \Big\{(L, A, f) : L \in [0.2,\, 0.6],\, A \in [0.005,\, 0.040],\, f \in [0.2,\, 1.0]\Big\}. \tag{23}$$

Accounting for the computational cost related to unsteady flow simulations, we consider a total computational budget of 40 simulations, including 8 points for the DOE phase followed by 32 simulations for the optimization phase.

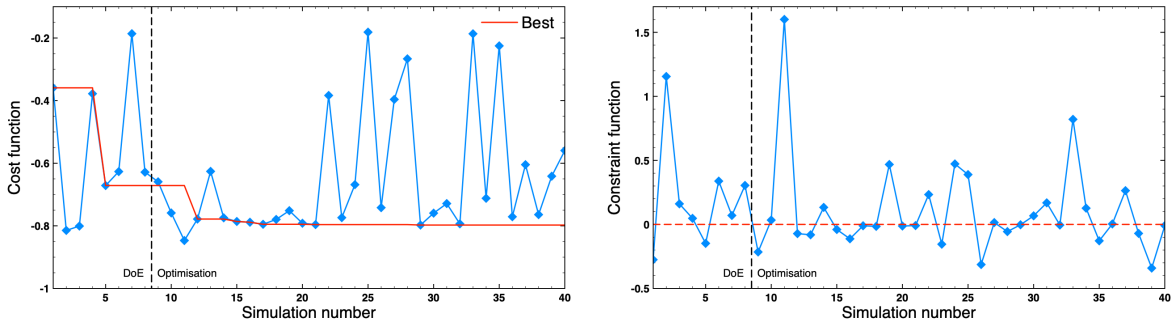


Figure 20: Single-objective optimization of morphing: evolution of cost function and constraint

The evolution of the cost and constraint functions is illustrated in Fig. 20. We can remark that the constraint plays a significant role in the optimization. Among the 8 DOE points, only 2 respect the constraint. For nearly half of the 40 simulations a negative value of $\check{C}_l$ is found. Moreover, the first iterations that present an improvement of the average lift coefficient do not satisfy the constraint and are therefore discarded. As the Gaussian process models are enriched, the algorithm is able to identify the region where the constraint is satisfied and the cost function is minimised. The optimum is found after 21 simulations during the optimization phase, corresponding to the evaluation index $N_s = 29$. The optimal set of morphing parameters is (0.2312, 0.0394, 0.7197), with $\bar{C}_l = 0.7975$ and $\check{C}_l = 0.0027$.

We then validate the results of the optimisation by performing a comparison with the fine mesh, for three relevant configurations found during the search:

- $N_s = 26$: best trade-off between $\bar{C}_l$ and $\check{C}_l$,
- $N_s = 29$: optimal design,
- $N_s = 39$: best value of $\check{C}_l$.

We report in table 5 the results of the validation test. Some small variations of the aerodynamic coefficients are registered between the two grids. However, for all the selected design

points the results obtained with the fine mesh are in line with the predictions of the optimisation loop and, more importantly, the flow physics is not altered when the mesh is refined.

				Optimised mesh			Fine mesh		
N_s	L	A	f	$\bar{C}_l$	$\check{C}_l$	$\bar{C}_p$	$\bar{C}_l$	$\check{C}_l$	$\bar{C}_p$
26	0.2004	0.0306	0.6615	0.7419	0.3133	0.0557	0.7411	0.3241	0.0543
29	0.2312	0.0394	0.7197	0.7975	0.0027	0.1777	0.8011	0.0204	0.1737
39	0.2000	0.0258	0.5176	0.6415	0.3410	0.0141	0.6329	0.3415	0.0137

Table 5: Single-objective optimization of morphing: validation of three selected design points

8.3 Multi-objective optimization

Using the data collected during the single-objective study as a Design of Experiment, we perform 40 more iterations considering a multi-objective criterion. The first objective is still the maximisation of the time-averaged lift coefficient. In order to further control the time evolution of the lift, we maximise $\check{C}_l$ as well. In contrast, high values of $\bar{C}_l$ appears to be characterised by elevated power consumptions of the morphing actuation. It is thus interesting to minimise the average power coefficient $\bar{C}_P$ to find the most energetically efficient morphing design. Therefore, we adopt the following set of cost functions:

$$\begin{cases} g_1(L, A, f) = -\bar{C}_l, \\ g_2(L, A, f) = -\check{C}_l, \\ g_3(L, A, f) = \bar{C}_P. \end{cases} \tag{24}$$

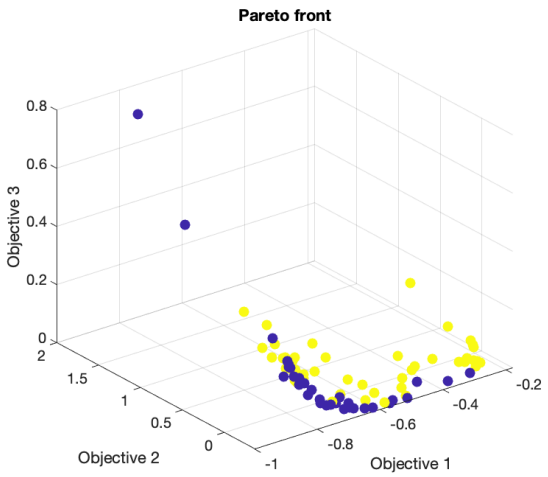


Figure 21: Visualisation of the Pareto front

The distribution of the evaluations in the objective space is shown in Fig. 21. The Pareto set is coloured in violet, whereas the dominated points are represented in yellow. Among the 80 observations, 32 are on the Pareto front. Interestingly, all of the three configurations

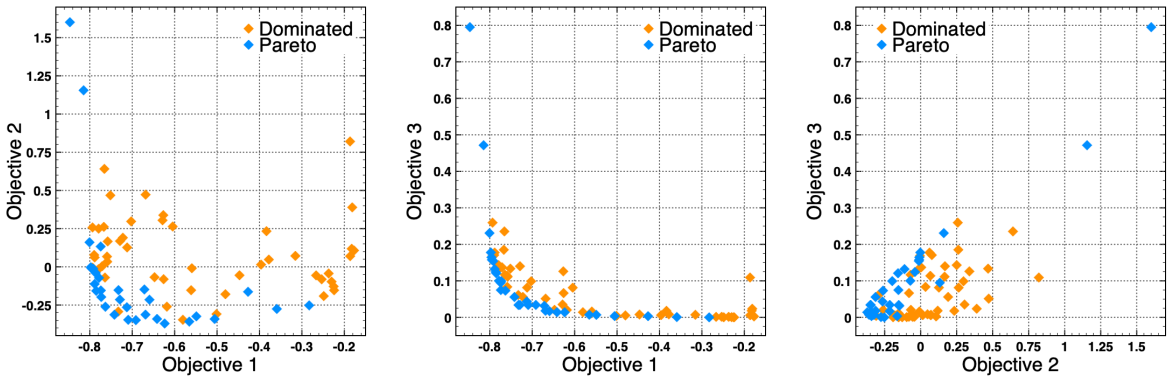


Figure 22: Bi-objective scatter plots

analysed in Table 5 are Pareto optimal. For a better interpretation of the shape of the Pareto front, we compare each pair of objectives in Fig. 22. We can clearly observe that two extreme configurations stand out from the rest, with high average lift, but very poor performance in terms of the other two objectives. In the design space, the two said observations are close to the upper amplitude, length and frequency bound, meaning that the fluid receives a large amount of energy from the deformation of the airfoil, causing large oscillations of lift. There is certainly a lack of exploration of the Pareto front in this region, however these configurations are clearly not interesting for the application. We can also observe that the region of interest, characterized by a positive minimum lift, a low power coefficient and a significant improvement of the time-averaged lift, counts several simulated configurations.

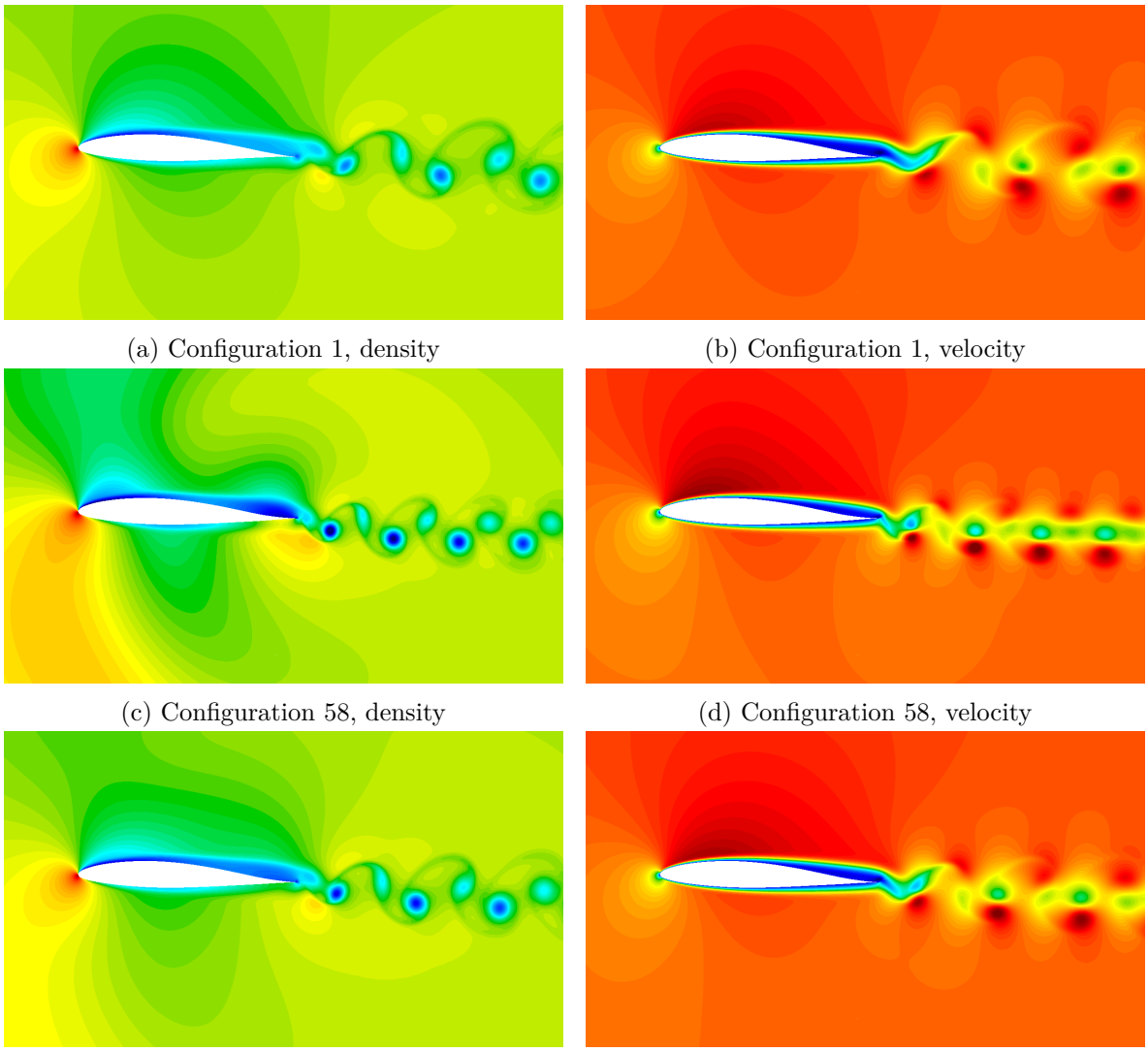
We then select three configurations from the Pareto set for a deeper investigation:

- $N_s = 1$: best $\bar{C}_P$ for which $\bar{C}_l$ is improved w.r.t. the rigid airfoil;
- $N_s = 58$: best $\check{C}_l$;
- $N_s = 76$: balanced trade-off between the three objectives.

The flow computations for the considered observations are validated using the fine mesh. We can remark in table 6 that no significant discrepancies are found when the mesh is refined. We can see that both $\check{C}_l$ and $\bar{C}_l$ are enhanced with respect to the rigid airfoil for all the analysed morphing designs. The smallest improvement is found for $N_S = 1$, which, on the other hand, is extremely efficient in terms of power consumption. Configuration 76 performs well in all the three objectives, requiring 4 times less actuation power than $N_S = 58$, for only 1.25 times less average lift.

				Optimised mesh			Fine mesh		
N_s	L	A	f	$\bar{C}_l$	$\check{C}_l$	$\bar{C}_p$	$\bar{C}_l$	$\check{C}_l$	$\bar{C}_p$
1	0.2134	0.0111	0.4013	0.3591	0.2754	0.00077	0.3539	0.2701	0.00078
58	0.2000	0.0219	0.5850	0.6235	0.3710	0.0139	0.6117	0.3672	0.0136
76	0.2000	0.0178	0.4721	0.5057	0.3413	0.0035	0.4947	0.3351	0.0034

Table 6: Multi-objective optimization of morphing: validation of three selected design points



(a) Configuration 1, density

(b) Configuration 1, velocity

(c) Configuration 58, density

(d) Configuration 58, velocity

(e) Configuration 76, density

(f) Configuration 76, velocity

Figure 23: Multi-objective optimisation: solution fields

Finally, we illustrate the flow fields of the three analysed configurations in Fig. 23. It is clearly possible to observe the effect of the morphing frequency on the spacing of the vortex pairs forming the wake. Moreover, a significantly larger recirculation region is found for $N_S = 1$, probably due to the smaller amplitude and frequency. The configuration with the strongest oscillations of lift ($N_S = 58$) is associated with larger density and velocity gradients.

9 Conclusion

A fully integrated geometry-simulation-optimization framework has been presented in this work. The whole design optimization loop relies on a unified geometry management, yielding a CAD-consistent aerodynamic design procedure. This has been achieved thanks to the properties of NURBS representations, allowing refinement, splitting, degree elevation, combined

with a Discontinuous Galerkin method modified to account for these representations. The complexity overhead in the solver, mostly related to the integration of non-linear domains, is clearly exceeded by the flexibility offered by the resulting tool and the ease of implementation of the design chain.

The proposed methodology has been applied to the shape optimization of an airfoil in transonic regime, including 16 design parameters. A global optimization has been achieved using less than one hundred simulations. The local refinement associated to the preservation of the geometry allowed to initiate the computations with very coarse grids while ensuring an accurate shock capturing. The second application concerned morphing airfoils, which are characterized by strongly non-linear unsteady flows. A multi-objective optimization was carried out, demonstrating the robustness and efficiency of the proposed methodology.

Future developments will target the resolution of coupled problems, such as fluid-structure interactions, for which an accurate treatment of the interface is mandatory.

Code Repository

The developed methodology is implemented in the **Igloo** software suite, which has been employed to perform all the presented computations. The source code and data are available, under the GNU General Public Licence v3, at the following repository: `https://gitlab.inria.fr/igloo/igloo/-/wikis/home`.

The adopted optimization algorithms are available in a set of open-source R packages (36; 37).

Acknowledgements

The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

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

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

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

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

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

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

[7] 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).

[8] A. Keane, P. Nair, Computational Approaches for Aerospace Design: The Pursuit of Excellence, John-Wiley and Sons, 2005.

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

[10] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes, G. Sangalli, Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes, Mathematical Methods and Models in Applied Sciences 16 (2006) 1031–1090.

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

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

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

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

[15] C. Giannelli, B. Jüttler, H. Speleers, Thb-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (2012) 485–498.

[16] J. Evans, T. Hughes, Isogeometric divergence-conforming B-Splines for the steady Navier–Stokes equations, Mathematical Models and Methods in Applied Sciences 23 (2013).

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

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

[19] B. S. Hosseini, M. Möller, S. Turek, Isogeometric analysis of the navier-stokes equations with taylor-hood b-spline elements, Applied Mathematics and Computation 267 (2015).

[20] M. Möller, A. Jaeschke, High-Order Isogeometric Methods for Compressible Flows, Springer International Publishing, Cham, 2020, pp. 31–39.

[21] S. Yu, R. Feng, T. Liu, An isogeometric discontinuous galerkin method for euler equations, Mathematical Methods and Models in Applied Sciences 40 (2017).

[22] R. Duvigneau, Isogeometric analysis for compressible flows using a Discontinuous Galerkin method, Computer Methods in Applied Mechanics and Engineering 333 (2018).

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

[24] S. Pezzano, R. Duvigneau, A NURBS-based Discontinuous Galerkin method for conservation laws with high-order moving meshes, Journal of Computational Physics 434 (2021).

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

[26] 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.

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

[28] R. Garnett, Bayesian Optimization, Cambridge University Press, 2022. In preparation.

[29] D. Jones, Efficient global optimization of expensive black-box functions, Journal of Global Optimization 13 (1998).

[30] M. T. Emmerich, K. C. Giannakoglou, B. Naujoks, Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels, IEEE Transactions on Evolutionary Computation 10 (2006) 421–439.

[31] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[32] M. D. McKay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.

[33] D. Jones, A taxonomy of global optimization methods based on response surfaces, Journal of Global Optimization 21 (2001) 345–383.

[34] M. T. Emmerich, A. H. Deutz, J. W. Klinkenberg, Hypervolume-based expected improvement: Monotonicity properties and exact computation, in: Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE, 2011, pp. 2147–2154.

[35] M. Schonlau, W. J. Welch, D. R. Jones, Global versus local search in constrained optimization of computer models, 1998.

[36] O. Roustant, D. Ginsbourger, Y. Deville, Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization, Journal of Statistical Software 51 (2012) 1–55.

[37] M. Binois, V. Picheny, GPareto: An R package for Gaussian-process-based multiobjective optimization and analysis, Journal of Statistical Software, Articles 89 (2019) 1–30.

[38] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, D. J. Inman, A review of morphing aircraft, Journal of Intelligent Material Systems and Structures 22 (2011) 823–877.

[39] J. Fincham, M. Friswell, Aerodynamic optimisation of a camber morphing aerofoil, Aerospace Science and Technology 43 (2015) 245–255.

[40] N. Simiriotis, K. Diakakis, G. Jodin, F. Kramer, A. Marouf, Y. Hoarau, J.-F. Rouchon, G. Tzabiras, M. Braza, Synthesis on High-Fidelity Numerical simulation of a morphing A320 wing in subsonic speeds and sensitivity evaluation, 2019. doi:10.2514/6.2019-2911.