11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

ADJOINT OPTIMIZATION OF 2D-AIRFOILS IN INCOMPRESSIBLE FLOWS

M. Schramm[1*], B. Stoevesandt[1] and J. Peinke[2]

[1] Fraunhofer IWES, Ammerländer Heerstr. 136, 26129 Oldenburg,
matthias.schramm@iwes.fraunhofer.de, bernhard.stoevesandt@iwes.fraunhofer.de,
www.iwes.fraunhofer.de
[2] ForWind, Ammerländer Heerstr. 136, 26129 Oldenburg, joachim.peinke@forwind.de,
www.forwind.de

Key words: Continuous Adjoint Method, Constrained Optimization, Airfoil, Wind Energy, Wind Turbine, OpenFOAM®, Incompressible Flow.

Abstract. Gradient-based optimization using the adjoint approach is a common method in shape optimization. Here, it is used to maximize the lift to drag ratio of a two-dimensional DU 91-W2-250 profile. The methods principle is explained as well as the implementation in the open source CFD code OpenFOAM. First, the optimization is done for different angles of attack without any constraints. Then, the cross-section area of the airfoil is held constant and also the effect of approximate gradients is shown.

1 INTRODUCTION

In aerodynamic shape design a high lift to drag ratio is desired in many cases such as wind turbine blade layout. Furthermore, several constraints are necessary for reasons of stability, roughness insensitivity, noise and others [1]. This problem can be modelled by an optimization process, which is the minimization (or maximization) of one or more defined cost functions with respect to given constraints. In CFD it may be formulated as:

$$Minimize \;\; I(\mathbf{u}, p, \boldsymbol{\beta}) \;\; subject\ to \;\; \mathbf{R}(\mathbf{u}, p, \boldsymbol{\beta}) = 0 \;\; and \;\; \mathbf{G}(\mathbf{u}, p, \boldsymbol{\beta}) \leq 0 \;\; , \tag{1}$$

where I is the cost or objective function, $\boldsymbol{R}$ are the Navier-Stokes equations and $\boldsymbol{G}$ are the geometric inequality constraints. The arguments are the velocity $\boldsymbol{u}$, the pressure p and the design parameters $\boldsymbol{\beta}$ of the optimization. Due to the relatively low tip speed of wind turbines, the flow can be considered as incompressible and the steady Navier-Stokes equations may be written as:

$$\mathbf{R}^{\mathbf{u}} = (\mathbf{u} \cdot \bigtriangledown)\mathbf{u} + \bigtriangledown p - \bigtriangledown \cdot 2\nu D(\mathbf{u}) \tag{2}$$

$$R^p = - \bigtriangledown \cdot \mathbf{u} \quad , \tag{3}$$

where ν is the kinematic viscosity and $D(\mathbf{u})$ refers to the rate-of-strain tensor $1/2[\bigtriangledown \mathbf{u} + (\bigtriangledown \mathbf{u})^T]$. Note the negative sign in the continuity Equation (3), which is chosen according to the notation of Othmer [2] and is continued in the following.

2 OPTIMIZATION USING THE ADJOINT APPROACH

In order to solve the problem (1) several methods are available, whereas common approaches use genetic algorithms or gradient-based algorithms [3]. Here, we assume to start with an airfoil relatively close to the optimum, which is why the gradient-based approach is chosen. Using the method of steepest descent [4], the new geometry is given by:

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} - \alpha \cdot \delta L \quad , \tag{4}$$

wherein δL is the variation of the functional L, which will be defined later, and α is the step size, which is limited by the maximum mesh movement. A trade-off between fast convergence and smooth or stable mesh movement has to be found. For instance, good results are obtained using an initial step size of 0.001 of the chord length.
Using the chain rule, the gradient of the cost function I can be calculated as:

$$\delta I = \frac{\partial I}{\partial \boldsymbol{\beta}} \delta \boldsymbol{\beta} + \frac{\partial I}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial I}{\partial p} \delta p \quad , \tag{5}$$

which means that for every change of a design parameter a new evaluation of the flow field is necessary. This is usually very expensive and limits the use of Equation (5) to few design parameters. A common approach to solve this problem is the use of a Lagrangian function, including the Navier-Stokes equations as equality constraints [5]:

$$L := I + \int_{\Omega} (\boldsymbol{\Psi}^u, \Psi^p) \cdot \mathbf{R} \; d\Omega \quad , \tag{6}$$

wherein Ω is the computational domain and the Lagrangian multipliers $\boldsymbol{\Psi}^u$ and Ψ^p refer to adjoint velocity and adjoint pressure, respectively. The variation of the Lagrangian then leads to:

$$\delta L = \frac{\partial L}{\partial \boldsymbol{\beta}} \delta \boldsymbol{\beta} + \frac{\partial L}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial L}{\partial p} \delta p \quad . \tag{7}$$

And the principle of the adjoint approach is to choose $(\boldsymbol{\Psi}^u, \Psi^p)$ in such a way that the second and third term of the right hand side of Equation (7) become zero:

$$\frac{\partial L}{\partial \mathbf{u}} \delta \mathbf{u} + \frac{\partial L}{\partial p} \delta p = 0 \quad . \tag{8}$$

Then a variation of the flow field $(\delta \boldsymbol{u}, \delta p)$ is not affecting the gradient δL, which means that $\delta L = \delta L / \delta \boldsymbol{\beta}$, and the solution is independent of the number of design parameters. This is why the adjoint approach is often used in case of a detailed parameterization, which is also done in this work. Here, each boundary face of the shape is free to move and has two design variables defining the position of the face center (leading to 720 design parameters in this work).
Using the variation of the Navier-Stokes equations and integration by parts, Equation (8) delivers a new set of partial differential equations, the adjoint Navier-Stokes equations:

$$-\nabla \boldsymbol{\Psi}^{\mathbf{u}} \cdot \mathbf{u} - (\mathbf{u} \cdot \nabla) \boldsymbol{\Psi}^{\mathbf{u}} = -\nabla \Psi^{p} + \nabla \cdot (2\nu \mathbf{D}(\boldsymbol{\Psi}^{\mathbf{u}})) \tag{9}$$

$$\nabla \cdot \boldsymbol{\Psi}^{\mathbf{u}} = 0 \quad . \tag{10}$$

Their detailed derivation is omitted, but can be found in literature [2, 5, 6]. Due to the notation in Equation (3) the pressure gradient in Equation (9) is negative [2]. The variation of the eddy viscosity ν is neglected, which is commonly done in case of the continuous adjoint approach and is referred to as "frozen turbulence" [7]. It is worth to note that in most cases as in shape optimization Equations (9) and (10) are not depending on the cost functions, but need specific boundary conditions, which include derivatives of these functions.

2.1 Cost function

The interest of this work is the determination of an airfoil shape that maximizes the ratio of lift to drag coefficients. Instead of choosing $I = c_l / c_d$, it can be reformulated using a weighted function with the both coefficients, which is done for simplicity of the cost function derivatives. Then, one possible formulation could be:

$$I = w_d \cdot c_d - w_l \cdot c_l \quad . \tag{11}$$

Lift and drag usually are of different order, which is why the weighting factors w_d and w_l are used. They are calculated in such a way that the two terms on the right hand side in Equation (11) are of similar size. A minimization of I then leads to a minimization of drag and a maximization of lift.

2.2 Boundary conditions

The boundary conditions corresponding to the cost function in Equation (11) need the derivative w.r.t. the primal pressure p:

$$\frac{\partial I}{\partial p} = w_d \frac{\partial c_d}{\partial p} - w_l \frac{\partial c_l}{\partial p} \quad . \tag{12}$$

The objective function is only defined on the airfoil (wall boundary), which means that derivatives of the cost function are zero on other boundaries. Referring to Othmer [2]

the boundary conditions then can be calculated as follows (the indices t and n indicate tangential and normal components, respectively):

Inlet:

$$\mathbf{\Psi^u_t} = 0 \quad , \quad {\Psi^u}_n = -\frac{\partial I}{\partial p} = 0 \quad \Longrightarrow \mathbf{\Psi^u} = 0 \tag{13}$$

$$\mathbf{n} \cdot \nabla \Psi^p = 0 \tag{14}$$

Wall:

$$\mathbf{\Psi^u_t} = 0 \quad , \quad \Psi^u_n = -\frac{\partial I}{\partial p} \quad \Longrightarrow \quad \mathbf{\Psi^u} = \Psi^u_n \cdot \mathbf{n} = -\big(w_d \frac{\partial c_d}{\partial p} - w_l \frac{\partial c_l}{\partial p}\big) \cdot \mathbf{n} \tag{15}$$

$$\mathbf{n} \cdot \nabla \Psi^p = 0 \tag{16}$$

Outlet:

$$0 = u_n \mathbf{\Psi^u_t} + \nu(\mathbf{n} \cdot \nabla)\mathbf{\Psi^u_t} + \frac{\partial I}{\partial \mathbf{u_t}} = u_n \mathbf{\Psi^u_t} + \nu(\mathbf{n} \cdot \nabla)\mathbf{\Psi^u_t} \tag{17}$$

$$\Psi^p = \mathbf{\Psi^u} \cdot \mathbf{u} + u\Psi^u_n + \nu(\mathbf{n} \cdot \nabla)\Psi^u_n + \frac{\partial I}{\partial u_n} = \mathbf{\Psi^u} \cdot \mathbf{u} + u\Psi^u_n + \nu(\mathbf{n} \cdot \nabla)\Psi^u_n \quad . \tag{18}$$

Note that for reasons of numerical stability the boundary condition for the adjoint inflow velocity $\mathbf{\Psi^u}$ in Equation (13) is set to zero gradient instead.

2.3 Gradient calculation

In order to calculate the new geometry as in Equation (4) the gradient of the Lagrangian is needed. Using the adjoint approach it can be calculated from boundary integrals only, referring to Soto and Löhner [5]:

$$\begin{aligned}\delta L =& \frac{\partial I}{\partial \boldsymbol{\beta}} \delta \boldsymbol{\beta} + \int_S (\Psi^p \mathbf{n})(\mathbf{n} \cdot \nabla \mathbf{u}) \; dS \\ &+ \int_S 2\nu \mathbf{n} \cdot \mathbf{D}(\mathbf{n} \cdot \nabla \mathbf{u}) \cdot \mathbf{\Psi^u} \; dS - \int_S 2\nu \Big(\mathbf{n} \cdot \mathbf{D}(\mathbf{\Psi^u})\Big)(\mathbf{n} \cdot \nabla \mathbf{u}) \; dS \quad ,\end{aligned} \tag{19}$$

where S is the airfoil boundary. Compared to the original paper of Soto and Löhner [5] the normal pressure gradient cancels due to the pressure boundary condition on the wall. Note, the sign of the first integral is positive corresponding to the notation of continuity in Equation (3). The geometric part of the gradient in Equation (19) can be calculated depending on the cost function as:

$$\frac{\partial I}{\partial \boldsymbol{\beta}} = w_d \frac{\partial c_d}{\partial \boldsymbol{\beta}} - w_l \frac{\partial c_l}{\partial \boldsymbol{\beta}} \quad . \tag{20}$$

This is straightforward in case the design parameters are the positions of the boundary faces centres.

2.4 Constrained cross-section area

In principle, the cross-section area A is calculated as the integral over the whole plane area:

$$A = \int_A dA \quad . \tag{21}$$

As shown in Figure 1, A is not meshed in two-dimensional cases and we propose to use the divergence or Gauss's theorem for calculating the area. Löhner [8] used this approach to calculate a meshed volume and we adopt this technique in order to compute the cross-section area A. The area of an extruded geometry, as it is the case in two-dimensional

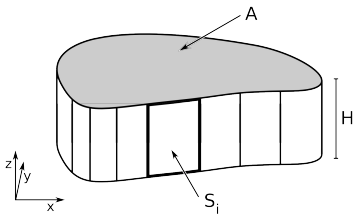


Figure 1: Cross-section area of a two-dimensional mesh

meshes, can be calculated by its volume V divided by the height H in extrusion direction. Applying the divergence theorem to the volume gives ($\partial V = S$):

$$A = \frac{1}{H} V = \frac{1}{H} \int_V \nabla \cdot \mathbf{a} \; dV = \frac{1}{H} \int_S \mathbf{a} \cdot \mathbf{n} \; dS \quad , \tag{22}$$

which only holds when $\nabla \cdot \mathbf{a} = 1$. For instance, the vector $\mathbf{a}$ can be obtained by $\mathbf{a} = {}^1\!/_2 \cdot (x, y, 0)$ and the cross-section area is:

$$A = \frac{1}{2\ H} \int_S (x, y, 0) \cdot \mathbf{n} \; dS \quad . \tag{23}$$

Although dS covers a three-dimensional body, the elements normal to the z-direction cancel by the dot product. Writing Equation (23) numerically lead to:

$$A = \frac{1}{2\ H} \sum_i (x, y, 0)_i \cdot \mathbf{n}_i \; S_i = \sum_i A_i \quad , \tag{24}$$

from which the area elements A_i and also their derivatives can be calculated. Note that the area A is not meshed and thus A_i is not explicitly known, but can be computed by the relation in Equation (24).

In order to handle geometric constraints the conditions of Karush-Kuhn-Tucker (KKT) can be employed [9], which means that violation terms are added to the Lagrangian of Equation (6). In contrast to the Navier-Stokes equations $\mathbf{R}$, those terms are embedded as inequality constraints:

$$L := I \; + \; \int_{\Omega} (\mathbf{\Psi}^u, \Psi^p) \cdot \boldsymbol{R} \; d\Omega \; + \; \lambda_{max} \frac{(A_{max} - A)}{A^0} \; + \; \lambda_{min} \frac{(A - A_{min})}{A^0} \quad , \tag{25}$$

where λ_{max} and λ_{min} are positive KKT multipliers for the constraints A_{max} and A_{min}, respectively. The additional terms are normalized with the initial area A^0 and are only active, when one of the constraints is violated. Then, a constant cross-section area can be gained by choosing: $A_{max} - A_{min} \to 0$.
Note, the area constraints are added to the Lagrangian and not to the cost function, thus the boundary conditions are not changed[1].

3 ADJOINT SOLVER IN OPENFOAM

The CFD code used in this work is OpenFOAM® [10], which is open source and based on C++. The code can be extended by each user and thereby it is convenient for creating new solvers.

In OpenFOAM, an adjoint solver for internal flows already exists [11], but for external aerodynamics the boundary conditions have to be changed as shown in the previous section. Furthermore, a mesh movement has to be used, which is done with a vertex based method of the extended version of OpenFOAM. Then each grid point of the airfoil shape can be moved independently, which leads to a high flexibility. Although the mesh is moved pointwise, the gradient is calculated on the boundary faces only and an interpolation from faces to points is applied. The solver is based on the Reynolds-Averaged-Navier-Stokes equations (RANS) for incompressible flows and runs in parallel to achieve results faster.

In order to obtain a smooth surface the gradient is averaged and normalized. As in the adjoint solver for internal flows the one-shot approach is used, which means that flow and adjoint variables are solved in the same time step [12]. This may lead to a faster optimization procedure, but still a defined convergence of the flow field has to be achieved before the next optimization step is executed.

The optimization is done via a steepest descent method as in Equation (4). Whenever the current cost function is smaller than its value of the last optimization step, the step size is reduced and the sign of the gradient is changed. The process is stopped, when the relative change of the cost function becomes sufficiently small.

[1] Anyway, the derivative of a geometric constraint w.r.t. the pressure, as it is necessary for the boundary conditions in Equations (13) to (18), is zero.

4 RESULTS

Basis of the optimization in this work is the profile DU 91-W2-250 developed by Delft University [13], which has a maximum thickness of 25%. It was specially designed for wind energy purposes and is used by various manufacturers.
The simulations use wall functions with a dimensionless wall distance of $y^+ \approx 100$ and turbulence is modelled applying the k-ω-SST model. The Reynolds number is 3 million and the computational grid consists of approx. 55,000 cells, where 360 are on the airfoil. This means that 720 design parameters are used in the optimization process.
For completeness, it must be said that the aim of this work is to show the functionality of the optimization process and not a validation of the used CFD code, which means that aerodynamic coefficients might differ from experimental values[2].

4.1 Non-constant cross-section area

The optimization of lift over drag as in Equation (11) is done for three different angles of attack (AoA), where experimental data is available: 6.15°, 8.18°, 9.66°.

For an angle of 6.15°, the evolution of the objective c_l/c_d is plotted against the solver iteration steps in Figure 2.

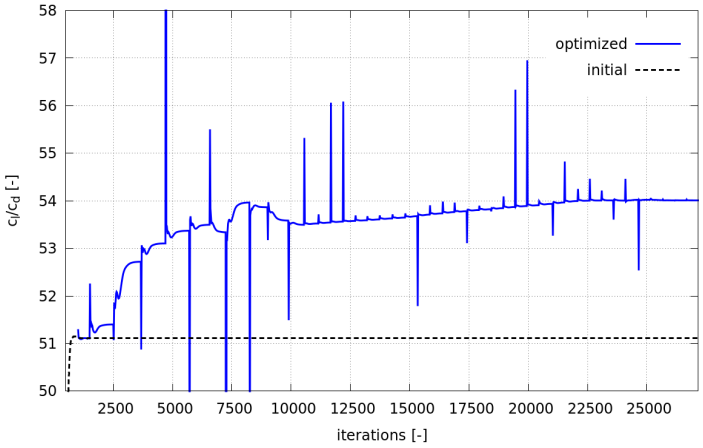


Figure 2: Lift over drag evolution at $AoA = 6.15°$

The graph shows a strong increase in the beginning of the process and becomes smoother after 10,000 iterations, which also occurs due to a decreasing step size. The oscillations of the "optimized" ratio result from a relatively big mesh movement and accordingly several iterations are needed for convergence. Although the purpose is to increase the lift to drag ratio continuously, a decrease at 8,000 iterations can be seen. This might result from a change of sign in the gradient described in the previous section,

[2]Since a fully turbulent flow is modelled the numerical results are in better agreement to the experiments performed with zigzag tapes.

because this change is done globally, which means that it can lead to an improvement at one position, but could lead to an impairment at some other position. Another reason for the decrease in Figure 2 could be that the step size in this point is still too big.
Compared to the initial airfoil, the ratio of lift over drag is increased by 5.7%, which is the result of a drag reduction of 4.9% and a lift increase of 0.5%.

The evolutions of the lift over drag ratio for an angle of attack of 8.18° and 9.66° are plotted in Figure 3.

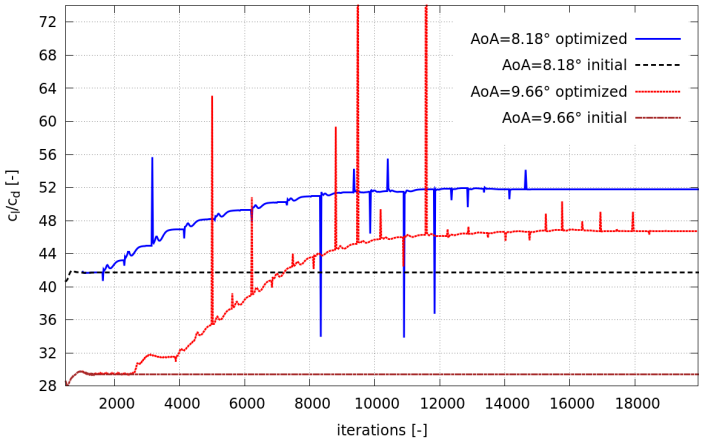


Figure 3: Lift over drag evolution at $AoA = 8.18°$ and $AoA = 9.66°$

In principle, they show a similar behaviour as in the case of 6.15°, although the improvements are bigger [3]. At an angle of attack of 8.18°, the lift to drag ratio increases by 24%, resulting from a drag reduction of 15.5% and an increase in lift by 4.9%. The coefficient changes are higher compared to the previous case with an angle of attack of 6.15°.

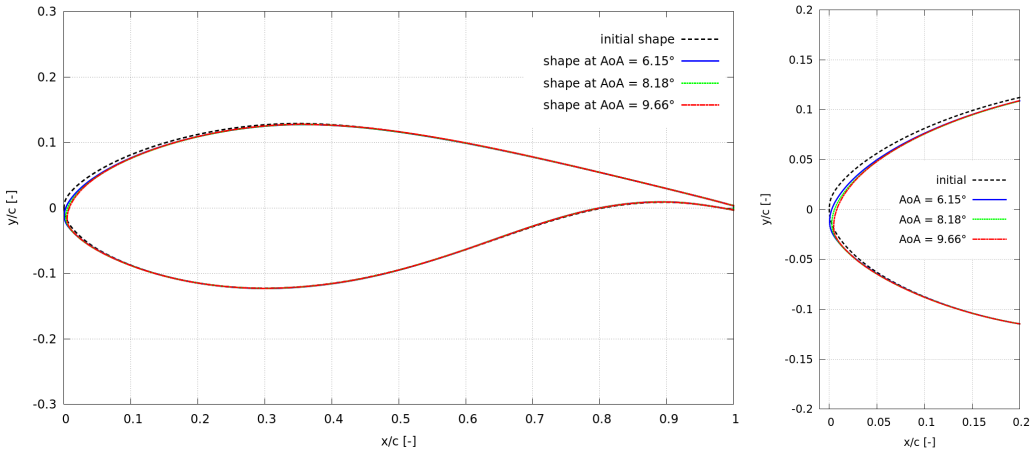
At an angle of attack of 9.66°, no difference to the initial airfoil can be seen in the very beginning of the optimization (until 2,500 iterations), because the initial solution might not be fully converged. After 2,500 iterations the lift to drag ratio increases steeply and becomes smoother, when convergence is reached. The lift to drag ratio increases by 59%, resulting from an increase in lift of 12% and a decrease in drag of 29%.
The changes in aerodynamic coefficients are again higher than in the previous cases. This might be because the maximum lift to drag ratio of the DU 91-W2-250 is at 6.7° and the first angle of attack is closer to this point [13]. From this it follows that there is a higher potential in the optimization the further the angle of attack is away from 6.7°.

For completeness and a better overview the resulting change of coefficients are listed in Table 1 and finally Figure 4 shows the optimized shapes of the described cases compared to the initial airfoil shape.

[3] Although in case of 8.18° the optimization converges and stops at approx. 15,000 iterations, the lines are extended for a better comparison.

Table 1: Change of aerodynamic coefficients

AoA	$\Delta c_l/c_d$	Δc_l	Δc_d
6.15°	5.7%	4.9%	-0.5%
8.18°	24%	4.9%	-15.5%
9.66°	59%	12%	-29%



(a) (b)

Figure 4: Shapes of optimized and initial airfoils (complete and detail)

It can be seen that the leading edge is deformed whereas there are no significant changes at the trailing edge. This is not very surprising, because the shape of the leading edge has a high sensitivity to lift and drag. Furthermore, the changes are bigger with increasing angle of attack, which corresponds with the higher change of lift to drag ratios in Figures 2 to 3.

An interpolation of these optimized shapes could be a first step to a new airfoil. However, this goes beyond the scope of this work and is not done. Also, it must be noted that the higher efficiency at the examined angles of attack might lead to a lower efficiency or poorer aerodynamic behaviour at other angles.

4.2 Constant cross-section area

The geometric constraint in this work is the cross-section area, which is used in order to obtain a shape considering structural stability and manufacturing. To show the working principle, the area is held constant by applying two inequality constraints in the Lagrangian function (25).

For an angle of attack of 8.18°, the area evolution is plotted against the optimization

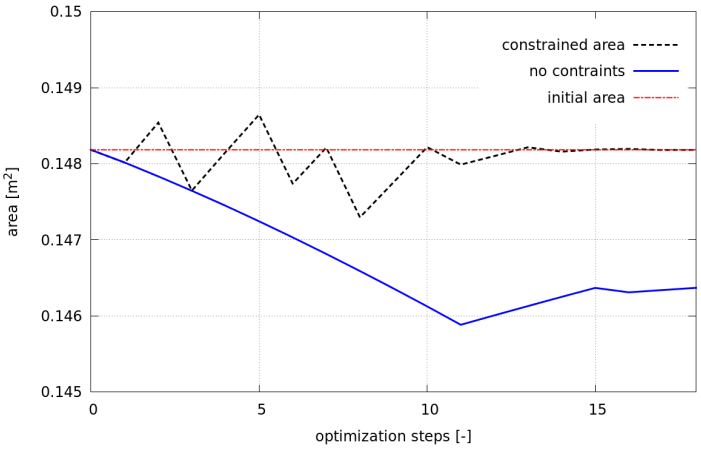


Figure 5: Cross-section area at $AoA = 8.18°$

steps in Figure 5 and compared with the non-constant cross-section area from the previous section. It can be seen that the area in the unconstrained case goes down, where it converges to a certain value. The area in the constrained case varies around the initial value and stays constant after some iterations. The fluctuations in the beginning become smaller after some optimization steps, which could also result from smaller mesh movements in general. The cross-section area in the unconstrained case decreases by 1.2%, but the improvement of the lift to drag ratio is nearly 6.5 percentage points higher than in the constrained case. Compared to the loss in improvement of lift over drag, the achievement of a constant cross-section area might be of less importance.

This is an exemplary case only and the results will be different for other airfoils or configurations. Nevertheless, the working principle of a constrained optimization is shown and can be applied for other geometric constraints such as minimum thickness etc.

4.3 Approximate gradients

As Medic et al. [14] already wrote "the dominant part in the gradient comes from the partial derivative with respect to the shape and not to the state". This may not be valid in general, but it is for many cost functions, which are defined on the deformed boundary wall as it is the case in this work. Then the gradient simplifies and the integrals in Equation (19) are cancelled. Thereby the computation of adjoint variables is not necessary and the speed of the optimization is more than two times higher[4].

For instance, Figure 6 shows the evolution of the lift to drag ratio comparing approximate and complete gradients for an angle of attack of 8.18° without geometric constraints. The improvement is less than 1 percentage point lower than using complete gradients, but the optimization is approx. 2.5 times faster. The differences in the resulting shapes are

[4]A doubled speed might be expectable, but for stability reasons the adjoint velocity has a smaller relaxation factor than the primal velocity, which leads to a relatively slower computation.

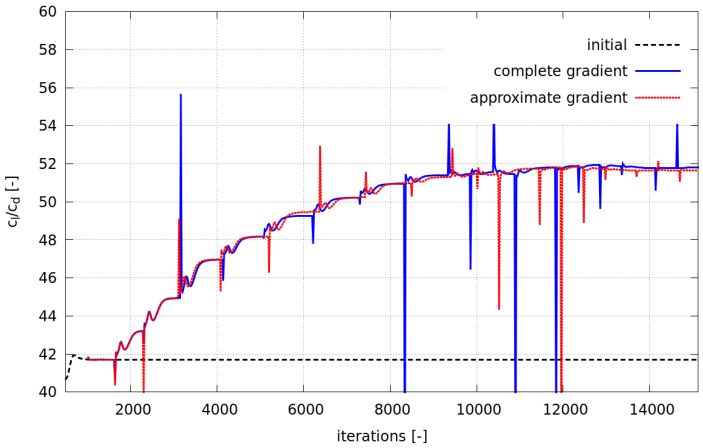


Figure 6: Lift over drag evolution by approximate and complete gradients at $AoA = 8.18°$

equally small, which is why a comparison is not explicitly shown.

The results might have a bigger difference, when the angle of attack rises into the stall region, but then also transient effects become more important, which makes an optimization more challenging.

5 CONCLUSIONS

A gradient-based optimization of a two-dimensional airfoil has been done by the use of the adjoint approach. It was shown that a bigger distance to the initial maximum of lift to drag ratio, which is at 6.7°, lead to bigger changes in the aerodynamic properties (up to 59% at 9.66°). Accordingly, the changes of the resulting shapes are bigger, e.g. the cross-section area becomes approx. 1.2% smaller at 9.66°.

The optimization was combined with two inequality constraints in order to achieve a constant cross-section area. The geometric constrained could be fulfilled, but with the drawback of a less improved cost function. In the shown case of 8.18°, the lift to drag ratio was increased by nearly 6.5 percentage points less compared to the unconstrained optimization, which might be important enough to disregard the area change (the area decreased by 1.2% in the unconstrained case).

The effect of approximate gradients was examined and the improvements of the objective function were similar to the optimization using complete gradients. When comparing the increase of the cost function to the computational costs, at least in the shown case of 8.18°, it is not necessary to compute the adjoint field. This might be different in other cases, e.g. other cost functions, airfoils or angles of attack.

REFERENCES

[1] Dahl, K. S., Fuglsang, P. Design of the Wind Turbine Airfoil Family RISØ-A-XX. *Risø-R-1024*, Risø National Laboratory, Roskilde, (1998).

[2] Othmer, C. A continuous adjoint formulation for the computation of topoligical and surface sensitivities of ducted flows. *International Journal for Numerical Methods in Fluids*, Wiley InterScience, (2008).

[3] Thévenin, D. and Janiga, G. *Optimization and Computational Fluid Dynamics*. Springer, Berlin, Heidelberg, (2008).

[4] Nocedal, J. and Wright, St. J. *Numerical Optimization*. 2nd edition, Springer, (2008).

[5] Soto, O. and Löhner, R. On the Computation of Flow Sensitivities from Boundary Integrals. 42nd *AIAA Aerospace Sciences Meeting and Exhibit*, (2004).

[6] Zymaris, A. S., Papadimitriou, D. I., Giannakoglou, K. C. and C. Othmer, C. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, Elsevier, (2010).

[7] Dwight, R. P. and Brezillon, J. Effect of Various Approximations of the Discrete Adjoint on Gradient-Based Optimization. 44th *AIAA Aerospace Sciences Meeting and Exhibit*, Reno, (2006).

[8] Löhner, R. *Applied CFD Techniques - An Introduction based on Finite Element Methods*. 2nd edition, John Wiley & Sons, (2008).

[9] Arora, J. S. *Introduction to Optimum Design*. 2nd edition, Elsevier Academic Press, (2004).

[10] Openfoam finite volume programming environment for CFD. *www.openfoam.com*.

[11] Othmer, C., de Villiers, E. and Weller, H. G. Implementation of a continuous adjoint for topology optimization of ducted flows. 18th *AIAA Computational Fluid Dynamics Conference*, Miami, (2007).

[12] Ta'asan, S., Kuruvila, G. and M. D. Salas, M. D. Aerodynamic design and optimization in one shot. *AIAA-92-0025*, (1992).

[13] Timmer, W. A. and van Rooji, R. P. J. O. M. Summary of the Delft University Wind Turbine dedicated Airfoils. *AIAA-2003-0352*, (2003).

[14] Medic, G., Mohammadi, B., Moreau, S. and Stanciu, M. Optimal Airfoil and Blade Design in compressible and incompressible Flows. *AIAA-98-2898*, (1998).