Model Order Reduction for Hyperbolic Problems: a new framework

N Cagniart, R Crisovan, Yvon Maday, R Abgrall

Model Order Reduction for Hyperbolic Problems: a new framework

N. Cagniart* R. Crisovan† Y. Maday* R. Abgrall†

August 25, 2017

Abstract

We are interested in adapting standard model reduction techniques to hyperbolic problems. More precisely, this paper proposes a calibration procedure that allows to use standard ROM techniques to solve the two dimensional Euler equation around an airfoil. We propose a complete framework. First, an offline calibration procedure that reduces the Kolmogorov n-width of the solution manifold studied. Then a cheap reduced scheme approximating the truth solver. It uses L1-norm minimization and uses the calibrated manifold constructed in the offline phase. We discuss it's computational complexity. Finally, we present numerical simulations that illustrate the overall feasability of the method.

Keywords Reduced Order Modeling, Hyperbolic problems, Calibration, L^1-norm minimization, Gordon-Hall, Hyper-Reduction

1 Introduction

Fast reliable solutions to many queries parametric Partial Differential Equations (PDE) have many applications among which real time systems, optimization problems and optimal control. Many different methods for reducing the complexity of the computations when such many queries are required have blossomed for answering this specific need. One of the approaches that have emerged is reduced order modeling (ROM). Methods in this category have been developed and are now well understood and set on firm grounds, both for steady cases or time dependent problems where time can be considered as another parameter.

The reduced basis method, which is the method that we focus in on this paper, enters in this frame and consists in: i) defining a sequence of low dimensional spaces for the approximation of the whole set of the solutions to the parametric PDE when the parameters vary (called hereafter the solution manifold associated to our problem); ii) once such a sequence of low dimensional spaces (known as reduced basis spaces) is determined, an approximate solution is sought in such a chosen reduced space to the PDE for the values of the parameter we are interested in. The approximation is often based on a Galerkin formulation. For such reduced basis methods, both the variety of applications and the theory are now quite sound. For instance, reliable algorithms with a priori estimates and certified a posteriori errors have been developed for elliptic and parabolic problems,

*University Pierre and Marie Curie - Paris 6, Laboratoire Jacques-Louis Lions (LJLL)
†University of Zürich, Institut für Mathematik (I-MATH)

with or without the so-called affine parameter dependence, see e.g.the two recent books on the subject [1] and [2] and, of course, the publications therein.

Reduced basis methods, classically, consider the solution manifold associated to the parametrized problem as outlined above and are appropriate if this manifold can be approximated accurately by a sequence of finite dimensional spaces. The mathematical frame for this is inherently linked to the notion of *Kolmogorov width* of solution manifolds, i.e. how well the solution manifold can be approached by a finite dimensional linear space. More precisely, let $\mathcal{M}$ be a manifold embedded in some normed linear space X. The Kolmogorov n-width of $\mathcal{M}$ is defined as:

$$d_n(\mathcal{M}, X) = \inf_{E_n} \sup_{f \in \mathcal{M}} \inf_{g \in E_n} \|f - g\|_X, \tag{1}$$

the first infimum being taken over all linear subspaces E_n of dimension n embedded in X.

Even if, from the practical point of view, there are various ways for checking that $\mathcal{M}$ can be approximated by a series of reduced spaced with small dimension, the first natural mathematical question is to provide an estimation of the Kolmogorov n-width of $\mathcal{M}$. Second, the question of an applied mathematician is if one can actually build an optimal, or close to optimal sequence of basis sets for these spaces?

Of course, in the vast majority of real cases, there is no analytical expression for this dimension but there are some papers giving bounds for some restricted classes of problems in the literature. For instance, in [3], bounds on d_n are found for solution manifolds corresponding to regular elliptic problems and where the parameter dependence is on the forcing term. More general cases can be handled using the results in [4]. The hypothesis therein is on the regularity of the solution with respect to the parameter dependence and it is proven that, under analyticity assumption on the behavior of the parameters in the PDE, the small Kolmogorov n-width of the manifold of parameters $\mathcal{D}$ ($\leq cn^{-t}$, $t > 1$) implies the smallness of the Kolmogorov n-width of the associated solutions manifold $\mathcal{M}_{\mathcal{D}}$ ($\leq cn^{-s}$, $s \leq t - 1$).

In practice, instead of the "optimal" linear subspace of dimension n in the sense described earlier, we build a "good" linear subspace. In the literature, the two most classical algorithms are the greedy method based on a certified (or at least fair enough) a posteriori estimator, and the Proper Orthogonal Decomposition (POD). We proceed assuming that the chosen algorithm has given a "good" basis "close" to the optimal one, that is, we assume that our reduced family of spaces $\{X_n\}_n$ satisfies:

$$d_n(\mathcal{M}, X) \approx \sup_{f \in \mathcal{M}} \inf_{g \in X_n} \|f - g\|_X \tag{2}$$

A first paper on this subject is [5], where the authors derived error bounds of the error for the Reduced Basis Method (RBM) approximation in case of a single parameter dependent elliptic PDE. More general results have been obtained more recently for the greedy approach of the RBM [6, 7]. The optimality considered in the case of POD is slightly different. The POD focuses on minimizing the average error (parameter wise), in some norm. More precisely, we have the well known relation

$$\int_{\mathcal{D}} \|u(\mu) - \Pi_{POD} u(\mu)\|^2 \, d\mu = \sum_{i > N_{POD}} \lambda_i, \tag{3}$$

where Π_{POD} is the orthogonal projection onto the POD reduced space of dimension N_{POD} and the λ_i's are the eigenvalues of the associated correlation operator, in decreasing order. The faster the decay of the eigenvalues, the fewer modes are needed for a good (in average) reconstruction of the solution manifold.

Up to now, most of the literature on the subject, deals with problems where one can expect/check/prove/ or hope, that the solution manifold $\mathcal{M}_{\mathcal{D}}$ has a small Kolmogorov n-width. There are however cases where the plain approach does not work and some transformation of $\mathcal{M}_{\mathcal{D}}$ needs to be done.

An example is for instance the use of the Piola transform in the processing of the velocity field when the PDE is the Stokes or Navier Stokes problem and the parameter includes the geometry of the computational problem (see e.g. [8]). The choice of the Piola transform indeed provides better reduction than a simple change of variables.

A lot of convection dominated problems also suffer the same characteristics. A freezing method, dealing with this issue was developed in [9, 10], and later adapted in [11]. Both methods rely on the notion of calibration. That is, they start with a "preconditioning" step where they target a family of (smooth) invertible mappings

$$\mathcal{F}_{\mathcal{D}} = \left\{ F : \overline{\Omega} \mapsto \overline{\Omega} \right\}, \tag{4}$$

in which there exists well chosen applications

$$\begin{array}{ccccc} [0,T] & \times & \mathcal{D} & \rightarrow & \mathcal{F}_{\mathcal{D}} \\ (t,\mu) & & & \mapsto & F_{t;\mu} \end{array} \tag{5}$$

such that the corresponding preconditioned solution manifold, defined as:

$$\mathcal{M}_{\mathcal{F},\mathcal{D}} := \left\{ u(F_{t;\mu}^{-1}(\cdot), t; \mu),\ \mu \in \mathcal{D},\ t \in [0,T] \right\} \tag{6}$$

has a smaller Kolmogorov n-width than $\mathcal{M}_{\mathcal{D}}$.

Behind this abstract formulation is the generalization of a simple idea. For periodic convection dominated problems, translation is a trivial calibration. In [11] is developed a full online reduced scheme.

The objective of this paper is to apply this abstract calibration idea to more realistic problems than the one dimensional Burgers equation. We have decided to focus on the steady two dimensional Euler equation around an airfoil. The precise setting will be discussed in Section 2. To motivate the calibration idea in this specific setting, we show in Figure 1 the shock positions for various pair of parameters: Mach number and angle of attacks (AoA). It is obvious that this example suffers the same problem as the one dimensional Burgers case. Because of the moving shock, the Kolmogorov n-width of the raw data set $\mathcal{M}_{\mathcal{D}}$ will not have the good decay properties required for standard ROM. We thus need a preconditioning step, and will propose a calibration.

The choice in this note is to follow the steps of [11], for this 2D hyperbolic problem. That is, we want to:

- calibrate the offline computed solution, to get a reduced basis as small as possible;
- have an online scheme that builds a "calibrated problem", which can use the calibrated reduced basis.

We highlight in here the differences:

- they were solving Burgers' equation in 1D with one propagating front i.e, there was only one calibration parameter. The shock's position shape might require more calibration parameters;

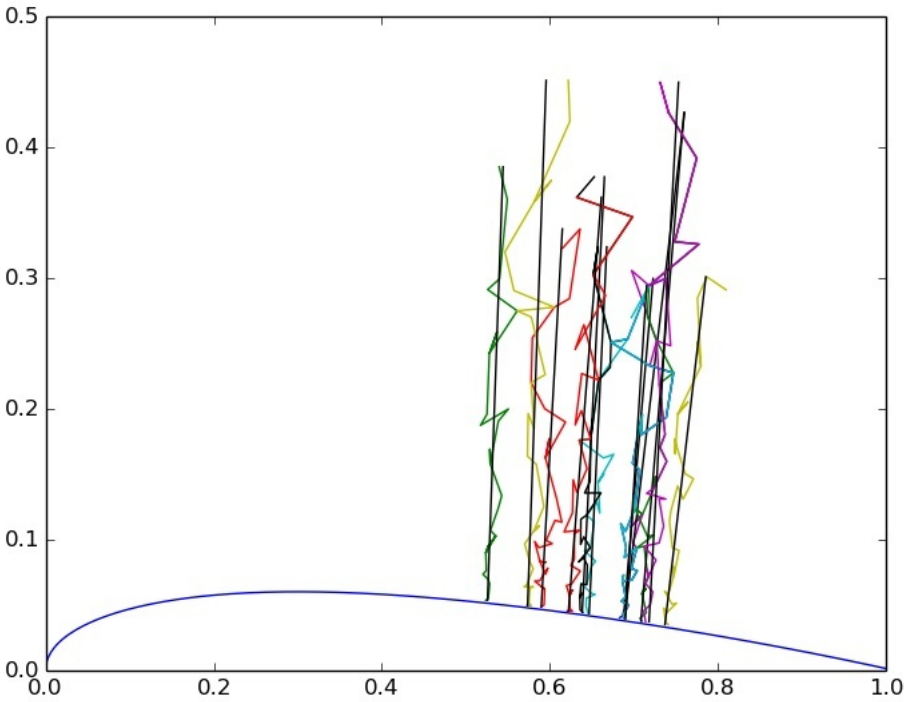


Figure 1: Position of the shock for various AoA and Mach numbers

- they use periodic boundary conditions, so no matching with exterior domain was needed. As a result, the calibrated problem was just a translated version of the initial problem;
- their problem was a parabolic problem. We will see that dealing with discontinuous solutions adds complexity to the online section;
- the fine computations were done with an upwind finite volume scheme, but a raw Galerkin reduced scheme gave decent results. We have no guarantee in here that a reduced scheme with no stabilization nor upwinding will work. We will discuss these issues in the next sections.

In the first section of this paper, we will completely described the problem we want to solve. We will give details on the 'truth' scheme we are using. In the second section, we describe our choice of family of mappings $\mathcal{F}$, as well as one possible choice for $\mu \to F_\mu$. We use this to perform the 'offline phase'. We will make sure that the calibration reduces the Kolmogorov n-width of the solution manifold. In the third section, we propose a cheap 'online' algorithm. This is the central part of this paper, as most related work simply perform the offline calibration, and do not propose any numerical scheme actually using the calibrated manifold $\mathcal{M}_{\mathcal{F},\mathcal{D}}$ (see [12, 13, 14]). We will also discuss the methods described in the online phase in terms of $L1$ minimization as it was advised

in [15] and we will use ideas close to Hyper-Reduction. In Section 5, we propose an optimization procedure in order to derive the minimization coefficients and the relative mapping. The final section is devoted to numerical experiments where we present different mappings and we show the importance of the smoothness of the mappings in $\mathcal{F}$. In the end, we will conclude our paper and we will present some ideas that can be further investigated and implemented.

2 Problem setting

2.1 Naca0012 test case

We have chosen to perform our calibration ideas on the following well documented external flow test-case: the two-dimensional, inviscid, transonic flow past the NACA 0012 airfoil. The explicit form of the wing is given as:

$$y = w(x) := 0.6 \cdot \Big(0.2969 \cdot \sqrt{x} - 0.1260 \cdot x - 0.3516 \cdot x^2 + 0.2843 \cdot x^3 - 0.1015 \cdot x^4\Big). \tag{7}$$

We are using subsonic boundary conditions on the outside boundary and slip boundary conditions on the wing. The last Neumann type boundary condition imposes that the velocity of the fluid is tangent to the wing. It is commonly known, that from a certain threshold of Mach number, a shock appears. The position and the form of the shock depends on many parameters among which the Mach number and the angle of attack (AoA), i.e the inflow mean direction.

2.2 2 dimensional Euler equation

We are interested in the numerical approximation of the two dimensional Euler equations. Let Ω some domain around the airfoil described in the previous section, W the state vector of conserved variables and $f = (f_x, f_y)$ the flux:

$$\begin{aligned} W =& (\rho, \rho u, \rho v, E)^T \\ f_x(W) =& (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T \\ f_y(W) =& (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T, \end{aligned}$$

ρ is the density, u and v are the components of the velocity, $E = \rho\epsilon + \frac{1}{2}\rho(u^2 + v^2)$ is the total energy and ϵ is the specific internal energy. The system is closed by the equation of state relating the pressure p to the conserved variables:

$$p = (\gamma - 1)\big(E - \frac{1}{2}\rho(u^2 + v^2)\big) = (\gamma - 1)\rho\epsilon,$$

where the ratio of the specific heats γ is constant, with $\gamma = 1.4$ in our applications.

We are trying to solve the following conservation law :

$$\begin{cases} \frac{\partial W}{\partial t} + div f(W) = 0, & t > 0, x \in \Omega \\ W(x,0) = W_0(x), & x \in \Omega. \end{cases}$$

supplemented with the boundary conditions specified in the previous section.

We will give a quick glance to the method we are using, the Residual Distribution (RD) scheme. It is a second order oscillation free method. A complete description of this method for steady problems can be found, for example, in [16, 17].

Remark 1. *Reduced Order Modeling does not necessarily require a deep understanding of the underlying truth solver. We give these details about the CFD code because we intend to use it as part of our online scheme.*

2.3 Residual distribution scheme

We are concerned with the construction of algorithms for the approximation of solutions (8) on a conformal mesh whose elements are triangles in 2D. We will denote with T some generic element in the mesh, and by M a generic vertex.

In the RD schemes, the data are stored at the vertices. Denote W_i an approximation of $W(M_i)$ and $(f(W))^h$ an approximation of the flux $f(W)$. We consider the solution of the steady limit of (8). Based on [17], a Residual Distribution scheme is defined as follows:

Definition 1. *Let some current state W_i. Denote $(f(W))^h$ a continuous approximation of the flux. Evolve the nodal values of W_i toward a steady solution of the problem as follows:*

1. $\forall T$ compute the residual

$$\Phi^T = \int_T div((f(W))^h)dx = \int_{\partial T} (f(W))^h \cdot \vec{n}\ d\tilde{x}. \tag{8}$$

2. $\forall M_i \in T$ distribute the functions of Φ^T to each node of T. Denoting by Φ_i^T the local nodal residual for the node $M_i \in T$, by construction one must have

$$\sum_{M_i \in T} \Phi_i^T = \Phi^T. \tag{9}$$

Equivalently, denoting by β_i^T the distribution coefficient of node M_i:

$$\beta_i^T = \frac{\Phi_i^T}{\Phi^T} \tag{10}$$

with

$$\sum_{M_i \in T} \beta_i^T = 1. \tag{11}$$

3. Assemble the contribution $\forall M_i \in T$ and evolve W_i in time by integrating the ODE:

$$\sum_{M_i \in T} \Phi_i^T = 0. \tag{12}$$

The last sum has the role of an iterative process in order to get to the steady solution of (8). In other words, we seek for an approximation of the unsteady Euler equation when the time goes to infinity.

This is a very general formulation . Many classical schemes can be formulated within this framework. This variability hides mostly in how the residual of each triangle is distributed among nodes, that is, on the choice of the β_i. For instance, distributing it evenly among nodes corresponds to a Lax-Friedriech type of scheme, whereas schemes with upwinding would take into account the direction of the information. We have chosen the Lax-Friedriech type of scheme. This, as we will see in the online section, makes our online scheme easier.

Similarly to most CFD schemes, one can add 'numerical dissipation' in order to enforce the stability. The CFD code we have chosen to use, uses an SUPG type stabilization. That is, instead of Φ_i^T in equation (10), one uses

$$\Phi_i^T \leftarrow \Phi_i^T + h_T \int_T (\nabla_W f(W^h) \cdot \nabla V_i^h) \tau (\nabla_W f(W^h) \cdot \nabla W^h) dx, \tag{13}$$

where W^h is the continuous piece-wise linear interpolant of $(W_i^h)_{i=1,\cdots n_s}$, V^h is any continuous piece-wise linear test function and $\tau > 0$ is in practice set to

$$\tau = (\sum_i \max(\nabla_W f(\bar{W^h}) \nabla V_i^h, 0))^{-1}.$$

The used CFD mesh has 4510 grid points which corresponds to a total of 18040 unknowns. Snapshots in this solution manifold can be visualized in Figure 2. We have identified a range of parameters, for which the sensitivity of the shock position to Mach and AoA is high :

$$\mathcal{D} := \left\{ \begin{array}{lcl} \text{Mach} & \in & [0.81, 0.83] \\ \text{AoA} & \in & [0.0^\circ, 3.0^\circ]. \end{array} \right.$$

The positions of the shock for these specific parameters can be seen in Figure 1. This problem has been already studied in [15] in the context of model reduction using $L1$-norm minimization. It has been shown that when shock exists, discrepancies in the reduced solution are appearing. As a consequence, something else has to be done, namely calibration of the shock.

In the rest of this paper, we will denote u a generic component of the state vector W. For instance, one component of the output of the CFD code for parameter μ will be denoted $u(\cdot; \mu)$. This choice of notation is not made to confuse the reader, but rather to match the standard notation in the ROM community.

3 Offline phase

As we will use a POD method to construct a reduced basis, we first need to select a (moderate) but representative snapshot set inside $\mathcal{M}_\mathcal{D}$. We have chosen the following set of cardinal 12 :

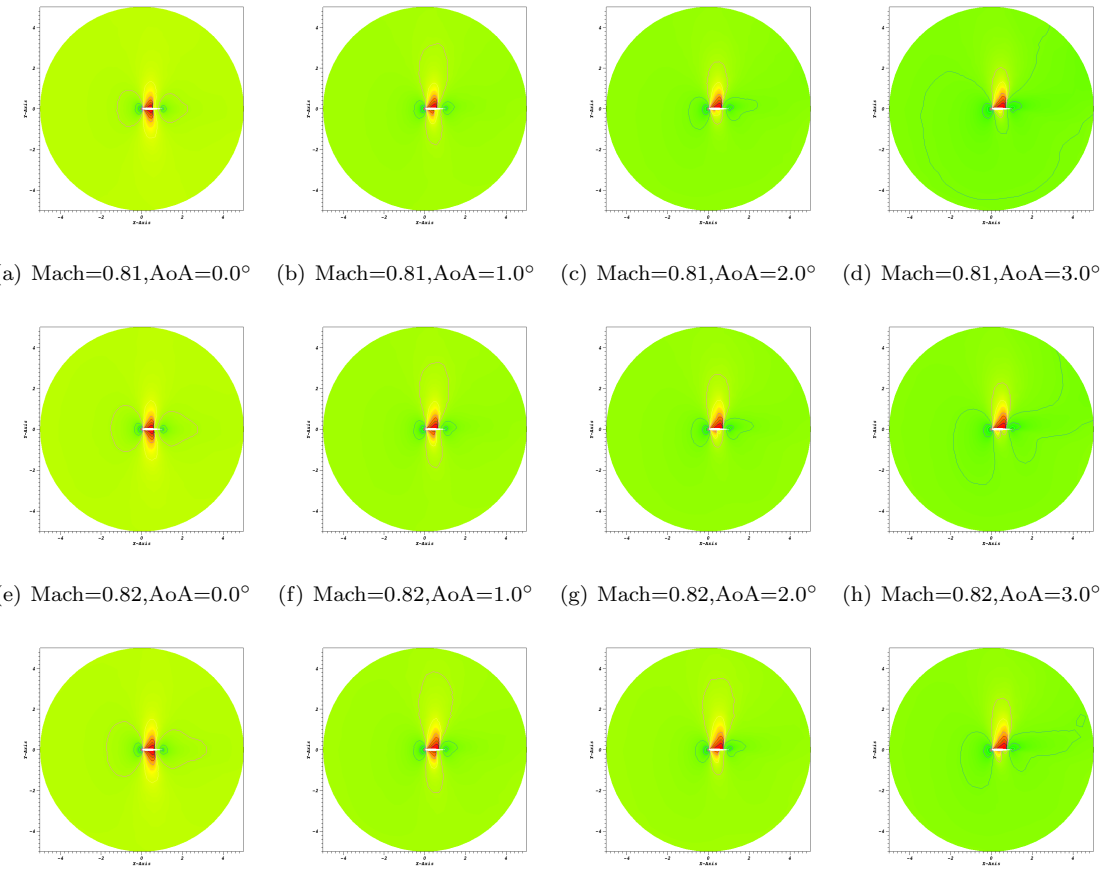
$$\begin{array}{ll} \text{Mach} & \in \{0.81, 0.82, 0.83\} \\ \text{AoA} & \in \{0.0^\circ, 1.0^\circ, 2.0^\circ, 3.0^\circ\}. \end{array}$$

The snapshots are presented on Figure 2.

The first objective of this section is to propose a calibration procedure, namely a family of mappings $\mathcal{F}$ as well as $\mu \to F_\mu$. We will show that the Kolmogorov n-width of the calibrated set is decaying faster than the one of the original set.

By plotting some POD basis of the original data set (see Figure 4), one can observe that just as in the 1D Burgers' case, in order to take into account the variability of the shocks' position and shape, the reduced basis tends to oscillate. This behaviour is even clearer when looking at the restriction of the POD basis at the wing, see Figure 3.

As mentioned in the introduction, calibration starts with some a priori knowledge on the solution manifold. By analogy with the first dimensional Burgers' case, we choose the following calibration



(a) Mach=0.81,AoA=0.0° (b) Mach=0.81,AoA=1.0° (c) Mach=0.81,AoA=2.0° (d) Mach=0.81,AoA=3.0°

(e) Mach=0.82,AoA=0.0° (f) Mach=0.82,AoA=1.0° (g) Mach=0.82,AoA=2.0° (h) Mach=0.82,AoA=3.0°

(i) Mach=0.83,AoA=0.0° (j) Mach=0.83,AoA=1.0° (k) Mach=0.83,AoA=2.0° (l) Mach=0.83,AoA=3.0°

Figure 2: The solutions of the problem for AoA=$\{0.0°, 1.0°, 2.0°, 3.0°\}$ and Mach=$\{0.81, 0.82, 0.83\ \}$

: Let $\hat{\Omega}$ some reference domain. Let $\hat{x}_0$ some abscissa in $\hat{\Omega}$. Construct $\mathcal{F}$ a family of mappings such that

$$\forall \mu \in \mathcal{D},\ \exists F_\mu \in \mathcal{F},\ \left\{(\hat{x}, \hat{y}) \text{ where } u(\cdot;\mu) \circ F_\mu^{-1} \text{ is discontinuous } \right\} \subset \{(\hat{x}_0, \hat{y})\}$$

To put it in other words, with this choice of calibration, the solutions in the calibrated solution manifold

$$\mathcal{M}_{\mathcal{F},\mathcal{D}} := \left\{u\left(F_\mu^{-1}(\cdot);\mu\right),\ \mu \in \mathcal{D}\right\}$$

have vertical shocks, at position $\hat{x}_0$. Again, with analogy with the one dimensional Burgers case, we should have

$$d_n(\mathcal{M}_{\mathcal{F},\mathcal{D}}) \leq d_n(\mathcal{M}_{\mathcal{D}})$$

How do we achieve this calibration ? The first task is to locate the position of the shock. We have chosen the following simple strategy: first find the boundary element (on the wing) where the

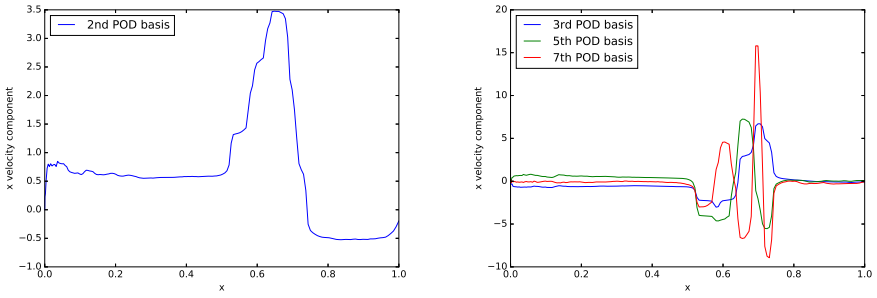


Figure 3: The x velocity component at the wing in the uncalibrated case : a few POD basis

quantity of interest has the highest gradient. Then look at neighboring elements and pick the one with the highest gradient. Iterate until the end of the shock (i.e some condition on the gradient) or until one reaches some predefined distance to the wing. One can use other methods in order to locate more precisely the shock. For instance, in [18], they use ENO related ideas to locate the inter cell position of the shock.

We denote as $x = s(y;\mu), \mu \in \mathcal{D}$ the true shape of the shock and we will make the following assumption :

$$\exists k \text{ small }, \ \forall \mu \in \mathcal{D}, \exists P_\mu \in \mathcal{P}_k(\mathbb{R}), \ s(y;\mu) = \mathcal{P}_\mu(y). \tag{14}$$

That is, the shock can be represented by a low order polynomial. All numerical experiments presented in this paper have been done using a one degree polynomial of form:

$$P_\mu(y) = a_0(\mu) + a_1(\mu) * y. \tag{15}$$

In Figure 1, the colored lines are the barycenters of the control volumes with the highest gradient. In black, is the fitted polynomial, characterized by two parameters.

Second step now, we need to construct the family $\mathcal{F}$. The global picture is presented on figure 5. We decompose Ω into three subdomains : Ω_0 where we will use the identity mapping. The calibration is going to be done in Ω_L and Ω_R. We have chosen to use a Gordon-Hall (G-H) [19] type mapping. Their accuracy and stability have been studied in [20]. Examples in fluid dynamics have been numerically studied in [8]. There are multiple reasons for this choice. For the offline part, what is important is its simplicity and its flexibility. Indeed, what is presented in the following section is not dependent on the shape of the shock nor on the shape of the wing. We will give computational cost related arguments in the online section below. The rest of this section will detail the application of the Gordon-Hall method onto Ω_L. Similar work has to be performed on the right subdomain.

The reference domain $\hat{\Omega}$ has to be a rectangle in the original G-H algorithm (we refer to Section 6.2 for another choice). This fits in our framework, as we want the calibrated shock to be a vertical line. The situation is depicted in Figure 6.

One possible instance of $F_\mu^{-1}(\hat{\Omega})$ is presented on Figure 7. Contrary to most examples using Gordon-Hall type method in the literature, our domain of interest is embedded in a bigger domain.

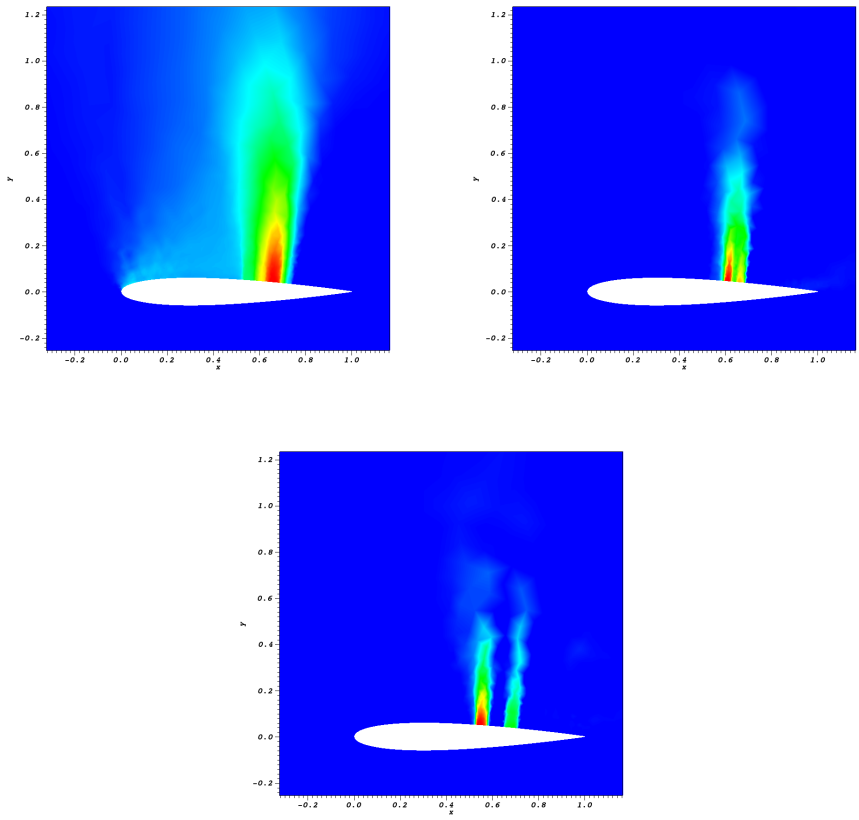


Figure 4: 1st, 3th and 5th POD basis at the wing in the uncalibrated case for the full domain Ω

The mapping thus need to be (at least) continuous on $\partial\Omega_L$, and $\partial\Omega_R$. More precisely, we need $x_1 = \hat{x}_1$, $x_3 = \hat{x}_3$, $y_1 = \hat{y}_1$ and $y_2 = \hat{y}_2$.

The G-H method is conceptually easy to understand. We denote with Γ_i the edges of Ω_L where i is the number of edges and we choose a clockwise numbering, starting from the left boundary. Their counterparts on $\hat{\Omega}_L$ are denoted $\hat{\Gamma}_i$. The steps are the following:

- map each edge of $\hat{\Omega}_L$ onto its counterpart on Ω_L. That is define f such that :

$$\forall i,\ f_\mu|_{\hat{\Gamma}_i} = \Gamma_i$$

- define the weights functions ϕ_i:

$$\begin{array}{lcl} \hat{\Omega} & \mapsto & [0,1] \\ (\hat{x},\hat{y}) & \to & \phi_i \end{array}$$

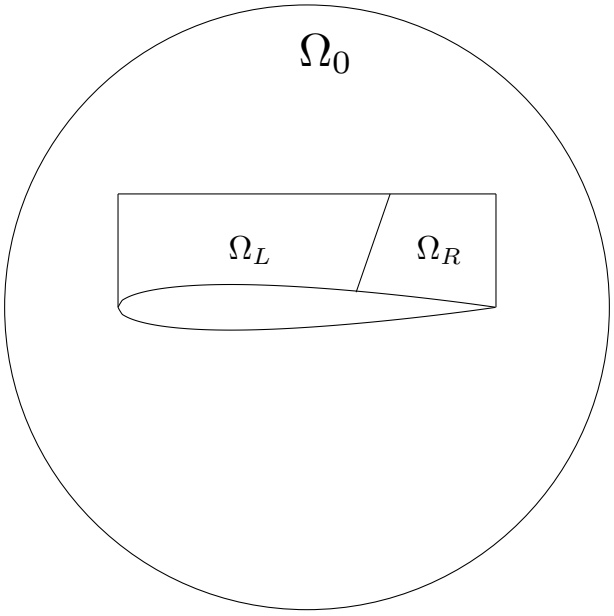


Figure 5: Physical domain Ω

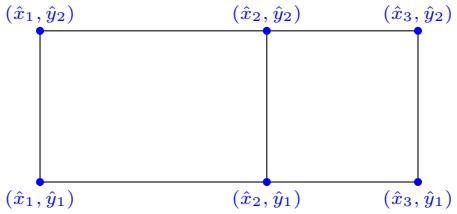


Figure 6: Reference domain $\hat{\Omega}$

Satisfying the following necessary conditions :

$$\begin{cases} \forall i,\ \phi_i + \phi_{i+2} = 1 \\ \forall i,\ \phi_i|_{\hat{\Gamma}_i} = 1 \end{cases}$$

These functions represent the relative positioning between the opposing edges. As such, in order to have a well behaved mapping, these need to be ' pas delirante '.

- define the projection functions π_i;

$$\begin{array}{lll} \hat{\Omega} & \mapsto & [0,1] \\ (\hat{x}, \hat{y}) & \to & \pi_i \end{array}$$

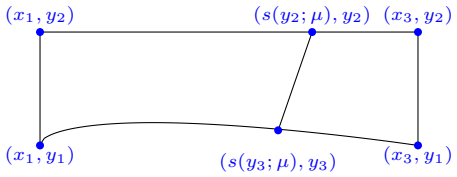


Figure 7: Physical domain Ω

Satisfying the following necessary condition :

$$\begin{cases} \forall i,\ \pi_i|_{\hat{\Gamma}_{i+1}} = 1 \\ \forall i,\ \pi_i|_{\hat{\Gamma}_{i-1}} = 0 \\ \forall i,\ \pi_i|_{\hat{\Gamma}_i} = [0,1] \end{cases}$$

these functions define a new coordinate system in $\hat{\Omega}$. As the ϕ's, they also need not be 'delirante'.

- for any point $(\hat{x}, \hat{y})$ on $\hat{\Omega}$, compute the projection on each edge $\pi_i(\hat{x}, \hat{y})$. Then, use a weighted combination of the $f_\mu(\pi_i(\hat{x}, \hat{y}))$

As a first easy step, we have chosen the following parametrization of the edges:

$$\begin{array}{l} f_\mu|_{\hat{\Gamma}_1} : (\hat{x}_1, \hat{y}) \to (x_1, \hat{y}) \\ f_\mu|_{\hat{\Gamma}_2} : (\hat{x}, \hat{y}_2) \to (x_1 + \hat{x} \cdot (s(y_2;\mu) - x_1), y_2) \\ f_\mu|_{\hat{\Gamma}_3} : (\hat{x}_2, \hat{y}) \to (s(y_2 + \hat{y} \cdot (y_3 - y_2); \mu), y_2 + \hat{y} \cdot (y_3 - y_2)) \\ f_\mu|_{\hat{\Gamma}_4} : (\hat{x}, \hat{y}_1) \to (s(y_3;\mu) + \hat{x} \cdot (x_1 - s(y_3;\mu)), w(s(y_3;\mu) + \hat{x} \cdot (x_1 - s(y_3;\mu))). \end{array} \tag{16}$$

For example, take the left edge $\hat{\Gamma}_1$ of the reference domain. It is le lieu des $\{(\hat{x}, \hat{y}),\ \text{ s.t } \hat{y} \in [\hat{y}_1, \hat{y}_2] \text{ and } \hat{x} = \hat{x}_1\}$. The vector valued function $f_\mu|_{\hat{\Gamma}_1}$ will be one possible parametrization of Γ_1.

We have chosen, at first, the same weight functions and the same projection functions as in the original G-H formulation:

$$\phi_1 = \frac{\hat{y} - \hat{y}_1}{\hat{y}_2 - \hat{y}_1} \quad \phi_3 = 1 - \frac{\hat{y} - \hat{y}_1}{\hat{y}_2 - \hat{y}_1}$$
$$\phi_2 = \frac{\hat{x} - \hat{x}_1}{\hat{x}_2 - \hat{x}_1} \quad \phi_4 = 1 - \frac{\hat{x} - \hat{x}_1}{\hat{x}_2 - \hat{x}_1}.$$

and

$$\pi_1(\hat{x}, \hat{y}) = \frac{\hat{x} - \hat{x}_1}{\hat{x}_2 - \hat{x}_1} \quad \pi_3(\hat{x}, \hat{y}) = \frac{\hat{y} - \hat{y}_2}{\hat{y}_3 - \hat{y}_2}$$
$$\pi_2(\hat{x}, \hat{y}) = \frac{\hat{x} - \hat{x}_2}{\hat{x}_3 - \hat{x}_2} \quad \pi_4(\hat{x}, \hat{y}) = \frac{\hat{x}_3 - \hat{x}}{\hat{x}_3 - \hat{x}_1}.$$

We insist that this method is flexible. Other choices will be presented in Section 6.2.

We will highlight the parameter dependency by using μ as a subscript. The Gordon-Hall mapping is given by :

$$\begin{aligned} GH(\hat{x},\hat{y};\mu) = &\phi_1(\hat{x},\hat{y})\cdot f_\mu(\hat{x}_1,\hat{y}) + \phi_2(\hat{x},\hat{y})\cdot f_\mu(\hat{x},\hat{y}_2) \\ &+ \phi_3(\hat{x},\hat{y})\cdot f_\mu(\hat{x}_2,\hat{y}) + \phi_4(\hat{x},\hat{y})\cdot f_\mu(\hat{x},\hat{y}_1) \\ &- \sum_{i=1}^{4} \phi_i(\hat{x},\hat{y})\cdot\phi_{i+1}(\hat{x},\hat{y})\cdot f_{i;\mu}, \end{aligned} \tag{17}$$

where $f_{i;\mu}$ is the value of f_μ in the corner between Γ_i and Γ_{i+1}. Here, we have

$$\begin{aligned} f_{1;\mu} = (x_1, y_1), \quad f_{2;\mu} = (x_1, y_2) \\ f_{3;\mu} = (x_3, y_2), \quad f_{4;\mu} = (x_3, y_1). \end{aligned}$$

We will use, in the course of this paper, the following notation :

$$\begin{array}{lcl} \mathbb{R}^2 & \mapsto & \mathcal{F} \\ (a_0, a_1) & \to & GH(\cdot; a_0, a_1) \end{array}$$

That is, the application takes as argument a shock position, and returns the corresponding G-H mapping.

Remark 2. *It is important to know that the π's, the ϕ's and f_μ can be chosen independently from each other. This will be made clearer in Section 6.2 when we try to improve the method.*

Remark 3. *We remind the reader that from the Gordon-Hall formulation, the mapping (17) is depending only on the coefficients of the low order polynomial $\mathcal{P}_\mu(y)$ defined in (15), namely $a_0(\mu)$ and $a_1(\mu)$. Moreover, we can generalize and to consider a polynomial of any order:*

$$P_\mu^n(y) = a_0(\mu) + a_1(\mu) * y + a_2(\mu) * y^2 + \cdots + a_n(\mu) * y^n. \tag{18}$$

This change will not affect the construction of the mapping.

It is clear that this mapping suffers from major drawbacks :

- this mapping is continuous at the boundary, but has discontinuous derivatives;
- this mapping linearly stretches the domain; this is not the best choice to diminish the Kolmogorov n-width;
- we have not taken into account the curvature of the wing;
- in x_1 and x_3, the boundary $\partial\hat{\Omega}$ is not C^1.

We will fix these issues in the numerical section 6.2. We will nevertheless numerically illustrate the usefulness of calibration using this rough mapping. We have computed separate POD basis on $\hat{\Omega}_L$ and $\hat{\Omega}_R$. We present on Figure 8 the counterpart of Figure 3, that is, the x component of the velocity at the wing. As one can see, using calibration we got rid of the oscillations. We present in Figure 9 the first, third and fifth POD basis in the calibrated case, as a counterpart of Figure 4. As expected, the calibrated POD captures most of the information in the first 4 basis. The 5th basis only contains numerical noise.

We will present in the next section a reduced scheme. This scheme should be cheap, i.e it should not have a computational complexity depending on the size of the truth problem. This scheme will use the calibrated basis that we have just constructed.

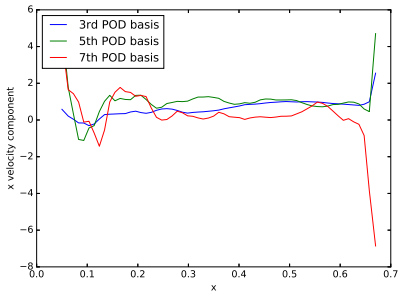


Figure 8: The x velocity component at the wing in the calibrated case : a few POD basis

4 Online phase

Now we enter the difficult section. Can we fit the previous pieces together to build a fully functioning reduced scheme? Three different methods will be mentioned here. We can think of them as increasing in difficulty. The first two will use some existing CFD codes. The last method describes a fully reduced scheme.

For the rest of this section, we will drop out the μ dependency, as we are focused on reducing one particular simulation. It will reappear in the offline/online decomposition section. Throughout this section, we will use the following notation :

- t^n is the pseudo time stepping, to reach steady state;
- $\hat{w}_i$ a generic control volume on the reference mesh $\hat{\Omega}$;
- w_i a generic control volumes on Ω;
- F_n the mapping chosen at time step n. It maps Ω onto $\hat{\Omega}$. The inverse mapping will be denoted F_n^{-1};
- $\{\phi_i\}$ some reduced basis on the reference mesh, of cardinal N^{red}. The one construted in section 3.

We will denote by u^n the solution at pseudo time step t^n and $\hat{u}^n$ it's counterpart on the reference mesh. That is, we have

$$\hat{u}^n = u^n \circ F_n^{-1} \text{ on } \hat{\Omega}$$

- The easiest method we can think of is the following :
  - Suppose we have some reduced solution at iteration n, $\hat{u}^n$, defined on the reference domain $\hat{\Omega}$, and a "well chosen" mapping F_n;
  - Map this reduced solution onto the real mesh, using F_n;
  - Use the CFD code, on Ω, using $\hat{u}^n \circ F_n$ as initial condition, to get u^{n+1};
  - Map u^{n+1} back onto $\hat{\Omega}$, find a "good" (in some sense) mapping F_{n+1}, and the corresponding reduced coordinates.

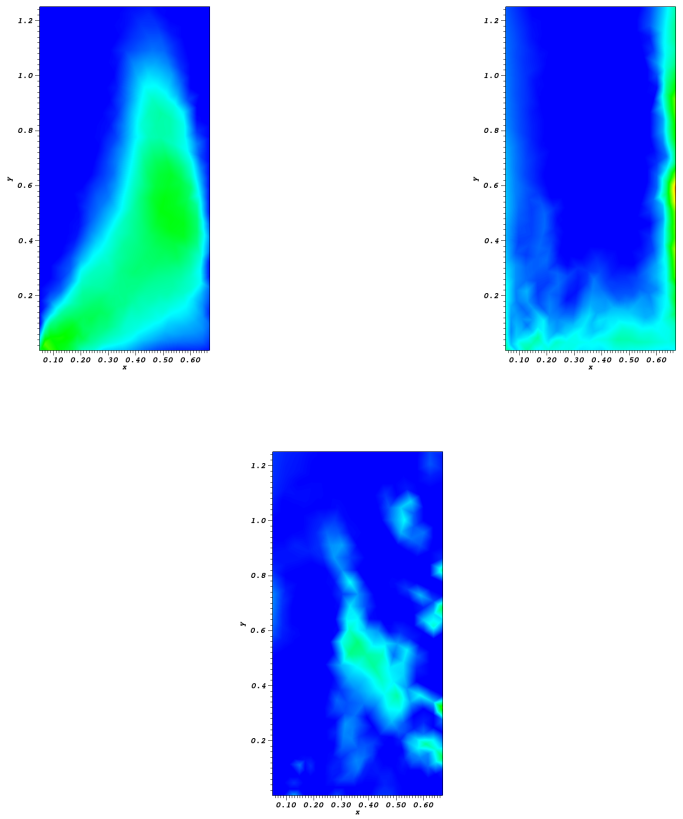


Figure 9: 1st, 3th and 5th POD basis in the calibrated case for the left subdomain

- The second method is smarter, and more in the spirit of what has been done in the 1D Burgers' case in [11].
  - Just as for the first method, suppose we have some reduced solution at iteration n, $\hat{u}^n$, defined on the reference domain $\hat{\Omega}$, and a "well chosen" mapping F_n;
  - Use the CFD code on $\hat{\Omega}$ using $\hat{u}^n$ as initial condition. This time, modify the parameters of the problem to make this problem equivalent to the original one. The objective is to be as little intrusive into the CFD code as possible. We will show that we fulfill the objective modifying only the flux and the boundary conditions. Denote $\tilde{u}^{n+1}$ the output. By construction, we have :
  $$\tilde{u}^{n+1} \approx u(\cdot, t^n) \circ F_n^{-1};$$
  - deduce a new "relative" mapping : $F_{n+1} \circ F_n^{-1}$ best suited to represent $\tilde{u}^{n+1}$. From this, compute a better calibrated solution $\hat{u}^{n+1}$ and the corresponding mapping F_{n+1} such

that

$$u(\cdot, t^{n+1}) \approx \hat{u}^{n+1} \circ F_{n+1};$$

- The third method is the ultimate goal to all reduce basis models. We want to construct a self sufficient reduced scheme, i.e we do not want to use the black box CFD code. Two paths can be taken. One idea is that since we are working with reduced basis, we do not need stability ingredients as smart as the ones used for the fine scheme. This argument has been studied in [21]. In our case, the fine scheme is using the advanced "Residual Distributed scheme" with SUPG type stabilization described in Section 2. The goal would be to use a simple scheme, say a raw Lax-Friedrichs method, which has no local components such as stabilization or "upwinding". A rectification step is then used to go from the reduced solution to the 'truth' original solution. The underlying idea is that the two stabilization methods, even if they arrive to different solutions, still represent the same underlying solution. Another direction is to assume that using calibration makes the complicated local components manageable by ROM.

  For instance, to ensure the TVD property, schemes often involve some gradient limiter near shocks. Can this be handled by standard ROM? Denote by $d_n(\nabla_{lim})$ it's Kolmogorov n-width, for some norm. Suppose that the shocks are order one polynomials, whose coefficients vary in $A_0 := [a_0^{min}, a_0^{max}]$, $A_1 := [a_1^{min}, a_1^{max}]$. That is,

  $$\forall \mu,\ \exists (a_0(\mu), a_1(\mu)) \in A_0 \times A_1,\ \text{s.t},\ \text{s}(y;\mu) = a_0(\mu) + a_1(\mu) * y$$

  We can estimate $d_n(\nabla_{lim})$. Let h some characteristic size of the mesh.

  $$d_n(\nabla_{lim}) \approx \frac{mes(A_0) * mes(A_1)}{h^2}$$

  There is no hope in trying to handle this term using the Empirical Interpolation Method (EIM).

  As calibration reduces the geometric variability of the shock position, the coefficients in the calibrated problem $\hat{A}_0$ and $\hat{A}_1$ will both be of order h. This drastically diminishing the Kolmogorov n-width of $d_n(\nabla_{lim})$. The same kind of arguments could be used for upwinding type of terms.

The first method will not be further discussed here, as the numerous mesh interpolations imply very high computational costs, as well as numerical errors. The third method is out of the scope of this paper. It is discussed in chapter 5 of the thesis [22]. We will focus on the second method.

Second method

This method assumes the existence of a fully functioning CFD code. In the lines of what has been done [23], the idea is to keep the stability and accuracy properties of the truth scheme by running the same code. The computational savings are obtained by using EIM/hyper reduction ideas.

The objective is to recast the original problem defined on Ω, onto an equivalent problem defined on $\hat{\Omega}$. We start with a step common to Finite Volume schemes and Residual Distribution schemes. Let w_i one control volume in Ω. Integrate the conservation law in space and time. Again, u is any the state variable.

$$\int_{w_i} u(w, t^{n+1}) dw - \int_{w_i} u(w, t^n) dw + \int_{w_i} \int_{t^n}^{t^{n+1}} \nabla \cdot f(u) dt dw = 0 \tag{19}$$

We will handle the boundary condition later.

What is usually done in the elliptic or parabolic communities [24, 25] is to transform the original problem, into a problem defined on the reference mesh. Let $\hat{w} = F_n(w)$. We use the standard change of variable and assume smoothness properties on the mapping. For $F \in \mathcal{F}$, we denote J_F the Jacobian of the transformation.

$$\int_{\hat{w}_i} \hat{u}^{n+1} |J_{F_n^{-1}}| d\hat{w} - \int_{\hat{w}_i} \hat{u}^n |J_{F_n^{-1}}| d\hat{w} + \int_{\hat{w}_i} \int_{t^n}^{t^{n+1}} \hat{u}(f \nabla \hat{u}) |J_{F_n^{-1}}| dt d\hat{w} = 0$$

The proof can be found in the references given above. There are two issues with this approach. The first one is that this derivation is not rigorous in the case of hyperbolic problems as some of the quantities (∇u) lose meaning when shock appear. Also, this formulation is not suited for our purpose. In the literature it has only been used for Galerkin type methods. Our intent is still to be as little intrusive as possible into a CFD code. We would like to find modified flux and boundary conditions such that the problem on the reference mesh is still casted as a finite volume problem. Using arguments that are detailed in the appendix, we can show that solving the initial problem (19) is equivalent to solving the following :

$$\int_{\hat{w}_i} \hat{u}(\hat{w}, t^{n+1}) |J_{F_n^{-1}}| d\hat{w} - \int_{\hat{w}_i} \hat{u}(\hat{w}, t^n) |J_{F_n^{-1}}| d\hat{w} + \int_{\hat{w}_i} \int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T f(\hat{u})) dt d\hat{w} = 0$$

where the $\{\hat{w}_i\}_i$ form a partition of $\hat{\Omega}$ and where $N_n^T f$ is the correct modified flux with

$$N_n^T = \begin{bmatrix} (J_{F_n^{-1}})_{22} & -(J_{F_n^{-1}})_{12} \\ -(J_{F_n^{-1}})_{21} & (J_{F_n^{-1}})_{11} \end{bmatrix}_n .$$

We will make the following assumption : the determinant of the Jacobian is sufficiently smooth and the mesh is fine enough so that we can consider it constant per element. The approximation error will not be investigated in this paper.

Remark 4. *Some more rigorous approaches could be developed, but would lead to more intrusion into the CFD code. In [26] for instance, they choose to work with $\hat{u}|J_{F_n^{-1}}|$ instead of $\hat{u}$.*

We are arriving then to the following equation in each control volume $\hat{w}_i$.

$$\int_{\hat{w}_i} \hat{u}(\hat{w}, t^{n+1}) d\hat{w} - \int_{\hat{w}_i} \hat{u}(\hat{w}, t^n) d\hat{w} + \frac{1}{|J_{F_n^{-1}}|_i} \int_{\hat{w}_i} \int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T f(\hat{u})) dt d\hat{w} = 0$$

We have all the ingredients to feed the CFD code :

- a mesh : here it is the reference mesh, over $\hat{\Omega}$;
- the average of the solution over control volumes :

$$\hat{\mathbf{u}}_i = \frac{1}{\text{mes}(\hat{w}_i)} \int_{\hat{w}_i} \hat{u}(\hat{w}, t^n)$$

- a flux, in a closed form: with the Piola transform, here it just amounts to

$$N_n^T f$$

  where the N^T term will depend on the time step and is not constant over $\hat{\Omega}$. We will see in Section 5.1 that using G-H type mapping allows offline/online decomposition.

- boundary conditions: we do not need to worry about the outside boundary conditions, as they will not be affected by the mapping. The slip boundary conditions for the original problem are weakly imposed boundary conditions and are given by

$$u \cdot \vec{n} = 0 \text{ on the wing.}$$

In our case, these are imposed as follows : treat the boundary nodes as any other node. Add the correct quantity to impose the slip boundary condition. More precisely, let $n = (n_1, n_2)$ the norm at the boundary . The following flux is computed at nodes on the boundary :

$$(f_x, f_y) \cdot n = \begin{pmatrix} \rho\Big((u,v)\cdot n\Big) \\ \rho u\Big((u,v)\cdot n\Big) + pn_1 \\ \rho v\Big((u,v)\cdot n\Big) + pn_2 \\ \Big((u,v)\cdot n\Big)(E+p) \end{pmatrix}$$

We enforce the slip boundary condition by subtracting the following quantity :

$$(\tilde{f}_x, \tilde{f}_y) \cdot n = \begin{pmatrix} \rho\Big((u,v)\cdot n\Big) \\ \rho u\Big((u,v)\cdot n\Big) \\ \rho v\Big((u,v)\cdot n\Big) \\ \Big((u,v)\cdot n\Big)(E+p) \end{pmatrix}$$

We can use the Piola transform again for these terms. In the original transformation, the following quantity is subtracted from the residual at boundary elements :

$$\int_{\partial K} (\tilde{f}_x(u^h), \tilde{f}_y(u^h)) \cdot n,$$

and formulated in terms of reference variables as,

$$\int_{\partial \hat{K}} (\tilde{f}_x(\hat{u}^h), \tilde{f}_y(\hat{u}^h)) \cdot (N^T \cdot n).$$

The conclusion from this analysis is the following : under the assumption that the determinant of the Jacobian is constant per element, changing the normals in the CFD code is enough to compute the residual in each triangle. As mentioned in the offline section, the CFD codes is of Lax-Friedriechs type, that is we distribute the residual evenly among nodes inside each element. Thus, the distribution procedure is done independently of the mesh, or of the solution. For an upwinding scheme, this is a much more difficult problem to tackle, not in the scope of this paper.

As mentioned in section 2.3, the truth scheme uses SUPG type stabilization. We have not studied in this paper how to modify this term in order to have an equivalent stabilization procedure on $\hat{u}$. We will discuss this approximation in the numerical experiment section.

We now assume that we have performed the $n+1$ iteration with the CFD code. That is, we have $\tilde{u}^{n+1} \circ F_n \approx u^{n+1}$. We are looking simultaneously for :

- a better suited mapping F_{n+1}
- the corresponding coordinates $\{\alpha_k^{n+1}\}_k$.

Following [11], define the following objective function, for $p \in \{1, 2\}$:

$$J^p : \left\{ \begin{array}{lll} \mathcal{F} \times \mathbb{R}^{N^{red}} & \mapsto & \mathbb{R} \\ F, \{\alpha_k\}_k & \to & \left\| \tilde{u}^{n+1} \circ F_n - \sum_k \alpha_k \phi_k \circ F \right\|_{L^p} \end{array} \right. \tag{20}$$

We will see in the next section that we need to modify the objective function before performing any standard optimization algorithm. Indeed, because of the discontinuity at the shock, J^p is non differentiable. We will propose one (of the many) possible options to solve this problem.

Remark 5. *In [11], the authors were studying the viscous Burgers equation. Thus, they did not have this specific issue.*

5 Coordinates

We first propose an optimization procedure when the mapping is assumed to be known. We will then see how to find it. Let $F \in \mathcal{F}$. Define J_F^p as :

$$J_F^p : \left\{ \begin{array}{lll} \mathbb{R}^{N^{red}} & \mapsto & \mathbb{R} \\ \{\alpha_k\}_k & \to & \left\| \tilde{u}^{n+1} \circ F_n - \sum_k \alpha_k \phi_k \circ F \right\|_{L^p(\Omega)} \end{array} \right. \tag{21}$$

L^2 minimization, standard Galerkin projection

$$J_F^2 = \left\| \tilde{u}^{n+1} \circ F_n - \sum_k \alpha_k \phi_k \circ F \right\|_{L^2(\Omega)}$$

First order optimality condition gives us the αs. Attention, the basis $\{\phi_k \circ F\}_k$ will most probably not be an orthogonal basis.

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N^{red}} \end{pmatrix} = A \begin{pmatrix} < \tilde{u}^{n+1} \circ F_n, \phi_1 \circ F >_X \\ < \tilde{u}^{n+1} \circ F_n, \phi_2 \circ F >_X \\ \vdots \\ < \tilde{u}^{n+1} \circ F_n, \phi_{N^{red}} \circ F >_X \end{pmatrix}$$

where $A_{i,j} :=< \phi_i \circ F, \phi_j \circ F >_X \in \mathbb{R}^{N^{red}}$. Define $\delta_F := F \circ F_n^{-1}$. We have

$$\begin{array}{lll} < \tilde{u}^{n+1} \circ F_n, \phi_k \circ F >_X & = & \int_{\hat{\Omega}} \tilde{u}^{n+1} \phi_k \circ \delta_F |J_{F_n^{-1}}| \\ < \phi_i \circ F, \phi_j \circ F >_X & = & \int_{\hat{\Omega}} \phi_i \phi_j |J_{F^{-1}}| \end{array}$$

We will see in section 5.1 how to achieve efficient offline/online decomposition.

$L1$ **minimization**

$$\forall \alpha \in \mathbb{R}^{N^{red}},\ J_F^1(\alpha) = \sum_i \int_{\hat{w}_i} \left| \tilde{u}^{n+1} \circ F_n - \sum_k \alpha_k \phi_k \circ F \right|.$$

Once again, standard change of variable :

$$\forall \alpha \in \mathbb{R}^{N^{red}},\ J_F^1(\alpha) = \sum_i \int_{\hat{w}_i} \left| \tilde{u}^{n+1} \circ \delta_F^{-1} - \sum_k \alpha_k \phi_k \right| |J_{F^{-1}}| \tag{22}$$

We use some quadrature formula. Denote the quadrature points $\hat{x}_{i,l}$ and the weights $\gamma_{i,l}$. We have :

$$J_{\mathcal{F}}^1(\alpha) = \sum_i \sum_l \gamma_{i,l} \left| \tilde{u}^{n+1}(\delta_F^{-1}(\hat{x}_{i,l})) - \sum_k \alpha_k \phi_k(\hat{x}_{i,l}) \right| |J_{F^{-1}}(\hat{x}_{i,l})|. \tag{23}$$

This is handled as in [15] by recasting it as a linear programming problem. For now, the size of the problem is of order $\mathcal{N}$, the number of degrees of freedom of the truth space. We will see in section 5.1 how to reduce the computational cost.

How do we find this relative mapping? One important assumption is to have smoothness in the choice of mappings. This just amounts to saying that the shock has not moved a lot from pseudo time step t^n to pseudo time step t^{n+1}. This can be justified by Rankine-Hugoniot conditions.

Let $\hat{A}_0$ and $\hat{A}_1$ the maximum values for the relative variation in the position of the shock. These can easily be deduced from RK conditions.

$$\hat{A}_i = (t^{n+1} - t^n) * \text{ maximum shock speed}$$

Define the following neighborhood of the identity in $\mathcal{F}$:

$$\mathcal{F}^{rel} := \left\{ \text{G-H}(\hat{a}_0, \hat{a}_1),\ \hat{a}_i \in \hat{A}_i \right\}$$

Let $u \in \mathcal{M}_{\mathcal{D}}$. It is clear that for solutions with shocks, the following application is not smooth.

$$\begin{array}{ccc} \mathcal{F}^{rel} & \mapsto & X \\ \delta_F & \to & u(\delta_F(\cdot)) \end{array}$$

More precisely, the derivative in the sense of distributions we will have a Dirac mass at the shock. We give here a formal proof, and refer to [27, 28] for a rigorous one. Decompose the solution into a smooth part and one discontinuity. Denote $\Sigma(u) = \{(\hat{x}, \hat{y}),\ u \text{ is discontinuous } \}$ and $v \to [v]$ the standard jump operator.

$$\forall u \in \mathcal{M}_{\mathcal{D}},\ u = u_{smooth} + [u_j]|_{\Sigma(u)}$$

$$u - u \circ \delta_F = u_{smooth} - u_{smooth} \circ \delta F + [u_j]|_{\Sigma(u)} - [u_j \circ F]|_{\Sigma(u \circ F)}$$

At first order, the derivative in the sense of distributions is given by :

$$u - u \circ \delta_F = \partial u_{smooth} + \delta|_{\Sigma(u)} \frac{\partial \Sigma}{\delta F}$$

where $\delta|_{\Sigma(u)}$ is the Dirac mass at $\Sigma(u)$.

We propose a method for $p = 1$. Suppose that we pick the $\hat{x}_{i,l}$ sufficiently far from the shock, so that

$$\forall \delta_F \in \mathcal{F}^{rel},\ \delta_F(\hat{x}_{i,l}) \text{ is on the same side of the shock as } \hat{x}_{i,l}$$

Then, the following application will be differentiable :

$$\left\{ \begin{array}{lll} \mathcal{F}^{rel} & \mapsto & \mathbb{R}^{\mathcal{N}*N_{quad}} \\ \delta_F & \to & \{\tilde{u}^{n+1}(\delta_F(\hat{x}_{i,l}))\}_{i,l} \end{array} \right.$$

The same idea can be applied to the case $p = 2$.

We will denote $\hat{\Omega}_d$ the subdomain of $\hat{\Omega}$ where we have removed some neighborhood of the shock. We denote Ω_d it's counterpart in the physical domain.

Remark 6. *For the L^1 norm, the overall problem as presented is not differentiable. This can be solved using Huber type minimization instead of the raw L^1 [15].*

Remark 7. *For L^2, we must not forget to normalize, as $\|\phi_k\|_{L^2(\Omega_d)}$ are not of unit norm. This is a Gappy-POD type of algorithm, see [29].*

From now on, we define a 'smaller' but differentiable objective function. For all Ω_{sub} subdomains of Ω, define the following $J_{\Omega_{sub}}$:

$$J^p_{\Omega_{sub},F}(\alpha) = \left\| \tilde{u}^{n+1} \circ F_n - \sum_k \alpha_k \phi_k \circ F \right\|_{L^p(\Omega_{sub})}.$$

The differentiable issue leads us to replace the original problem J^p_F by $J^p_{\Omega_d,F}$.

This smoothness issue being solved, we can now perform standard optimization algorithm to get the desired mapping δ_F. Indeed, we know that the following function is smooth:

$$\begin{array}{lll} \hat{A}_0 \times \hat{A}_1 \times \mathbb{R}^{N_{red}} & \mapsto & \mathbb{R} \\ \hat{a}_0, \hat{a}_1, \{\alpha_k\} & \to & J_{\Omega_d, \text{G-H}(\hat{a}_0,\hat{a}_1)}(\alpha) \end{array} \tag{24}$$

This is of course computationally not reasonable. We present in the next section one method to make the whole thing computationally tractable.

5.1 Online/offline decomposition

We will first show how to perform in practice the otpimization of the quantity defined in (24). Discretize the set $\hat{A}_0$ and $\hat{A}_1$: $\{\hat{a}^k_0\}$ and $\{\hat{a}^k_1\}$. Define the following set of mappings :

$$\Psi_{\mathcal{F}^{rel}} := \{\text{G-H}(\hat{a}^k_0,\ \hat{a}^p_1)\}_{k,p}$$

This space if is discretized version of $\mathcal{F}^{rel}$ and is also embedded in $\mathcal{F}$.

Compute the coordinates for all mappings in $\Psi_{\mathcal{F}^{rel}}$. Compute the corresponding value of the objective function :

$$\forall \delta_F \in \Psi_{\mathcal{F}^{rel}},\ \inf_\alpha J_{\Omega_d,\delta_F}(\alpha)$$

Interpolate to get an approximation of $\hat{a}^{opt}_0$ and $\hat{a}^{opt}_1$. Compute the reduced coordinates on G-H($\hat{a}^{opt}_0$, $\hat{a}^{opt}_1$)

Remark 8. *Other ideas to find F_{n+1} can be implemented. They are however less natural in our framework.*

- ***Shock fitting:*** *close to what has been described in the offline section. Find the highest gradient elements, fit a polynomial. This is made computationally efficient because we do not need to look for highest gradient all over Ω. Once again, the smoothness of $t \to s(\mu; t)$, for all μ means that we only need to look in the restricted domain.*
- ***RK condition:*** *enforce Rankine-Hugoniot type conditions at the shock.*

We have not yet discussed the computational complexity of our full algorithm. For now, at each time step, we need full run of the CFD code, to get $\tilde{u}^{n+1}$ over Ω. Until we manage to build a full reduced scheme, see method three described in section 4 , this computational time is not easily reducible. The only ideas available in the literature are hyper reduction [30].

In the previous section, we have restricted the problem from Ω to Ω_d for smoothness reasons. Here, we replace $J^p_{\Omega_d}$ by an even smaller, denoted generically $J^p_{\Omega_{sub}}$ because of computational cost.

How does one pick a 'good' Ω_{sub}? This is what hyper reduction is. To put it simply, we are looking for Ω_{sub} such that :

$$\forall \hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}},\ \forall F \in \Psi_{\mathcal{F}^{rel}},\ \underset{\alpha}{\text{arginf}}\ J^p_{\Omega_{sub},F}(\alpha) \approx \underset{\alpha}{\text{arginf}}\ J^p_{\Omega_d,F}(\alpha).$$

That is, the optimization is not affected too much by the reduction of the size of the problem. Of course, we do not now the continuous set $\mathcal{M}_{\mathcal{F},\mathcal{D}}$. Denote $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$ some samples. We will assume that they are good representative of the continuous set.

We then perform the following greedy algorithm. Let ϵ some threshold.

Data: $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$,$\{\phi_k\}_k$
Result: Ω_{hyper},N^{hyper}
Initialize $\Omega_{hyper} := \bigcup_{i \in I_{ini}} \hat{w}_i$;
while *convergence* **do**
  $\forall \hat{u} \in \Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}},\ \{\alpha_k\}_k = \underset{\alpha}{\text{arginf}}\ J^p_{\Omega_{hyper},\mathcal{F}^{truth}}(\alpha)$;
  $i := \underset{p}{\text{argsup}}\ \| \sum_k \alpha_k \phi_k - \hat{u} \|_{L^p(\hat{w}_p)}$;
  $\Omega_{hyper} := \Omega_{hyper} \bigcup \hat{w}_i$
end
$N^{hyper} := \text{card}(\Omega_{hyper})$;

Algorithm 1: One possible choice to pick a good Ω_{sub}

The idea of hyper reduction is that on the solution manifold there is a one to one correspondence between the restriction of the solution on Ω^{hyper} and the full solution, for a specific quantity. The domain Ω^{hyper} depends of course on the quantity that is beeing studied. The accuracy of the overall process can be estimated but not guaranteed. Indeed, the greedy algorithm is performed on sampled spaces : $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$ instead of $\mathcal{M}_{\mathcal{F},\mathcal{D}}$ and $\Psi_{\mathcal{F}^{rel}}$ instead of $\mathcal{F}^{rel}$. One possible output of the algorithm is illustrated in Figure 10, where Ω_{hyper} is the reunion of black elements.

What the calibration has achieved, at the same time as it has reduced the Kolmogorov n-width of the solution set, is to localize spatially the interesting part of the solution.

Remark 9. *These two properties are equivalent for discontinuous solutions. This is not true for smooth solutions.*

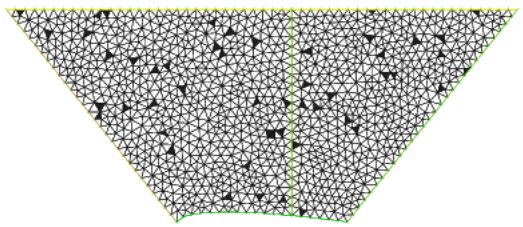


Figure 10: One possible output of algorithm 1

We can thus anticipate that the control volumes chosen will be accumulated around the calibrated shock, as depicted in Figure 11. In the non calibrated case, to retrieve the information

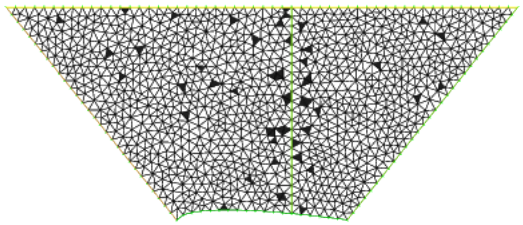


Figure 11: A more realistic output of algorithm 1

around the shock, one would need to use a bigger hyper reduced domain, namely something like

$$\Omega_{sum} := \left\{ (x,y), \quad \text{s.t} \ , \ (x < \min_{\mu} P_\mu(y)) \text{ and } (x > \max_{\mu} P_\mu(y)) \right\}.$$

Also, the method used to enforce the differentiability of J^p would not be applicable.

Implementation details?

For our choice of online implementation, the computation of the N^T terms is not a pressing issue, as these are only required in a moderate number of cells, denoted by N^{hyper}. We will nevertheless emphasize that this terms, because of the choice of Gordon-Hall type mapping, would not be a computational problem even with no hyper reduction. To put it in the ROM framework, we will simply highlight the fact that these terms are trivially affinely decomposable. We need to show that

$$\mu \to N^T(\mu)$$

is affinely decomposable, which is equivalent to saying that the entris of the Jacobian are affinely decomposable. This is a consequence of the structure of the G-H mapping, see (17). The weights and the projection functions are not parameter dependent. We have

$$\mu \to \psi_i,$$

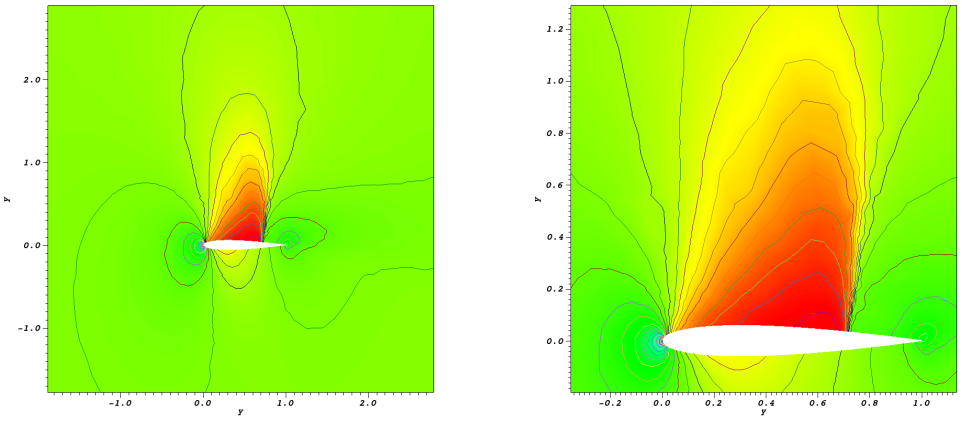
with ψ_i for $i \in 1,2,3$ linear function in μ. Because of the square root in the definition of NACA airfoil, we have something not affinely decomposable for ψ_4. This will be taken care by using the modifications to the G-H method described in section 6.2. We will enforce $F = Id$ in a neighborhood of Γ_1, thus removing the issue. The rest of the wing can be approximated by a polynomial.

6 Numerical Experiments

6.1 Mapping on a flat domain

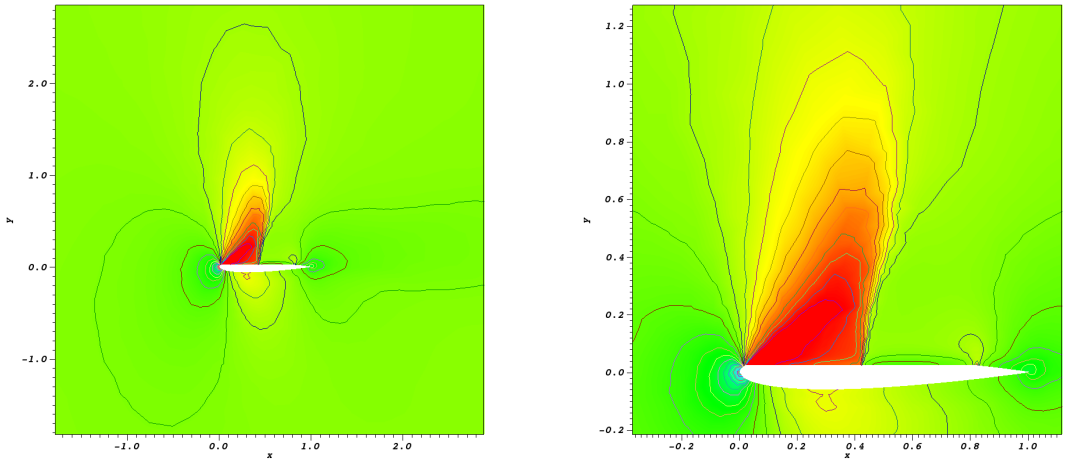
The first thing we will do is a necessary step. We will try to reproduce a simple test case using the mapping presented in the offline Section 3 and the second method presented in section 4. We will consider that we know the correct mapping from the begining of the simulation. That is, we perform the whole CFD code with the same transformed flux $N^T f$ where N^T corresponds to the correct, truth mapping. We will not try to quantify the results for now, but rather to check that the global calibration idea fits in this framework.

We are running the CFD code for Mach = 0.81 and AoA = 3.0°. The truth solution that we are trying to recover is presented in Figure 12. We first perform a 'control sample' test. That is, we run the CFD code using the identity mapping. The output solution is presented in Figure 13. As expected, the output is not comparable with the truth solution. We present in Figure 14 the solution for the mapping described in equation (17). One can observe that the general behaviour is correct. The shock is more or less located at the correct position and it has been straightened. Nevertheless, we can see that we have some non physical behaviour close to the wing. This could have been anticipated, as the mapping constructed in Section 3 suffers major issues. The biggest problem at the wing seems to be a consequence of the high gradient at the bottom left corner of the domain of $\hat{\Omega}_L$. From this preliminary analysis, we conclude that we need a smoother (more than continuous) mapping.



(a) (b)

Figure 12: Truth solution for velocity component with Mach=0.81 and AoA=3.0°

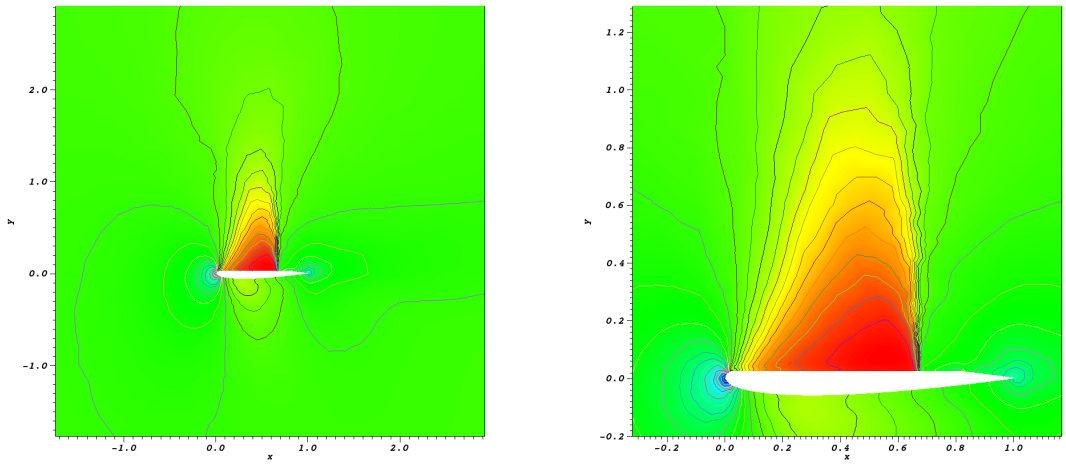


(a) (b)

Figure 13: The identity mapping velocity component on a flat domain

6.2 Mapping on a curved domain

The choice of a flat wing in the previous subsection was intentional. This reminds that the reference domain on which we are solving the problem is not the physical domain. Nevertheless, because of



(a) (b)

Figure 14: The mapped solution for velocity component on a flat domain

the smoothness of the mapping issue, we have decided to use a more advanced mapping than the raw Gordon-Hall. We will use a mapping from a curved domain to a curved domain (see Figure 15, 16). For this, we have to build a more advanced G-H type mapping. For that, we are following the same steps as in [20].

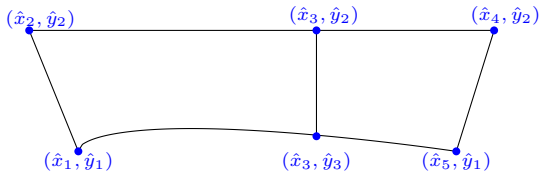


Figure 15: Reference domain $\hat{\Omega}$

Again, as in the case presented in section 3, we consider here that $x_1 = \hat{x}_1$, $x_2 = \hat{x}_2$, $x_4 = \hat{x}_4$, $x_5 = \hat{x}_5$, $y_1 = \hat{y}_1$ and $y_2 = \hat{y}_2$. In this case, the new G-H mapping or the generalized transfinite extension is described as follows:

$$GH(\hat{x}, \hat{y}) = \sum_{i=1}^{n} [\phi_i(\hat{x}, \hat{y})\psi_i(\pi_i(\hat{x}, \hat{y}), \mu) - \phi_i(\hat{x}, \hat{y})\phi_{i+1}(\hat{x}, \hat{y})\psi_i(1, \mu)], \tag{25}$$

where n is the number of the sides of the domain (in our case $n = 4$). If we denote with

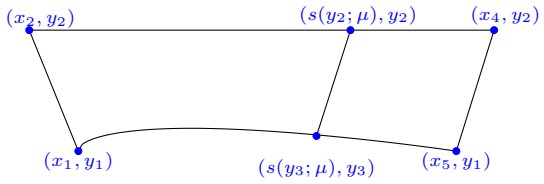


Figure 16: Physical domain Ω

$\hat{\Gamma}_i, i = 1, \cdots, n$ the sides numbered clockwise, we associate to each side a weight function ϕ_i and a projection function π_i, both defined over the reference domain $\hat{\Omega}$. We also assume that the value of f along $\hat{\Gamma}_i$ may be determined by the parametrization $\psi_i(t, \mu) : [0, 1] \times \mathcal{D} \to \mathbb{R}$, where t is the normalized arc-length of $\hat{\Gamma}_i$.

We will first present the original version proposed in [20]. This was done in a very general case, and the focus was put on the smoothness of the overall mapping. We will see why some ingredients need to be modified to match our objectives. The weights functions are taken as the solutions of the following Laplace problems :

$$\begin{cases} \Delta\phi_i &= 0 \text{ in} & \hat{\Omega} \\ \phi &= 1 \text{ on} & \hat{\Gamma}_i \\ \phi &= 0 \text{ on} & \hat{\Gamma}_{i+2} \\ \frac{\partial\phi}{\partial n} &= 0 \text{ on} & \hat{\Gamma}_{i-1} \cup \hat{\Gamma}_{i+1}. \end{cases} \tag{26}$$

We also need the projection from the interior onto each side $\hat{\Gamma}_i$. On a general domain, we compute the projection function π_i onto the side $\hat{\Gamma}_i$, by solving the Laplace problem (for more details, see [20]):

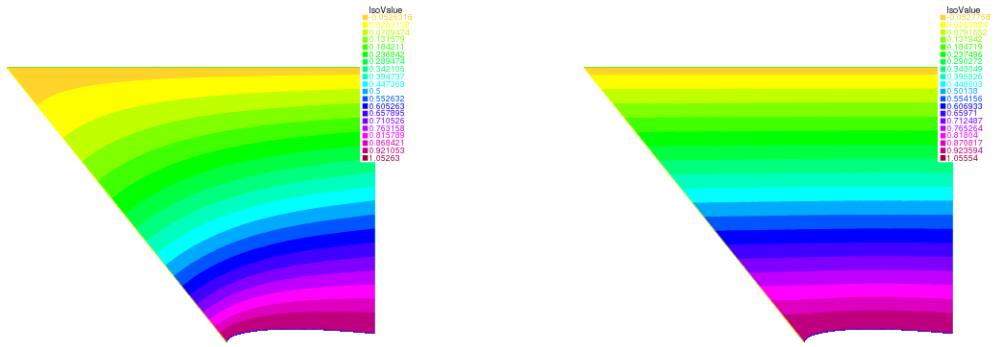
$$\begin{cases} \Delta\pi_i &= 0 \text{ in} & \hat{\Omega} \\ \pi &= t \text{ on} & \hat{\Gamma}_i, t \text{ monotone and smooth} \\ \pi &= 1 \text{ on} & \hat{\Gamma}_{i+1} \\ \pi &= 0 \text{ on} & \hat{\Gamma}_{i-1} \\ \frac{\partial\pi}{\partial n} &= 0 \text{ on} & \hat{\Gamma}_{i+2}. \end{cases} \tag{27}$$

We will now deal with the issues mentioned in Section 3 one by one. Firstly, we solve the smoothness of $\partial\hat{\Omega}$: we had noticed in our rough flat approximation that not taking into account the curvature of the wing was a too big approximation and as a consequence, the projection π_4 onto the wing had a non desired behaviour. The proper way of dealing with this curved boundary is to use the standard arclength definition as a projection function on the wing :

$$\pi_4|_{\Gamma_4} : (\hat{x}, \hat{y}) \to \int_0^{\hat{x}} \sqrt{1 + \left(\frac{\partial w}{\partial \hat{x}}\right)^2}.$$

We also need to be closer to the identity mapping on the left boundary. In the original formulation, homogeneous Neumann boundary conditions are imposed on neighboring edges when

computing the projection function. This choice is not the right one for our particular problem. We present in Figure 17 on the left, the projection function π_3 in the transfinite version of [20] described above. Remember, π_3 is the projection onto the edge $\hat{\Gamma}_3$ and this deforms the coordinate system. We could have anticipated that with this choice, our mapping will not match the outside boundary (where we have the identity mapping). The right picture in Figure 17 presents the solution for an alternative Neumann boundary condition, tailored for this specific $\hat{\Omega}$.



(a) (b)

Figure 17: Homogeneous Neumann boundary condition is not always the correct choice

Towards the same objective, we do not want any stretching of the solution around the left boundary and close to the shock. Indeed, we need smooth transitions to neighboring domains. How one can enforce this? One necessary step is to modify ψ_2 and ψ_4 from the original version. Denote with $H(x)$ some smoothed Heaviside step function and write it for ψ_2. The same could be done for ψ_4. We pick the following :

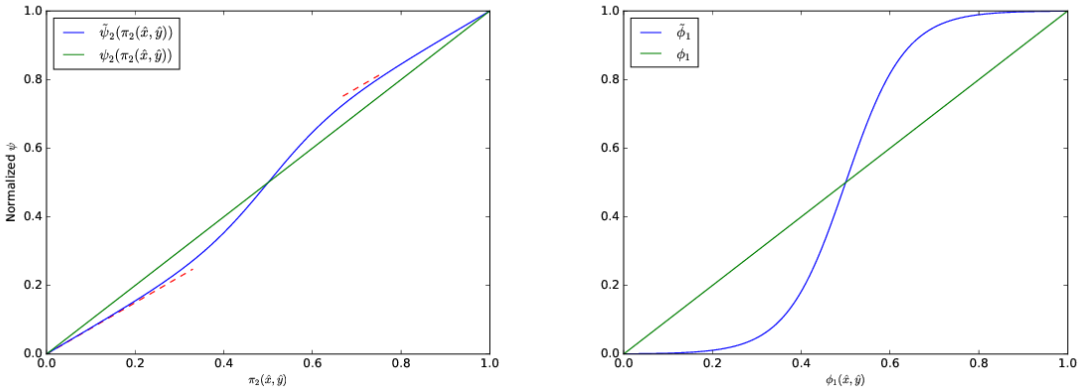
$$\tilde{\psi}_2(\hat{x},\hat{y},\mu) = \pi_2(\hat{x},\hat{y})\cdot\frac{\hat{x}_3-\hat{x}_2}{s(y_2;\mu)-x_2}\cdot(1-H(\pi_2(\hat{x},\hat{y})))+\left(1+(\pi_2(\hat{x},\hat{y})-1)\cdot\frac{\hat{x}_3-\hat{x}_2}{s(y_2;\mu)-x_2}\right)\cdot H(\pi_2(\hat{x},\hat{y})).$$

That is, we want no stretching for $\pi_2(\hat{x},\hat{y}) \approx 0$ or 1. A numerical example is presented in Figure 18 for an hypothetical stretching of $4/3$, where the dashed red lines correspond to a non stretched mapping.

We need to modify one more ingredient. We take steeper weight functions for boundaries 1 and 3. For instance, we can pick:

$$\tilde{\phi}_1(\hat{x},\hat{y}) = H\left(\phi_1(\hat{x},\hat{y})\right).$$

What this achieves is that close to left and right boundaries, the slopes are the same as the outside boundaries.



(a) (b)

Figure 18: We want to be closer to the identity mapping on $\hat{\Gamma}_1$ and $\hat{\Gamma}_3$

Finally, we choose the following ψ functions :

$$\text{On } \hat{\Gamma}_1 : \psi_1(\pi_1(\hat{x},\hat{y}),\mu) = \Big(x_1 + \pi_1(\hat{x},\hat{y})\cdot(x_2 - x_1), y_1 + \pi_1(\hat{x},\hat{y})\cdot(y_2 - y_1)\Big)$$
$$\text{On } \hat{\Gamma}_2 : \psi_2(\pi_2(\hat{x},\hat{y}),\mu) = \Big(x_2 + \pi_2(\hat{x},\hat{y})\cdot(s(y_2;\mu) - x_2), y_2\Big)$$
$$\text{On } \hat{\Gamma}_3 : \psi_3(\pi_3(\hat{x},\hat{y}),\mu) = \Big(s(y_2 + \pi_3(\hat{x},\hat{y})\cdot(y_3 - y_2);\mu), y_2 + \pi_3(\hat{x},\hat{y})\cdot(y_3 - y_2)\Big)$$
$$\text{On } \hat{\Gamma}_4 : \psi_4(\pi_4(\hat{x},\hat{y}),\mu) = \Big(arclen^{-1}(\pi_4(\hat{x},\hat{y})), y_3\Big)$$

Remark 10. *The offline/online decomposition of the global method will strongly depend on the way we pick the set of ϕ_i's, π_i's and ψ_i's.*

Remark 11. *This 'smarter' choice of functions not only makes the G-H mapping smoother but it also makes*

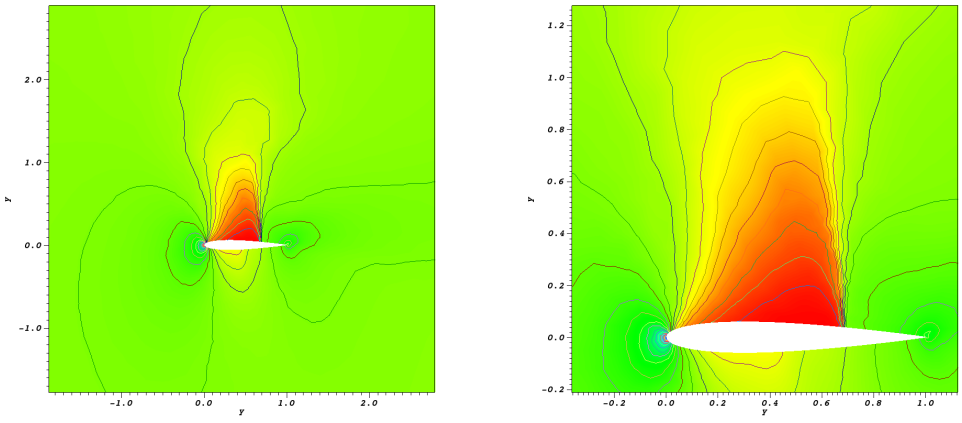
$$\begin{array}{ccc} \mathcal{D} & \mapsto & \mathcal{F} \\ \mu & \to & F_\mu \end{array}$$

smoother. This can be an interesting property in a optimal control context (see [22]).

We test this new mapping. We present the results in Figure 19. This has to be compared with the truth solution, presented in Figure 12. We can see that we are still straightening the shock. Also we have removed the non physical behavior at the boundary that we had in the raw G-H scenario (see Figure 14).

Remark 12. *One must not forget that this case is no different from the flat boundary scenario of subsection 6.1. The fact that the reference domain has the same body as the physical domain was just needed for smoothness of the mapping purposes.*

Before a more involved test run, with a more quantitive analysis of the results, we present yet another type of mapping. This goes one step further in building a smooth at boundaries mapping.



(a) (b)

Figure 19: The mapped solution for velocity component on a curved domain

One recent development on transfinite maps is defined in [31] and is called boundary displacement dependent transfinite map (BDD TM). The idea is not to construct the whole mapping, but to construct a relative displacement with respect to the identity. Most of the method is the same, the only difference is that instead of ψ_i function, which represent the position on the physical domain, a new function $d_i : [0,1] \times \mathcal{D} \to \mathbb{R}$ is introduced and it will represent the displacement:

$$d_i(t,\mu) = \psi_i(t,\mu) - \hat{\psi}_i(t),$$

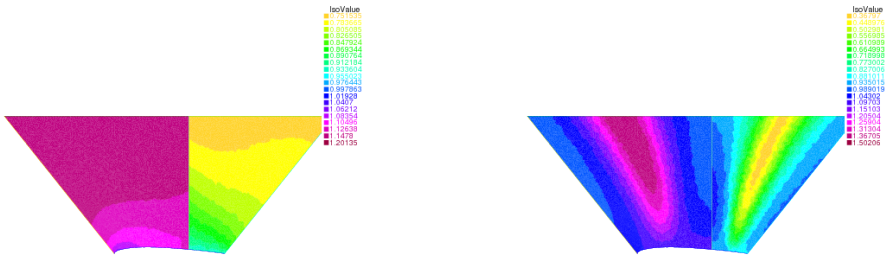
where each of the boundaries in the reference domain is parametrized by $\hat{\psi}_i : [0,1] \to \mathbb{R}$. Like this, the mapping will keep into account the original positions of the points in the reference domain $\hat{\Omega}$ and will move them by weighting only the difference between the original boundaries and the deformed ones. Now, if $(\hat{x},\hat{y})$ is a point in the reference domain $\hat{\Omega}$, the idea of BDD TM is to displace it through the quantity $(\hat{x},\hat{y}) + \sum_{i=1}^{n} \phi_i(\hat{x},\hat{y}) d_i(\pi_i(\hat{x},\hat{y}),\mu)$. In the end, the BDD transfinite mapping is defined as:

$$GH_{BDDTM}(\hat{x},\hat{y}) = (\hat{x},\hat{y}) + \sum_{i=1}^{n} \Big(\phi_i(\hat{x},\hat{y}) d_i(\pi_i(\hat{x},\hat{y}),\mu) - \phi_i(\hat{x},\hat{y})\phi_{i+1}(\hat{x},\hat{y}) d_i(1,\mu) \Big) \tag{28}$$

This has one major effect, on the left boundary for instance, where we want zero displacement. The resulting mapping restricted to a neighborhood of this boundary will be the identity, which guarantees the matching with the outside domain.

Remark 13. *The ameliorations on ϕ's and ψ's presented above will still apply to the BDD TM.*

We will illustrate numerically the gain from these ameliorations. We present in Figure 20 the comparaison between the original method and the final one, taylored for our specific application. We show the entry of N^T that varies the most, i.e $\frac{x}{\hat{x}}$.



(a) (b)

Figure 20: One of the entries of the Jacobian matrix, namely $(J_{F_n^{-1}})_{11}$. Left : with no additional smoothing ingredients; Right : with some smoothing ingredients

6.3 Final experiment

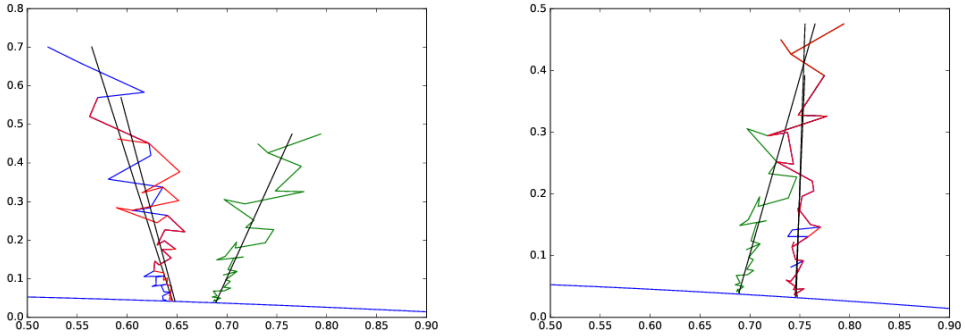
A full implementation of the method described so far is out of the scope of this paper. We will instead present numerical evidence on the new aspect of this method, which is the resolution of a CFD problem on a mapped mesh. We will show in this section, that by playing with the coefficients of the mapping, we can put the shock anywhere we want inside $\hat{\Omega}$. More precisely, we will construct several mappings G-H(a_0^k, a_1^k) and launch the modified CFD code. We insist that the mapping will be constant throughout each simulation.

We can then compare the recovered position of the shock with the mapped 'truth' solution. That is, we compare $u^{truth} \circ \left(\text{G-H}(a_0^k, a_1^k)\right)^{-1}$ with the output of our algorithm using the modified flux. Some results are presented in Figure 21. Blue represents the truth solution mapped onto the reference domain. Red is the output of the CFD code. Green is the original position of the shock. We have fitted one degree polynomials through each shock. The differences between our result and the mapped truth scheme can be due to many factors:

- numerical errors on the computation of the N^T terms;
- the SUPG stabilization has not been touched, to avoid too much intrusion in the code. This means that we are not using the same stabilization procedure as the truth scheme. We refer to [21] for a study of this situation. They advise using an a posteriori procedure, called rectification;
- our method to locate the shock is basic. We would need something more involved to quantify the error

7 Conclusion

The purpose of this paper was to propose a complete calibration procedure to make standard ROM methods fitted for solving the two dimensional Euler equation around an airfoil. We have proposed



(a) (b)

Figure 21: Comparaison of the outputs

an offline calibration procedure, have shown numerically that it reduces the Kolmogorov n-width and leads to non oscillatory basis. We have then proposed a fully functioning reduced scheme. The computational complexity and the optimization procedures have been theoretically studied. We have finally developed numerical experiments that serve as a proof of concept for the global method.

Most of the stages in this paper can be further investigated. Future work could involve :

- a deeper study of the offline calibration and it's effect on the Kolmogorov n-width. We have proposed online some advanced mappings, where we impose no stretching in the vicinity of a shock. A numerical investigation can be done.
- a fully reduced scheme, with the procedure advised in Section 4 has to be constructed. This could lead to an interesting comparison between L^1 and L^2 minimization. Also, the optimization procedure can be studied. The differentiability issues can be also tested.
- the hyper reduction procedure can be numerically investigated. The conjectures made on the resulting $\hat{\Omega}_{hyper}$, namely that the interesting control volumes are close to the shock, which is fixed in $\hat{\Omega}$ can be tested as the subject of a future work.
- the smoothness of the entries of the N^T matrix has to be studied. All the smoothing ingredients proposed in section 6 could be further investigated.
- a more long term objective could be to use this method to try new airfoils shapes. As mentioned, the G-H is a very flexible algorithm. We could use the NACA 00012 as reference domain, but use fantasist choices for physical domain. This is a very well fitted framework for optimal control (for more details, see Chapter 5 of [22]).

Appendix

Let's formulate a problem in a general setting. Let two domains Ω and $\hat{\Omega}$ and F_n some smooth mapping between them. $\hat{w} = F_n(w) \Rightarrow w = F_n^{-1}(\hat{w})$. We have some generic smooth flux defined on Ω, denoted $f : \Omega \to \mathbb{R}^2$. The objective is to prove that there exists a vector field $\tilde{f} : \hat{\Omega} \to \mathbb{R}^2$, function of f and of F_n such that

$$\nabla_w \cdot f = \frac{1}{J_{F_n^{-1}} \circ F_n} \left(\nabla_{\hat{w}} \cdot \tilde{f} \right) \circ F_n \text{ on } \Omega$$

where as usual, J_F denotes the Jacobian of a the mapping F. Here

$$J_{F_n^{-1}}(\hat{w}) = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix}.$$

We will first compute the representation of f in terms of curvilinear coordinates and the frame

$$\boldsymbol{\rho}_1(\hat{w}) = \left(\frac{\partial x}{\partial \hat{x}}, \frac{\partial y}{\partial \hat{x}} \right), \boldsymbol{\rho}_2(\hat{w}) = \left(\frac{\partial x}{\partial \hat{y}}, \frac{\partial y}{\partial \hat{y}} \right)$$

associated with them. Thus, we treat the vector field $f(w)$ by first expressing it in the form: $\mathbf{f} \circ F_n^{-1}(\hat{w}) = \hat{f}_1(\hat{w})\boldsymbol{\rho}_1 + \hat{f}_2(\hat{w})\boldsymbol{\rho}_2$. On the other side, we know that the fluxes are $f(w) = (f_1, f_2)$. Then,

$$\Rightarrow \begin{pmatrix} f_1 \circ F_n^{-1} \\ f_2 \circ F_n^{-1} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{pmatrix} \begin{pmatrix} \hat{f}_1 \\ \hat{f}_2 \end{pmatrix}$$

Solving the system of equations, we obtain the following values for $\hat{f}_1$ and $\hat{f}_2$:

$$\hat{f}_1(\hat{w}) = \frac{1}{\frac{\partial x}{\partial \hat{x}}\frac{\partial y}{\partial \hat{y}} - \frac{\partial x}{\partial \hat{y}}\frac{\partial y}{\partial \hat{x}}} \left(\frac{\partial y}{\partial \hat{y}} f_1 \circ F_n^{-1} - \frac{\partial x}{\partial \hat{y}} f_2 \circ F_n^{-1} \right)$$

$$\hat{f}_2(\hat{w}) = \frac{1}{\frac{\partial x}{\partial \hat{x}}\frac{\partial y}{\partial \hat{y}} - \frac{\partial x}{\partial \hat{y}}\frac{\partial y}{\partial \hat{x}}} \left(-\frac{\partial y}{\partial \hat{x}} f_1 \circ F_n^{-1} + \frac{\partial x}{\partial \hat{x}} f_2 \circ F_n^{-1} \right)$$

We have

$$\nabla_w \cdot \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \nabla_w \cdot (\hat{f}_1 \circ F_n \ \boldsymbol{\rho}_1 \circ F_n) + \nabla_w \cdot (\hat{f}_2 \circ F_n \ \boldsymbol{\rho}_2 \circ F_n)$$

Let $i \in \{1, 2\}$.

$$\begin{aligned} \nabla_w \cdot (\hat{f}_i \circ F_n \ \boldsymbol{\rho}_i \circ F_n) &= \nabla_w \left(J_{F_n^{-1}} \circ F_n \ \hat{f}_i \circ F_n \right) \cdot \left(\frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n \right) \\ &+ \left(J_{F_n^{-1}} \circ F_n \ \hat{f}_i \circ F_n \right) \nabla_w \cdot \left(\frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n \right) \end{aligned}$$

The second operand is zero. We will sketch the proof and refer to [26] for a rigorous one. We will consider ϕ smooth real valued test functions with compact support in Ω. We want to show the following :

$$\forall \phi \in \mathcal{D}(\Omega), \ \int_\Omega \phi(w) \nabla_w \cdot \left(\frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n \right) dw = 0$$

For this, use product rule :

$$\begin{aligned}\forall \phi \in \mathcal{D}(\Omega),\ \int_\Omega \phi(w) \nabla_w \cdot (\tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n) dw \quad &= \int_\Omega \nabla_w \cdot (\phi(w) \tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n dw) \\ &- \int_\Omega \tfrac{1}{J_{F_n^{-1}} \circ F_n} (\boldsymbol{\rho}_i \circ F_n) \cdot \nabla_w \phi(w) dw\end{aligned}$$

We have one easy equality :

$$\int_\Omega \nabla_w \cdot \left(\phi(w) \frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n \right) dw = 0$$

This uses the divergence theorem and the fact that ϕ has a compact support in Ω. The other equality requires more attention :

$$\begin{aligned}\int_\Omega \tfrac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_i \circ F_n(w) \cdot \nabla_w \phi(w) \quad &= \int_{\hat{\Omega}} \tfrac{1}{J_{F_n^{-1}}} \boldsymbol{\rho}_i \cdot \nabla_w \left(\phi \circ F_n^{-1} \right)(\hat{w}) |J_{F_n^{-1}}| d\hat{w} \\ &= \pm \int_{\hat{\Omega}} \boldsymbol{\rho}_i \cdot \nabla_w \left(\phi \circ F_n^{-1} \right)(\hat{w}) d\hat{w}\end{aligned}$$

We will use the following equality two times :

$$\forall \psi \in \mathcal{D}(\hat{\Omega}),\ \nabla_w \psi \cdot \boldsymbol{\rho}_1 = \frac{\partial \psi}{\partial x} \frac{\partial x}{\partial \hat{x}} + \frac{\partial \psi}{\partial y} \frac{\partial y}{\partial \hat{x}} = \frac{\partial \psi}{\partial \hat{x}} \tag{29}$$

This is precisely the reason why we have defined the ρ_i in the first place. The previous equality becomes :

$$\int_\Omega \frac{1}{J_{F_n^{-1}} \circ F_n} \boldsymbol{\rho}_1 \circ F_n(w) \cdot \nabla_w \phi(w) = \int_{\hat{\Omega}} \frac{\partial (\phi \circ F_n^{-1})}{\partial \hat{x}} (\hat{w}) d\hat{w}$$

We conclude using the fact that $\phi \circ F_n^{-1}$ has compact support in $\hat{\Omega}$.

We finally have the annouced result :

$$\forall i \in \{1, 2\},\ \nabla_w \cdot (\hat{f}_i \circ F_n\ \boldsymbol{\rho}_i \circ F_n) = \frac{1}{J_{F_n^{-1}} \circ F_n} \nabla_w \left(J_{F_n^{-1}} \circ F_n \hat{f}_i \circ F_n \right) \cdot (\boldsymbol{\rho}_i \circ F_n)$$

Using the same argument as in (29), we have

$$\sum_{i=1}^{2} \nabla_w \cdot (\hat{f}_i \circ F_n\ \boldsymbol{\rho}_i \circ F_n) = \frac{1}{J_{F_n^{-1}} \circ F_n} \nabla_{\hat{w}} \left(J_{F_n^{-1}} \hat{f}_i \right) \circ F_n$$

We conclude

$$\nabla_w \cdot f = \frac{1}{J_{F_n^{-1}} \circ F_n} \nabla_{\hat{w}} \cdot \left(N^T f \right) \circ F_n$$

We have found our transformed flux function: $\tilde{f} := N^T \cdot f$, where

$$N^T = \begin{bmatrix} (J_{F_n^{-1}})_{22} & -(J_{F_n^{-1}})_{12} \\ -(J_{F_n^{-1}})_{21} & (J_{F_n^{-1}})_{11}. \end{bmatrix}$$

In the end, we obtain the following finite volume formulation (also known as Piola transformation [8]):

$$\int_{\hat{w}_i} \hat{u}(\hat{w}, t^{n+1})|J_{F_n^{-1}}(\hat{w})|d\hat{w} - \int_{\hat{w}_i} \hat{u}(\hat{w}, t^n)|J_{F_n^{-1}}(\hat{w})|d\hat{w} + \int_{\hat{w}_i}\int_{t^n}^{t^{n+1}} \frac{1}{|J_{F_n^{-1}}|}\nabla_{\hat{w}} \cdot (N^T f(\hat{w}))|J_{F_n^{-1}}|dtd\hat{w} = 0$$

$$\Leftrightarrow \int_{\hat{w}_i} \hat{u}(\hat{w}, t^{n+1})|J_{F_n^{-1}}(\hat{w})|d\hat{w} - \int_{\hat{w}_i} \hat{u}(\hat{w}, t^n)|J_{F_n^{-1}}(\hat{w})|d\hat{w} + \int_{\hat{w}_i}\int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N^T f(\hat{w}))dtd\hat{w} = 0$$

References

[1] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, 2016.

[2] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, 2016.

[3] J. Melenk, "On n-widths for elliptic problems," *Elsevier*, vol. 247, no. 1, pp. 272–289, 2000.

[4] A. Cohen and R. DeVore, "Kolmogorov widths under holomorphic mappings," *IMA Journal of Numerical Analysis*, vol. 36, no. 1, pp. 1–12, 2016.

[5] Y. Maday, A. Patera, and G. Turinici, "A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations," *Journal of Scientific Computing*, vol. 17, no. 1-4, pp. 437–446, 2002.

[6] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, "Convergence rates for greedy algorithms in reduced basis methods," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 3, pp. 1457–1472, 2011.

[7] R. DeVore, G. Petrova, and P. Wojtaszczyk, "Greedy algorithms for reduced bases in banach spaces," *Constructive Approximation*, vol. 37, no. 3, pp. 455–466, 2013.

[8] A. Løvgren, Y. Maday, and E. Rønquist, *The Reduced Basis Element Method for Fluid Flows*, 2006.

[9] W. Beyn and V. Thümmler, "Freezing solutions of equivariant evolution equations," *SIAM Journal on Applied Dynamical Systems*, vol. 3, no. 2, pp. 85–116, 2004.

[10] M. Ohlberger and S. Rave, "Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing," *Comptes Rendus Mathématique*, vol. 351, no. 23, pp. 901–906, 2013.

[11] N. Cagniart, Y. Maday, and B. Stamm. (2016, Nov.) Model order reduction for problems with large convection effects. [Online]. Available: http://hal.upmc.fr/hal-01395571

[12] A. Iollo and D. Lombardi, "Advection modes by optimal mass transfer," *Physical Review E*, vol. 89, no. 2, p. 022923, 2014.

[13] G. Welper, "Interpolation of functions with parameter dependent jumps by transformed snapshots," *SIAM Journal on Scientific Computing*, vol. 39, no. 4, p. A1225A1250, 2017.

[14] D. Rim, S. Moe, and R. J. LeVeque. (2017, Jan.) Transport reversal for model reduction of hyperbolic partial differential equations. [Online]. Available: arXiv:1701.07529

[15] R. Abgrall, D. Amsallem, and R. Crisovan, "Robust model reduction by L1-norm minimization and approximation via dictionaries: application to nonlinear hyperbolic problems," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, no. 1, 2016.

[16] R. Abgrall, "Residual distribution schemes: current status and future trends," *Computers & Fluids*, vol. 35, no. 7, pp. 641–669, 2006.

[17] H. Deconinck and M. Ricchiuto, "Residual distribution schemes: foundations and analysis," *Encyclopedia of computational mechanics*, 2007.

[18] K. Siddiqi, B. B. Kimia, and C.-W. Shu, "Geometric shock-capturing eno schemes for subpixel interpolation, computation, and curve evolution," in *Computer Vision, 1995. Proceedings., International Symposium on Computer Vision - ISCV*. IEEE, 1995, pp. 437–442.

[19] W. J. Gordon and C. A. Hall, "Transfinite element methods: blending-function interpolation over arbitrary curved element domains," *Numerische Mathematik*, vol. 21, no. 2, pp. 109–129, 1973.

[20] A. E. Løvgren, Y. Maday, and E. M. Rønquist, "Global c1 maps on general domains," *Mathematical Models and Methods in Applied Sciences*, vol. 19, no. 5, pp. 803–832, 2009.

[21] Y. Maday, A. Manzoni, and A. Quarteroni, "An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems," *Comptes Rendus Mathématique*, vol. 354, no. 12, pp. 1188–1194, 2016.

[22] Cagniart, "N. Cagniart PhD," 2017.

[23] K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient non-linear model reduction via a least-squares petrovgalerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 2, pp. 155–181, 2011.

[24] A. Quarteroni and G. Rozza, "Numerical solution of parametrized navier–stokes equations by reduced basis methods," *Numerical Methods for Partial Differential Equations*, vol. 23, no. 4, pp. 923–948, 2007.

[25] A. Løvgren, Y. Maday, and E. Rønquist, "The reduced basis element method: Offline-online decomposition in the nonconforming, nonaffine case," in *Spectral and High Order Methods for Partial Differential Equations*. Springer, 2011, pp. 247–254.

[26] P. Colella, M. R. Dorr, J. A. Hittinger, and D. F. Martin, "High-order, finite-volume methods in mapped coordinates," *Journal of Computational Physics*, vol. 230, no. 8, pp. 2952–2976, 2011.

[27] C. Bardos and O. Pironneau, "Derivatives and control in the presence of shocks," *Computational Fluid Dynamics Journal*, vol. 11, no. 4, pp. 383–391, 2003.

[28] N. Allahverdi, A. Pozo, and E. Zuazua, "Numerical aspects of large-time optimal control of burgers equation," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 50, no. 5, pp. 1371–1401, 2016.

[29] R. Everson and L. Sirovich, "Karhunen–loeve procedure for gappy data," *JOSA A*, vol. 12, no. 8, pp. 1657–1664, 1995.

[30] D. Ryckelynck, "Hyper-reduction of mechanical models involving internal variables," *International Journal for Numerical Methods in Engineering*, vol. 77, no. 1, pp. 75–89, 2009.

[31] C. Jäggli, L. Iapichino, and G. Rozza, "An improvement on geometrical parameterizations by transfinite maps," *Comptes Rendus Mathématique*, vol. 352, no. 3, pp. 263–268, 2014.