The Eighth International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, Sichuan, China, July 14–18, 2014

ICCFD8-2014-0203

Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization

Zhoujie Lyu[1]*, Zelu Xu[1] and Joaquim R. R. A. Martins[1]

[1] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA

Abstract: Aerodynamic design optimization requires large computational resources, since each design evaluation requires the solution of a system of partial differential equations in a three dimensional domain. Thus, the choice of optimization algorithm is critical, as it directly affects the number of required design evaluations to reach the optimum design. To help designers make an informed choice, we benchmark several optimization algorithms, including gradient-based and gradient-free methods using three test problems of increasing difficulty: a multi-dimensional Rosenbrock function, a RANS-based aerodynamic twist optimization problem and an aerodynamic shape optimization problem. The majority of the gradient-based optimizers successfully solved all three test problems, while the gradient-free methods require two to three orders of magnitude more computational effort when compared to the gradient-based methods. Thus, gradient-based algorithms are the only viable option for solving large-scale aerodynamic design optimization problems.

Keywords: aerodynamic shape optimization, benchmarking, wing design, adjoint method.

1 Introduction

With the advancement of high performance computing, numerical simulation and optimization of complex and large-scale aircraft design problems has become possible. Aerodynamic shape optimization of a transonic wing is an example of such complex design problems, often solved with respect to hundreds of design variables. Computational fluid dynamics (CFD) with a Reynolds-averaged Navier–Stokes (RANS) model is often used in aerodynamic shape optimization to accurately capture the flow, which makes computing the objective function computationally intensive. Therefore, performing high-fidelity aerodynamic shape optimization remains a challenging and expensive task [1].

The optimization can be performed with gradient-based or gradient-free methods. Gradient-based methods are best when an efficient gradient evaluation is available. The computational expense of evaluating the gradient using finite difference or complex step methods [2] is prohibitive for aerodynamic shape optimization with respect to hundreds of variables. The adjoint method, however, can provide accurate and efficient gradient evaluations [3, 4], and adjoint-based aerodynamic shape optimization has been widely used [5, 6, 7, 8, 9]. The gradient-free methods are generally simpler to implement, and claim to find the global optimum, but the computational cost is higher. In this paper, we investigate the local optima in aerodynamic shape optimization of a transonic wing. In addition, we also compare the optimization algorithms using this benchmark.

The aerodynamic shape optimization has been well-studied using various approaches. Sasaki *et al.* [10] applied an adaptive range multiobjective genetic algorithm (ARMOGA) to aerodynamic wing design. A four-objective optimization of wing shape and planform were presented using 72 design variables, subject to thickness and planform shape constraints. Moigne and Qin [11] studied aerodynamic shape optimization based on a discrete adjoint of a Reynolds-averaged Navier–Stokes (RANS) solver. A variable-fidelity optimization method combining low- and high-fidelity models was used. The optimization reduced 23% drag on a RAE2822 airfoil and 15% on a ONERA M6 wing. Their results showed that using a variable-fidelity

*Corresponding author's email: peterlyu@peterlyu.com.

method that performs most of the optimization on a low-fidelity, low-cost model (Euler equations on a coarse grid) reduces the overall computing time.

Lyu *et al.* [8] presented the results of lift-constrained drag minimization of the AIAA Aerodynamic Design Optimization Discussion Group (ADODG) Common Research Model wing[1] using a RANS solver. A 8.5% drag reduction was achieved using a multilevel optimization approach. The same optimization was also performed starting from a randomly generated initial design, and closely spaced local optima were observed.

Several authors compared the performance of different optimization methods. Foster and Dulikravich [12] compared a hybrid gradient method and a hybrid genetic algorithm for a three dimensional aerodynamic lifting body design. Zingg *et al.* [13] performed a comparison of genetic algorithm and gradient methods in aerodynamics airfoil optimization. Genetic algorithm required 5–200 times more function evaluations than gradient-based methods with adjoint sensitivity. They suggested genetic algorithm was more suited for preliminary design with low-fidelity models. Gradient-based optimizers may be more appropriate for detailed designs with high-fidelity models. Obayashi and Tsukahara [14] compared a gradient-based method with simulated annealing, and a genetic algorithm on an airfoil lift maximization problem. The genetic algorithm required the highest number of function evaluation. However, the genetic algorithm achieved the best design compared to the other two methods. Frank and Shubin [15] compared one-dimensional duct flow optimization with finite-difference gradients, optimization with analytic gradients, and an all-at once method where the flow and design variables are simultaneously altered. They concluded that the optimization with analytic gradients was the best approach that can be retrofitted to most existing codes.

In this paper, we extend the previous studies of optimizer comparison and local optima using high-fidelity aerodynamic shape optimization. We compare several optimization algorithms including 6 gradient-based methods—SNOPT, PSQP, SLSQP, IPOPT, CONMIN, GCMMA—and 2 gradient-free methods—ALPSO, NSGA2. We test those optimizers using a multi-dimensional Rosenbrock function, a wing twist optimization problem, and a wing shape optimization problem. The strengths and weaknesses of each method are discussed. This paper is organized as follows. In Section 2, we discuss the computational tools used in this study. The results of multi-dimension Rosenbrock function are presented in Section 3. The aerodynamic twist optimization is shown in Section 4, and finally, the aerodynamic shape optimization is discussed in Section 5, followed by the conclusions.

2 Methodology

This section describes the numerical tools and methods that we used for the aerodynamic shape optimization studies. These tools are components of the framework for multidisciplinary design optimization (MDO) of aircraft configurations with high fidelity (MACH) [16]. MACH can perform the simultaneous optimization of aerodynamic shape and structural sizing variables considering aeroelastic deflections [17]. However, in this paper we use only the components of MACH that are relevant for aerodynamic shape optimization: the geometric parametrization, mesh perturbation, CFD solver, and optimization algorithm.

2.1 Geometric Parametrization

We use a free-form deformation (FFD) volume approach to parametrize the wing geometry [18]. The FFD volume parametrizes the geometry changes rather than the geometry itself, resulting in a more efficient and compact set of geometry design variables, thus making it easier to manipulate complex geometries. Any geometry may be embedded inside the volume by performing a Newton search to map the parameter space to the physical space. All the geometric changes are performed on the outer boundary of the FFD volume. Any modification of this outer boundary indirectly modifies the embedded objects. The key assumption of the FFD approach is that the geometry has constant topology throughout the optimization process, which is usually the case in wing design. In addition, since FFD volumes are trivariate B-spline volumes, the derivatives of any point inside the volume can be easily computed.

[1] `https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/default.aspx`

2.2 Mesh Perturbation

Since FFD volumes modify the geometry during the optimization, we must perturb the mesh for the CFD to solve for the modified geometry. The mesh perturbation scheme used in this work is a hybridization of algebraic and linear elasticity methods, developed by Kenway *et al.* [18]. The idea behind the hybrid scheme is to apply a linear-elasticity-based perturbation scheme to a coarse approximation of the mesh to account for large, low-frequency perturbations, and to use the algebraic warping approach to attenuate small, high-frequency perturbations. For the results in this paper, the additional robustness of the hybrid scheme is not required, so we use only the algebraic scheme.

2.3 CFD Solver

We use SUmb [19] as the CFD solver, which is a finite-volume, cell-centered multiblock solver for the compressible Euler, laminar Navier–Stokes, and RANS equations (steady, unsteady, and time-periodic). SUmb provides options for a variety of turbulence models with one, two, or four equations, and options for adaptive wall functions. The Jameson–Schmidt–Turkel (JST) scheme [20] augmented with artificial dissipation is used for the spatial discretization. The main flow is solved using an explicit multi-stage Runge–Kutta method, along with geometric multigrid. A segregated Spalart–Allmaras turbulence equation is iterated with the diagonally dominant alternating direction implicit (DDADI) method.

To efficiently compute the gradients required for the optimization, we have developed and implemented a discrete adjoint method for the Euler and RANS equations within SUmb [21, 4]. The adjoint implementation supports both the full-turbulence and frozen-turbulence modes, but in the present work we use the full-turbulence adjoint exclusively. The adjoint is verified against complex-step method. [2] We solve the adjoint equations with preconditioned GMRES [22] using PETSc [23, 24, 25]. We have previously performed extensive Euler-based aerodynamic shape optimization [26, 27] and aerostructural optimization [17, 28]. However, we have observed serious issues with Euler-based optimal designs due to the missing viscous effects. While Euler-based optimization can provide design insights, we found that the resulting optimal Euler shapes are significantly different from those obtained with RANS [4]. Euler-optimized shapes tend to exhibit a sharp pressure recovery near the trailing edge, which is non-physical because such conditions near the trailing edge would cause separation. Thus, RANS-based shape optimization is necessary to achieve realistic designs.

2.4 Optimizer

A number of optimizers are studied in this paper. We use the pyOpt framework [29], which is an open source framework that provides a common interface to all optimizers. Both gradient-based and gradient-free methods are studied. In this section, we briefly describe each optimizer used in this paper.

2.4.1 SNOPT

SNOPT is a sequential quadratic programming (SQP) method that uses a smooth augmented Lagrangian merit function and reduced-Hessian semi-definite QP solver for the QP subproblems [30]. It solves large-scale problems with nonlinear constraints and a smooth objective.

2.4.2 SLSQP

SLSQP is a sequential least squares programming algorithm [31] that evolved from the least squares solver of Lawson and Hanson [32]. The optimizer uses a quasi-Newton Hessian approximation and an L1-test function in the line search algorithm. Kraft [33] also applied this method to aerodynamic and robotic trajectory optimization.

2.4.3 PSQP

PSQP is a preconditioned SQP method with a BFGS variable metric update. It can handle large scale problems with nonlinear constraints.

2.4.4 IPOPT

IPOPT implements a primal-dual interior-point algorithm with a filter line search method [34]. The barrier problem is solved using a damped Newton's method. The line search method includes a second order correction.

2.4.5 CONMIN

CONMIN solves linear or nonlinear optimization problems using the method of feasible directions [35]. It minimizes the objective function until it reaches an infeasible region. The optimization then continues by following the constraint boundaries in a descent direction.

2.4.6 GCMMA

GCMMA is a modified version of the method of moving asymptotes, designed for nonlinear programming and structural optimization [36]. It solves a strictly convex approximating sub-problem at each iteration. GCMMA guarantees convergence to a local minimum from any feasible starting point.

2.4.7 ALPSO

ALPSO is a parallel augmented Lagrange multiplier particle swarm optimization (PSO) solver written in Python [37]. This method takes advantage of PSO methods, which can solve non-smooth objective functions and is more likely to find the global minimum. Augmented Lagrange multipliers are used to handle constraints. ALPSO can be used for nonlinear, non-differentiable, and non-convex problems. Perez and Behdinan [38] applied this method to a non-convex, constrained structural problem. Other applications include aerostructural optimization of nonplanar lifting surfaces [39] and aeroservoelastic design optimization of a flexible wing [40].

2.4.8 NSGA2

NSGA2 is a non-dominant sorting based multi-objective evolutionary algorithm [41]. The optimizer enforces constraints by tournament selection. It can solve non-smooth and non-convex multi-objective functions and tends to approach the global minimum.

3 Multi-dimensional Rosenbrock Function

To examine the effectiveness of the optimizers listed above, we first minimize a multi-dimensional Rosenbrock function [42]. In addition, a nonlinear constraint is added to the formulation, and the complete problem is:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n-1} 100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \\ \text{with respect to} \quad & x \in \mathbb{R}^n \\ \text{subject to} \quad & \hat{\sum}_{i=1}^{n-1} (1.1 - (x_i - 2)^3 - x_{i+1}) \geq 0 \end{aligned}$$

The constraint is always active at the optimum. For a two-dimensional problem, the feasible optimum is at $[1.2402, 1.5385]$ with an objective value of 0.0577244. The optimizations are started from $x_i = 4$, and the design variables are bounded so that they remain in the interval $[-5.12, 5.12]$.

We set the options for each optimizer based on our best knowledge. For example, we use a swarm size of 8 and a maximum outer iteration of 4000 for ALPSO. We use a population size of 24 and 200 generations for NSGA2. We terminate all optimizations with 10^{-6} relative tolerance of 3 consecutive iterations and 10^{-6} feasibility tolerance. In this study, we investigate the computational cost and effect of increasing number of design variables. In addition, we compare results found using finite-difference gradients with those found using analytical derivatives.

Figure 1 shows the optimization path taken by each optimizer. Gradient-based methods follow through the Rosenbrock valley and converge toward the optimum. Gradient-free methods converge their population toward the optimum in a more scattered way. The convergence history of selected optimizers of the two dimensional Rosenbrock function is plotted in Figure 2. For gradient-free methods, only the best point is plotted for each iteration or generation. Most of the gradient-based optimizers converge to an objective tolerance of 10^{-5} within 150 iterations, while ALPSO converges to the same tolerance using $3,368$ iterations and NSGA2 can not converge to the same tolerance before we terminate the computation. NSGA2 terminates when the maximum number of generation (200) is reached. SLSQP is the fastest, with 34 function evaluations.

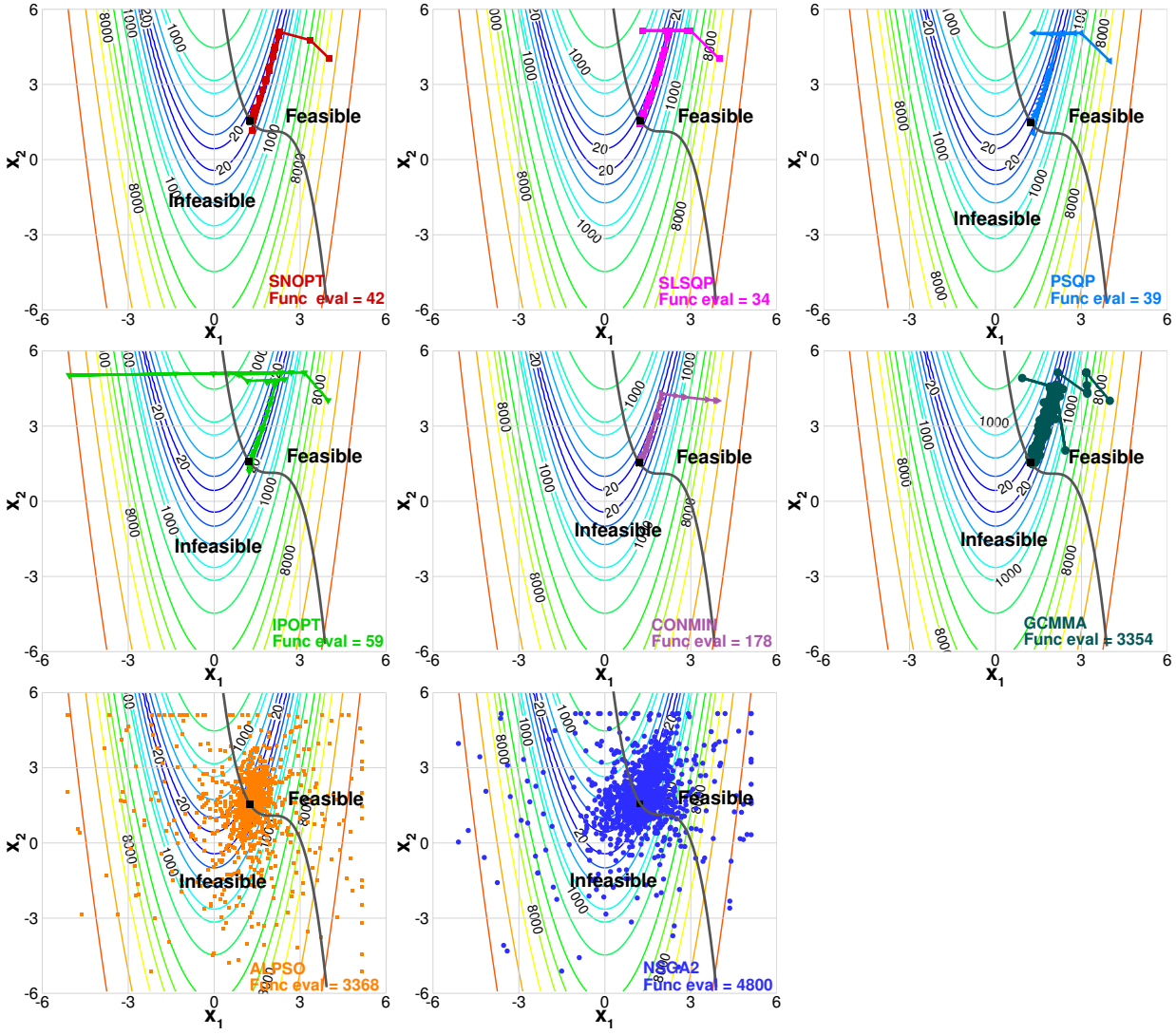

Figure 1: Optimization paths for the constrained 2D Rosenbrock function

To visualize the effect of increasing the dimensionality of the problem, we also plot the number of function evaluations required to converge the optimization for an increasing number of design variables. As shown in Figure 3, the gradient-free methods tend to have quadratic or cubic growth of function evaluations with increasing dimensionality, while the gradient-based methods follow a linear trend. The difference between gradient-based methods with finite-difference gradients and gradient-based methods with analytical gradients is significant, motivating the use of the adjoint method in aerodynamic shape optimization that we discuss later.

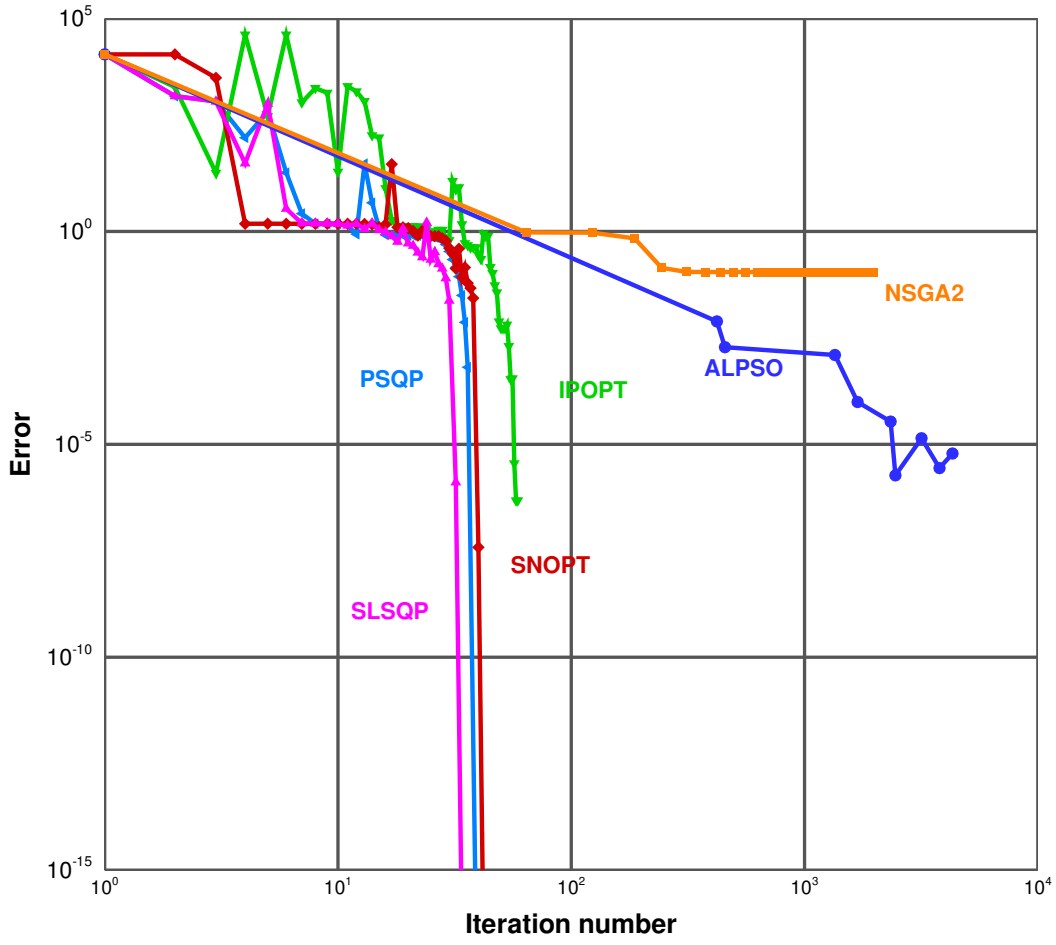

Figure 2: Gradient-based methods converge faster for the Rosenbrock problem

To investigate the local minima, we remove the constraint. Then, two minima (one local and one global) occur for higher dimensions [43]. The local minimum is at $[-1, 1, 1, \ldots]$, and the global minimum is at $[1, 1, 1, \ldots]$. We start the optimization at $[-0.8, 0.8, 0.8, \ldots]$, which is relatively close to the local minimum at $[-1, 1, 1, \ldots]$. All optimizers were able to converge to the global minimum for 2 and 4 design variables. However, for 8 design variables or more, gradient-based methods converge to the local minimum, while the gradient-free methods find the global minimum, as shown in Figure 4.

In this study, we compare the optimizers using a multi-dimensional Rosenbrock function. Gradient-free methods take 2 to 4 orders of magnitude more function evaluations to converge the optimization than most gradient-based methods. NSGA2 cannot achieve the required accuracy within 200 generations. The gradient-free methods have a higher probability of converging to a point near a global optimum. However, it requires high number of function evaluations with large number of design variables, making it infeasible for large-scale aerodynamic shape optimization. Thus, gradient-based methods with efficient gradient computations are a better choice for large-scale optimizations.

4 Aerodynamic Twist Optimization

In this study, the objective is to perform a lift-constrained drag minimization of the Common Research Model (CRM) wing [44, 45, 46]. The flow is solved using RANS equations. The adjont method is used to solve the gradients including the linearization of the turbulence model. The baseline geometry is the same

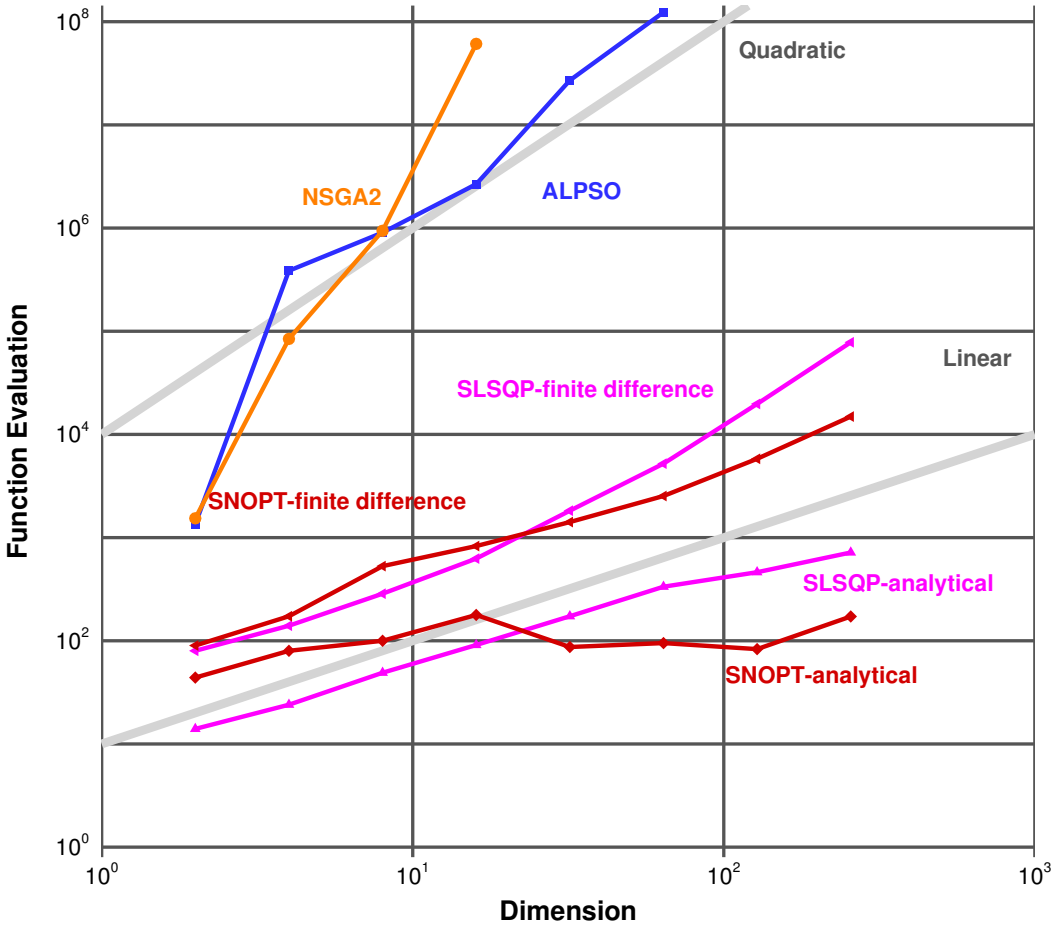

Figure 3: The gradient-free methods require an excessive number of function evaluations for large numbers of variables.

as the one used by Lyu *et al.* [8]. The specifications are given by the Aerodynamic Design Optimization Discussion Group [2]. The mesh is generated using an O-grid topology, extruded to a farfield at a distance equal to 25 times the wing span. The grid size and y^+ are listed in Table 1. We use level 3 and level 2 grids in this study.

Grid level	Grid size	y+
$L2$	$450,560$	2.213
$L3$	$56,320$	8.4086

Table 1: Grid size used in aerodynamic twist optimization

For this problem, we use 8 wing twist design variables to provide a reasonable run time to compare gradient-based and gradient-free optimizers. A lift coefficient constraint of $C_L = 0.5$ is imposed. The initial wing has zero twist. The coarse L3 grid is used. We also perform the optimization on the L2 grid using gradient-based methods.

The optimized twist, lift and pressure distributions using each optimizer and the L3 grid are shown in

[2] https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/default.aspx

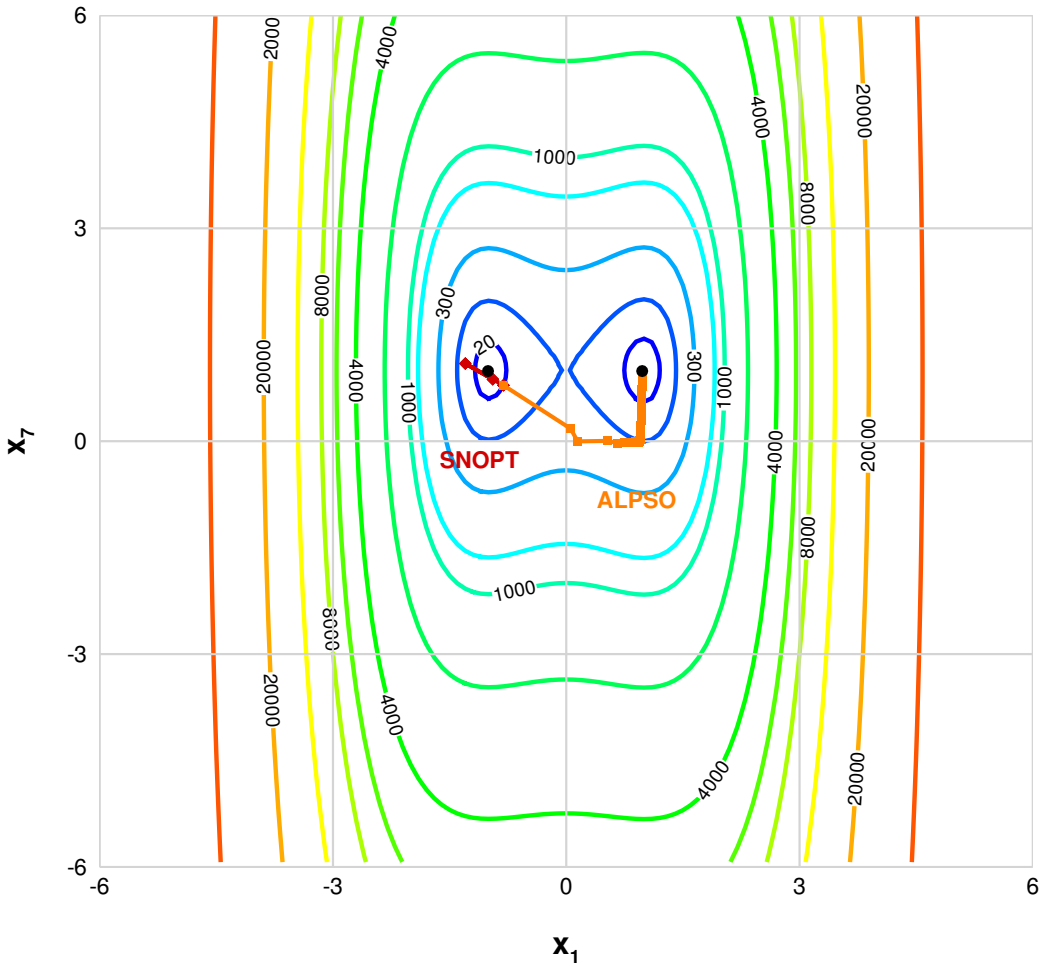

Figure 4: Visualization of the local and global minimum of 8-dimensional Rosenbrock function

Figure 5. All optimizers except NSGA2 converge to the same drag value. The difference is within 0.1 of a drag count, and the twist distributions are nearly identical.

The gradient-free optimizers take 3 orders of magnitude more iterations than the gradient-based optimizers. We compare the computational cost of the optimizers in Table 2. The relative convergence tolerances for gradient-based methods are 10^{-5} for the objective, and 10^{-4} for the constraints. The corresponding values for the ALPSO optimizer are 10^{-2} for the objective and 10^{-3} for the constraints. For this case, SLSQP, PSQP and IPOPT perform well. CONMIN is slower and did not achieve the required tolerance. For non-gradient methods, ALPSO performs better than NSGA2, as it takes half of the time of NSGA2, and converges to a better design. ALPSO converges to the same optimum as the gradient-based methods, while the optimum obtained by NSGA2 is 0.8 drag count higher with a different twist distribution.

The convergence history of the optimization is shown in Figure 6 and 7. Since the number of function evaluations for the gradient-based methods are two orders of magnitude lower than gradient-free methods, we plot the convergence of the gradient-based methods and the gradient-free methods separately. For gradient-based methods, we plot the value of objective function with respect to the number of function evaluations. For the gradient-free methods, only the best point of each iteration or generation is plotted.

After performing the comparison for the L3 grid, the same optimization is verified using the finer L2 grid. Figure 8 shows the optimized results of the gradient-based optimizers using the L2 grid. The L2 optimization

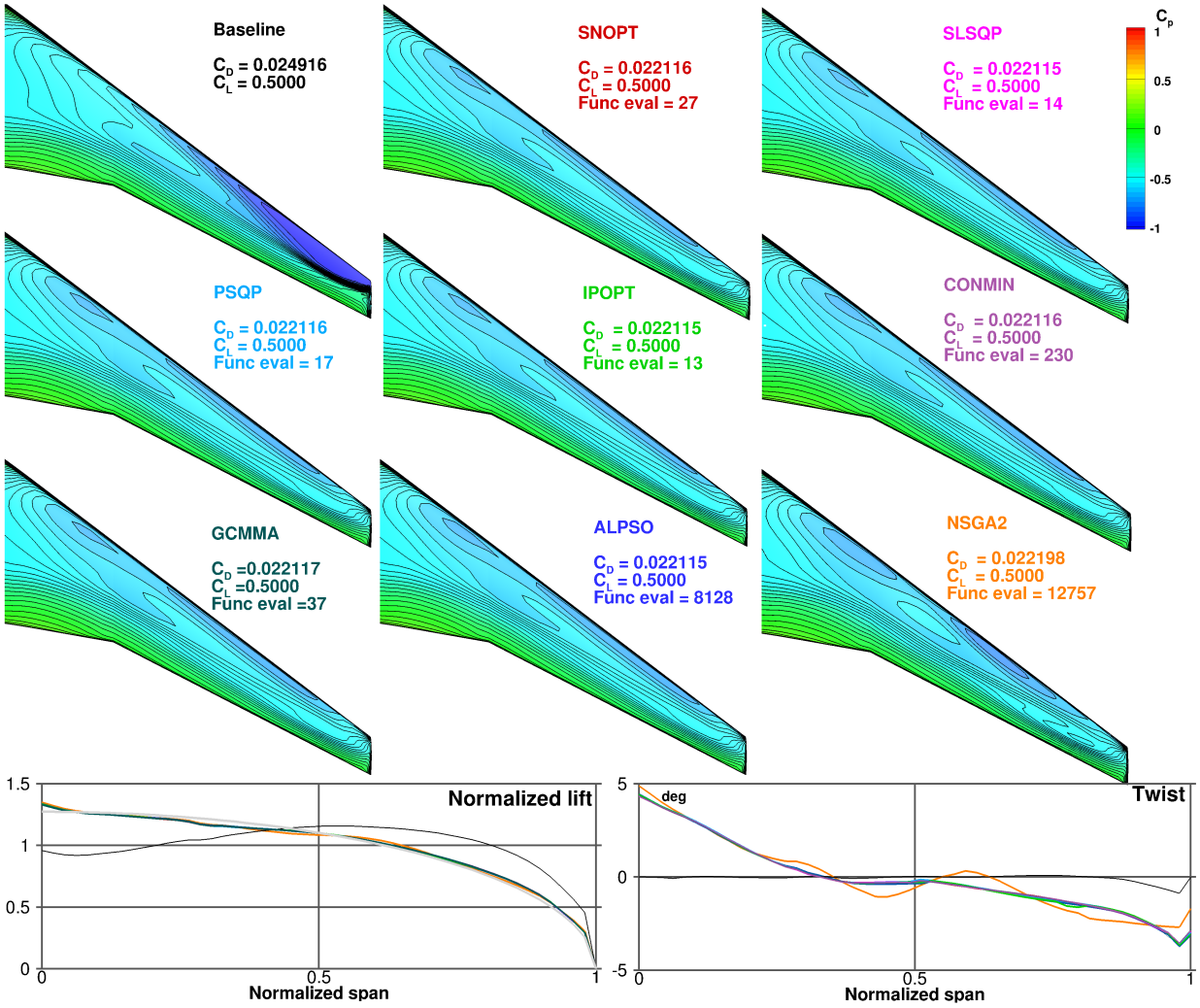

Figure 5: Aerodynamic twist optimization comparison on the L3 grid

Optimizer	Iteration numbers	Proc hours
SNOPT	27	5.81
SLSQP	14	2.56
PSQP	17	3.70
IPOPT	13	2.69
CONMIN	230	33.61
GCMMA	37	4.57
ALPSO	8129	1695.72
NSGA2	12757	2744.16

Table 2: Computational cost comparison of the twist optimization for the L3 grid

is too costly to be implemented with gradient-free methods. Using only twist design variables, the shock on the wing can not be completely removed. The drag is reduced by 29 counts. Similarly, the difference in drag between the optimizers is within 0.1 count. Thus, it seems as if the twist optimization problem has

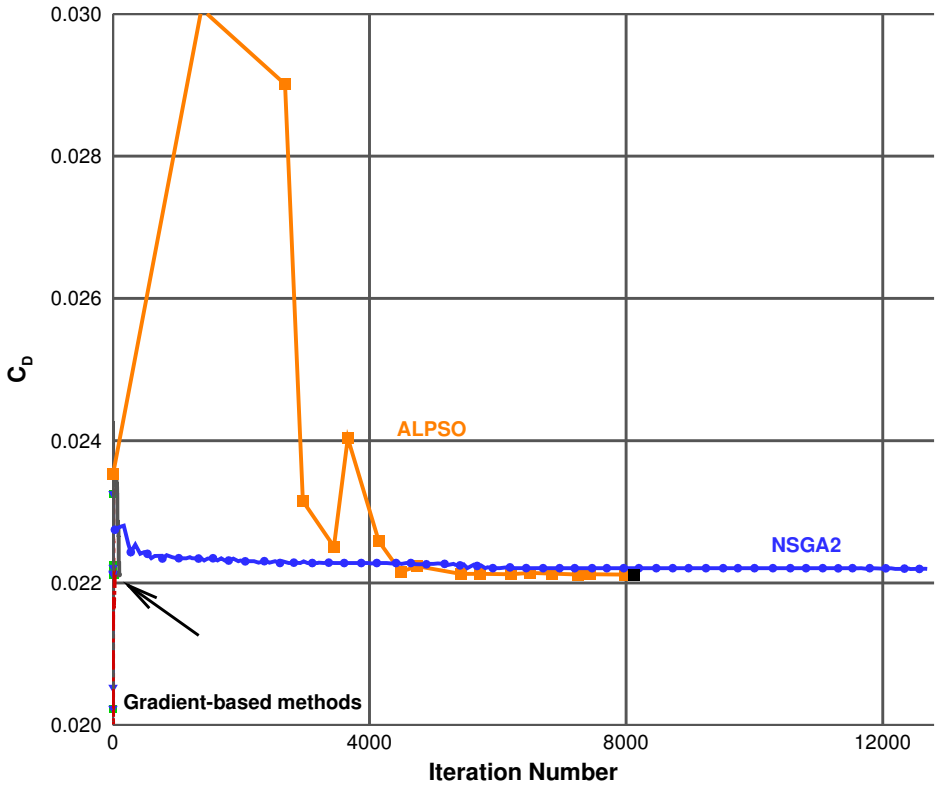

Figure 6: Twist optimization convergence history for gradient-free methods for the L3 grid

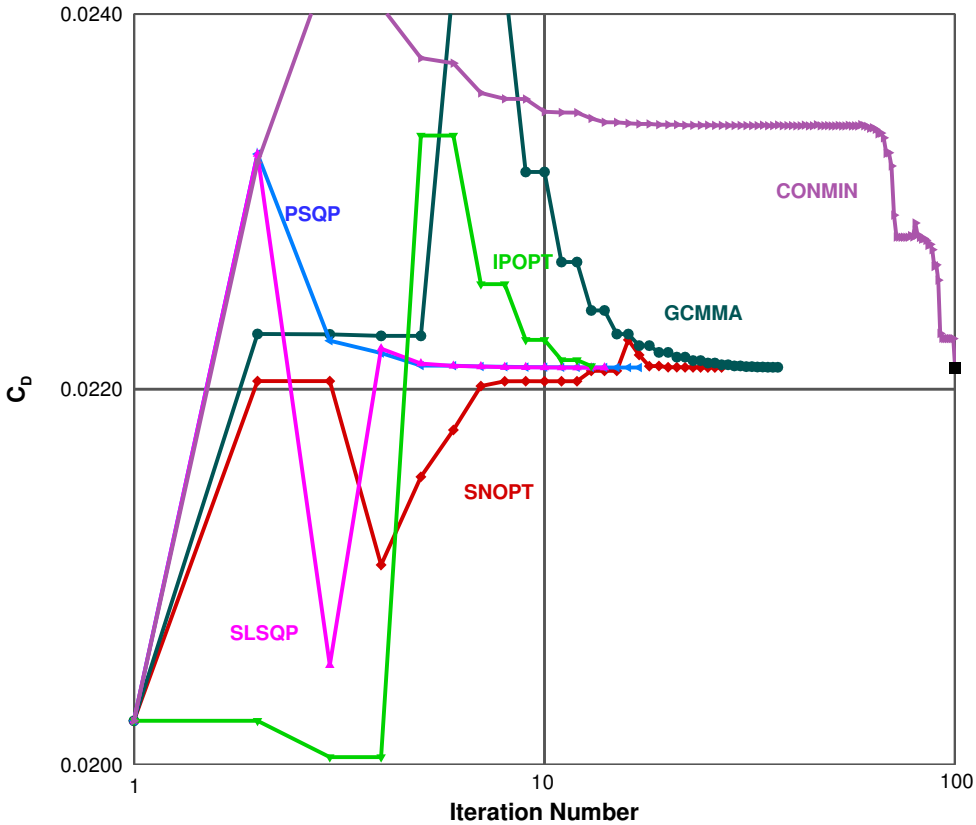

Figure 7: Twist optimization convergence history for gradient-based methods for the L3 grid

only one optimum. The lift distributions of optimized design using L2 and L3 grids are shown in Figure 9. The difference in grid resolutions results in a difference in the optimized twist distribution. The optimized design increases lift at the root and reduces lift at the tip, thus moving towards an elliptical lift distribution. However, since the optimizers minimize the total drag with only 8 twist design variables, the optimal trade-off between induced drag with wave and viscous drag is not obvious, resulting in a non-elliptical lift distribution.

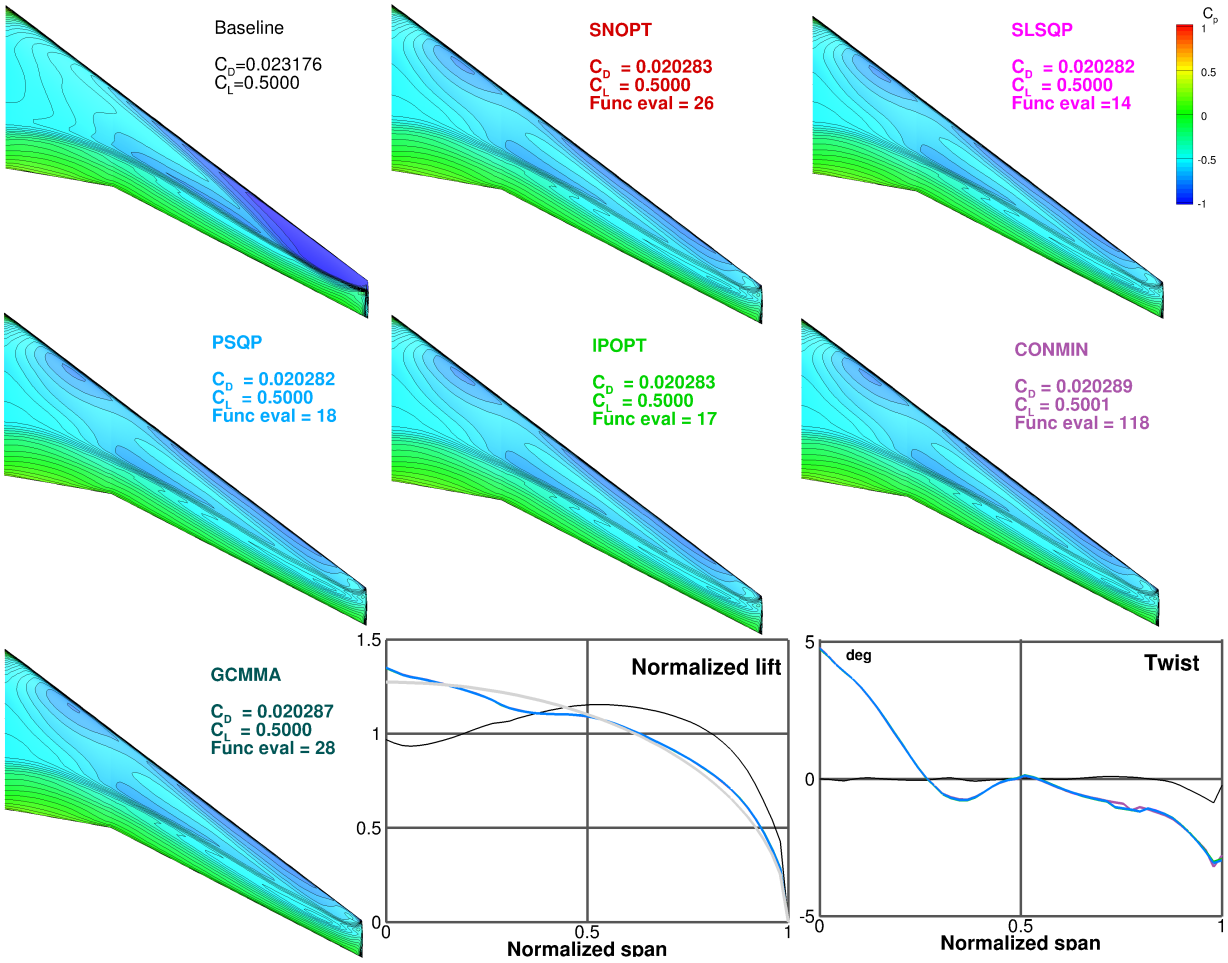

Figure 8: Aerodynamic twist optimization comparison for the L2 grid

In this study, we examined a twist design problem. We used 8 design variables subject to a lift constraint. We compared the optimized results using different optimizers on two grid levels. All optimizers converged to a similar optimum. A single global minimum is observed. The gradient-based methods converged significantly faster than the gradient-free methods.

5 Aerodynamic Shape Optimization

In this study, we use the same geometry as the twist optimization case discussed above. Instead of using just 8 twist design variables, a total of 192 shape design variables are considered. As in the previous case, the angle-of-attack is also allowed to vary, and we perform drag minimization subject to a lift constraint of $C_L = 0.5$. The wing thickness is constrained from reducing relative to the initial geometry by imposing 750 thickness constraints. In addition, a volume constraint is imposed to ensure that the internal volume does not decrease beyond the baseline volume. This problem requires significantly more computational resources than the previous case. We perform the shape optimization using 4 different gradient-based optimizers on the L2 grid. The convergence tolerance is 10^{-6} for the objective and 10^{-4} for the constraints.

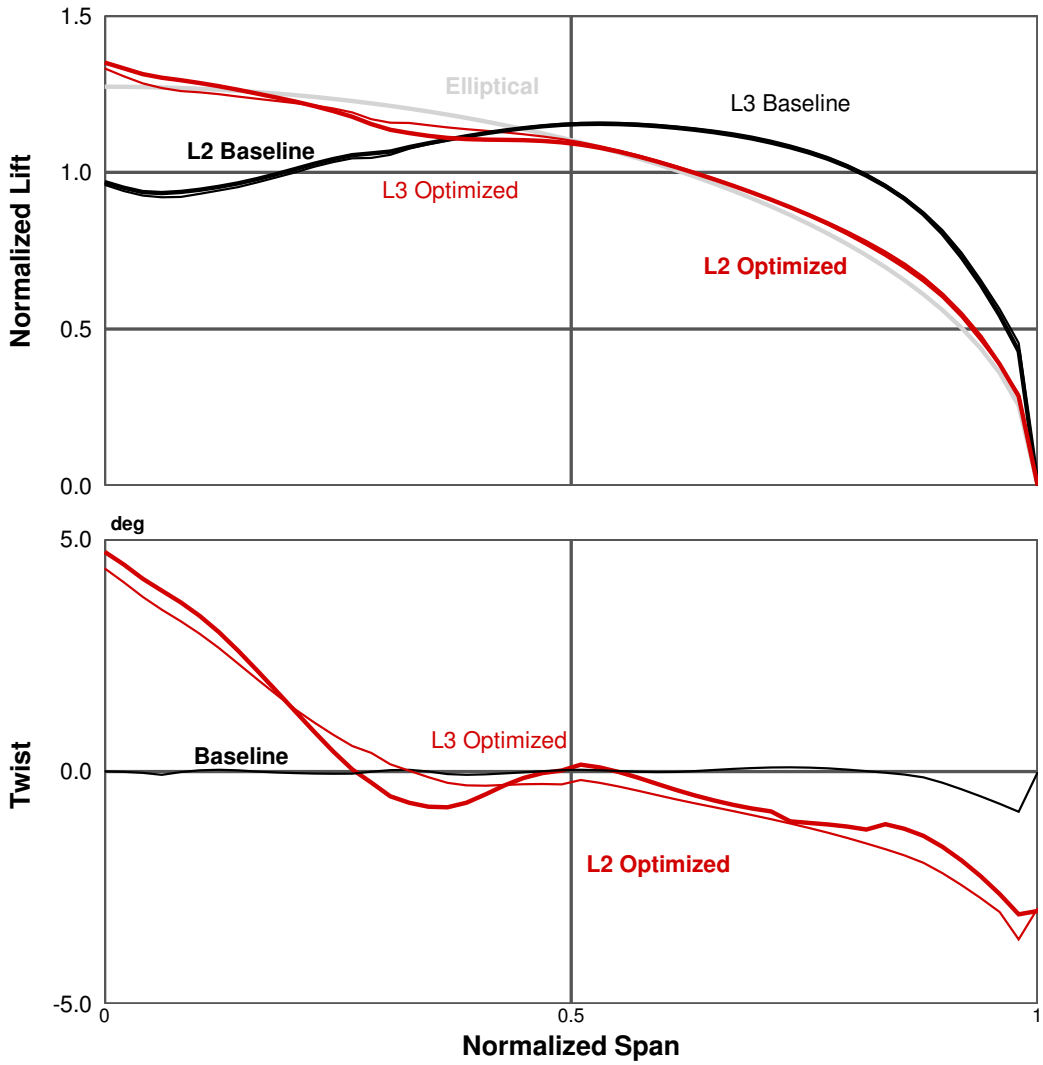

Figure 9: Lift distribution comparison of the twist optimized designs

Figure 10 shows the final design resulting from the use of different optimizers. The results from the baseline wing are shown in black. More detailed comparisons for each optimizer are shown in Figures 12– 15. The drag is reduced by 4.84%, from 206.7 to 196.6 counts, which is similar to the previous result [8].

We can see that all optimizers achieve a shock-free wing with an elliptical lift distribution. The baseline design has a strong shock, as evidenced by closely spaced C_p contours, while the optimized designs have a parallel, equally spaced pressure contours. The variation in C_D is within 1 drag count between the various optimizers. All optimized shapes are similar to each other, and only small difference in shape are observed. The comparison of the computational time for various optimizers is shown in Table 3. SNOPT converges the fastest among all optimizers. The optimized results using GCMMA is 0.2 drag count higher than the others. The convergence history is plotted in Figure 11.

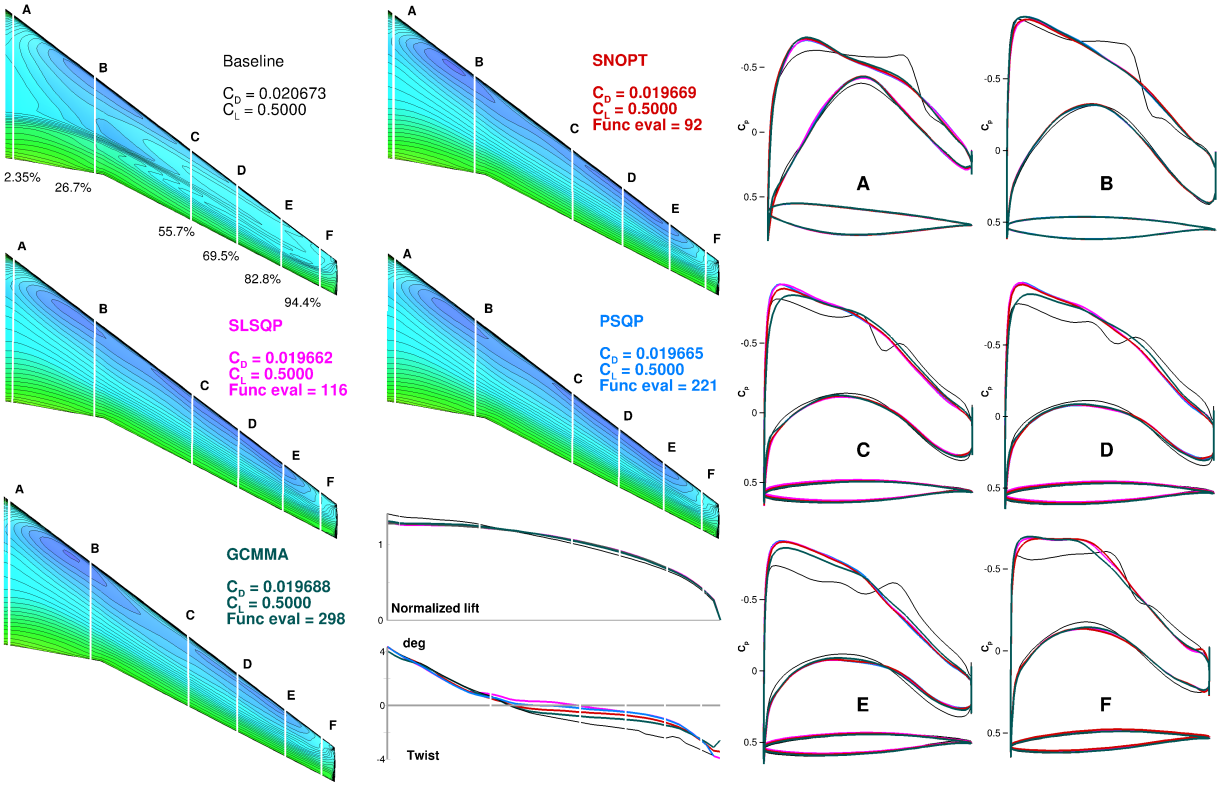

Figure 10: Aerodynamic shape optimization comparison for the L2 grid

Optimizer	Function evaluations	Proc hour
SNOPT	92	224.98
SLSQP	116	306.38
PSQP	221	562.60
GCMMA	298	772.60

Table 3: Computational cost comparison of the shape optimization for the L2 grid

6 Conclusion

We evaluated several optimization algorithms for three different aerodynamic shape optimization problems. The algorithms we considered included gradient-based methods with adjoint gradients and gradient-free methods (a particle swarm optimization and a genetic algorithm). The gradient-free methods required 2 to 4 orders of magnitude more iterations than gradient-based methods. We conclude that gradient-based methods with adjoint gradients are the best choice for solving large-scale aerodynamic design optimization problems.

7 Acknowledgments

The computations were performed on the Flux HPC cluster at the University of Michigan Center of Advanced Computing, and on the Gordon cluster of the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575.

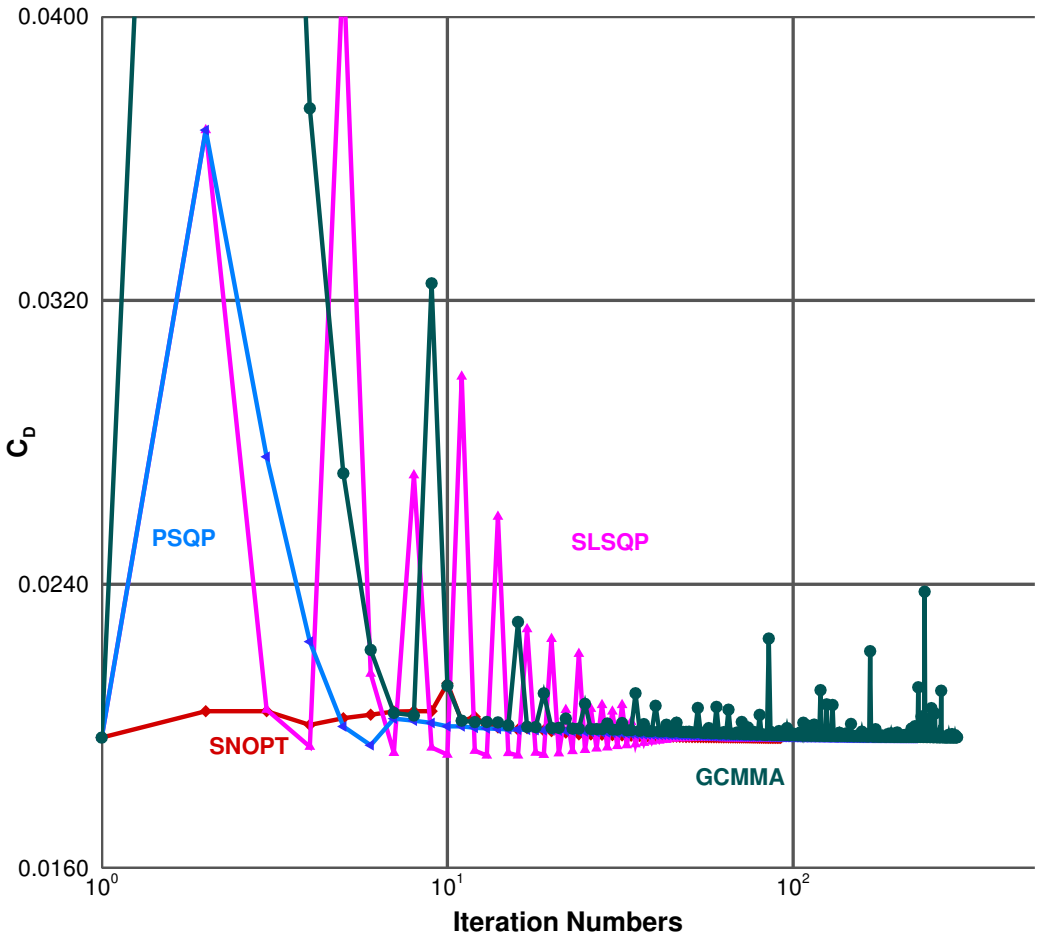

Figure 11: Shape optimization convergence history for gradient-based methods for the L2 grid

References

[1] Zhoujie Lyu and Joaquim R. R. A. Martins. Strategies for Solving High-Fidelity Aerodynamic Shape Optimization Problems. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, June 2014. AIAA 2014-2594.

[2] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. The Complex-Step Derivative Approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, 2003.

[3] Antony Jameson. Aerodynamic Design via Control Theory. 3(3):233–260, 1988.

[4] Zhoujie Lyu, Gaetan Kenway, Cody Paige, and Joaquim R. R. A. Martins. Automatic Differentiation Adjoint of the Reynolds-Averaged Navier-Stokes Equations with a Turbulence Model. In *43rd AIAA Fluid Dynamics Conference and Exhibit*, June 2013.

[5] Antony Jameson, L Martinelli, and NA Pierce. Optimum Aerodynamic Design using the Navier–Stokes Equations. *Theoretical and computational fluid dynamics*, 10(1-4):213–237, 1998.

[6] Jason E Hicken and David W Zingg. Induced-Drag Minimization of Nonplanar Geometries Based on the Euler Equations. *AIAA journal*, 48(11):2564–2575, 2010.

[7] Zhoujie Lyu and Joaquim R. R. A. Martins. Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft. *Journal of Aircraft*. (In press).

[8] Zhoujie Lyu, Gaetan K. W. Kenway, and Joaquim R. R. A. Martins. Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark. *AIAA Journal*, 2014. (In press).

[9] Zhoujie Lyu and Joaquim R. R. A. Martins. Aerodynamic Shape Optimization of an Adaptive Morphing

Trailing Edge Wing. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, June 2014. AIAA 2014-3275.

[10] Daisuke Sasaki, Masashi Morikawa, Shigeru Obayashi, and Kazuhiro Nakahashi. Aerodynamic Shape Optimization of Supersonic Wings by Adaptive Range Multiobjective Genetic Algorithms. pages 639–652, 2001.

[11] Alan Le Moigne and Ning Qin. Variable-Fidelity Aerodynamic Optimization for Turbulent Flows using a Discrete Adjoint Formulation. *AIAA Journal*, 42(7):1281–1292, 2004.

[12] Norman F Foster and George S Dulikravich. Three-Dimensional Aerodynamic Shape Optimization using Genetic and Gradient Search Algorithms. *Journal of Spacecraft and Rockets*, 34(1):36–42, 1997.

[13] D. W. Zingg, M. Nemec, and T. H. Pulliam. A Comparative Evaluation of Genetic and Gradient-Based Algorithms Applied to Aerodynamic Optimization. *European Journal of Computational Mechanics*, 17(1–2):103–126, January 2008.

[14] Shigeru Obayashi and Takanori Tsukahara. Comparison of Optimization Algorithms for Aerodynamic Shape Design. *AIAA journal*, 35(8):1413–1415, 1997.

[15] Paul D Frank and Gregory R Shubin. A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamics Design Problem. *Journal of Computational Physics*, 98(1):74–89, 1992.

[16] Gaetan K. W. Kenway, Graeme J. Kennedy, and Joaquim R. R. A. Martins. Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations. *AIAA Journal*, 52(5):935–951, 2014.

[17] Gaetan K. W. Kenway and Joaquim R. R. A. Martins. Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration. *Journal of Aircraft*, 51(1):144–160, 2014.

[18] Gaetan KW Kenway, Graeme J Kennedy, and J. R. R. A. Martins. A Cad-free Approach to High-Fidelity Aerostructural Optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX*, 2010.

[19] Edwin van der Weide, Georgi Kalitzin, Jorg Schluter, and Juan Alonso. Unsteady Turbomachinery Computations Using Massively Parallel Platforms. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.

[20] A Jameson, Wolfgang Schmidt, and Eli Turkel. Numerical Solution of the Euler Equations by Finite Volume Methods using Runge Kutta Time Stepping Schemes. In *14th AIAA, Fluid and Plasma Dynamics Conference*, 1981.

[21] Charles A. Mader, Joaquim R. R. A. Martins, Juan J. Alonso, and Edwin van der Weide. ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers. *AIAA Journal*, 46(4):863–873, April 2008.

[22] Youcef Saad and Martin H Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[23] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[24] Satish Balay, Jed Brown, , Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[25] Satish Balay, Jed Brown, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web Page, 2013. http://www.mcs.anl.gov/petsc.

[26] Charles A. Mader and Joaquim R. R. A. Martins. Stability-Constrained Aerodynamic Shape Optimization of Flying Wings. *Journal of Aircraft*, 50(5):1431–1449, September 2013.

[27] Zhoujie Lyu and Joaquim R. R. A. Martins. Aerodynamic Shape Optimization of a Blended-Wing-Body Aircraft. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.

[28] Rhea P. Liem, Gaetan K. W. Kenway, and Joaquim R. R. A. Martins. Multi-Mission Aircraft Fuel Burn Minimization via Multi-Point Aerostructural Optimization. *AIAA Journal*, 2014. (Submitted).

[29] Ruben E. Perez, Peter W. Jansen, and Joaquim R. R. A. Martins. pyOpt: A Python-Based Object-

Oriented Framework for Nonlinear Constrained Optimization. *Structures and Multidisciplinary Optimization*, 45(1):101–118, 2012.

[30] Philip E Gill, Walter Murray, and Michael A Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM journal on optimization*, 12(4):979–1006, 2002.

[31] Dieter Kraft. A Software Package for Sequential Quadratic Programming. Technical report, Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center, 1988.

[32] Charles L Lawson and Richard J Hanson. *Solving Least Squares Problems*, volume 161. SIAM, 1974.

[33] Dieter Kraft. Algorithm 733: TOMP–Fortran Modules for Optimal Control Calculations. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):262–281, 1994.

[34] Andreas Wächter and Lorenz T Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical programming*, 106(1):25–57, 2006.

[35] Garret N Vanderplaats. *CONMIN, a FORTRAN Program for Constrained Function Minimization: User's Manual*, volume 62282. Ames Research Center and US Army Air Mobility R&D Laboratory, 1973.

[36] Krister Svanberg. The Method of Moving Asymptotes–a New Method for Structural Optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.

[37] P. W. Jansen and R. E. Perez. Constrained Structural Design Optimization via a Parallel Augmented Lagrangian Particle Swarm Optimization Approach. *Computers & Structures*, 89(13–14):1352–1366, 7 2011.

[38] RE Perez and K Behdinan. Particle Swarm Approach for Structural Design Optimization. *Computers & Structures*, 85(19):1579–1588, 2007.

[39] Peter Jansen, Ruben. E. Perez, and Joaquim R. R. A. Martins. Aerostructural Optimization of Nonplanar Lifting Surfaces. *Journal of Aircraft*, 47(5):1491–1503, 2010.

[40] Sohrab Haghighat, Joaquim R. R. A. Martins, and Hugh H. T. Liu. Aeroservoelastic Design Optimization of a Flexible Wing. *Journal of Aircraft*, 49(2):432–443, 2012.

[41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[42] H. H. Rosenbrock. An Automatic Method for Finding the Greatest or Least Value of a Function. *The Computer Journal*, 3(3):175–184, 1960.

[43] Jorge J Moré, Burton S Garbow, and Kenneth E Hillstrom. Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software (TOMS)*, 7(1):17–41, 1981.

[44] J. Vassberg. Introduction: Drag Prediction Workshop. *Journal of Aircraft*, 45(3):737–737, Jun 2008.

[45] John Vassberg, Mark Dehaan, Melissa Rivers, and Richard Wahls. Development of a Common Research Model for Applied CFD Validation Studies. In *26th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, August 2008.

[46] John Vassberg. A Unified Baseline Grid about the Common Research Model Wing/Body for the Fifth AIAA CFD Drag Prediction Workshop (Invited). In *29th AIAA Applied Aerodynamics Conference*, Jul 2011.

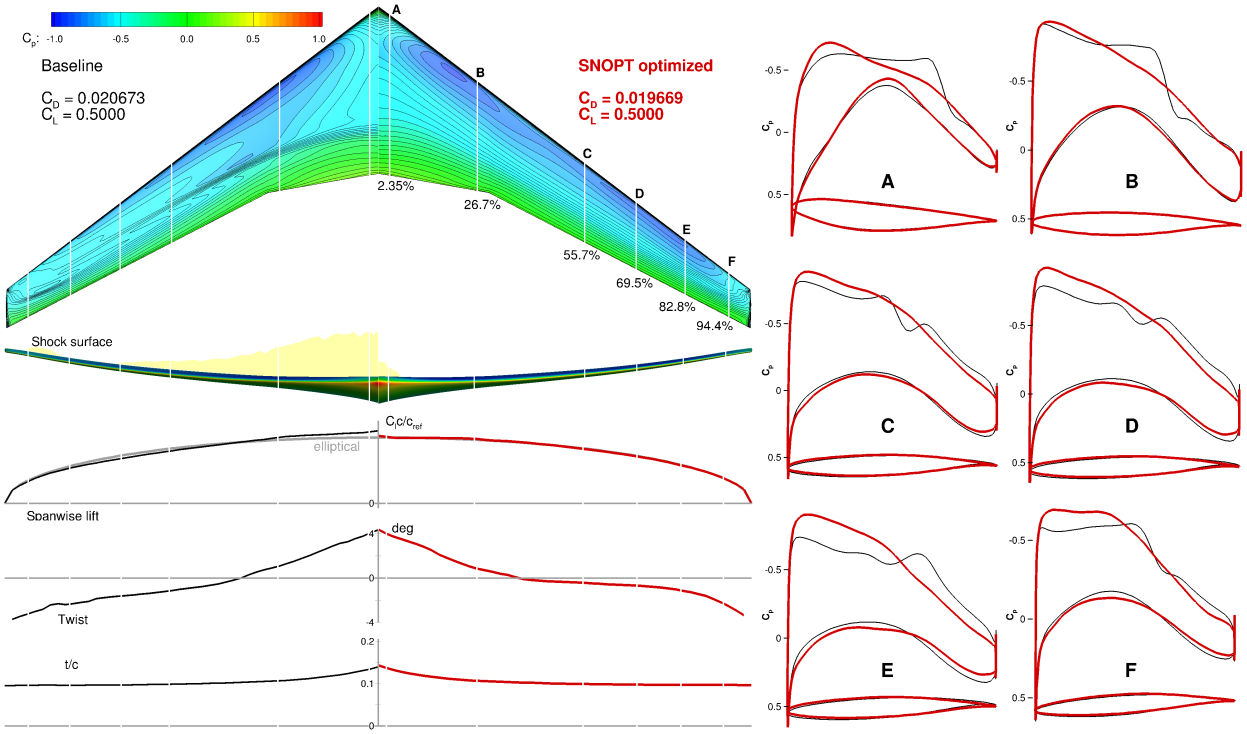

Figure 12: Wing shape optimization using SNOPT

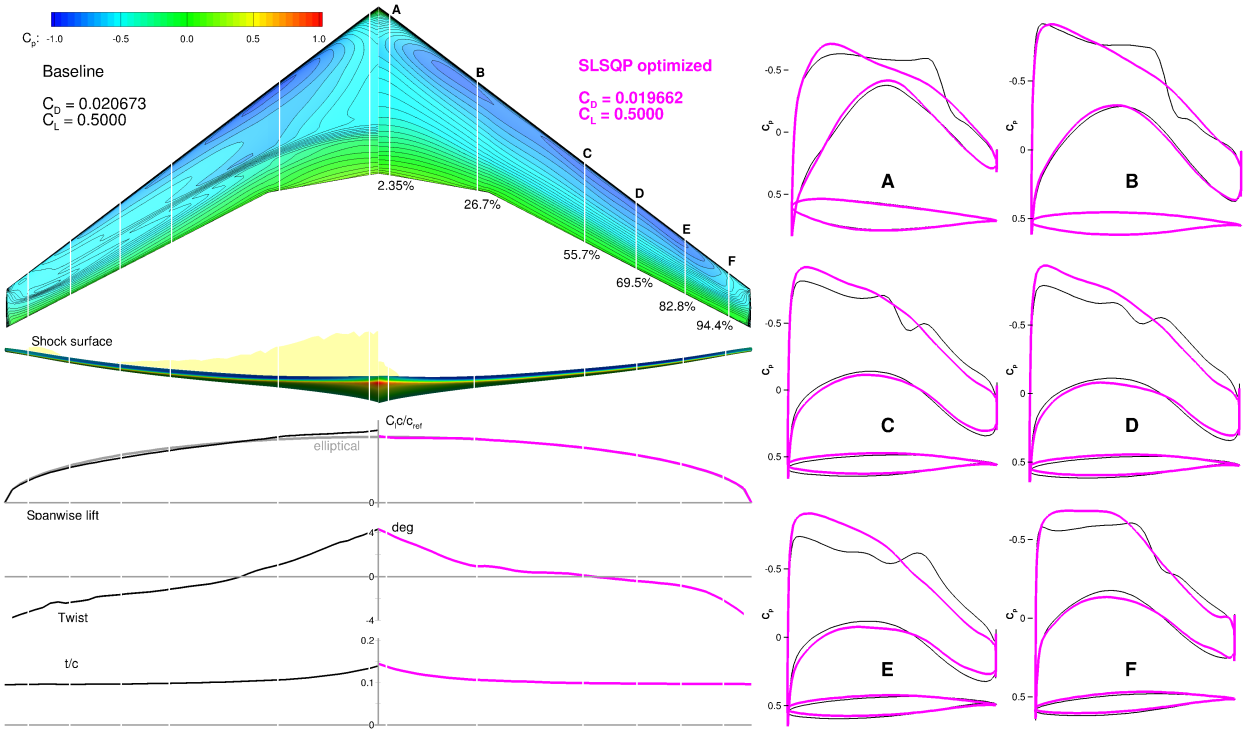

Figure 13: Wing shape optimization using SLSQP

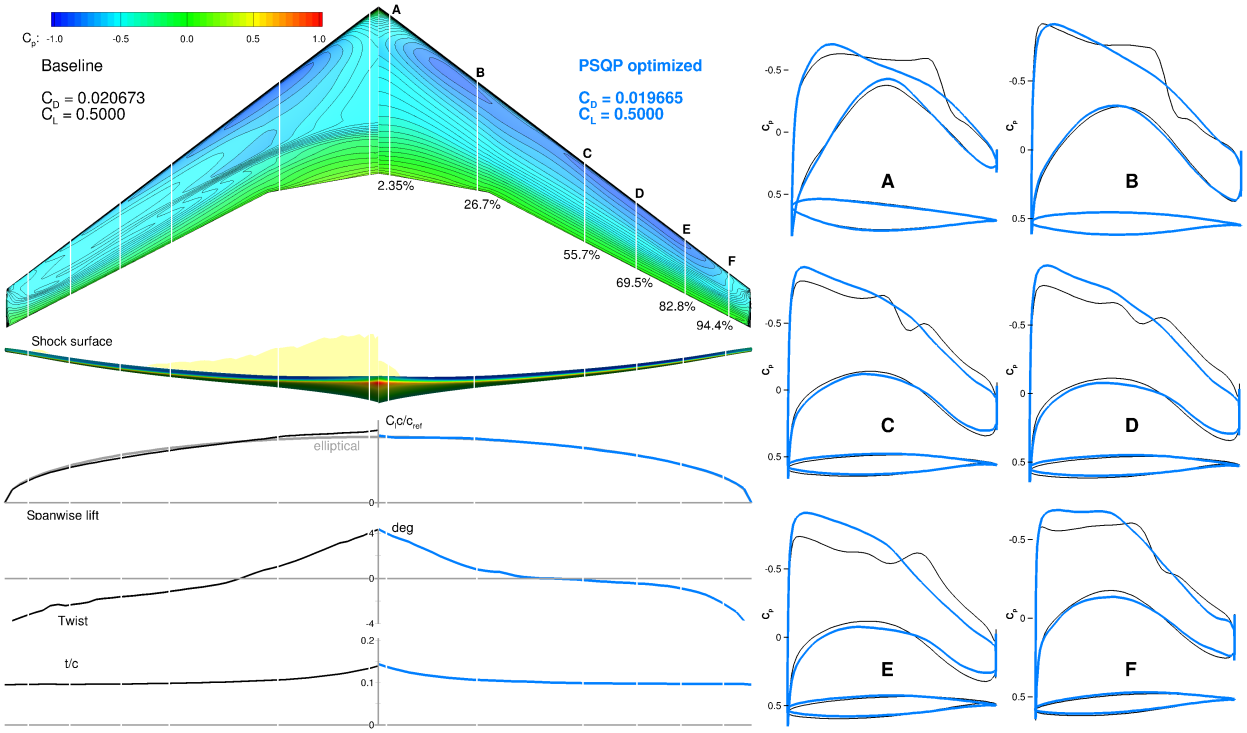

Figure 14: Wing shape optimization using PSQP

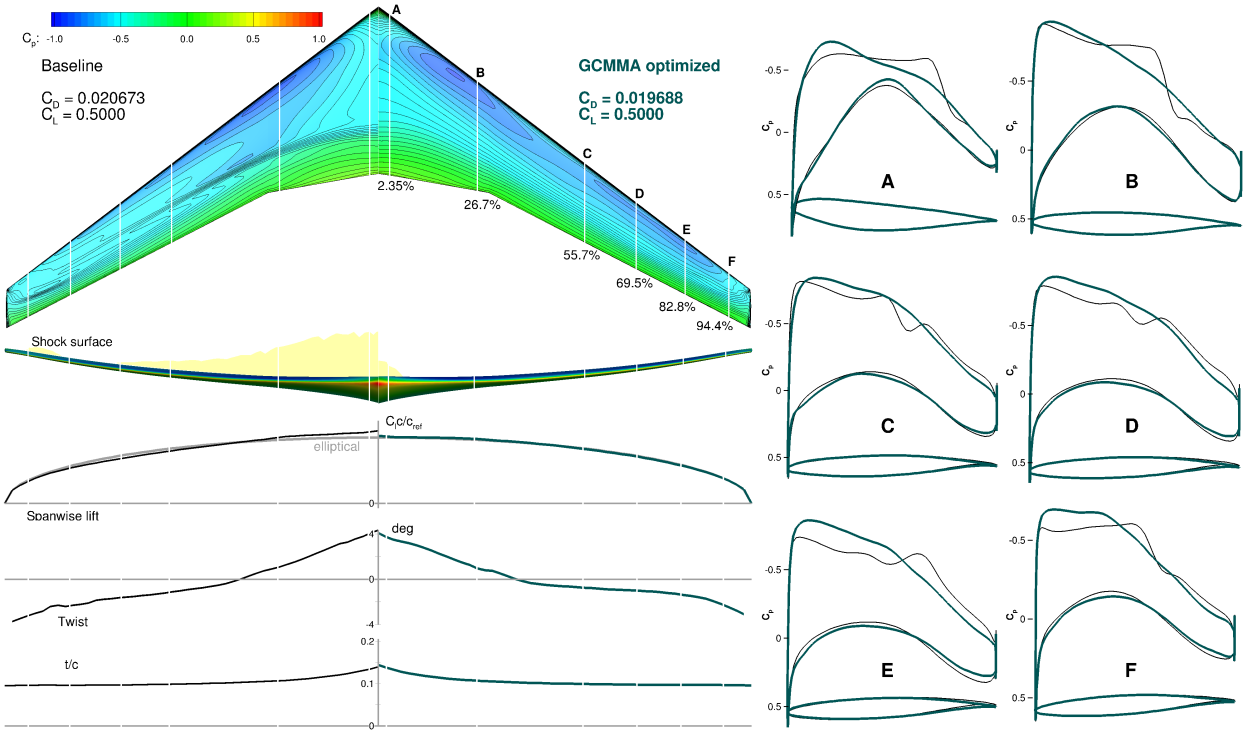

Figure 15: Wing shape optimization using GCMMA