ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

EVOLUTIONARY AERODYNAMIC SHAPE OPTIMIZATION THROUGH THE RBF4AERO PLATFORM

Dimitrios H. Kapsoulis[1], Varvara G. Asouti[1], Kyriakos C. Giannakoglou[1], Stefano Porziani[2], Emiliano Costa[2], Corrado Groth[3], Ubaldo Cella[3] and Marco E. Biancolini[3]

[1]National Technical University of Athens (NTUA), School of Mech. Eng.,
Parallel CFD & Optimization Unit, Athens, Greece
e-mail: jim.kapsoulis@gmail.com, vasouti@mail.ntua.gr, kgianna@central.ntua.gr

[2] D'Appolonia S.p.A.
Viale Cesare Pavese 305, 00144 Rome, Italy
e-mail: {stefano.porziani, emiliano.costa}@dappolonia.it

[3] University of Rome Tor Vergata, Dept. of Enterprise Engineering
Via Politecnico 1, 00133, Rome Italy
e-mail: biancolini@ing.uniroma2.it

Keywords: Aerodynamic Optimization, Metamodels, Evolutionary Algorithms, Radial Basis Functions, Mesh Morphing.

Abstract. *This paper demonstrates a way of solving industrial aerodynamic shape optimization problems using the RBF4AERO platform, developed in the framework of the EU–funded RBF4AERO project. The platform combines optimization algorithms (stochastic and gradient–based ones), a mesh morphing tool based on Radial Basis Functions (RBFs) and various evaluation tools (CFD, CSD, etc). In this paper, the Evolutionary Algorithms (EAs) based tool assisted by metamodels trained on a sampling performed during the Design of Experiment phase is used along with a CFD evaluation tool. The use of Response Surface Models (RSM) significantly reduces the number of CFD runs required to reach the optimal solution(s), while the use of RBF–based mesh morphing avoids re–meshing prior to each CFD–based evaluation. In optimization problems, the platform starts by selecting a number of individuals to undergo CFD–based evaluations. The latter constitute the training patterns for the RSM which is, then, used as a low-cost evaluation tool within the EA–based optimization. "Optimal" solutions found by the EA–based search exclusively based on the trained metamodel are then re-evaluated by means of the CFD tool. The database of already evaluated individuals is updated, the RSM is re–trained and the EA–based optimization is repeated. The optimization terminates when convergence criteria related to the RSM prediction accuracy are met or the computational budget is exhausted. The optimization of an ultra-light aircraft and that of the DrivAer car model for minimum drag are showcased. In all cases presented, the simpleFoam solver of OpenFOAM is used to evaluate candidate solutions.*

1 INTRODUCTION

During the last decades, Evolutionary Algorithms (EAs) have widely been applied to shape optimization problems. Though EAs can easily accommodate any evaluation software as a black–box tool, they ask for a great number of evaluations to reach the optimal solution(s) which often makes their utilization in optimization problems prohibitively expensive, especially in industrial problems with excessive computational cost per evaluation. A known remedy to this problem is the use of surrogate evaluation models or metamodels, leading to the so–called metamodel–assisted EAs (MAEAs). Metamodels replace the exact evaluation tool, thus reducing the total number of evaluations required to reach the optimal solution(s).

MAEAs can be classified to on–line and off–line trained ones, based on whether the metamodel training takes place during the evolution or not. On MAEAs with on–line trained metamodels, the EA relies upon the interleaving usage of the problem–specific (to be considered as the high–fidelity) evaluation tool and the metamodel. The entire EA population can be evaluated with any of these tools, [1, 2], by switching either periodically or based on some criteria. Another option is to pre–evaluate all the generation members on the metamodel and use the high–fidelity tool only for the most promising, according to the metamodel, individuals out of them [3, 4, 5, 6]. Metamodels used can be of either local or global support. On the other side, off–line trained metamodels usually rely on a single global metamodel and a sampling technique, often referred to as Design of Experiments (DoE), [7] which defines the appropriate training patterns. Once the metamodel has been trained, this becomes the only evaluation tool during the EA-based search.

In the RBF4AERO platform, developed in the framework of the EU–funded RBF4AERO project, the optimization algorithm relies on off–line trained metamodels. In particular, a polynomial-based Response Surface Model (RSM), trained on appropriately selected samples in the design space, is implemented.

Another important feature of the platform is the mesh morphing tool. Generally, mesh morphing techniques are based on shape parameterization, which can parameterize the surface along with the surrounding nodes of the interior mesh. These techniques allow the interior of the computational mesh to be deformed, avoiding, thus, costly re-meshing. Methods like Radial Basis Functions (RBFs) [8, 9] and volumetric B-splines or NURBS [10] can provide the required parameterization and mesh deformation. The RBF4AERO platform makes use of a morphing tool implementing RBFs, [11]. A number of parameters controlling the positions of groups of RBF control points are used as design variables. Though the RBF4AERO platform may accommodate several CFD tools, for the applications presented in this paper the simpleFoam solver of the OpenFOAM is used, since we are dealing with low speed applications.

In what follows, the basic features of the platform are described and, then, this is used for the re–design/shape optimization of an ultra–light aircraft and a car model.

2 OPTIMIZATION THROUGH THE RBF4AERO PLATFORM

The EA–based optimization algorithm of the RBF4AERO platform is summarized below:

1. Define the RBF shape modifications that require a specific set–up and a steerable parametric mesh.
2. Sample the design space with a DoE technique.
3. Evaluate the above samples on the high–fidelity (herein CFD) tool, after appropriately morphing a pre–existing computational mesh using the RBF–based morpher tool. Store

the so–computed objective function value(s) in the database (DB), paired with the corresponding values of the design variables.

4. Train the RSM using the DB entries as training patterns.
5. Perform EA–based optimization, exclusively based on evaluations on the RSM.
6. Re–evaluate ”optimal” solution(s) resulting from step 5 on the exact evaluation tool and update DB.
7. Return to step 4 until convergence.

In the re–evaluation phase, step 6, there is the possibility to perform additional high–fidelity evaluations, if one area of the design space is not explored properly. Below, the basic features of each step are further described. Note that the optimization algorithm settings, i.e. DoE, RSM and EA parameters are user–defined through the RBF4AERO platform Graphical User Interface.

2.1 Design space sampling

The RBF4AERO platform offers three DoE [12, 13] options to sample the design space, namely the full factorial, partial factorial and randomized designs. Either option decides how many and which candidate solutions will be evaluated on the CFD tool. To perform DoE, each design variable is discretized to a number of user–defined levels. In case of a full factorial design, all possible combinations of design variables' levels are considered. Full factorial designs may result to high CPU cost since all possible combinations must then be evaluated on the computationally expensive tool. The partial (or fractional) factorial design extracts a sub–group from the corresponding full factorial design after cutting some user–selected "less important" variables off. To obtain the partial factorial, a full factorial design is firstly defined using only the "important" variables and the levels of the cut–off variables results from combinations of the other variables' levels. The partial factorial is a better compromise in terms of CPU cost but requires a very good knowledge of the problem in hand in order to correctly decide the design variables to cut–off. Finally, in the randomized design, a user–defined number of designs must be generated. To do so the algorithm computes all possible partial factorial designs and keeps the one with number of members less or equal to the user–defined number of individuals. If the number of collected members is less than the user–defined one, the remaining design vectors are selected from the remaining members of the corresponding full factorial design at random.

2.2 Individuals mesh morphing and evaluation

The individuals determined by the DoE are evaluated on the CFD tool. In this paper, all evaluations are carried out using the steady state solver of OpenFOAM. Instead of re–meshing the computational domain for each and every geometry change during the DoE, an RBF–based morpher undertakes the morphing of a baseline mesh before delivering it to the evaluation manager for the CFD run. In all cases, the baseline mesh is generated using the snappyHexMesh tool of OpenFOAM.

RBFs are mathematical functions able to interpolate data defined at discrete points only (source points) in an n-dimensional environment. The RBF function has the following form

$$s(\mathbf{x}) = \sum_{k=1}^{K} \gamma_k \, \phi(||\mathbf{x} - \mathbf{x}_k||) + h(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ the position vector of a mesh node and γ_k the coefficients of the polynomial, fitted imposing the known values at source points $\mathbf{x}_k$. The interpolation quality and its behavior depends on the chosen ϕ function, being of either global or compact support. In 3D mesh morphing, the bi–harmonic function $\phi(r)=r$ is adopted for its smoothing abilities. The solution of the RBF mathematical problem consists of the computation of the scalar coefficients of a linear system of order equal to the number of considered source points (RBF centers). Once the RBF system coefficients have been computed, the displacement of an arbitrary mesh node can be expressed as a function of the distance-based contributions from the RBF centers.

The RBF method has several advantages that make it very attractive for mesh morphing. Being a mesh–less method, where only grid points are moved regardless their connectivity, its parallelization is quite straightforward. Over and above, it is able to exactly prescribe known deformations onto the surface mesh. This can be achieved by using all the mesh nodes as RBF centers with prescribed displacements, including the simple zero field to denote a surface which is left untouched by the morphing action.

The industrial implementation of the RBF mesh morphing poses two challenges, (a) the numerical complexity related to the solution of the RBF problem for a large number of centers and (b) the definition of suitable paradigms to effectively control shapes using RBF. The RBF Morph software, [11], included in the RBF4AERO platform addresses both challenges as it comes with a fast RBF solver capable to fit large dataset (hundreds of thousands of RBF points can be fitted in a few minutes) and with a suite of modeling tools that allows the user to easily set–up each shape modification.

2.3 Response surface model training

The surrogate evaluation model (or metamodel) implemented in the platform is a Response Surface Model [7] based on polynomial functions. The mathematical expression of the response is

$$\hat{F}(\vec{x}) = b_0 + \sum_{i=1}^{N}\sum_{j=1}^{P_i} b_{ij}x_i^j + \sum_{j=0}^{M} a_j \prod_{i=1}^{N} x_i^{I_i} \tag{2}$$

where $\hat{F}$ is the approximate objective function value, N is the number of design variables, x_i is the i^{th} design variable, M the number of interactions, I_i the power that the i^{th} design variable is raised to and P_i the maximum power for each variable. The RSM coefficients b_{ij}, a_j are computed during the training phase. Interactions, [14], denote the relationship among design variables and are mathematically expressed as multiplications among the related/interacting variables. In case that the number of training patterns exceeds the number of coefficients to be computed, the least–squares method is used during the RSM training. The number of equations constructed by the least–squares method is equal to the number of coefficients. As a result, the cost of the training depends on the number of used coefficients.

The P_i and I_i values in equation 2 can either be selected by the user or defined automatically by minimizing the RSM's error. Note that a different RSM, i.e. with a different configuration (maximum powers, interactions, etc) is trained for each objective function and constraint.

2.4 Optimization using evolutionary algorithms

Once the RSM has been trained, a (μ,λ)EA, with μ parents and λ offspring, undertakes the optimization by exclusively using the RSM as the evaluation tool. A real coded EA with tournament selection for the parents population and simulated binary crossover scheme is implemented. Note that the computational cost of this phase is negligible since the previously trained RSM is used to approximately evaluate all candidate solutions. Upon the termination of the evolution, the "optimal" solution(s) resulting from EA are re–evaluated on the high-fidelity tool and added to the DB.

2.5 Convergence/Termination criteria

The optimization platform includes three convergence criteria, checked upon completion of each optimization cycle. The first convergence criterion is related to the CPU clock time of the optimization (computational budget) and is defined as the maximum number of high-fidelity evaluations that can be performed with the provided budget. This limits the size of the initial sampling and the number of optimization cycles to be performed. The second convergence criterion is related to the RSM prediction accuracy. The optimal solution is considered as found, if the RSM error is very small and its prediction practically replicates the objective function value which results from the problem–specific evaluation tool. The third convergence criterion checks if the "optimal" solution found by the EA is not improving during a user-defined number of evaluations.

3 OPTIMIZATION OF AN ULTRA–LIGHT AIRCRAFT

The first case is concerned with the re–design/optimization of an ultra–light aircraft [15] aiming at the minimization of its drag coefficient. The aircraft geometry was provided by Pipistrel, a light aircraft manufacturer, partner in the RBF4AERO project. In particular, the re–design focuses on the wing root–body junction by defining two boxes. The larger one is used to limit the morphing action and remains fixed thanks to the proper spacing of RBF points. The smaller box, which includes the whole wing, is allowed to move along all three directions, leading to 3 design variables in total (the degrees of freedom are the displacements of the control box in the x, y and z axes). The mesh lying in the space between the small box (which is allowed to change) and the large one (still), including part of the fuselage and wing surfaces, is deformed by the RBF model. A close–up view of the wing root–body junction and the morphing boxes is shown in figure 1.

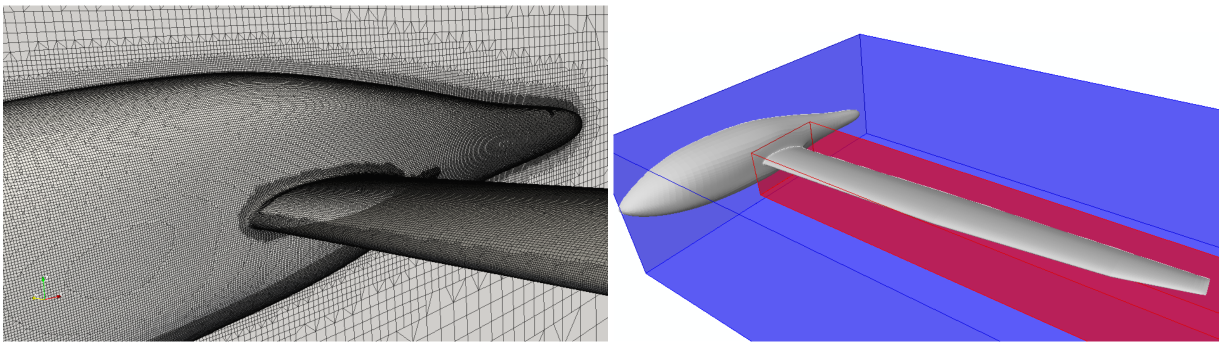


Figure 1: Optimization of an ultra–light aircraft. Close–up view of the wing root–body junction on a surface mesh (left) and RBF control boxes (right).

The flow conditions are $M_\infty = 0.08$, flow angle 10^o and $Re = 10^6$ (based on the wing chord). Each candidate solution is evaluated on the OpenFOAM incompressible solver (simpleFoam) coupled with the Spalart–Allmaras turbulence model with wall functions. The computational mesh around the aircraft is unstructured and consists of about 4.7M cells.

The optimization starts by evaluating, on the aforementioned CFD tool, 45 samples which are used for training a sixth degree ($P_i = 6$ in equation 2) RSM. Then a $(\mu, \lambda) = (15, 30)$ EA undertakes the optimization with a termination criterion of 500 evaluations on the RSM. The "optimal" solution resulting from the EA is re–evaluated on the CFD tool and the RSM is trained anew. Note that each re–training results in different degree and coefficients in equation 2, since the algorithm automatically defines the parameters needed (see section 2.3). This procedure is repeated ten times before meeting the convergence criteria of the optimization procedure (herein, RSM error criterion is firstly met), resulting into a total cost of 55 evaluations on the CFD tool. The convergence history of the optimization is shown in figure 2. Note that the horizontal axis starts from 45 since the optimization started after evaluating 45 samples during the DoE phase.

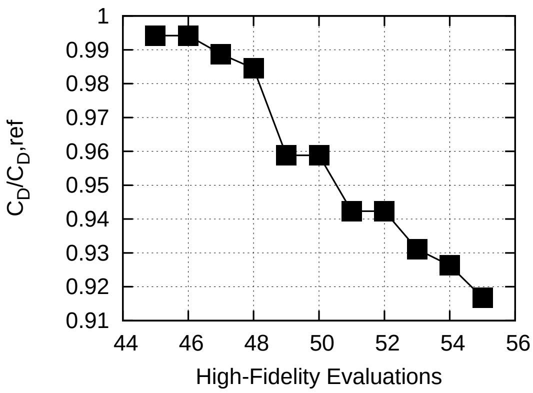


Figure 2: Optimization of an ultra–light aircraft. Convergence history of the optimization.

The optimized geometry yields drag coefficient which is lower by 9% compared to the reference one. The displacement of the junction towards the rear and bottom part of the fuselage (figure 3) is responsible for the observed objective function reduction. A by–product of this optimization is that the lift of the optimized geometry becomes higher, without being included in the objective function. Figures 3, 4 and 5 compare the reference and the optimized aircraft geometries. In all figures, the pressure field on the aircraft surface is plotted.

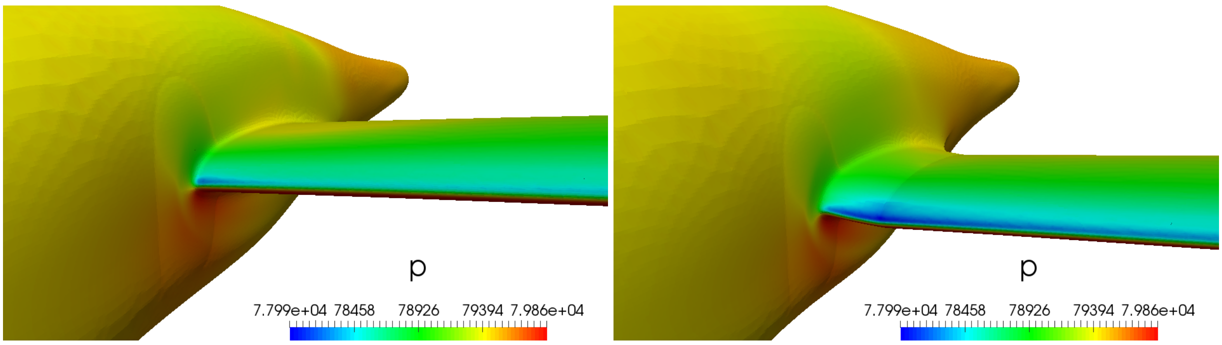


Figure 3: Optimization of an ultra–light aircraft. Comparison of the reference (left) and optimized (right) aircraft geometries. Close–up front view of the wing root–body junction.

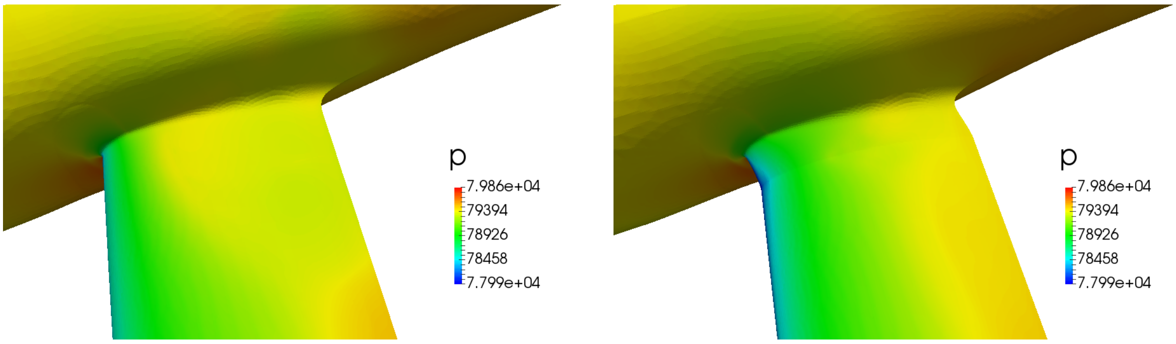


Figure 4: Optimization of an ultra–light aircraft. Comparison of the reference (left) and optimized (right) aircraft geometries. Upper view.

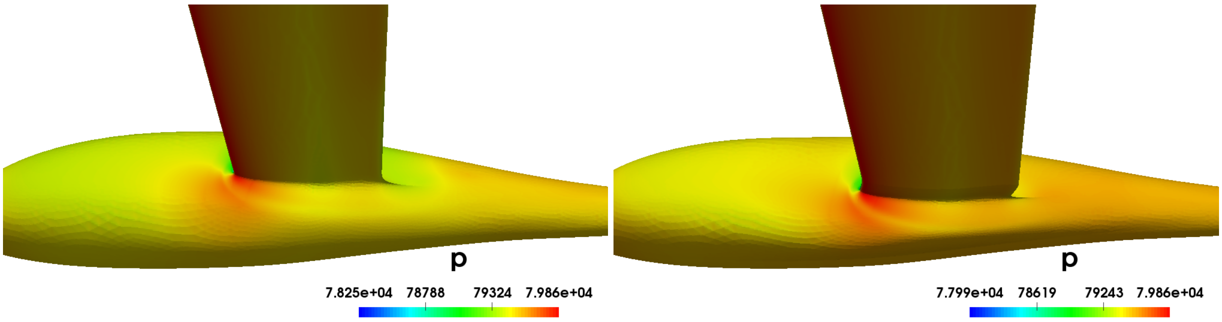


Figure 5: Optimization of an ultra–light aircraft. Comparison of the reference (left) and optimized (right) aircraft geometries. Bottom view.

4 OPTIMIZATION OF THE DRIVAER CAR MODEL

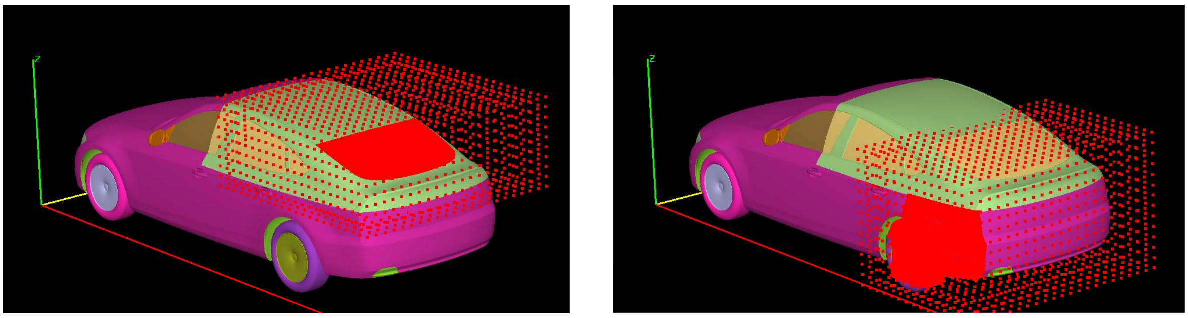
The second case is dealing with the drag minimization of a specific configuration of the DrivAer car model, a generic car model developed at the Institute of Aerodynamics and Fluid Mechanics of TU München, [16], to facilitate aerodynamic investigations of passenger vehicles. Herein, the fast–back configuration with a smooth underbody, mirrors and stationary wheels is used [17]. For this configuration, the generated unstructured mesh consists of about 3.8M cells.

Six design variables related the shape deformation of the mirror, the rear window, the front and back underbody, the car's distance from the road and the boat tail are used. The RBF set–up for two out of the six design variables is shown in figure 6. The evaluation tool is the steady state solver of OpenFOAM with the Spalart–Allmaras turbulence model with wall functions. In some candidate solutions, for which the flow becomes unsteady, the objective function value results from averaging results of the 100 last iterations.

The optimization starts with 20 samples, resulting from a randomized design which are then used to train the initial RSM. A $(25, 50)$EA with 500 evaluations on the RSM is used as described before. A total of 10 optimization cycles was needed to meet the convergence criteria. The convergence history is depicted in figure 7.

After 30 high-fidelity evaluations (including those in the DoE phase), the best solution yields 7% reduction in mean drag. Figures 8 and 9 compare the reference and the optimized car geometries. In all figures, the pressure over the DrivAer model surface is plotted.

Since, the shape optimization of this car model with the same design variables has previously been presented by the same team in [17], using a gradient–based optimization algorithm and the continuous adjoint method for the computation of the objective function gradient, it is interest-



(a) Rear window. (b) Boat tail.

Figure 6: DrivAer car shape optimization. Preview of the RBF source points, before applying morphing, for two out of the six design variables.

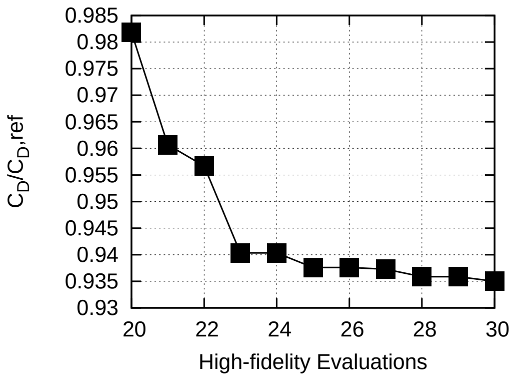
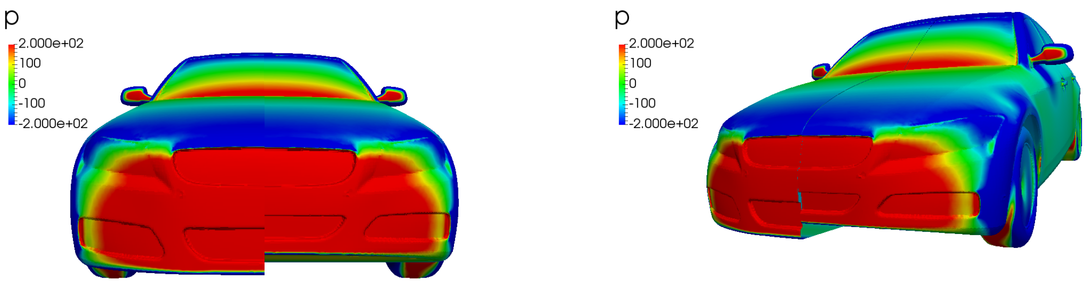


Figure 7: DrivAer car shape optimization. Convergence history of the optimization.



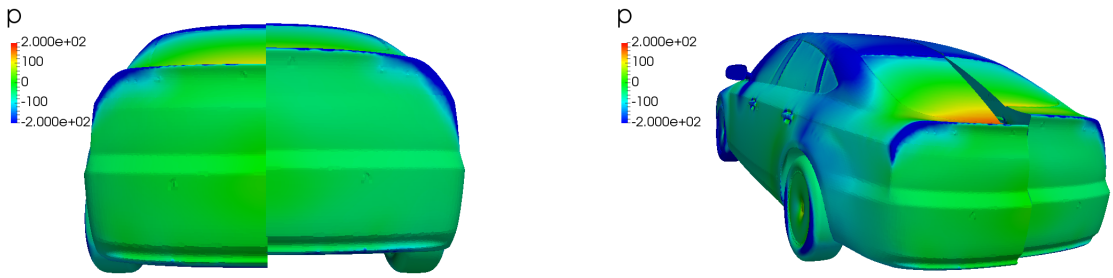
(a) Front view. (b) Side front view.

Figure 8: DrivAer car shape optimization. Comparison of the reference (left) and the optimized (right) car geometries.

ing to compare the resulting optimal solutions. Both methods yield almost similar results at the same computational cost and reduce the objective function value by about 7%. The resulting optimal geometries are similar and small differences can be identified in the front bumper and the spoiler, see figure 10.

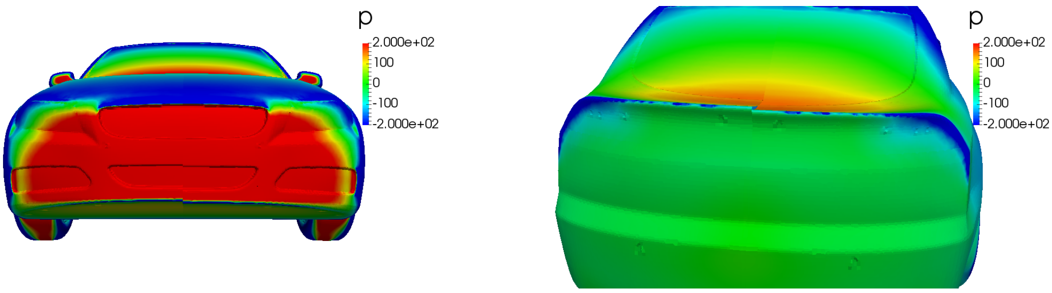
5 CONCLUSIONS

This paper presented EA–based features of the RBF4AERO optimization platform along with real–world applications. The gradient–based features are presented in a companion paper, [18]. The use of RSM as a surrogate evaluation model during the EA–based search allows for



(a) Rear view. (b) Side rear view.

Figure 9: DrivAer car shape optimization. Comparison of the reference (right) and the optimized (left) car geometries.



(a) Front view. EA (left), adjoint (right) (b) Rear view. Adjoint (left), EA (right).

Figure 10: DrivAer car shape optimization. Comparison of optimized geometries resulted from the EA assisted by the RSM and a gradient–based optimization in conjunction with the adjoint method, as presented by the same group in [17].

reduced optimization time. Over and above, the RBF–based mesh morphing reduces further the wall clock time per evaluation to be performed on the high–fidelity (CFD) tool. The overall optimization algorithm described is fully automated and quite user–friendly thanks to the graphic user interface of the RBF4AERO platform.

ACKNOWLEDGMENT

This work was financially supported by the RBF4AERO Project, funded by the European Union 7th Framework Programme (FP7-AAT, 2007 - 2013) under Grant Agreement no. 605396.

REFERENCES

[1] Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, 2002.

[2] D. Büche, N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(2):183–194, 2005.

[3] M.K. Karakasis, A.P. Giotis, and K.C. Giannakoglou. Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization. *International Journal for Numerical Methods in Fluids*, 43(10-11):1149–1166, 2003.

[4] M. Emmerich, K.C. Giannakoglou, and B. Naujoks. Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

[5] M.K. Karakasis and K.C. Giannakoglou. On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Engineering Optimization*, 38(8):941–957, 2006.

[6] J. Branke and C. Schmidt. Faster convergence by means of fitness estimation. *Soft Computing*, 9(1):13–20, 2005.

[7] R.H. Myers and D.C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 2nd Ed.* New York, 2002.

[8] M.E. Biancolini, I. Viola, and M. Riotte. Sails trim optimisation using CFD and RBF mesh morphing. *Computers & Fluids*, (36):1119–1136, 2007.

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

[10] M.J. Martin, E. Andres, C. Lozano, and E. Valero. Volumetric b-splines shape parametrization for aerodynamic shape design. *Aerospace Science and Technology*, (37):26–36, 2014.

[11] M.E. Biancolini. Mesh morphing and smoothing by means of radial basis functions (RBF): A practical example using Fluent and RBF Morph. *Handbook of Research on Computational Science and Engineering: Theory and Practice (vol. 2)*, pages 347–380, 2012.

[12] V.V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.

[13] D.C. Montgomery. *Design and Analysis of Experiments, 5th Ed.* John Willey & Sons, Inc., New York, 2008.

[14] J. F. Kenney and E. S. Keeping. Linear regression and correlation. In *Mathematics of Statistics Pt. 1 3rd ed.*, chapter 15, pages 252–285. Princeton, NJ: Van Nostrand, 1962.

[15] M.E. Biancolini, E. Costa, U. Cella, C. Groth, G. Veble, and M. Andrejasic. Glider fuselage-wing junction optimization using CFD and RBF mesh morphing. *Aircraft Engineering and Aerospace Technology (to appear), DOI: 10.1108/AEAT-12-2014-0211.R1*.

[16] A. Heft, T. Indinger, and N. Adams. Experimental and numerical investigation of the DrivAer model. In *ASME 2012, Symposium on Issues and Perspectives in Automotive Flows*, pages 41–51, Puerto Rico, USA, July 8-12 2012.

[17] E.M. Papoutsis-Kiachagias, S. Porziani, C. Groth, M.E. Biancolini, E. Costa, and K.C Giannakoglou. Aerodynamic optimization of car shapes using the continuous adjoint method and an rbf morpher. In *EUROGEN 2015*, Glasgow, UK, September 14-16 2015.

[18] E. M. Papoutsis-Kiachagias, M. Andrejašič, S. Porziani, C. Groth, D. Eržen, M.E Biancolini, E. Costa, and K.C. Giannakoglou. Combining an RBF-based morpher with continuous adjoint for low-speed aeronautical optimization applications. In *ECCOMAS*, Crete, Greece, June 5-10 2016.