AeroBest 2021
Lisboa, 21-23 July 2021

AUTOMATIC SHAPE OPTIMIZATION OF A NON-PLANAR WING BASED ON DISCRETE ADJOINT AND RADIAL BASIS FUNCTION MESH DEFORMATION

Luca Abergo[1*], Myles Morelli[2] and Alberto Guardone[3]

Department of Aerospace Science and Technologyr
Politecnico di Milano
Building B12, Via La Masa 34, Milano, MI 20156, Italy

1:Master Student, luca.abergo@mail.polimi.it
2: Postdoctoral Researcher, mylescarlo.morelli@polimi.it
3:Professor, alberto.guardone@polimi.it

Abstract. *The results obtained by an Automatic Shape Optimization tool are strongly effected by the mesh deformation method used. A computationally efficient Radial Basis Function (RBF) grid deformation is coded with two data reduction schemes: multi-level greedy surface point selection algorithms and volume point reduction methods. Following, it is combined with the discrete adjoint inside the open-source software SU2. The robustness of the method, the ability to handle complex shapes and apply large deformations makes possible to optimize also a non planar geometry like a wing provided with a winglet. The surface sensitivity, besides also the final value of the objective function, depends on how the grid is updated, since for the computation of the adjoint variables the process is differentiated by Automatic Differentiation. Finally, the gradient based algorithm "Sequential Least Squares Programming" drives the research of a new local minimum by gradually morphing the geometrical shape*

Keywords: discrete adjoint, design, optimization, mesh deformation, radial basis functions

1 INTRODUCTION

Automatic Shape Optimization (ASO), based on Computational Fluid Dynamics (CFD) and guided by a gradient method, is becoming a powerful tool available to engineers for aircraft design. It can be applied to improve the aero-performance of airfoils, wings and rotors. The Optimization chain described in this paper is implemented in the open-source multiphysics solver SU2 [1], the several modulus needed are implemented in C++ and executed in the correct sequence by a python routine. The search of the optimal shape is driven by a gradient based algorithms named "Sequential Least Squares Programming" (SLSQP) [2], which has been proved to deliver the most robust solution [3]. In this context "optimal" is intended as a morphed form of the original body which provides a significant reduction of the objective function selected J, usually an aerodynamic coefficient or a combination of them. The general problem can be expressed as:

$$\begin{aligned} \min_{\alpha} \quad & J(U(\alpha), X(\alpha)) \\ \text{subject to} \quad & U(\alpha) = G(U(\alpha), X(\alpha)) \\ & X(\alpha) = M(\alpha)\,. \end{aligned} \tag{1}$$

Where X is the computational grid which can be considered as sum of two different components $X = X_{\text{vol}} + X_{\text{surf}} = M(\alpha)$ where $X_{\text{vol}} = X_{\text{vol}}(X_{\text{surf}})$. It means that the volume grid depends explicitly on the wall mesh, that depends in a smooth way on the problem variables α. No assumptions on the structure of M are considered, except that is differentiable. Instead, U is the Reynolds Average Navier-Stokes (RANS) flow state that can be found solving the fixed point equation $U^{n+1} =: G(U^n)$. The local minimum found depends on the starting point and belongs to its close neighbour, besides the final value of J is linked to the amount of geometrical and aero-constraints and the number of design variables.

First, a mathematical description of the wall surface is needed, in this paper the interpolation method Free Form Deformation (FFD) is selected [4]. The body is immersed inside a plastic box, split in smaller bricks which vertices are selected as design variables α whose positions control the geometry. The displacement of the Design Variables (DVs) is obtained projecting into the design space the sensitivity of the target function with respect to the them. FFD can be applied to any kind of mesh and does not modify the connectivity. One the body is properly parametrize, the flow variables U and the current value of the objective function J must be computed, therefore Reynolds Averaged Navier-Stokes equations (RANS) are solved. Regarding this work, the one-equation Spalart Allmaras[5] turbulence model is used without wall functions. The following step is to determine how the objective function is going to change to a perturbation of one of the design variable, thus the surface sensitivity must be computed how it is done in SU2 is explained in Sec. 2. Once it is known how the body has to be morphed, the surface and fluid volume grid has to be adapted. This step is crucial for two reasons. First, the mesh deformation method selected is differentiated inside the process for computing the surface sensitivity therefore the direction of search in the design space is method dependent. Second, how the grid is updated, especially in case of large displacements and complex geometry, can generate a progressive lost in the quality of the mesh jeopardising the entire design process. Historically the first methods were based on an analogy to a continuum: springs or linear elasticity (ELA) [6, 7]. They both require the connectivity of the grid, they well handle

only small deformations and are computationally expensive [8]. Therefore a method for morphing the grid based on interpolation is introduced and coupled with the adjoint, the technique is extensively described in Sec. 3. Concerning shape optimization, ELA is widely used also in SU2 [9–11], thus in this work it is used as comparison with the new optimization chain.
The design loop is repeated until the Karush-Kuhn-Tucker conditions (KKT) are respected or the maximum number of iterations is reached [12]. Most of the times the convergence history is not monotone, the maximum reduction of the objective function is reached before the last loop.

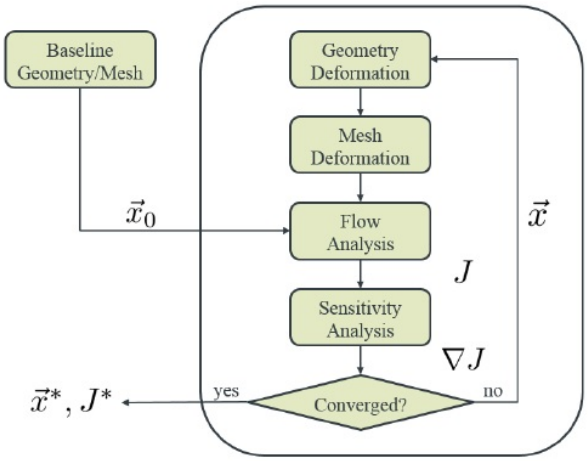

Figure 1: Optimization Design Loop

The comparison between the two ASO is made with the 3D commonly used planar wing the Onera M6 in Sec. 4.1. The cad is than modify to obtain a non planar wing to optimize in Sec. 4.3. The optimization histories are compared also the virtual memory consumption during the mesh deformation step is monitored.

2 SENSITIVITY COMPUTATION

Regarding SU2, $\frac{\mathrm{d}J}{\mathrm{d}\alpha}$ is computed through the discrete adjoint, which has been implemented by Albring, Saugeman and Gauge [13] with a state of art double system method [14]. The required Jacobians are constructed by Automatic Differentiation (AD), coded taking advantage of expression templates and operator overloading. This choice makes possible to differentiate viscous RANS equations with any turbulence model or easily extend the optimization problem to multiphysics pdes. Taking advantage of the library named CoDiPack [15], AD is implemented at statement-level, where the information that need to be stored are independent from the number of operations internally involved [16]. In particular "Reverse Mode" is used since it is attractive for all that kind of problems where a single or a small number of objective functions depend on a large set of variables. Discrete Adjoint is a perfect example since we need the sensitivity of J with respect to the long vector α. In contrast to the forward modality, a single output is selected and the first-order derivative with respect to each of the intermediate variables and the input variables is calculated in an unique process. Reverse AD, coded using expression templates, is 2.7-4 times slower than a direct simulation, making it comparable to the hand-written Jacobians. However, reverse AD requires a large amount of physical memory, some tech-

niques as local preaccumulation and the usage of checkpoints can help to contain the ram consumption [17].
Discrete Adjoint provides the sensitivity with a computational cost scaling with the number of objective functions, therefore not with the length of α as finite difference, and without computing the volume mesh sensitivity.

$$\begin{aligned} \Lambda_f &= \frac{\partial J^T}{\partial U} + \frac{\partial G^T}{\partial U}\Lambda_f \\ \Lambda_g &= \frac{\partial J^T}{\partial X} + \frac{\partial G^T}{\partial X}\Lambda_f \end{aligned} \tag{2}$$

where Λ_f represents the adjoint variables linked with the flow state, meanwhile Λ_g is a set of dual variables related to the grid movement problem.
It must be highlighted that the surface sensitivity still depends on the grid deformation method selected. Moreover, it strongly influence the ability to properly explore the design space, the complexity of the geometry manageable and the dimension of the displacements applied, resulting in a final value of the target function dependent on the method used. Mesh deformation algorithms can generate a slow degradation of the mesh quality which could cause the divergence of the RANS simulation thus even of the optimization process.

3 RADIAL BASIS FUNCTION

In this work, a Radial Basis Function technique, implemented by Morelli and Bellosta [18] for ice accretion prediction, is introduced in the ASO for its robustness and the ability to transfer a large displacement from the surface into the fluid grid. This helps to preserve the quality of the original grid and subsequently improve the robustness of the whole optimization process. As FFD it does not need the grid connectivity, therefore the described optimization chain can handle any type of mesh. It belongs to the class of interpolation methods, it takes as inputs a known boundary displacement field and a continuous base function $\phi(\mathbf{r}, \mathbf{r}_i) = \phi\left(\|\mathbf{r} - \mathbf{r}_i\|\right)$, where the distance corresponds to the radial basis centre, $\mathbf{r}_i$ between two nodes. A certain amounts of control points are selected then, the wall movement is interpolated and propagated in the whole fluid by a weighted sum of basis functions:

$$f(\mathbf{r}) = \sum_{i=1}^{N} \alpha_i \phi\left(\|\mathbf{r} - \mathbf{r}_i\|\right) \tag{3}$$

A linear system to obtain the weight coefficients need to be solved. This method requires the knowledge of the desired displacement of the entire surface grid. Concerning the optimization process implemented in SU2, from the adjoint solution projected into the design space the movement of the vertices of the FFD box is computed. Subsequently, with the free form deformation routine, which is also in this case an interpolation inside the control volume, the surface displacement is obtained. The vector ΔX collect the surface nodes movement which is underlined by the subscript "s", it is described by:

$$\begin{aligned} \Delta X_s &= [\Delta x_{s_1}, \Delta x_{s_2}, \ldots, \Delta x_{N_s}] \\ \Delta Y_s &= [\Delta y_{s_1}, \Delta y_{s_2}, \ldots, \Delta y_{N_s}] \\ \Delta Z_s &= [\Delta z_{s_1}, \Delta z_{s_2}, \ldots, \Delta z_{N_s}] \end{aligned} \tag{4}$$

The three Cartesian directions can be combined in a more simplified formulation:

$$\Delta S = \Delta X_s \hat{x} + \Delta Y_s \hat{y} + \Delta Z_s \hat{z} \tag{5}$$

In analogy also the weight coefficients are collected in a vector:

$$\alpha_x = [\alpha_{x,s_1}, \alpha_{x,s_2}, \ldots, \alpha x, N_s]^T \tag{6}$$

The y and z coefficients are analogous. Following, the weights can be extracted by solving the linear system:

$$\Delta S = \Phi_{s,s} \alpha \tag{7}$$

where Φ is the universal basis matrix, it is generated with the radial basis function evaluated at each surface nodes, meaning that the matrix has size of N_s^2. The compact form of the universal basis function is expressed as:

$$\Phi_{s_j,s_i} = \phi ||r_{s_i} - r_{s_j}|| \tag{8}$$

The next step is to compute the volume base matrix $\Phi_{v,s}$ of size $N_v \times N_s$, where "v" indicates a volume point. Finally, the volume displacement can be interpolated multiplying the above-mentioned matrix with the weights previously computed:

$$\Delta V = \Phi_{v,s} \alpha \tag{9}$$

RBF in the context of design optimization is not widely used due to the elevated computational cost. Even if its potential has been already highlighted by [19], there is no notion of RBF coupled with the discrete version of the adjoint and automatically differentiated. The SU2 versions becomes practicable since it introduces two data reduction schemes to increase the efficiency: multi-level greedy surface point selection algorithms [20] and volume point reduction methods [21]. To guide the greedy algorithm when selecting control points an error vector E is introduced, which is based on the difference between actual surface displacements and computed surface displacements:

$$\mathbf{E} = \Delta S - \mathbf{\Phi}_{s,c} \boldsymbol{\alpha} \tag{10}$$

The node with the largest error is selected as new control point. Denoting the control, surface, and volume points respectively by subscripts c, s, and v and the number of levels is described by the superscript l.Each time that a node is selected the linear system to obtain the weights must be solved. Therefore, the CPU cost of the greedy algorithm is of the order of N_c^4, where the final number of control points selected is the number of iterations of the process plus one. In the case of large displacement of complex geometry the computational cost of the simple greedy scheme becomes too large. The problem has been overtaken by introducing a multi-level subspace radial basis function interpolation, firstly introduced by Wang [20]. The object for the second level of interpolation is set equal to the error of the first step $E^{(0)}$. In a general form, it can be expressed as:

$$\Delta S_{l+1} = E^{(l)} \tag{11}$$

where the next step of the multi-level selection process is indicated by the subscript "l+1". The residual of Eq. 7 at the second level can be expressed as:

$$\Delta S^{(1)} = \Delta S^{(0)} - \Phi W^{(1)} = \Delta S - \Phi(\alpha^{(0)} + \alpha^{(1)}) \tag{12}$$

the size of the displacement is strongly reduced $\Delta S_{l+1} << \Delta S_l$. The computational cost for the multilevel greedy algorithm is now of order of $N_l \times N_c^4$ instead of $(N_l \times N_c)^4$ for the single step. The multi-level selection process can be summarised as follows:

$$\Delta S = \sum_{i=0}^{i=N_l-1} \Delta S^{(i)} = \sum_{i=0}^{i=N_l-1} \mathbf{\Phi}_{s,c}^{(i)} \boldsymbol{\alpha}^{(i)} \tag{13}$$

$$\Delta V = \sum_{i=0}^{i=N_l-1} \Delta V^{(i)} = \sum_{i=0}^{i=N_l-1} \mathbf{\Phi}_{v,c}^{(i)} \boldsymbol{\alpha}^{(i)} \tag{14}$$

The CPU cost of the volume interpolation, after the multi greedy point selection, is now of the order of $N_l \times N_c \times N_v$. In the case of large scale geometry, it is of interest to find how to decrease N_v. One method was proposed by Xie and Liu [21]. They introduced a function which value is based on the distance from the closest wall:

$$\psi = \psi\left(\frac{d(r)}{D}\right). \tag{15}$$

where $d(r)$ is the space distance and D a support value imposed. We define the ratio between the two distances as ξ. The function decays in value when the distance increase and is zero outside the supported distance, so it is a compact support function. It can be expressed as:

$$\psi(\xi) = \begin{cases} (1-\xi) & 0 \leq \xi < 1 \\ 0 & \xi \geq 1 \end{cases} \tag{16}$$

The value of support distance D depends on the maximum surface displacement multiplied for a volume reduction factor k imposed by the used. Mathematically it can be expressed as:

$$D = k(\Delta S_l)^{\max} \tag{17}$$

k in practice defines the range around the body inside which the flow nodes are going to be shifted, besides we are requiring that the elements outside this volume are not affected by the surface movement. It can be notice that is exactly the opposite behaviour respect to the linear elasticity analogy of the previous section where the higher stiffness of the grid's elements close to the wall boundary cases that the distortion is absorbed by the largest elements which are close to the farfield.
In the end the interpolation Eq. 3 is modified in order to include the wall distance correction:

$$f(r) = \psi\left(\frac{d(r)}{D}\right) \sum_{i=1}^{N} \alpha_i \varphi(||r - r_i||) \tag{18}$$

It is attractive the possibility to directly control the intrinsic interpolation error, the consumption of virtual memory and the wall time selecting the base function, the maximum amount of control points and the number of levels for the greedy algorithm.

4 RESULTS

In this section the old framework, with Ela as deformation mesh method, and the new one with RBF are compared. The optimization of a 3D planar wing is presented in Sec. 4.1. Moreover, to show the improved robustness of the new ASO the optimization of a full non planar wing is conducted in Sec. 4.3. Finally, the virtual memory allocation during the single deformation process is monitored and reported in Sec. 4.2

4.1 ONERA M6

The ONERA M6 can be described as a swept, semi-span wing with no twist. The symmetric ONERA D section is used as an airfoil. It is a typical test case for turbulence flow over a transonic wing, widely adopted for CFD validation. Experimental data for the comparison of the pressure distribution are provided in [22]. The flight conditions are chosen to deal with a strong shock on the upper part of the wing collocated close to the 25% of the chord.

Mach	0.84
AoA	3.06
Re	$14.6E6$
Temperature	$300K$

Table 1: Free Stream Conditions Onera M6

Firstly, mesh convergence has been performed, four hybrid meshes are generated. The smallest one is used for optimization. This choice is dictated by the computational power available. The first three layers of the mesh have constant height than the growth ratio decrease from the coarse to the finer grid. The number of layers of the structured part has an opposite behaviour. Since the surface grid is almost everywhere structured, the number of points in the x,y direction are easily multiplied by a factor of 1.5 for the convergence. The following figures report the aerodynamic coefficients obtained and the relative error also the predicted value for an infinity dense mesh is marked. Moreover, the C_p distribution at four different stations is monitored.

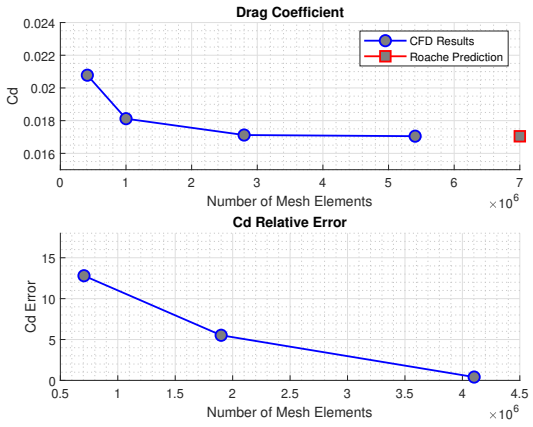

Figure 2: Onera M6: C_d Convergence

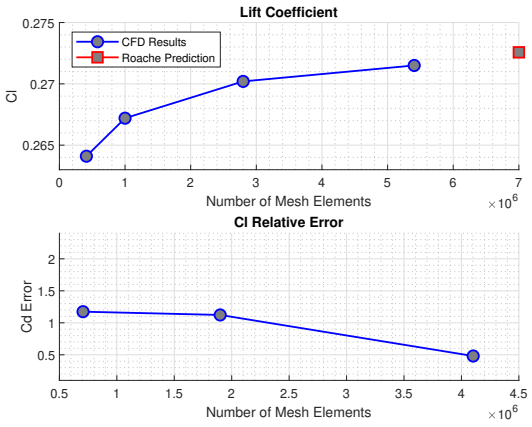

Figure 3: Onera M6: C_l Convergence

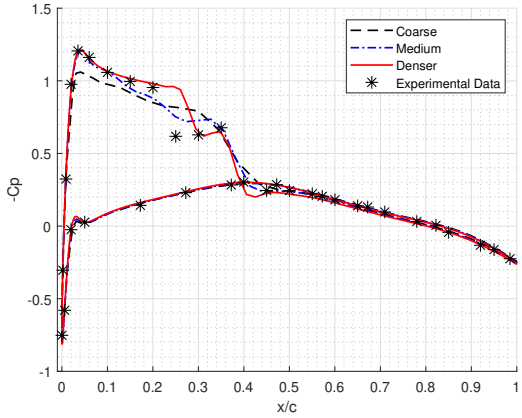

Figure 4: Onera M6: C_p Convergence y=80%b

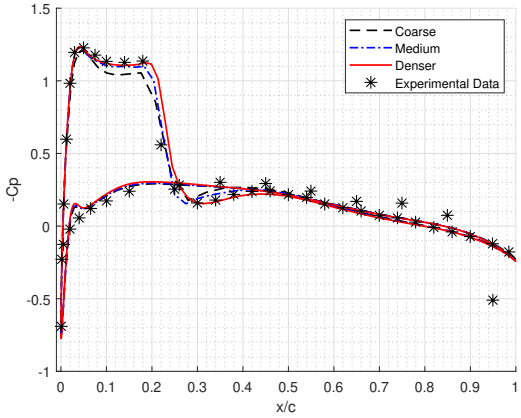

Figure 5: Onera M6: C_p Convergence y=95%b

The goal of the optimization is to reduce the drag without decreasing the lift and the volume of the body. The angle of attack is free to change, the variation of the torque moment is only monitored. Six geometrical constraints are imposed: the final maximum thickness at different span stations cannot be lower than the 75% of the initial value. The gradient based optimization using the SLSQP algorithm is performed once with RBF as mesh deformation method, then with ELA. Regarding RBF, the maximum number of control points selectable is the 10% of the surface's nodes, the volume reduction factor k is set at 5 and the Wendland $C0$ is selected as the interpolation function. This configuration makes RBF really robust, fast, and computationally cheap. Instead, considering ELA a final residual of $10^{(-10)}$ is required for the solution of the linear system and the stiffness of the cells is computed inversely with respect to their volume. Results are reported in the following figures:

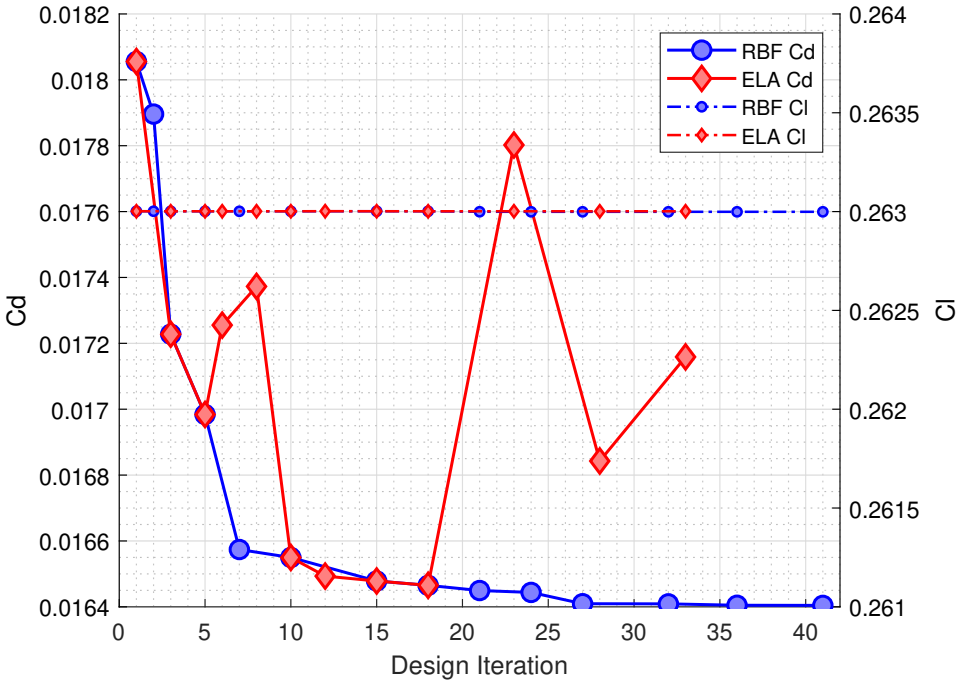

Figure 6: Onera M6: C_d Variation

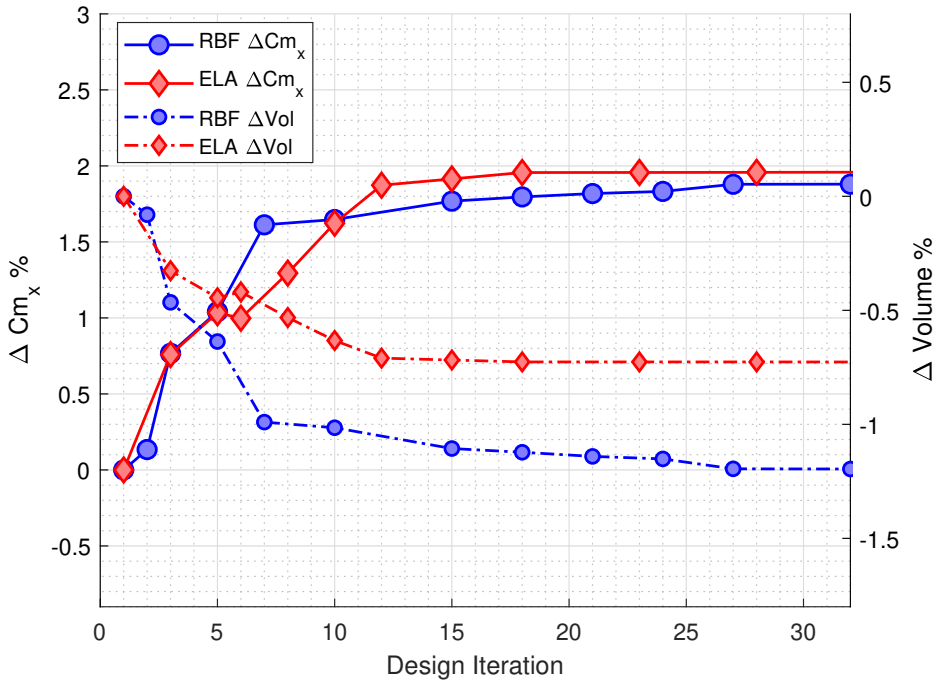

Figure 7: Onera M6: volume and torque variation

Fig 6 clearly shows that both methods are able to conserve the lift and reduce the drag, however the optimization history seems to be more robust using RBF with a more clear convergence. Regarding RBF, the overall drag reduction is around 9.15%, particularly the

drag reduction between the last two loops is insignificant underlying the achievement of a minimum. Instead, ELA obtains a slightly smaller reduction of the C_d around 8.57%. The evolution of the drag with respect to the deformation loops is oscillating, this behaviour has been observed also by other researchers [23]. Further, the optimum is obtained at the $18th$ iteration than a big jump happens and this seems to be typical of the old SU2 optimization chain [24]. Both method increase the torque of the 2% and slightly reduce the volume even if a specific geometrical constraint is imposed. How the shape of the sections and thus the pressure distribution are modified after the optimization is shown in the next pics. The sections close to the root are more morphed, instead the tip of the wing is just more twisted. Especially close to the wing's tip, the peak of suction is decreased and this results in a less intense shock wave, accordingly a lower jump of pressure Fig. 10.

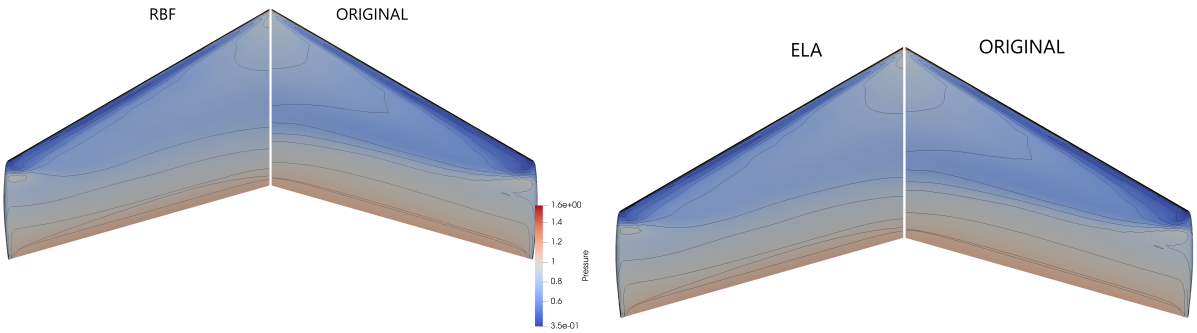

Figure 8: Onera M6: Isopressure Lines Upper Surface

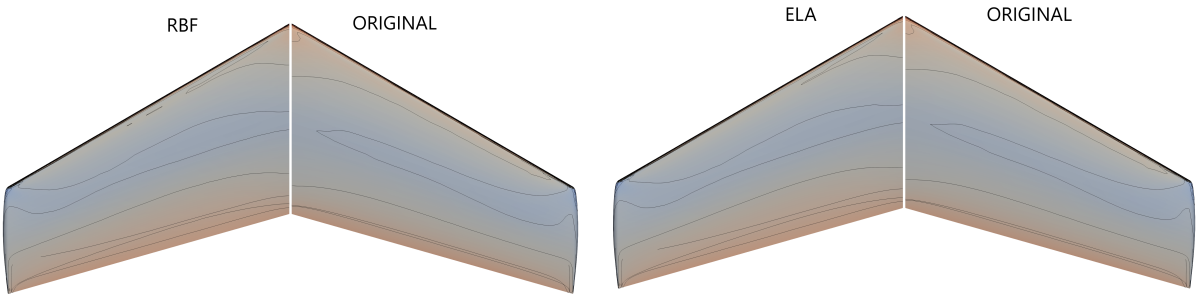

Figure 9: Onera M6: Isopressure Lines Bottom Surface

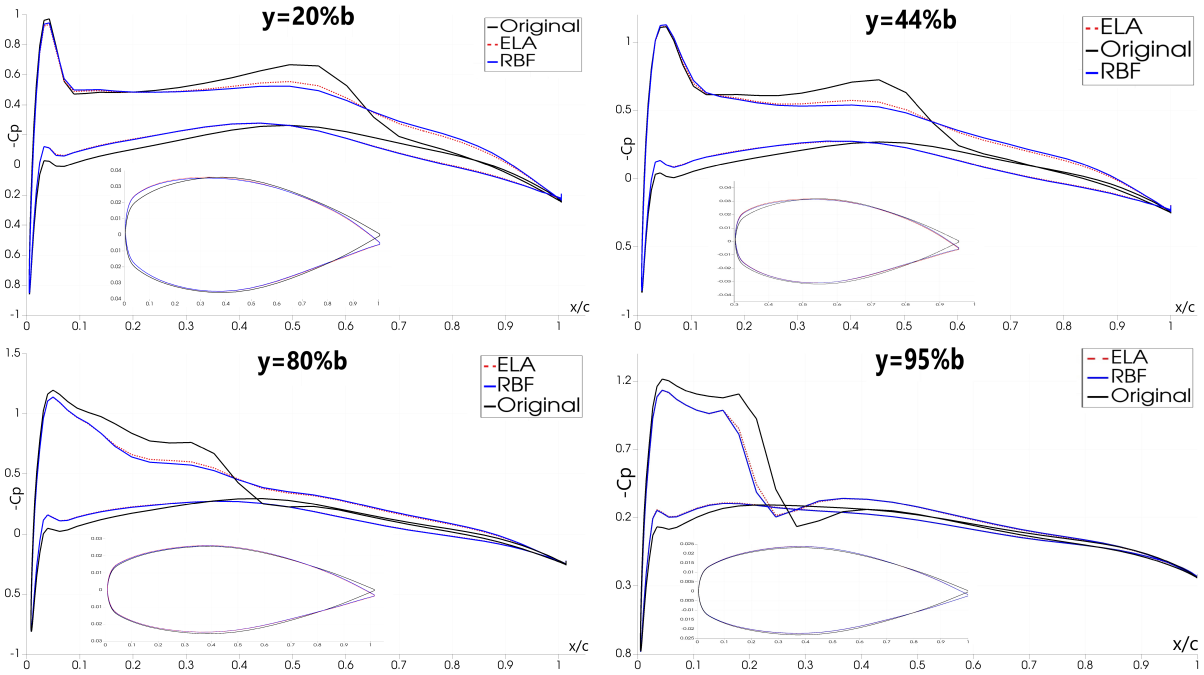

Figure 10: Onera M6: C_p Optimization

Moreover, Fig. 11 shows that the control points selected on the upper part of the wing, the distribution is strictly linked to the shape of the deformation wanted. In this case, the movement was the result of the first loop of optimization and it is evident that the tip is not touched instead the leading edge and the central part of the wing are more affected. The region of greater displacement should correspond to a zone of the surface with higher sensitive. It is confirmed by Fig. 12 where is shown the value of the adjoint variables for the last equation of the adjoint system and the magnitude of the three momentum equations combined.

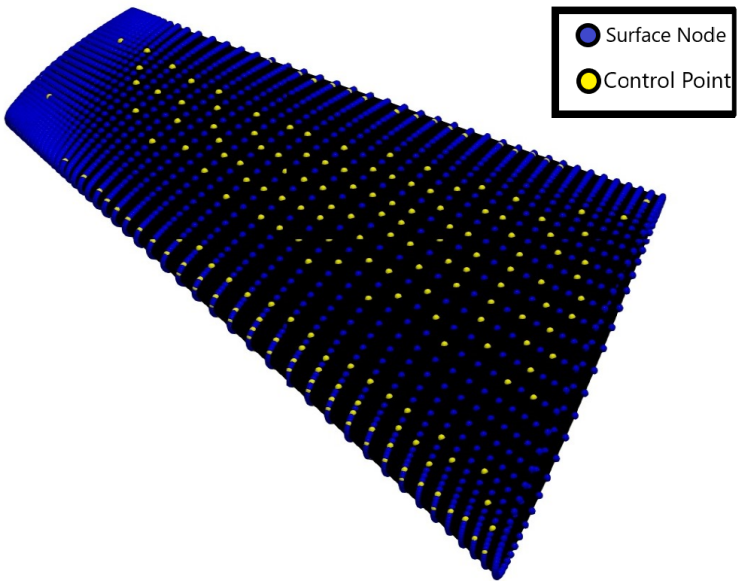

Figure 11: Onera M6: RBF Case 1 Control Points

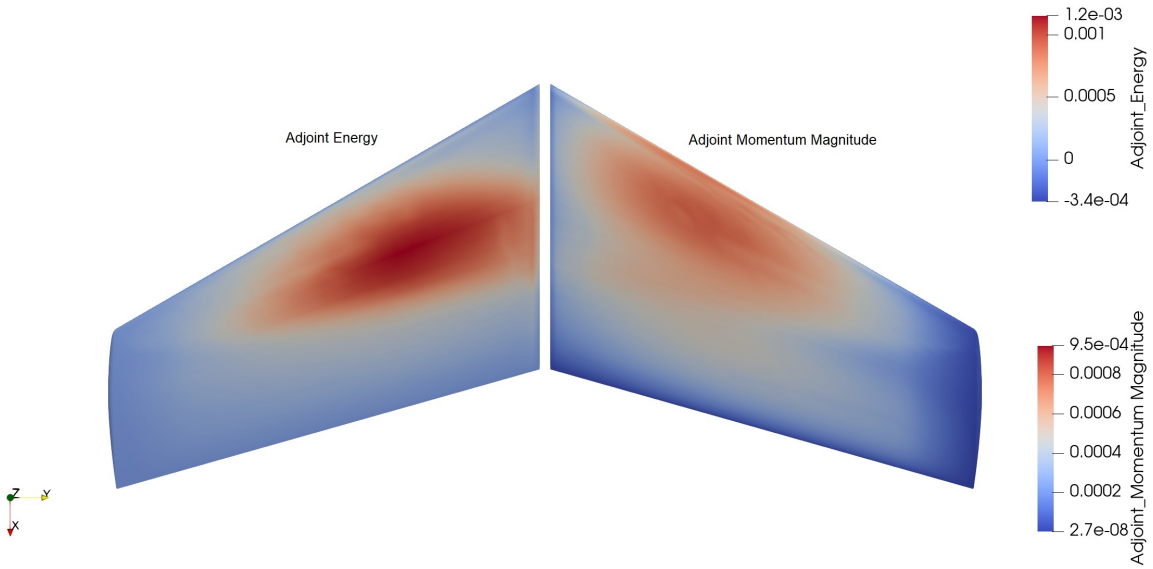

Figure 12: Onera M6: Adjoint Variable

4.2 MESH DEFORMATION PERFORMANCE

The Onera M6 test case has been used to compare the wall time and the ram consumption of the two mesh deformation considered in this text. The grid considered is hybrid, with 7319 surface nodes and 446685 cells. The performances of RBF are measured considering different settings: interpolation function, k, number of levels, amount of control points in percentage of the total surface nodes. The mesh deformation processes are executed on a single core of an AMD EPYC^{TM} of 2.4 GHz. The next table reports the combination of the parameters tested and the results obtained.

N.	Wendland	Control Points	K	N. Levels
0	$C0$	5%	5	one
1	$C0$	10%	5	one
2	$C0$	15%	5	one
3	$C0$	10%	10	one
4	$C0$	10%	5	two
5	$C2$	10%	5	one

Table 2: RBF Parameters

Three outputs are monitored: the maximum virtual memory allocated, the wall time excluded the deallocation of the data, and the error due to the surface interpolation. Regarding ELA, the stiffness of the cells is set with an inverse volume logical. The final linear system has to be solved with a final residual of $10^{(-10)}$ in maximum 800 iterations, this is the classical setting proposed in the SU2 tutorials. The results are reported in the next table:

N.	RAM (G_b)	Interpolation Error	Cpu Time (min)
RBF.0	1.94	2.18%	1.81
RBF.1	2.55	1.07%	3.33
RBF.2	5.41	0.51%	7.89
RBF.3	2.55	0.98%	3.4
RBF.4	3.68	0.47%	6.46
RBF.5	2.74	0.15%	6.53
ELA	14.8	0%	13.66

Table 3: Performance Results

The comparison between ELA and the RBF number one, which is used for the optimization, shows remarkable results. The RAM usage is almost six times lower and the time employed is one quarter. Considering only RBF, in order to have a very low interpolation error maintaining excellent performance, the comparison shows that is better to increase the order of the interpolation function instead of the number of levels of the greedy algorithm. In every 3D optimization ELA has shown a better ability to conserve the volume of the wing, it could be linked to the interpolation error introduced using RBF. This underline the necessity to contain as much as possible the approximation introduced. The following figures show the allocation of the data in the virtual memory with respect to the cpu time. It can be noticed that the behavior of RBF and ELA are completely different, RBF is progressive, instead ELA quickly allocate all necessary information, then the ram consumption remains constant until the linear system is solved.

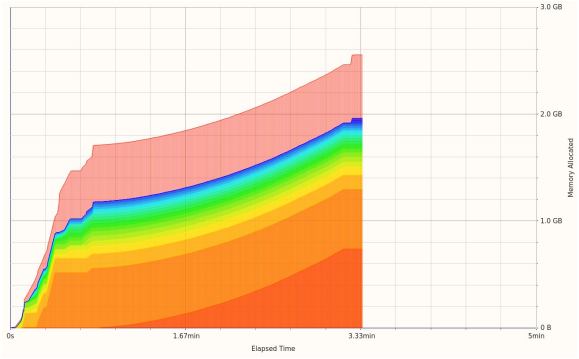

Figure 13: Onera M6: Case 1 Rbf

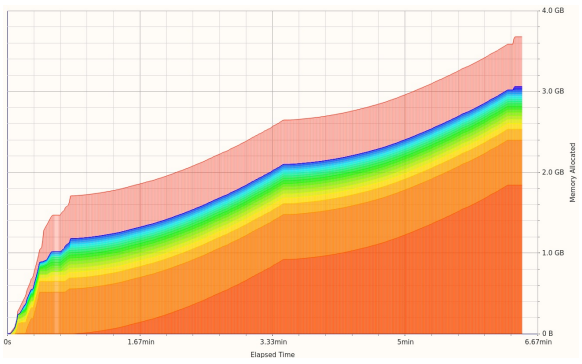

Figure 14: Onera M6: Case 4 Rbf

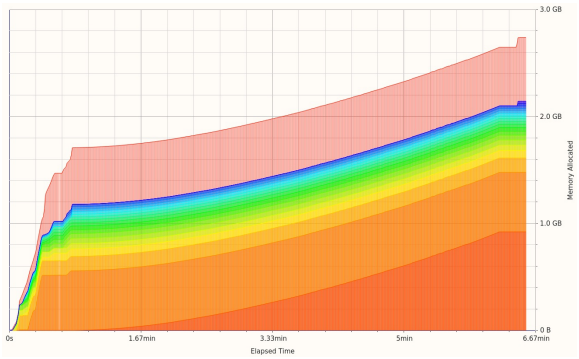

Figure 15: Onera M6: Case 5 Rbf

Figure 16: Onera M6: Performance Ela

4.3 ONERA M6 WITH WINGLET

The optimization of a non planar wing has been conducted in SU2 previously by Palacio in the project NERONE[25]. However, only Euler equations are solved and a new unstructured grid is generated at each iteration loop. In this section a RANS optimization is performed, with SA as turbulence model and RBF for updating the grid which can be of arbitrary topology. The winglet is obtained with a loft from the tip of the old planar configuration, generating a contraction of the final section.

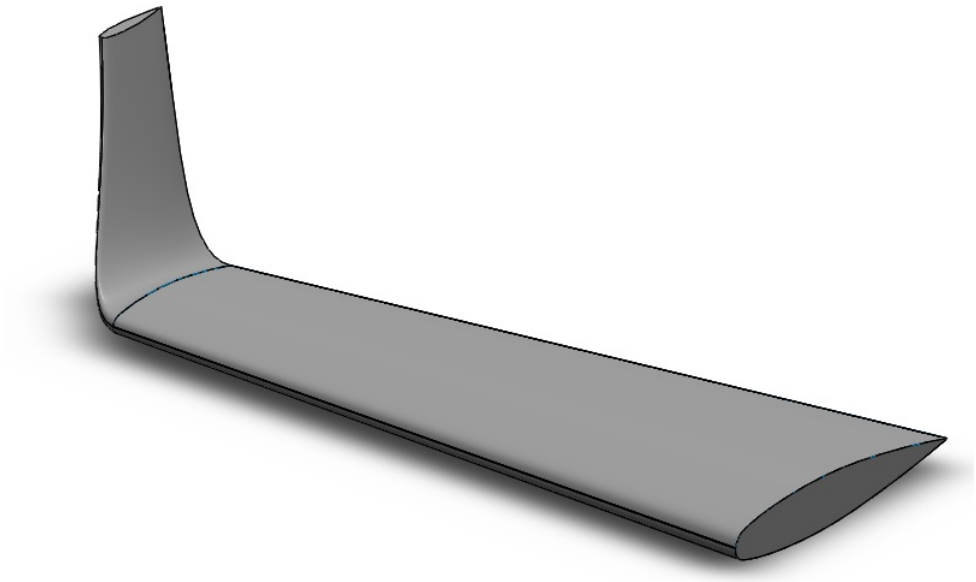

Figure 17: Onera M6 With Winglet Attached CAD

The free stream conditions and the target of the optimizations are the same of Sec. 4.1. The section constraints are imposed also for two station of the winglet so on two planes with z norm. With the same criteria previously used the convergence of the mesh is conducted at fixed C_l and the coarser one is used due to the limited computational resources available. The results has to be intended as the proof that the new framework is able to improve the aero-coefficient of a more complex geometry.

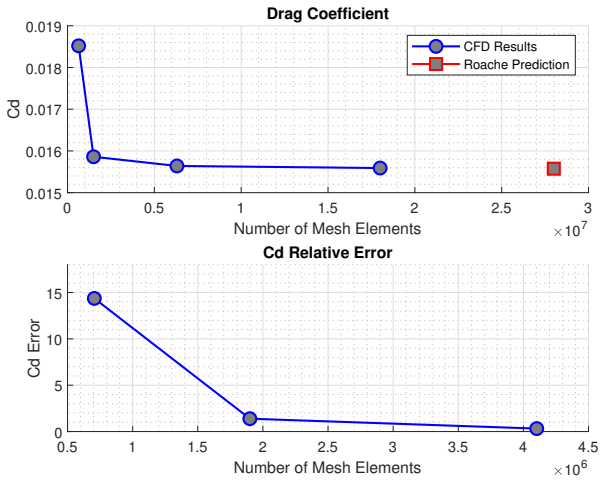

Figure 18: Onera M6 With Winglet Mesh Convergence

The design process is divided in two. At the start, two FFD boxes are selected, the first containing the planar part of the wing using 144 DVs free to translate only in z-direction and the second including only the winglet with 70 DVs that can shift in y,x direction. The sensitivity of the wing DVs is much higher, therefore in this phase the nodes of the planar part are shifted, the winglet is almost untouched. When the drag reduction becomes negligible, the optimization restarts from the last output grid but using only one FFD around the winglet. The DVs are increased to 160, in this phase the winglet is proper morphed and the final shape can be seen in Fig.19.

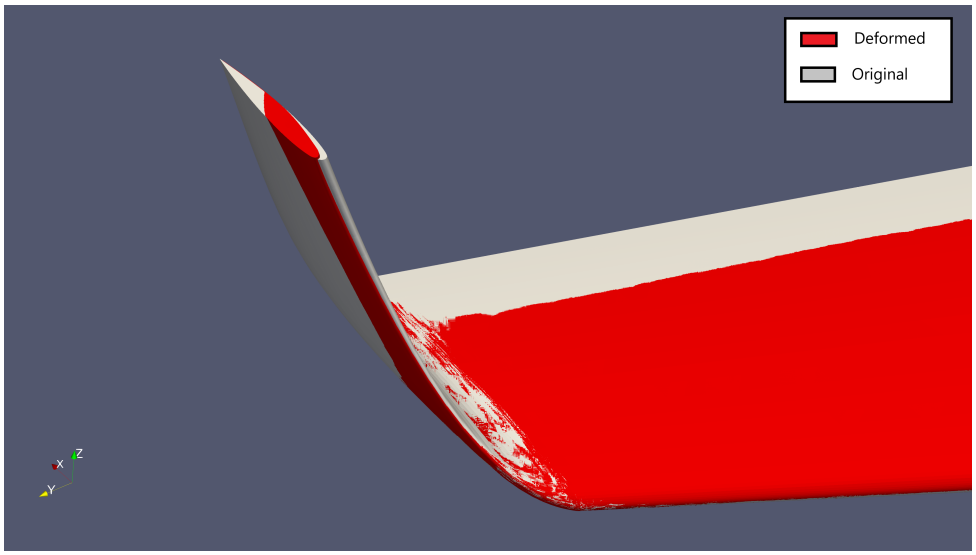

Figure 19: Onera M6 Original and Deformed Winglet

The history of the optimization is reported in Fig.20. A reduction of the drag from 185.34 counts to 163.43 is achieved, therefore the C_d is decreased of 11.82%. The lift and volume constraints are perfectly respected. It must be signaled that the y-momentum, which is not part of the optimization problem but only monitored, strongly increases from -0.112 to -0.148. Figs.21 shows how the C_p distribution and the shape section is modified for four locations, two concerning the winglet and two to the planar part.

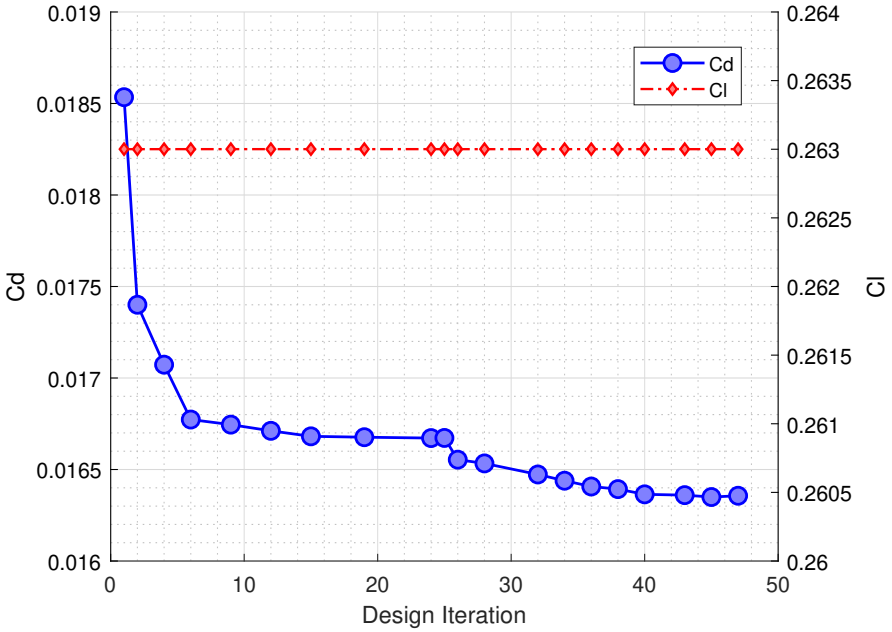

Figure 20: Onera M6 Winglet C_d Optimization

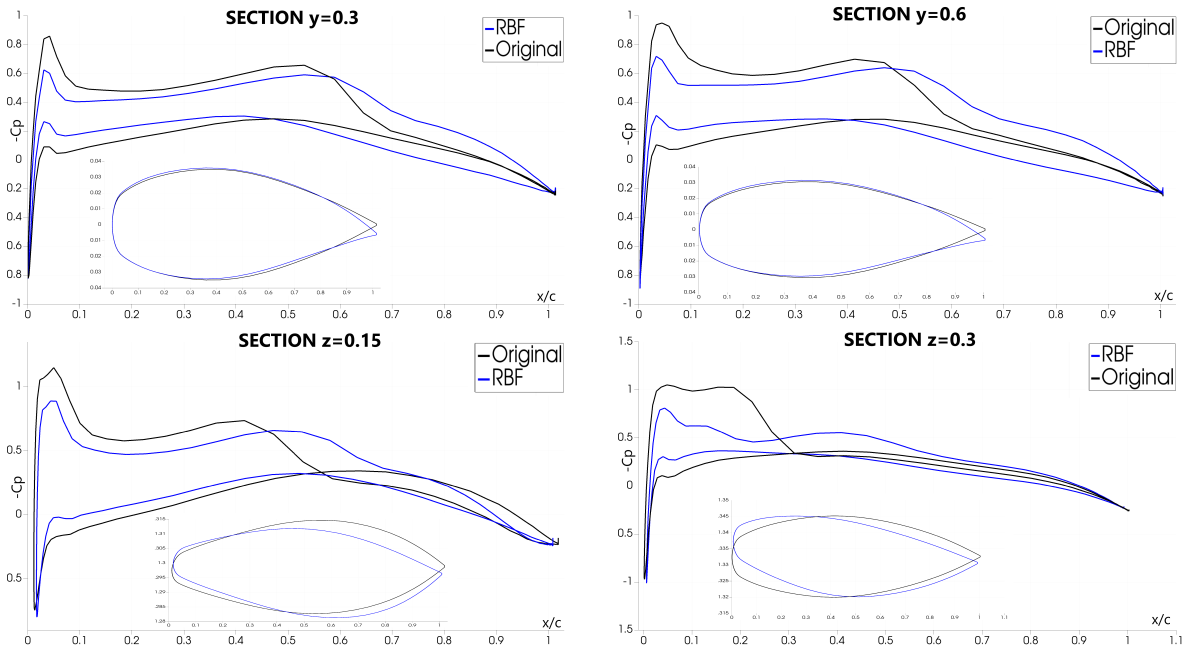

Figure 21: Onera Winglet: C_p Optimization

The optimization was possible only with RBF since it better preserves the quality of the mesh. It is tempted also with ELA where the stiffness is set with an inverse volume criteria. Two other criteria are, constant stiffness and wall distance are also tested, in these cases the lost of quality is more progressive and the design process is jeopardise only after some loops. In the next table it is only reported the orthogonality of the mesh after the first update of the grid.

Method	N_c	N_v	Min Orthogonality
RBF C0	1100	392619	6.32
ELA inv volume	12676	645590	-43.92
ELA constant	12676	645590	6.21
ELA wall distance	12676	645590	6.28

Table 4: Mesh Orthogonality

5 CONCLUSIONS

The aim of this thesis was to increase the robustness and the possibility of the gradient based aerodynamic optimization chain implemented inside the open-source SU2. As optimization in this context it is intended the research of the body shape that provides a lower value of the objective function selected. Strictly concerning SU2 the optimizations using RBF with respects to ELA show a strong reduction of the virtual memory consumption, of the time needed and a better preserved orthogonality of the mesh. The reason is that the version of RBF implemented in SU2 includes two data reduction schemes: a multilevel greedy algorithms and a volume reduction system. Selecting the maximum number of the control points permitted, the levels and the base function it is possible to regulate the computational time and the virtual memory usage. This is extremely useful in case of scarce computational resource available.
The different value of the adjoint variables computed could drive the gradient based SLSQP to a complete different local minimum. RBF showed the possibility to obtain the same or even improve the drag reduction. The viscous optimization in transonic regime of a non planar wing, specifically an Onera M6 with a winglet attached, it is achieved only thanks to the combination of discrete adjoint and RBF, making a step forward respect to the work done in NERONE [25] where the optimization was done only solving Euler equations and regenerating the mesh from zero at each design loop.

REFERENCES

[1] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. Taylor, and J. J. Alonso. Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. *AIAA paper*, 287:2013, 2013.

[2] T. A. Johansen, T. I. Fossen, and S. P. Berge. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *IEEE Transactions on Control Systems Technology*, 12(1):211–216, 2004.

[3] Z. Lyu, Z. Xu, and J. Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *Proceedings of the 8th International Conference on Computational Fluid Dynamics, Chengdu, Sichuan, China*, volume 11. Citeseer, 2014.

[4] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.

[5] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.

[6] J. T. Batina. Unsteady euler airfoil solutions using unstructured dynamic meshes. *AIAA journal*, 28(8):1381–1388, 1990.

[7] R. P. Dwight. Robust mesh deformation using the linear elasticity equations. In *Computational fluid dynamics 2006*, pages 401–406. Springer, 2009.

[8] F. J. Blom. Considerations on the spring analogy. *International journal for numerical methods in fluids*, 32(6):647–668, 2000.

[9] A. H. Truong, C. A. Oldfield, and D. W. Zingg. Mesh movement for a discrete-adjoint newton-krylov algorithm for aerodynamic optimization. *AIAA journal*, 46 (7):1695–1704, 2008.

[10] C. Lee, D. Koo, K. Telidetzki, H. Buckley, H. Gagnon, and D. W. Zingg. Aerodynamic shape optimization of benchmark problems using jetstream. In *53rd AIAA Aerospace Sciences Meeting*, page 0262, 2015.

[11] G. Yang and A. Da Ronch. Aerodynamic shape optimisation of benchmark problems using su2. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0412, 2018.

[12] H. W. Kuhn. Nonlinear programming: a historical view. In *Traces and Emergence of Nonlinear Programming*, pages 393–414. Springer, 2014.

[13] T. A. Albring, M. Sagebaum, and N. R. Gauger. Efficient aerodynamic design using the discrete adjoint method in su2. In *17th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 3518, 2016.

[14] E. J. Nielsen and M. A. Park. Using an adjoint approach to eliminate mesh sensitivities in computational design. *AIAA journal*, 44(5):948–953, 2006.

[15] M. Sagebaum, T. Albring, and N. R. Gauger. High-performance derivative computations using codipack. *ACM Transactions on Mathematical Software (TOMS)*, 45 (4):1–26, 2019.

[16] T. A. Albring, M. Sagebaum, and N. R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. In *16th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 3240, 2015.

[17] A. Griewank and A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[18] M. Morelli, T. Bellosta, and A. Guardone. Efficient radial basis function mesh deformation methods for aircraft icing. *Journal of Computational and Applied Mathematics*, page 113492, 2021.

[19] S. Jakobsson and O. Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, 36(6):1119–1136, 2007.

[20] G. Wang, H. H. Mian, Z.-Y. Ye, and J.-D. Lee. Improved Point Selection Method for Hybrid-Unstructured Mesh Deformation Using Radial Basis Functions. *AIAA Journal*, 53(4):1016–1025, 2015. https://doi.org/10.2514/1.J053304.

[21] L. Xie and H. Liu. Efficient Mesh Motion Using Radial Basis Functions With Volume Grid Points Reduction Algorithm. *Journal of Computational Physics*, 348:401–415, 2017. https://doi.org/10.1016/j.jcp.2017.07.042.

[22] V. Schmitt. Pressure distributions on the onera m6-wing at transonic mach numbers, experimental data base for computer program assessment. *AGARD AR-138*, 1979.

[23] Y. Yu, Z. Lyu, Z. Xu, and J. R. Martins. On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. *Aerospace Science and Technology*, 75:183–199, 2018.

[24] F. Palacios, T. D. Economon, and J. J. Alonso. Large-scale aircraft design using su2. In *53rd AIAA aerospace sciences meeting*, page 1946, 2015.

[25] L. Pustina, R. Cavallaro, and G. Bernardini. Nerone: An open-source based tool for aerodynamic transonic optimization of nonplanar wings. *Aerotecnica Missili & Spazio*, 98(1):85–104, 2019.