

# Aerodynamic Inverse Design Framework using Discrete Adjoint Method

Joël Brezillon, Mohammad Abu-Zurayk

DLR, Institute of Aerodynamics and Flow Technology  
Lilienthalplatz 7, D-38108 Braunschweig

Joel.Brezillon@dlr.de, Mohammad.Abu-Zurayk@dlr.de

**Abstract** The aerodynamic inverse design approach consists of finding the geometry that produces a desired (target) pressure distribution. Such design problem is here solved using optimization strategies based on CFD solver to minimize the pressure residual. It is observed that the key ingredients for solving this problem are the mesh point parametrization combined with the adjoint approach for efficient and reliable design. The resulting optimization framework is finally successfully assessed on representative 2D design problems ranking from viscous subsonic to inviscid supersonic flows.

## 1 Introduction

Usual aerodynamic optimization is based on global quantity like drag and lift coefficients and tends to improve a given figure of merit, like the lift to drag ratio, by changing the shape of the wing or the airfoil. For the design of laminar profile, specific pressure distribution - such as a smooth continuous pressure decrease at the suction side - is desired to delay as far as possible the Tollmien-Schlichting laminar-turbulent transition point. There is therefore a high need for tools allowing finding the geometry that produces a prescribe target pressure distribution. Such approach is called in this paper the Aerodynamic Inverse Design process.

At the DLR-Institute of Aerodynamics and Flow Technology, a process was developed in the past which relies on an iterative "residual-correction" concept, originally proposed by Takanashi [10]: the pressure residual, defined as the difference between the computed and the prescribed target pressure distributions, is determined by the use of a computational fluid dynamic (CFD) code, while the correction, used to decrease the residual, is approximately obtained by solving the transonic integral equation. Such approach turns to be very fast but may fail on specific cases where strong shock appears.

The basic idea developed here is to solve the Inverse Design problem using an appropriate optimization strategy to minimize the pressure residual. Such approach has the advantage to be robust since no approximation is introduced; furthermore if the designer defined a non feasible pressure distribution (ill posed problem), the optimizer will try finding the geometry that matches as close as physically possible the target pressure. However, the problem is stiff since a small local change of the geometry can have a large impact on the pressure residual: the optimizer needs therefore a large number of evaluations to converge. Furthermore, the airfoil shape has to be parametrised in a way to cover a large range of possible configurations in a smooth way.

## **2 Pyranha: a python based framework for optimization**

An optimization framework is classically composed of two parts: the optimizer and the analysis module, also called task in this paper. The first part drives the analysis module which will pass the goal function to minimize (or to maximize), the constraints and eventually the associated gradients at the design point specified by the optimizer. In most of the time, both parts are separated in a sense that the optimizer may be borrowed from third party and adapted to the analysis module.

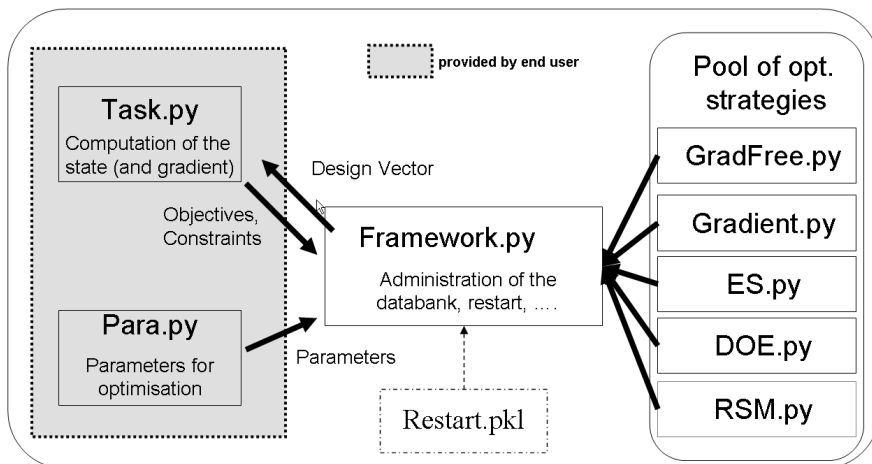
DLR-AS has a long tradition in shape optimization based on CFD and mainly two approaches have been followed so far: the development of frameworks in C or in Fortran with a system call to the analysis module written in shell [1, 13] or the use of commercial frameworks involving shell or Python scripts [4]. These systems proved their capability in solving aerodynamic design problems, but the structures were either not flexible enough to be able to easily integrate new modules, like the adjoint approach, or difficult to be deployed on various operating system or limited by the number of simultaneous use due to the license policy and price.

Alternatively, a new optimization framework based on Python interpreted language has been developed with the major aims to be easily deployed and to allow easy integration of new modules. The main advantages of the Python language rely on its capability to support object-oriented programming constructs, can be advantageously used as scripting language to set-up complex analysis model and is easily extended with new functions implemented in C/C++/Fortran: the whole optimization framework is now unified under a same programming language and the communication between modules is greatly simplified. During the development of the framework, care was taken to rely solely on the standard Python library packages to ensure the maximum flexibility across different operating systems and computers: the minimum restriction is to have Python 2.4 installing.

Regarding the optimizers, various approaches ranking from non-deterministic evolutionary strategies to gradient based and gradient free strategies have been re-coded in Python. The link to the analysis module is ensured via a central frame-

work responsible among other things for the administration of the database automatically generated during the design process. Such database, written in platform independent Python-specific format using the “pickle” module, contains the full history of the optimization and is either used to restart the design process if necessary or to perform post-processing analysis. The inherent parallelism of some of the optimization strategies is ensured via the “multiprocessing” Python module (from Python 2.6) which allows user-defined number of simultaneous evaluation of the analysis module. The central framework already supports connection with (external) gradients, multi-objective, multi-constraint and multi-fidelity data.

The target field of applications for such optimization framework is to conduct shape designs with CFD solvers. The analysis module will therefore contain inhomogeneous program that may run on several platforms. In order to ease the set-up of the analysis module, specific Python modules have been accordingly developed to facilitate communication with external program via command system, the administration of the process with automatic generation of history, error and log files and the launch of processes on different computers.



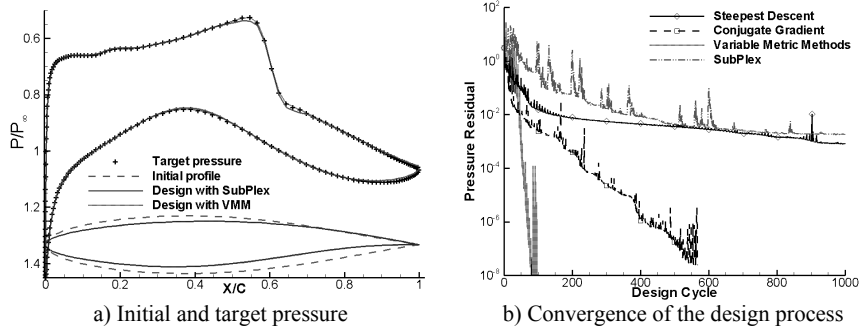
**Fig.1** Structure of the Python based optimization framework Pyranha  
(Python based framework for optimizations relying on high-fidelity approach)

### 3 Analysis module for inverse design shape

The analysis module is responsible for the evaluation of the pressure residual, and eventually of the gradient, for a given design vector. In the present work, the main ingredients are the geometry parametrization, the mesh deformation and the CFD solver. Each component, embedded in a Python script, is now introduced in more details:

- Several parametrization strategies to modify the initial airfoil shape have been implemented. They are either based on B-Splines, Hicks-Henne bump functions, on surface mesh point, or on direct representation of the geometry with the Class Shape Transformation (CST) [7] methods, Bezier curves. In all cases presented here, the initial geometry used is a symmetric NACA type profile.
- The mesh is adapted to the new geometry by deforming an initial mesh using the *TAU* volume deformation module. For viscous computation, the initial mesh is a 2D hybrid unstructured mesh, with 19'981 points, 41 level of hexahedron around the profile for good discretisation of the boundary layer and 200 points at the surface.
- The aerodynamic state is evaluated using the DLR *TAU*-code [5] in viscous mode with the Spallart-Almaras-Edwards one-equation turbulence model. The pressure residual is internally evaluated by the flow solver during the computation. To decrease the turn around time, only a reduced number of CFD iterations is performed and an appropriate use of the CFD restart capability has been developed to ensure the convergence of the optimization process and the CFD solutions at the same time. Following this approach, the flow solution is obtained after 22 seconds of computation on single 2.4GHz Sun AMD Opteron processor. For better stability of the optimization process, the pressure residual is averaged over the last 10 iterations.
- The gradient of the pressure residual relies on the discrete adjoint approach implemented in *TAU* [3]. For the 2D cases here presented, the full discrete flux-Jacobian, including the turbulence model, is stored explicitly and the resulting linear system is solved using an ILU preconditioner to a Krylov subspace method [2]. This method has no restart capability and solves the linear system always from scratch, but needs few seconds (about 90s) to compute the adjoint variables. The mesh sensitivities, needed to get each component of the gradient, are currently evaluated with finite differences. This approach provides accurate gradients but can be time prohibitive on problems with large number of design variables (up to several minutes). To solve this bottleneck in the future, parallel work has already been conducted following the idea of Nielsen and Park [8] and has demonstrated that the resolution of the mesh deformation adjoint permits an accurate evaluation of the mesh sensitivities within few seconds [12].
- The adjoint approach allows the practical design based on mesh point parametrization where the gradient gives the direct shape modification to apply on the airfoil shape. Since the airfoil should remain smooth during the whole design to ensure good conditioning of the design problems, the gradient has to be smoothed to remove the high frequency components. Following the work from Jameson [6] a Sobolev smoother has been implemented in the analysis module and is used only when the mesh point parametrization is employed. The other parametrizations are used with a few number of design parameters and do not required particular smoothing since higher frequencies are not admitted by construction and the design spaces are naturally well posed.

#### 4 Selection of the optimization strategy



**Fig. 2** Demonstration case: parametrization with CST method, 10 design variables ( $M_\infty=0.75$ ,  $AoA=2.0^\circ$ ,  $Re=6.2 \times 10^6$ )

The capability of the optimization framework is first assessed on a feasible problem, where the initial and target pressures are obtained using the same parametrization, mesh topology and flow solver. For the purpose of the demonstration, the target pressure is obtained at transonic case, see Fig. 2a, on an approximation of the RAE2822 geometry using the Class Shape Transformation methods with 10 design variables, equally distributed at the upper and lower face of the profile. Four optimization strategies are selected to check their capabilities to find the feasible target pressure: a Steepest Descent, a Conjugate Gradient, a quasi-Newton approach (Variable Metric Method) [11] and a gradient free Nelder-Mead optimizer working on subspace domains (SubPlex) [9].

According to the convergence of the design process presented in Fig. 2b, the SubPlex approach has a similar convergence rate to the steepest descent and both allow three orders of pressure residual decrease after 1,000 evaluations of the analysis module, also called design cycle. In opposite, the quasi-Newton approach converges extremely fast and matches exactly the target pressure within 100 design cycles, involving 22 evaluations of the gradients. The Conjugate Gradient is indeed slower than the VMM to reach similar low level of pressure residual, but turns to be the fastest approach until pressure residual up to  $10^{-2}$ . The quasi-Newton approach is in fact very efficient once close to the (feasible) solution, where - from the aerodynamic design point of view - the pressure distribution is already close enough to the target pressure. It can be indeed observed in Fig. 2a that the best solution from the SubPlex run is extremely close to the target pressure, with little deviations close to the shock.

This demonstration case has permitted to demonstrate the capability of the framework to retrieve the correct geometry for a target pressure presenting a very strong shock. The quasi-Newton approach is the most efficient strategies tested so far but the Conjugate Gradient was still preferred due to its robustness and efficiency at the beginning of the design.

## 5 Selection of the parametrization

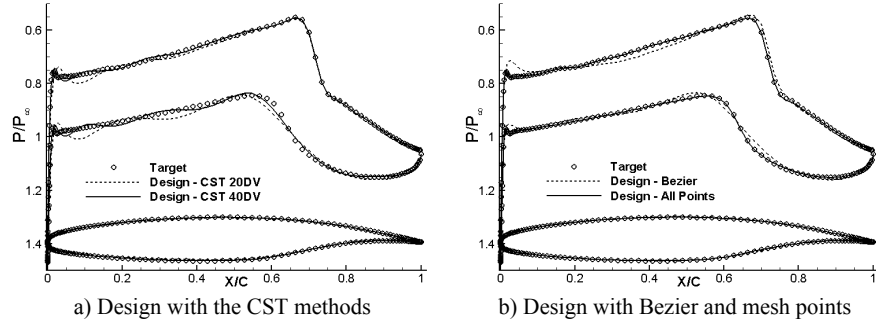


Fig. 3 LV2 test case -  $M_\infty=0.76$ ,  $AoA=0.5^\circ$ ,  $Re=15 \times 10^6$

Since the purpose of application is the design of laminar profile, the DLR-LV2 profile is here selected as target profile to quantify the capability of various type of parametrizations. The profile is characterized by a round nose with large radius and a rear loading. The target pressure was obtained at transonic condition where the profile presents constant pressure increase at the upper and lower side and a strong shock, see Fig. 3a. All optimizations are stopped after 300 design cycles.

The CST method is first tested with increasing number of design parameters from 10 to 40, equally distributed on the upper and lower side of the geometry. Table 1 presents the final designs obtained for each case: as expected the use of larger number of design parameters allows decreasing the pressure residual, the design profile is close to the target geometry but the resulting pressure distributions presents oscillations.

In a second attempt, BSpline and Hicks-Henne bump functions, direct Bezier representation and mesh points parametrization are tested. All designs are well converged after 300 design cycles and Table 1 summarizes the best pressure residuals for each parametrization. For a given number of variables, the direct Bezier representation gives the best matching with almost no pressure oscillation, see Fig. 3b. As expected, the mesh points parametrization allows retrieving the target pressure without oscillation and is thus retained as shape parametrization for the next cases. With this parametrization, the turn around time is currently a little bit less than 15 hours to reach this high level of convergence, but should be reduced to about 4.5 hours solely by using the mesh adjoint approach.

**Table 1** Best pressure residual according to the geometry parametrization

Parametrization	Best	Parametrization	Best
CST – 10DV	0.248	Bezier – 20DV	0.030
CST – 20DV	0.086	BSpline – 20DV	0.061
CST – 40DV	0.018	HH – 20DV	0.097
		Mesh points	0.001

## 6 Demonstration on low speed/transonic/supersonic conditions

Finally, the design process is demonstrated on several conditions, covering low to high Mach numbers, and using exactly the same optimization procedure. The design parameters are the angle of attack and the mesh nodes of the profiles. To assess the capability of the method, only the target pressure, Mach and Reynolds numbers were provided by third party while the target profile and the angle of attack were unknown at the beginning of the design. It is worth to mention that the target pressures are not necessary feasible with the *TAU* code.

For the subsonic case target pressure should be obtained at  $M_\infty=0.3$  and  $Re=10$  millions. After 300 design cycles, the pressure residual decreases by more than 3 orders of magnitude and the resulting pressure distribution matches very well the target pressure, see Fig. 4. In fact the final design profile matches the well known laminar NASA-NLF415 profile at the correct angle of incidence.

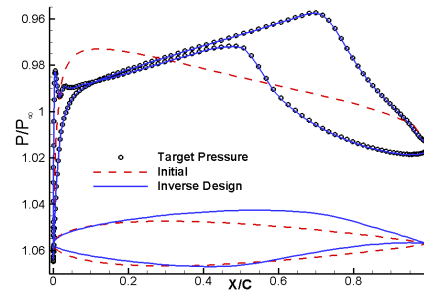


Fig. 4 Subsonic test case

The second test case is defined for  $M_\infty=0.70$  and  $Re=15$  millions. The design process took about 400 design cycles to perfectly retrieve the target pressure, including the correct shock position and strength as well as the rear loading. The resulting supercritical (Whitcomb) profile is characterized by a blunt nose and a very sharp trailing edge at flow condition close to separation near upper trailing edge.

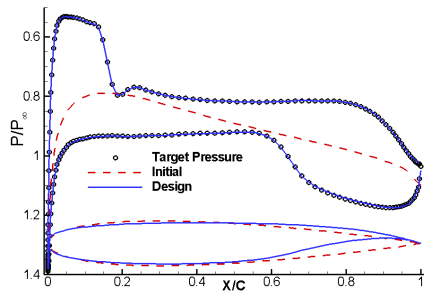


Fig. 5 Transonic test case

The last test case is a pressure distribution to be obtained at  $M_\infty=1.6$ , characterized by a straight pressure increase at the upper side, see Fig. 6. The case is solved using Euler simulations and the pressure residual decreases by more than 5 order magnitude within 400 design cycles. The final geometry produces perfectly the target pressure, including the angular pressure distribution at the leading edge.

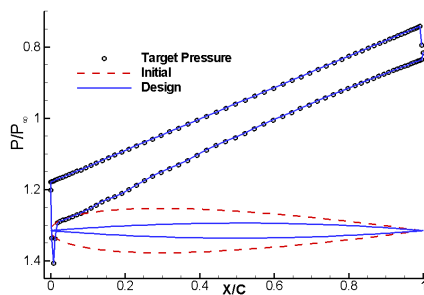


Fig. 6 Supersonic test case

## 7 Conclusion

An optimization framework for solving aerodynamic inverse design problems has been presented and successfully evaluated on various 2D cases. The use of the mesh point parametrization combined with the discrete adjoint appears to be the key elements for successful inverse design in a (relative) limited turn around time.

Future works will concern the implementation of the mesh deformation adjoint for even more efficient gradient computations, the inclusion of global quantities like lift and pitching moment into the optimization problem and the extension of the formulation for 3D problems.

**Acknowledgments** The authors wish to thanks especially Martin Kruse for providing the target pressure cases, Arne Seitz for the fruitful advices on laminar flow and Jen-Der Lee for the discussions on optimization strategies and mesh points parametrization.

## Reference

- 1 Axmann J.K., Hadenfeld M. and Frommann O. (1997) "Parallel Numerical Airplane Wing Design". New Results in Numerical and Experimental Fluid Mechanics: Contributions to the 10th AG STAB/DGLR Symposium", Vieweg Verlag, 1997.
- 2 Balay S., Buschelman K., Gropp W.D., Kaushik D., Knepley M.G., Curfman-McInnes L., Smith B.F. and Zhang, H. (2006) PETSc Web page, 2006. [http://www.-mcs.anl.gov/petsc](http://www.mcs.anl.gov/petsc).
- 3 Dwight R. (2006) "Efficiency Improvements of RANS-Based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids", School of Mathematics, University of Manchester, 2006
- 4 Frommann O. (2002) "SynapsPointer Pro V2.50", Synaps Ingenieur-GmbH, Bremen, Germany 2002.
- 5 Gerhold T., Galle M., Friedrich O., Evans, J. (1997) "Calculation of Complex 3D Configurations Employing the DLR TAU-Code", AIAA paper 97-0167, 1997.
- 6 Jameson A (1990) "Automatic design of transonic airfoils to reduce the shock induced pressure drag", In Proceedings of the 31<sup>st</sup> Israel Annual Conference on Aviation and Aeronautics, Tel Aviv, pages 5-17, February 1990.
- 7 Kulfan B. M. (2001) "Universal Parametric Geometry Representation Method," Journal of Aircraft, Vol. 45, No. 1, pp. 877–884, 2001.
- 8 Nielsen E. J. and Park, M. A. (2005) "Using an adjoint approach to eliminate mesh sensitivities." AIAA-2005-0491, 2005.
- 9 Rowan T. (1990) Functional Stability Analysis of Numerical Algorithms. Dissertation, Department of Computer Sciences, University of Texas at Austin, USA, 1990.
- 10 Takanashi S. (1985) "Iterative Three-Dimensional Transonic Wing Design Using Integral Equations", Journal of Aircraft, Vol. 22, No. 8, August 1985
- 11 Vanderplaats G.N. (1984) "Numerical Optimization Techniques for Engineering Design", McGraw-Hill Series in Mechanical Engineering, ISBN 0-07-066964-3, 1984.
- 12 Widhalm M., Brezillon J., Caslav I. and Leicht T (2010) "Investigation on Adjoint Based Gradient Computations for Realistic 3d Aero-Optimization.", 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, 13 - 15 Sep 2010, Fort Worth, Texas.
- 13 Wild J. (2000) "Validation of Numerical Optimization of High-Lift Multi-Element Airfoils based on Navier-Stokes-Equations", Paper 2000-2939, AIAA, 2000.