Progressive Subdivision Curves for Aerodynamic Shape Optimisation

D. A. Masters*,
Department of Aerospace Engineering, University of Bristol

N. J. Taylor†,
MBDA UK Ltd, Filton

T. C. S. Rendall‡ and C. B. Allen§
Department of Aerospace Engineering, University of Bristol

This work presents a shape parameterisation method based on multi-resolutional subdivision curves and investigates its application to aerodynamic optimisation. Subdivision curves are defined as the limit curve of a recursive application of a subdivision rule, which provides an intrinsically hierarchical set of control polygons that can be used to provide surface control at varying levels of fidelity. This is used to construct a progressive aerofoil parameterisation that allows an optimisation to be initialised with a small number of design variables, and then periodically increased in resolution through the optimisation. This brings the benefits of a low dimensional design space (high convergence rate, increased robustness, low cost finite-difference gradients) while still allowing the final results to be from a high-dimensional design space. In this work the progressive refinement technique is tested on a variety of optimisation problems. For each problem a range of 'static' (non-progressive) subdivision schemes (equivalent to cubic B-splines) are also used as a control group. For all the optimisation cases the progressive schemes perform comparably or better than the static methods, often providing a significant computational advantage, and in many cases allowing a solution to be found when the static method would otherwise finish prematurely in a local optimum.

I. Introduction and Background

With optimisation becoming more common in aerodynamic design, a significant effort is being made to improve both its effectiveness and its efficiency[1, 2]. Within an optimisation procedure the choice of shape parameterisation controls the relationship between the optimisation design variables and the aerodynamic surface itself. Consequently the choice of shape parameterisation method can have a significant impact on the effectiveness and efficiency of the overall procedure[3]. Many different methods have been used within an aerodynamic optimisation framework, from standard geometric curve definitions such as B-Splines[4] or NURBS[5] to aerospace-specific methods such as CST[6, 7], Hicks-Henne bump functions[8, 9] or PARSEC[9, 10] to Free-Form Deformation[11, 12, 13, 14], proper orthogonal decomposition[2, 15, 16] or the discrete method[17]. A whole variety of options are available, however all are subject to the 'curse of dimensionality'. In the context of aerodynamic optimisation this refers to the problems associated with increasing the number of design variables used in the optimisation procedure. For many optimisation schemes the number of objective function evaluations is proportional to the number of design variables used, in conjunction with this a large number of design variables can lead to poor convergence rates and poor design space conditioning. Considering that for aerodynamic optimisation each objective function evaluation equates to a single, often expensive, aerodynamic solution the impact of dimensionality can be huge. On the other hand, the fidelity of the parameterisation, and therefore the design space of the problem, is directly linked to the number of design variables. This often leads to a compromise between available resources and desired accuracy of the results.

One approach to reducing this effect is to control the shape with a series of nested, hierarchical parameterisation schemes and increase the fidelity at intervals throughout the optimisation process. This approach is akin to a multi-grid method for parameterisation and was first used in an aerodynamic optimisation setting by Beux and Dervieux[18]. It

*Graduate Student, AIAA Student Member, dominic.masters@bristol.ac.uk, Bristol, BS8 1TR, UK
†Capability Leader, Aerodynamic Tools & Methods, AIAA Senior Member, nigel.j.taylor@mbda-systems.com, WG3, PO Box 5, Filton, Bristol, BS34 7QW, UK
‡Lecturer, AIAA Member, thomas.rendall@bristol.ac.uk, Bristol, BS8 1TR, UK
§Professor of Computational Aerodynamics, AIAA Senior Member, c.b.allen@bristol.ac.uk, Bristol, BS8 1TR, UK

has since been applied to a range of aerofoil optimisation problems using a variety of different parameterisation frameworks such as Bèzier curves[19, 20, 21, 22], Bèzier surface FFD[23, 24, 25, 26], RBFs[27] and B-Splines with a knot insertion algorithm[28, 29]. In general, they have shown that implementation of progressive nested parameterisations can improve the convergence rate, robustness and final solution of an optimisation procedure. This paper investigates the application of progressive shape parameterisation techniques to multi-resolution subdivision curves for aerodynamic optimisation procedures.

Subdivision curves and surfaces are a shape parameterisation method used predominantly in computer graphics and animation. They describe a smooth curve or surface based on an initial coarse network of points and a simple subdivision rule of refinement. By successively applying the subdivision rule increasingly fine networks are created which at the refinement limit create a curve or surface. In some cases these limit curves are equivalent to B-Splines, for example Chaikin's corner cutting scheme[30] is equivalent to uniform quadratic B-Splines and further extensions to higher order uniform B-Splines can also be derived[31]. For subdivision surfaces there are similar equivalences; for a set of regular, rectangular control points Doo-Sabin subdivision[32] is equivalent to bi-quadratic B-spline surfaces and Cattmull-Clark subdivision[33] is equivalent to uniform bi-cubic B-spline surfaces. However, subdivision surfaces are generalisable to arbitrary topologies while B-Splines are not, which is a key benefit of the subdivision framework. B-Splines and subdivisions share many characteristics; the method of implementation is, however, one area of major difference. B-splines utilise continuous parametric representation whereas subdivisions use a hierarchical process of discrete refinement. It is the innately hierarchical nature of subdivisions that make them easily applicable to multi-resolution analysis.

Multi-resolution analysis utilises hierarchical nested data sets to increase efficiency by allowing operations to be performed at varying levels of detail. For geometry and shape parameterisation applications this typically means the ability to implement either coarse geometry changes while maintaining the fine detail, or fine geometry changes while maintaining the overall shape. This approach has been implemented comparably from both B-spline[34] and subdivision[35] perspectives in both their two-dimensional and three-dimensional forms, though it would seem that in three dimensions the advantage of being able to represent arbitrary topologies with subdivision surfaces has lead to it becoming the industry standard choice in multi-resolution computer animation[36]. It is also slowly being incorporated into some computer aided design (CAD) packages[37, 38, 39].

The aim of this work is to explore the use of multi-resolution subdivision curves for aerodynamic shape optimisation with particular emphasis on how they can be used to improve both the efficiency and accuracy of current optimisation schemes.

II. Subdivision Curves

Subdivision curves are defined as the limit of a process of repeated subdivision refinement of an initial control polygon. Each subdivision refinement defines a new set of smoother, denser points as a linear combination of the old points. For this reason the refinements can conveniently be expressed as a simple matrix transformation

$$C_{n+1} = P_n C_n \tag{1}$$

from old points C_n to new points C_{n+1}. For simple subdivision schemes on closed polygons these matrices are just the two row offset repetition of a subdivision 'mask'. Two common examples are Chaikin's rule [30] (equations 2 and 3) and the Cubic B-spline rule (equations 4 and 5). Figure 2 shows a simple implementation of Chaikin's rule on a closed polygon.

Note how the rows of the transformation matrices all sum to 1 and therefore describe a weighted averaging of the previous points, this is a key feature of all subdivision transformation matrices. Equations 2 and 4 describe the matrix for areas of smooth subdivision whereas equations 3 and 5 represent an area with a corner or endpoint, where the non-averaged matrix row represents the corner or endpoint itself.

Given a set of subdivision matrices P the N^{th} subdivision level C_N can be expressed as

$$C_N = P_{N-1} \dots P_{n+1} P_n C_n \tag{6}$$

for some $n < N$. The limit surface can therefore be described as

$$C_\infty = \lim_{N \to \infty} C_N = \dots P_{n+1} P_n C_n. \tag{7}$$

In practice this calculation must be truncated at some point and the limit surface calculated. A typical solution to this is to calculate an evaluation matrix based on eigenanalysis[40] that pushes the subdivision points to their limit locations. With this method however the user cannot directly parameterise the surface itself and therefore cannot specify the

Chaikin's rule:

$$P_n = \begin{bmatrix} \ddots & \ddots & & & & \\ 0.25 & 0.75 & & & & \\ & 0.75 & 0.25 & & & \\ & 0.25 & 0.75 & & & \\ & & 0.75 & 0.25 & & \\ & & 0.25 & 0.75 & & \\ & & & \ddots & \ddots & \end{bmatrix}, \quad (2) \qquad P_n = \begin{bmatrix} \ddots & \ddots & & \\ 0.5 & 0.5 & & \\ & 1 & & \\ & 0.5 & 0.5 & \\ & 0.25 & 0.75 & \\ & & \ddots & \end{bmatrix}. \quad (3)$$

Cubic B-Spline rule:

$$P_n = \begin{bmatrix} \ddots & \ddots & & & & \\ 0.125 & 0.75 & 0.125 & & & \\ & 0.5 & 0.5 & & & \\ & 0.125 & 0.75 & 0.125 & & \\ & & 0.5 & 0.5 & & \\ & & 0.125 & 0.75 & 0.125 & \\ & & & \ddots & \ddots & \end{bmatrix}, \quad (4) \qquad P_n = \begin{bmatrix} \ddots & \ddots & & & & \\ 0.5 & 0.5 & & & & \\ & 1 & & & & \\ & 0.5 & 0.5 & & & \\ & & 0.75 & 0.25 & & \\ & & 0.1875 & 0.6875 & 0.125 & \\ & & & \ddots & \ddots & \end{bmatrix}. \quad (5)$$

Figure 1. Matrix representations of the Chaikin and Cubic B-Spline subdivision schemes.

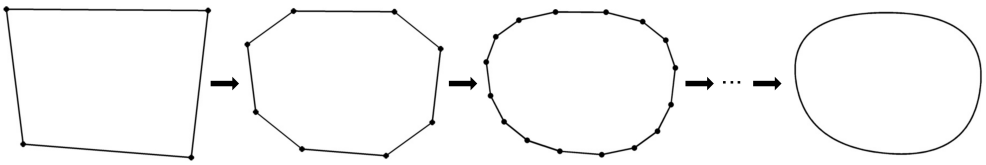


Figure 2. Simple closed subdivision using Chaikin's Scheme

distribution of points around the surface. As the distribution of points around an aerofoil can be very important an alternative method is used in this work. This is done by exploiting the equivalency between B-spline curves and some subdivision formulations, and creating a continuous B-spline transformation between the a maximum subdivision level N and the limit curve. This B-Spline transformation can then be formulated as matrix P_N^{BS} such that a desired point distribution is achieved. This method is however constrained to subdivisions with B-spline equivalents.

It is then convenient to define

$$\phi_n = P_N^{BS} P_{N-1} \dots P_n \quad (8)$$

and therefore the relationship between any level of control polygon and the limit surface can be expressed as

$$C_\infty = \phi_n C_n. \quad (9)$$

The columns of ϕ_i represent the basis functions of the subdivision scheme. Figure 3 shows the basis functions for a Chaikin subdivision with fixed endpoints, these are identical to the basis functions of an equivalent quadratic B-spline. From this it can be seen that the different subdivision levels can be used to control and deform the limit surface at varying levels of fidelity.

III. Reverse Subdivision Curves

Given a set of points it can often be desirable to obtain the subdivision polygon (or closest possible match) that produced them. Consider a fine set of points C_n; the coarser set of points C_{n-1} can simple be calculated as the least squares solution of equation 1,

$$C_{n-1} = P_{n-1}^{+} C_n, \quad (10)$$

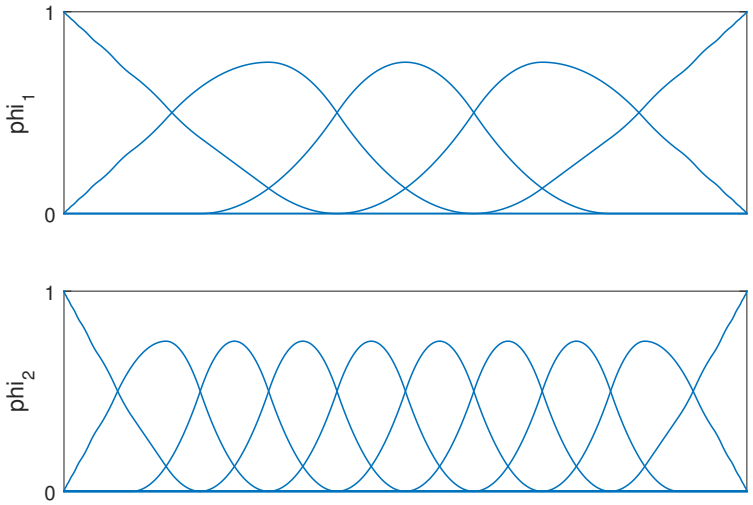


Figure 3. The basis functions for the first two levels of a Chaikin subdivision with fixed endpoint conditions with a 4 point initial control polygon

where + denotes the Moore-Penrose pseudo-inverse. However as P_n is a non-square, non-invertible matrix this leads to some loss of information with the result that, for almost all cases,

$$P_{n-1}C_{n-1} \neq C_n. \tag{11}$$

For this reason it is important to retain any errors created through the least squares process and include them in any subsequent refinement. This can be done very conveniently and efficiently by extending the refinement matrices P_n by any orthogonal compliment $Q_n = null(P_n^T)$ and extending the subdivision refinement equation such that

$$\begin{aligned} C_{n+1} &= \begin{bmatrix} P_n & Q_n \end{bmatrix} \begin{bmatrix} C_n \\ D_n \end{bmatrix} \\ &= P_n C_n + Q_n D_n, \end{aligned} \tag{12}$$

for some set of error coefficients D_n. If we then let $\begin{bmatrix} P_n & Q_n \end{bmatrix}^{-1} = \begin{bmatrix} A_n \\ B_n \end{bmatrix}$ we can then re-express equation 10 as the reverse subdivision equations

$$C_{n-1} = A_n C_n, \tag{13}$$

$$D_{n-1} = B_n C_n. \tag{14}$$

This importantly creates a one-to-one relationship between the subdivision refinement levels and thus allows information to be propagated uniquely and exactly in either the refinement or coarsening direction (figure 4).

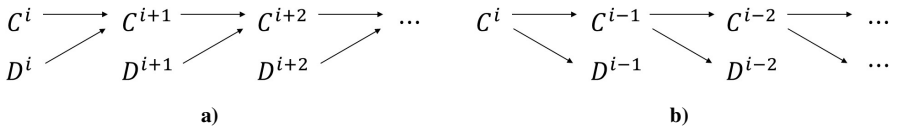


Figure 4. Process for subdivision refinement (a) and reconstruction (b)

Equation 9 can then be reformed using equation 12 such that

$$C_\infty = \phi_n C_n + \sum_{i=n}^{N} \phi_{i+1} Q_i D_i. \tag{15}$$

By storing and including these error terms it means that any shape can now be represented by any subdivision rule as long as the correct error terms D_n are used.

IV. Aerofoil Parameterisation

To parameterise the aerofoils in this work a cubic B-Spline subdivision scheme is used with a single, closed initial polygon with 'corners' at the leading and trailing edges. This ensures that the position of the leading and trailing edges are equal to the 'corner' control points at every subdivision level. This is equivalent to using two distinct subdivision curves for the upper and lower surfaces with shared endpoints at the leading and trailing edges. In physical terms the trailing edge is a corner, however the leading edge is not, for this reason the control points closest to the leading edge are constrained to lie directly above and below it; this enforces the vertical surface required.

In this work the first subdivision level has been defined by six control points with two fixed points defining the leading and trailing edges at $[x/c, z/c] = [0, 0]$ and $[1, 0]$ respectively, and points at $x/c = 0$ and $x/c = 0.5$ on each surface free to move in the z direction. The number of control points in the final subdivision level is then restricted to less than the number of discrete points in the initial/final aerofoil, otherwise equation 12 cannot be formed such that $[P_n\ Q_n]$ is square for all n. In this work, unless otherwise specified, the aerofoils have been defined by 301 cosine distributed points which implies that the maximum level is defined by the eighth subdivision polygon containing 260 points. Table 1 shows the total number of control points in each level used.

Subdivision Level	Number of Control Points
1	6
2	8
3	12
4	20
5	36
6	68
7	132
8	260

Table 1. Number of control points at each subdivision level

Given an initial aerofoil $C^{initial}$, the starting initial subdivision positions can then be calculated by the recursive application of equation 13. The resulting control point positions, $C_n^{initial}$, represent the least-squares approximations of the initial aerofoil for the limit surfaces $\phi_n C_n^{initial}$. At each of these subdivision levels a set of fixed error terms D_n can then be calculated by applying equation 14. Then for the set of control points $C_n^{initial}$ calculated, equation 15 implies that

$$C^{initial} \equiv \phi_n C_n^{initial} + \sum_{i=n}^{N} \phi_{i+1} Q_i D_i, \quad \forall n \leq N. \tag{16}$$

The benefit of this is that any subdivision optimisation can be set to start from exactly the prescribed initial aerofoil rather than a best approximation. This approach has been used for all the optimisations in this paper. Figure 5 shows the initial positions for the first four subdivision levels for a NACA0012.

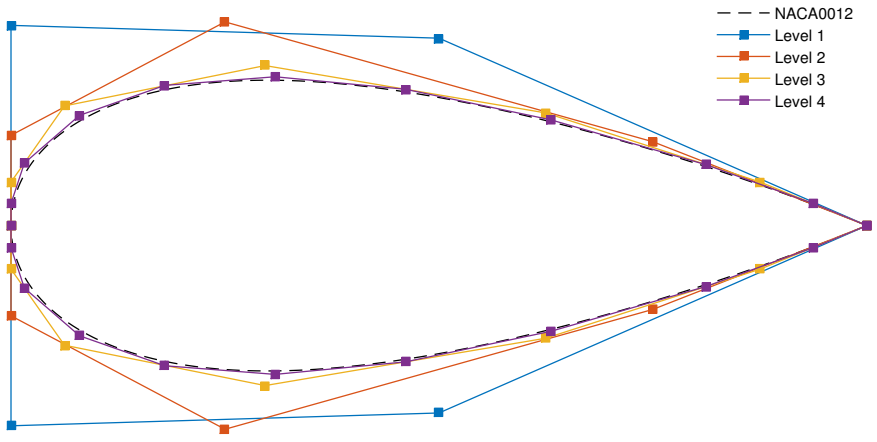


Figure 5. Initial control point positions for the first four subdivision levels for a NACA0012.

Each of these available subdivision levels can then be used to parametrise the aerofoil independently, in what have been described hereafter as 'static' schemes, or in an ascending series, which have been described as 'progressive'. For the static schemes the aerofoils are defined by the equation

$$C^{aero} = \phi_n C_n + \sum_{i=n}^{N} \phi_{i+1} Q_i D_i \tag{17}$$

for the desired subdivision level n with control points C_n. For the progressive schemes the same definition applies but when a 'refinement trigger' is activated $n \rightarrow n+1$. This increases the number of design variables, and therefore available fidelity, while maintaining the aerofoil shape exactly. The aim of this is to use fewer design variables in the early stages of optimisation to increase the rate of convergence, robustness and, for finite-difference gradients, reduce gradient calculation times. Then, as the lower dimensional design space looks to be fully exploited, the refinement process is applied to allow the larger design space to be explored.

For each optimisation case investigated a variety of subdivision schemes have been applied. Each available subdivision level has been applied as a static parameterisation as well as three progressive schemes starting from the first, second and third levels respectively and refining up to the final level. The static schemes are equivalent to using normal cubic B-Splines so act as the control group for which the benefits of the progressive tests have been compared against.

V. Optimisation Methodology

In this work subdivion curves have been applied to a range of aerodynamic optimisation problems. For all of these tests the multi-purpose large-scale optimiser SNOPT[41] was used. This is a gradient-based sequential-quadratic programming (SQP) method that employs a reduced-Hessian BFGS search-direction and, in this work, a non-derivative line-search technique. This was coupled with three different aerodynamic solvers; a potential flow solver, the full-potential, boundary-coupled solver VGK[42] and the Euler solver from SU^2[43, 44]. For the potential flow solver and VGK the objective function gradients were calculated by finite-difference, importantly this means that the cost of calculating the gradients is proportional to the number of design variables used. For SU^2 however, the gradients were calculated using the adjoint equations, and consequentially the cost of calculating the gradients is independent of the number of design variables used.

Optimiser convergence was determined based on the activation of one of three criteria:

1. The Karush-Kuhn-Tucker (KKT) first-order optimality condition[41] satisfying the tolerance of 10^{-6}
2. The optimiser unable to improve the objective function
3. For the progressive schemes *not* on the final level, satisfying the refinement condition (equation 18)

The refinement condition aims to trigger the refinement of the subdivision scheme when the optimisation has exploited most of the available gains from the current design space and is approaching the local minimum. This moment can be very hard to identify as it is very hard to differentiate between the optimiser converging to a local optimum and the optimiser traversing a difficult area of the design space. If refinement is triggered too early the under-exploitation of design space can result in slower convergence and possibly a poorer final result, and if it is triggered too late, over-exploitation of the design space can waste resources. A methods for approximating the optimum refinement time was proposed by Anderson[27] where refinement was triggered when the convergence of the objective function with respect to the iterations dropped below some proportion $t < 1$ of the maximum attained. A slightly modified version of this is implemented in this work. This triggers refinement if the rolling average of the log-scaled gradients is less than a proportion t of the max rolling average of the log scaled gradients; i.e.

$$\frac{t}{w} \sum_{j=0}^{w-1} |G_{k-j}| < \max_{m \leq l \leq k} \frac{1}{m} \sum_{j=0}^{m-1} |G_{l-j}| \tag{18}$$

$$\text{where} \qquad G_i = \log_{10}(J_{i-j}) - \log_{10}(J_{i-j-1}) \tag{19}$$

for objective function J, iteration k (at current refinement level), $t < 1$ and integers w and m. The parameter t controls the change in gradient required to trigger the scheme and w and m control the size of the rolling average windows for the maximum and current gradients. If small values are used for w and m this defines a very aggressive triggering system, for well behaved, consistently converging optimisations this ensures that iterations are not wasted converging areas close to a local minimum. For more complex optimisation procedures this can however cause premature triggering

when the optimiser only makes a small improvement through a highly non-linear area. For this reason these parameters can be increased to average the gradients and only trigger refinement when improvements are consistently small over a given window. In this work a value of $t = 0.1$ has been used throughout and w, m have been set to 1 or 3 depending on the complexity of the optimisation problem.

VI. Optimisation Results

Results for a range of optimisation problems are presented here with the aim of assessing the benefits of using progressive subdivision curves in aerodynamic optimisation. A range of problems and solvers have been used in order to investigate the performance of the subdivision curves at varied levels of complexity.

A. Inverse Design

Three inverse design problems have been considered and each one has been completed with the potential flow solver and VGK. Inverse design problems usually pose well conditioned optimisation which, when coupled potential flow solver, present the simplest optimisation problems investigated. The addition of the shocked solutions with VGK then presents a slight increase in overall complexity.

For all of these inverse design optimisations the objective function is defined as the root mean squared difference between the current and target pressure distributions, i.e.,

$$J = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Cp_i - Cp_i^{target}\right)^2}. \tag{20}$$

All of the control points except the trailing edge point are allowed to vary in the vertical, z, direction thus the number of design variables is equal to one less than the number of control points in the active subdivision level.

1. *NACA0012 → NACA4410*

TEST CONFIGURATION

Initial Aerofoil:	*NACA0012*,
Target Aerofoil:	*NACA4410*,
Flow Conitions:	$\alpha = 0$,
	$M = 0.7$ (VGK only),
Refinement Parameters:	
Potential Flow:	$t = 0.1, \quad w = 1, \quad m = 1,$
VGK:	$t = 0.1, \quad w = 3, \quad m = 1.$

This first inverse design problem starts from an initial NACA0012 aerofoil and minimises the difference between its pressure distribution and that of a NACA4410. The initial and target pressure distributions are displayed in figure 6a. Figure 6b shows how each of the different subdivision configurations converge for increasing iterations, figure 6c then displays the same data but plotted against number of solver evaluations required.

In each figure the solid lines represent the static subdivision schemes and the dashed lines the progressive schemes. In figure 6b the static schemes follow the trend that, as the level, and therefore the number of design variables, is increased the rate of convergence decreases. As well as this the final optimisation results improve up to level five, then for the levels above that, the optimisations fail to exploit the full design space and stagnate. The three progressive schemes show comparable convergence rates to the first five static configurations and look to converge to similar results for equivalent design variables. However, after they have refined to the sixth level, with 67 design variables, it can be seen that the objective function keeps reducing. This enables all three of the progressive methods to access area of the design space unobtainable with the static methods and thus achieve better final results..

Figure 6c then shows the same data but for number of solver evaluations required throughout the optimisation. As the design variable gradients are calculated using finite-differences the number of solver evaluations is proportional to the number of design variables. As a result of this it can be seen that the computational cost associated with the static methods increases with the subdivision level used. For the progressive methods the computational cost is lower for the earlier iterations and consequentially the convergence rate, relative to the static methods, is improved in this region.

This results in the progressive methods displaying an objective-to-cost convergence rate better or comparable to all of the static methods.

Figures 7b and 7c show the results for the same problem but with the solutions calculated with VGK at $M = 0.7$. In this case a shock is present on target aerofoil solution (figure 7a); this increases the complexity and difficulty of the optimisation problem. It can be seen that the static results improve up to level three but then for levels four and above they fail to attain a reasonable solution. The progressive methods however continue to improve throughout the refinement phases converging to a final result approximately ten times better than the best static solution.

2. *NACA0012 → RAE2822*

TEST CONFIGURATION

Initial Aerofoil:	*NACA0012*,
Target Aerofoil:	*RAE2822*,
Flow Conitions:	$\alpha = 0$,
	$M = 0.78$ (VGK only),
Refinement Parameters:	
Potential Flow:	$t = 0.1, \quad w = 1, \quad m = 1,$
VGK:	$t = 0.1, \quad w = 3, \quad m = 1.$

The second inverse design problem matches the pressure distribution to a RAE2822 at zero incidence starting from a NACA0012. The pressure distributions for initial and target aerofoils are shown in figure 8a. Figure 8b shows the convergence of the objective function with respect to the major iterations. It can be seen that the static methods show a steady improvement in results, apart from for level 5 which stagnates and fails early on. The eighth level provides the best static final result with the sixth and seventh levels closely matching it. It should however be noted that the seventh and eight levels require double and quadruple the number of iterations required by the sixth. The '1 → 8' and '2 → 8' progressive methods then give results comparable to the eighth level static method with the '3 → 8' progressive method improving on it by a further 65%.

Figure 8c then shows the convergence history with respect to the number of solver evaluations. While the progressive results were comparable to the sixth level static method relative to the total number of iterations, relative to the number of solver evaluations they can be seen to display a significantly better convergence rate. It should also be noted that the eight level static optimisation requires around 15 times as many solver evaluations as the sixth level case yet achieves a very similar result. This emphasises the large number of evaluations that can be required to fully converge a static, high-fidelity surface parameterisation.

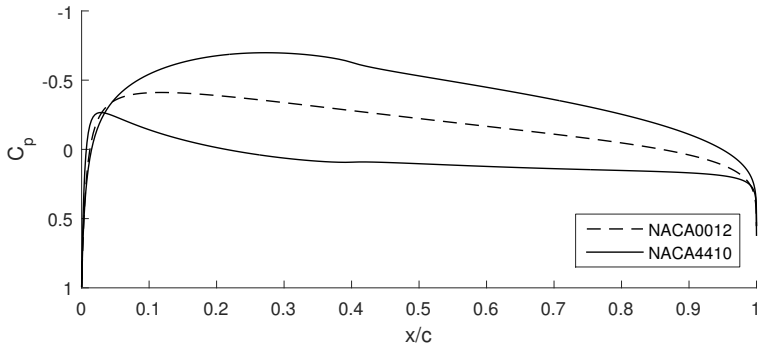
For the associated VGK optimisations (figures 9a-9c) the best static result is achieved by the fourth level scheme with 19 design variables; cases for 35 with more design variables stagnate at poor results. The progressive methods starting from the second and third levels achieve results approximately 20 times better than the best static results. The progressive method starting from the first level, however, seems to stagnate; this may be an indication that the refinements were triggered too early or too late.

3. *NACA0012 → ONERA M6*

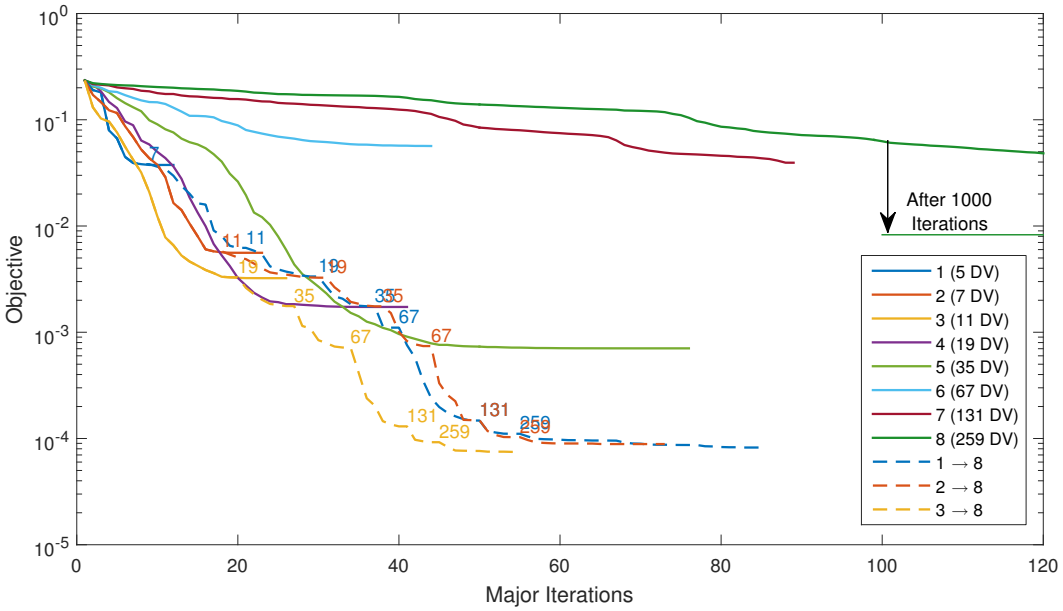
TEST CONFIGURATION

Initial Aerofoil:	*NACA0012*,
Target Aerofoil:	*ONERA M6 (D section)*,
Flow Conitions:	$\alpha = 3$,
	$M = 0.7$ (VGK only),
Refinement Parameters:	
Potential Flow:	$t = 0.1, \quad w = 1, \quad m = 1,$
VGK:	$t = 0.1, \quad w = 3, \quad m = 1.$

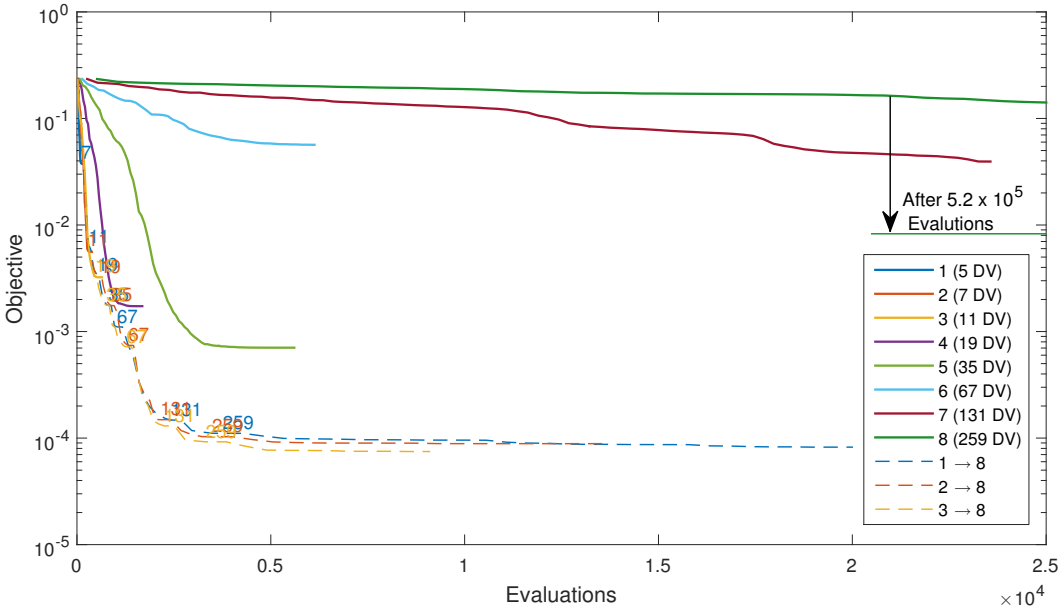
The final inverse design case minimises the difference in pressure distributions between an ONERA M6 at three degrees incidence again, starting from a NACA0012. Figure 10b and 10c shows the convergence of the objective function with respect to the number of iterations and solver evaluations when the potential flow solver is used and the



a) Initial and target pressure distributions

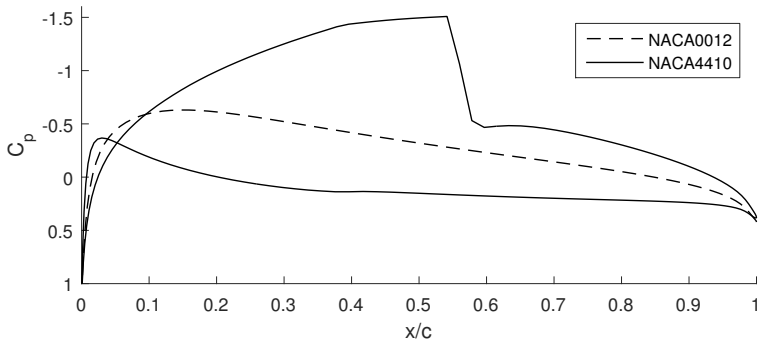


b) Convergence of the objective function w.r.t. the number of iterations

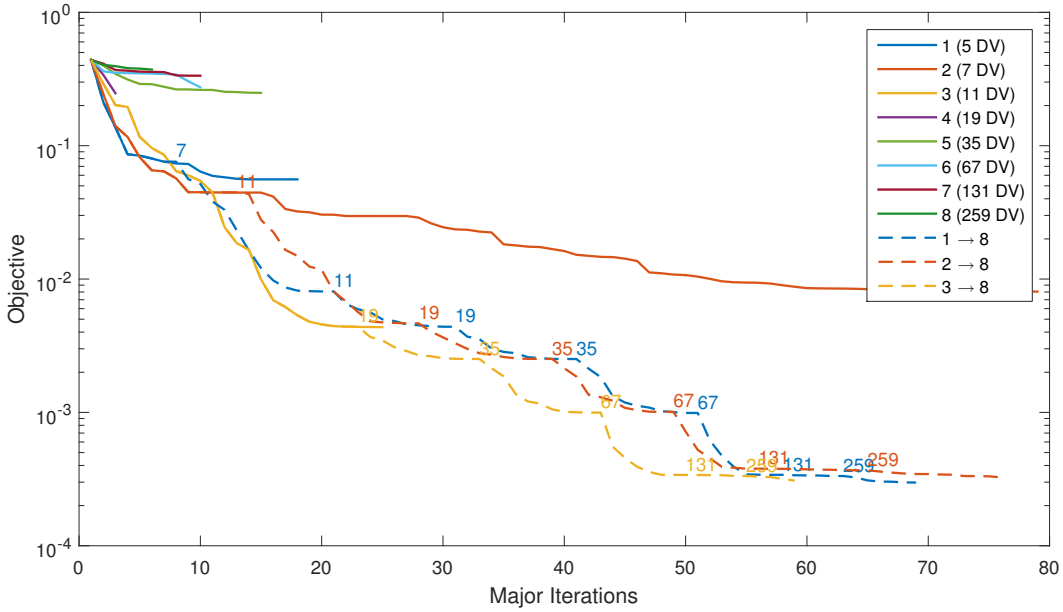


c) Convergence of the objective function w.r.t. the number of solver evaluations

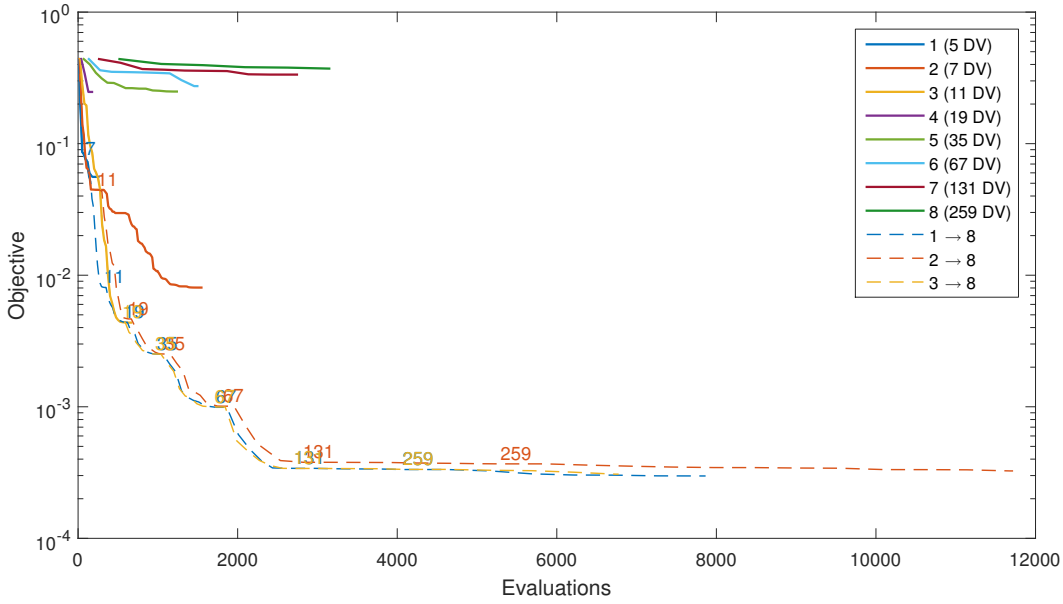
Figure 6. Results for the inverse design from a NACA0012 to a NACA4410 using a potential flow solver at $\alpha = 0$. Numbers represent the number of design variables after a refinement.



a) Initial and target pressure distributions

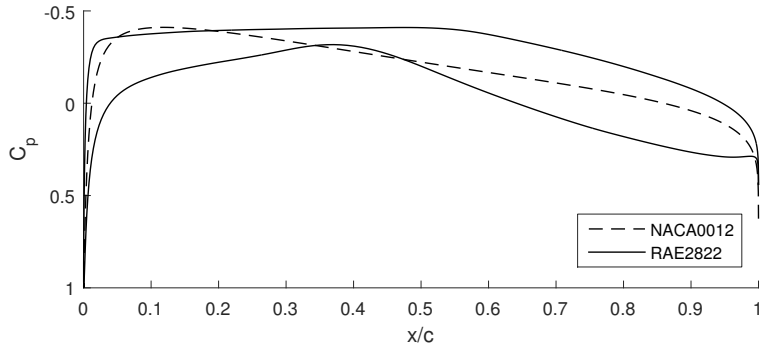


b) Convergence of the objective function w.r.t. the number of iterations

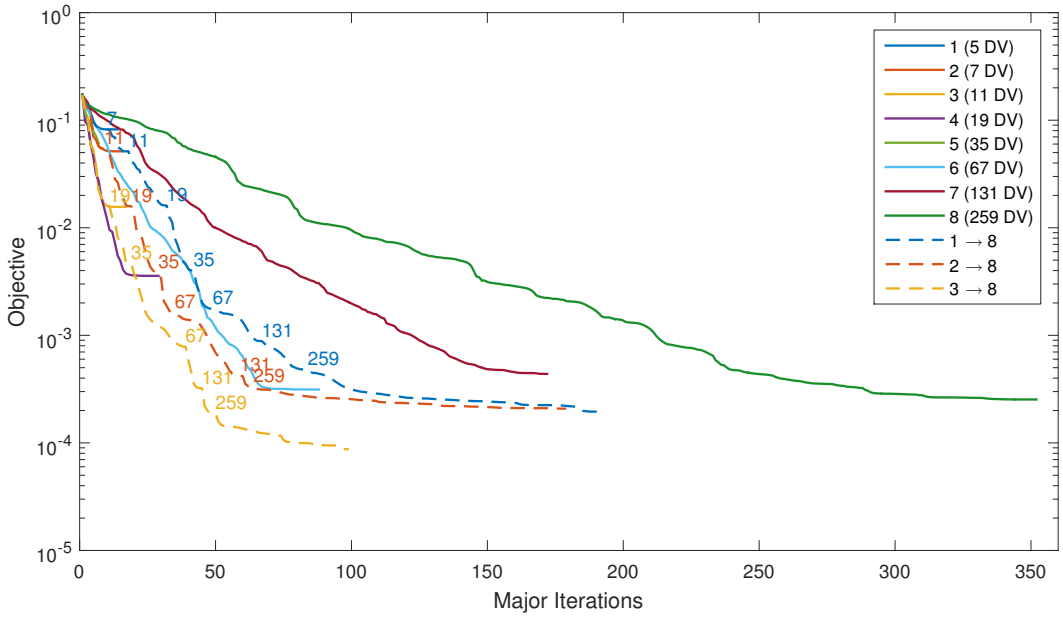


c) Convergence of the objective function w.r.t. the number of solver evaluations

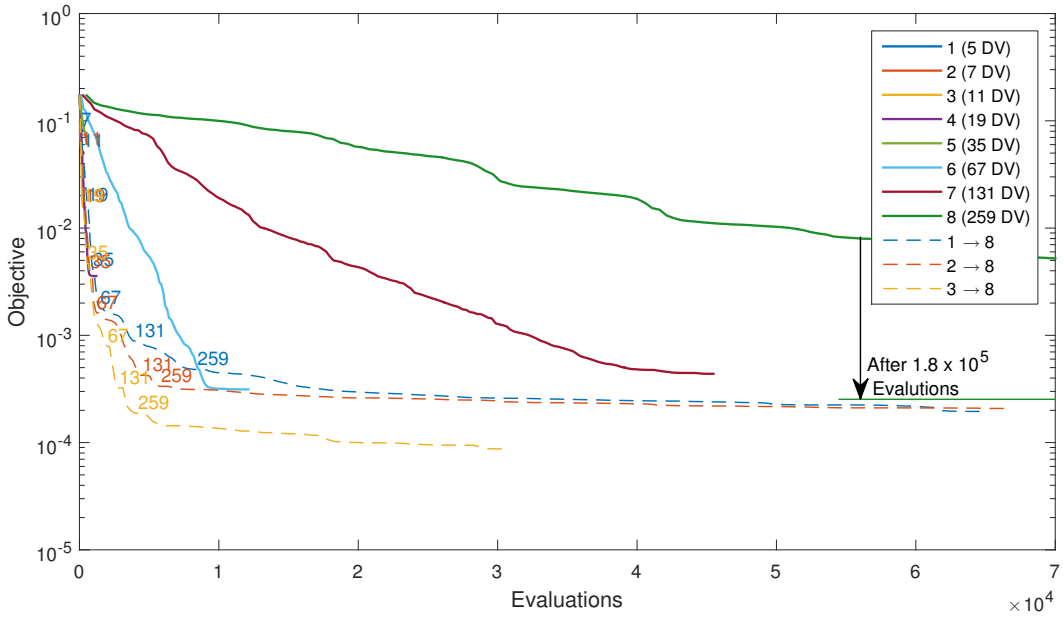
Figure 7. Results for the inverse design from a NACA0012 to a NACA4410 using VGK at $\alpha = 0$, $M = 0.7$. Numbers represent the number of design variables after a refinement.



a) Initial and target pressure distributions

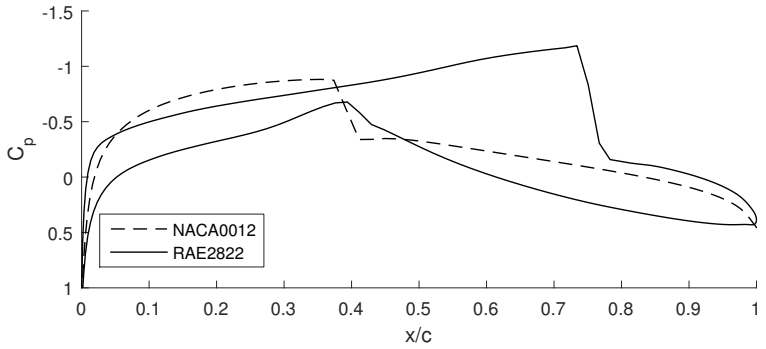


b) Convergence of the objective function w.r.t. the number of iterations

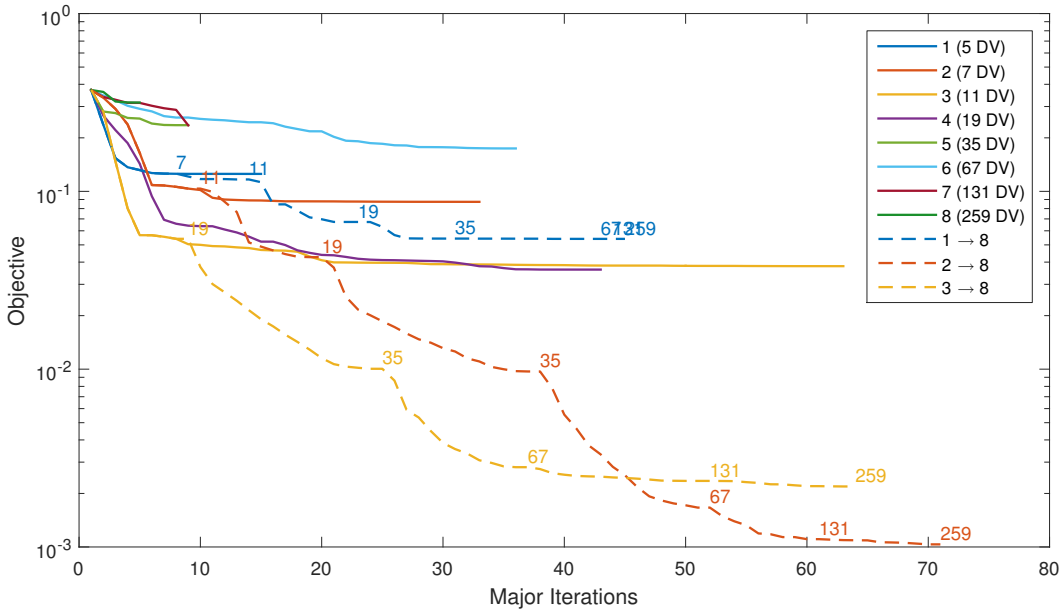


c) Convergence of the objective function w.r.t. the number of solver evaluations

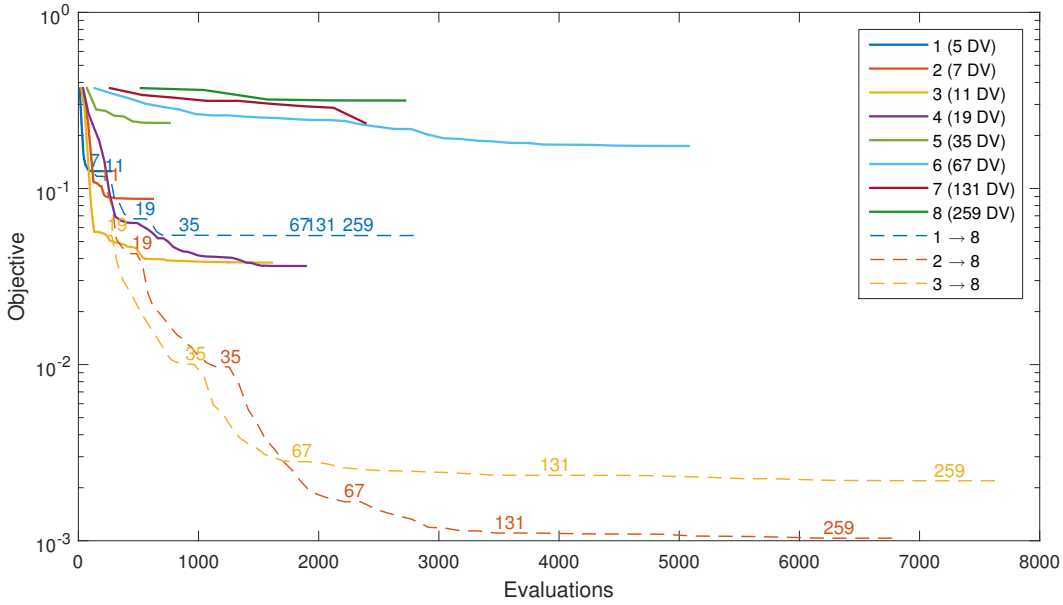
Figure 8. Results for the inverse design from a NACA0012 to a RAE2822 using a potential flow solver at $\alpha = 0$. Numbers represent the number of design variables after a refinement.



a) Initial and target pressure distributions



b) Convergence of the objective function w.r.t. the number of iterations



c) Convergence of the objective function w.r.t. the number of solver evaluations

Figure 9. Results for the inverse design from a NACA0012 to a RAE2822 using VGK at $\alpha = 0$, $M = 0.78$. Numbers represent the number of design variables after a refinement.

corresponding initial and target pressure distributions are shown in figure 10a. Similar results are shown for the static schemes as for the other cases, with improving results to an optimum level then dropping off; in this case the fifth static level gives the best final result. The three progressive results then all show similar behaviour, improving at a rate of convergence comparable to the static results though converging to results an order lower than the best static case.

Similar conclusions can be drawn for the static schemes for the VGK case (figures 11a-11c), though this time the best result is achieved at level four. All three progressive cases exceed the results of the static schemes though show some differences, with the case starting from the second level obtaining a result 10 times better than the other two.

Table 2 shows a summary of all the inverse results obtained with the best results for the static and progressive methods highlighted. It can be seen that for every case the best progressive method is better than the best static case and for all but the 'NACA0012 → RAE2822' case with VGK all three progressive cases are better. It can also be seen that, for the static methods, there is a high variability in the quality of the results. For example the fifth level static method gives the best results for two of the tests but poor results for the other four. There was also some instances of the progressive methods converging to different solutions, indicating that the global optimum was not found by all three cases, the '2 → 8' option was however consistently the best, or close to the best, result achieved. In general, however, these results highlight the significant improvement in robustness achieved by using the progressive parameterisation as well as the improved exploitation of the higher dimensional design spaces.

	NACA4410		RAE2822		ONERA M6	
	Potential	VGK	Potential	VGK	Potential	VGK
Static Level 1 (5 DV)	3.75E-02	5.59E-02	8.22E-02	1.25E-01	8.22E-02	1.29E-01
Static Level 2 (7 DV)	5.60E-03	8.05E-03	5.14E-02	8.73E-02	5.79E-02	1.09E-01
Static Level 3 (11 DV)	3.23E-03	4.36E-03	1.56E-02	3.79E-02	2.79E-02	2.66E-02
Static Level 4 (19 DV)	1.73E-03	2.47E-01	3.57E-03	3.63E-02	1.56E-02	2.59E-02
Static Level 5 (35 DV)	7.05E-04	2.49E-01	6.09E-02	2.36E-01	1.35E-03	1.95E-01
Static Level 6 (67 DV)	5.66E-02	2.74E-01	3.14E-04	1.74E-01	5.87E-02	1.71E-01
Static Level 7 (131 DV)	3.94E-02	3.35E-01	4.38E-04	2.35E-01	2.21E-01	1.82E-01
Static Level 8 (259 DV)	8.27E-03	3.72E-01	2.53E-04	3.16E-01	2.83E-01	2.35E-01
Progressive 1 → 8	8.22E-05	2.98E-04	1.95E-04	5.40E-02	1.80E-04	4.29E-03
Progressive 2 → 8	8.87E-05	3.25E-04	2.08E-04	1.03E-03	2.07E-04	5.80E-04
Progressive 3 → 8	7.48E-05	3.07E-04	8.74E-05	2.19E-03	2.63E-04	4.86E-03

Table 2. Summary of inverse design final results. Best results for the static and progressive methods are highlighted.

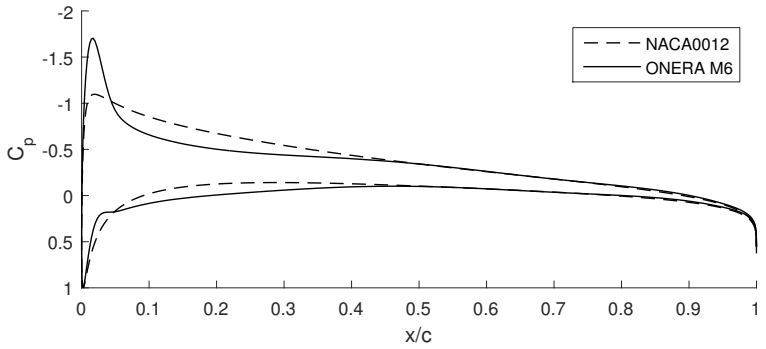
B. Drag Reduction

Drag reduction optimisations typically present a much more difficult problem than inverse design procedures. This is due to the highly non-linear relationship between the aerofoil surface and the drag as well as the range of constraints that are commonly enforced. Three drag reduction optimisations have also been investigated in this work, a non-symmetric inviscid drag reduction on a NACA0012, a viscous drag reduction on a RAE2822 and a symmetric inviscid drag reduction on a NACA0012. For each of these cases the leading edge control point has also been held fixed in addition to the trailing edge point; this is to ensure a constant angle of attack.

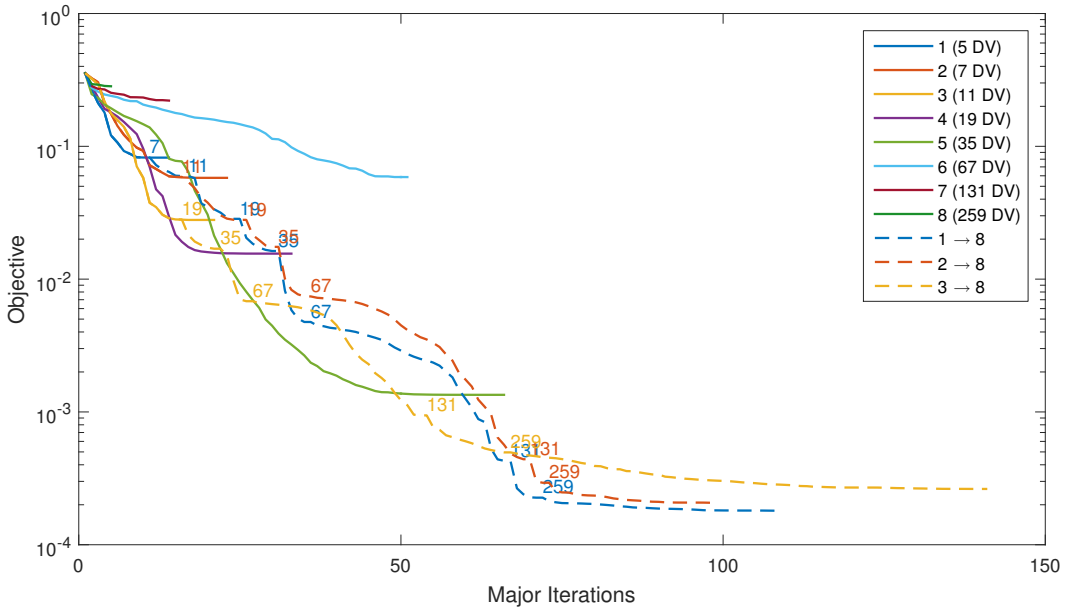
1. NACA0012 Non-Symmetric Inviscid Drag Reduction (VGK)

TEST CONFIGURATION

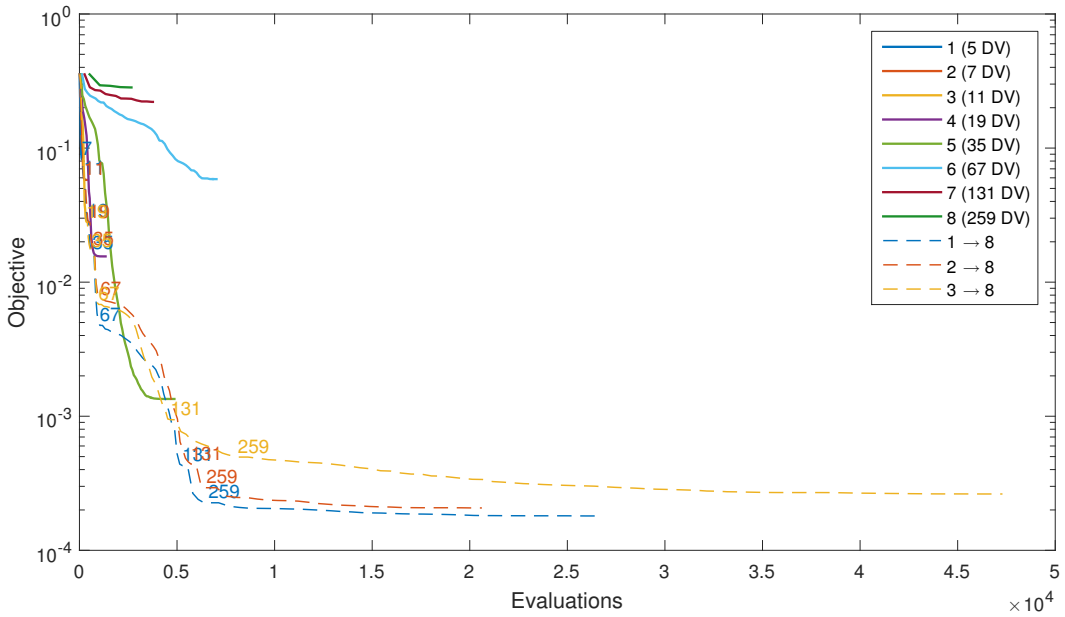
Initial Aerofoil:	*NACA0012*,
Flow Conitions:	$\alpha = 0.5$,
	$M = 0.85$,
Minimise:	C_D,
Subject to:	$C_L \geq C_{Lorig}$ $(= 0.206)$,
	$A = A_{orig}$,
Refinement Parameters:	$t = 0.1, \quad w = 3, \quad m = 1.$



a) Initial and target pressure distributions

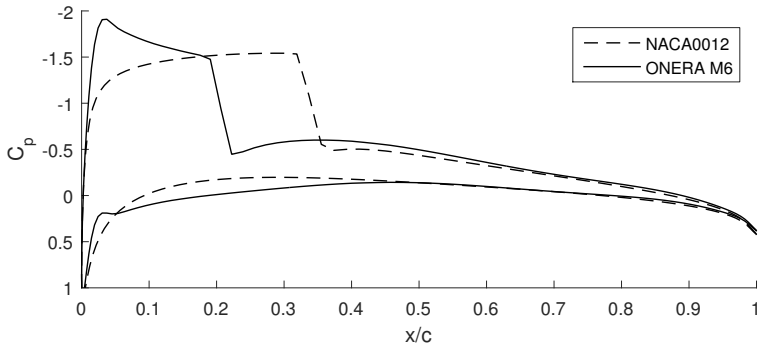


b) Convergence of the objective function w.r.t. the number of iterations

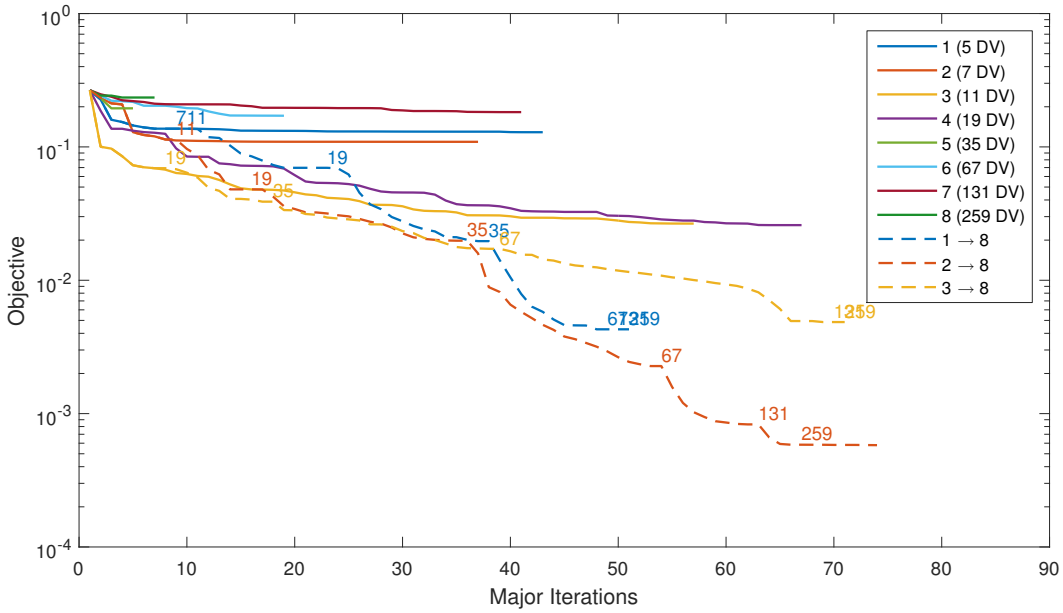


c) Convergence of the objective function w.r.t. the number of solver evaluations

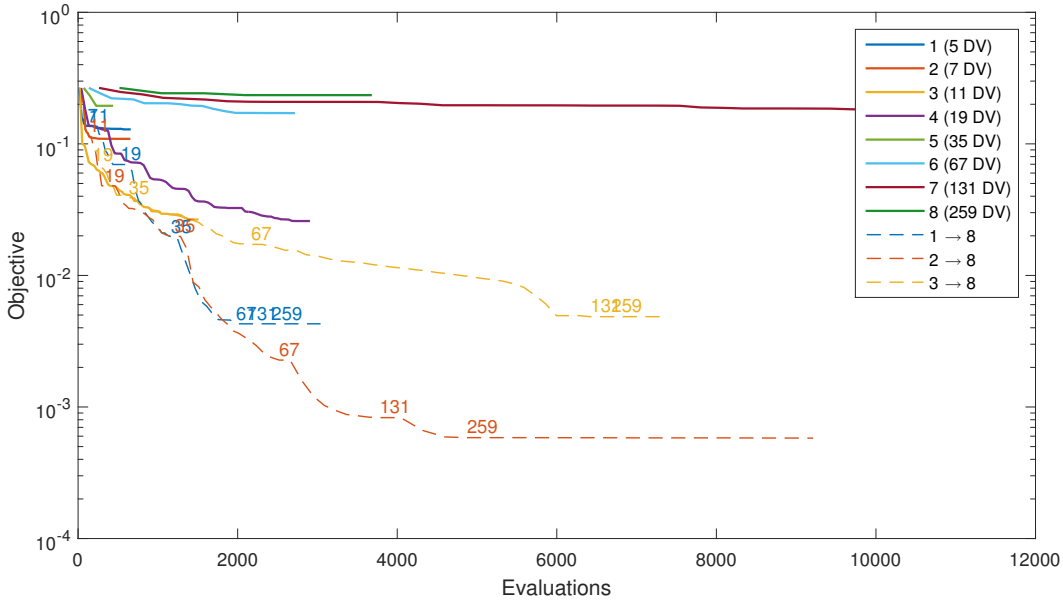
Figure 10. Results for the inverse design from a NACA0012 to an ONERA M6 using a potential flow solver at $\alpha = 3$. Numbers represent the number of design variables after a refinement.



a) Initial and target pressure distributions



b) Convergence of the objective function w.r.t. the number of iterations



c) Convergence of the objective function w.r.t. the number of solver evaluations

Figure 11. Results for the inverse design from a NACA0012 to an ONERA M6 using VGK at $\alpha = 3$, $M = 0.7$. Numbers represent the number of design variables after a refinement.

This first optimisation case minimises the drag for a NACA0012 at a low angle of attack under inviscid, transonic flow conditions. This has been done under the constraints that the lift coefficient, C_L, should not be reduced below its original value and the total area, A, should be held constant. These have been enforced through the inclusion of penalties in the objective function. This problem has therefore been reduced to the unconstrained minimisation of the objective function

$$J = C_d + 0.1\beta_{C_L} + 0.1\beta_{area}, \tag{21}$$

where

$$\beta_{C_L} = \begin{cases} C_{Lorig} - C_L & if \quad C_L < C_{Lorig}, \\ 0 & otherwise, \end{cases} \tag{22}$$

$$\beta_{area} = \frac{|A_{orig} - A|}{A_{orig}}. \tag{23}$$

Figures 12b and 12c show the convergence history with respect to the major iterations and function evaluations. For the static results a similar trend can be seen as for the inverse design cases with the results improving up to a best case (level three here), then declining for further increases in fidelity. The progressive results starting from the first and second levels give the best overall results and both show improvement at each refinement level up to the sixth, 66 design variable increment, after which no further gains are made. Interestingly the third level progressive scheme performs worse than the third level static scheme. This implies that the refinement must have been triggered at a non optimal iteration; it would appear that it was done too early.

Selected final aerofoil and pressure distribution plots are shown in figure 12a. They show that all of the cases try to reduce the strength of the upper and lower surface shocks by increasing the thickness at the rear section and the best results coincide with those with the largest boat-tail angle.

2. *RAE2882 Viscous Drag Reduction (VGK)*

TEST CONFIGURATION

Initial Aerofoil:	*RAE2822*,
Flow Conitions:	$\alpha = 2.78$,
	$M = 0.734$,
	$Re = 6.5 \times 10^6$,
Minimise:	C_D,
Subject to:	$C_L = C_{Lorig}$ $(= 0.8238)$,
	$C_M \geq C_{Morig}$ $(= -0.0945)$,
	$A \geq A_{orig}$,
Refinement Parameters:	$t = 0.1, \quad w = 3, \quad m = 3.$

The second optimisation is a slight variation on a common test case specified by the Aerodynamic Design Optimisation Discussion Group[45]. The case looks to minimise C_D at $\alpha = 2.78$, $M = 0.734$ and $Re = 6.5{\times}10^6$ under the constraints that $C_L = C_{Lorig}$, $C_M > C_{Morig}$ and $A \geq A_{orig}$. These constraints have been enforced using penalty functions so the optimisation is therefore reduced to the unconstrained minimisation of

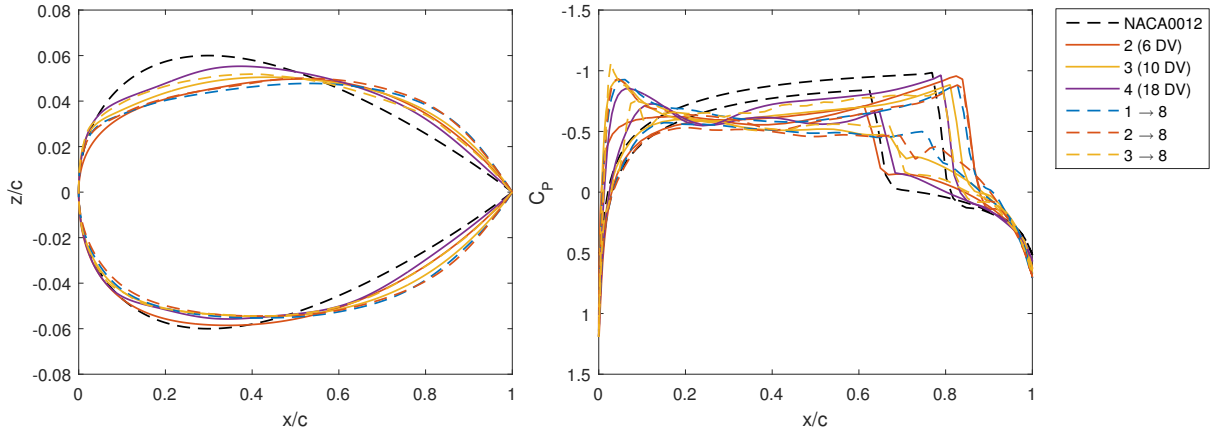
$$J = C_d + 0.1\beta_{C_L} + 0.1\beta_{area} + 0.1\beta_{C_M}, \tag{24}$$

where

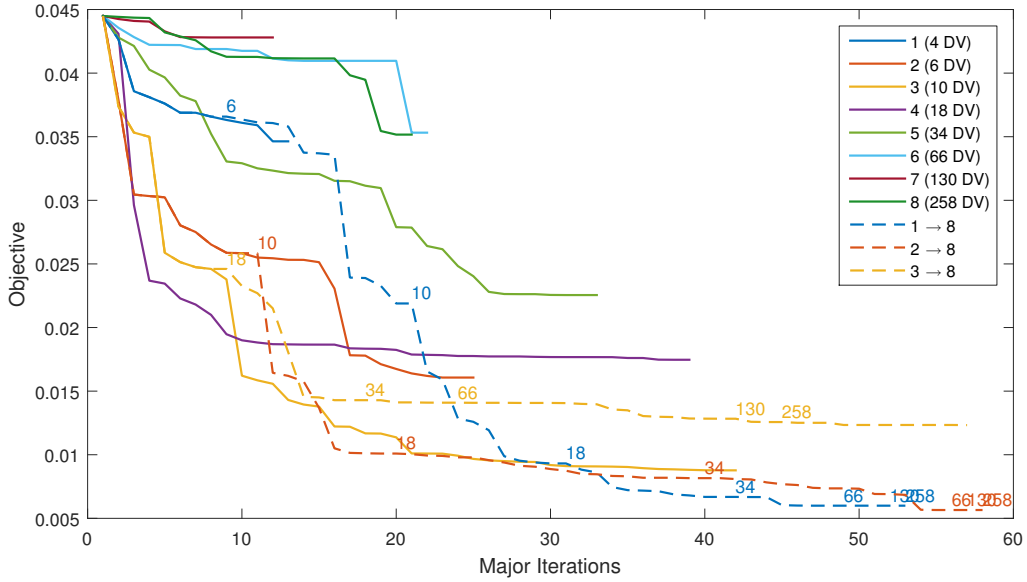
$$\beta_{C_L} = |C_{Lorig} - C_L|, \tag{25}$$

$$\beta_{area} = \begin{cases} \dfrac{A_{orig} - A}{A_{orig}} & if \quad A < A_{orig}, \\ 0 & otherwise, \end{cases} \tag{26}$$

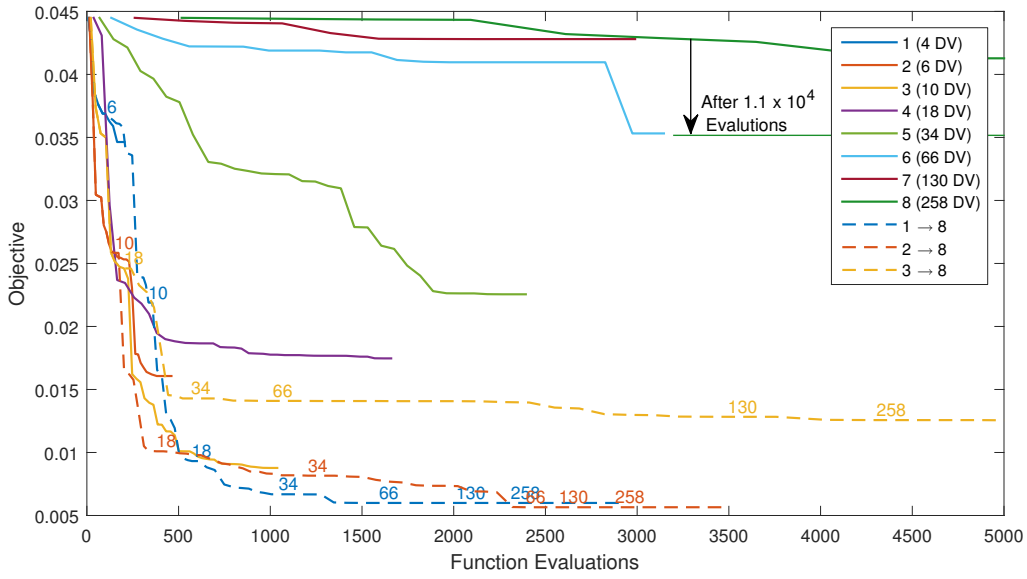
$$\beta_{C_M} = \begin{cases} C_{Morig} - C_M & if \quad C_M < C_{Morig}, \\ 0 & otherwise. \end{cases} \tag{27}$$



a) Optimum aerofoil shapes and corresponding pressure distributions for selected results



b) Convergence of the objective function w.r.t. the number of iterations



c) Convergence of the objective function w.r.t. the number of solver evaluations

Figure 12. Results for the inviscid optimisation of a NACA0012 at $\alpha = 0.5$, $M = 0.85$. Numbers represent the number of design variables after a refinement.

Figure 13b shows that the second, third and fourth static levels all achieve good results with the second level attaining the best for the non-progressive configurations. Additionally this is only narrowly worse than the best progressive result which starts from the same second level configuration. For the progressive methods all three schemes achieve reasonable final results. The convergence of the '1 → 8' progressive method at the first level is however quite slow which greatly increases the number of iterations required to obtain a good answer. Though due to the small relative cost of these iterations this does not influence its convergence with respect to the number of flow evaluations (figure 13c) as significantly.

The final aerofoils and pressure coefficients of the six best results are shown in figure 13a. It can be seen that for all of the cases the shock has been reduced or removed by increasing the upper surface thickness after the initial shock position.

A summary of the results for the first two optimisation cases is shown in table 3. For the static results they show that the lower design dimensional methods perform best with the second and third levels giving the best result. They also highlights the general difficulty in getting the higher design variable methods to converge to a good solution. For the progressive methods, they show that the '2 → 8' method performs best for both optimisations followed by the '1 → 8' and '3 → 8' methods. It should also be noted that even for the worst, '3 → 8', progressive scheme, it is only beaten by one static method in each optimisation case. In general these results again emphasises the benefits in robustness achieved through using the progressive parameterisation.

	Non- Symmetric NACA0012	Viscous RAE2822
Static Level 1 (4 DV)	3.46E-02	1.66E-02
Static Level 2 (6 DV)	1.61E-02	9.80E-03
Static Level 3 (10 DV)	8.77E-03	1.08E-02
Static Level 4 (18 DV)	1.75E-02	1.11E-02
Static Level 5 (34 DV)	2.26E-02	1.11E-02
Static Level 6 (66 DV)	3.53E-02	1.17E-02
Static Level 7 (130 DV)	4.28E-02	1.78E-02
Static Level 8 (258 DV)	3.52E-02	1.78E-02
Progressive 1 → 8	6.00E-03	9.81E-03
Progressive 2 → 8	5.65E-03	9.79E-03
Progressive 3 → 8	1.23E-02	1.08E-02

Table 3. Summary of final objective function results for the VGK drag optimisations. Best results for the static and progressive methods are highlighted.

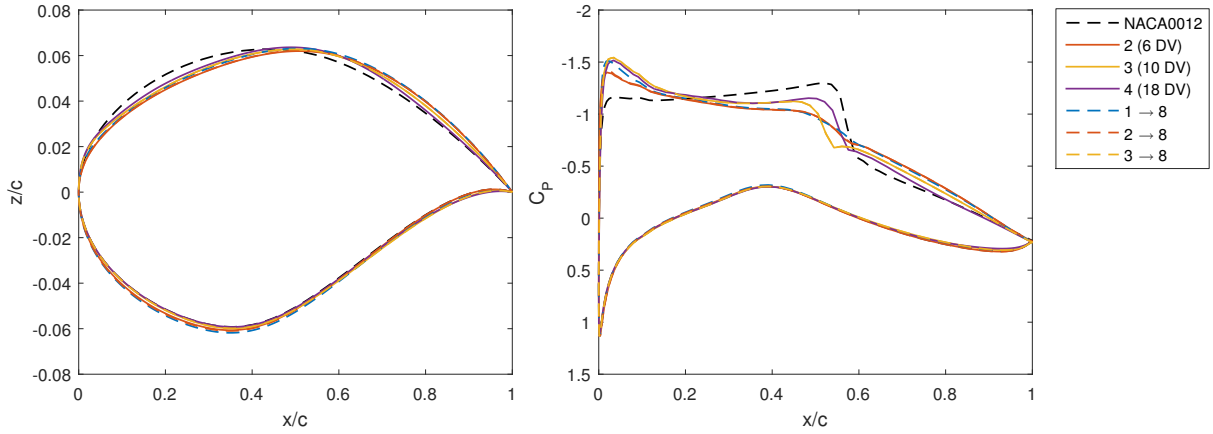
3. NACA0012 Symmetric Inviscid Drag Reduction (Euler)

TEST CONFIGURATION

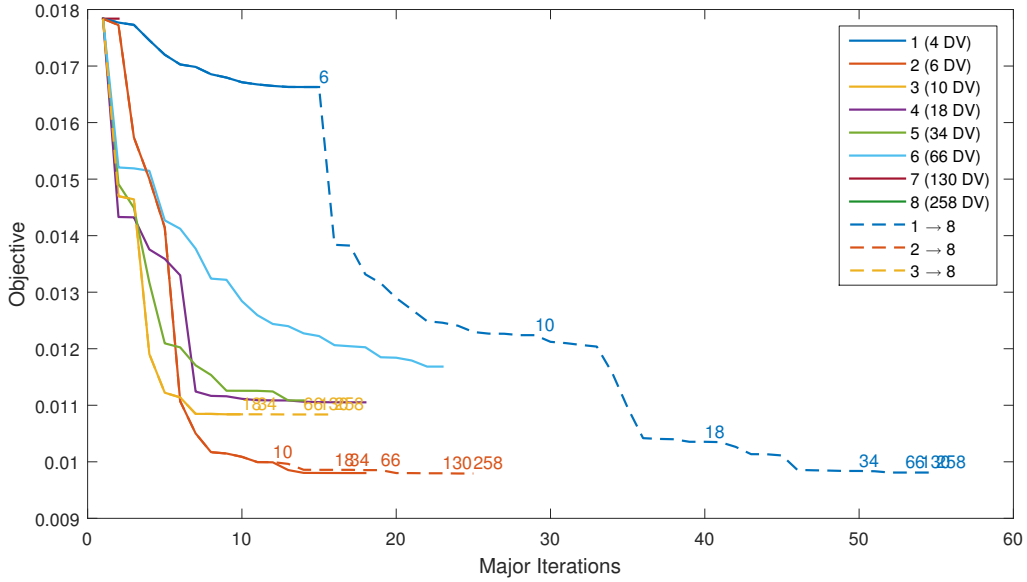
$$
\begin{aligned}
\text{Initial Aerofoil:} \quad & \textit{NACA0012}, \\
\text{Flow Conitions:} \quad & \alpha = 0, \\
& M = 0.85, \\
\text{Minimise:} \quad & C_D, \\
\text{Subject to:} \quad & C_L = 0, \\
& y \geq y_{NACA0012} \quad \forall x \in [0, 1], \\
\text{Refinement Parameters:} \quad & t = 0.1, \quad w = 3, \quad m = 3.
\end{aligned}
$$

The final optimisation is a further test case outline by the Aerodynamic Design Optimisation Discussion Group[45]. This is the inviscid drag reduction of a NACA0012 at $M = 0.85$ and $\alpha = 0$. A considerable number of papers have been published on this case[1, 2, 46, 47, 48, 49, 50, 51, 52, 53, 54] and it has been shown to be a particularly difficult problem to optimise due to a range of characteristics such as multiple local minima[52], non-symmetric solutions[53] and hysteresis[54].

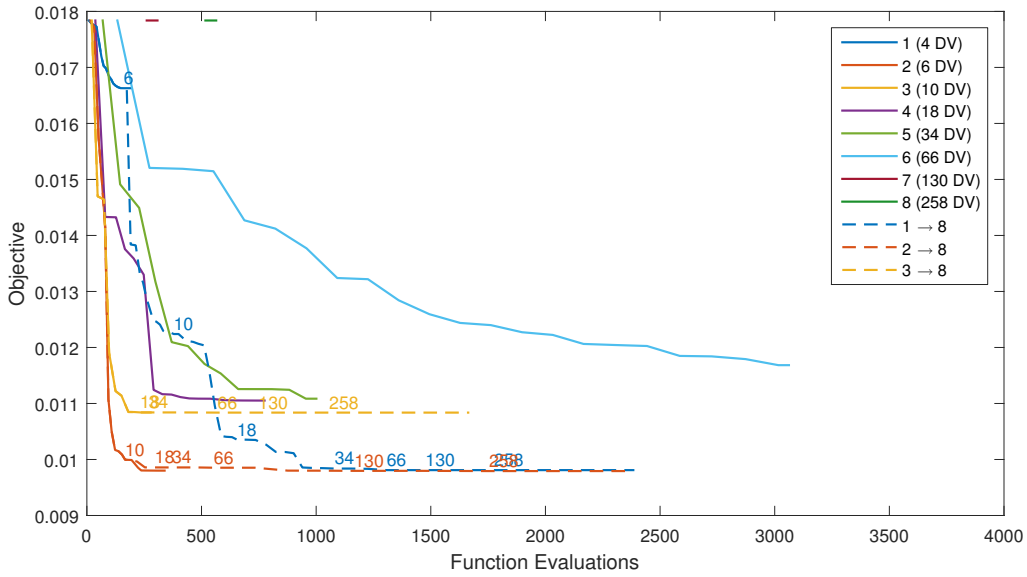
These tests have been performed with the unstructured CFD solver SU^2 with the local thickness constraint applied as linear constraints in SNOPT at every fifth chord-wise percentile. To calculate the design variable gradients the



a) Optimum aerofoil shapes and corresponding pressure distributions for selected results



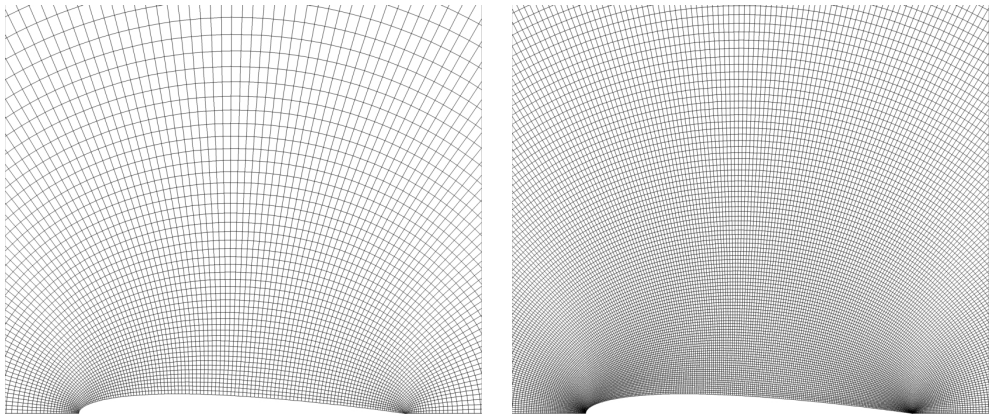
b) Convergence of the objective function w.r.t. the number of iterations



c) Convergence of the objective function w.r.t. the number of solver evaluations

Figure 13. Results for the viscous optimisation of an RAE2822 at $\alpha = 2.78$, $M = 0.734$, $Re = 6.5 \times 10^6$. Numbers represent the number of design variables after a refinement.

adjoint equations[55] are solved to calculate the surface sensitivities which are then projected, in the z direction, on to the design variables. The benefit of this method is that the calculation of the gradients is independent of the number of design variables used. Due to the symmetry of the problem a symmetry plane was aligned with the aerofoil chord and just half the mesh was solved. This meant that only half of the aerofoil surface, and therefore only half of the subdivision surface, needed to be modelled. For this reason each subdivision level has half the number of control points and design variables used in the other optimisation cases. The full optimisation study was run on two different mesh sizes, a 129×129 coarse mesh and a 257×257 fine mesh (equivalent to 257 and 513 points around the full aerofoil, respectively). These are shown in figure 14.



a) 129×129 **Coarse Half-Mesh** **b)** 257×257 **Fine Half-Mesh**

Figure 14. Two mesh resolutions used for the inviscid, symmetric NACA0012 optimisation.

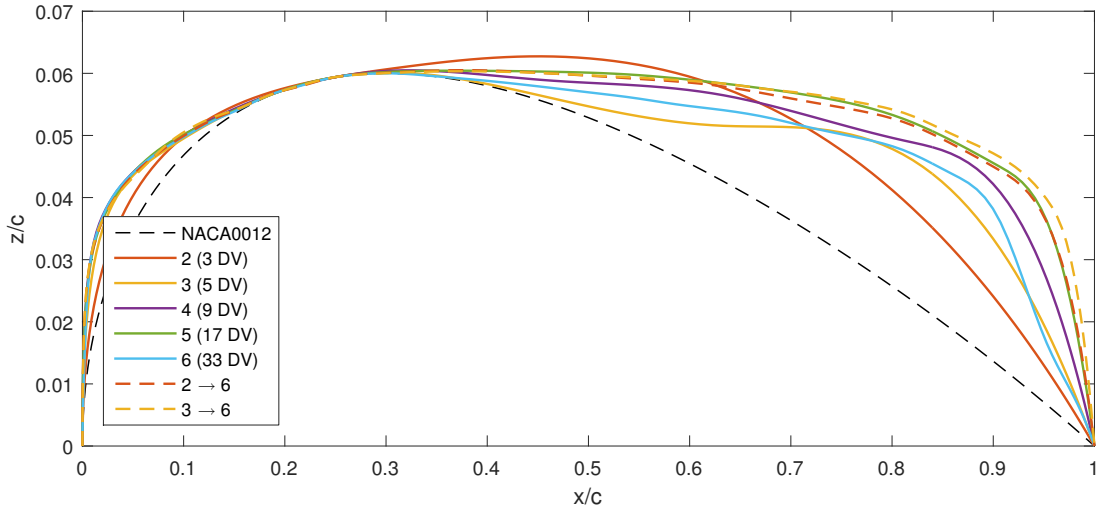
For both mesh sizes no improvement could be made at subdivision level 1 (with 2 design variables). For this reason only levels two and above are included in the results for this test case. As well as this the maximum level used will be level six for the coarse mesh and level seven for the fine.

The convergence history for the coarse mesh optimisations can be seen in figure 15c. It shows that, for the static methods, the results improve consistently up to a level 5 result of 54 drag counts. The '$2 \rightarrow 6$' and '$3 \rightarrow 6$' progressive methods show a slight improvement on that result with final drag values of 51 and 49 counts, respectively. The '$3 \rightarrow 6$' case also shows an improved convergence path, dropping to 106 counts after 10 iterations relative to 180 counts for the fifth level static case at the equivalent point. The optimum surfaces and pressure distributions are then shown in figures 15a and 15b. This shows that the best results are those with the largest boat-tail angle and trailing edge thickness profile, in line with other results for this problem[54].

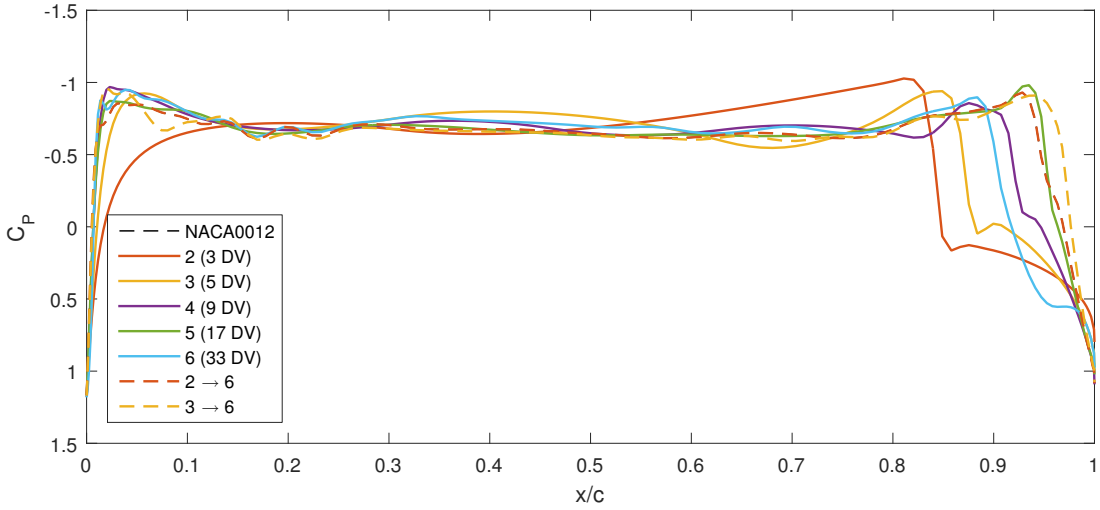
The convergence results for the fine, 257×257 case are then shown in figure 16c. Again, for the static results, the fifth level shows the best final answer, converging to a result of 36 drag counts. The '$2 \rightarrow 7$' and '$3 \rightarrow 7$' progressive methods then converge to results of 39 and 42 counts. However, neither make any improvement at the final, seventh, level, this indicates that refinement may have been triggered too early. For this reason an additional two cases for levels '$2 \rightarrow 5$' and '$3 \rightarrow 5$' were run. They both converged to improved final results of 36 drag counts. This confirms that the refinement must have been triggered too early for the cases refining up to level 7. The optimum shapes and pressure distributions for each method are then shown in figures 16a and 16b. These figures show that the level 5 static, and four progressive methods converged to very similar shapes.

Tables 4 and 5 show a summary of the results for the two symmetric NACA0012 optimisations. For both of these cases the fifth static level and the various progressive methods give very similar results. It should however be noted that only one of the static methods achieved these result in each case yet all of the progressive methods did. This again highlights the benefits in robustness gained through using the progressive schemes.

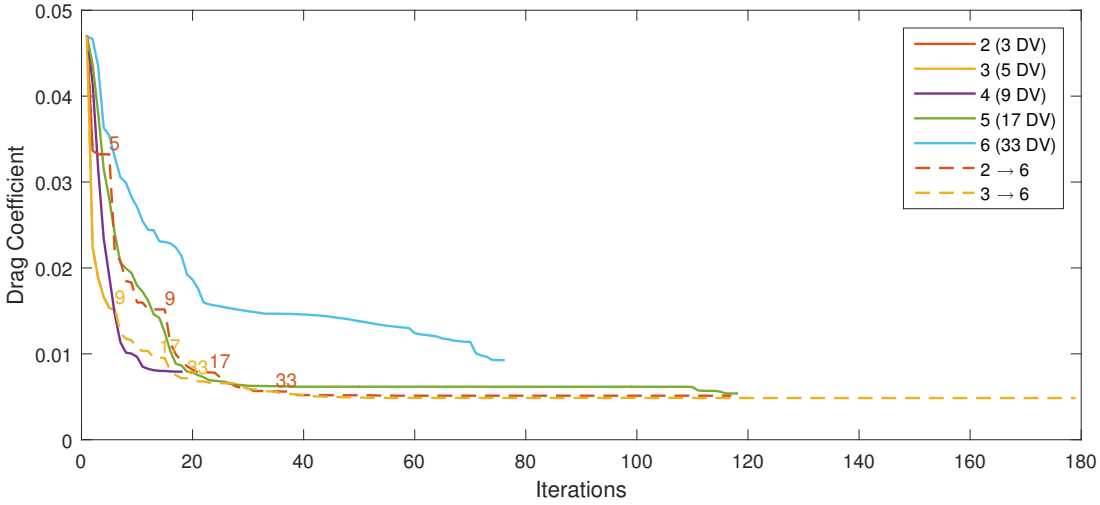
A comparison of the best static and progressive methods for each optimisation case is shown in table 6. For every case investigated it can be seen that the progressive methods have exceeded or matched the results of the static methods. In addition to this it shows that, for the inverse design cases, they, on average, improved on the best static result by 88.3% and for the optimisation cases reduced the drag coefficient by a further 11.4%. This emphasis the general improvements in final optimisation results attained using the progressive methods.



a) Optimum aerofoil shapes

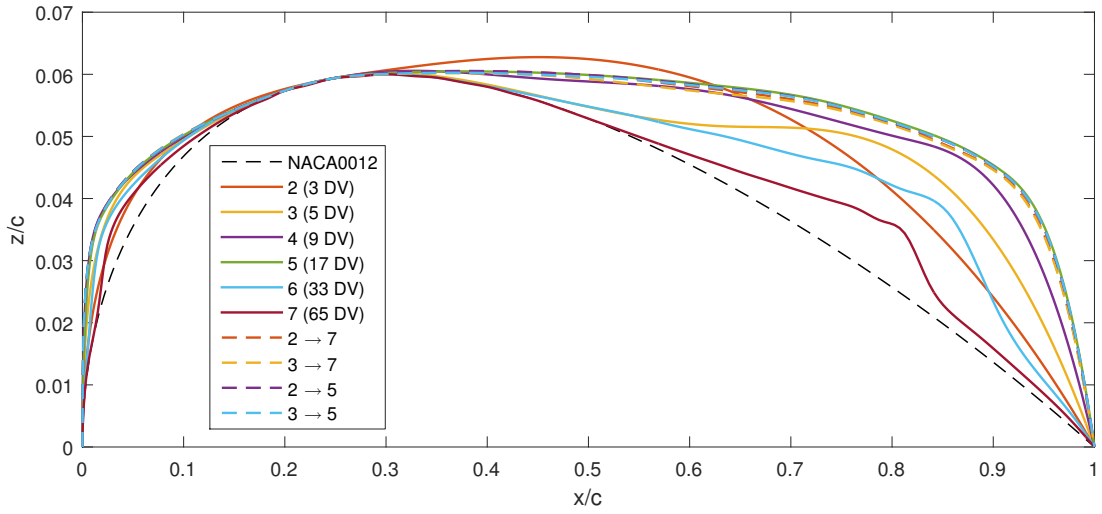


b) Pressure distributions for optimum aerofoil shapes

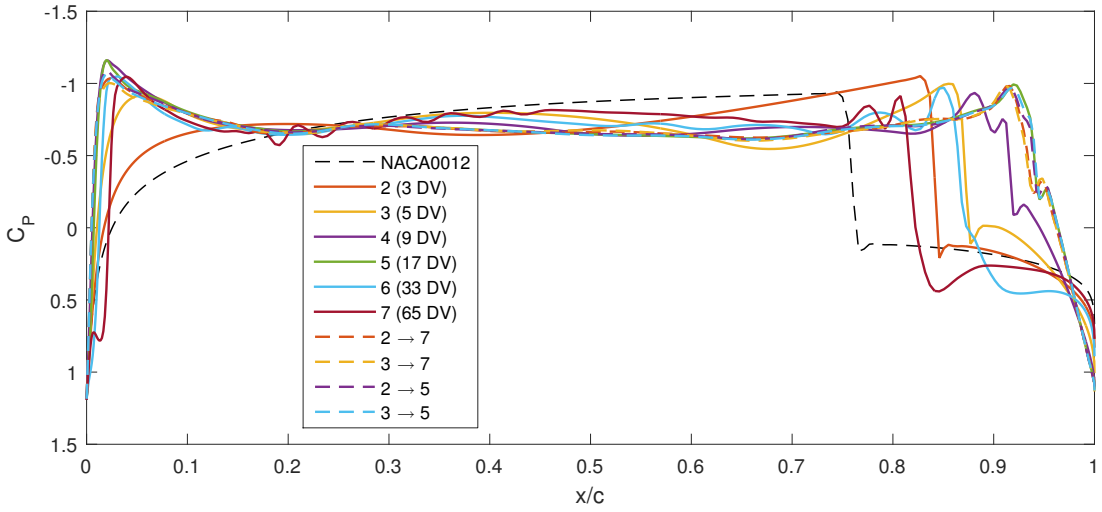


c) Convergence of the drag coefficient w.r.t. the number of iterations

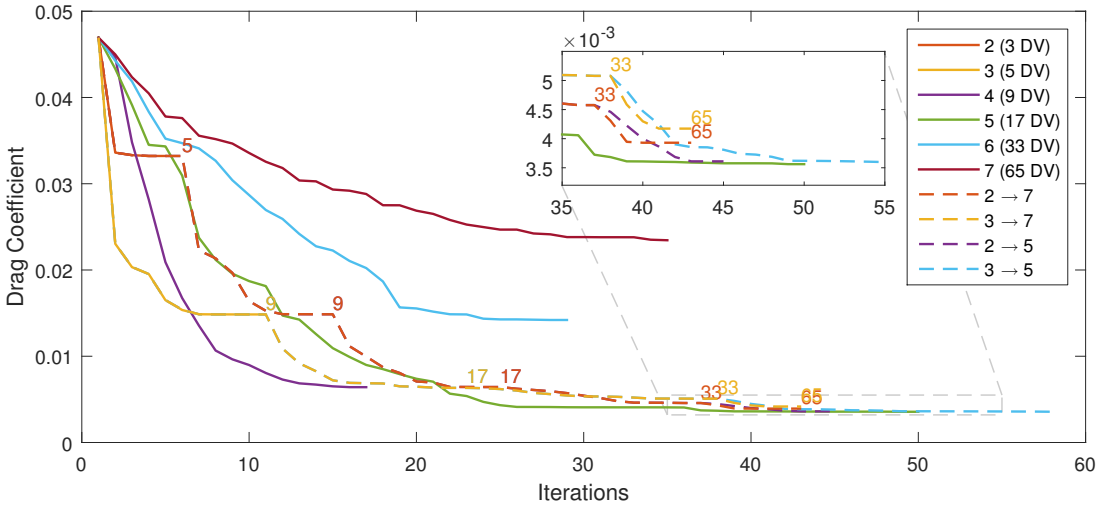
Figure 15. Results for the inviscid optimisation of a NACA0012 at $\alpha = 0$ and $M = 0.85$ run on the coarse 129×129 half-mesh.



a) Optimum aerofoil shapes



b) Pressure distributions for optimum aerofoil shapes



c) Convergence of the drag coefficient w.r.t. the number of iterations

Figure 16. Results for the inviscid optimisation of a NACA0012 at $\alpha = 0$ and $M = 0.85$ run on the fine 257×257 half-mesh.

	Symmetric NACA0012 (129×129 Coarse Mesh)
Level 2 (3 DV)	3.32E-02
Level 3 (5 DV)	1.51E-02
Level 4 (9 DV)	7.92E-03
Level 5 (17 DV)	5.39E-03
Level 6 (33 DV)	9.27E-03
Progressive 2→6	5.13E-03
Progressive 3→6	4.85E-03

Table 4. Table of results for the inviscid optimisation of a NACA0012 at $\alpha = 0$ and $M = 0.85$ run on the coarse 129×129 half-mesh.

	Symmetric NACA0012 (257×257 Fine Mesh)
Level 2 (3 DV)	3.32E-02
Level 3 (5 DV)	1.48E-02
Level 4 (9 DV)	6.41E-03
Level 5 (17 DV)	3.56E-03
Level 6 (33 DV)	1.42E-02
Level 7 (65 DV)	2.34E-02
Progressive 2→7	3.93E-03
Progressive 3→7	4.17E-03
Progressive 2→5	3.61E-03
Progressive 3→5	3.56E-03

Table 5. Table of results for the inviscid optimisation of a NACA0012 at $\alpha = 0$ and $M = 0.85$ run on the fine 257×257 half-mesh.

		Best Static		Best Progressive		Relative
Optimisation Case	Solver	Case	Obj. Fun	Case	Obj. Fun	Difference
NACA0012 → NACA4410	Potential Flow	Level 5	7.05E-04	3→8	7.48E-05	−89.4%
NACA0012 → NACA4410	VGK	Level 3	4.36E-03	1→8	2.98E-04	−93.2%
NACA0012 → RAE2822	Potential Flow	Level 8	2.53E-04	3→8	8.74E-05	−65.5%
NACA0012 → RAE2822	VGK	Level 4	3.63E-02	2→8	1.03E-03	−97.2%
NACA0012 → ONERAM6	Potential Flow	Level 5	1.35E-03	1→8	1.80E-04	−86.7%
NACA0012 → ONERAM6	VGK	Level 4	2.59E-02	2→8	5.80E-04	−97.8%
					Mean	*−88.3%*
Non-symmetric NACA0012	VGK	Level 3	8.77E-03	2→8	5.65E-03	−35.6%
RAE2822	VGK	Level 2	9.80E-03	2→8	9.79E-03	−0.1%
Symmetric NACA0012	Euler (Coarse Mesh)	Level 5	5.39E-03	3→6	4.85E-03	−10.0%
Symmetric NACA0012	Euler (Fine Mesh)	Level 5	3.56E-03	3→5	3.56E-03	0.0%
					Mean	*−11.4%*

Table 6. Table showing the best results attained for the static and progressive parameterisation schemes for each optimisation case as well as the relative difference between them.

VII. Conclusions

In this work a progressive subdivision parameterisation scheme has been tested on a series of optimisation problems. They have been compared to a range of static subdivision parameterisations that are equivalent to cubic B-Splines. Ten optimisation problems of varying complexity have been used; three inverse design tests, each performed with a potential flow solver and the full potential boundary-layer coupled solver VGK, and four drag reduction optimisations, two with VGK and two with an Euler solver.

For the inverse design problems the progressive schemes obtained significantly better results than any of the static, control tests. On average the best progressive result achieved a 88% improvement on the best static case. This was because the progressive nature of the parameterisations allowed the high-dimensional design space to be successfully exploited in contrast to the static methods that would often stagnate. As well as this the best static method was different for each optimisation case and no single level gave good results for all of them. For the progressive methods however, all three methods gave consistently good results with the '2 → 8' case giving the best overall performance.

For the all four drag optimisation cases the best progressive methods improved on or matched the best static methods with an average improvement of 11% in drag reduction. The robustness of the progressive schemes was again shown, with all methods achieving reasonable results for all cases. For the static methods the best result was often comparable to the progressive schemes however the level which achieved it was usually unpredictable.

The convergence path of the optimisations was also often improved by using the progressive methods, as the low dimensional space used in the early stages of the optimisation was easier to navigate and therefore accelerated the

process. This also provided benefits computationally if the gradients were calculated using finite differences.

In summary the use of progressive subdivision surfaces has been shown to significantly improve the robustness of an optimisation process by allowing predictable lower dimensional design spaces to be utilised in the early optimisation stages and higher dimensional design spaces in the latter. This also lead to improvements in the final optimisation results through improved exploitation of high-dimensional design spaces as well as improvements to convergence rates and computational efficiency.

VIII. Acknowledgements

This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol - http://www.bris.ac.uk/acrc/. The authors also wish to acknowledge the financial support provided by Innovate UK: the work reported herein has been undertaken in GHandI (TSB 101372), a UK Centre for Aerodynamics project.

References

[1] Carrier, G., Destarac, D., Dumont, A., Meheut, M., Salah El Din, I., Peter, J., Ben Khelil, S., Brezillon, J., and Pestana, M., "Gradient-based aerodynamic optimization with the elsA software," *52nd Aerospace Sciences Meeting*, Jan 2014, 10.2514/6.2014-0568.

[2] Poole, D. J., Allen, C. B., and Rendall, T. C. S., "Application of control point-based aerodynamic shape optimization to two-dimensional drag minimization," *52nd Aerospace Sciences Meeting*, 2014.

[3] Masters, D. A., Taylor, N. J., Rendall, T. C. S., Allen, C. B., and Poole, D. J., "Review of Aerofoil Parameterisation Methods for Aerodynamic Shape Optimisation," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[4] Braibant, V. and Fleury, C., "Shape optimal design using B-splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44(3), 1984, pp. 247–267.

[5] Lépine, J., Trépanier, J.-Y., and Pépin, F., "Wing aerodynamic design using an optimized NURBS geometrical representation," *The 38th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA–2000–699, Reno, Nevada, 2000.

[6] Kulfan, B. M. and Bussoletti, J. E., "Fundamental parametric geometry representations for aircraft component shapes," *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, September 2006.

[7] Kulfan, B. M., "A universal parametric geometry representation method-CST," *45th AIAA Aerospace Sciences Meeting and Exhibit*, January 2007.

[8] Hicks, R. M. and Henne, P. A., "Wing design by numerical optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[9] Wu, H.-Y., Yang, S., Liu, F., and Tsai, H.-M., "Comparison of three geometric representations of airfoils for aerodynamic optimization," *16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida*, 2003.

[10] Sobieczky, H., "Parametric airfoils and wings," *Recent Development of Aerodynamic Design Methodologies*, Springer, 1999, pp. 71–87.

[11] Morris, A. M., Allen, C. B., and Rendall, T. C. S., "CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation," *International Journal for Numerical Methods in Fluids*, Vol. 58, No. 8, Nov 2008, pp. 827–860.

[12] Morris, A. M., Allen, C. B., and Rendall, T. C. S., "Aerodynamic shape optimization of a modern transport wing using only planform variations," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 223, No. 6, 2009, pp. 843–851.

[13] Allen, C. B. and Rendall, T. C. S., "CFD-based optimization of hovering rotors using radial basis functions for shape parameterization and mesh deformation," *Optimization and Engineering*, Vol. 14, No. 1, Mar 2013, pp. 97–118.

[14] Chauhan, D., Praveen, C., and Duvigneau, R., "Wing shape optimization using FFD and twist parameterization," *12th Aerospace Society of India CFD Symposium*, 2010.

[15] Poole, D. J., Allen, C. B., and Rendall, T. C. S., "Metric-based mathematical derivation of efficient aerofoil design variables," *AIAA Journal*, 2015, Published online.

[16] Toal, D. J. J., Bressloff, N. W., Keane, A. J., and Holden, C. M. E., "Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization," *AIAA Journal*, Vol. 48, 2010, pp. 916–928.

[17] Jameson, A., "Aerodynamic design via control theory," *Journal of Scientifical Computing, also ICASE Report No.88-64*, Vol. 3, 1988, pp. 233–260.

[18] Beux, F. and Dervieux, A., "A hierarchical approach for shape optimization," *Engineering Computations*, Vol. 11, No. 1, 1994, pp. 25–48.

[19] Desideri, J. and Zolesio, J., "Inverse shape optimization problems and application to airfoils," *Control and Cybernetics*, Vol. 34, No. 1, 2005, pp. 165.

[20] Desideri, J. and Dervieux, A., "Hierarchical methods for shape optimization in aerodynamics I: Multilevel algorithms for parametric shape optimization," *LECTURE SERIES-VON KARMAN INSTITUTE FOR FLUID DYNAMICS*, Vol. 3, 2006, pp. 10.

[21] Désidéri, J.-A., El Majd, B. A., and Janka, A., "Nested and self-adaptive Bézier parameterizations for shape optimization," *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 117–131.

[22] Martinelli, M. and Beux, F., "Multi-level gradient-based methods and parametrisation in aerodynamic shape design," *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, Vol. 17, No. 1-2, 2008, pp. 169–197.

[23] Andreoli, M., Ales, J., and Désidéri, J.-A., "Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics," 2003.

[24] Duvigneau, R., Chaigne, B., and Désidéri, J.-A., "Multi-level parameterization for shape optimization in aerodynamics and electromagnetics using a particle swarm optimization algorithm," 2006.

[25] Duvigneau, R., "Adaptive parameterization using free-form deformation for aerodynamic shape optimization," 2006.

[26] El Majd, B. A., Duvigneau, R., and Désidéri, J., "Aerodynamic shape optimization using a full and adaptive multilevel algorithm," *ERCOFTAC Conference Design Optimization: Methods and Applications, Canary Island, Spain*, 2006.

[27] Anderson, G. R. and Aftosmis, M. J., "Adaptive Shape Control for Aerodynamic Design," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Jan 2015.

[28] Han, X. and Zingg, D. W., "An adaptive geometry parametrization for aerodynamic shape optimization," *Optimization and Engineering*, Vol. 15, No. 1, 2014, pp. 69–91.

[29] Sherar, P., Thompson, C., Xu, B., and Zhong, B., "An Optimization Method Based On B-spline Shape Functions & the Knot Insertion Algorithm." *World congress on engineering*, Citeseer, 2007, pp. 862–866.

[30] Chaikin, G., "An algorithm for high-speed curve generation," *Graphical Models /graphical Models and Image Processing /computer Vision, Graphics, and Image Processing*, Vol. 3, 1974, pp. 346–349.

[31] Cashman, T. J., Hormann, K., and Reif, U., "Generalized Lane–Riesenfeld algorithms," *Computer Aided Geometric Design*, Vol. 30, No. 4, 2013, pp. 398–409.

[32] Doo, D. and Sabin, M., "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, Vol. 10, No. 6, 1978, pp. 356–360.

[33] Catmull, E. and Clark, J., "Recursively generated B-spline surfaces on arbitrary topological meshes," *Computer-aided design*, Vol. 10, No. 6, 1978, pp. 350–355.

[34] Forsey, D. R. and Bartels, R. H., "Hierarchical B-spline refinement," *ACM SIGGRAPH Computer Graphics*, Vol. 22, ACM, 1988, pp. 205–212.

[35] Finkelstein, A. and Salesin, D. H., "Multiresolution curves," *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, 1994, pp. 261–268.

[36] DeRose, T., Kass, M., and Truong, T., "Subdivision surfaces in character animation," *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, 1998, pp. 85–94.

[37] Ma, W., "Subdivision surfaces for CAD," *Computer-Aided Design and Applications*, Vol. 1, No. 1-4, 2004, pp. 223–232.

[38] Cashman, T. J., *NURBS-compatible subdivision surfaces*, Ph.D. thesis, Cashman, Thomas J., 2010.

[39] Quinlan, G., "Using Freestyle in PTC Creo Parametric to Create Award-Winning Designs," http://creo.ptc.com/2013/12/06/using-ptc-creo-parametric-freestyle-to-create-award-winning-designs/, December 2013, Accessed: 01-Jun-2015.

[40] Sabin, M., "Eigenanalysis and artifacts of subdivision curves and surfaces," *Tutorials on multiresolution in geometric modelling*, Springer, 2002, pp. 69–92.

[41] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM journal on optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[42] Freestone, M., "VGK method for two-dimensional aerofoil sections Part 1: principles and results," April 2004, ESDU 96028.

[43] Palacios, F., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W., and Alonso, J. J., "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design," *AIAA Paper*, Vol. 287, 2013, pp. 2013.

[44] Palacios, F., Economon, T. D., Aranake, A. C., Copeland, S. R., Lonkar, A. K., Lukaczyk, T. W., Manosalvas, D. E., Naik, K. R., Padrón, A. S., Tracey, B., et al., "Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows," *AIAA paper*, Vol. 243, 2014, pp. 13–17.

[45] Nadarajah, S., "Aerodynamic Design Optimization: Drag Minimization of the NACA 0012 in Transonic Inviscid Flow," Retreived from https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/Test%20Cases/ADODG%20Case%201%20and%202%20NACA0012%20and%20RAE%202822.pdf.

[46] Zhang, M., Rizzi, A. W., and Nangia, R. K., "Transonic Airfoils and Wings Design Using Inverse and Direct Methods," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[47] Vassberg, J., Harrison, N., Roman, D., and Jameson, A., "A Systematic Study on the Impact of Dimensionality for a Two-Dimensional Aerodynamic Optimization Model Problem," *29th AIAA Applied Aerodynamics Conference*, Jun 2011.

[48] Bisson, F. and Nadarajah, S., "Adjoint-Based Aerodynamic Optimization of Benchmark Problems," *52nd Aerospace Sciences Meeting*, Jan 2014.

[49] Nadarajah, S., "Adjoint-Based Aerodynamic Optimization of Benchmark Problems," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[50] Poole, D. J., Allen, C. B., and Rendall, T. C. S., "Control Point-Based Aerodynamic Shape Optimization Applied to AIAA ADODG Test Cases," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[51] Anderson, G. R., Nemec, M., and Aftosmis, M. J., "Aerodynamic Shape Optimization Benchmarks with Error Control and Automatic Parameterization," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[52] Masters, D. A., Taylor, N. J., Rendall, T. C. S., and Allen, C. B., "Impact of Shape Parameterisation on Aerodynamic Optimisation of Benchmark Problems," *54rd AIAA Aerospace Sciences Meeting*, Jan 2016.

[53] Lee, C., Koo, D., Telidetzki, K., Buckley, H., Gagnon, H., and Zingg, D. W., "Aerodynamic Shape Optimization of Benchmark Problems Using Jetstream," *53rd AIAA Aerospace Sciences Meeting*, 2015.

[54] Meheut, M., Destarac, D., Carrier, G., Anderson, G., Nadarajah, S., Poole, D., Vassberg, J., and Zingg, D. W., "Gradient-Based Single and Multi-points Aerodynamic Optimizations with the elsA Software," *53rd AIAA Aerospace Sciences Meeting*, Jan 2015.

[55] Economon, T. D., Palacios, F., and Alonso, J. J., "A viscous continuous adjoint approach for the design of rotating engineering applications," *AIAA Paper*, Vol. 2580, 2013, pp. 24–27.