# WING OPTIMIZATION CONSTRAINED BY BOUNDARY LAYER SEPARATION USING NONLINEAR LIFTING LINE METHOD

J. HÁJEK
Aeronautical Research and Test Institute, Beranových 130, Prague, Czech Republic

**Summary**

An extension to the classical lifting line method based on Prandtl's theory is described. The extended method allows employing nonlinear section lift data, giving more realistic results at higher angles of attack. Amongst other quantities, the method is able to estimate the point where separation of boundary layer first occurs. We employ the method in an optimization problem constrained by the position of the separation point on the wing and present the results.

# 1 Introduction

The lifting line theory invented by Prandtl is classical ([1]). In this article, we present an extension of this method designed to exploit nonlinear 2D wing section lift data, supplied by sophisticated 2D solvers such as XFOIL ([2]). This extension seems to be a promising tool for obtaining estimates of various very useful characteristics of 3D configurations several orders of magnitude faster than possible using CFD solvers. The article is split into sections as follows. In Section 2, we give the governing mathematical equations that form the base of the method, describe the numerical procedure used for solving these equations, describe our open source implementation and show example results. Section 3 discusses the applicability of the results of our method to real world problems and comparisons with wind tunnel experiments. Section 4 shows an application of our method to wing design optimization.

# 2 Method description

### 2.1 Mathematical formulation

The method starts from the assumption that the flow around the wing can be modeled by a system of horseshoe vortices, as shown on Fig. 1. The bound segments of these vortices model the circulation around the wing, while the free segments form the wake. Given a single horseshoe vortex with known strength (circulation), its induced velocity in any point in the 3D space can be calculated using the Biot-Savart law:

$$d\vec{v}=\frac{\Gamma}{\varepsilon\,\pi}\;\vec{r}\times d\vec{s} \qquad (1)$$

This formula gives the induced velocity of an infinitesimal vortex segment. The induced velocity of a vortex thread is obtained by integrating along the thread. The velocities induced by a system of vortex threads (the horseshoe vortex system) are obtained by superposition.

Given the vortex strengths, we can, using the above method, calculate induced velocities in the midpoints of the bound vortex segments (the influence of a bound segment on itself is set to zero).

$$\vec{v}_i^x = \sum g_{ik}^x \Gamma_k \qquad (2)$$
$$\vec{v}_i^y = \sum g_{ik}^y \Gamma_k$$

These induced velocities are superimposed onto the freestream velocity to get local stream velocity and local angle of attack (in the following equations, we omit the subscript *i* for brevity)

$$v^{loc} = \sqrt{(\vec{v}^x + \vec{v}_\infty^y)^2 + (\vec{v}^y + \vec{v}_\infty^y)^2} \qquad (3)$$
$$\alpha^{loc} = \arctan\left(\frac{\vec{v}^x + \vec{v}_\infty^x}{\vec{v}^x + \vec{v}_\infty^y}\right) - \alpha^{twist}$$

The local lift coefficient is interpolated from the local polars:

$$c_L^{loc} = C_L^{loc}(\alpha^{loc}) \qquad (4)$$

and the equations are closed using the Kutta-Joukowski law (*loc* superscripts omitted):

$$c_L v c = 2\Gamma \qquad (5)$$

Choosing the local circulations as unknown variables, we end with a system of nonlinear equations of the form

$$\Gamma_i = F_i\left(\sum g_{ik}^x \Gamma_k, \sum g_{ik}^y \Gamma_k\right) \qquad (6)$$

where $F_j$ are scalar nonlinear functions of two variables. These functions depend not only on the wing geometry and the section polars, but also on the global angle of attack.

### 2.2 Solution method

As described above, the problem of determining the circulations for a given configuration can be transformed into a set of nonlinear equations dependent on the global angle of attack (the wing geometry and section polars as fixed throughout a single computation). In vector notation (bold symbols denote vector variables and vector functions), we can write it as

$$\boldsymbol{F}(\boldsymbol{\Gamma}(\alpha), \alpha) = \boldsymbol{0} \qquad (7)$$

The computation proceeds by continuously tracking $\Gamma(\alpha)$ using a *predictor-corrector* paradigm. At any angle of attack $\alpha$, we choose a step $\Delta\alpha$, and use approximate differentiation of the previous equation to obtain the *predictor* in the form:

$$\nabla \boldsymbol{F}(\boldsymbol{\Gamma}, \alpha) \cdot \Delta \boldsymbol{\Gamma} = -\boldsymbol{F}(\boldsymbol{\Gamma}, \alpha + \Delta\alpha) \qquad (8)$$

with $\Delta\Gamma$ unknown. Afterwards, we replace $\boldsymbol{\Gamma}+\Delta\boldsymbol{\Gamma}\rightarrow\boldsymbol{\Gamma}$, $\alpha+\Delta\alpha\rightarrow\alpha$ and use the new value of $\Gamma$ as a starting point for the *corrector* applied to the system of nonlinear equations (7). The corrector is the Levenberg-Marquardt method for solving nonlinear least squares problems. That is, the system (7) is solved in a least squares sense for better robustness.

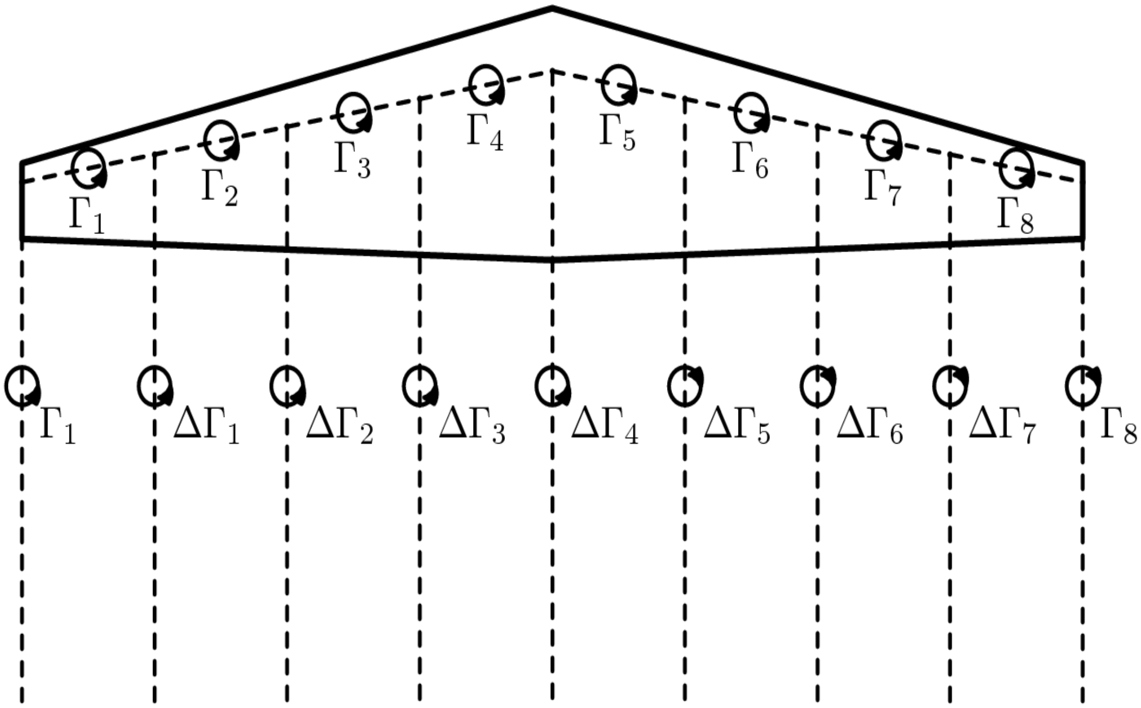


*Figure 1: The horsheshoe vortex model of a wing*

### 2.3 Implementation description

The above method has been implemented as a package for the Octave system ([3]), an open source “clone” of Matlab. The package is called NLWing2 and it is part of the OctaveForge project, that comprises many packages for Octave ([4]). The wing geometry is specified directly using 3D coordinates of the quarter-point line, along with local depth and twist distribution. As a consequence, curved and swept wings can be specified, as well as wings with dihedral angle (not necessarily constant), and they can be of arbitrary planform. It should be noted, however, that the horseshoe model may not be appropriate for wings with excessive sweep and dihedral angles, and thus the method may fail to yield sensible results in such cases. The package implements the predictor-corrector approach as described above, choosing step adaptively. A number of quantities are computed using the known span-wise local angle of attack distribution: the lift, viscous drag, induced drag & momentum coefficient of the wing (using local viscous drag and momentum interpolated from local polars). Local span-wise quantities, including local angle of attack, are also obtained.

Fig. 2 shows a comparison between the nonlinear lift curve of a wing produced by NLWing2 and a linear curve produced by solving the linear Prandtl's equation by Glauert's method.

Fig. 3 compares the span-wise lift distributions.

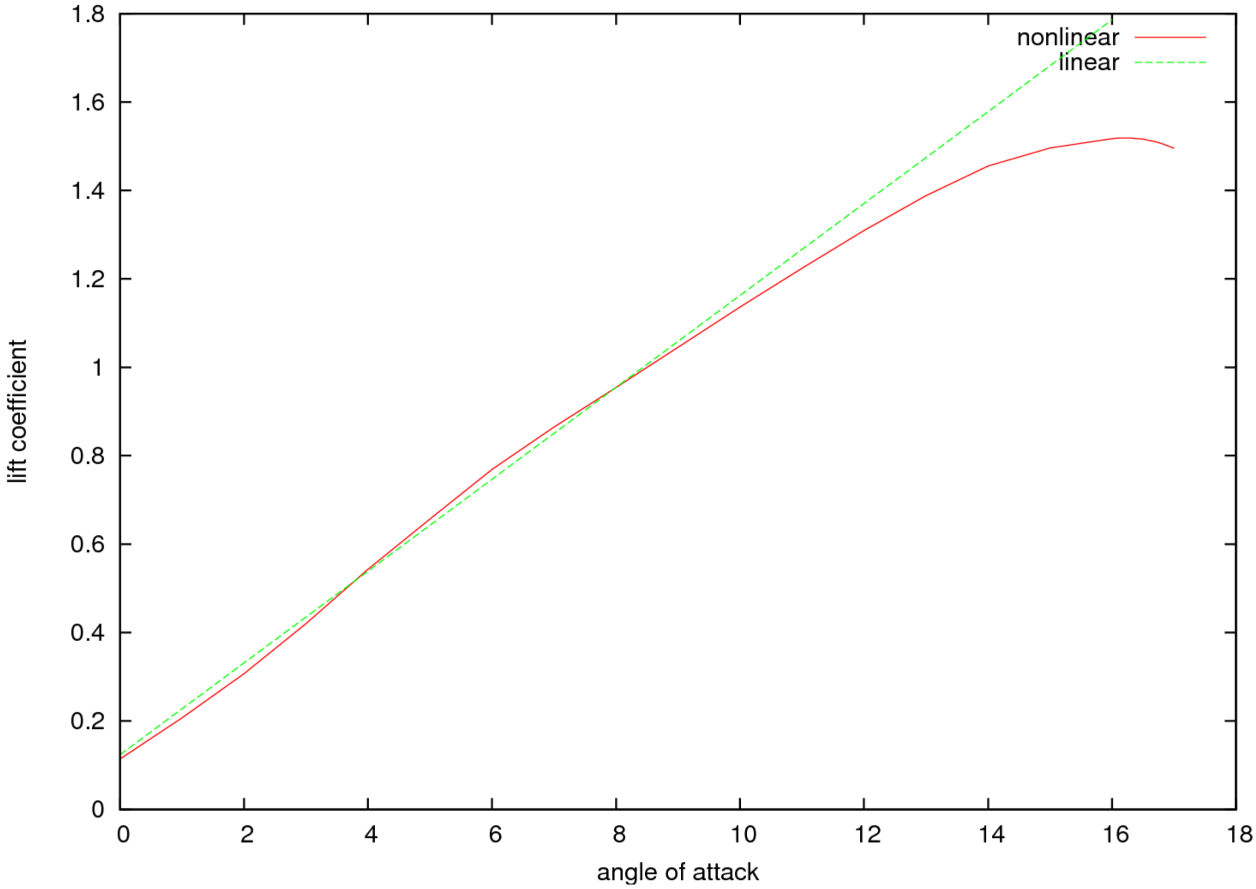

*Figure 2: Comparison of lift curves of a wing, linear vs. nonlinear method.*

# 3 Interpretation and verification of results

## 3.1 Behaviour at high angles of attack

Numerical problems always start occurring when local lifts approach local maximum lifts. The problem stems from the fact that in such areas, the lift does not uniquely determine the angle of attack. This causes spurious oscillations in the local angle of attack distribution, leading the process “the wrong way”. It does not seem possible to solve this problem within the current framework – no matter what we choose for control variables. As long as the equations are closed via the lift coefficient, the problem always occurs. We are not yet sure whether or not some kind of stabilization would help solve this issue. Certain experiments suggested that, rather than being a purely numerical issue, the oscillations may even be an inherent property of the governing nonlinear equations. That means, it may even be possible that there is a unique oscillating solution. That would, of course, mean that the model simply ceases to be feasible. Resolving this issue will be a subject of further research.

Nevertheless, for most wings it is possible to reach the point where the local angle of attack at a certain point along the span reaches the angle of attack of the maximum local lift, or even surpasses it by up to 0.5-1 degrees, before the solution deteriorates.

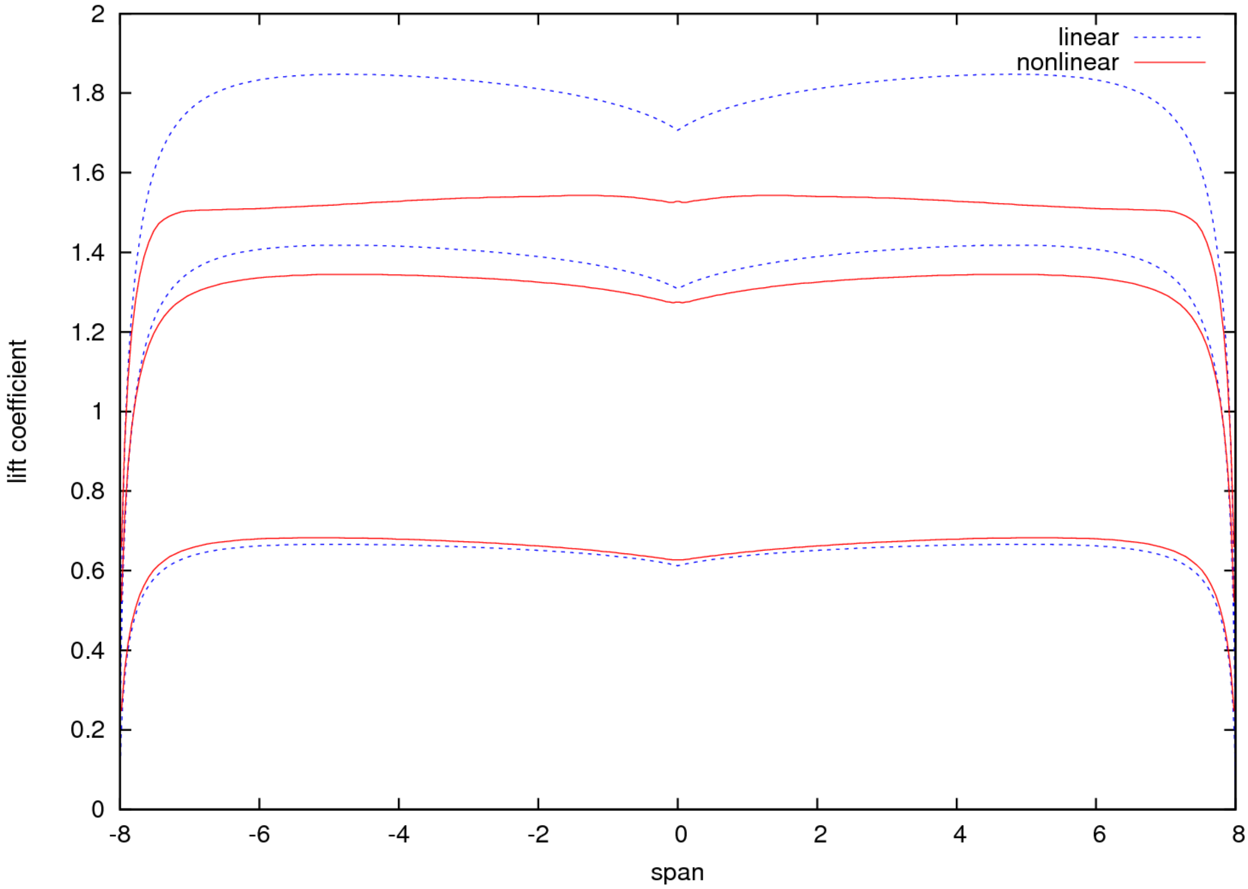

*Figure 3: Comparison of span-wise lift distributions, linear vs. nonlinear method.*

### 3.2 Boundary layer separation and stall of lift

Since we associate the decline of local lift (of a 2D wing section) with the separation of boundary layer, our method gives us a means to estimate the span-wise point where, with increasing angle of attack, the separation starts to appear (and it spreads from there).

This information is very valuable for flight stability and safety. The point is to keep the separation away from ailerons; otherwise, the pilot loses control over the aircraft.

A detailed investigation of accuracy of these estimates is a subject of current research. Preliminary comparisons with wind tunnel experiments described in the next section below show that the accuracy may be within 5% of the span. Given that our method is able to deliver the result within seconds, whereas CFD computations usually take hours (even days) to get a comparable result (several angles of attack need to be computed), verifying this accuracy (at least for certain classes of wings) would make our method a valuable tool for estimating this property of the wing.

### 3.3 Wind tunnel experiment

An experiment in the 3mLSWT wind tunnel in VZLU was carried out. The experiment was focused to show the point where the separation on the wing starts. A model of twin-engine turboprop with high-wing was used. For the purpose of comparison with the numerical computation, the engine nacelles were put off of the wing. The upper side of the wing was equipped with tufts for the flow visualization. The tests were performed at Reynold number of

700000 and Mach number of 0.15. Results showed, that the separation starts at 25-30% of the wing span, what was also predicted by the calculation.

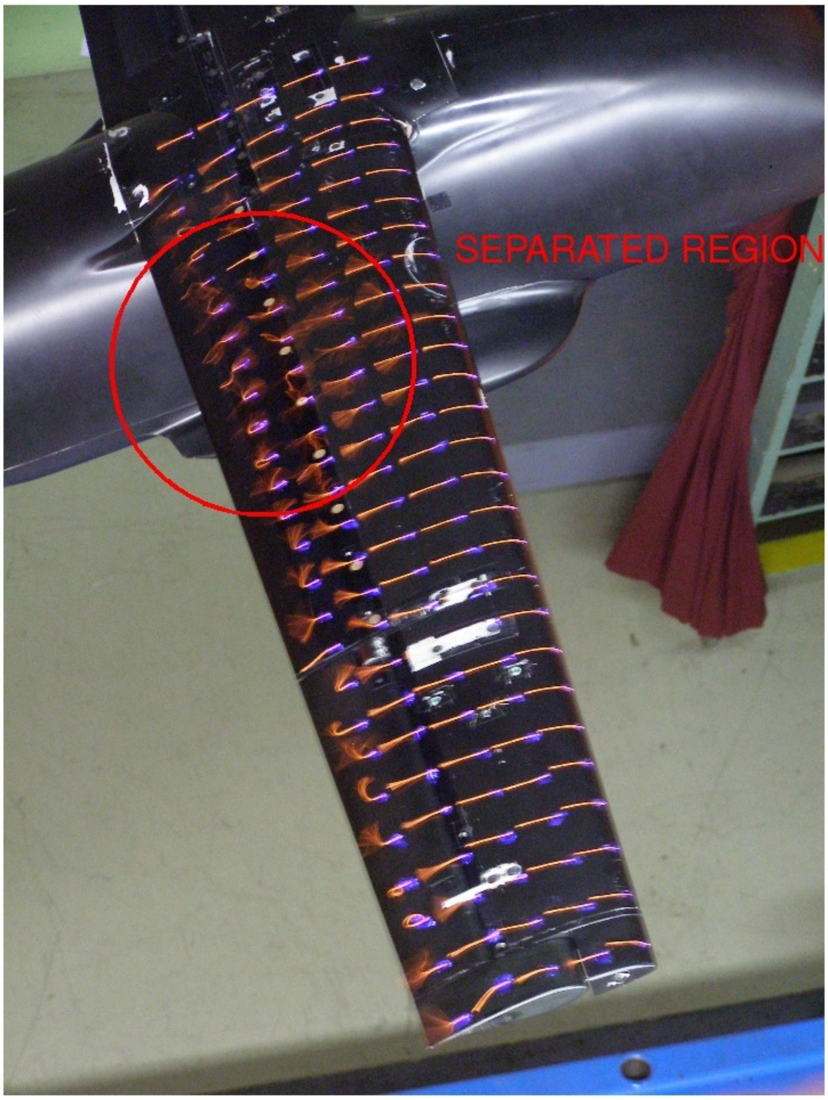


*Figure 4: Picture of tunnel measurements*

# 4 Application to wing optimization

## 4.1 Design variables and constraints

We used our method to solve a particular wing design optimization problem, carried out as part of the CESAR project at VZLÚ. The goal was to optimize the planform of a symmetric wing, consisting of two (for each half) trapezoidal sections. The geometry was to be determined by the following five design variables:

- Sw: The wing planform area 23 - 27 m$^2$
- TR: Taper ratio (tip chord / root chord) 0.4 – 0.6
- ttw: Tip twist 0 - 4°
- xbr: Relative break position 0 – 0.5
- btr: Deviation from trapezoidal wing: -1 – 1

The last two variables deserve a more precise description. Say that the root chord length is $c_{rw}$ , $c_{br}$ is the chord length separating the root (inner) and tip (outer) trapezoid, and $c_{tw}$ is the tip chord length. Further, let $b_r$ and $b_t$ be the root and tip trapezoid span, respectively. Then

$$xbr=\frac{b_r}{b_r+b_t} \tag{9}$$

and

$$btr=\frac{c_{br}-c_{b0}}{c_{rw}-c_{b0}}, \quad c_{b0}=xbr\,c_{tw}+(1-xbr)c_{rw}. \tag{10}$$

Further, $c_{rw}=\text{١.٩٥٥}\,m$ , dihedral angle is 1.5°, and the quarter-chord line is assumed straight. These constraints determine the geometry of the wing.

## 4.2 Objectives and constraints

The optimization was driven by the following objectives:

- minimize $C_D S_W$ at $C_L S_w=7.558$
- minimize $-C_M S_W C_{MAC}$ at $C_L S_w=\text{٧.٥٥٨}$

and the primary constraint

- $Z_{SEP}/(b/\text{٢})\leq 0.6$

Where $Z_{SEP}$ is the span-wise coordinate where the local lift coefficient first reaches its maximum (critical point). These objectives were earlier used in optimizing a wing formed by a single trapezoid (btr = 0), used as a preliminary problem, and worked well. It was, however, later discovered that in the more general design space, these objectives do not lead to a compromise surface but rather produce a single Pareto optimum. The reason is that in the single-trapezoid case, the primary mechanism to shift $Z_{SEP}$ towards the root is to increase the tip twist. Unfixing btr, however, can form a small downwards-oriented spike in the span-wise lift distribution and thus introduces a completely different mechanism to satisfy the primary constraint, by shifting this spike towards the root, as indicated in Fig. 5. This obviously comes at the cost of lowering the critical angle of attack, which was not reflected by our original objectives. We therefore added the following:

- maximize $C_L S_W$ at $\alpha_{CRIT}$

This turned our problem into a 3-objective one with a non-degenerate Pareto front (compromise surface), as shown in the following section.

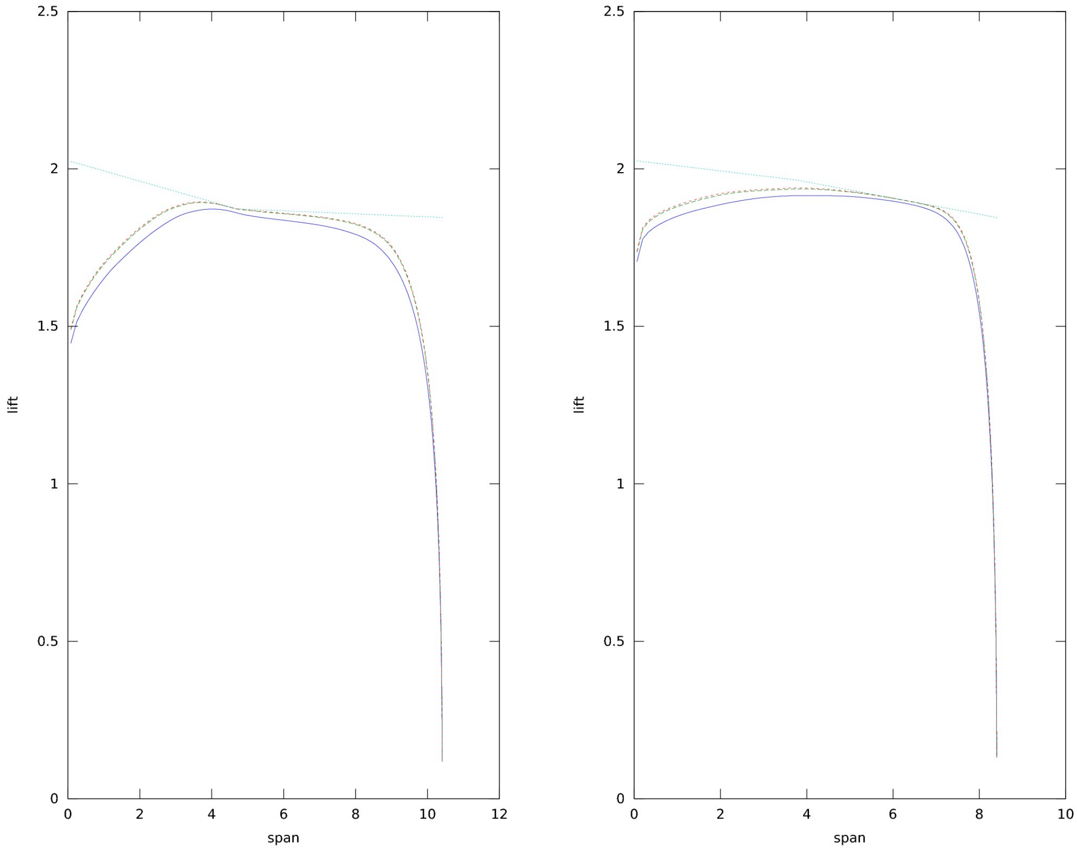


*Figure 5: The effect of chord length spike on the point of separation*

### 4.3 Results

Selected candidate designs were evaluated for objectives and constraints using NLWing2. Together with the recently developed parallel evaluation capabilities for Octave (`parcellfun` from OctaveForge/general), using 8 AMD Opteron processors (@2.4GHz), evaluation of $5^5$ = 3125 designs takes just 21 min 9s, giving an average of 0.4s per design (3.2s on a single CPU). It should be also noted that to evaluate the objectives, NLWing2 evaluates the wing performance at a sequence of angles of attack (operating points), in our case usually between 40 and 50.

The outstanding performance of our design evaluation procedure gives us the option to perform exhaustive sampling of the design space as an alternative to genuine optimization algorithms, by simply evaluating the objectives at a grid of designs and just picking out the optimal solutions. This can be thought of as a very simplistic optimization algorithm. Despite its simplicity, it has the advantage of being very robust and essentially fail-safe; it is guaranteed to find any sufficiently broad local optimum.
We also solved the same problem by employing a true multi-objective optimization algorithm the εμ-ARMOGA algorithm, described in [5].

Fig. 6 shows a result of sampling the 5-dimensional space using a Cartesian grid with 5 sample points in each dimension. All designs are shown as red, the Pareto-optimal (non-dominated) results are highlighted as blue.

The point in the left-lower-back corner corresponds to the unique Pareto optimum if only the first two objectives are considered, as described in the previous section.

Fig. 7 compares the Pareto fronts  achieved from grid sampling and by using the genetic algorithm. Since the sampling was rather coarse, the results from the genetic algorithm are significantly better. It is also more clearly visible how the Pareto front surface deteriorates to a spike when the third objective is low (again left-lower-back corner).

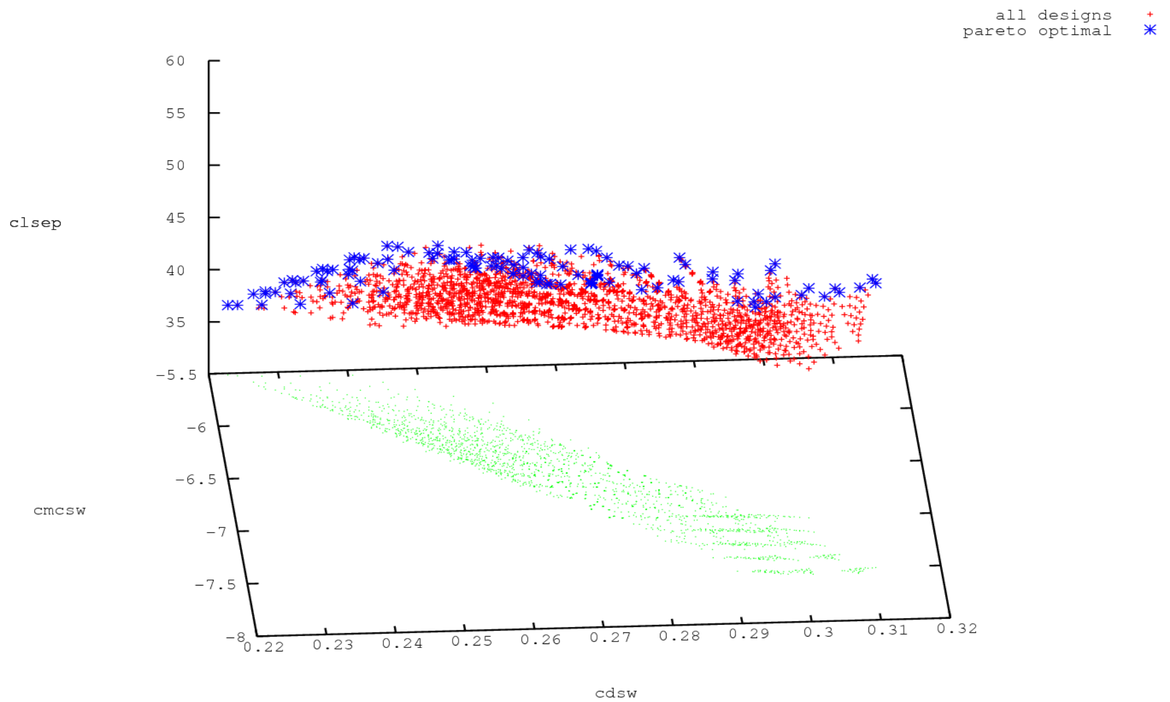


*Figure 6: Sampled design objectives with nondominated designs selected*

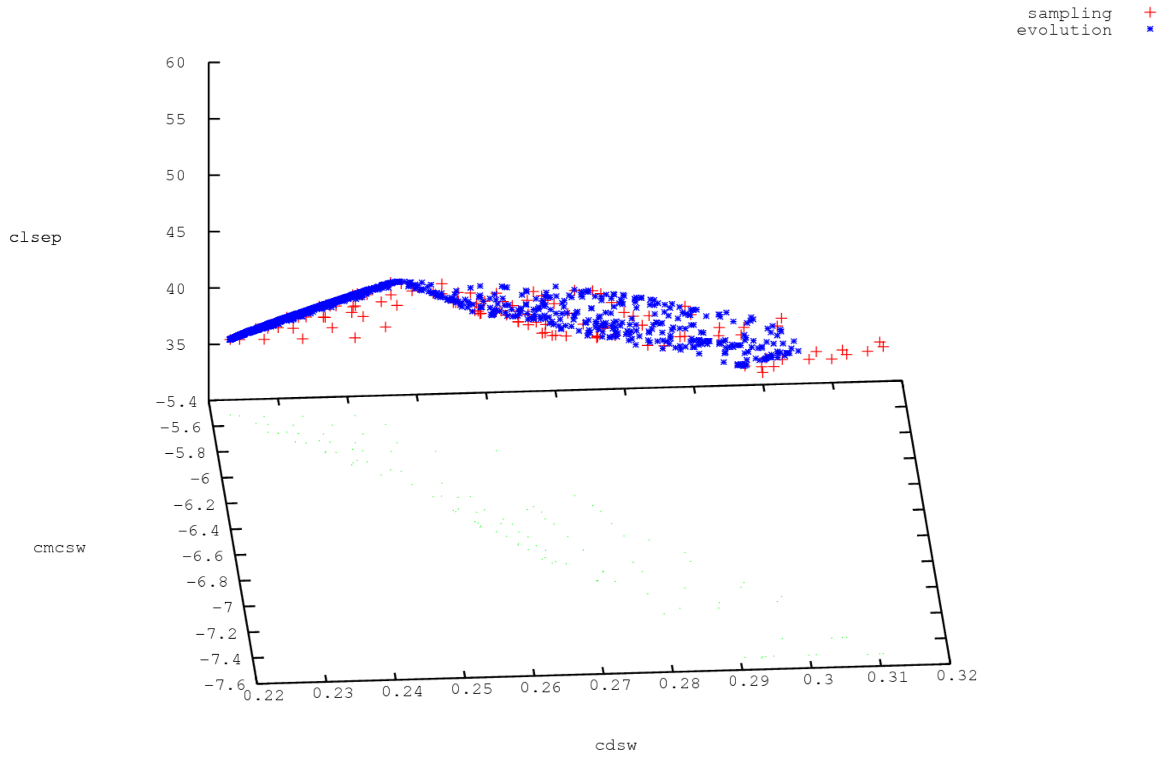


*Figure 7: Comparison of results from sampling and genetic algorithm*

# 5 Conclusion

We have shown a nonlinear extension of the classical lifting line method for computation of aerodynamic wing performance, and the capabilities of our implementation, especially the vastly superior performance compared to CFD approaches. We have pointed out the areas that are subject to ongoing research and demonstrated an application of the method to design optimization of a wing planform, constrained by conditions at the critical angle of attack. We have also shown how the choice of objectives significantly impacts the nature of the problem and its solution.

# Acknowledgment

The presented research was conducted in the frame of MSM 00010669 Rozvoj vnější aerodynamiky (Development of external aerodynamics) project granted by Ministerstvo školství, mládeže a tělovýchovy (Ministry of Education, Youth and Sports of the Czech Republic)..

# References

[1] Katz, J., Plotkin, A., Low-Speed Aerodynamics, Cambridge University Press, 2001, ISBN 0521665523

[2] XFOIL, http://web.mit.edu/drela/Public/web/xfoil/

[3] Octave, http://www.octave.org

[4] OctaveForge, http://octave.sourceforge.net

[5] Hájek,J., Szöllös, A., Šmíd, M.: Aerodynamic Optimization via Multi-Objective Micro-genetic Algorithm with Range-Adaptation, Knowledge-Based Reinitialization, ε-Dominance and Crowding, Advances in Engineering Software, 2008