

CAD-based Aerodynamic Shape Optimization using Geometry Surrogate Model and Radial Basis Functions Mesh Deformation coupled with Adjoint Method

Kamil Bobrowski*, Esteban Ferrer† and Eusebio Valero‡

ETSIAE (School of Aeronautics), Universidad Politécnica de Madrid, Madrid, Spain

Holger Barnewitz§

AIRBUS Operations GmbH, Bremen, Germany

Nowadays classical configurations of civil transportation aircraft are well optimized, with further improvements expected to be of the order of single drag counts. To achieve it, the whole aircraft configuration has to be taken into account during aerodynamic shape optimization, which require handling complex geometries with Computer Aided Design (CAD) software. So far, integration of CAD engine and aerodynamic shape optimization has been difficult, due to framework complexity and problematic gradient evaluation. In this paper we propose to substitute CAD engine with geometry surrogate model of the CAD surface. The resulting framework does not require calls to the CAD engine during the optimization, with design variables defined in CAD during preparation step. Furthermore, geometry surrogate model has been analytically differentiated, allowing coupling with Radial Basis Functions mesh adjoint method of objective function gradient evaluation.

Nomenclature

v_i	i -th element of the vector \mathbf{v}
\mathbf{v}^i	i -th vector from the set of vectors $\{\mathbf{v}^1, \mathbf{v}^2, \dots\}$
I	Objective function
\mathbf{D}	Design variables
ϕ	POD modes
$\mathbf{a}(\mathbf{D})$	POD coefficients (function of design variables)
$\tilde{\mathbf{a}}(\mathbf{D})$	Interpolated POD coefficients
φ	Radial Basis Function
β, γ	Coefficients of RBF interpolation (of POD coefficients)
η, ζ	Coefficients of RBF deformation (of volume mesh)
\mathbf{x}_i	Coordinates of i -th volume mesh node
$\Delta \mathbf{x}_i$	Deformation of i -th volume mesh node
$\mathbf{x}_{s,i}$	Coordinates of the i -th surface deformation base point
$\Delta \mathbf{x}_{s,i}$	Deformation in the i -th surface deformation base point
\mathbf{b}	weighting factors for RBF deformation
d	wall distances to the deformation zone
R_Z	Radius Zero Weight
R_F	Radius Full Weight
n_s	Number of sample points in the design space
n_m	Number of POD modes

*Ph.D. Student, Applied Mathematics Department

†Lecturer, Applied Mathematics Department

‡Professor, Applied Mathematics Department

§Optimization Expert

n_p	Number of surface deformation field points
n_d	Number of dimensions of the design space
N	Number of volume nodes
\mathbf{Q}	Vector of flow variables
\mathbf{R}	Flow residual vector
\mathbf{X}	Volume mesh deformation
$\mathbf{X}_s(\mathbf{D})$	Deformation field on the surface (function of design variables)
$\widetilde{\mathbf{X}}_s(\mathbf{D})$	Prediction of the deformation field from GEO-POD model
$\mathbf{\Lambda}$	Flow adjoint variables
$\mathbf{\Lambda}_g$	RBF mesh adjoint variables
C_L	Lift coefficient
C_D	Drag coefficient
C_M	Pitching moment coefficient
C_p	Pressure coefficient
M	Mach number
α	Angle of attack

Abbreviations

POD	Proper Orthogonal Decomposition
GEO-POD	Geometry surrogate model based on POD
RBF	Radial Basis Function
HTP	Horizontal Tailplane

I. Introduction

I.A. Modern aircraft design

Classical aircraft design methods are based on Cayley's design paradigm formulated in the nineteenth century at the very beginning of aviation development, which assumes that each of the aircraft components has a specific role (wing to provide lift, tail to provide control, stability and trimming etc.) and are only weakly coupled aerodynamically. Based on this principle, each of the aircraft component used to be optimized separately, with only some minor modifications performed after assembling and analysing the whole aircraft.

There are number of aircraft for which Cayley's design paradigm could not be fully applied, including flying wings or high-maneuverability fighter aircraft with strong non-linear aerodynamic coupling between different components, but the design of classical configurations followed Cayley's paradigm throughout the twentieth century. Following the shift from "higher, faster, further" to "better, faster, cheaper" paradigm which occurred in the 1990s, industry started to push design engineers to reduce design and manufacturing costs, while improving the aircraft performance in familiar flight conditions. Risk minimization became a requirement as well, with a goal to generate the design without extensive trial and error process and reduce or eliminate necessity for prototype models. This has led to a more integrated design approach, taking into account requirements and constraints of the whole design process from the beginning. To achieve that, high-fidelity model of the aircraft physical properties, also referred as Virtual Product¹ has to be available already at the early stage of the design process.

With the developments of computational clusters and advances in flow simulation methods it has become possible to bring high-fidelity, virtual representation of the aircraft physics earlier to the design process. Cayley's design paradigm has become less relevant, as complex interference effects induced by installation of the engine, flap-track fairings etc. can be taken into consideration.²

Bringing high-fidelity earlier to the design process in the civil transportation industry is facilitated by a step-by-step development approach, undertaken by both Airbus and Boeing. New aircraft are based to a large extent on the previous models, and many of the developments are smaller, but more frequent (derivative aircraft). Examples of this approach include Airbus A320 family development (A320 - A320 Enhanced - A320neo) as well as Boeing 737 family development (737 - 737 Classic - 737 Next Generation - 737 MAX).

I.B. Usage of CAD system in the aerodynamic optimization

As a response to the shift in design paradigms described in the previous section, advanced Computer Aided Design, Manufacturing and Engineering (CAD/CAM/CAE) systems have been developed, to enable more integrated design process. One of the implications for high-fidelity aerodynamic shape optimization was that both input and output of the optimization cycle should be represented by a CAD model.

Optimization based on CAD poses several challenges. Nowadays CAD systems are usually used on the designer's personal machine, while flow computations are performed on the remote cluster. Optimization framework which would directly access CAD engine running on the designer's computer would have to include interface between local and remote machines and most likely between different operating systems.³ Next challenge is the robustness of the CAD model - for complex models and specific design variables there might be no solution satisfying internal CAD constraints, which might result in a failure of the optimization algorithm. Finally, implementing analytical geometrical sensitivities (for the gradient-based optimization) in the CAD engine is difficult due to the number and complexity of possible operations, and has not yet been included in leading CAD software.

A popular approach creates a model of the CAD surface and uses this model in the optimization loop, without calls to the CAD engine. Examples of existing approaches include extraction of Non-Uniform Rational B-Splines (NURBS) from CAD surface and optimization of NURBS control points⁴ or optimization of every point of the surface mesh and fitting CAD surface to obtained optimal shape.⁵ Main advantages of these methods are simple framework (CAD is used outside the optimization loop) and possible calculation of surface gradients analytically. Design variables used in these methods allow for wide spectrum of shapes, although it might be a disadvantage if precise control over the parameterization (using parameters defined in CAD) is of interest. Furthermore, the CAD engine cannot be used to efficiently handle complex geometrical constraints or intersections.

The other notable approach is based on using the CAD engine in the optimization loop through a CAPRI interface.⁶ The latter has the advantage of using actual CAD parameterization as design variables and can handle complex geometries, although implementation is complicated, especially in industrial environments due to the frequent calls to the CAD engine. The latter approach has been used in gradient-based optimization with surface deformation gradient computed with finite-difference method.⁷ It is possible to obtain fully analytical geometry sensitivities,⁸ although it requires significant implementation effort (each of the operation that can be performed by CAD engine has to be analytically differentiated).

I.C. Surrogate model of CAD surface

The core idea of the approach presented in this work is to build an accurate, cheap to evaluate and robust surrogate model of the CAD surface, and to use this model instead of running CAD engine in the optimization loop. Geometry surrogate model based on Proper Orthogonal Decomposition (GEO-POD) is used for that task, which for given CAD design variables predicts discrete deformation field on the CAD surface. This kind of surrogate models has proven to be efficient for approximating aerodynamic data.^{9,10} Results show that for reconstructing geometrical data these models are even more accurate.

A set of consistent surface deformation fields, corresponding to various CAD design variables settings, is used to train the GEO-POD model. These deformation fields are obtained from CAD surface by discretizing the curves created by the designer in the regions subject to the optimization. Surface deformation fields can be created for any arbitrary set of design variables, although it is beneficial to use Design of Experiments (DoE) techniques¹¹ to automatically sample the design space. More details on GEO-POD model creation are given in chapter II.

Our method combines the advantages of previously described approaches - CAD engine is used only in the preparation step and not during the optimization, which eliminates the need for interface between the computational cluster and the machine operating CAD software. CAD model parameters serve as design variables, which gives the designer full control over the parameterization, and significantly reduces the effort manufacturing final designs. Complex geometries and intersections are handled accurately, and internal geometrical constraints defined in CAD are satisfied. Moreover, due to mathematical simplicity of that model, it can be easily analytically differentiated, providing input for adjoint method (see details in the next section).

Contrary to other approaches, the approximation of the CAD surface is optimized instead of the accurate machine-precision CAD representation. Accuracy of the GEO-POD model has to be ensured by using

sufficient number of points to sample the deformation field.

I.D. The adjoint method in optimization

Gradient-free optimization methods are commonly used in aerodynamic shape optimization. Only the objective function and optimization constraints have to be evaluated, without the need to calculate gradients. A popular gradient-free method is the Nelder-Mead Simplex,¹² which has proven to be robust for moderate number of design variables.³

Gradient-based optimizers converge significantly faster provided that the gradients are calculated efficiently. Widely used gradient-based optimizers which converge to a local minimum include steepest descent and its derivatives (e.g. conjugate gradients), as well as Newton's method based algorithms for which objective function is approximated by a quadratic form.¹³ Usually, the approximate Hessian matrix is constructed (e.g. Broyden-Fletcher-Goldfarb-Shanno¹⁴). The trust-region approach¹⁵ is the extension of these methods, where the step size is limited in the region where the function approximation matches the objective function with sufficient accuracy.

There are numerous methods to calculate gradients (classical finite difference, complex step¹⁶), among which the adjoint method has proven to be one of the most efficient for aerodynamic optimization, especially when the number of design variables significantly exceeds the number of objective functions. This method has been developed for both the Euler and viscous Navier-Stokes equations.^{17,18} A key property of the adjoint solution is that it provides gradient values with respect to the single objective function at a cost weakly dependent on the number of design variables.¹⁹ The adjoint solution can be calculated in discrete or continuous way. In the former the equations are initially discretized and then the adjoint equations are formulated, while in the latter the continuous equations are first formulated and then discretized. The continuous adjoint formulation is generally cheaper to derive,²⁰ although is not exempt of difficulties when considering viscous flows.²¹ On the other hand, the discrete adjoint approach requires more computer memory since the discretized adjoint problem needs to be kept in memory, thus approximations of this method have been developed.²² In this paper, a discrete adjoint implementation is used. This approach is implemented in the DLR TAU solver for both Euler and Reynold-Averaged Navier-Stokes equations.

I.E. Mesh deformation and mesh adjoint methods

A robust mesh deformation approach is crucial for aerodynamic optimizations, as recomputing 3D volume meshes at each iteration would significantly increase the optimization time and could incur in significant discretization errors that can in turn damage the optimization convergence. Mesh deformation using linear elasticity has proven to be robust even for large displacements,²³ although large sparse linear system need to be solved, which might increase the computational memory required. An alternative approach uses a Delaunay graph mapping²⁴ as a direct method to allow fast and memory efficient deformations.

In this work we apply a different direct method based on Radial Basis Functions (RBF).²⁵ The input for this method is a discrete deformation field on the surface, not necessarily matching the position of surface mesh nodes, which is well suited for coupling with the output of the GEO-POD model. The deformations of each of the volume nodes is a linear combination of Radial Basis Functions centered around a set of control points, multiplied by a scaling factor depending on a distance from the surface. Details of the implementation are given in chapter III.

To evaluate the gradients using the discrete adjoint approach, the volume mesh deformation sensitivities with respect to the design variables has to be taken into account. This can be done with finite difference method.²² Alternatively, mesh deformation sensitivities can be eliminated using the mesh adjoint, derived for elastic deformation by Nielsen.²⁶ The RBF deformation approach has been coupled with adjoint by Jakobsson,²⁷ providing objective function sensitivities in each of the control points. Following Nielsen's method, an alternative derivation is presented in chapter IV by introducing RBF adjoint variables. Furthermore, analytical derivative of the GEO-POD model enable objective function gradients to be calculated without finite differences in any place during the optimization process.

II. CAD geometry surrogate model based on POD

The first step to build the GEO-POD model is to discretize the CAD surface. We propose a simple, yet powerful method based on discretizing curves available in the model. Predefined number of points are

distributed equidistantly along specified curves. Change in the position of those points, due to CAD design variable modification, is a measure of the surface deformation. If the curves available in the model do not capture geometry change sufficiently well in the region of interest, additional curves can be easily created by cutting through available surfaces. In the second step, the CAD design variables space is sampled, generating a set of deformation fields. The goal is to build accurate surrogate model, which for any design variables which values lie in a sampled ranges predicts the deformation field.

II.A. Proper Orthogonal Decomposition

Each of the coordinates of the sampled deformation fields can be expressed as a linear combination of modes ϕ (1). These modes are obtained using Proper Orthogonal Decomposition (POD) method, which seeks a subspace approximating a given set of vectors, such that mean square error of a projection of the original vectors onto that subspace is minimized.^{10,28} Mathematical formulation of this optimization problem is given in Eq. (2), where Π_ϕ is an operator of an orthogonal projection onto the subspace spanned by modes ϕ .

$$\mathbf{X}_s(\mathbf{D}^j) = \sum_{i=1}^{n_m} a_i(\mathbf{D}^j) \phi^i \quad j = 1, \dots, n_s \quad (1)$$

$$\underset{\phi}{\text{minimize}} \quad I(\phi) = \sum_{j=1}^{n_s} \|\mathbf{X}_s(\mathbf{D}^j) - \Pi_\phi \mathbf{X}_s(\mathbf{D}^j)\| \quad (2)$$

Solution of this optimization problem is equivalent to solving an eigenvalue problem^{28,29} (Eq. (3)), where \mathbf{Y} is a snapshot matrix given in the Eq.(4).

$$\mathbf{Y}\mathbf{Y}^T \phi^j = \lambda_j \phi^j \in \mathbb{R}^{n_p} \quad j = 1, \dots, n_s \quad (3)$$

$$\mathbf{Y} = [\mathbf{X}_s(\mathbf{D}^1) \ \dots \ \mathbf{X}_s(\mathbf{D}^{n_s})] \in \mathbb{R}^{n_p \times n_s} \quad (4)$$

This $n_p \times n_p$ eigenvalue problem can be reduced to smaller $n_s \times n_s$ problem (usually $n_s \ll n_p$), by solving Eq. (5) and computing ϕ using Eq. (6).

$$\mathbf{Y}^T \mathbf{Y} \mathbf{V}^j = \lambda_j \mathbf{V}^j \in \mathbb{R}^{n_s} \quad j = 1, \dots, n_s \quad (5)$$

$$\phi^j = \frac{1}{\sqrt{\lambda_j}} \mathbf{Y} \mathbf{V}^j \in \mathbb{R}^{n_p} \quad j = 1, \dots, n_s \quad (6)$$

After the POD modes are calculated, POD coefficients $\mathbf{a}(\mathbf{D}^1), \dots, \mathbf{a}(\mathbf{D}^{n_s})$ are computed by projecting original snapshots onto the modes. For large amount of data it may be beneficial to reduce the number of modes ($n_m < n_s$) by keeping only the ones which correspond to the largest eigenvalues. This operation can significantly reduce the storage, but also may decrease the accuracy.

II.B. Multidimensional interpolation

The POD method described in previous section provides discrete function bounding design variables vectors $\mathbf{D}^1, \dots, \mathbf{D}^{n_s}$ with corresponding POD coefficients $\mathbf{a}(\mathbf{D}^1), \dots, \mathbf{a}(\mathbf{D}^{n_s})$. To obtain intermediate $\tilde{\mathbf{X}}_s(\mathbf{D})$ for design variables not present in the initial sampling, POD coefficients need to be interpolated (7).

$$\tilde{\mathbf{X}}_s(\mathbf{D}) = \sum_{i=1}^{n_m} \tilde{a}_i(\mathbf{D}) \phi^i \quad (7)$$

We use Radial Basis Function interpolation with Thin Plate Spline (TPS) kernel. The interpolated POD coefficient is a linear combination of radially symmetric basis function φ centered at the sampled points in the design variables space. It is usually beneficial to add first order polynomial to the expression (8). TPS function is given in Eq. (9) and GEO-POD prediction in Eq. (10).

$$\tilde{a}_i(\mathbf{D}) = \boldsymbol{\beta}^{iT} \begin{bmatrix} 1 \\ \mathbf{D} \end{bmatrix} + \sum_{k=1}^{n_s} \gamma_k^i \varphi(\|\mathbf{D} - \mathbf{D}^k\|) \quad i = 1, \dots, n_m \quad (8)$$

$$\varphi(r) = \begin{cases} r^2 \log(r), & \text{for } r \neq 0 \\ 0, & \text{for } r = 0 \end{cases} \quad (9)$$

$$\widetilde{\mathbf{X}}_s(\mathbf{D}) = \sum_{i=1}^{n_m} \left[\boldsymbol{\beta}^{iT} \begin{bmatrix} 1 \\ \mathbf{D} \end{bmatrix} + \sum_{k=1}^{n_s} \gamma_k^i \|\mathbf{D} - \mathbf{D}^k\|^2 \log(\|\mathbf{D} - \mathbf{D}^k\|) \right] \boldsymbol{\phi}^i \quad (10)$$

Constants $\boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^{n_m}$ and $\boldsymbol{\gamma}^1, \dots, \boldsymbol{\gamma}^{n_m}$ are chosen to satisfy condition that POD coefficients prediction at the sample points is exact (Eq. (11)). Additional condition given in Eq. (12) is added to allow unique interpolation.

$$a_i(\mathbf{D}^j) = \tilde{a}_i(\mathbf{D}^j) = \boldsymbol{\beta}^{iT} \begin{bmatrix} 1 \\ \mathbf{D}^j \end{bmatrix} + \sum_{k=1}^{n_s} \gamma_k^i \varphi(\|\mathbf{D}^j - \mathbf{D}^k\|) \quad (11)$$

$$\begin{aligned} 0 &= \sum_{k=1}^{n_s} \gamma_k^i \begin{bmatrix} 1 \\ \mathbf{D}^k \end{bmatrix} \\ i &= 1, \dots, n_m \\ j &= 1, \dots, n_s \end{aligned} \quad (12)$$

In matrix notation, $\boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^{n_m}$ and $\boldsymbol{\gamma}^1, \dots, \boldsymbol{\gamma}^{n_m}$ are calculated by introducing interpolation matrix H_D (Eq. (15)) and solving Eq. (18). Literature includes further explanation of POD surrogate modelling technique.^{9,10}

$$\mathbf{A}_D = \begin{bmatrix} \varphi(\|\mathbf{D}^1 - \mathbf{D}^1\|) & \dots & \varphi(\|\mathbf{D}^1 - \mathbf{D}^{n_s}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{D}^{n_s} - \mathbf{D}^1\|) & \dots & \varphi(\|\mathbf{D}^{n_s} - \mathbf{D}^{n_s}\|) \end{bmatrix} \in \mathbb{R}^{n_s \times n_s} \quad (13)$$

$$\mathbf{P}_D = \begin{bmatrix} 1 & \dots & 1 \\ \mathbf{D}^1 & \dots & \mathbf{D}^{n_s} \end{bmatrix} \in \mathbb{R}^{(n_d+1) \times n_s} \quad (14)$$

$$\mathbf{H}_D = \begin{bmatrix} \mathbf{0} & \mathbf{P}_D \\ \mathbf{P}_D^T & \mathbf{A}_D \end{bmatrix} \in \mathbb{R}^{(n_s+n_d+1) \times (n_s+n_d+1)} \quad (15)$$

$$\boldsymbol{\gamma}^i = \begin{bmatrix} \gamma_1^i \\ \vdots \\ \gamma_{n_s}^i \end{bmatrix} \in \mathbb{R}^{n_s} \quad (16)$$

$$\mathbf{a}^i = \begin{bmatrix} a_i(\mathbf{D}^1) \\ \vdots \\ a_i(\mathbf{D}^{n_s}) \end{bmatrix} \in \mathbb{R}^{n_s} \quad (17)$$

$$\mathbf{H}_D \begin{bmatrix} \boldsymbol{\beta}^i \\ \boldsymbol{\gamma}^i \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{a}^i \end{bmatrix} \quad i = 1, \dots, n_m \quad (18)$$

II.C. GEO-POD model gradient

For the purpose of gradient calculation with discrete adjoint method, GEO-POD prediction equation (Eq. (10)) has to be differentiated with respect to the design variables. Analytical differentiation results is given in Eq. (19), where vectors $\boldsymbol{\delta}^1, \dots, \boldsymbol{\delta}^{n_m}$ are defined in Eq. (20). The computational cost of predicting the

gradient is similar to the cost of deformation prediction itself and is weakly dependent on the number of design variables.

$$\begin{aligned}
\frac{\partial \widetilde{\mathbf{X}}_s}{\partial \mathbf{D}} &= \sum_{i=1}^{n_m} \frac{\partial \widetilde{a}_i(\mathbf{D})}{\partial \mathbf{D}} \phi^i \\
&= \sum_{i=1}^{n_m} \left[\boldsymbol{\delta}^i + \sum_{k=1}^{n_s} \gamma_k^i \frac{\partial \varphi(\|\mathbf{D} - \mathbf{D}^k\|)}{\partial \mathbf{D}} \right] \phi^i \\
&= \sum_{i=1}^{n_m} \left[\boldsymbol{\delta}^i + \sum_{k=1}^{n_s} \gamma_k^i (\mathbf{D} - \mathbf{D}^k) (2 \log(\|\mathbf{D} - \mathbf{D}^k\|) + 1) \right] \phi^i
\end{aligned} \tag{19}$$

$$\boldsymbol{\delta}^i = \begin{bmatrix} \beta_2^i \\ \vdots \\ \beta_{n_d+1}^i \end{bmatrix} \quad i = 1, \dots, n_m \tag{20}$$

III. Mesh deformation method using Radial Basis Functions

Mesh deformation using Radial Basis Functions extrapolates discrete deformation field given on the base points into the volume mesh. The resulting volume mesh deformation is a linear combination of radially symmetric basis function φ centered at the base points. For deformation which show a polynomial character, it is beneficial to add a polynomial to the expression. Furthermore, scaling factor dependent only on the distance d from the surface being deformed is applied, which linearly suppresses the deformation between R_F and R_Z . Expression for volume mesh deformation is given in Eq. (21) and Eq. (22). In this case we use linear radial basis function, given in Eq. (23).

$$\Delta \mathbf{x}_i = b_i \left(\boldsymbol{\eta}^T \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} + \sum_{j=1}^{n_p} \zeta_j \varphi(\|\mathbf{x}_i - \mathbf{x}_{s,j}\|) \right) \quad i = 1 \dots N \tag{21}$$

$$b_i = \begin{cases} 0 & \text{for } d_i \geq R_Z \\ 1 & \text{for } d_i \leq R_F \\ \frac{R_Z - d_i}{R_Z - R_F} & \text{for } R_F < d_i < R_Z \end{cases} \tag{22}$$

$$\varphi(r) = r \tag{23}$$

Constants $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ are chosen to satisfy condition that deformation at the base points should be exactly equal to the input deformation (Eq. (24)). Additional condition is added to allow unique interpolation (Eq. (25)).

$$\Delta \mathbf{x}_{s,i} = \boldsymbol{\eta}^T \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} + \sum_{j=1}^{n_p} \zeta_j \varphi(\|\mathbf{x}_{s,i} - \mathbf{x}_{s,j}\|) \quad i = 1 \dots n_p \tag{24}$$

$$0 = \sum_{j=1}^{n_p} \zeta_j \begin{bmatrix} 1 \\ \mathbf{x}_j \end{bmatrix} \tag{25}$$

Matrix notation is similar as for the POD coefficients interpolation. Coefficients $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ are calculated by solving Eq. (30), where interpolation matrix H is given in Eq. (28).

$$\mathbf{A} = \begin{bmatrix} \varphi(\|\mathbf{x}_{s,1} - \mathbf{x}_{s,1}\|) & \cdots & \varphi(\|\mathbf{x}_{s,1} - \mathbf{x}_{s,n}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_{s,n} - \mathbf{x}_{s,1}\|) & \cdots & \varphi(\|\mathbf{x}_{s,n} - \mathbf{x}_{s,n}\|) \end{bmatrix} \in \mathbb{R}^{n_p \times n_p} \quad (26)$$

$$\mathbf{P} = \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{x}_{s,1} & \cdots & \mathbf{x}_{s,n} \end{bmatrix} \in \mathbb{R}^{4 \times n_p} \quad (27)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{A} \end{bmatrix} \in \mathbb{R}^{(n_p+4) \times (n_p+4)} \quad (28)$$

$$\boldsymbol{\zeta} = \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_{n_p} \end{bmatrix} \in \mathbb{R}^{n_p} \quad (29)$$

$$\mathbf{H} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\zeta} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{X}_s \end{bmatrix} \quad (30)$$

Considering memory requirements, the most efficient way of computing the deformation is to use directly Eq. (21). For the purpose of deriving adjoint equations this expression is transformed to matrix notation by introducing matrices $\boldsymbol{\Phi}$, \mathbf{B} and $\widehat{\mathbf{X}}_s$ (Eq. (31), (32) and (33)), leading to the final expression given in Eq. (35).

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & \mathbf{x}_1 & \varphi(\|\mathbf{x}_1 - \mathbf{x}_{s,1}\|) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_{s,n_p}\|) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{x}_N & \varphi(\|\mathbf{x}_N - \mathbf{x}_{s,1}\|) & \cdots & \varphi(\|\mathbf{x}_N - \mathbf{x}_{s,n_p}\|) \end{bmatrix} \in \mathbb{R}^{N \times (n_p+4)} \quad (31)$$

$$\mathbf{B} = b\mathbf{I}_N \quad (32)$$

$$\widehat{\mathbf{X}}_s = \begin{bmatrix} \mathbf{0} \\ \mathbf{X}_s \end{bmatrix} \quad (33)$$

$$\begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\zeta} \end{bmatrix} = \mathbf{H}^{-1} \widehat{\mathbf{X}}_s \quad (34)$$

$$\mathbf{X} = \mathbf{B}\boldsymbol{\Phi}\mathbf{H}^{-1}\widehat{\mathbf{X}}_s \quad (35)$$

IV. Aerodynamic coefficients gradient formulation

An effective method to calculate gradients of the aerodynamic coefficient is by solving the adjoint equations. Here we derive expressions for flow and mesh adjoint variables, which enables to compute gradient efficiently for large number of design variables.

IV.A. Flow adjoint and mesh deformation adjoint equations

One of the methods to derive discrete adjoint equations is by introducing the Lagrangian function L (Eq. (36)), where I is the objective function $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$ are Lagrange multipliers multiplying flow residual and mesh deformation residual correspondingly. Eq. (37) represents differentiated Lagrangian function with respect to the vector of design variables.

$$L(\mathbf{D}, \mathbf{Q}, \mathbf{X}, \mathbf{\Lambda}, \mathbf{\Gamma}) = I(\mathbf{D}, \mathbf{Q}, \mathbf{X}) + \mathbf{\Lambda}^T \mathbf{R}(\mathbf{D}, \mathbf{Q}, \mathbf{X}) + \mathbf{\Gamma}^T (\mathbf{X} - \mathbf{B}\mathbf{\Phi}\mathbf{H}^{-1}\widehat{\mathbf{X}}_s) \quad (36)$$

$$\begin{aligned} \frac{dL}{d\mathbf{D}} = & \frac{\partial I}{\partial \mathbf{D}} + \mathbf{\Lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{D}} + \left(\frac{\partial \mathbf{Q}}{\partial \mathbf{D}} \right)^T \left(\frac{\partial I}{\partial \mathbf{Q}} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \right)^T \mathbf{\Lambda} \right) \\ & + \left(\frac{\partial \mathbf{X}}{\partial \mathbf{D}} \right)^T \left(\frac{\partial I}{\partial \mathbf{X}} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}} \right)^T \mathbf{\Lambda} + \mathbf{\Gamma} \right) - \mathbf{\Gamma}^T \mathbf{B}\mathbf{\Phi}\mathbf{H}^{-1} \frac{\partial \widehat{\mathbf{X}}_s}{\partial \mathbf{D}} \end{aligned} \quad (37)$$

Computation cost of Lagrangian gradient can be weakly dependent on the number of design variables by eliminating flow variables sensitivities $\partial \mathbf{Q}/\partial \mathbf{D}$ and mesh deformation sensitivities $\partial \mathbf{X}/\partial \mathbf{D}$ from the Eq. (37). It can be done by solving system of equations given in Eq. (38) for $\mathbf{\Lambda}$ and $\mathbf{\Gamma}$.

$$\begin{cases} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \right)^T \mathbf{\Lambda} = -\frac{\partial f}{\partial \mathbf{Q}} \\ \mathbf{\Gamma} = -\left(\frac{\partial f}{\partial \mathbf{X}} + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{X}} \right)^T \mathbf{\Lambda} \right) \end{cases} \quad (38)$$

By convention $\mathbf{\Lambda}$ are flow adjoint variables, and we define mesh adjoint variables $\mathbf{\Lambda}_g$ in Eq. (39), as this expression is more useful than $\mathbf{\Gamma}$ and correspond to similar derivation for elastic mesh deformation method by Nielsen.²⁶ To avoid calculation of the inversion of the H matrix, Eq. (40) can be used to solve for RBF mesh adjoint variables. It is worth noting, that for practical applications right-hand side of the Eq. (40) should not be calculated directly in given form, as matrix $\mathbf{\Phi}$ is large and dense, which might cause memory problems. Instead, iterative equivalent is preferred (Eq. (41)).

$$\mathbf{\Lambda}_g^T = \mathbf{\Gamma}^T \mathbf{B}\mathbf{\Phi}\mathbf{H}^{-1} \quad (39)$$

$$\mathbf{H}\mathbf{\Lambda}_g = \mathbf{\Phi}^T \mathbf{B}\mathbf{\Gamma} \quad (40)$$

$$\mathbf{\Phi}^T \mathbf{B}\mathbf{\Gamma} = \sum_{i=1}^N b_i \mathbf{\Phi}_i^T \mathbf{\Gamma}_i \quad (41)$$

Finally, Eq. (37) is simplified to Eq. (42). For aerodynamic design variables (angle of attack, Mach number), only first two terms are non-zero, and for shape design variables only the last term is relevant.

$$\frac{dL}{d\mathbf{D}} = \frac{\partial I}{\partial \mathbf{D}} + \mathbf{\Lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{D}} - \mathbf{\Lambda}_g^T \frac{\partial \widehat{\mathbf{X}}_s}{\partial \mathbf{D}} \quad (42)$$

The surface deformation field Jacobian (last term in the Eq. (42)) can be obtained using GEO-POD model (Eq. (19)).

V. Results

V.A. Optimization framework

The overview of the optimization framework is given in figure 1. CAD engine is not used in the optimization loop, as GEO-POD serves as the representation of CAD surface. Furthermore, gradients are calculated directly using adjoint and mesh adjoint approach, which avoids the need of performing finite difference calculation with respect to the design variables in any place during the process. Primal and adjoint equations are solved using the DLR TAU unstructured solver. L-BFGS-B algorithm³⁰ is used as an optimizer. This framework can be also used in straightforward way for gradient-free optimization to validate results obtained using GEO-POD + adjoint process.

V.B. DPW-W1 wing optimization

To illustrate this technique, DPW-W1^a CAD model has been parameterized in CATIA, introducing 11 design variables: twist angle at three sections and amplitudes of two Hicks-Henne bumps³¹ applied to the chord of

^aWing from 3rd AIAA Drag Prediction Workshop

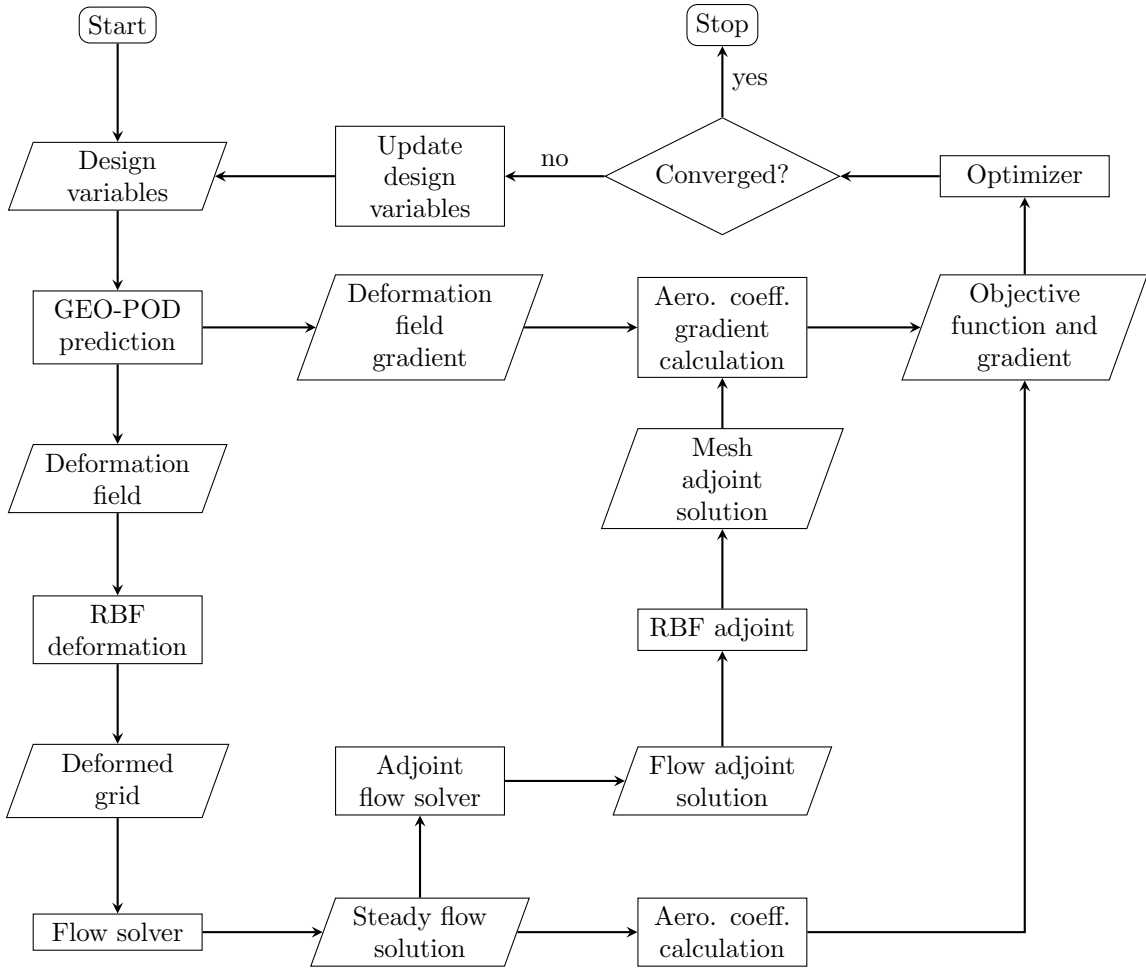


Figure 1: Optimization flow chart

the wing at four sections. Bumps are visualized in the figure 2, and example of wing modification in the figure 3.

V.B.1. Definition of the optimization objective and constraints

We consider the objective of minimizing the negative aerodynamic efficiency (43) with constrained lift coefficient (44). Calculations are performed in target lift coefficient mode (angle of attack is adjusted during convergence to match given C_L constrain). Although this optimization task is equivalent to minimizing C_D , for numerical reasons it is more accurate to consider aerodynamic efficiency, as C_L might differ slightly from the constrained value. Special care has to be taken in deriving objective function gradient, as $\partial C_D / \partial \mathbf{D}$ obtained with adjoint method is valid for unconstrained C_L . To account for constant C_L , a correction term has to be added (45).

$$f = -\frac{C_L}{C_D} \quad (43)$$

$$C_L = 0.5 \quad (44)$$

$$\begin{aligned} \left(\frac{\partial f}{\partial \mathbf{D}} \right)_{C_L=\text{const}} &= \frac{C_L}{C_D^2} \left(\frac{\partial C_D}{\partial \mathbf{D}} \right)_{C_L=\text{const}} \\ &= \frac{C_L}{C_D^2} \left(\frac{\partial C_D}{\partial \mathbf{D}} - \frac{\partial C_D / \partial \alpha}{\partial C_L / \partial \alpha} \frac{\partial C_L}{\partial \mathbf{D}} \right) \end{aligned} \quad (45)$$

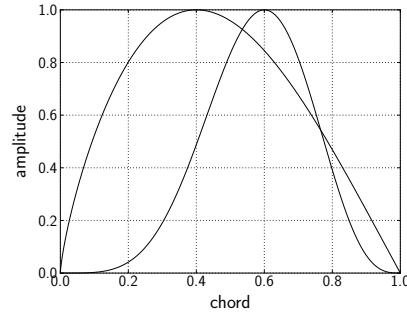


Figure 2: Hicks-Henne bumps applied to wing section chord

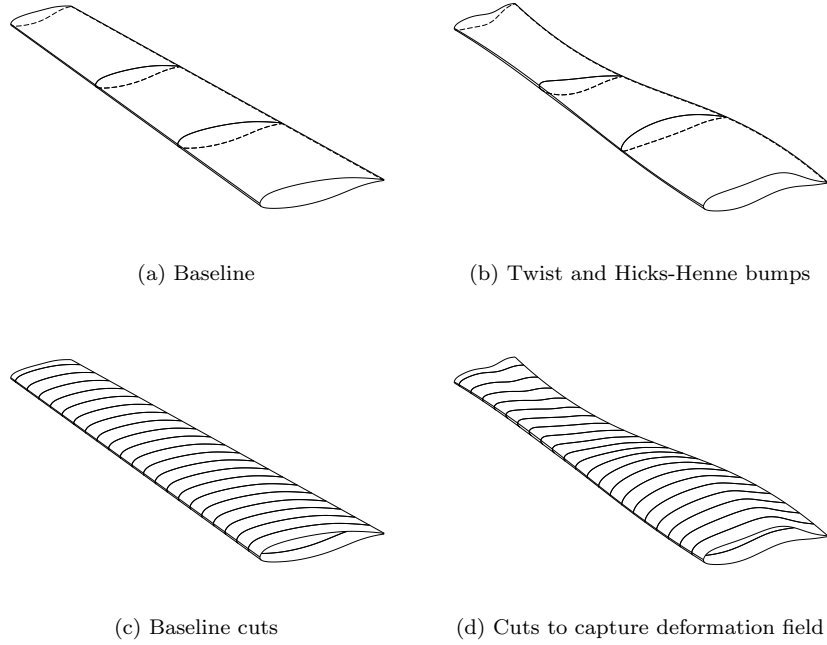


Figure 3: Wing sections parametrization

V.B.2. Optimization results validation with gradient-free method

Results obtained with gradient-based optimization using GEO-POD and adjoint has been validated against gradient-free optimization results. This validation has been performed for twist optimization (3 design variables) and Mach number equal to 0.8. Results show that optimization converges for both cases to the same minimum objective function value, with insignificant differences in final design variables (table 1).

Table 1: Twist optimization (3 design variables) results for $M = 0.8$, comparison to gradient-free method

	p_1 [°]	p_2 [°]	p_3 [°]	C_L	C_D	C_M	I	n_{iter}
Initial	0	0	0	0.5	0.02115	-0.02670	-23.642	
GEO-POD + Adjoint	-4.427	-3.508	-4.041	0.4999	0.02076	-0.02487	-24.073	10
Simplex	-4.512	-3.549	-4.101	0.4999	0.02076	-0.02488	-24.074	115

V.B.3. Optimization with full set of design variables

After validating optimization results for 3 design variables case, optimization has been performed using full set of 11 design variables for Mach number equal to 0.78. Deformation field has been sampled in CATIA using 20, 40 and 200 design points selected with Halton sequence. Optimization converged in about 12 iterations for all the cases (figure 4). Minimal optimal value has been achieved for the most accurate GEO-POD model (200 samples), although the deviation for less accurate models is not significant. Table 2 includes initial and final aerodynamic coefficients obtained using GEO-POD model created with 200 samples, improvement of 16 drag counts has been achieved. A more detailed look into optimization results is provided in the figure 5. As can be seen from pressure coefficient distributions in four sections along the wing span, the shock wave present in the initial design has been eliminated.

Table 2: Twist and Hicks-Henne bumps optimization (11 design variables) results for $M = 0.78$

	C_L	C_D	C_M	I	n_{iter}
Initial	0.5	0.01579	-0.02263	-31.669	
GEO-POD + Adjoint	0.5	0.01420	-0.02214	-35.188	12

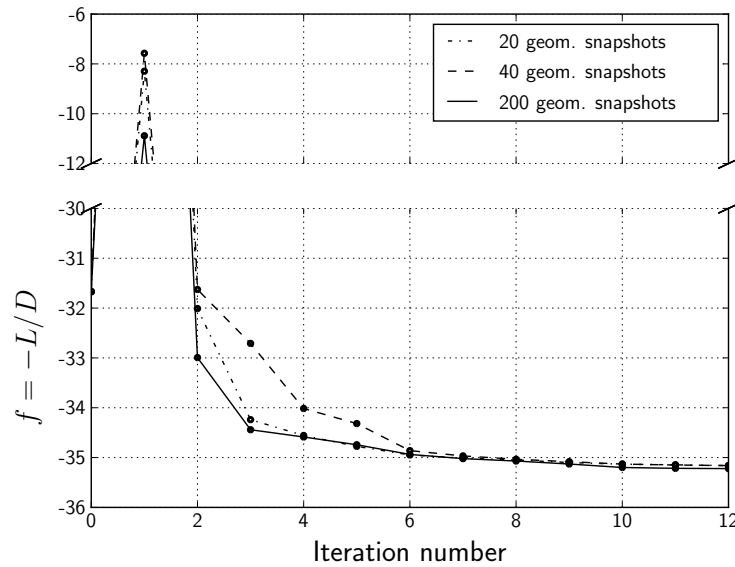


Figure 4: Optimization convergence history

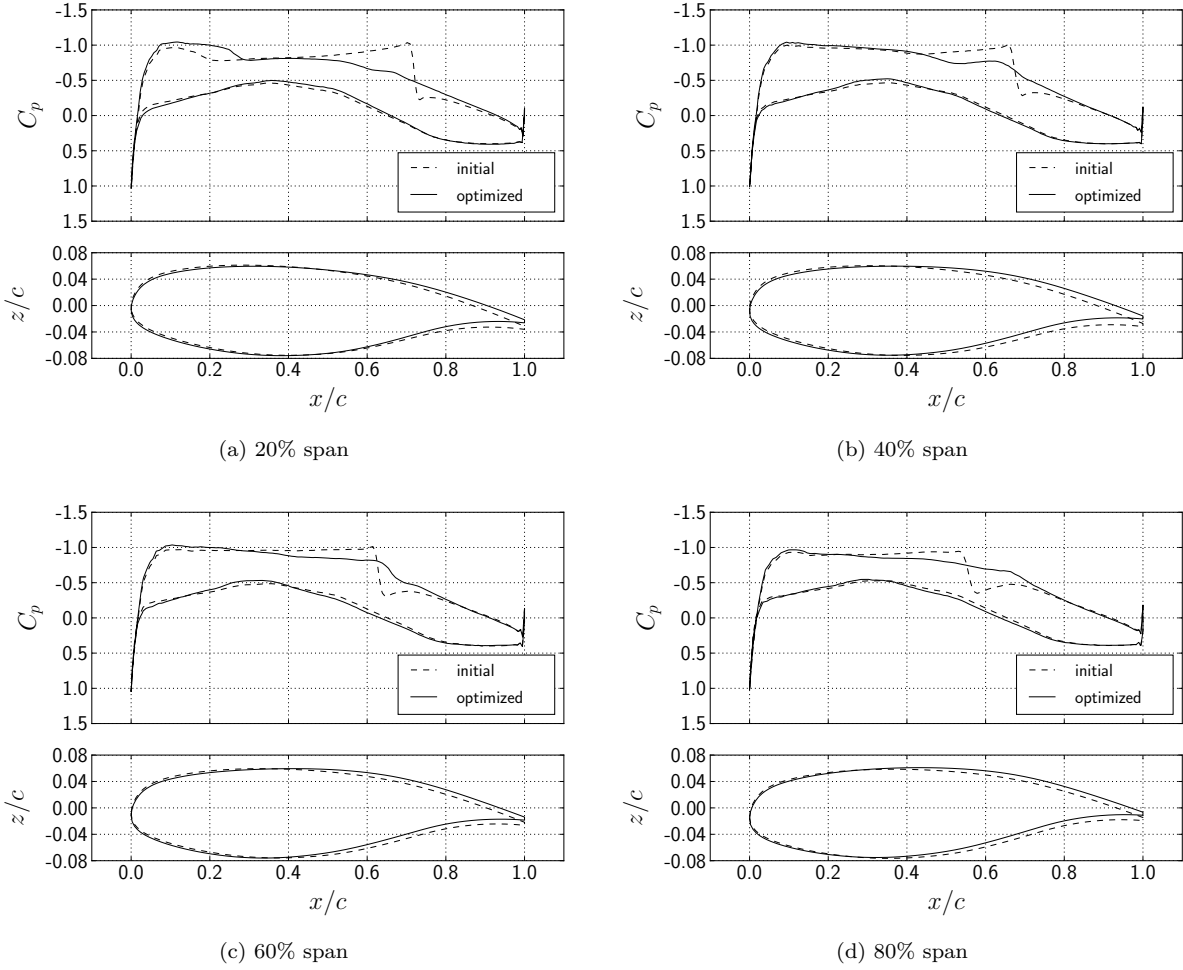


Figure 5: Optimization results, pressure coefficient in wing sections

V.C. Industrial test case: CRM wing-body optimization

To demonstrate applicability of our method to the industrial test cases, Common Research Model (CRM) wing-body configuration has been optimized. Wing has been parameterized similarly to the DPW case with two Hicks-Henne bumps in seven wing sections (overall 14 design variables). Wing sections to which bumps are applied are sketched in Figure 6(a), and CAD cuts to capture the deformation field in Figure 6(b). 500 snapshots were used to create GEO-POD model for wing deformation. Second GEO-POD model has been created for HTP rotation using 10 snapshots. CAD curves capturing the deformation field has been created on both HTP itself and on the fuselage section adjacent to the HTP (Figure 6(d)), since during mesh movement of the HTP surface nodes on the fuselage should be deformed parallel to the surface as well, to provide valid mesh. It is worth noting, that our technique allows for efficient and accurate handling of the intersection lines, in this case between HTP and the fuselage (Figure 6(c)).

Two optimizations have been performed, unconstrained (with HTP fixed in the initial position) and with $C_M = 0$ constraint (HTP rotation activated). Results of unconstrained optimization are presented in Table 3, drag has been reduced by 6.5 drag counts, but pitching moment became more negative, which has a negative effect in the trimmed state (HTP has to exert more downward force to balance it). Constrained optimization (Table 4) reduced drag by 7.6 drag counts with respect to the trimmed baseline configuration, keeping pitching moment equal to zero. Comparison of C_p distribution in four wing sections between trimmed baseline configuration and constrained optimization result is presented in Figure 7.

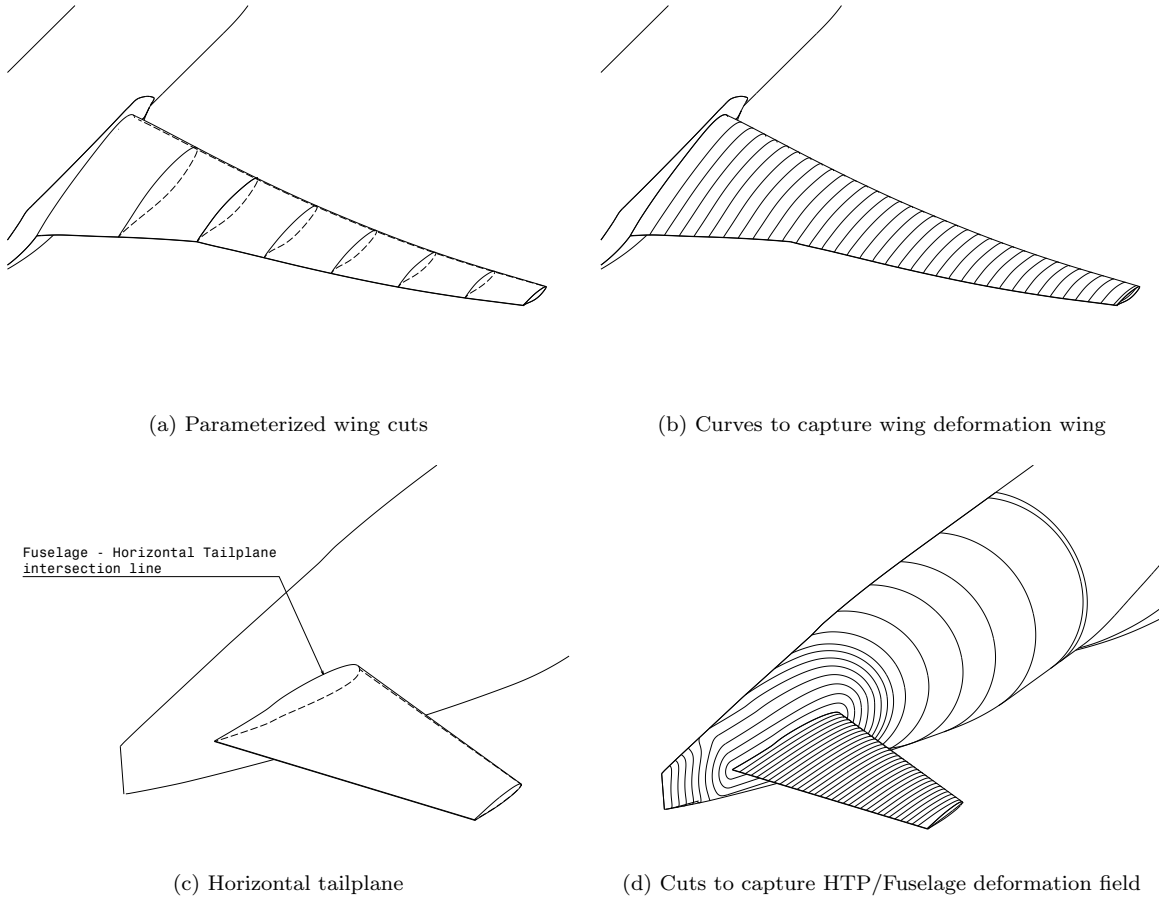


Figure 6: Wing sections parametrization

Table 3: Unconstrained wing optimization results (14 Hicks-Henne bumps)

	C_L	C_D	C_M	n_{iter}
Baseline	0.5	0.02754	-0.03823	
Optimized	0.5	0.02689	-0.04860	10

Table 4: Constrained wing optimization results (14 Hicks-Henne bumps and HTP rotation)

	C_L	C_D	C_M	n_{iter}
Baseline	0.5	0.02754	-0.03823	
Baseline trimmed	0.5	0.02789	0.0	3
Optimized	0.5	0.02713	0.0	22

VI. Conclusions

In this paper we introduce a new CAD-based optimization approach, with the core idea to substitute the CAD engine with surrogate model of CAD surface based on Proper Orthogonal Decomposition. The proposed optimization framework does not use CAD engine in the loop, although shape parameterization is defined in CAD by the designer in the preparation step. Furthermore, the method uses mesh adjoint approach based on Radial Basis Functions to evaluate the gradients. Surface deformation field gradients

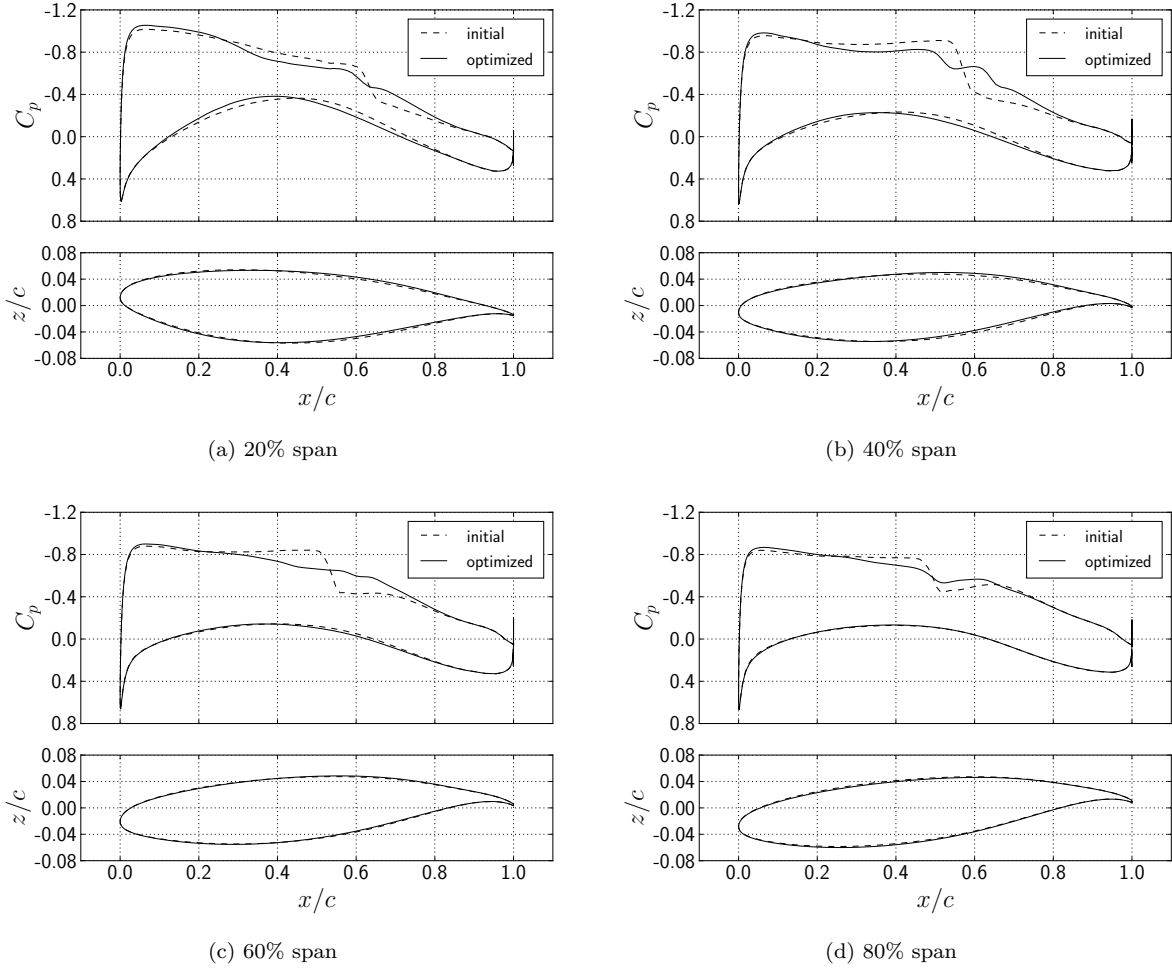


Figure 7: Optimization results (CRM optimization with trimming), pressure coefficient in wing sections

are calculated in an analytical way, which eliminates the need for performing finite difference computations. This significantly improves robustness - for some cases it is impossible to determine the proper step size for finite difference approach, as has been shown in the wing twist example.

Proposed optimization method has been demonstrated on DPW-W1 wing for transsonic case. Firstly, twist optimization results has been validated against gradient-free approach. POD surrogate model coupled with Radial Basis Functions interpolation has proven to predict both surface deformation field and its gradient accurately, leading gradient-based optimizer to the same optimum as achieved by the gradient-free method. In order to demonstrate performance of this method for a larger number of design variables, optimization with Hicks-Henne bumps applied to the wing chord has been performed. It required only few iterations more to converge compared to the twist optimization, leading from the initial solution with a strong shock wave to the final result with a smooth pressure distribution.

Acknowledgments

The first author acknowledge the funding support from the European Commission Seventh Framework Programme (FP7) through the AIRUP^b project (Grant Agreement number: 608087).

References

- ¹Hirschel, E. H., "Towards the virtual product in aircraft design?" *Fluid Dynamics and Aeronautics New Challenges. CIMNE Handbooks on Theory and Engineering Applications of Computational Methods*, Barcelona, Spain, 2003, pp. 453–464.
- ²Vos, J., Rizzi, a., Darracq, D., and Hirschel, E., "Navier–Stokes solvers in European aircraft design," *Progress in Aerospace Sciences*, Vol. 38, No. 8, 2002, pp. 601–697.
- ³Barnewitz, H., "Flexible wing optimisation based on shapes and structures," *MEGADESIGN and MegaOpt-German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, 2009, pp. 287–305.
- ⁴Yu, G., Müller, J. D., Jones, D., and Christakopoulos, F., "CAD-based shape optimisation using adjoint sensitivities," *Computers & Fluids*, Vol. 46, No. 1, 2011, pp. 512–516.
- ⁵Mohammadi, B. and Pironneau, O., "Shape Optimization in Fluid Mechanics," *Annual Review of Fluid Mechanics*, Vol. 36, No. 1, 2004, pp. 255–279.
- ⁶Haimes, R. and Follen, G., "Computational analysis programming interface," *Proc. 6th Int. Conf. On Numerical Grid Generation in Computational Field Simulations*, Eds. Cross, Eiseman, Hauser, Soni & Thompson, 1998.
- ⁷Brock, W. E., Burdyslaw, C., Karman, S. L., Betro, V. C., Hilbert, C. B., and Kyle, W., "Adjoint-Based Design Optimization Using CAD Parameterization Through CAPRI," *50th AIAA Aerospace Sciences Meeting*, 2012.
- ⁸Dannenhoffer, J. F. and Haimes, R., "Design Sensitivity Calculations Directly on CAD-based Geometry," *AIAA paper*, 2015.
- ⁹Damodaran, M. and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, No. 8, 2004.
- ¹⁰Lorente, L. S., Vega, J. M., and Velazquez, A., "Efficient computation of the POD manifold containing the information required to generate a multi-parameter aerodynamic database," *Aerospace Science and Technology*, Vol. 25, 2013, pp. 152–160.
- ¹¹Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and analysis of computer experiments," *Statistical Science*, 1989, pp. 409–423.
- ¹²Nelder, J. A. and Mead, R., "A simplex method for function minimization," *The computer journal*, Vol. 7, No. 4, 1965, pp. 308–313.
- ¹³Boyd, S. and Vandenberghe, L., *Convex optimization*, Cambridge university press, 2004.
- ¹⁴Fletcher, R., *Practical methods of optimization*, John Wiley & Sons, 2013.
- ¹⁵Sorensen, D. C., "Newton's Method with a Model Trust Region Modification," *SIAM Journal on Numerical Analysis*, Vol. 19, No. 2, 1982, pp. 409–426.
- ¹⁶Martins, J. R., Sturdza, P., and Alonso, J. J., "The complex-step derivative approximation," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 29, No. 3, 2003, pp. 245–262.
- ¹⁷Jameson, A., "Aerodynamic design via control theory," *Journal of scientific computing*, Vol. 3, No. 3, 1988, pp. 233–260.
- ¹⁸Jameson, A., Martinelli, L., and Pierce, N. A., "Optimum Aerodynamic Design Using the Navier-Stokes Equations," *Theoretical and Computational Fluid Dynamics*, Vol. 10, No. 1-4, 1998, pp. 213–237.
- ¹⁹Brezillon, J. and Dwight, R., "Discrete Adjoint of the Navier-Stokes Equations for Aerodynamic Shape Optimization," *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, EUROGEN 2005*, 2005.
- ²⁰Nadarajah, S. K. and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach To Automatic Aerodynamic Optimization," *AIAA*, 2000.
- ²¹Nadarajah, S. K. and Jameson, A., "Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization," *AIAA paper*, Vol. 2530, 2001.
- ²²Dwight, R. P. and Brezillon, J., "Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization," *AIAA Journal*, Vol. 44, No. 12, 2006, pp. 3022–3031.
- ²³Stein, K., Tezduyar, T., and Benney, R., "Mesh moving techniques for fluid-structure interactions with large displacements," *Journal of Applied Mechanics*, Vol. 70, No. 1, 2003, pp. 58–63.
- ²⁴Liu, X., Qin, N., and Xia, H., "Fast dynamic grid deformation based on Delaunay graph mapping," *Journal of Computational Physics*, Vol. 211, No. 2, 2006, pp. 405–423.
- ²⁵Barnewitz, H. and Sticka, B., "Improved mesh deformation," *Management and Minimisation of Uncertainties and Errors in Numerical Aerodynamics*, Springer, 2013, pp. 219–243.
- ²⁶Nielsen, E. J. and Park, M. a., "Using An Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design," *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 948 – 953.
- ²⁷Jakobsson, S. and Amoignon, O., "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization," *Computers & Fluids*, Vol. 36, No. 6, 2007, pp. 1119–1136.
- ²⁸Pinnau, R., "Model Reduction via Proper Orthogonal Decomposition," *Model Order Reduction: Theory, Research Aspects and Applications*, Vol. 13, 2008, pp. 95–109.
- ²⁹Zimmermann, R. and Götz, S., "Non-linear reduced order models for steady aerodynamics," *Procedia Computer Science*, Vol. 1, 2012, pp. 165–174.

^bAirbus-UPM European Industrial Doctorate in mathematical methods applied to aircraft design

- ³⁰Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C., “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on Scientific Computing*, Vol. 16, No. 5, 1995, pp. 1190–1208.
- ³¹Hicks, R. M. and Henne, P. A., “Wing design by numerical optimization,” *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.
- ³²Nemec, M., Aftosmis, M. J., and Pulliam, T. H., “CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method,” *AIAA paper*, Vol. 113, 2004.
- ³³Samareh, J. a., “Aerodynamic shape optimization based on free-form deformation,” *AIAA paper*, 2004.
- ³⁴Rendall, T. and Allen, C., “Efficient mesh motion using radial basis functions with data reduction algorithms,” *Journal of Computational Physics*, Vol. 228, No. 17, 2009, pp. 6231–6249.
- ³⁵Alonso, J. J., Martins, J. R. R. A., Reuther, J., Haimes, R., and Crawford, C. A., “High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model,” *AIAA paper*, 2003.
- ³⁶Peter, J. E. and Dwight, R. P., “Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches,” *Computers & Fluids*, Vol. 39, No. 3, 2010, pp. 373–391.
- ³⁷Fudge, D. M., Zingg, D. W., and Haimes, R., “A CAD-Free and a CAD-Based Geometry Control System for Aerodynamic Shape Optimization,” *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.
- ³⁸Jones, D. R., “A Taxonomy of Global Optimization Methods Based on Response Surfaces,” *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 39.
- ³⁹Kroll, N., Gauger, N. R., Brezillon, J., Dwight, R., Fazzolari, A., Vollmer, D., Becker, K., Barnewitz, H., Schulz, V., and Hazra, S., “Flow simulation and shape optimization for aircraft design,” *Journal of Computational and Applied Mathematics*, Vol. 203, No. 2 SPEC. ISS., 2007, pp. 397–411.
- ⁴⁰Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Kevin Tucker, P., “Surrogate-based analysis and optimization,” *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28.
- ⁴¹Nielsen, E. J. and Kleb, B., “Efficient Construction of Discrete Adjoint Operators on Unstructured Grids by Using Complex Variables,” Vol. 44, No. January, 2005, pp. 1–20.
- ⁴²Buhmann, M. D., *Radial basis functions: theory and implementations*, Vol. 12, Cambridge university press, 2003.
- ⁴³Rendall, T. and Allen, C., “Efficient mesh motion using radial basis functions with data reduction algorithms,” *Journal of Computational Physics*, Vol. 228, No. 17, 2009, pp. 6231–6249.
- ⁴⁴Morrison, J. H. and Hemsch, M. J., “Statistical analysis of CFD solutions from the third AIAA drag prediction workshop,” *AIAA paper*, Vol. 254, 2007, pp. 2007.
- ⁴⁵Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, Vol. 23, No. 4, 1997, pp. 550–560.