AIAA JOURNAL
Vol. 44, No. 9, September 2006

# Airfoil Shape Design and Optimization Using Multifidelity Analysis and Embedded Inverse Design

Thomas R. Barrett,* Neil W. Bressloff,† and Andy J. Keane‡
*University of Southampton, Southampton, SO17 1BJ England, United Kingdom*

DOI: 10.2514/1.18766

**Design optimization using high-fidelity computational fluid dynamics simulations is becoming increasingly popular, sustaining the desire to make these methods more computationally efficient. A reduction in problem dimensions as a result of improved parameterization techniques is a common contributor to this efficiency. The focus of this paper is on the high-fidelity aerodynamic design of airfoil shapes. A multifidelity design search method is presented which uses a parameterization of the airfoil pressure distribution followed by inverse design, giving a reduction in the number of design variables used in optimization. Although an expensive analysis code is used in evaluating airfoil performance, computational cost is reduced by using a low-fidelity code in the inverse design process. This method is run side by side with a method which is considered to be a current benchmark in design optimization. The two methods are described in detail, and their relative performance is compared and discussed. The newly presented method is found to converge towards the optimum design significantly more quickly than the benchmark method, providing designs with greater performance for a given computational expense.**

## Nomenclature

| | | |
|---|---|---|
| $a, b$ | = | constant coefficients of polynomial expressions |
| $C_D$ | = | force coefficient in the $x$ direction (drag coefficient) |
| $C_L$ | = | force coefficient in the $z$ direction (lift coefficient) |
| $C_p$ | = | airfoil pressure coefficient |
| $C_p^k$ | = | computed pressure coefficient at design iteration $k$ |
| $C_p^T$ | = | target pressure coefficient |
| $c$ | = | airfoil chord |
| $R$ | = | relaxation factor |
| $r_{LE}$ | = | airfoil leading edge radius |
| $t$ | = | airfoil maximum thickness |
| $x$ | = | airfoil ordinate in the horizontal (chordwise) direction |
| $z$ | = | airfoil ordinate in the vertical direction |
| $z_c$ | = | airfoil maximum mean thickness (camber) |
| $z''$ | = | airfoil surface second derivative with respect to $x$ |
| $\alpha$ | = | airfoil angle of attack |

## I. Introduction

DESPITE advances in computational fluid dynamics (CFD) simulations and computing power, the expense of high-fidelity CFD means that more efficient design optimization methods are still sought after for use in aerodynamic design. The process of design has been described as the *making of decisions that change the product definition* [1]. Traditionally, within *direct* CFD based design methods these decisions involve the manipulation of a component's geometry based on the results of analysis. An inherent limitation associated with high-fidelity aerodynamic shape optimization performed in this way is the requirement for a large number of design variables to define, in sufficient detail, the geometry of the component being studied. The resulting process can be very expensive computationally. As well as designing a component via the relationship between its geometry and its overall performance, it is also possible to design by matching flow features with a set specified by the designer [2,3]. These so-called *inverse* design methods have been used widely, particularly in the context of designing airfoils to prescribed surface pressure distributions. Inverse methods require that the flow features of the intended design are specified a priori; traditionally this would be the task of an experienced aerodynamicist, but this specification can also be the result of an optimization on these flow features [3]. Inverse methods have proven valuable because once the target pressure distribution is specified the required airfoil geometry can be obtained with very few CFD evaluations, compared to a direct optimization on the geometry [2,3]. Of course, the inverse process always requires a construction of the geometry to perform CFD analysis, but knowledge of the target surface pressure allows rapid convergence towards the final design.

The method introduced here does not approach inverse design from this traditional perspective. Rather, the inverse process is used as a tool within a more complex multifidelity search and is called upon whenever an airfoil geometry is required whose flow features match those specified by a parametric model, that is, the design search method uses a parameterization of the flow features coupled with inverse design. This is compared to a direct method which focuses purely on the geometry of a component and its resulting flow characteristics, such as lift and drag. The motivation for investigating this more involved strategy is the desire to reduce the number of design variables in the design search process. Thus, development using high-fidelity CFD analysis is accelerated without requiring the specification of the target pressure distribution ab initio, which is the hallmark of classical inverse design and one of its principal drawbacks. This alternative approach is based on the observation that the pressure distribution resulting from an airfoil shape can often be simpler to define than its geometry, that is, the flow features of a component can potentially be described by fewer design variables compared to the geometry, independent of simulation fidelity. This reduction in problem dimensions means that fewer CFD evaluations are required in the design search process, making it less computationally expensive to achieve a given level of performance. This approach also permits careful control of the design process via constraints applied to the flow features, and this is particularly useful if the designer is aiming to achieve a particular type of flow structure.

Here, the aerodynamic component being designed is a two-dimensional airfoil section in subsonic flow. This is a well-known and relatively simple problem, which simplifies the comparisons between the method under investigation and existing technology. The distribution of pressure over the airfoil surface is chosen as the

Received 12 July 2005; revision received 28 March 2006; accepted for publication 28 March 2006. 

*Graduate Research Student, School of Engineering Sciences. Member AIAA.
†Senior Research Fellow, School of Engineering Sciences.
‡Professor of Computational Engineering, School of Engineering Sciences.

design flow feature. This is an intuitive choice because small perturbations in pressure distribution can often require large variations in the entire geometry, for a given set of flow conditions [3]. In related work, Obayashi and Takanashi [4] used aerodynamic design relationships and constraints to relate the surface pressure distribution to certain airfoil performance parameters. A genetic algorithm (GA) was used to optimize the pressure profile for minimum drag, after which an inverse design code was employed to recover the corresponding airfoil shape. Jameson [5] optimized an airfoil shape using an Euler based CFD analysis, and used the pressure distribution of the resulting airfoil as a target for inverse design using a Reynolds-averaged Navier–Stokes (RANS) solver, because this gives more accurate viscous drag predictions. Ahn et al. [6,7] used lift and drag relations together with CFD analyses on a series of airfoil geometries to build a response surface model (RSM), able to relate the surface pressure distribution to predictions for airfoil lift and drag. The target pressure distribution could then be optimized by using a GA search over the RSM, and an inverse design code [6] used to determine the resulting airfoil shape. For each spanwise station of their three-dimensional wing, the number of design variables for the optimization of the pressure distribution was 15. The authors of [7] recognize that replacing airfoil section shape parameters with section pressure distributions gives a saving in computational cost. They report the computational cost of their work to be one-sixteenth of the cost of direct design methods, although no detailed comparisons are made.

The work presented here makes comparisons between two design methods. The first is treated as a benchmark in aerodynamic shape optimization. In this, the geometry model is defined by 13 design variables and the design is evaluated using high-fidelity CFD software to give a measure of performance (drag at a fixed value of lift). An efficient design search and optimization (DSO) method based on response surfaces is used to maximize this performance measure by varying the design variables. The alternative method adopted uses a parameterization of the pressure coefficient ($C_p$) profile for the airfoil section using six parameters, that is, considerably fewer than can be used for the direct optimization method. For each variation in the design variables of this model, an airfoil geometry which produces this prescribed pressure distribution must be found by inverse design. Crucially, a much less expensive CFD solver is used during the inverse design process: well-calibrated full potential solvers are able to accurately calculate the airfoil pressure distribution with very little computational effort. Once the resulting geometry is determined its performance is analyzed using the high-fidelity CFD software, adopting the same performance metric as in the benchmark method. The efficient DSO method is again employed to maximize the performance, but now by manipulating the variables of the $C_p$ profile.

Because it uses more design variables, the benchmark design method requires a relatively large number of full CFD evaluations for sufficient coverage of the design space, but acts directly on the geometry. The alternative method calls for fewer full CFD evaluations, but for each variation of the design variables an inverse design step using low-fidelity CFD is required to first recover the geometry. This work investigates the relative computational expense of these two methods in reaching a similar level of design performance. In the sections that follow, the use of the two design methods is described in full. The methods have been evaluated side by side for a 2-D airfoil design problem; their relative performances are discussed. Both design methods employ the same high-fidelity flow solver, whereas the alternative method additionally calls upon a low-fidelity code; the setup of the CFD analyses is the subject of the next section.

## II. CFD Solver Setup

In the current work, the high-fidelity solver used is FLUENT® [8] and the low-fidelity code (used in the newly proposed method) is the full potential flow solver VGK [9]. Calibration of the CFD solvers requires a set of accepted experimental results for a standard airfoil whose geometry can be reproduced. For the alternative design method, in particular, it is important that the airfoil pressure distribution must be reproduced accurately by both flow solvers. In the current study a subsonic flow regime is used, as this simplifies the pressure distribution to be parameterized in the alternative method. The NASA low-speed (LS) airfoil family [10] provides a suitable baseline for drawing comparisons with the results of a subsonic design study. These airfoils have undergone extensive development and have been used as wing sections on low-speed civil aircraft such as the Stoddard-Hamilton Glassair and the Adam Aircraft A500 and A700. The low-speed family is designed to operate at a lift coefficient of 0.4 and typically a flow speed Mach number of 0.15 and so these conditions are used in the present design studies. The starting airfoil is the first point in the design search and should be the same for both methods. To evaluate the ability of the two methods to improve the design, the starting geometry should also not perform particularly well at the selected operating conditions. In this case, a symmetrical geometry is unlikely to be optimal, hence the 13-% thick symmetrical NASA LS(1)-0013 airfoil is used as the initial design, for which experimental surface pressure results are given in [11]. These data have been collated such that comparisons can be made with results from FLUENT and VGK for the same airfoil (Fig. 1). Because the flow solvers are validated with the flow conditions specified in [11], these conditions are used in calculating the design objective for the two design methods; this also allows any resulting airfoil designs to be compared directly with the NASA low-speed airfoils. Further information about the configuration of the CFD solvers is given in Table 1.

**Table 1 Detailed information regarding the setup of the CFD solvers used in this study**

| | |
|---|---|
| **FLUENT** | |
| Mesh | 2-D unstructured, triangular |
| Boundary layer mesh: | Structured, rectangular |
| First cell thickness | 0.045% chord |
| Total thickness (approx.) | 2.5% chord |
| Total mesh elements (approx.) | 17,500 |
| Solver | Coupled implicit formulation of RANS equations, also solving the energy equation |
| Viscous model | Spalart–Allmaras |
| Discretization schemes | Second order upwind |
| Flow medium | Air as an ideal gas |
| Reynolds number | $4 \times 10^6$ |
| Flow Mach number | 0.15 |
| Solver Courant number | 5 |
| Wall $y^+$ range (approx.) | 30–60 |
| **VGK** | |
| Mesh | Conformal mapping onto inside of a circle |
| Radial lines | 160 |
| Circumferential lines | 30 |
| Solver | Finite-difference solution of full potential equations |

### A. FLUENT

The FLUENT model is set up with the aim of minimizing the computational effort required for the analysis, giving robust convergence, while providing accurate results in close agreement to the experimental data as shown in Fig. 1. The FLUENT geometry generation and meshing tool, GAMBIT®, is used to mesh the flow domain to be solved by FLUENT. The airfoil geometry is imported into GAMBIT in the form of a data file containing a matrix of $(x, z)$ coordinates, with 103 points on each surface. The extent of the flow domain is a square with 40 m edges, with the airfoil normalized to a chord of 1 m. A structured boundary layer mesh of 20 cells thick is attached to the airfoil surface, with the first cell given a thickness of 0.045% airfoil chord and the cells growing in size toward the freestream. The remaining flow domain consists of an unstructured triangular mesh, with a size function applied to the airfoil to grow the

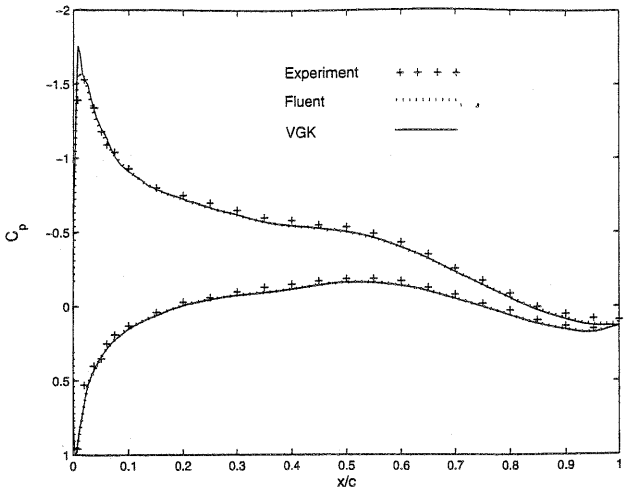


**Fig. 1 Comparison of pressure distributions generated using the FLUENT and VGK CFD solvers, for the NASA LS(1)-0013 airfoil. This is compared with experimental data.**

cell size with distance from the surface, thus reducing the computational expense of the analysis. Typically, the full mesh consists of 17,500 mesh elements, including the structured boundary layer. In this configuration an increase in the mesh size gives a negligible variation in the resulting force coefficients, providing good numerical accuracy.

In the CFD model described here, the equations of momentum and viscosity are solved in a coupled manner and the Spalart–Allmaras turbulence model [12] is employed, as this is known to be a relatively accurate method for external flow over an airfoil, giving robust convergence. This latter feature is desirable when a large number of calls to the analysis are made during the design search process. On studying the convergence history of the solver using this setup, it is observed that 2000 iterations of the RANS calculations are sufficient to provide a converged solution. At this point the variation in the drag coefficient for the airfoil is within $\pm 0.1$ counts ($\pm 0.00001 C_D$) of the fully converged value. This convergence criterion is used as the lift and drag coefficients form the airfoil design metric. For the converged solution the nondimensional distance $y^+$ over the airfoil surface lies in the range suitable for a log-law wall function representation of the boundary layer, that is, between 30 and 60.

Figure 1 shows the FLUENT results for the NASA LS(1)-0013 airfoil using the above setup and for the flow conditions specified in [11], that is, a flow speed Mach number of 0.15 and a Reynolds number of $4 \times 10^6$. Figure 1 confirms the strong similarity between the pressure profiles from FLUENT and the experimental data.

For each airfoil design analyzed using FLUENT, the desired performance metric is drag at a fixed value of lift. This is determined by running the analysis at a number of values of angle of attack until the desired lift is achieved. In practice, the analysis is run at two initial angles; at the first angle the solution is run for 2000 iterations and at the second it is run for a further 900 iterations. Following this, the correct angle is calculated from the lift curve and the analysis is run at this angle for a final 900 iterations, providing a converged solution at the desired value of lift to within $\pm 1\%$. Calculating the drag using the current CFD setup takes on average 21.4 min when running on a Xeon 2.8 GHz compute node with 2 Gb memory, and this process is used to calculate the airfoil performance metric in both design methods under consideration.

### B. VGK

The method newly presented in the current work requires a computationally inexpensive CFD solver to compute the airfoil pressure distributions during the inverse design step. The low-fidelity CFD code used here is viscous Garabedian and Korn (VGK) method, written by DRA Farnborough and distributed by the Engineering Sciences Data Unit (ESDU). VGK is a 2-dimensional viscous coupled finite-difference code which solves the full potential equations, written specifically for the analysis of airfoils. The airfoil geometry is input as a matrix of coordinates; following this a computational mesh is built in the flow domain using a series of radial and circumferential grid lines. The full potential equations are solved iteratively over the grid using a finite-difference approach. Because a potential formulation is used, VGK cannot produce results for flows where the boundary layer has separated from the surface. The code has been calibrated against experimental data, which gives it the ability to estimate the location of the separation boundary and which also results in more accurate drag estimations. Typically, each airfoil analysis takes around 2 s on a Xeon 2.8 GHz compute node with 2 Gb memory.

The VGK CFD model is set up as a viscous solve with the same Reynolds number and flow speed as used when running FLUENT. As with the FLUENT CFD model, the solver was run for the standard NASA LS(1)-0013 airfoil for comparison with the experimental data; this is also shown in Fig. 1. With the exception of the pressure *spike* at the leading edge, the VGK method gives accurate pressure distribution results compared to the experimental data.

## III. Design Strategies

### A. Direct Optimization

A direct optimization design method is used here to act as the benchmark in aerodynamic shape optimization, against which the alternative method is compared. In the benchmark method (Fig. 2a) the airfoil geometry is parameterized and design iterations, requiring high-fidelity CFD analysis, are automated using an optimization algorithm. Ultimately, one hopes to arrive at an *optimum* design, but in reality when there are more than two or three variables this becomes an exercise in design improvement.

The choice of parameterization method is a critical factor in the performance of direct searches such as the benchmark design method used here. This has been the subject of extensive research [13], seeking representations which reduce the number of design variables while retaining the ability to capture a global range of designs. The use of $B$ spline or Bézier curves generally facilitates a large design space, but with a correspondingly large number of design variables. Some methods use existing airfoil shapes, either as a linear combination or to derive a series of orthogonal shape functions [14], yielding a design space which leads to evolution rather than innovation. Shape modifying functions such as Hicks–Henne functions [15] are also commonly used. Song and Keane [16] compared an interpolating $B$-spline method with an orthogonal shape function based method and reported that although the spline approach is computationally expensive it is able to capture a larger range of geometries accurately; Samereh [13] also reports that the use of polynomial splines is well suited to a two-dimensional study. An interpolating segmented cubic polynomial spline is chosen here to parameterize the airfoil for the direct search method.

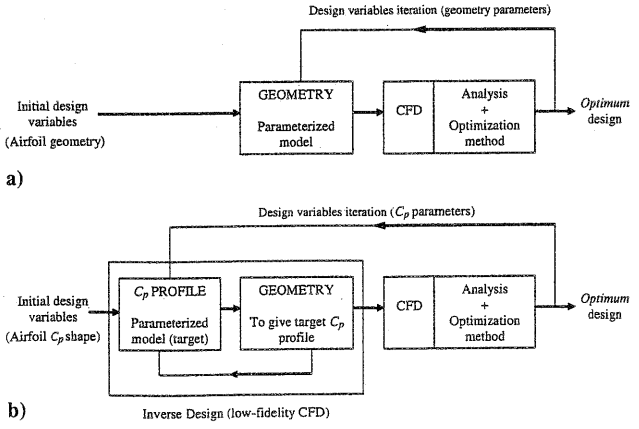


**Fig. 2 a) Benchmark *direct* aerodynamic design optimization method. b) Alternative *inverse* design method based on parameterizing the $C_p$ profile (EMFID).**

A single segment of the polynomial spline describes the variation of $z$ with $x$ in the form:

$$z(x) = \sum_{i=1}^{4} a_i x^{i-1}, \qquad x_1 \le x \le x_2 \tag{1}$$

The constants $x_1$ and $x_2$ are the extents of the spline segment in $x$, and $a_i$ are coefficients to be determined by specifying boundary conditions for the segment. These boundary conditions arise by requiring adjacent spline segments to have the same value of the first and second derivatives at the point at which they join. The gradient is specified at the start point of the first segment and the endpoint of the last segment of the complete spline; this completes the curve. An advantage of this segmented curve over a $B$ spline is the ability to directly control the trailing edge angle. By specifying the coordinates of the segment joins, or control points, the coefficients of the polynomials are found and thus the surface is defined. When this curve representation is applied to an airfoil shape, problems arise at the leading edge where the gradient often approaches infinity. The use of a cubic polynomial in this region is unnatural; the designers of the NACA airfoil sections overcame this by using a polynomial expression which included a $\sqrt{x}$ term [17]. Thus, for the leading edge segment of the airfoil the terms with $x$ of order 2 and 3 have been replaced with a $\sqrt{x}$ term. Therefore, the gradient at the start of the spline does not have to be specified; the singularity at the leading edge gives an *infinite* gradient.

The airfoil shape is thus defined by 10 spline segments or 11 control points in ($x$, $z$) space, which are interpolated by the curve (Fig. 3). The design variables are selected from the possible $x$ and $z$ movements of the control points, with the aim of minimizing the number of design variables while retaining the ability to produce smooth and varied airfoil shapes. The airfoil is separated into upper and lower surfaces, while the leading edge point (0, 0) is shared by both surfaces and remains fixed. Of the remaining five points on the upper surface, the near leading edge point (point A in Fig. 3) is free to move in both $x$ and $z$ directions and the trailing edge point B is fixed, whereas the other three points are constrained to movement in the $z$ direction only. The same applies to the lower surface, with the exception of the point adjacent to the trailing edge point C, which is free to move in both directions. The trailing edge thickness is fixed at the value corresponding to the initial design in the search process. Additionally, the gradient of each surface at the trailing edge is added to the list of design variables, because the exit angle is important in the governing aerodynamics. Thus, there are 13 design variables in total defining the geometry of the airfoil, which must be searched by the optimizer and analyzed using CFD software. It is thought that 13 is a reasonable number of variables for an airfoil design problem of this nature; it is not uncommon for such a problem to make use of 22 or more variables [16,18,19].

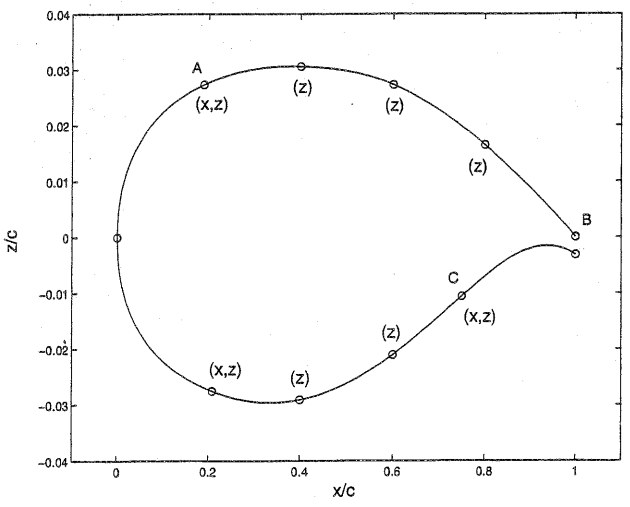


**Fig. 3 Airfoil geometry representation for the direct optimization method, showing control point degrees of freedom.**

Using the high-fidelity CFD solver (FLUENT), at each airfoil design iteration the lift and drag coefficients ($C_L$ and $C_D$) can be calculated, which are then used as metrics of performance. As already noted, the design objective of the optimization procedure is to minimize $C_D$ calculated at a constant value of $C_L$, allowing the angle of attack to float. The optimization strategy in the benchmark design method uses a RSM approach. For optimization using high-fidelity CFD, it is imperative to minimize the number of objective function calls. The RSM method is therefore a popular choice for global optimization using expensive functions [20,21] because it can be used to predict promising areas of the design space with relatively few objective function calls, compared to other design search methods. Further savings in overall time can be made by constructing the RSM with simultaneous calls to the objective function, something that is not always possible in other approaches to design optimization. Initial design iterations are generated as dictated by a formal *design of experiments* [1] (DOE) to populate the designated search space, and the objective function is calculated for each point. A RSM is built based on the initial DOE and searched to predict areas where improved designs may be found. The objective function is evaluated at these points and the surface is updated. The process of building, searching, and updating the surface is repeated, providing convergence towards an optimum in the design space. The RSM optimization routine for the current method is implemented using the OPTIONS design exploration system,[§] operating in the MATLAB®[‖] environment using the GEODISE toolkit.[¶] The setup of this procedure is described next.

The DOE used to seed the initial database is a Latin hypercube, which has good coverage of the design space and has the advantage of representing each variable's range equally [22]. Additionally, OPTIONS allows the random number sequence to be changed giving different, but repeatable, initial DOE sets. The response surface model used is an interpolating cubic spline radial basis function (RBF).

Once the RSM has been built, it is searched using a GA, implemented in OPTIONS. The GA gives a relatively thorough search of the whole design space, which can be tolerated since calls to the response surface are very fast; here 5000 search evaluations are used. Instead of searching for a single optimum point on the surface, parallel update points are extracted from the search. Not only does this speed up the optimization procedure by allowing the update points to be evaluated simultaneously, it also reduces the chances of stalling on a local minimum in the surface [22]. For this method, five parallel update points are requested, taken from a cluster analysis of the final population in the GA search. After evaluating the new design points, the RSM is updated.

For a given iteration of the design variables and call to the CFD analysis, it is possible that the calculations may fail for some reason (if the CFD calculations diverge, for example). An important decision concerns the handling of these failed design points. For the current method, design points in the initial DOE which fail are not included when constructing the RSM. When updating the response surface, however, failed points are included in the RSM and given an objective function value equal to the average of the objective values recorded so far. In this way, the coverage of the data set is statistically unaltered, while the RSM is such that the areas of failure are not sampled again by the search algorithm.

### B. Embedded Multifidelity Inverse Design (EMFID)

Increasingly, multifidelity approaches to design optimization are being used both to improve the reliability of the analysis and reduce the computational expense of a design search [1]. Multiple levels of CFD model complexity have been used simultaneously in an automated way in previous research, for example, by Keane [23] and Alexandrov [24]. The alternative method described in the current work is illustrated in Fig. 2b. Instead of directly parameterizing the

[§]Data available on-line at http://www.soton.ac.uk/~ajk [cited April 2005].
[‖]Data available on-line at http://www.mathworks.com/ [cited April 2005].
[¶]Data available on-line at http://www.geodise.org [cited April 2005].

airfoil geometry to generate design iterations, the pressure distribution over the airfoil surface is parameterized. The geometry which produces this target pressure profile is then recovered using an inverse design method; following this the airfoil performance is calculated via the high-fidelity CFD software. The optimization algorithm is used to maximize this performance by manipulating the design variables defining the pressure distribution. Although high-fidelity aerodynamic optimization is desired and hence an expensive solver is used to evaluate the airfoil performance, the key to the effectiveness of this method is the use of a lower-fidelity CFD solver for the inverse design steps; thus we have a multifidelity search procedure. In using rather few design variables to describe the pressure distribution, the number of calls to the full CFD solver required to populate the design space is reduced considerably over the benchmark optimization method. However, for each call to the objective function an inverse design step must be performed. The effectiveness of this method relies on the savings made in reducing the number of CFD evaluations being greater than the relative cost of the inverse design steps. The configuration of EMFID is now described in full, starting with the parameterization of the airfoil pressure distribution.

### 1. *Parameterization Technique*

The representation of the pressure (or velocity) distribution for an airfoil has been attempted by various authors using different approaches, almost invariably applied to the design of transonic airfoils. van Egmond [3] formulated a set of aerodynamic shape functions for a transonic pressure distribution, capable of representing a wide range of airfoil flows (this was also used by Ahn and Kim [7]). Obayashi and Takanashi [4] used $B$-spline curves to represent the pressure profile. Gopalarathnam [25] chose to parameterize the velocity distribution, by splitting the surfaces into segments over which the velocity difference was specified. All of these methods use in excess of 12 design variables, partly owing to their ability to represent transonic as well as subsonic flow regimes. Here, the focus is on subsonic flow regimes only, which simplifies the type of pressure distribution to be represented. Further reduction in the dimensionality of this model can be achieved by an appropriate choice of parameterization. To make comparisons between the EMFID method and the benchmark optimization, the same initial design must be used. This means that the pressure distribution for the NASA LS(1)-0013 airfoil under the chosen design conditions must be capable of being represented reasonably well by the parameterization, with the corresponding parameter values being used to start the DSO process.

With respect to the parameterization method, the objective is to allow the generation of a wide range of realistic subsonic $C_p$ distributions, while also limiting the dimensionality of this model. Here, the $C_p$ distribution is parameterized using one $B$-spline curve for each airfoil surface. Each $B$-spline curve contains four knots and four control points. These are constructed on a knot vector of four zeros and four ones, giving a cubic Bézier curve. The control point locations are determined by specifying that the curve must interpolate four data points. The chordwise positions of these data points are approximately $x/c = \{0, 0.6, 0.8, 1\}$. The height of the trailing edge point is fixed whereas the heights of the three remaining points are the profile design variables, giving a total of six variables for the complete $C_p$ distribution.

As noted above, the two design search methods evaluated in this work are started from a parameterization of the same initial design, NASA LS(1)-0013. Figure 4 compares the geometry and $C_p$ distribution for each of these parameterizations with the original NASA shape. This original NASA geometry is shown as a thick solid line. The parameterization used in the benchmark method is shown as a thin dashed line, and this corresponds closely with the shape and $C_p$ profile of the original airfoil (to the extent that its geometry is hidden in Fig. 4). The EMFID parameterized representation of the NASA 0013 $C_p$ distribution is shown as a dotted line in Fig. 4. This profile was found by running a search on the six pressure profile variables to minimize the error with respect to the NASA $C_p$ profile.

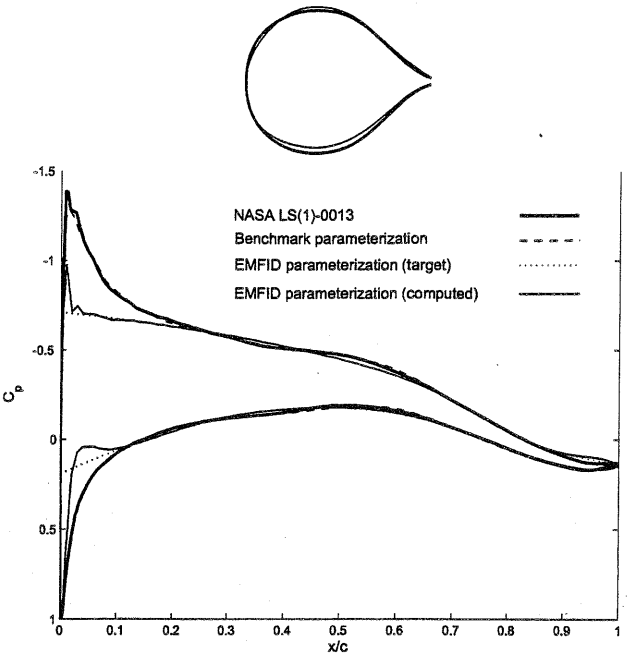


**Fig. 4 Airfoil geometry and pressure distribution for the parameterizations of the NASA LS(1)-0013 used by the two methods, shown with those of the original shape.**

The resulting profile is close to that of NASA airfoil, except in the leading edge region; in this region the pressure is sensitive to the tradeoff between airfoil angle of attack and camber. The airfoil geometry which realizes this parameterized target is found by running the inverse design process (detailed later); this geometry and its $C_p$ distribution are shown as a thin solid line in Fig. 4. It is seen that the disparity between the NASA 0013 pressure profile and the parameterized target has resulted in an airfoil design with a slightly different geometry, in particular, an increase in camber. This highlights a limitation of the EMFID method using the current parametric model. A model could be configured such that the initial design is replicated more accurately, however, such a model would involve more variables, defeating the object of the EMFID process. Moreover, it turns out that the current parameterization is sufficiently flexible for the design of a general subsonic airfoil. The discrepancy between the geometries shown in Fig. 4 is not thought to disadvantage either method. Figure 4 shows that the inverse design produces a $C_p$ profile which closely agrees with the parameterized target. Of course, it is not immediately apparent how close one must get to the target profile on each inverse design step, to provide a meaningful design search process. However, experience with this process has shown that the target must be sufficiently well reproduced, such that small changes made to the target $C_p$ are represented appropriately in the resulting airfoil shape.

### 2. *Inverse Design*

The EMFID method requires an inverse design process to derive an airfoil geometry which produces the specified target pressure distribution, or at least a reasonable approximation to it, at each iteration of the design search. Here, the full potential code VGK is used to calculate the pressure distributions for this inverse process. It must be able to accurately converge onto the target pressure distribution while minimizing the number of CFD calls.

A significant research effort has been put into inverse design methods, much of which focuses on surface flow design, that is, matching a prescribed surface pressure distribution [2]. LeGresley and Alonso [26] use proper orthogonal decomposition to calculate gradients for the minimization of the least-squares error between the target and computed pressure profiles. Dulikravich [2] reports that the modified Garabedian method is popular, but is slow to converge. Stream function based methods have proven very efficient, giving a fully converged geometry within 10–20 design iterations. The three-

dimensional method described by Takanashi [27] uses an integral formulation of the full potential equations and reports excellent inverse design results in only 10 design iterations. This is similar to a method proposed by Davis [28] for two-dimensional transonic airfoils. The inverse design method used here is similar to that used by Davis and is described next.

The method is based on an iterative residual-correction concept. For a given chordwise station, the residual is the difference between the target and the computed pressure distributions. At the $k$th iteration the second derivatives of the airfoil shape are corrected according to the following expression:

$$z''_k = z''_{k-1} + \left(\frac{dz''}{dC_p}\right)_{\text{approx}} (C_p^T - C_p^{k-1}) \tag{2}$$

For each chordwise ordinate, $C_p^{k-1}$ is the design pressure coefficient computed by a flow solver at iteration $k-1$. The gradient $(dz''/dC_p)$ is calculated using approximate flow formulas, which can be relatively crude given the iterative nature of the design process. The flow formulation adopted here is the small disturbance theory used by Davis [28]. Thus, the inverse design process proceeds as follows: 1) the pressure distribution is calculated for an initial airfoil using the potential flow solver VGK, and the difference to the target pressure distribution is calculated; 2) an approximation to the pressure distribution is calculated using small disturbance theory, allowing the gradient $(dz''/dC_p)$ to be determined; 3) the surface second derivatives are corrected using Eq. (2); 4) integration yields the corresponding airfoil surface. The process is repeated until convergence is reached.

The inverse design procedure requires calculation of surface second derivatives, and these must also be integrated twice once they are corrected. To avoid errors with numerical differentiation/integration, particularly at the leading and trailing edges, an analytical approach is used here. A tenth-order polynomial is fitted to the airfoil surface coordinates in a least-squares sense, employing a singular value decomposition algorithm. This is similar to the approach used in [7]. To ensure a smooth leading edge radius, the polynomial spline terms used for the leading edge in the benchmark parametric model are used in this curve also. The second derivative of this polynomial can thus be computed exactly. When these derivatives are corrected using Eq. (2), the polynomial coefficients no longer apply, hence the new surface derivatives must be fitted with a new curve. An eighth-order polynomial is used so that when the derivatives are integrated twice the polynomial coefficients for the corrected surface can be carried over into the next iteration without needing to recalculate them. Integrating the second derivatives twice yields two constants of integration, which are found from the known position of the leading and trailing edges. This allows all the polynomial coefficients to be determined, specifying the new airfoil.

The objective of the design strategies described here is to minimize drag for a fixed quantity of lift while allowing the angle of attack, $\alpha$, to be varied: $\alpha$ therefore becomes a variable in the inverse design process. As a consequence, the process is also given increased capability in matching a given target $C_p$ distribution. Increasing $\alpha$ gives a monotonic increase in lift, and because this increases the area between the $C_p$ curves for the upper and lower surfaces, $\alpha$ can be adjusted at each inverse design iteration using an expression similar to Eq. (2):

$$\alpha_k = \alpha_{k-1} + R\sum_S (C_p^T - C_p^{k-1}) \tag{3}$$

where $\alpha_k$ denotes the angle of attack at iteration $k$ and $S$ represents the airfoil surface.

In Eq. (3), $R$ is a relaxation factor applied at each iteration to prevent excessive corrections to the angle. A value for $R$ of 0.2 deg is used in the current study. The use of relaxation provides a facility to control the convergence of an iterative procedure, a bigger factor gives faster convergence but increases the risk of instability. A relaxation factor is also applied to the pressure residual term in Eq. (2), because experience with this inverse design method revealed that the surface corrections at each iteration can be overly large. Relaxation is used in [28], although its magnitude is not specified; a factor of 0.4 is used here. The inverse design approach requires a relaxation factor because of its iterative nature and because of the simplicity of the surface pressure approximation. The use of such factors is not uncommon in design optimization [1], although the appropriate magnitude is likely to depend on the problem set up. The magnitudes specified here have been found by monitoring the number of iterations required for convergence as the relaxation factor is varied.

Using the method described above, the inverse design step uses approximately 10 VGK calls to capture the target pressure profile. However, it may require as few as five VGK calls or as many as 30 for certain target profiles, particularly when matching features at the leading edge. The least-squares error between the target and the computed profile is used as a measure of the convergence of the inverse design process. This error diminishes rapidly as the iterative procedure converges onto the desired pressure profile. Eventually, the computed pressure profile is unable to match the target any closer, and the error increases fractionally; at this point the process is deemed to be converged and is halted. If this convergence criterion is not met after 40 inverse design iterations the process is stopped. If, during the procedure, a geometry is passed to VGK which causes the calculations to fail (for example, if the solution diverges), the geometry passed to the high-fidelity CFD is the last one for which VGK gave a converged solution.

#### 3. *EMFID Search Setup*

To make fair comparisons between the alternative method and the benchmark direct search method described above, the optimization algorithm used and its implementation are set the same for both methods. Therefore the RSM approach described for the benchmark strategy is also used as the optimization method shown in Fig. 2b. For the EMFID method, the design search involves manipulation of the shape of the $C_p$ distribution, and because the integral of surface pressure is lift this results in a different target lift depending on the variable values. To allow for this, the pressure coefficient values, used as the target during the inverse design process, are scaled such that the area between the curves is equal to the desired target lift.

In summary, a call to the objective function in EMFID takes the design variables, calculates the corresponding pressure distribution, and then scales it to give the required total area and target lift. The inverse design code (low-fidelity CFD) is then used to calculate an airfoil geometry which realizes the specified pressure variation, and this shape is passed to the high-fidelity CFD analysis for calculation of $C_D$ and $\alpha$ at the required value of $C_L$. The high-fidelity CFD code is set up in the same way as in the benchmark method.

### C. Comparing the Two Methods

The two design methods described above are here evaluated side by side to assess their efficiency. The more efficient design method is the one which reaches the highest level of design performance (minimum $C_D$) given a fixed computational budget. Also of interest is the convergence of the design search process; does the method converge, and if so to what level of performance? Each method has therefore been run 5 times using different Latin hypercube DOE seeds; their optimization histories, showing the progress of the optimizer in maximizing performance against the number of designs evaluated, are presented.

An important factor when comparing the two methods is the bounds placed on the design variables. These must be equivalent for each method such that one method is not forced to search a much larger or smaller design space than the other. It is also desirable to maintain a conceptual design approach and permit innovative and radical designs. To allow a direct comparison, the EMFID method must have bounds in pressure profile terms which are equivalent to the geometrical bounds of the benchmark method. However, this is difficult to achieve because a modification to the shape of one surface affects the pressure over both surfaces. To constrain the two methods fairly, bounds must be placed on design parameters which exist in

both methods. In the current study, constraints are therefore placed on the airfoil shape directly, namely, on the maximum thickness, maximum camber, and leading edge radius.

The variables for the benchmark method control the position of spline points on the airfoil surface, where those for the EMFID method control spline points which establish the shape of the target pressure distribution. These variables are therefore each given a relatively larger range, resulting in a large potential design space. For each set of variables selected by the optimization process of either method, if these variables generate an airfoil geometry which violates the constraints on thickness, camber, and leading edge radius, the variable set and resulting geometry are rejected. The objective function is not calculated for rejected geometries, and they are not included when constructing the RSM. If, however, all five update points requested from the search of the RSM are rejected, one of these points is added to the surface and treated as a failed design point. This action prevents an identical RSM being generated which would stall the optimization process. In addition to equating the search bounds of the two methods, the mathematical form of the airfoil shapes is made to be the same by fitting the benchmark parametric model to the EMFID geometries. These measures enable a direct comparison to be made between the optimization performance of the two methods, and their resulting geometries.

### D. Computational Expense

The two design methods described above are here run with equal amounts of computational effort; the one which is able to reach a higher level of design performance is deemed to be the more efficient method. A single objective function evaluation for the benchmark method takes on average 21.4 min when running on a Xeon 2.8 GHz compute node. An objective function evaluation for the EMFID method uses this effort plus the effort required in the inverse design step. Each VGK evaluation requires approximately 2 s; if on average 10 iterations are used in the inverse design step the total objective function call demands 21.7 min computational time. Thus, the ratio of computational expense for the two methods is 1:1.014 benchmark to EMFID evaluations.

## IV. Case Study

The benchmark design search method (Fig. 2a) and the EMFID method (Fig. 2b) have been run side by side to compare their relative performance. The benchmark method is given a budget of 300 calls to the high-fidelity CFD code, which equates to 295 objective function evaluations in EMFID after taking into account the effort required for the inverse design. Based on the recommendation of Jones [20], the number of points in the initial DOE should be 10 times the number of design variables. For the benchmark method this gives 130 design points, and for the EMFID method this requires 60 points. The methods use their remaining budget to evaluate designs during the update process, giving 170 update points for the benchmark method and 235 update points in EMFID. The design constraints on airfoil shape maximum thickness, maximum camber, leading edge radius, and total lift, respectively, are

$$12.5\% \leq t/c \leq 15\%, \qquad 0\% \leq z_c/c \leq 2.5\%$$
$$r_{\text{LE initial}} \leq r_{\text{LE}} \qquad C_L = 0.4 \tag{4}$$

In these constraints $r_{\text{LE initial}}$ is the leading edge radius of the initial shape in the design process, that is, the NASA LS(1)-0013 airfoil. The constraints are arranged with relatively narrow ranges such that the design methods converge more quickly onto promising designs. In addition to the above, a constraint is applied which states that the airfoil thickness at 95% chord shall be no smaller than 0.8% chord. With these constraints applied, the airfoil designs are forced to take a conventional form. As already noted, the target lift and flow conditions are the same as those used in [10,11], allowing comparisons to be made with the NASA low-speed airfoils.

The optimization-iteration histories for the two methods are shown in Fig. 5. The progress of the optimizer in each case is plotted against the number of benchmark iterations; the number of iterations of EMFID has been scaled in this plot such that the $x$ axis can be interpreted as equal computational expense. The design objective is drag, which has been normalized by multiplying by the ratio of lift output from the CFD code to the target lift; this removes the numerical error generated if the airfoil $C_L$ is not exactly 0.4. From Fig. 5 it is seen that for a given computational cost, the EMFID method finds a better design performance than the benchmark search method. The rate of convergence towards the optimum is faster for the EMFID method. This is because EMFID is able to find significantly better designs in its initial DOE, and this in turn is due to the reduced problem dimensionality. Although neither method has provided convergence onto a single optimum design, the five runs of the EMFID method show greater convergence, indicated by the reduced range of objective values at the end of the search process.

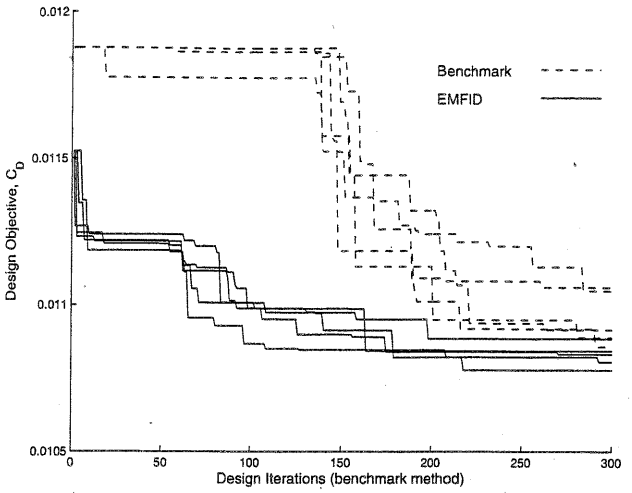


**Fig. 5 The five optimization histories for the benchmark and EMFID methods.**

The geometries which were found to give the best performance for each of the five runs of the benchmark procedure are shown in Fig. 6. It is clear that each initial DOE set has given a different final result for the computational budget used, but all designs have similar features. Table 2 gives data on these geometries. Although the shapes in Fig. 6 are not converged, the thicknesses of the five airfoils are similar, suggesting the beginnings of a converged design. However, the combinations of maximum camber and angle of attack of these airfoils are quite different, possibly a result of the optimization procedure locating many local optima. This is a symptom of the high dimensional design space.

The five best geometries from the computations using the EMFID method are shown in Fig. 7. The more advanced state of convergence is made clear by the greater similarity in the five geometries than the benchmark method achieved. The design searches have converged almost onto a single optimum geometry, and this is possible because of the reduced dimensionality of the design space in EMFID. Although the description of the $C_p$ distribution is simple, the resulting geometry after inverse design can be of high detail and complexity.

Table 3 gives the airfoil design data for the five geometries resulting from the EMFID method. Contrary to the benchmark method, the maximum thickness, camber, and angle of attack in each

**Table 2 Airfoil design data for the five geometries resulting from the benchmark design search: maximum thickness, maximum camber, angle of attack, lift coefficient, and drag coefficient**

| Run | $t$ | $z_c$ | $\alpha$ | $C_L$ | $C_D$ (counts) |
|---|---|---|---|---|---|
| 1 | 0.1334 | 0.0144 | 1.54 | 0.3999 | 108.5 |
| 2 | 0.1371 | 0.0142 | 0.72 | 0.3999 | 109.1 |
| 3 | 0.1288 | 0.0155 | 1.05 | 0.3999 | 108.9 |
| 4 | 0.1359 | 0.0241 | 2.01 | 0.3998 | 110.4 |
| 5 | 0.1279 | 0.0214 | 0.73 | 0.3999 | 110.6 |

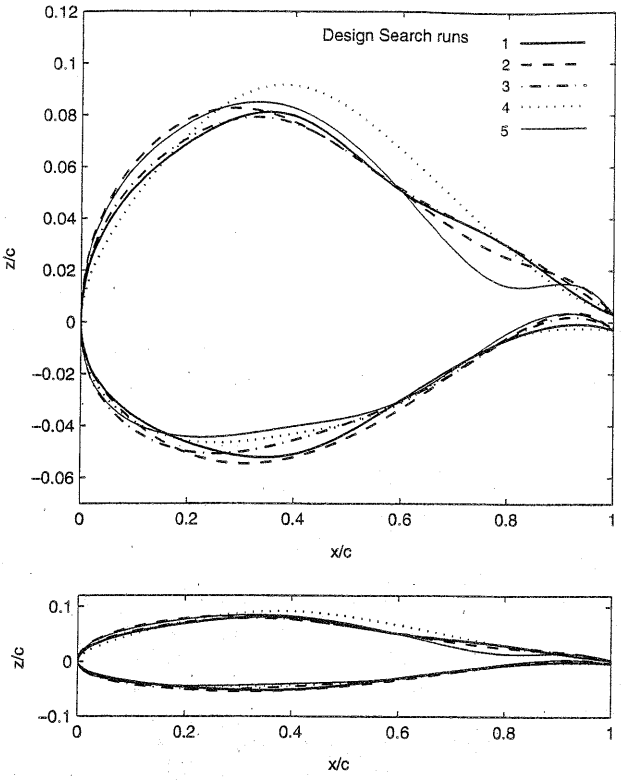


**Fig. 6 Final five geometries generated by the benchmark method. The lower figure shows the airfoils on equally scaled axes.**

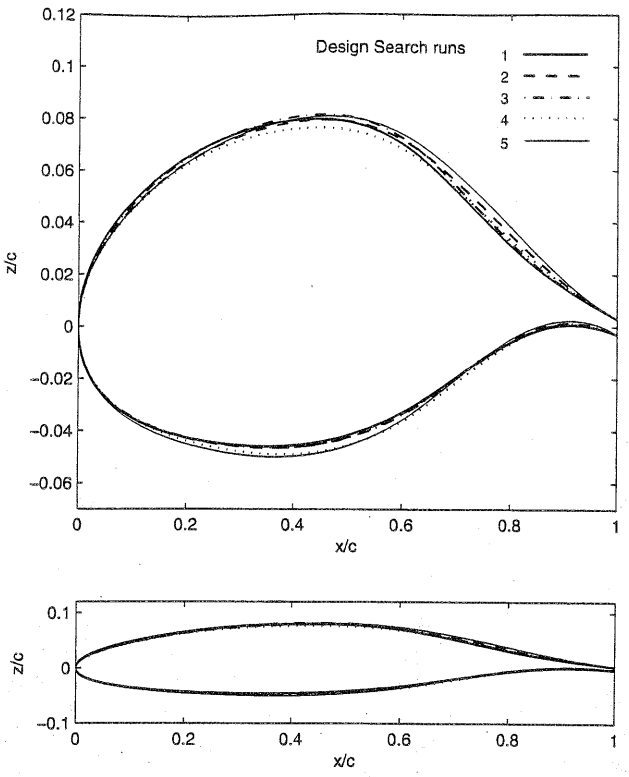


**Fig. 7 Final five geometries generated by the EMFID method.**

case are very similar, again signifying a greater level of convergence. Crucially, and rather predictably from looking at the airfoil shapes, the drag for the EMFID designed shapes is lower than those from the benchmark search method. In all five cases the maximum thickness is made to be almost as small as the constraints allow (12.5%), and this is to be expected because the objective is to minimize drag for a single angle of attack. The lower problem dimensionality in EMFID has allowed the optimization procedure to explore promising areas of the design space more thoroughly than is possible in the benchmark method for the same computational effort.

Figure 8 compares the best two geometries from the five benchmark and EMFID searches. Two NASA low-speed airfoils from [10] are also shown: the initial design in the search process, NASA LS(1)-0013, and NASA LS(1)-0413, also 13% thick. The 0413 airfoil has 2.2% camber and is designed to give a $C_L$ of 0.4 at 0 deg angle of attack; the 0013 airfoil has zero camber and gives a $C_L$ of 0.4 at approximately 3.5 deg angle of attack. The two best airfoils resulting from the benchmark and EMFID methods are 13.34 and 12.5% thick with maximum cambers of 1.4 and 1.6%, respectively. Both designed airfoils feature a cambered adverse pressure region at the lower surface trailing edge, similar to that on the 0413 airfoil. The lower surface of the EMFID airfoil closely resembles that of the 0413 shape, while the aft upper surface features the inflection point present on the 0013 shape which results in reduced drag. The benchmark airfoil features a slight *rippling*, associated with the incomplete convergence of the high dimensional search space.

Figure 9 compares the $C_p$ distributions from the best benchmark and EMFID airfoils (i.e., those in Fig. 8) with the NASA 0413 $C_p$ distribution, computed using VGK. In the case of the EMFID method two profiles are shown: the best target profile and the achieved profile which most closely matched this during the inverse design process. Note the erratic profile of the benchmark airfoil, an effect of the aforementioned rippling.

It is also useful to compare the performance of the designed airfoils in a more general sense. Figure 10 illustrates this as a lift-drag polar plot, showing the best designs from the benchmark and EMFID processes along with plots for the NASA airfoils 0013 and 0413. Of the NASA examples, the 0013 gives the lowest drag; this is at zero angle of attack because it is symmetrical; however, the 0413 airfoil

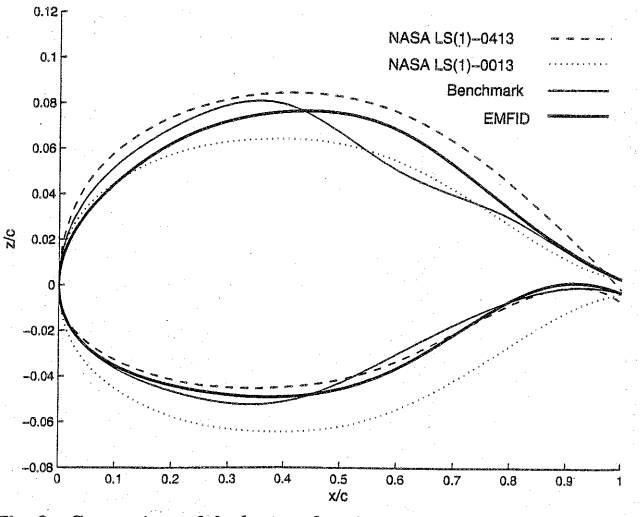


**Fig. 8 Comparison of the best performing geometry from each of the two methods, shown with two NASA low-speed airfoils of 13% thickness.**

gives lower drag at values of $C_L$ greater than around 0.3, due to its improved lifting performance. The airfoils designed by the benchmark and EMFID methods give lower drag at their design lift coefficient, 0.4, but their drag increases more rapidly with lift compared to the NASA airfoils. This result is an artifact of the design objective used: the NASA airfoils are designed to perform well over a range of angles, whereas the geometries generated here are designed to minimize drag at a single angle of attack (or a single $C_L$). It is not the purpose of this work to improve on the NASA airfoils; rather it is to demonstrate the use of the EMFID method in design.

As a final consideration, it is worth discussing the general applicability of the EMFID approach. The $B$-spline parameterization of the airfoil $C_p$ distribution used in the current work has been shown to perform well, but the successful application of the EMFID method is not limited to this model. The model is by no means optimal, but serves as an illustration of the simplicity with which the subsonic pressure distribution may be represented. A number of different

**Table 3 Airfoil design data for the five geometries resulting from the EMFID design search: maximum thickness, maximum camber, angle of attack, lift coefficient, and drag coefficient**

| Run | $t$ | $z_c$ | $\alpha$ | $C_L$ | $C_D$ (counts) |
|---|---|---|---|---|---|
| 1 | 0.1250 | 0.0189 | 1.15 | 0.3999 | 108.4 |
| 2 | 0.1251 | 0.0191 | 0.96 | 0.3999 | 108.0 |
| 3 | 0.1271 | 0.0195 | 1.08 | 0.3999 | 108.3 |
| 4 | 0.1252 | 0.0160 | 1.17 | 0.3999 | 107.8 |
| 5 | 0.1297 | 0.0195 | 0.78 | 0.3999 | 108.8 |

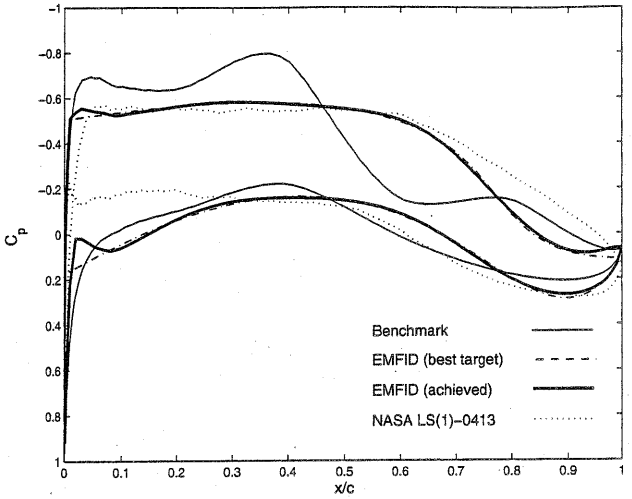


**Fig. 9 The final target pressure distribution and achieved profile, with the pressure distribution due to the best performing benchmark geometry.**

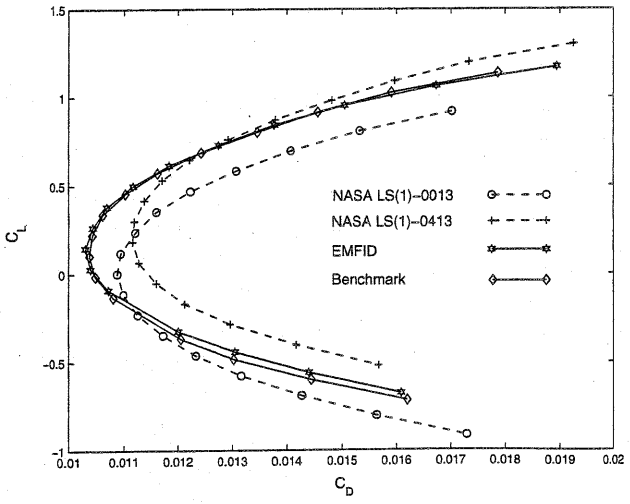


**Fig. 10 Lift-drag polar plot for the best designs from the benchmark and EMFID methods, shown with two NASA airfoils.**

representations have been used by the authors, and the results can be shown to be largely independent of the model used.

The method in the current work has been applied to the design of airfoils in subsonic flow conditions. This flow regime was chosen as the pressure distribution is more straightforward to parameterize, compared to the distribution when running at transonic flow speeds. A transonic pressure distribution can contain a shock, that is, a step in the profile upper surface. The present $B$-spline parameterization would not be suitable to represent such a pressure profile. However, it would not be difficult to facilitate the design of a shock in a reworked $B$-spline model. This modified model would probably call for more than six design variables and it is not known what impact this would have on its performance as compared to the benchmark method. The design of airfoils at transonic Mach numbers would also necessitate modifications to the inverse design code, which may result in a marginal increase in the expense of inverse design. The use of the EMFID method for the design of airfoils in more complex flow regimes is the subject of ongoing research by the authors.

As noted above with reference to the NASA airfoils, practical airfoils are designed to work well at multiple operating points. For example, one might calculate an airfoil performance metric at a number of different angles of attack, or at different flow speeds. In this case, the EMFID method would use the same objective function, but the inverse design step would be performed at just one of the operating points. For multipoint design optimization, the attraction of the EMFID approach is likely to be greater because fewer objective function calls are permitted for a given computational budget.

## V. Conclusions

An airfoil design search method has been presented which uses a parameterization of the subsonic pressure distribution to reduce the problem dimensions. The pressure distribution is relatively simple to describe parametrically, and the airfoil shape resulting from the subsequent inverse design can be of high quality and detail. This method has been compared directly with a method which focuses purely on the geometry of the airfoil. For a given computational cost, the proposed EMFID method is able to return airfoil geometries which perform noticeably better than those of the benchmark method, in terms of aerodynamic efficiency. The use of just six design variables compared to 13 in the benchmark process allows the EMFID method to converge significantly more quickly than the benchmark method. The reduction in the number of design variables has been shown to allow more thorough exploration of promising areas of the design space. The method may also prove to be successful when applied to transonic designs or to the design of a three-dimensional wing, where the relative cost of high-fidelity CFD analysis increases significantly over the cost of performing inverse design at the wing sections. This is the subject of future work in this area.

## Acknowledgments

This work has been funded by a scholarship from the University of Southampton School of Engineering Sciences. The support from Robert Lewis at Advantage CFD is gratefully acknowledged. In addition, the encouragement of Alexander Forrester, Nicola Hoyle, and Andrs Sbester is greatly appreciated.

## References

[1] Keane, A. J., and Nair, P. B., *Computational Approaches to Aerospace Design: The Pursuit of Excellence*, 1st ed., John Wiley & Sons Ltd., Chichester, U.K., 2005.

[2] Dulikravich, G. S., "Aerodynamic Shape Design," *Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD, Rept. 780, Nov. 1990.

[3] van Egmond, J. A., "Numerical Optimization of Target Pressure Distributions for Subsonic and Transonic Airfoils Design," *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD, Rept. 463, March 1990.

[4] Obayashi, S., and Takanashi, S., "Genetic Optimization of Target Pressure Distributions for Inverse Design Methods," *AIAA Journal*, Vol. 34, No. 5, May 1996, pp. 881–886.

[5] Jameson, A., "Re-Engineering the Design Process Through Computation," *Journal of Aircraft*, Vol. 36, No. 1, Jan.–Feb. 1999, pp. 36–50.

[6] Kim, H. J., and Rho, O. H., "Aerodynamic Design of Transonic Wings Using the Target Pressure Optimization Approach," *Journal of Aircraft*, Vol. 35, No. 5, Sept.–Oct. 1998, pp. 671–677.

[7] Ahn, T., Kim, H. J., Kim, C., and Rho, O. H., "Inverse Design of Transonic Wings Using Wing Planform and Target Pressure Optimization," *Journal of Aircraft*, Vol. 38, No. 4, July–Aug. 2001, pp. 644–652.

[8] Fluent, *FLUENT 5 User's Guide*, Fluent Inc., Lebanon, 2003.

[9] Engineering Sciences Data Unit, "The VGK Method for Two-

Dimensional Aerofoil Sections," ESDU, Rept. 96028, 1996.
[10] McGhee, R. J., Beasley, W. D., and Whitcomb, R. T., "NASA Low- and Medium-Speed Airfoil Development," NASA TM-78709, 1979.
[11] Ferris, J. C., McGhee, R. J., and Barnwell, R. W., "Low-Speed Wind-Tunnel Results for Symmetrical NASA LS(1)-0013 Airfoil," NASA TM-4003, 1987.
[12] Sai, V. A., and Lutfy, F. M., "Analysis of the Baldwin-Barth and Spalart-Allmaras One-Equation Turbulence Models," *AIAA Journal*, Vol. 33, No. 10, Oct. 1995, pp. 1971–1974.
[13] Samareh, J. A., "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, Vol. 39, No. 5, May 2001, pp. 877–884.
[14] Robinson, G. M., and Keane, A. J., "Concise Orthogonal Representation of Supercritical Airfoils," *Journal of Aircraft*, Vol. 38, No. 3, May–June 2001, pp. 580–583.
[15] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, July 1978, pp. 407–412.
[16] Song, W., and Keane, A. J., "A Study of Shape Parameterisation Methods for Airfoil Optimisation," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2004-4482, Aug. 2004.
[17] Jacobs, E. N., Ward, K. E., and Pinkerton, R. M., "The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel," NACA TR-460, 1933.
[18] Lépine, J., Guibault, F., and Trépanier, J. Y., "Optimized Nonuniform Rational *B*-Spline Geometrical Representation for Aerodynamic Design of Wings," *AIAA Journal*, Vol. 39, No. 11, Nov. 2001, pp. 2033–2041.
[19] Li, W., Krist, S., and Campbell, R., "Transonic Airfoil Shape Optimization in Preliminary Design Environment," AIAA Paper 2004-4629, Aug. 2004.
[20] Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, Dec. 2001, pp. 345–383.
[21] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization* Vol. 13, No. 4, Dec. 1998, pp. 455–492.
[22] Sóbester, A., Leary, S. J., and Keane, A. J., "A Parallel Updating Scheme for Approximating and Optimizing High Fidelity Computer Simulations," *Structural and Multidisciplinary Optimization*, Vol. 27, No. 5, July 2004, pp. 371–383.
[23] Keane, A. J., "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods," *Journal of Aircraft*, Vol. 40, No. 4, July–Aug. 2003, pp. 741–750.
[24] Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., "Optimization with Variable-Fidelity Models Applied to Wing Design," AIAA Paper 2000-0841, Jan. 2000.
[25] Gopalarathnam, A., and Selig, M. S., "Hybrid Inverse Airfoil Design Method for Complex Three-Dimensional Lifting Surfaces," *Journal of Aircraft*, Vol. 39, No. 3, May–June 2002, pp. 409–417.
[26] LeGresley, P. A., and Alonso, J. J., "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition" AIAA Paper 2000-2545, June 2000.
[27] Takanashi, S., "Iterative Three-Dimensional Transonic Wing Design Using Integral Equations," *Journal of Aircraft*, Vol. 22, No. 8, Aug. 1985, pp. 655–660.
[28] Davis, W. H., Jr., "Technique for Developing Design Tools from the Analysis Methods of Computational Aerodynamics," *AIAA Journal*, Vol. 18, No. 9, Sept. 1980, pp. 1080–1087.

N. Alexandrov
*Associate Editor*