



**HAL**  
open science

# Shape sensitivity analysis in aerodynamics using an isogeometric Discontinuous Galerkin method

Maxime Stauffert, Régis Duvigneau

► **To cite this version:**

Maxime Stauffert, Régis Duvigneau. Shape sensitivity analysis in aerodynamics using an isogeometric Discontinuous Galerkin method. *SIAM Journal on Scientific Computing*, 2021, 43 (5), 10.1137/20M1356269 . hal-02962207v2

**HAL Id: hal-02962207**

**<https://inria.hal.science/hal-02962207v2>**

Submitted on 27 Sep 2021

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

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

# SHAPE SENSITIVITY ANALYSIS IN AERODYNAMICS USING AN ISOGEOMETRIC DISCONTINUOUS GALERKIN METHOD

M. STAUFFERT\* AND R. DUVIGNEAU†

**Abstract.** The sensitivity equation method aims at estimating the derivative of the solution of partial differential equations with respect to a parameter of interest. The objective of this work is to investigate the ability of an isogeometric Discontinuous Galerkin (DG) method to evaluate accurately sensitivities with respect to shape parameters originating from Computer-Aided Design (CAD), in the context of compressible aerodynamics. The isogeometric DG method relies on Non-Uniform Rational B-Spline representations, which allow to define a high-order numerical scheme for Euler/Navier-stokes equations, fully consistent with CAD geometries. We detail how this formulation can be exploited to construct an efficient and accurate approach to evaluate shape sensitivities. A particular attention is paid to the treatment of boundary conditions for sensitivities, which are more tedious in the case of geometrical parameters. The proposed methodology is first verified on a test-case with analytical solution and then applied to two more demanding problems, that concern the inviscid flow around an airfoil with its camber as shape parameter and the unsteady viscous flow around a three-element airfoil with the positions of slat and flap as parameters.

**Key words.** Sensitivity analysis, Discontinuous Galerkin, isogeometric analysis

**AMS subject classifications.** 35L60, 65M50, 65M60, 76N25

**1. Introduction.** Sensitivity Analysis (SA) refers to the study of how a model output is influenced by a change of an input parameter. It is therefore a classical topic in engineering, in the perspective of optimization of the performance of a complex system, quantification of the output uncertainty according to the knowledge of input statistics, etc. This task can be achieved in several ways, which depend on the nature of the model of interest, the amplitude of the parameter perturbation or its stochastic / deterministic origin [18]. In this work, we focus on models governed by Partial Differential Equations (PDEs), and more specifically on non-linear systems of conservation laws. Such models are employed in transport problems, like in fluid mechanics, wave propagation or even crowd dynamics. In this context, the current study concerns the estimation of the derivative of a PDE solution with respect to an input parameter. Thus, SA is considered here as a local approach and only small perturbations of the parameter are under consideration. Obviously, we make the assumption that the PDE solution is regular enough to be differentiated. Note that it is not always the case with non-linear hyperbolic systems, as studied in [1, 14]. Two different approaches can be carried out to estimate the derivative of a PDE model with respect to an input parameter: the adjoint and the sensitivity equation methods. In the former, a single functional output can be differentiated with respect to several input parameters with a computational cost independent from the number of parameters. In the latter, the whole solution field can be differentiated with respect to a single parameter, for a similar cost. As consequence, the adjoint approach is typically suited to optimization involving a large number of design parameters. On the contrary, the sensitivity approach is suited to the propagation of uncertainty of an input parameter to the whole solution field. Both can be performed at the continuous level (*differentiate-then-discretize*) or at the discrete level (*discretize-then-differentiate*).

---

\*Université Côte d'Azur, Inria, CNRS, LJAD, 2004 route des lucioles 06902 Sophia-Antipolis, France, ([contact@maxime-stauffert.fr](mailto:contact@maxime-stauffert.fr)).

†Université Côte d'Azur, Inria, CNRS, LJAD, 2004 route des lucioles 06902 Sophia-Antipolis, France, ([regis.duvigneau@inria.fr](mailto:regis.duvigneau@inria.fr)).

The adjoint method has been studied in depth in several works, from the pioneer developments using the continuous [28, 19] or discrete [2, 25] approach, and applied to optimization in several disciplines [5, 20, 21]. However, less attention has been paid to the sensitivity equation method, although it has been pointed out that this approach can be powerful for uncertainty propagation [32, 33] or low-cost evaluation of neighboring solutions [12]. The current work focuses on this methodology.

The Sensitivity Equation Method (SEM) consists of solving a new system of PDEs, for each parameter of interest, obtained by differentiating the original PDE model. After resolution of the resulting system, one obtains as solution the derivative of the solution fields with respect to the parameter. As mentioned above, there are two ways to derive a numerical scheme for the sensitivity equations, the *discretize-then-differentiate* approach and the *differentiate-then-discretize* one. The first approach is referred as the Discrete SEM (DSEM) [23] and can be carried out using Automatic Differentiation (AD) software [25, 29], which can generate the computational code automatically and thus facilitate the implementation. However, the main issue concerning the DSEM is related to the fact that the mesh is involved in the *discretize* step and, consequently, some mesh derivative terms appear in the *differentiate* step. Alternatively, we prefer to use the Continuous SEM (CSEM) [3], for which the *differentiate* step is achieved on the continuous equations, thus no mesh derivatives need to be computed and the approach can be considered as more flexible. For a deeper comparison between DSEM and CSEM, one refers to [4] and [22].

Nevertheless, a difficulty arises in the CSEM when the parameter of interest controls the geometry of the computational domain. In this case, the boundary conditions for the sensitivity are not known exactly but depend on the gradients of the solution at the boundary, which can be tedious to estimate accurately and significantly impacts the accuracy of the sensitivity fields. To remedy to this situation, some specific techniques have been introduced, such as a local projection reconstruction [35, 26] or a least square reconstruction in the neighborhood of the boundary [11]. More recently, an alternate formulation, namely the Lagrangian SEM, has been proposed [6]. It consists of replacing the estimation of the flow derivative with respect to the boundary shape by the estimation of the derivative with respect to the domain deformation. This change of viewpoint allows to set an exact boundary condition but necessitates two modifications: firstly, a mapping should be introduced to relate the deformed and undeformed configurations, for instance using a physical elasticity model [6] or a geometrical approach like radial basis functions [8]. Secondly, additional terms should be inserted in the sensitivity system, reflecting the Lagrangian nature of the differentiation and using the derivative of this geometrical mapping. In the current work, this approach is not retained. Alternatively, we intend to investigate more in depth the Eulerian formulation in the specific context of the isogeometric Discontinuous Galerkin (DG) framework and overcome the issue of the boundary conditions for the sensitivities by using very high-order schemes associated with an exact description of the geometry.

IsoGeometric Analysis (IGA) has been introduced in [17] as a way to solve PDE systems consistently with Computer-Aided Design (CAD) representations. The core idea of IGA is to use the same Non-Uniform Rational B-Spline (NURBS) basis to represent both the geometry and the solution, in a standard Finite-Element Method (FEM). Indeed, NURBS are widely used in CAD systems to define complex geometries [27]. As consequence, IGA allows a seamless integration of CAD and FEM. Recently, IGA has been extended to DG methods, for a better numerical treatment of hyperbolic systems of conservation laws [9, 24, 30, 34]. The current study is conducted

in this framework. As shown in previous works [9, 10], the NURBS-based DG scheme allows to solve hyperbolic systems with a high accuracy and is fully consistent with CAD representations. Therefore, we intend to evaluate its potentiality for sensitivity analysis including shape parameters derived from CAD representations. The content of this paper is organized as follows: we first present the principle of the CSEM, with a particular emphasis on the treatment of shape parameters. Then, we summarize the main properties of NURBS representations and describe the NURBS-based DG scheme for compressible Euler/Navier-Stokes equations. In a third part, we derive the sensitivity equations. Finally, we demonstrate the capabilities of the proposed approach for a problem with an analytical solution and for two more demanding applications. The first one concerns the inviscid flow around an airfoil with its camber as shape parameter. In particular, we discuss the accuracy of a linear extrapolation of the flow with respect to the shape. The second application deals with the viscous flow around a three-element airfoil with the positions of slat and flap as parameters. This illustrates the flexibility of the proposed methodology to handle complicated geometrical configurations and its potentiality to study complex physic behaviors.

## 2. Principle of continuous sensitivity analysis.

**2.1. General formulation.** For the sake of genericity, we consider first that the governing equations can be expressed as a system of conservation laws, that writes in the physical domain  $\Omega$  as:

$$(2.1) \quad \partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{0},$$

where  $\mathbf{U}: (\mathbf{x}, t) \in \Omega \times \mathbb{R}^+ \mapsto \mathbf{U}(\mathbf{x}, t) \in \mathbb{R}^n$  is the vector of state variables at any point and any time and  $\mathbf{F}: \mathbf{U} \in \mathbb{R}^n \mapsto \mathbf{F}(\mathbf{U}) \in (\mathbb{R}^n)^d$  the corresponding physical flux. The integers  $d$  and  $n$  are respectively the dimension of  $\Omega$  (which means  $\Omega \subset \mathbb{R}^d$ ) and the number of state variables (or equations). Finally, this system is associated with boundary conditions, which can be of Dirichlet type:  $\mathbf{U}(\mathbf{x}_{\Gamma_D}, t) = \mathbf{U}_D$  on the boundary  $\Gamma_D$ , or Neumann type:  $\partial_{\mathbf{n}} \mathbf{U}(\mathbf{x}_{\Gamma_N}, t) = \delta \mathbf{U}_N$  on  $\Gamma_N$ , with  $\partial\Omega =: \Gamma = \Gamma_D \cup \Gamma_N$ .

We consider a parameter of the problem  $a \in \mathbb{R}$ , which impacts the solution. As consequence, the state variables depend now on  $a$ , which is denoted  $\mathbf{U}(x, t; a)$ . The objective of continuous sensitivity analysis is to estimate the derivative of the solution with respect to the parameter  $a$ . This quantity is referred as the sensitivity of  $\mathbf{U}$  and denoted:

$$(2.2) \quad \mathbf{U}' = \partial_a \mathbf{U}.$$

We assume obviously that the state vector is regular enough with respect to  $a$ , so that its sensitivity exists. Theoretical foundations can be found in [1]. We differentiate now the state equation (2.1) with respect to  $a$  in order to obtain a new system governing the sensitivity. All the above functions  $\mathbf{U}$ ,  $\mathbf{F}$ ,  $\mathbf{U}_D$  and  $\mathbf{U}_N$  depend *a priori* on  $a$ . Moreover, in the case of smooth solutions, one can commute  $\partial_a$  with  $\partial_t$  and  $\nabla$ . Note that this commutation is usually not possible in presence of discontinuities in the solution. For a study of sensitivity analysis in the context of discontinuous solutions, one can refer to [14]. Finally, the differentiation of the state equation (2.1) with respect to  $a$  yields therefore:

$$\partial_a (\partial_t \mathbf{U}) + \partial_a (\nabla \cdot \mathbf{F}(\mathbf{U})) = \partial_t (\partial_a \mathbf{U}) + \nabla \cdot (\partial_a [\mathbf{F}(\mathbf{U})]) = \mathbf{0},$$

or, after simplification:

$$(2.3) \quad \partial_t \mathbf{U}' + \nabla \cdot (\mathbf{F}'(\mathbf{U}, \mathbf{U}')) = \mathbf{0},$$

where  $\mathbf{F}'$  is the sensitivity flux, i.e. the derivative of the physical flux with respect to  $a$ . One can notice that the sensitivity equation (2.3) has the same conservative form as the state equation (2.1). Moreover, the sensitivity flux can be expressed as:

$$\mathbf{F}' = \partial_a [\mathbf{F}(\mathbf{U})] = \partial_{\mathbf{U}} \mathbf{F} \cdot \partial_a \mathbf{U} = \partial_{\mathbf{U}} \mathbf{F} \cdot \mathbf{U}'.$$

As consequence, the sensitivity flux is linear with respect to  $\mathbf{U}'$  and its Jacobian is identical to the Jacobian of the physical flux:

$$\partial_{\mathbf{U}'} \mathbf{F}' = \partial_{\mathbf{U}} \mathbf{F}$$

This property is important because it allows to establish that, if the state equation is hyperbolic, the sensitivity equation is hyperbolic too, with the same characteristic speeds. This (linear) system can be solved independently from the state equation, after evaluation of the solution  $\mathbf{U}$ . However, as will be detailed later, it may be convenient to gather the two systems into a global conservation law for a concatenated variable:

$$(2.4) \quad \partial_t \mathbf{W} + \nabla \cdot \mathbf{G}(\mathbf{W}) = \mathbf{0},$$

with:

$$\mathbf{W} = \begin{pmatrix} \mathbf{U} \\ \partial_a \mathbf{U} \end{pmatrix} = \begin{pmatrix} \mathbf{U} \\ \mathbf{U}' \end{pmatrix} \quad \text{and} \quad \mathbf{G}(\mathbf{W}) = \begin{pmatrix} \mathbf{F}(\mathbf{U}) \\ \partial_a [\mathbf{F}(\mathbf{U})] \end{pmatrix} = \begin{pmatrix} \mathbf{F}(\mathbf{U}) \\ \mathbf{F}'(\mathbf{U}, \mathbf{U}') \end{pmatrix}.$$

The boundary conditions for the sensitivity can now be derived, by differentiating the boundary conditions for the state variables. However, two different cases should be considered, depending if the parameter  $a$  controls the geometry of the domain, i.e.  $\Omega(a)$  (shape parameter), or not (value parameter).

**2.2. Boundary conditions for value parameters.** If the parameter  $a$  only influences the value of the boundary condition, i.e.  $\mathbf{U}(\mathbf{x}_{\Gamma_D}, t; a) = \mathbf{U}_D$  on  $\Gamma_D$  or  $\partial_{\mathbf{n}} \mathbf{U}(\mathbf{x}_{\Gamma_N}, t; a) = \delta \mathbf{U}_N$  on  $\Gamma_N$ , one simply obtains by differentiation:

$$\mathbf{U}'(\mathbf{x}_{\Gamma_D}, t; a) = \partial_a (\mathbf{U}_D),$$

for the sensitivity Dirichlet condition and:

$$\partial_{\mathbf{n}} \mathbf{U}'(\mathbf{x}_{\Gamma_N}, t; a) = \partial_a (\delta \mathbf{U}_N),$$

for the sensitivity Neumann condition.

**2.3. Boundary conditions for shape parameters.** If the geometry of the domain depends on  $a$ , the derivation of the boundary conditions for the sensitivity is more complex, because it should account for the spatial dependency of the boundary condition [11]. The Dirichlet boundary condition for the state variables thus writes  $\mathbf{U}(\mathbf{x}_{\Gamma_D}(a), t; a) = \mathbf{U}_D$ , with  $\mathbf{x}_{\Gamma_D}(a)$  the coordinates of the boundary curve parameterized by  $a$ . Its differentiation yields:

$$\mathbf{U}'(\mathbf{x}_{\Gamma_D}, t; a) + \nabla \mathbf{U} \cdot \partial_a \mathbf{x}_{\Gamma_D} = D_a (\mathbf{U}_D),$$

where  $D_a (\mathbf{U}_D)$  represents the total derivative of  $\mathbf{U}_D$  with respect to  $a$ . The Dirichlet condition for the sensitivity is then:

$$(2.5) \quad \mathbf{U}'(\mathbf{x}_{\Gamma_D}, t; a) = D_a (\mathbf{U}_D) - \nabla \mathbf{U} \cdot \partial_a \mathbf{x}_{\Gamma_D}.$$

The Neumann condition is even more tedious because it implies the normal derivative of the state variables, which depends on the geometry of the boundary:

$$\partial_{\mathbf{n}}\mathbf{U}(\mathbf{x}_{\Gamma_N}(a), t; a) = \nabla\mathbf{U}(\mathbf{x}_{\Gamma_N}(a), t; a) \cdot \mathbf{n}(a) = \delta\mathbf{U}_N.$$

The differentiation of this condition yields finally:

$$(2.6) \quad \partial_{\mathbf{n}}\mathbf{U}'(\mathbf{x}_{\Gamma_N}, t; a) = D_a(\delta\mathbf{U}_N) - (\nabla^2\mathbf{U} \cdot \partial_a\mathbf{x}_{\Gamma_N}) \cdot \mathbf{n} - \nabla\mathbf{U} \cdot \partial_a\mathbf{n}(a).$$

It is worth noticing that the additional terms make the boundary conditions for the sensitivity dependent on the derivatives of the solution. They reflect that any change in the parameter  $a$  modifies the location of the boundary condition. In the forthcoming sections, we describe how this general approach can be implemented in the context of an isogeometric DG method applied to compressible flow equations.

### 3. NURBS-based Discontinuous Galerkin method.

**3.1. Compressible Euler equations.** We consider now as conservation laws the compressible Euler equations, which can be expressed for the 2D case as:

$$(3.1) \quad \partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ u(E + p) & v(E + p) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where  $\rho$  is the density,  $\mathbf{u} = (u, v)$  the velocity vector and  $E$  the total energy per unit of volume. The pressure  $p$  is given by the perfect gas law:

$$(3.2) \quad p = (\gamma - 1) \left( E - \frac{1}{2}\rho(u^2 + v^2) \right),$$

with  $\gamma = 1.4$ . A classical slip condition is prescribed at the wall boundaries, which consists of imposing in a weak sense  $\mathbf{u} \cdot \mathbf{n} = un_x + vn_y = 0$ . At inlet or outlet boundaries, state variables are weakly imposed, according to the local flow direction.

**3.2. Compressible Navier-Stokes equations.** Accounting now for viscous effects, one obtains compressible Navier-Stokes equations, which can be expressed as:

$$(3.3) \quad \partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ u(E + p) & v(E + p) \end{pmatrix} = \nabla \cdot \begin{pmatrix} 0 & 0 \\ \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \\ u\tau_{xx} + v\tau_{yx} & u\tau_{xy} + v\tau_{yy} \end{pmatrix},$$

where the symmetric viscous stress tensor  $\bar{\tau}$  for Newtonian fluids reads:

$$(3.4) \quad \tau_{xx} = \frac{2}{3}\mu \left( 2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \quad \tau_{xy} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad \tau_{yy} = \frac{2}{3}\mu \left( 2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right).$$

Note that the heat flux has been neglected here and the viscosity  $\mu$  is considered as constant for the sake of simplicity. Contrary to the Euler equations, no-slip conditions are prescribed at the wall boundaries, which consist of imposing  $\mathbf{u} = \mathbf{0}$  in a weak sense.

**3.3. Discontinuous Galerkin discretization.** For any state variable  $w \in \{\rho, \rho u, \rho v, E\}$ , a weak formulation is obtained by multiplying equation (3.1) or (3.3) by a test function  $\varphi$  and integrating over the physical domain  $\Omega$ :

$$(3.5) \quad \int_{\Omega} (\partial_t w + \nabla \cdot \mathbf{f}) \varphi \, d\Omega = 0,$$

where  $\mathbf{f}$  is the physical flux associated to  $w$ . By integrating by parts, one obtains classically:

$$(3.6) \quad \int_{\Omega} (\partial_t w) \varphi \, d\Omega - \int_{\Omega} \mathbf{f} \cdot \nabla \varphi \, d\Omega + \int_{\Gamma} (\mathbf{f} \cdot \mathbf{n}) \varphi \, d\Gamma = 0.$$

The domain is then split in a set of elements  $\Omega = \cup_{j=1}^N \Omega_j$ , for which a local functional basis of size  $p + 1$  is defined:

$$\mathcal{P}^j = \{\mathbf{x} \in \Omega_j \mapsto P_{i,j}(\mathbf{x}) \in \mathbb{R}, i \in \{0, \dots, p\}\}.$$

The components of the solution  $\mathbf{U}$  are finally decomposed into a finite space of piecewise-defined functions:

$$w(\mathbf{x}, t) = \sum_{i=0}^p w_{i,j}(t) P_{i,j}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_j \quad \forall t > 0,$$

where  $w_{i,j}$  are the time dependent coefficients to be solved. If one chooses  $\varphi$  such that  $\varphi|_{\Omega_j} = P_{k,j}$  and  $\varphi|_{\Omega \setminus \Omega_j} = 0$ , then equation (3.6) yields:

$$(3.7) \quad \sum_{i=0}^p \partial_t w_{i,j} \int_{\Omega_j} P_{i,j} P_{k,j} \, d\Omega_j = \int_{\Omega_j} \mathbf{f} \cdot \nabla P_{k,j} \, d\Omega_j - \int_{\Gamma_j} (\mathbf{f}^* \cdot \mathbf{n}) P_{k,j} \, d\Gamma_j.$$

The left-hand side is the temporal term multiplied by the so-called mass matrix, whereas the right-hand side corresponds to the volume and surface integrals. The last term exhibits a numerical flux function  $\mathbf{f}^*$  instead of the physical flux  $\mathbf{f}$ , due to the fact that the solution is not uniquely defined at element interfaces. In the present work, the Harten-Lax-van Leer (HLL) Riemann solver [31] is employed for the convective flux, whereas the Local Discontinuous Galerkin (LDG) method is used for the diffusive flux [7]. All integrals are computed numerically using Gauss-Lagrange quadratures. Finally, the time-integration is carried out using the explicit forth-order Runge-Kutta method. All details can be found in [9, 10]. While most DG methods rely on piecewise polynomial functions, the present work adopts bases defined from CAD for a consistent shape description, as explained in the next section.

**3.4. NURBS basis.** In most modern CAD software, shapes are defined thanks to NURBS representations, which are the rational extensions of B-Splines [27]. These functions are polynomials of degree  $p$  defined recursively, with an initialization as piecewise constant functions:

$$(3.8) \quad N_i^0(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0, & \text{otherwise,} \end{cases}$$

and by the following recursive formula:

$$(3.9) \quad N_i^p(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_i^{p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1}^{p-1}(\xi),$$

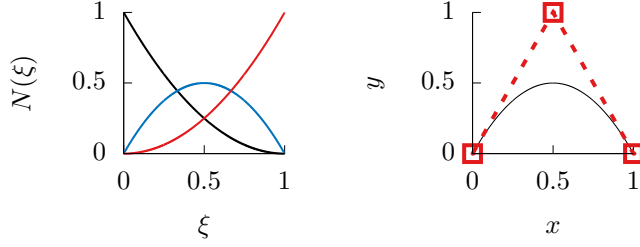


Figure 3.1: B-spline basis on left, curve with control points on right.

where  $\Xi = \{\xi_0, \dots, \xi_{n+p}\}$  is called the knot vector, defined in the parametric domain. Multivariate NURBS functions are then obtained by using tensor products of B-Spline functions multiplied by weight coefficients. For example, uni- and bi-variate NURBS functions are defined as:

$$R_i^p(\xi) = \frac{\omega_i N_i^p(\xi)}{\sum_{j=0}^{n-1} \omega_j N_j^p(\xi)} \quad R_{i_1, i_2}^p(\xi, \eta) = \frac{\omega_{i_1, i_2} N_{i_1}^p(\xi) N_{i_2}^p(\eta)}{\sum_{j_1=0}^{n-1} \sum_{j_2=0}^{n-1} \omega_{j_1, j_2} N_{j_1}^p(\xi) N_{j_2}^p(\eta)}.$$

Note that the specific case  $n = p + 1$  yields rational Bézier representations.

Curves, surfaces and volumes can then be introduced by associating coefficients  $\{\mathbf{B}_0, \dots, \mathbf{B}_{n-1}\}$ , referred as control points, to the basis functions. For instance, NURBS curves and surfaces are defined as:

$$(3.10) \quad C(\xi) = \sum_{i=0}^{n-1} R_i^p(\xi) \mathbf{B}_i \quad S(\xi, \eta) = \sum_{i_1=0}^{n-1} \sum_{i_2=0}^{n-1} R_{i_1, i_2}^p(\xi, \eta) \mathbf{B}_{i_1, i_2}.$$

This representation is illustrated by a simple example:  $p = 2$ ,  $n = 3$ ,  $\Xi = \{0, 0, 0, 1, 1, 1\}$  and  $\mathcal{B} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$ . B-Spline functions are constructed as follows  $\forall \xi \in [0, 1]$ :

$$\begin{aligned} N_0^0(\xi) &= 0, & N_1^0(\xi) &= 0, & N_2^0(\xi) &= 1, & N_3^0(\xi) &= 0, & N_4^0(\xi) &= 0, \\ N_0^1(\xi) &= 0, & N_1^1(\xi) &= (1 - \xi), & N_2^1(\xi) &= \xi, & N_3^1(\xi) &= 0, & & \\ N_0^2(\xi) &= (1 - \xi)^2, & N_1^2(\xi) &= 2(1 - \xi)\xi, & N_2^2(\xi) &= \xi^2, & & & & \end{aligned}$$

and the corresponding curve writes (we suppose here unitary weights):

$$C(\xi) = (1 - \xi)^2 \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 2(1 - \xi)\xi \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} + \xi^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \xi \\ 2(1 - \xi)\xi \end{pmatrix}.$$

In Figure (3.1), the B-spline basis is shown on the left and the curve on the right.

NURBS representations have several interesting properties, that can be found in [27]. Among them, we underline the knot insertion property, which is a key ingredient to apply these bases to a DG discretization. According to this property, one can insert a knot  $\tilde{\xi} \in [\xi_k, \xi_{k+1})$  for any  $k$ , without altering the geometry of a curve:

$$C(\xi) = \sum_{i=0}^{n-1} N_i^p(\xi) \mathbf{B}_i = \sum_{i=0}^n \tilde{N}_i^p(\xi) \tilde{\mathbf{B}}_i,$$



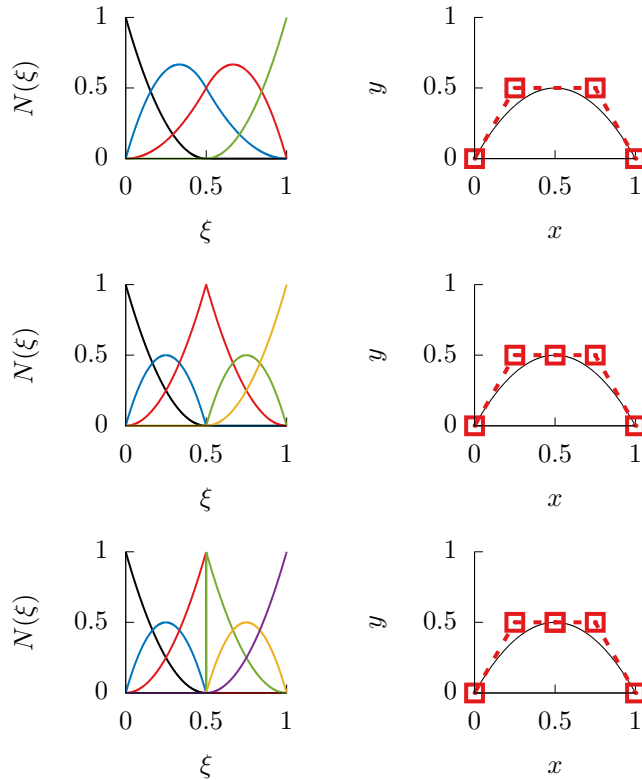


Figure 3.2: B-spline basis on left, curve with control points on right. Knot 0.5 respectively added 1 time on top, 2 times in the middle and 3 times on bottom.

where the B-spline basis  $\tilde{N}_i^p$ ,  $i \in \{0, \dots, n\}$  corresponds to the knot vector:

$$\tilde{\Xi} = \left\{ \xi_0, \dots, \xi_k, \tilde{\xi}, \xi_{k+1}, \dots, \xi_{n+p} \right\},$$

and  $\tilde{\mathbf{B}}_i = \alpha_i \mathbf{B}_i + (1 - \alpha_i) \mathbf{B}_{i-1}$  with:

$$\alpha_i = \begin{cases} 1, & \text{if } i \leq k - p, \\ \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i}, & \text{if } k - p + 1 \leq i \leq k, \\ 0, & \text{if } i \geq k + 1. \end{cases}$$

Moreover, it can be shown [27] that the curve exhibits a regularity  $\mathcal{C}^{(p-m)}$  at any knot of multiplicity  $m$ . Thus, by multiple knot insertion, one can generate a discontinuous B-spline basis, without changing the geometry of the curve  $C$ . This can be easily generalized to multi-variate NURBS. The resulting procedure is named Bézier extraction and allows to split any NURBS curve, surface or volume into a set of corresponding rational Bézier curves, surfaces or volumes without modifying the geometry. This property is illustrated on the simple example shown above, by inserting 3 times the knot 0.5, in Figure (3.2).

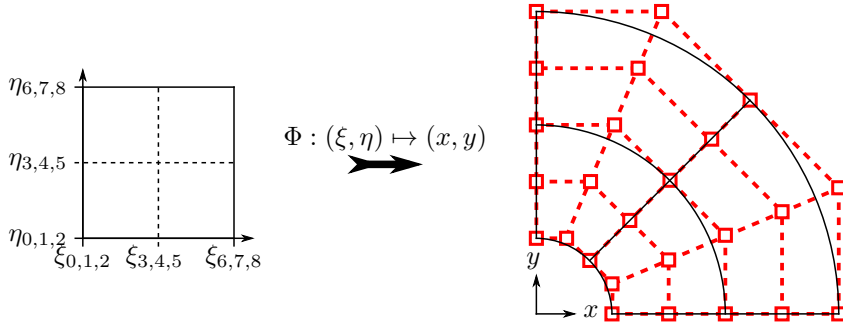


Figure 3.3: Parametric domain on left, physical domain using NURBS on right.

**3.5. Isogeometric Discontinuous Galerkin method.** This approach simply consists of employing NURBS representations, described in section 3.4, in the framework of the general DG discretization presented in section 3.3. This choice is motivated by the will to achieve CAD-consistent flow simulations. As shown in previous works [9, 10], the resulting method exhibits a higher accuracy with respect to a classical DG approach, based on piecewise linear grids, in presence of curved boundaries.

The approach is described here for two-dimensional problems, but the scheme could be extended to other dimensions. The core idea of the method is to replace the polynomial functions (like Lagrange basis) and the corresponding geometrical elements (like triangles) traditionally used in DG, by NURBS basis functions and NURBS surfaces. However, NURBS surfaces employed in CAD to represent shapes are usually too large to be considered directly as elements in a DG framework. Therefore, the proposed approach consists firstly of constructing a CAD-consistent grid, by splitting the NURBS surfaces that define originally the computational domain into a set of smaller rational Bézier surfaces, that will be considered as elements in the DG discretization. We underline again that this transformation modifies only the basis employed to describe the geometry, and not the geometry itself. From practical point of view, this phase relies on the Bézier extraction procedure explained previously, all the details being provided in [9].

This is illustrated in Figure (3.3), for a geometry corresponding to a quarter of cylinder. It is originally defined using a single bi-quadratic NURBS surface. Thanks to the Bézier extraction procedure, it is split into four rational Bézier surfaces without altering the geometry. The figure depicts the split parametric domain and the rational Bézier elements with their control points.

As a result, the computational domain is represented by a set of rational Bézier surfaces, each of them being defined by its own rational Bézier basis, which enables to generate discontinuous solutions at element interfaces. The second phase consists thus of writing the general DG formulation in equation (3.7) for a rational Bézier surface  $\Omega_j$  and its corresponding rational Bézier basis (for clarity, we omit the subscript  $j$  for the functions):

$$(3.11) \quad \sum_{i_1=0}^p \sum_{i_2=0}^p \partial_t w_{i,j} \int_{\hat{\Omega}_j} R_{i_1, i_2}^p R_{k_1, k_2}^p J d\hat{\Omega}_j = \int_{\hat{\Omega}_j} \mathbf{f} \cdot \nabla R_{k_1, k_2}^p J d\hat{\Omega}_j - \int_{\hat{\Gamma}_j} (\mathbf{f}^* \cdot \mathbf{n}) R_{k_1, k_2}^p J d\hat{\Gamma}_j.$$

Note that these integrations are achieved in the parametric domain  $\hat{\Omega}_j$ , as all quantities are now defined in this space. All other numerical ingredients implied in the DG method are maintained, only the bases being modified. We underline that the resulting approach can be referred as isogeometric, because the geometry as well as the solution are defined using the same CAD-consistent basis. In the following section, we explain how this approach can be employed for CAD-consistent sensitivity analysis.

#### 4. Shape sensitivity analysis for compressible flows.

**4.1. Sensitivity equations.** In the following, we still denote by  $'$  the derivative with respect to the parameter  $a$ . If one differentiates the Euler system (3.1), one obtains:

$$(4.1) \quad \begin{cases} \partial_t \rho' + \partial_x (\rho u)' & + \partial_y (\rho v)' & = 0, \\ \partial_t (\rho u)' + \partial_x [(\rho u)'u + (\rho u)u' + p'] & + \partial_y [(\rho u)'v + (\rho u)v'] & = 0, \\ \partial_t (\rho v)' + \partial_x [(\rho v)'u + (\rho v)u'] & + \partial_y [(\rho v)'v + (\rho v)v' + p'] & = 0, \\ \partial_t E' + \partial_x [u'(E+p) + u(E'+p')] & + \partial_y [v'(E+p) + v(E'+p')] & = 0, \end{cases}$$

with:

$$p' = (\gamma - 1) \left[ E' - \frac{1}{2} ((\rho u)'u + (\rho u)u' + (\rho v)'v + (\rho v)v') \right].$$

The conservative form established in equation (2.4) can therefore be retrieved by using the following global variables and fluxes:

$$(4.2) \quad \mathbf{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \\ \rho' \\ (\rho u)' \\ (\rho v)' \\ E' \end{pmatrix} \quad \mathbf{G}(\mathbf{W}) = \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho u v \\ \rho u v & \rho v^2 + p \\ u(E+p) & v(E+p) \\ (\rho u)' & (\rho v)' \\ (\rho u)'u + (\rho u)u' + p' & (\rho u)'v + (\rho u)v' \\ (\rho v)'u + (\rho v)u' & (\rho v)'v + (\rho v)v' + p' \\ u'(E+p) + u(E'+p') & v'(E+p) + v(E'+p') \end{pmatrix}.$$

For the Navier-Stokes equations, one should add the diffusive flux and its sensitivity which read:

$$(4.3) \quad \mathbf{G}_{\text{diff}}(\mathbf{W}) = \begin{pmatrix} 0 & 0 \\ -\tau_{xx} & -\tau_{xy} \\ -\tau_{yx} & -\tau_{yy} \\ -u\tau_{xx} - v\tau_{yx} & -u\tau_{xy} - v\tau_{yy} \\ 0 & 0 \\ -\tau_{xx}' & -\tau_{xy}' \\ -\tau_{yx}' & -\tau_{yy}' \\ -u'\tau_{xx} - v'\tau_{yx} - u\tau_{xx}' - v\tau_{yx}' & -u'\tau_{xy} - v'\tau_{yy} - u\tau_{xy}' - v\tau_{yy}' \end{pmatrix}.$$

Since the viscosity is not supposed to be parameter dependent, the sensitivity of the viscous stress tensor is simply the stress tensor applied to the sensitivity fields.

Thus, in both cases, sensitivity equations can be easily solved, simultaneously to the state equations, by using the DG discretization given in equation (3.11). The

computations are achieved component by component, with the same explicit time-marching procedure. As the state and the sensitivity equations have the same flux Jacobian, the same stability conditions hold. We underline that this global formalism facilitates significantly the extension of an existing solver to sensitivity analysis.

**4.2. Implementation of sensitivity boundary conditions.** In the context of the DG method employed here, the boundary conditions are weakly imposed, by prescribing a suitable flux on boundary interfaces. We detail below the case of wall and inlet / outlet boundary conditions for the state and sensitivity variables. In what follows, we suppose that the sensitivity parameter controls the geometry of the wall boundary. In all cases, we impose at the boundary the physical flux applied to so-called boundary state variables  $\mathbf{W}_{\text{bnd}}$ :

$$(4.4) \quad \mathbf{G} \cdot \mathbf{n}|_{\text{bnd}} = \mathbf{G}(\mathbf{W}_{\text{bnd}}) \cdot \mathbf{n},$$

with  $\mathbf{n} = (n_x, n_y)$  the unitary normal vector pointing outwards. This generic expression is then adapted to the different boundary conditions. For the wall condition with inviscid flows, the slip condition writes  $\mathbf{u} \cdot \mathbf{n} = 0$ , whereas the corresponding condition for the sensitivity is obtained by differentiating this expression:

$$(4.5) \quad (\mathbf{u} \cdot \mathbf{n})' = (\mathbf{u}' + \nabla \mathbf{u} \cdot \mathbf{x}') \cdot \mathbf{n} + \mathbf{u} \cdot \mathbf{n}' = 0,$$

where  $\mathbf{x}' = \partial_a \mathbf{x}$  represents the derivative of the boundary coordinates with respect to the shape parameter. In the context of the isogeometric DG, this implies the derivative of the boundary curve:

$$(4.6) \quad \mathbf{x}' = \partial_a C(\xi) = \sum_{i=0}^{n-1} R_i^p(\xi) \partial_a \mathbf{B}_i,$$

where  $\partial_a \mathbf{B}_i$  is the derivative of the control point of index  $i$  with respect to the parameter  $a$ . This quantity is obviously case dependent and its estimation will be detailed in forthcoming application sections. As explained in section 2.3 and illustrated in Eq. (4.5), the gradient of the velocity appears during the differentiation because the condition is written at the boundary location, which depends on the parameter  $a$ . Finally, the global boundary state  $\mathbf{W}_{\text{bnd}}$  used to compute the flux is obtained by correcting the normal component of the velocity and its sensitivity according to:

$$(4.7) \quad \mathbf{u}_{\text{bnd}} = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n}) \mathbf{n} \quad \mathbf{u}'_{\text{bnd}} = \mathbf{u}' - ((\mathbf{u}' + \nabla \mathbf{u} \cdot \mathbf{x}') \cdot \mathbf{n} + \mathbf{u} \cdot \mathbf{n}') \mathbf{n}.$$

Similarly, for the wall condition with viscous flows, the no-slip condition writes  $\mathbf{u} = \mathbf{0}$ , whereas the corresponding condition for the sensitivity is  $\mathbf{u}' + \nabla \mathbf{u} \cdot \mathbf{x}' = \mathbf{0}$ . Thus, the global boundary state  $\mathbf{W}_{\text{bnd}}$  is here obtained by modifying all the components of the velocity and its sensitivity:

$$(4.8) \quad \mathbf{u}_{\text{bnd}} = \mathbf{0} \quad \mathbf{u}'_{\text{bnd}} = -\nabla \mathbf{u} \cdot \mathbf{x}'.$$

As explained in introduction, the accuracy of the velocity gradient is critical to impose a correct boundary condition for the sensitivity. Contrary to some previous studies [26, 11], we simply rely here on the local evaluation of the solution gradient in each element. Thus, at any boundary point of parametric coordinates  $(\xi_{\text{bnd}}, \eta_{\text{bnd}})$ , the gradient reads:

$$(4.9) \quad \nabla \mathbf{u}(\xi_{\text{bnd}}, \eta_{\text{bnd}}) = \sum_{i_1=0}^{n-1} \sum_{i_2=0}^{n-1} \nabla R_{i_1, i_2}^p(\xi_{\text{bnd}}, \eta_{\text{bnd}}) \mathbf{U}_{i_1, i_2},$$

where  $\mathbf{U}_{i_1, i_2}$  denote velocity control points. The proposed approach is therefore very simple to implement. The accuracy is ensured by the use of very high-order elements associated with the exact description of the geometry permitted by the isogeometric method, as will be shown in the application section.

Regarding the inlet / outlet conditions, a similar approach is carried out to derive sensitivity boundary conditions. It is however simpler, because we make the assumption that the geometry of the boundary does not depend on  $a$ . Several formulations exist for inlet / outlet conditions in the context of DG discretizations. In this work, we impose at the inlet / outlet boundaries the flow variables  $\mathbf{U}_{\text{bnd}}$ , which are computed according to the interior  $\mathbf{U}_{\text{in}}$  and the user-defined exterior  $\mathbf{U}_{\text{out}}$  states. Formally, it writes:

$$(4.10) \quad \mathbf{U}_{\text{bnd}} = \Phi(\mathbf{U}_{\text{in}}, \mathbf{U}_{\text{out}}),$$

The operator  $\Phi$  defines how the boundary states result from a mix of interior and exterior quantities. The sensitivity variables at inlet / outlet boundaries  $\mathbf{U}'_{\text{bnd}}$  are directly obtained by differentiating the previous expression:

$$(4.11) \quad \mathbf{U}'_{\text{bnd}} = \partial_{\mathbf{U}_{\text{in}}} \Phi(\mathbf{U}_{\text{in}}, \mathbf{U}_{\text{out}}) \mathbf{U}'_{\text{in}} + \partial_{\mathbf{U}_{\text{out}}} \Phi(\mathbf{U}_{\text{in}}, \mathbf{U}_{\text{out}}) \mathbf{U}'_{\text{out}}.$$

To be more practical, we consider below a simple case: at inlet, one imposes the density  $\rho_{\text{out}}$  and velocity vector  $\mathbf{u}_{\text{out}}$ , whereas the pressure is obtained from the interior domain  $p_{\text{in}}$ . The global boundary state used to compute the flux is therefore:

$$(4.12) \quad \mathbf{W}_{\text{inlet}} = \begin{pmatrix} \rho_{\text{out}} \\ \rho_{\text{out}} u_{\text{out}} \\ \rho_{\text{out}} v_{\text{out}} \\ \frac{p_{\text{in}}}{\gamma-1} + \frac{1}{2} \rho_{\text{out}} (u_{\text{out}}^2 + v_{\text{out}}^2) \\ 0 \\ 0 \\ 0 \\ \frac{p'_{\text{in}}}{\gamma-1} \end{pmatrix},$$

because the exterior states do not depend on  $a$  and the term  $\mathbf{U}'_{\text{out}}$  vanishes in equation (4.11). Other types of inlet / outlet conditions can be treated with the same approach, for instance when one uses Riemann invariants to estimate boundary state variables. In the case the geometry of the inlet / outlet boundary depends on  $a$ , additional terms including the solution gradients appear in equation (4.11), expressing the fact that the location of the boundary condition depends on  $a$ , as already explained for the wall condition.

**4.3. Post-processing: sensitivity of efforts.** Finally, we describe here how to compute the sensitivity of flow efforts exerted on an obstacle. Contrary to the adjoint method, for which the efforts are part of the adjoint formulation, the computation of the efforts sensitivity is independent from the analysis and can be considered as a post-process. Nevertheless, it should be achieved carefully in the case of shape parameters, because efforts are obtained from a parameter-dependent integral along the boundary  $\Gamma_{\text{wall}}(a)$ . Thanks to the isogeometric mapping, it can however be expressed in terms of the independent parametric domain  $\hat{\Gamma}_{\text{wall}}$  according to (we restrict here to the inviscid case):

$$(4.13) \quad \mathcal{F} = \int_{\Gamma_{\text{wall}}(a)} p \mathbf{n} d\Gamma(a) = \int_{\hat{\Gamma}_{\text{wall}}} p \mathbf{n} J d\hat{\Gamma},$$

where  $J$  is the Jacobian of the boundary representation (3.10). The sensitivity of the efforts can then be obtained by differentiating this expression, accounting for the fact that the pressure is evaluated at a boundary point depending on the shape parameter, yielding:

$$(4.14) \quad \mathcal{F}' = \int_{\hat{\Gamma}_{\text{wall}}} (p' + \nabla p \cdot \mathbf{x}') \mathbf{n} J + p \mathbf{n}' J + p \mathbf{n} J' d\hat{\Gamma}.$$

As can be seen, the first term involves the Lagrangian derivative of the pressure at the wall, the second term represents the change of the wall normal and the last term the change of the integration domain (via the isogeometric mapping). Details can be found in [6, 15].

**5. Applications.** Three test-cases are considered to assess the proposed approach. First, the inviscid flow between two cylinders is studied, with a shape parameter defined as the radius of the inner cylinder. This problem allows a rigorous verification of the results because the solution for the flow and the sensitivity is known analytically. Then, one considers the inviscid flow around an airfoil, which is originally symmetric, while a shape parameter controlling its camber is introduced. Contrary to the previous case, no analytical solution is provided for this problem. The results will therefore be validated by comparison of neighbouring solutions, obtained by linear extrapolation based on sensitivities and full non-linear resolution. Finally, the potentiality of the proposed approach is demonstrated for a more complex problem, involving the unsteady viscous flow around a three-element airfoil, for which the relative positions of the elements are considered as parameters.

**5.1. Inviscid flow between two cylinders with radius change.** This problem, which has a steady analytical solution for the Euler system, is described in [16]. The flow is confined between two concentric cylinders of radii  $r_0 < r_1$ . The exact solution is given by:

$$\begin{cases} \rho = \rho_0, \\ \rho u = -\rho_0 \sin(\theta) u_\theta, \\ \rho v = \rho_0 \cos(\theta) u_\theta, \\ E = \frac{p}{\gamma - 1} + \frac{\rho_0}{2} u_\theta^2, \end{cases}$$

where

$$u_\theta(r) = \frac{0.2}{\left(\frac{r_1}{r_0} - \frac{r_0}{r_1}\right)} \left(\frac{r_1}{r} - \frac{r}{r_1}\right),$$

$$p(r) = \rho_0 \left[ 1 + \frac{0.2^2}{\left(\frac{r_1}{r_0} - \frac{r_0}{r_1}\right)^2} \left(\frac{r^2}{2r_1^2} - 2\ln(r) - \frac{r_1^2}{2r^2}\right) \right],$$

and the polar coordinates are given by  $r = \sqrt{x^2 + y^2}$  and  $\theta = \arctan\left(\frac{y}{x}\right)$ .

One defines the shape parameter as the inner radius  $a = r_0$ . The analytical solution for the sensitivity can easily be obtained by differentiating the solution for

the flow with respect to  $r_0$ , yielding:

$$\begin{cases} \rho' = 0, \\ (\rho u)' = -\rho_0 \sin(\theta) \partial_{r_0} u_\theta(r), \\ (\rho v)' = \rho_0 \cos(\theta) \partial_{r_0} u_\theta(r), \\ E' = \frac{\partial_{r_0} p(r)}{\gamma - 1} + \rho_0 u_\theta(r) \partial_{r_0} u_\theta(r), \end{cases}$$

where:

$$\begin{aligned} \partial_{r_0} u_\theta(r) &= \frac{u_\theta(r)}{\left(\frac{r_1}{r_0} - \frac{r_0}{r_1}\right)} \left(\frac{r_1}{r_0^2} + \frac{1}{r_1}\right) \\ \partial_{r_0} p(r) &= \frac{2(p(r) - \rho_0)}{\left(\frac{r_1}{r_0} - \frac{r_0}{r_1}\right)} \left(\frac{r_1}{r_0^2} + \frac{1}{r_1}\right). \end{aligned}$$

The geometry of the inner cylinder  $\mathbf{x} = (r_0 \cos(\theta), r_0 \sin(\theta))$  depends linearly on the parameter  $r_0$ , therefore the sensitivity of the boundary coordinates is simply  $\mathbf{x}' = \mathbf{x}/r_0$ .

Due to the symmetries of the problem, only a quarter of the domain is simulated. As consequence, the computational domain can be represented exactly using a single quadratic NURBS surface. As explained in section 3.5 and illustrated in Figure (3.3), a CAD-consistent rational Bézier grid is generated by multiple knot insertion.

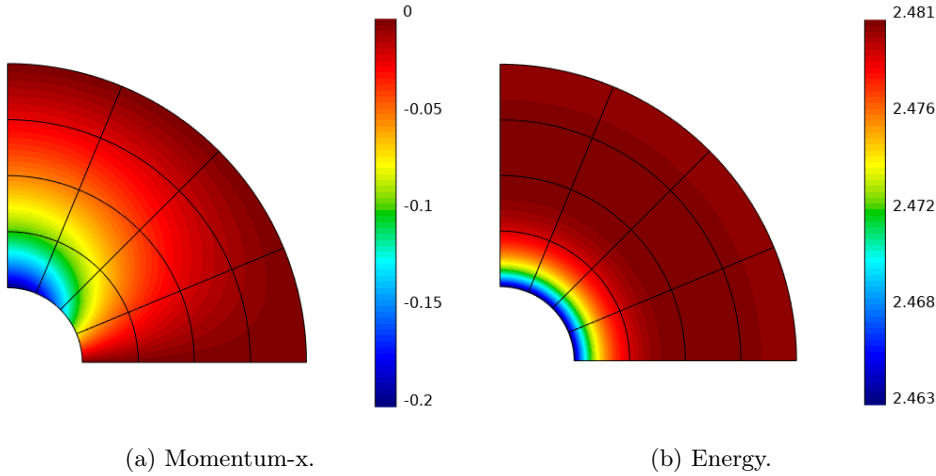


Figure 5.1: Flow solution for the cylinder test-case.

Wall boundary conditions are imposed on the two cylinders, whereas the flux corresponding to the exact solution is imposed at inlet and outlet boundaries. By splitting the computational domain in different numbers of rational Bézier surfaces, one generates a sequence of grids, from coarse to fine, which allows to carry out a rigorous assessment of the computations, for different basis degrees.

The solutions for the flow are provided in Figure (5.1), for a basis of degree five and a grid composed of four by four elements. The evolution of the error in  $L^2$ -norm

is represented in Figure (5.2) for  $\rho u$ , for degree  $p$  ranging from one to four. As can be seen, the numerical solution converges to the analytical one, with a convergence rate of value  $p + 1$ . One can notice a significant improvement between degree one and two, due to the fact that for degree two and above the geometry is exactly taken into account during the resolution, whereas for degree one the cylinder geometry is only approximated. This can also explain why the asymptotic convergence rate is more difficult to reach for the linear basis. Then, the accuracy of the sensitivity fields is similarly studied. Note that the numerical solution of the flow is used to solve the sensitivity equations. The error for  $(\rho u)'$  is depicted in Figure (5.3). One can notice that, contrary to the flow variables, a convergence rate of value  $p$  is only obtained for the sensitivity. This loss of accuracy is in fact due to the presence of the gradient term  $\nabla \mathbf{u}$  in the boundary conditions. Indeed, this quantity is computed with an accuracy of order  $p$ , by differentiation of the numerical solution of order  $p + 1$ . The sub-optimal convergence rate for the sensitivity was therefore expected. Thus, it is critical to use a high-accurate discretization method for the flow, in order to derive an accurate boundary condition for the sensitivity. This issue was studied in depth in [11, 12, 13], in the context of a finite-element method.

This test-case is an opportunity to quantify the benefit of using an isogeometric formulation, for the computation of both the flow and the shape sensitivity. In this perspective, we perform again the accuracy study, but using a sequence of (classical) piecewise linear grids generated by linearization of the grids previously used. The evolution of the error is plotted in Figures (5.4) and (5.5) for the first momentum component and its sensitivity. As can be observed, the error for the flow variables is deeply impacted by the geometrical error and the convergence rate is limited to a value lower than two, whatever the basis employed. As already observed in [9], it is due to the presence of spurious rarefaction waves, which are generated at each boundary vertex because of the artificial change of wall slope. This illustrates how the geometry can be critical when using high-order schemes. This impact is larger for the sensitivity variables, because their boundary condition at the wall involves the velocity gradients. Then, the errors in the velocity fields at the boundary propagate to the sensitivity boundary conditions. These errors are even stronger when the basis degree increases. One can conclude that the computation of accurate shape sensitivity requires a high-order treatment of the geometry.

Finally, the verification of the efforts sensitivity is presented in table (5.1). For a grid of size  $8 \times 8$  and degree  $p = 5$ , one compares the sensitivity of the horizontal force obtained using the sensitivity equation method, finite-difference formulas with different perturbation amplitudes and the analytical solution. As can be seen, the accuracy of the proposed approach is impressive, even for a quite coarse grid.

analyt.	sensit. eq.	f.d. $\delta a = 10^{-1}$	f.d. $\delta a = 10^{-2}$	f.d. $\delta a = 10^{-3}$
-1.03745	-1.03744	-1.04040	-1.03775	-1.03748

Table 5.1: Verification of efforts sensitivity



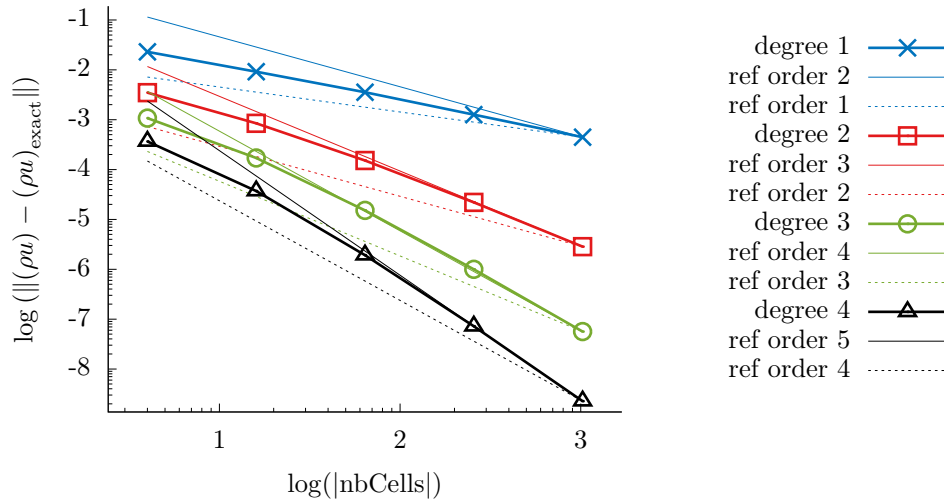


Figure 5.2: Errors for variable  $(\rho u)$  between computed and exact solutions.

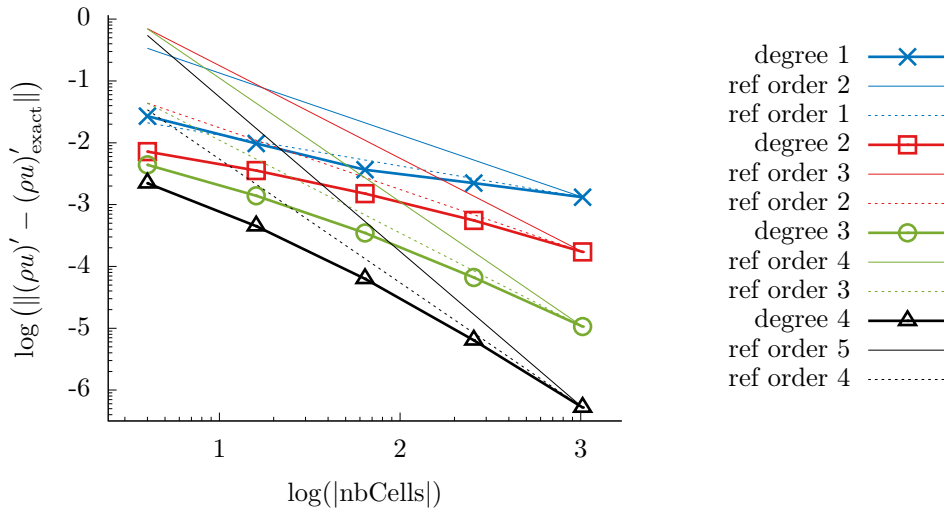


Figure 5.3: Errors for variable  $(\rho u)'$  between computed and exact solutions.

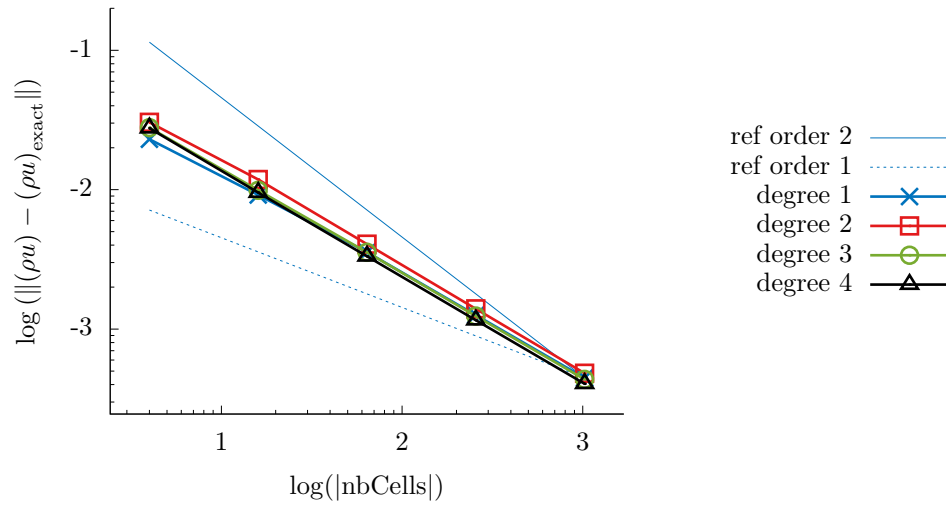


Figure 5.4: Errors for variable  $(\rho u)$  between computed and exact solutions on linear meshes.

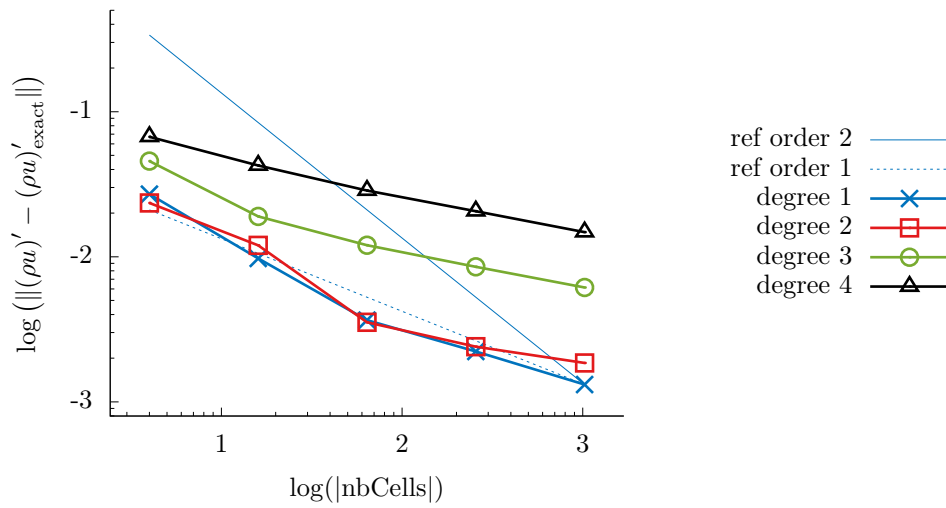


Figure 5.5: Errors for variable  $(\rho u)'$  between computed and exact solutions on linear meshes.

**5.2. Inviscid flow around an airfoil with camber change.** We consider as second test-case the flow around an airfoil in subsonic regime. The objective is to study the ability of the proposed approach to evaluate the sensitivity of the flow with respect to the camber of the airfoil. The first step is to define the airfoil parametric shape and the camber parameter in this context. In this perspective, we adopt a NURBS representation that mimics the four-digit NACA airfoils, which are defined by setting the maximum camber  $m$ , the position  $p$  of this maximum and the maximum airfoil thickness  $t$ . We recall that, for all NACA airfoils, the coordinates of the upper (resp. lower) curve  $(x_U, y_U)$  (resp.  $(x_L, y_L)$ ) are given by:

$$\begin{aligned} x_U &= x - y_t \sin \theta, & y_U &= y_c + y_t \cos \theta, \\ x_L &= x + y_t \sin \theta, & y_L &= y_c - y_t \cos \theta, \end{aligned}$$

where  $x \in [0, c]$  is the position along the chord of length  $c$ ,  $y_c$  the mean camber is defined as:

$$y_c = \begin{cases} \frac{m}{p^2} \left( 2p \left( \frac{x}{c} \right) - \left( \frac{x}{c} \right)^2 \right), & \text{if } 0 \leq x \leq pc, \\ \frac{m}{(1-p)^2} \left( (1-2p) + 2p \left( \frac{x}{c} \right) - \left( \frac{x}{c} \right)^2 \right), & \text{if } pc \leq x \leq c, \end{cases}$$

the angle  $\theta$  equals  $\arctan\left(\frac{dy_c}{dx}\right)$  and, finally,  $y_t$  is the half thickness:

$$y_t = 5t \left( 0.2969 \sqrt{\frac{x}{c}} - 0.1260 \left( \frac{x}{c} \right) - 0.3516 \left( \frac{x}{c} \right)^2 + 0.2843 \left( \frac{x}{c} \right)^3 - 0.1036 \left( \frac{x}{c} \right)^4 \right).$$

We are not using directly this definition because it does not match the NURBS representation used in this work. Nevertheless, we are considering as airfoil definition a very close approximation of the NACA 0012 airfoil ( $m = 0$ ,  $p = 0$ ,  $t = 0.12c$ ), resulting from a least-squares fitting of the lower and upper NACA curves by two cubic NURBS curves. The Figure (5.6) shows a coarse grid around the airfoil constructed by using  $16 \times 4$  cubic rational Bézier elements, obtained after splitting of the two fitted NURBS curves. The coordinates of the control points of the two NURBS curves defining the airfoil are denoted  $\mathbf{x}_U^{0012}$  and  $\mathbf{x}_L^{0012}$ . To define the shape parameter controlling the airfoil camber, we introduce a linear mapping between these two curves and the ones approximating the NACA 2412 airfoil ( $m = 0.2c$ ,  $p = 0.4c$ ,  $t = 0.12c$ ), obtained with a similar least-squares fitting procedure and characterized by a camber of value  $m = 0.2c$ . The coordinates of the control points of the two NURBS curves defining this second airfoil are denoted  $\mathbf{x}_U^{2412}$  and  $\mathbf{x}_L^{2412}$ . This allows to generate a set of NURBS-based airfoils of different camber values, whose control points have the coordinates  $\mathbf{x}_U = (1-\omega)\mathbf{x}_U^{0012} + \omega\mathbf{x}_U^{2412}$  and  $\mathbf{x}_L = (1-\omega)\mathbf{x}_L^{0012} + \omega\mathbf{x}_L^{2412}$ . The weighting coefficient  $\omega$  controls the global camber of the airfoil, as  $m$  for NACA airfoils, it is thus chosen as shape parameter for this study:  $a = \omega$ . As consequence, the sensitivities of the boundary coordinates are simply  $\mathbf{x}'_U = \mathbf{x}_U^{2412} - \mathbf{x}_U^{0012}$  and  $\mathbf{x}'_L = \mathbf{x}_L^{2412} - \mathbf{x}_L^{0012}$ .

The flow and the sensitivity with respect to  $\omega$  are computed for a subsonic regime corresponding to a far-field Mach number  $M_\infty = 0.2$ . The mesh employed is similar to the one shown in Figure (5.6), but it is more extended spatially and includes a refined area around the airfoil. It counts 6818 cubic elements. Wall conditions are imposed at the airfoil curves, whereas inlet / outlet conditions are prescribed at the outer boundary, with exterior state defined as  $\rho_{\text{out}} = \gamma$ ,  $u_{\text{out}} = M_\infty$ ,  $v_{\text{out}} = 0$  and  $p_{\text{out}} = 1$ .

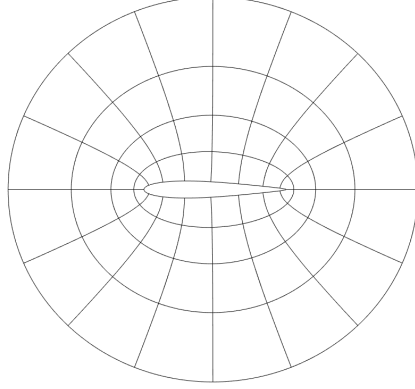


Figure 5.6: Example of cubic grid fitting the NACA 0012 airfoil.

In Figure (5.7) are plotted the density field around the airfoil and its sensitivity with respect to  $\omega$ , for the symmetric airfoil configuration corresponding to  $\omega = \omega_0 = 0$ . Since the airfoil is symmetric and has no incidence with respect to the far-field flow, the density field is also symmetric. However, the sensitivity field is asymmetrical, because the sensitivity of the boundary coordinates involved in the boundary conditions is asymmetrical.

To assess these computations, we propose to use the sensitivity to extrapolate linearly the flow fields from the original symmetric airfoil to a sequence of airfoils with non-zero camber values. This is achieved in a straightforward way by combining the discrete flow variables  $\mathbf{U}(\omega_0)$  and sensitivity variables  $\mathbf{U}'(\omega_0)$  computed for the symmetric airfoil:

$$(5.1) \quad \mathbf{U}(\delta\omega)^{\text{lin}} = \mathbf{U}(\omega_0) + \delta\omega\mathbf{U}'(\omega_0),$$

where the parameter  $\delta\omega$  represents the camber change and  $\mathbf{U}^{\text{lin}}$  the predicted fields. According to this first-order Taylor expansion, the error in the extrapolated fields should be of order  $\delta\omega^2$ . Therefore, to validate the sensitivity computation, we solve the flow equations for a sequence of different camber values and plot the extrapolation error with respect to the perturbation  $\delta\omega$  in logarithm scale, as shown in Figure 5.8 for the variables  $E$ . As can be seen, a satisfactory agreement is obtained. It should be underlined that the theoretical order  $\delta\omega^2$  is difficult to obtain in practice, because of the presence of discretization errors, which should be made negligible with respect to the extrapolation error using a very fine grid and a high-accurate scheme. Moreover, the numerical evaluation of the error is tedious, because the two solutions (extrapolated one and resolved one) are achieved on different grids. Here, the grids corresponding to the airfoils with camber are generated using local deformations of the reference grid, thanks to the geometrical mapping described above. As consequence, the error curve presented in Figure 5.8 is evaluated only in the areas of the computational domain that are identical for the two configurations. This may also explain the discrepancies between the error curves and the theoretical rates.

Finally, the potentiality of such linear extrapolations to estimate the flow for a neighboring shape at a low computational cost is demonstrated. A comparison of the density fields obtained by extrapolation from the symmetric airfoil and full resolution using an airfoil with camber is shown in Figure 5.9, for a camber of value  $\omega = 1$ .

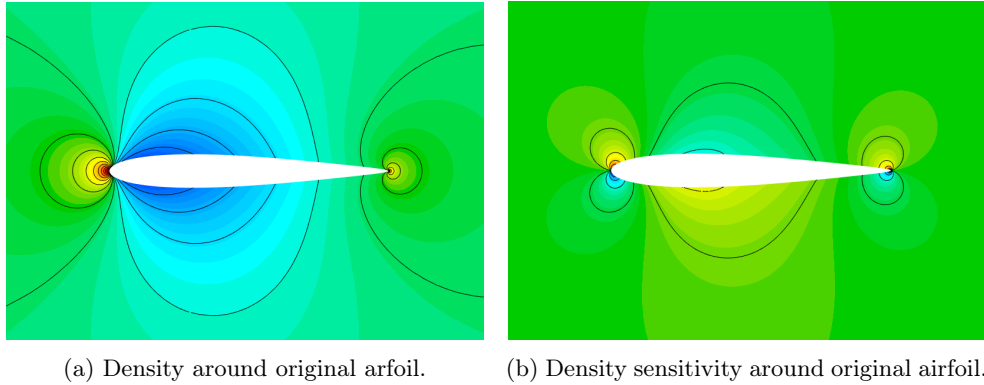


Figure 5.7: Flow and sensitivity solution for original airfoil.

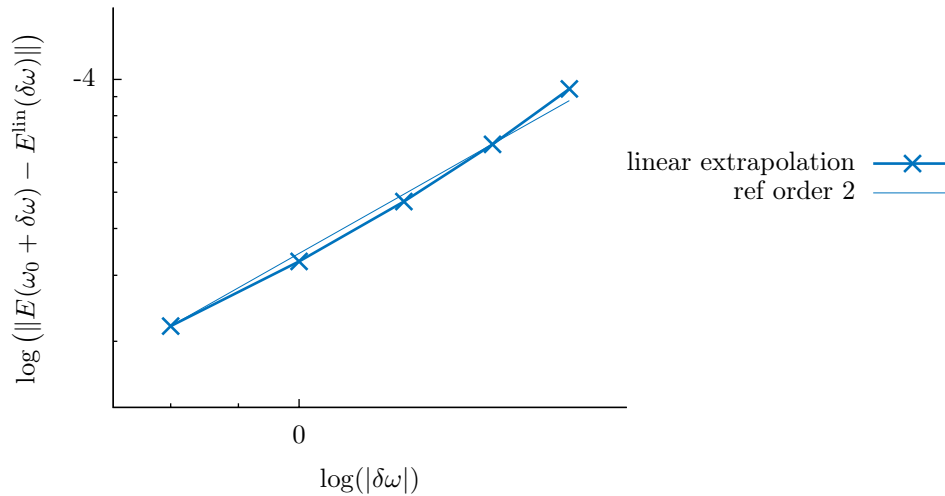


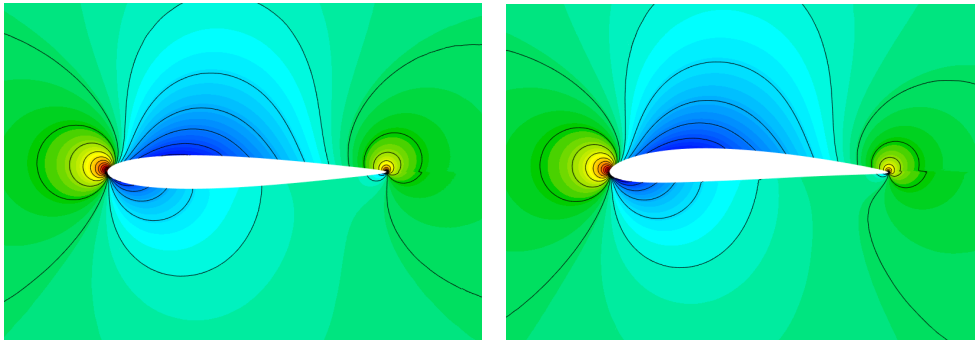
Figure 5.8: Extrapolation error for variable  $E$ .

Note that the extrapolated field is plotted for the original symmetric configuration, because it is evaluated on the corresponding computational domain. As consequence, it does not make sense at some locations close to the airfoil boundary. The flow around the airfoil with camber is directly computed using a geometrical domain that fits the airfoil. Note that, for visualization purpose, the extrapolated fields could be projected onto the mesh corresponding to the airfoil with camber, or possibly transported using local gradient data in the case of mesh deformation [15]. As can be seen, the camber yields a significant change in the solution field, which is well predicted by the linear extrapolation. Although some discrepancies can be found, it shows that an accurate computation of the shape sensitivity allows to predict the global characteristics of the flow for a quite important shape perturbation. Beyond the prediction of the flow fields, it is also interesting to predict the evolution of the efforts exerted by the flow on the

airfoil. We focus here on the lift, which has a zero value for the reference symmetric shape and increases with the camber of the airfoil. The lift sensitivity value is provided in Table 5.2 for two configurations: the symmetric case  $\omega = 0$  and a cambered case corresponding to  $\omega = 0.5$ . In both cases, the lift sensitivities obtained by solving the sensitivity equations are compared to those estimated by finite-differences, based on two non-linear simulations, with different perturbation amplitudes. As can be seen, a satisfactory accuracy is observed, in particular for the configuration  $\omega = 0$  which may benefit from the symmetry of the grid. As a consequence, the linear extrapolation of the lift from  $\omega = 0$  to  $\omega = 1$  (case of the Figure 5.9) yields an error of 3%, which seems reasonable given the magnitude of the shape change.

	sensit. eq.	finite differences			
		$\delta\omega = 10^0$	$\delta\omega = 10^{-1}$	$\delta\omega = 10^{-2}$	$\delta\omega = 10^{-3}$
$\omega = 0$	0.001274	0.001313	0.001274	0.001273	0.001273
$\omega = 0.5$	0.001459	0.001698	0.001548	0.001442	0.001442

Table 5.2: Lift sensitivity



(a) Extrapolation of the density field. (b) Density around an airfoil with camber.

Figure 5.9: Comparison of extrapolated and solved density fields.

### 5.3. Viscous flow around a three-element airfoil with position change.

After a verification of the developed method and a validation for a quite simple configuration, the last application aims at providing a more complex illustration of the potentiality of the proposed approach, in terms of geometry and physics. We consider the unsteady viscous flow around a three-element airfoil (slat, main foil, flap), with the positions of slat and flap as shape parameters.

The geometry of each foil, as depicted in Figure (5.10), is composed of two cubic NURBS curves, one for the upper side and one for the lower side. A baseline grid is obtained by splitting these curves into rational Bézier ones and constructing a set of elements based on them. This process is illustrated in Figure (5.11). To obtain a suitable computational grid, some elements are split again several times (local h-refinement) and their degree is elevated to quartic (global p-refinement), as shown in Figure (5.12). Remember that, during all this process, the boundary geometry

remains identical. This illustrates the flexibility offered by the NURBS representation. Finally the mesh is rotated, yielding an incidence angle of  $3^\circ$ . The grid counts 2480 quartic elements and 62000 degrees of freedom. The flow is then simulated, for a Reynolds number  $R_\infty = 3000$  and a Mach number  $M_\infty = 0.3$ . Note that the flow conditions considered here are not fully representative, because of the low Reynolds number, but this case illustrates anyway the potentiality of the sensitivity analysis. The flow is characterized by a periodic vortex shedding occurring at the flap, due to the acceleration of the flow between the flap and the main foil. The first component of the momentum is depicted at non-dimensional time  $t = 50$  in Figure (5.13).

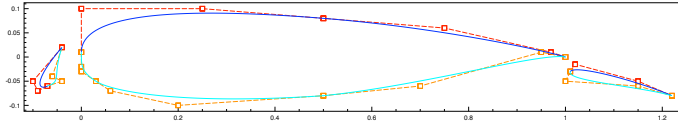


Figure 5.10: Geometry of the three-element airfoil (curves and control points polygons).

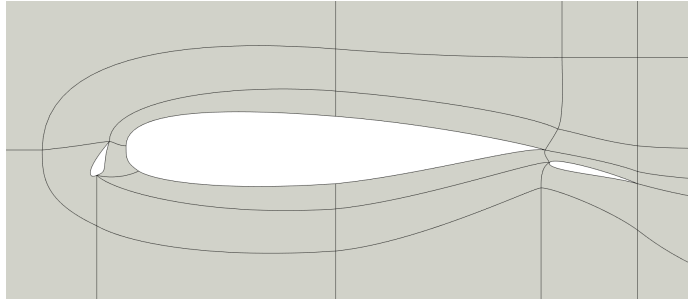


Figure 5.11: Baseline mesh.

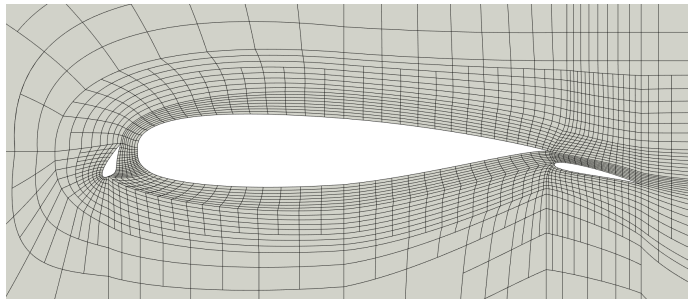


Figure 5.12: Locally refined computational mesh.

We consider now two shape parameters, defined as the vertical position of the slat and flap. We choose them because they represent critical parameters for the design of high-lift devices in aerodynamics. For these parameters, the sensitivities of

the boundary coordinates are simply  $\mathbf{x}' = (0, 1)$  for the slat or flap, respectively. We examine first the results obtained for the sensitivity with respect to the flap position. The sensitivity of the first momentum is depicted in Figure (5.14). As can be observed, the sensitivity analysis predicts that a vertical displacement of the flap yields a change of the near wall velocity. This also impacts the vortex generation, due to a change of the flow between the two foils. One can also notice some small perturbations in the vicinity of the slat, because of the presence of upstreaming pressure waves, but these are negligible. Now, looking at the sensitivity of the first momentum with respect to the slat position, as shown in Figure (5.15), a different situation is reported. As for the flap, the velocity field in the near wall region is obviously influenced by the slat location. But one can also observe a propagation downstream that finally impacts the vortex generation. Although this influence is lower than that observed for the flap location, it was not expected and illustrates the interest of the sensitivity analysis to study the flow dependency in such complex configurations.

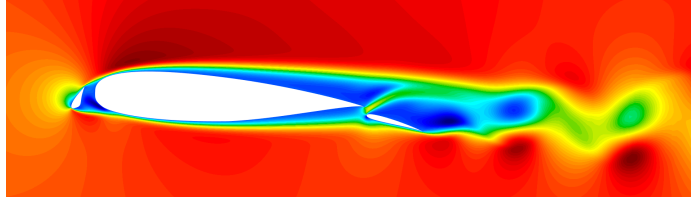


Figure 5.13: First momentum field  $\rho u$  (range [0.15; 0.6]).

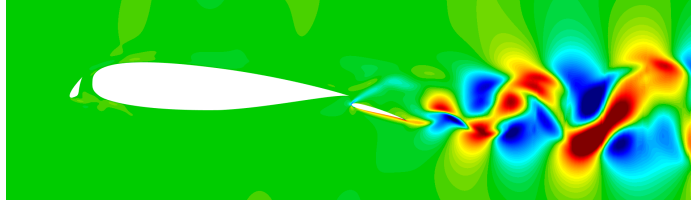


Figure 5.14: Field  $(\rho u)'$  with respect to the flap position (range [-60; 60]).

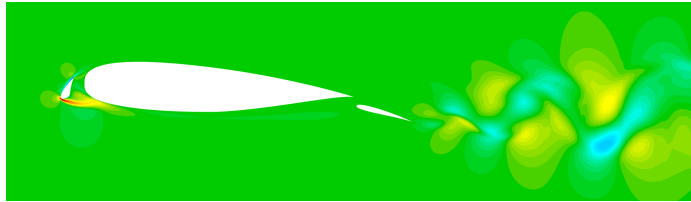


Figure 5.15: Field  $(\rho u)'$  with respect to the slat position (range [-60; 60]).



**6. Conclusion.** The objective of this work was to investigate the shape sensitivity analysis in the framework of an isogeometric DG method. After a summary of the NURBS-based discretization technique was provided, the extension to the resolution of the continuous sensitivity equations for the compressible Euler/Navier-Stokes system was detailed. A particular emphasis was given to the description of the boundary conditions in the case of shape parameters. The resulting approach allows to solve in a compact form both the flow and sensitivity equations with a high-order CAD-consistent methodology.

A verification exercise was then carried out, for a test-case concerning the flow between two cylinders. It was demonstrated that a convergence rate of value  $p + 1$  was obtained for the flow variables, whereas only a rate of value  $p$  was observed for sensitivity variables, due to the presence of flow gradients in the sensitivity boundary conditions. The critical impact of the use of high-order geometrical representations was reported. Then, two more complex problems were studied, regarding the inviscid flow around an airfoil with camber change and the viscous flow around a three-element airfoil with flap and slat position change. It was shown that accurate shape sensitivity computations allow to predict the flow modifications, as well as efforts, by linear extrapolation. Finally, we illustrated the potentiality of sensitivity analysis to study non-trivial dependencies occurring in complex flows.

Several issues remain to use such an approach for industrial applications. In particular, the difficulty to account for discontinuous solutions like shocks in the framework of the sensitivity equation method is well known. A possible approach was found in [14] for one-dimensional problems. Its extension to higher dimension is currently investigated.

**Code and data.** The source code and data used in this work are freely distributed, under the GNU General Public Licence v3, at the following repository: <https://gitlab.inria.fr/igloo/igloo/-/wikis/home>.

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

#### REFERENCES

- [1] C. BARDOS AND O. PIRONNEAU, *A formalism for the differentiation of conservation laws*, *Compte rendu de l’Académie des Sciences*, 335 (2002), pp. 839–845.
- [2] O. BAYSAL AND G. W. BURGEEEN, *Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis*, *AIAA Journal*, 34 (1996), pp. 1761–1770.
- [3] J. BORGGAARD AND J. BURNS, *A PDE sensitivity equation method for optimal aerodynamic design*, *Journal of Computational Physics*, 136 (1997), pp. 366–384.
- [4] J. BORGGAARD AND A. VERMA, *On efficient solutions to the continuous sensitivity equation using automatic differentiation*, *SIAM Journal on Scientific Computing*, 22 (2000), pp. 39–62.
- [5] A. BORZI AND V. SCHULZ, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2012.
- [6] L. CHARLOT, S. ETIENNE, AND D. PELLETIER, *A continuous lagrangian sensitivity equation method for incompressible flow*, *Journal of Computational Physics*, 231 (2012).
- [7] B. COCKBURN AND C.-W. SHU, *The local discontinuous galerkin method for time-dependent convection-diffusion systems*, *SIAM Journal of Num. An.*, 35 (1998), pp. 2440–2463.
- [8] A. DE BOER, M. VAN DER SCHOOT, AND H. BIJL, *Mesh deformation based on radial basis function interpolation*, *Computers & Structures*, 85 (2007), pp. 784–795.
- [9] R. DUVIGNEAU, *Isogeometric analysis for compressible flows using a Discontinuous Galerkin method*, *Computer Methods in Applied Mechanics and Engineering*, 333 (2018), pp. 443–461.

- [10] R. DUVIGNEAU, *CAD-consistent adaptive refinement using a NURBS-based Discontinuous Galerkin method*, International Journal for Numerical Methods in Fluids, (2020).
- [11] R. DUVIGNEAU AND D. PELLETIER, *On accurate boundary conditions for a shape sensitivity equation method*, Int. J. for Numerical Methods in Fluids, 50 (2006).
- [12] R. DUVIGNEAU AND D. PELLETIER, *A sensitivity equation method for fast evaluation of nearby flows and uncertainty analysis for shape parameters*, International Journal of Computational Fluid Dynamics, 20 (2006), pp. 497–512.
- [13] R. DUVIGNEAU, D. PELLETIER, AND J. BORGGAARD, *An improved continuous sensitivity equation method for optimal shape design in mixed convection*, Numerical Heat Transfer part B : Fundamentals, 50 (2006), pp. 1–24.
- [14] C. FIORINI, C. CHALONS, AND R. DUVIGNEAU, *A modified sensitivity equation method for the euler equations in presence of shocks*, Numerical Methods for Partial Differential Equations, (2019).
- [15] A. HAY, S. ETIENNES, R. DUVIGNEAU, AND D. PELLETIER, *Evaluation of flows on nearby geometries by a shape sensitivity equation method*, in 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 2006.
- [16] J. S. HESTHAVEN AND T. WARBURTON, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, Springer Science & Business Media, 2007.
- [17] T. J. HUGHES, J. A. COTTRELL, AND Y. BAZILEVS, *Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement*, Computer methods in applied mechanics and engineering, 194 (2005), pp. 4135–4195.
- [18] B. IOOSS AND P. LEMAITRE, *Uncertainty management in Simulation-Optimization of Complex Systems: Algorithms and Applications*, Springer, 2015, ch. A Review on Global Sensitivity Analysis Methods, pp. 101–122.
- [19] A. JAMESON, *Aerodynamic design via control theory*, Journal of Scientific Computing, 3 (1988), pp. 233–260.
- [20] A. JAMESON, L. MARTINELLI, AND N. A. PIERCE, *Optimum aerodynamic design using the Navier-Stokes equation*, Theoretical and Computational Fluid Dynamics, 10 (1998), pp. 213–237.
- [21] A. KEANE AND P. NAIR, *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, John-Wiley and Sons, 2005.
- [22] M. KLEIBER, T. D. HIEN, H. ANTÚNEZ, AND P. KOWALCZYK, *Parameter sensitivity in nonlinear mechanics: Theory and finite element computations*, Wiley, 1997.
- [23] V. KOMKOV, K. K. CHOI, AND E. J. HAUG, *Design sensitivity analysis of structural systems*, vol. 177, Academic press, 1986.
- [24] C. MICHOSKI, J. CHAN, L. ENGVALL, AND J. A. EVANS, *Foundations of the blended isogeometric discontinuous Galerkin (BIDG) method*, Computer Methods in Applied Mechanics and Engineering, 305 (2016), pp. 658–681.
- [25] B. MOHAMMADI AND O. PIRONEAU, *Mesh adaptation and automatic differentiation in a cad-free framework for optimal shape*, International Journal for Numerical Methods in Engineering, 30 (1999), pp. 127–136.
- [26] D. PELLETIER, É. TURGEON, S. ETIENNE, AND J. BORGGAARD, *Reliable sensitivity analysis via an adaptive sensitivity equation method*, in 3rd AIAA Theoretical Fluid Mechanics Meeting, Saint Louis, MO, 2002.
- [27] L. PIEGL AND W. TILLER, *The NURBS book*, Springer Science & Business Media, 2012.
- [28] O. PIRONEAU, *Optimal shape design for elliptic systems*, Springer-Verlag, 1983.
- [29] L. L. SHERMAN, A. C. TAYLOR III, L. L. GREEN, P. A. NEWMAN, G. W. HOU, AND V. M. KORIVI, *First-and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods*, Journal of Computational Physics, 129 (1996), pp. 307–331.
- [30] A. SILVEIRA, R. MOURA, A. SILVA, AND M. ORTEGA, *Higher-order surface treatment for discontinuous Galerkin methods with applications to aerodynamics*, International Journal for Numerical Methods in Fluids, 79 (2015), pp. 323–342.
- [31] E. TORO, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer Berlin Heidelberg, 1997.
- [32] É. TURGEON, D. PELLETIER, AND J. BORGGAARD, *Application of a sensitivity equation method to the  $k - \epsilon$  model of turbulence*, in 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, Jun. 2001.
- [33] É. TURGEON, D. PELLETIER, AND J. BORGGAARD, *Sensitivity and uncertainty analysis for variable property flows*, in 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 2001.
- [34] S. YU, R. FENG, AND T. LIU, *An isogeometric discontinuous galerkin method for euler equa-*

- tions, *Mathematical Methods and Models in Applied Sciences*, 40 (2017).
- [35] O. C. ZIENKIEWICZ AND J. Z. ZHU, *The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique*, *International Journal for Numerical Methods in Engineering*, 33 (1992), pp. 1331–1364.