

A DIFFUSE-MORPHING ALGORITHM FOR SHAPE OPTIMIZATION

Balaji Raghavan¹, Piotr Breitkopf¹, Pierre Villon¹

¹ Laboratoire Roberval UMR 7337 UTC-CNRS, Labex MS2T
 {balaji.raghavan,piotr.breitkopf,pierre.villon}@utc.fr

Abstract. *Shape optimization typically involves geometries characterized by several dozen design variables and a possibly high number of explicit/implicit constraints restricting the design space to admissible shapes. In this work, instead of working with parametrized CAD models, the idea is to interpolate between admissible instances of finite element/CFD meshes. We show that a properly chosen surrogate model can replace the numerous geometry-based design variables with a more compact set permitting a global understanding of the admissible shapes spanning the design domain, thus reducing the size of the optimization problem. To this end, we present a two-level mesh parametrization approach for the design domain geometry based on Diffuse Approximation in a properly chosen locally linearized space, and replace the geometry-based variables with the smallest set of variables needed to represent a manifold of admissible shapes for a chosen precision. We demonstrate this approach in the problem of designing the section of an A/C duct to maximize the permeability evaluated using CFD.*

Keywords: *Model reduction, CFD, Diffuse Approximation, rasterization.*

1. Nomenclature

S^i	i^{th} geometric snapshot	\bar{X}	vector of geometric parameters
$\Phi, \bar{\phi}_i$	POD basis, i^{th} mode vector	\bar{L}_B, \bar{U}_B	lower/upper bounds on \bar{X}
$\bar{\alpha}$	vector of PCA coefficients	$\bar{\alpha}^{opt}$	optimal solution in α -space
m	number of modes retained	$\tilde{S}(\bar{\alpha}^{opt})$	Shape corresponding to $\bar{\alpha}^{opt}$
M	Number of snapshots	C_v	Covariance matrix of snapshots
N_c	size of pixel map	\tilde{S}	shape approximation with m modes
P_{flow}	flow permeability	\bar{S}	mean snapshot
h, b	density filter parameters	λ_i	i^{th} eigenvalue of C_v
\bar{t}	reduced design parameters	p	Number of reduced design parameters

2. INTRODUCTION

Shape optimization may be viewed as the task of combining a parameterized geometric model with a numerical simulation code in order to predict the geometric state that minimizes

a given cost function while respecting a set of equality/inequality constraints. In this paper we consider the task of shape/mesh interpolation or hypothesizing the structure, which occurs between shape/mesh instances given by a sequence of parameter values. The need for this arose during the development of multidisciplinary optimization techniques, because CAD parameterized models involved in automatized computing chains suffered from excessive design space dimensionality eventually leading to crashes of either the mesh generator or the solver. This phenomenon is due to the difficulties in expressing all the technological and common sense constraints (needed to convert a set of geometric parameters to an admissible shape) within existing parameterization methods.

Most current approaches to shape parameterization require hand-constructed CAD models. We are interested in developing an alternative approach in which the interpolation system builds up structural shapes automatically by learning from existing examples. One of the central components of this kind of learning is the abstract problem of inducing a smooth non-linear constraint manifold from a set of the examples, called "Manifold Learning" by Bregler [2] who developed approaches closely related to neural networks for doing it. [8] proposed a similar approach in the domain of Reduced Order Modeling (ROM) for complex flow problems. In this paper we apply manifold learning to the shape interpolation problem to develop a parametrization scheme tailored to the structural optimization problem (e.g. airplane wing, A/C duct, engine inlet, etc)

Several techniques [5,26] have been used to replace a complicated numerical model by a lower-order meta-model, usually based on polynomial response surface methodology (RSM), kriging, least-squares regression and moving least squares [4]. Surrogate functions and reduced-order meta-models have also been used in the field of control systems to reduce the order of the overall transfer function [26]. A very popular physics-based meta-modeling technique consists of carrying out the approximation on the full vector fields using PCA and Galerkin projection [1] in CFD [21,27] as well as in structural analysis [12] and has been successfully applied to a number of areas such as flow modeling [23,13], optimal flow control [21], aerodynamics design optimization [18,11] or structural mechanics [14]. In [7], a snapshot-weighting scheme introduced using vector sensitivities as system snapshots to compute a robust reduced order model well-suited to optimization. [8] also demonstrated a goal-oriented local POD approach that is computationally less expensive than using a global POD approach.

However, we have not observed much if any research into using decomposition-based surrogate models to reducing dimensionality of the design domain in shape optimization, and for that matter, structural optimization of any type. This area, we feel is promising considering the obvious advantages of having far fewer parameters describing the domain: easier visualization, more flexibility in the choice of admissible shapes, better applicability to gradient-based solvers due to reduced dimensionality and thus a reduction in the overall size of the optimization, and of course a separation between the CAD and the optimization phases in system design by giving the optimization group a protocol to reparametrize structural shapes for a given set of admissible shapes/meshes that can be generated by the CAD group, and using the presented algorithm (or a variant thereof) on these to get the new set of design variables.

In this paper, we present what can best be described as a manifold learning approach combining Diffuse Approximation and Principal Component Analysis, whose performance is easily

compared to that of simple linear interpolation, classical morphing [25] and a posteriori mesh parametrization [9].

We propose a four-step "a posteriori" reparametrization approach to reduce the number of design variables needed while describing the shape of a structure:

- Pixellization: the protocol first uses the method of snapshots to generate M admissible shapes (or read a set of structural meshes) sweeping the design space. In order to obtain an indicator function for the design domain, a step called "pixellization" is next performed by mapping the snapshot boundaries/edges onto a reference grid with a certain resolution, to be then stored as a binary array S^i of 0's and 1's, as is typically done in image-storing/manipulation [15].

- Decomposition of the M snapshots by Principal Components Analysis.

- Two-level dimensionality reduction: In the first reduction phase, the snapshot "pixel arrays" (or "voxels" in 3D) are then reduced to obtain a small number of dominant basis vectors $(\bar{\phi}_1 \dots \bar{\phi}_m)$ spanning the physical design domain, and the vector of coefficients $\bar{\alpha} \in R^m$, $m \ll M$ is then obtained by projecting a structural shape onto the basis Φ .

In the second reduction phase, the coefficients $\alpha_1 \dots \alpha_m$ corresponding to the snapshots are analyzed to understand the shape of the feasible region, allowing us to deduce the true dimensionality of the physical design domain. A Diffuse Approximation performed in the α -space gives the final minimal set of parameters $t_1 \dots t_p$, $p \leq m$, thus our approach involves a two-level model reduction. Since they have been obtained from an "a posteriori" sweep of the design domain followed by decomposition, these new variables can be directly used in an optimization algorithm to obtain the optimal shape (pixel array) for a given performance objective.

- Shape Interpolation to obtain a smooth structural shape from \bar{t} .

The methodology is described in the next section with the overall algorithm, and the test-case from the automotive field, the numerical model used to calculate the objective function are then described in section 4. The optimization problem is formally presented in section 5. Section 6 presents some results with a discussion of the different stages, and we close with a discussion of possible future work.

3. A POSTERIORI GRID PARAMETRIZATION METHODOLOGY

3.1. Creation of snapshots

We build the parametrization scheme after studying the full range of admissible shapes (i.e. snapshots [10]) constituting the design domain. For structural optimization problems for a fixed topology, these admissible shapes could be obtained in a Lagrangian description by a sampling of the geometry-based design variables within their feasible range $\bar{X} \in [\bar{L}_B, \bar{U}_B] \subset R^N$, or simply from the finite set of points describing the edges/boundaries of a series of CFD meshes/grid points for an initial random sampling of M designs.

3.2. Pixelization of snapshots

This step refers to mapping the edges/boundaries for each snapshot onto a reference grid and store it as a binary array [15]. This is typically performed by finding the cells (in the reference grid) penetrated by the edges/boundary of the structure or mesh and assigning a

value 1 to these boundary cells as well as all the cells inside the boundary cells as shown in figure 1, and 0 to the cells outside the boundary, thus allowing us to store the pixel maps as arrays ($S^i \in \mathcal{R}^{N_c}, i = 1..M$) of 1s and 0s. It goes without saying that pixelization captures the actual shape better with higher resolution compared to a lower resolution.

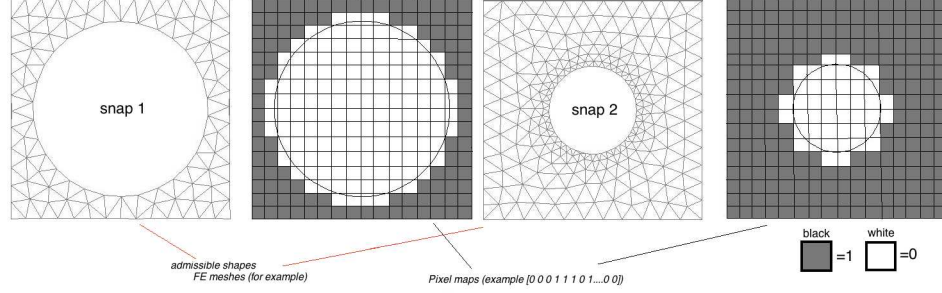


Figure 1. Reference grid mapping for pixel map: plate with circular hole

3.3. Principal Components Analysis

This is the first phase of model reduction. We first calculate the deviation matrix D_S for the snapshots using:

$$D_s = [S^1 - \bar{S} \quad S^2 - \bar{S} \quad \dots \quad S^M - \bar{S}] \quad (1)$$

where $M \ll N_c = \text{number of snapshots}$, $S^i = i\text{th individual snapshot binary array (pixel map)}$ and \bar{S} is the mean of all the snapshots. Next, the covariance matrix C_v is calculated

$$C_v = D_S \cdot D_S^T \quad (2)$$

allowing us to express any S^j in terms of the eigenvectors $\bar{\phi}_i$ of C_v .

$$S^j = \bar{S} + \sum_{i=1}^M \alpha_{ij} \bar{\phi}_i, \alpha_{ij} = \bar{\phi}_i^T S^j \quad (3)$$

for the j th pixel map.

In the first reduction phase, we limit the basis to the first $m \ll M$ most "energetic" modes

$$\tilde{S}^j = \bar{S} + \sum_{i=1}^m \alpha_{ij} \bar{\phi}_i \text{ and } \epsilon(m) = 1 - \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^M \lambda_i} \quad (4)$$

3.4. Model reduction and design domain dimensionality

Equation (4) does not provide a sufficient basis for establishing the value of m as one needs to specify the threshold value for ϵ . Also the α_{ij} may not be taken as design variables without taking into account the possible relationships between them so as to render feasible shapes.

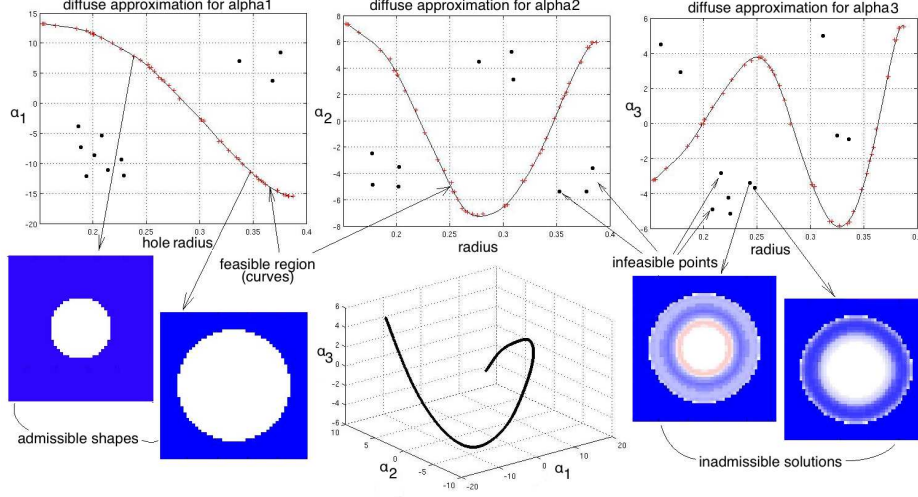


Figure 2. Feasible region for a plate with a circular hole of varying radius

Let us consider the same system (plate with circular hole $R_{min} \leq r \leq R_{max}$). Ignoring the fact that the dimensionality is 1, we construct 50 random snapshots by varying the radius r . The pixelization and PCA are then performed in succession giving us a set of α 's corresponding to each snapshot. As illustrated in figure 2, the α 's form a set of one-dimensional manifolds, clearly indicating that the design domain is parametrized by ONE single parameter t , which in this case happens to be the hole radius (in the general case we obtain a vector $\bar{t} \in R^p, p \leq m$), i.e. $\alpha_1 = \alpha_1(t), \alpha_2 = \alpha_2(t) \dots$. These manifolds are easily obtained by performing a Diffuse Approximation [4,20] over all the $\alpha_1 \dots \alpha_M$ obtained from snapshots S_1 to S_M . Furthermore, the curves of $\alpha_1, \alpha_2, \dots$ vs t may be interpreted as possible "constraints" (direct geometric constraints, technological constraints etc that are difficult to express mathematically) on the geometric parameters \bar{X} (here simply r) in the α -space, since points lying outside the manifolds will produce inadmissible shapes as shown. Thus, in the second reduction phase, we locally introduce the parametric expression of the α -manifolds.

3.5. Manifold approximation and updating

We present here a formal approach to locally identify the system dimensionality from the α -manifolds. Consider a system of M pixel snapshots converted to the PCA-space retaining $m < M$ coefficients thus giving us a set of points $\bar{\alpha}^1, \dots, \bar{\alpha}^M \in \mathcal{R}^m$. We would like to implement an algorithm that:

1. Detects the "true" dimensionality ($p \leq m$) from the local rank of the α -manifold in the vicinity of the evaluation point, so that the feasible region may (locally) be expressed as $\alpha_1 = \alpha_1(t_1 \dots t_p), \dots, \alpha_m = \alpha_m(t_1 \dots t_p)$.
2. Constrains the evaluation point ($\bar{\alpha}^{ev}$) to stay on the feasible region of admissible shapes, during the course of the optimization.

3.5.1 Local Rank Detection of α -manifold

To locally detect the dimensionality of the $\alpha_1 \dots \alpha_m$ hyper-surface in the neighborhood of $\bar{\alpha}^{ev}$, we first establish the local neighborhood, this may be done in the original geometric space (if available) or, if the original parameters are unavailable which is what this approach is intended for, by using the α values if the neighborhood is sufficiently dense. So if $\bar{\beta}_1 \dots \bar{\beta}_{nbd}$ are neighboring points in α -space, we next use a polynomial basis centered around the evaluation point

$$P = \begin{bmatrix} 1 & \beta_1^1 - \alpha_1^{ev} & \beta_2^1 - \alpha_2^{ev} & \dots & \beta_m^1 - \alpha_m^{ev} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta_1^{nbd} - \alpha_1^{ev} & \beta_2^{nbd} - \alpha_2^{ev} & \dots & \beta_m^{nbd} - \alpha_m^{ev} \end{bmatrix} \quad (5)$$

with an appropriate weighting function, for example gaussian $w(d) = \exp(-c * d^2)$ and assemble the moment matrix $A = P^T W P$, where W is the diagonal matrix whose elements correspond to the weighted contributions of the nodes $\bar{\beta}_1 \dots \bar{\beta}_{nbd}$.

Next, we detect the local rank of the manifold by calculating the singular values of the moment matrix A , this gives us the dimensionality $p \leq m$.

To demonstrate this, we consider again the M snapshots for a plate with a circular hole of varying radius. Considering the first three modes, the points corresponding to the snapshots are shown to the left in figure 3. We assemble the moment matrix A and calculate its rank = 1 (only one significant value as seen in the middle figure 3) and thus the dimensionality of the plate with circular hole of varying radius is $p = 1$, thus allowing us to parametrize the curve with a single parameter $\alpha_1 = \alpha_1(t_1), \dots \alpha_m = \alpha_m(t_1)$ as seen in the third diagram in the same figure.

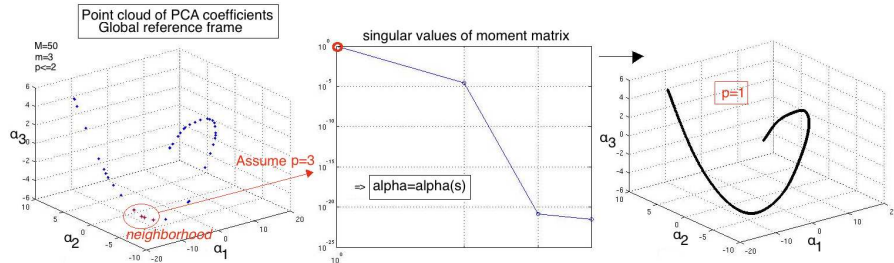


Figure 3. Detecting dimensionality for a plate with a circular hole of varying radius

3.5.2 Tangent plane construction and Diffuse "walking"

The idea is to bring the current design point given by the optimization algorithm in subsequent iterations, down to the surface, which represents locally the manifold of admissible shapes. The local surface tangent to the manifold is defined with respect to the tangent plane iteratively updated. To achieve this, we use a Diffuse Approximation-based manifold "walking" scheme consisting of the following steps shown in figure 4.

1. Let P_i be the evaluation point (on the α -manifold), and P_{i+1}^0 be the new candidate point

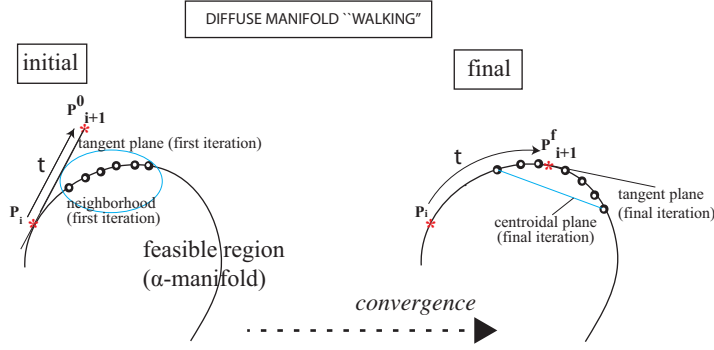


Figure 4. Walking the evaluation point along the α -manifold using diffuse approximation and tangent plane construction

(that needs to be brought back on to the manifold/feasible region). We first establish the neighborhood $\bar{\beta}_1 \dots \bar{\beta}_{nbd}$ of P_{i+1}^0 .

2. Calculate the centroid $\bar{\beta}_m = (\sum_{i=1}^{nbd} \bar{\beta}_i) / nbd$.

3. Find the centroidal plane for the neighborhood from the eigenvectors of the covariance matrix C_{nbd} , the first eigenvector representing the plane normal:

$$C_{nbd} = (1/nbd) \sum_{i=1}^{nbd} (\bar{\beta}_i - \bar{\beta}_m)(\bar{\beta}_i - \bar{\beta}_m)^T, (\bar{v}_1, \bar{v}_2, \dots) = \text{eigenvectors}(C_{nbd}) \quad (6)$$

4. Project the evaluation point as well as the neighborhood points in the local coordinate system $\bar{v}_1, \bar{v}_2, \dots$ (origin at centroid $\bar{\beta}_m$) to get the local co-ordinates h, t_1, t_2, \dots, t_p where h is the height over the centroidal plane using the equations (for a general point $\bar{\alpha}$).

$$\begin{aligned} h &= \bar{v}_1^T (\bar{\alpha} - \bar{\beta}_m) \\ t_1 &= \bar{v}_2^T \bar{\alpha}, \text{ etc} \end{aligned} \quad (7)$$

5. Perform a diffuse approximation for the nbd points, to obtain the local surface $h = h(t_1 \dots t_p)$ using a polynomial basis P centered around $\bar{\alpha}^{ev}$, with a weighting matrix W .

$$\begin{bmatrix} h(t_1^{ev}, \dots) \\ \frac{\partial h}{\partial t_1}(t_1^{ev}, \dots) \\ \frac{\partial h}{\partial t_2}(t_1^{ev}, \dots) \\ \dots \end{bmatrix} = (P^T W P)^{-1} P^T W \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_{nbd} \end{bmatrix} \quad (8)$$

where $[\frac{\partial h}{\partial t_1}(t_1^{ev}, \dots), \frac{\partial h}{\partial t_2}(t_1^{ev}, \dots), \dots]^T$ is the local tangent hyper-plane at P_0 in the neighborhood $\bar{\beta}^1 \dots \bar{\beta}^{nbd}$.

6. We then project the point P_{i+1}^0 onto this tangent plane to get the adjusted evaluation point P_{i+1}^1 , and then repeat the process by finding the new neighborhood and new tangent plane and new projection point P_2 till the evaluation point stops changing P_{i+1}^f .

In other words, we "walk" along the surface of the α -manifold to ensure that we stay in the domain of feasible solutions. To illustrate the approach, consider the α coefficients obtained for a plate with two circular holes of varying radii r_1 and r_2 , with $M = 1000$ snapshots, shown in figure 5. We see clearly from the shape of the manifolds, that all the parameters $\alpha_1 \dots \alpha_M$

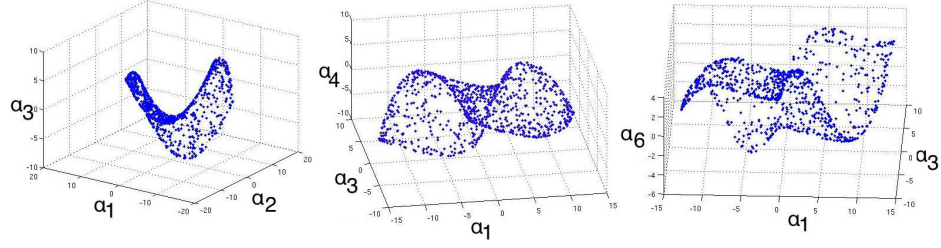


Figure 5. α -manifolds for plate with two circular holes of fixed centers and independently varying radii

are locally controlled by two parameters ($p = 2$) and this is verified by the rank detection explained earlier. We perform the diffuse approximation locally with $m = 3$ and project the evaluation point onto the tangent plane, and the main steps are shown in figure 6.

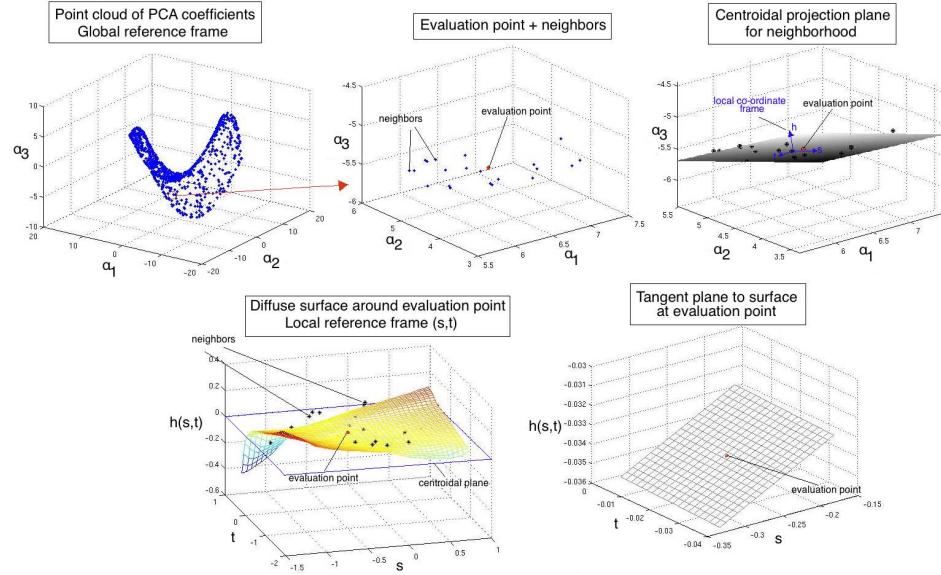


Figure 6. Diffuse α -manifold walking for the plate with two circular holes example

3.6. Shape interpolation

In this step, performed at every single function call within the optimization subroutine, we recreate the structural shape for an arbitrary design point (\bar{t}).

3.6.1 PCA reconstruction

In the first step, the α coefficients are obtained from the values of \bar{t} (location on the α -manifold), and thus the pixel maps:

$$\tilde{S}(\bar{t}) = \bar{S} + \sum_{i=1}^m \alpha_i(\bar{t}) \bar{\phi}_i \quad (9)$$

While we expect the interpolated map to contain values of 0 and 1 based on the previous section which appears to guarantee an admissible solution map as long as we stay on the α -manifold, it is still possible for the processes of averaging snapshots, singular value decomposition and truncation to deliver intermediate values (grayscale) around the boundaries as intermediate values during the optimization. The next step deals with this problem in case it surfaces.

3.6.2 Density filtering

Due to the truncation of the basis vectors, it is possible that in the course of the optimization we may pass through certain points slightly outside the feasible surfaces. A typical example is presented in Figure 7 showing the conversion from pixel map to boundary pixels. In this simple example, we use a worst-case pixel map with values of 0, 0.5 and 1. Here, there is one single boundary/edge to be located, but in complex shapes the density filter needs to allow the user to capture every possible edge. For the purposes of shape optimization, the authors have found Canny's algorithm [6] to be an appropriate density filter, with a pretreatment as presented below.

Any density gradient-based filter can easily be thrown off by certain situations, like the one shown in figure 7 where the filter throws up two boundaries, since both edges (the real one as well as the false one) represent a strong gradient, no matter the filter threshold. One would expect that a sufficiently diverse family of snapshots, a grid of sufficient resolution and a good optimization algorithm would limit the occurrence of such situations, but we might need to deal with situations of this sort during intermediate stages in the optimization.

In order to be able to distinguish between two gradients of the same strength based on the actual density values, we recommend pretreating each element of the snapshot S_{old} according to (10) in order to attenuate the stronger gradients further away from values of 1 (since the true boundary should be close to the front corresponding to a value of 1 for the approach presented in this paper).

$$S_i^{new} = S_i^{old} + h(1 - S_i^{old})^b \quad (10)$$

where h, b are constants that can be adjusted according to the type of problem being reduced. This is a simple but effective pretreatment somewhat inspired by grayscale suppression methods in topology optimization [24,19].

3.6.3 Pixel map boundary and Moving least squares smoothing

The next step is to locate the co-ordinates of the corner points/vertices of the boundary pixels using one of various possible methods [3], as seen in Figure 8 for a part of the pixel map.

Next, a local moving least square approximation using radial basis weighting functions [20,4] is performed to construct the boundaries/edges from the vertices of the (reconstructed) pixel map.

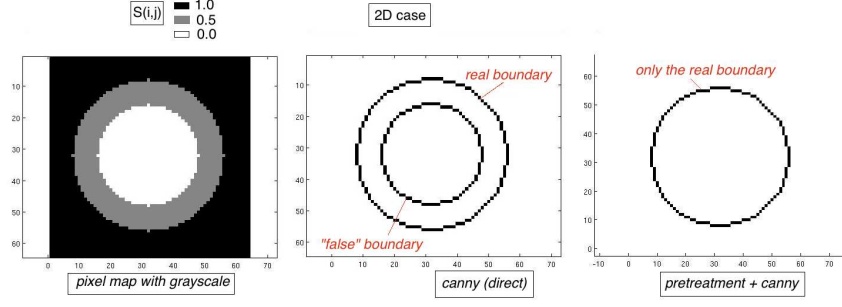


Figure 7. Canny filter for 2D case with and without proposed pretreatment

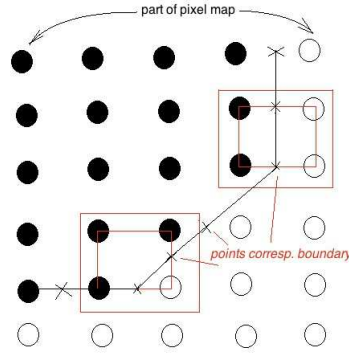


Figure 8. Finding the pixel map boundary

The boundary curve of the shape may be represented parametrically by $x(l), y(l)$

$$x(l) \approx x_{app}(l) = b^T(l)a_x(l) \text{ and } y(l) \approx y_{app}(l) = b^T(l)a_y(l) \quad (11)$$

where l = parameter representing the curve.

$$b(l) = [1 \quad l \quad l^2 \quad l^3 \quad \dots]^T \quad (12)$$

The coefficients $a_x(l)$ and $a_y(l)$ are not constant over the domain but depend on the values of the design variables, and are chosen to minimize the functionals $J_x(a)$ and $J_y(a)$ defined by:

$$\begin{aligned} J_x(a) &= \frac{1}{2} \sum_{i=1}^{N_n} w_i(l_i, l) (b^T(l_i) a_x - x(l_i))^2 \\ J_y(a) &= \frac{1}{2} \sum_{i=1}^{N_n} w_i(l_i, l) (b^T(l_i) a_y - y(l_i))^2 \end{aligned} \quad (13)$$

where N_n = number of vertices in the local neighborhood of each evaluation point used to describe the reconstructed pixel map, located by the marching cubes approach [3].

A typical radial basis form for the weighting function is $w_i(l_i, l) = \exp(-(l_i - l)^2)$, but one could also choose an interpolating polynomial form to better capture a somewhat irregular shape due to local effects.

3.7. Objective function evaluation

The reconstructed shape is meshed and the numerical analysis is performed using a method chosen based on the disciplines involved in the analysis i.e. CFD/Navier-Stokes for incompressible flows [17,16], FEA for structural analysis [22], etc. The only difference is that instead of obtaining X_i^{opt} we attempt to find the final governing parameters \bar{t}^{opt} and thus the coefficients $\bar{\alpha}(\bar{t}^{opt})$ that optimize the performance objective. An important phase here is remeshing the surfaces obtained in section 2.4. Chappuis et al [9] developed an approach of calculating principal curvatures from an existing mesh or shape using a secondary local model with Diffuse Interpolation, and then using these curvatures to identify shape primitives such as cylinders, torus, etc for the purpose of meshing. In this paper, we have calculated the curvature energy for the reconstructed edges and used this to position the nodes/blocks for meshing the newer shape.

3.8. Algorithm for overall procedure

The complete algorithm is shown in figure 9.

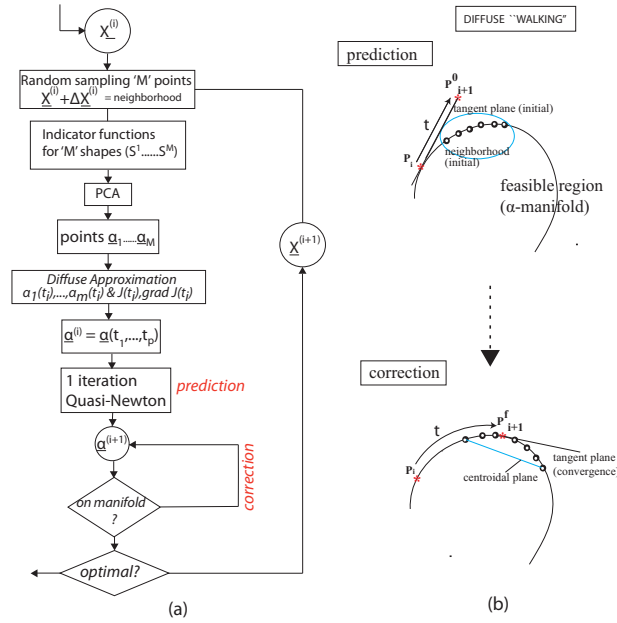


Figure 9. Schematic diagram for the overall algorithm

4. OPTIMIZATION TEST-CASE: AIR-CONDITIONING DUCT

4.1. Duct geometry

The inlet and outlet portions of the air-conditioning duct have fixed geometries, while the middle portion allows for modification of the shape and thus performance of the duct. The duct geometry as shown in Figure 10 is completely described by the relative positions of points P1 to P11. P1 to P4 and P9 to P11 are assumed fixed and the positions of P5, P6,

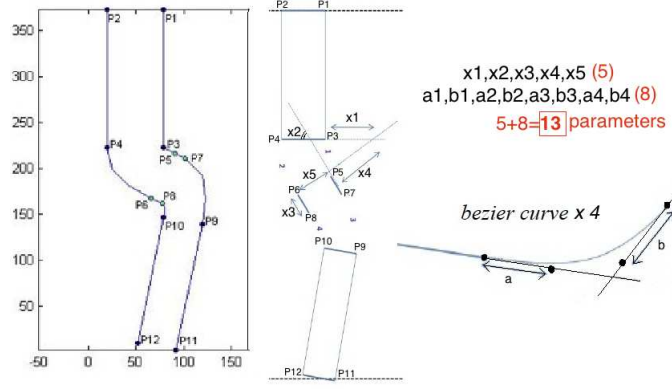


Figure 10. Duct geometry showing four different regions

P7 and P8 are required to determine the geometry of the portion of the duct critical to performance. The parameters that determine the locations of P5 to P8 are obtained by the geometric constructions shown. Parameters X1 to X5 allow us to locate P5 to P8, while parameters a1 to a4 and b1 to b4 allow us to draw Bezier curves passing through these points tracing out the whole geometry of the curved portion of the duct. The geometry of the curved portion of the duct in 2D (the only part of the duct that is variable) may thus be characterized by 13 parameters in all: x1 to x5, a1 to a4 and b1 to b4, and the design and thus performance can be changed by altering these parameters. In order to retain a laminar flow and for other design considerations, there are upper and lower bounds on these 13 parameters and thus on the possible designs for the duct. The design variables are thus: X_1, X_2, X_3, X_4, X_5 (for $P_1 \dots P_8$) and $X_6 = a_1, X_7 = b_1, X_8 = a_2, X_9 = b_2, X_{10} = a_3, X_{11} = b_3, X_{12} = a_4, X_{13} = b_4$.

4.2. CFD model and mesh

Since the Reynolds number for the situation is typically low, the air flow is modeled using OpenFoam CFD for incompressible 2D laminar flow. For every possible design resulting from particular choices of the 13 parameters described earlier in section 4.1, we set up a CFD grid with 39000 grid points and 17250 hexahedral cells. The physical domain is split into 23 different blocks for the purpose of meshing. Boundary conditions enforced for the CFD set the pressure at the duct outlet = 0 (atmospheric pressure) and flow speed along the walls (straight as well as curved portions) = 0. The CFD analysis is run for 500 iterations to ensure convergence, and the converged pressure and velocity fields in the duct are obtained for each hexahedral cell, and are assumed to be evaluated at the midpoint of each cell for post-processing and surrogate function calls. The performance-related objective function (permeability) may be directly evaluated from the pressure/velocity fields (both CFD as well as surrogate) for each design.

5. Optimization problem

The optimization problem in the geometric space may be written as:

$$\text{Find } \bar{X}^{opt} = \text{Argmax } P_{flow}(X_1, \dots, X_{13}) \quad \text{s.t. } \bar{L}_B \leq \bar{X} \leq \bar{U}_B \quad (14)$$

where P_{flow} = flow permeability = $1/(\text{pressure loss from inlet to outlet}) = 1/(P_{inlet} - P_{outlet})$, and \bar{U}_B and \bar{L}_B are the upper and lower bounds on the 13 design variables. Once we switch over to the reduced space, we can express the objective function as a function of the PCA coefficients (α_i) and hence the new design variables \bar{t} , thus the optimization problem may be written as:

$$\begin{aligned} \text{Find } \bar{t}^{opt} = & \text{Argmax } P_{flow}(\bar{\alpha}(t_1 \dots t_p)) \\ \text{s.t } g_{min} \leq & g_i(\bar{\alpha}(\bar{t})) \leq g_{max} \text{ and } h(\bar{\alpha}(\bar{t})) = 0 \end{aligned} \quad (15)$$

So the optimization problem is now one of finding the pixelized shape $S(\bar{\alpha}(\bar{t}^{opt}))$. The constraints $g_i, i \in [1, N]$ are obtained by transferring the bounds \bar{U}_B and \bar{L}_B on the α -space, while h represents the feasible region (set of manifolds in $\bar{\alpha}$ -space). Both g_i and h are taken into account implicitly with the Diffuse Approximation-based approach outlined earlier allowing the α 's to be expressed locally as functions of the final parameters t_1, \dots, t_p .

6. Results and discussion

Since the mapping to the new space is highly non-linear, we need to ensure we stay in the feasible region during the course of the optimization. Clearly there will be a loss of accuracy due to the pixelization and PCA phases [10] that gave the intermediate design variables ($\alpha_1 \dots \alpha_m$) and the reconstruction that allows us to capture the duct geometry from the α 's, but the author's contention is that the error will be consistent throughout and not seriously affect the results of the optimization, and this has been observed. Of course the effect of the error can be estimated very easily using a reconstruction of the original snapshot geometries after pixelization and decomposition, using the truncated basis. In the problem solved, we have used only $M = 102$ snapshots as mentioned previously. If M was chosen to be higher, i.e. 1000 or more snapshots using a Latin Hypercube Sampling of $M = 1000$ designs between the bounds, the quality of the truncation with say $m = 5$ modes would improve. In this paper, we have focused more on the approach of two-level model reduction using the basis truncation and the α -space Diffuse Approximation.

6.1. Dimensionality and model reduction

Figure 11 shows the dimensionality deduction approach first presented in section 3. After analyzing the individual snapshots in α -space, it is clear from the set of 2D surfaces obtained that the behavior of the various α 's is governed by just TWO parameters (say t_1 and t_2 that can be easily found by a Diffuse Approximation [20,4] over the retained α 's. The feasible regions are represented by the α -manifolds, and as explained in section 3, staying on the manifold ensures an admissible solution, even though we may need to invoke the density filter from time to time during the optimization. This also means that $\bar{\alpha} = [\alpha_1(t_1, t_2), \alpha_2(t_1, t_2) \dots \alpha_m(t_1, t_2)]$ if using a truncated basis of size m . So $\bar{\alpha}^{opt} = \bar{\alpha}(t_1^{opt}, t_2^{opt})$ transferring the problem into the \bar{t} -space where \bar{t} are the final parameters that control the overall design domain.

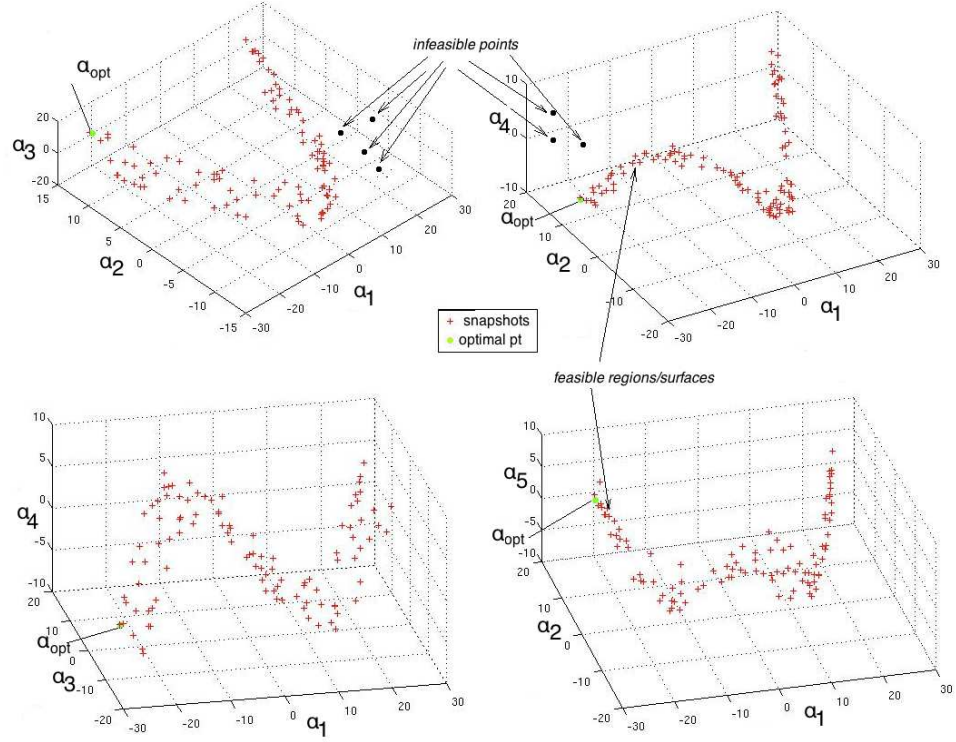


Figure 11. Dimensionality (= 2) and model reduction for A/C duct

6.2. Interpolation of duct shape

This is an important step consisting of first using the α coefficients to determine the pixel map \tilde{S} representing the shape of the duct, followed by passing the reconstructed pixel map \tilde{S} (if grayscale is present for whatever reason) through a Canny density filter as explained previously to get the boundary/edge pixels, followed by a "marching cubes" to extract the vertices of the boundary/edge pixels, and finally obtaining the actual smooth geometric shape (reconstructed) of the duct using the moving least squares approximation described in section 3. The weighting function needs to be carefully chosen while using a global (or local) Moving Least Squares (Diffuse Approximation) since a delicate balance between smoothness and precision is required and the duct geometry is composed of different types of sections. Since the boundary curves will next be used to create a CFD mesh the authors feel it is better to err somewhat in favor of smoothness and try to control precision by increasing resolution. Figure 12 shows an enlarged view of the MLS smoothing in the two curved portions of the duct. Figure 13 shows the quality of reconstruction with increased number of modes retained after truncation. As expected, the accuracy increases with the additional modes retained, but the increase in precision drops off quickly with added modes as explained in section 3. Finally, even the small loss of accuracy will be consistent for all the designs so by building a response surface between X_i and α_i or simply by inspection of the optimal shape obtained, one could easily extract the optimal design variables in the geometric design space (\bar{X}_{opt}) from the optimal variables $\bar{\alpha}_{opt}$. Of course, regardless of the number m of modes retained, these are all expressed as functions of the true design variables, i.e. $\alpha_1(t_1, t_2) \dots \alpha_m(t_1, t_2)$ with the diffuse

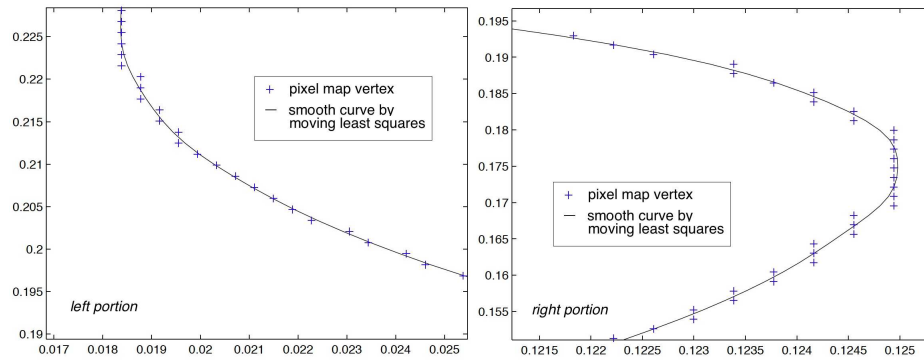


Figure 12. vertices and moving least square curve (enlarged)

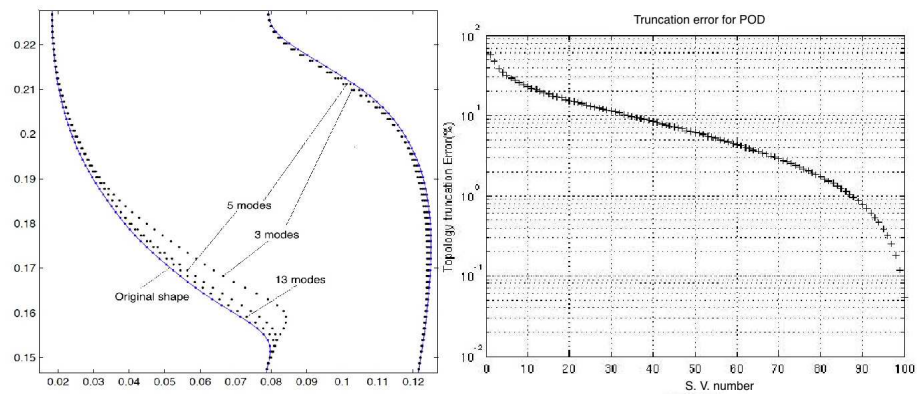


Figure 13. Reconstruction precision with increasing modes and truncation error

approximation.

6.3. Optimization in the reduced-space

The goal is to first perform the optimization in the reduced-space getting \bar{t}_{opt} , and then calculating $\bar{\alpha}(\bar{t}_{opt})$, and next to estimate \bar{X}_{opt} (original geometric parameters) from the values of $\bar{\alpha}(\bar{t}_{opt})$ either by inspection of the optimized shape $S(\bar{\alpha}(\bar{t}_{opt}))$ or using an RSM between the \bar{X}_i and α_i . The permeability P_{flow} for every possible design was calculated using the inverse of the total pressure drop across the duct length (inlet to outlet). The next step was to obtain the optimal shape using 5-8 modes, followed by identification by response surface methodology over the values of the original 13 geometric design parameters for each $\bar{\alpha}_{opt}$, i.e. getting $\bar{X}(\bar{\alpha}_{opt})$. The optimal solution obtained has been added to figure 10 and as expected, it lies on the edge of the constraint/feasible region. This is followed by "reverse look-up" (Table 1) by projecting the pixel array obtained by shape generation, meshing and then pixelization, on to the truncated basis of m modes.

$$\bar{\alpha}_{rev}(S(\bar{\alpha}^{opt})) = (\bar{S} + \sum_1^m \alpha_i^{opt} \phi_i^S - \bar{S})\Phi^S = (\sum_1^m \alpha_i^{opt} \phi_i^S)\Phi^S \quad (16)$$

where $\bar{\alpha}^{opt}$ = optimal coefficients using m modes and $\bar{\alpha}_{rev}$ = coefficients obtained by reverse identification from $\bar{X}(\bar{\alpha}^{opt})$.

This reverse look-up of α coefficients from the identified geometric parameters $\bar{X}(\bar{\alpha}_{opt})$ is needed to account for the error introduced by the moving least squares approximation needed to map the $\bar{\alpha}_{opt}$ (m coefficients) to the $\bar{X}(\bar{\alpha}_{opt})$.

The velocity fields for the optimal shapes obtained using 5-8 modes are shown in figure 14 and the values of $\bar{\alpha}^{opt}$, $\bar{X}(\bar{\alpha}^{opt})$ and $\bar{\alpha}_{rev}$ are shown in Table 2, where the pressure drop needs to be minimized for optimal performance. The velocity field gets increasingly more regular with additional modes retained. We also note that the $\bar{\alpha}_{rev}(S(\bar{\alpha}^{opt}))$ and $\bar{\alpha}^{opt}$ are fairly close to each other, the slight discrepancy being due to the error introduced by the moving least squares for the response surface between the α 's and the geometric parameters. This error can be completely or mostly avoided by one of two approaches:

1. Using direct identification to extract the geometric design parameters directly from the structural shape/mesh created using the $\bar{\alpha}_{opt}$ with m modes.
2. The $\bar{\alpha}_{opt}$ can be used to directly plot the structural shape and this can be directly used for design purposes instead of trying to first extract the geometric design parameters. This can and should be the method of choice, but the RSM between the two sets of design variables is directly programmable in most cases and always an option for the design engineer.

Conclusions

In this paper, the authors have introduced an "a posteriori" scheme with a two-level model reduction to replace the geometry-based variables with a more compact and normalized set of variables and replace the higher-dimensional design space with a newer design space of lower-dimension.

The overall interpolation technique is nonlinear, and is constrained to produce only shapes

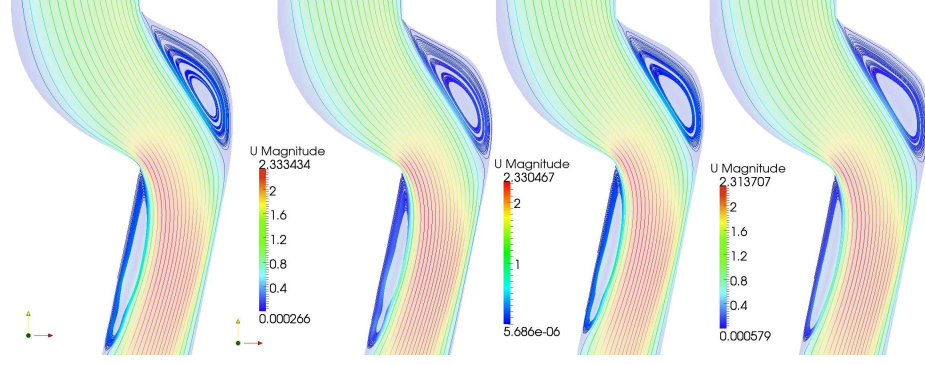


Figure 14. Velocity field ($m=5$, $m=6$, $m=7$ and $m=8$)

modes (m)	$\bar{\alpha}^{opt}$	$\bar{\alpha}_{rev}(S(\bar{\alpha}^{opt}))$
5	-22.6226,13.4301,5.8149,- 9.4755,3.1751	-21.8192,13.7669,6.6731,- 8.8569,3.2854
6	-22.4873,14.2647,6.2187,- 9.3431,3.0856,3.7891	-22.0924,13.274,5.8194,- 9.2671,3.0162,4.2320
7	-22.7028,14.4708,5.6493,- 9.5091,3.1863,3.9603,5.2279	-22.1531,13.8087,5.2442,- 9.1170,2.8065,4.2241,4.6575
8	-22.3745,14.7165,5.1094,- 9.7710,3.2653,3.9784,5.2527,- 4.2108	-22.0974,13.8794,5.6017,- 9.4491,2.4682,5.2825,4.7886,- 2.3602

Table 1. Reverse look-up: comparison between α^{opt} and α_{rev} for different m

from an abstract manifold in shape space induced by learning. The non-varying zones used for boundary conditions are naturally preserved and additional constraints may be imposed using constrained versions of Proper Orthogonal Decomposition. Since the approach is "a posteriori", it is clearly dependent on the information contained in the snapshot database, and thus the initial sampling used to create the database. While it is clear that the analysis of the α manifolds using a diffuse approximation is the main reduction phase, the truncation to m modes is also important to reduce computational effort. The results showed how the geometry could be very closely described even with 5 modes (ultimately depending on 2 parameters) even with a modest snapshot database with a sampling size of 102, and it is obvious that the accuracy of the truncation can be directly influenced by increasing the sampling size. One could also combine this approach with using a surrogate model for the numerical analysis to greatly increase the performance as well as reduce overall computation time.

The presented methodology has a few possible areas of improvement. The first is in resolving the difficulty in setting upper and lower bounds on the α -based design variables. The second area is in the treatment of possible degenerate cases for the structural shape. The third area is studying the efficacy of the Canny and pretreatment filter in "bounding" 3D pixel maps.

modes (m)	5	6	7	8	\bar{L}_B	\bar{U}_B
$X1(\alpha^{opt})$	7.8528	8.2695	8.4067	8.0782	6.1096	9.1644
$X2(\alpha^{opt})$	25.7555	28.9527	30.1149	28.7854	21.3080	31.962
$X3(\alpha^{opt})$	13.3979	14.1076	13.8989	13.8898	10.1464	15.2196
$X4(\alpha^{opt})$	7.8255	8.3505	8.4015	8.6217	6.0832	9.1248
$X5(\alpha^{opt})$	64.2028	65.4544	65.4339	64.5703	43.6	65.4
$X6(\alpha^{opt})$	1.9279	2.2016	2.2544	2.3511	1.6	2.4
$X7(\alpha^{opt})$	2.0009	2.0609	2.0628	1.9503	1.6	2.4
$X8(\alpha^{opt})$	35.6519	29.3352	30.7665	33.2524	24	36
$X9(\alpha^{opt})$	35.9963	33.4178	34.0429	35.0498	24	36
$X10(\alpha^{opt})$	37.1714	36.7973	37.8723	36.76	24	36
$X11(\alpha^{opt})$	26.4866	30.8812	30.5599	33.0657	24	36
$X12(\alpha^{opt})$	4.9232	5.5084	5.4175	5.7500	4	6
$X13(\alpha^{opt})$	5.4837	5.0370	5.3058	5.4318	4	6
$J1(\alpha^{opt})$	1.9684	1.9754	1.9626	1.9580		
$J2(X(\alpha^{opt}))$	2.1344	2.1335	2.1389	2.1312		
Shape error (%)	0.847	0.33	0.248	0.163		

Table 2. optimization in α -space and identification of geometric parameters \bar{X}

Acknowledgements

This work has been supported by the French National Research Agency (ANR), through the COSINUS program (project OMD2 no. ANR-08-COSI-007). The authors acknowledge the Projet Pluri-Formations PILCAM2 at the Universite de Technologie de Compiègne (URL: <http://pilcam2.wikispaces.com>) for providing HPC resources as well as Maryan Sidorkiewicz, Direction de la Recherche, Renault, France and Mr. V. Picheny, Ecole des Mines, France for contributing the CFD model.

7. REFERENCES

- [1] Berkooz, G., Holmes, P., Lumley, J.L., "The proper orthogonal decomposition in the analysis of turbulent flows". *Annu Rev Fluid Mech* 25(1), 539-575, 1993.
- [2] Bregler, C., Omohundro, S.M., "Nonlinear image interpolation using manifold learning" *Neural Information Processing Systems* 973-980, 1995.
- [3] Breitskopf, P., "An algorithm for construction of iso-valued surfaces for finite elements." *Engineering with Computers* 14(2), 146-149, 1998.
- [4] Breitskopf, P., Naceur, H., Rassineux, A., Villon, P., "Moving least squares response surface approximation: Formulation and metal forming applications" *Computers and Structures* 83(17-18), 1411-1428, 2005.
- [5] Bui-Thanh, T., Willcox, K., Ghattas, O., van Bloemen Waanders, B., "Goal-oriented, model-constrained optimization for reduction of large-scale systems" *Journal of Computational Physics* 224(2), 880 -896, 2007.
- [6] Canny, J., "A computational approach to edge detection" *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6), 679-698, 1986.
- [7] Carlberg, K., Farhat, C., "A compact proper orthogonal decomposition basis for optimization-oriented reduced-order models" *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* Victoria, Canada, 2008.
- [8] Carlberg, K., Farhat, C., "A low-cost, goal-oriented compact proper orthogonal decomposition basis for model reduction of static systems" *International Journal for Numerical Methods in Engineering* 86(3), 381-402, 2010.
- [9] Chappuis, C., Rassineux, A., Breitskopf, P., Villon, P., "Improving surface remeshing by feature recognition" *Engineering with Computers* 20(3), 202209, 2004.
- [10] Chatterjee, A., "An introduction to the proper orthogonal decomposition" *Current Science, Special Section: Computational Science* 78(7), 808-817, 2005.
- [11] Coelho, R.F., Breitskopf, P., Knopf-Lenoir, C., "Bi-level model reduction for coupled problems" *Int J Struc Multidisc Optim* 39(4), 401-418, 2009.
- [12] Cordier, L., El Majd, B.A., Favier, J., "Calibration of pod reduced order models using tikhonov regularization" *International Journal for Numerical Methods in Fluids* 63(2), 269-296, 2010.
- [13] Couplet, M., Basdevant, C., Sagaut, P., "Calibrated reduced-order pod-galerkin system for fluid flow modeling" *Journal of Computational Physics* 207(1), 192-220, 2005.
- [14] Dulong, J.L., Druesne, F., Villon, P., "A model reduction approach for real-time part deformation with nonlinear mechanical behavior" *International Journal on Interactive Design and Manufacturing* 1(4), 229-238, 2007.
- [15] Kaufman, A., Cohen, D., Yagel, R., "Volume graphics" *IEEE Computer* 26(7), 51-64, 1993.
- [16] Larsson, F., Diez, P., Huerta, A., "A flux-free a posteriori error estimator for the incompressible stokes problem using a mixed fe formulation" *AIAA Journal* 199(37-40), 2383-2402, 2010.
- [17] Launder, B.E., Spalding, D.B., "The numerical computation of turbulent flows" *Computer Methods in Applied Mechanics and Engineering* 3(2), 269-289, 1974.

- [18] LeGresley, P., Alonso, J., "Airfoil design optimization using reduced order models based on proper orthogonal decomposition" *Proceedings of the Fluids 2000 Conference and Exhibit* Denver, CO, 2000.
- [19] Mahdavi, A., Balaji, R., Frecker, M., Mockensturm, E., "Topology optimization of 2d continua for minimum compliance using parallel computing" *Int J Struc Multidisc Optim* 32(2), 121-132, 2005.
- [20] Nayroles, B., Touzot, G., Villon, P., "Generalizing the finite element method: diffuse approximation and diffuse elements" *Computational Mechanics* 10(5), 307-318, 1992.
- [21] Raghavan, B., Breitzkopf, P., "Asynchronous evolutionary shape optimization using high-quality surrogates: application to an air-conditioning duct." *Engineering with Computers* doi: 10.1007/s00366-012-0263-0
- [22] Rodriguez, H.C., "Shape optimal design of elastic bodies using a mixed variational formulation" *Computer Methods in Applied Mechanics and Engineering* 69(1), 29-44, 1988.
- [23] Sahan, R.A., Gunes, H., Liakopoulos, A., "A modeling approach to transitional channel flow" *Computers and Fluids* 27(1), 121-136, 1998.
- [24] Sigmund, O., "A 99 line topology optimization code in MATLAB" *Int J Struc Multidisc Optim* 21(2), 120-127, 2001.
- [25] Sofia, A.Y.N., Meguid, S.A., Tan, K.T., "Shape morphing of aircraft wing: Status and challenges" *Materials & Design* 31(3), 1284-1292, 2010.
- [26] Willcox, K., Peraire, J., "Balanced model reduction via the proper orthogonal decomposition" *AIAA Journal* 40(11), 2323-2330, 2002.
- [27] Xiao, M., Breitzkopf, P., Coelho, R.F., Knopf-Lenoir, C., Villon, P., "Enhanced POD projection basis with application to shape optimization of car engine intake port" *Int J Struc Multidisc Optim* doi: 10.1007/s00158-011-0757-1