

Wing aerodynamic optimization using efficient mathematically-extracted modal design variables

Christian B. Allen¹ · Daniel J. Poole¹ · Thomas C. S. Rendall¹

Received: 13 December 2016 / Revised: 13 October 2017 / Accepted: 10 February 2018 /
Published online: 1 March 2018
© The Author(s) 2018. This article is an open access publication

Abstract Aerodynamic shape optimization of a transonic wing using mathematically-extracted modal design variables is presented. A novel approach is used for deriving design variables using a singular value decomposition of a set of training aerofoils to obtain an efficient, reduced set of orthogonal ‘modes’ that represent typical aerodynamic design parameters. These design parameters have previously been tested on geometric shape recovery problems and aerodynamic shape optimization in two dimensions, and shown to be efficient at covering a large portion of the design space; the work is extended here to consider their use in three dimensions. Wing shape optimization in transonic flow is performed using an upwind flow-solver and parallel gradient-based optimizer, and a small number of global deformation modes are compared to a section-based local application of these modes and to a previously-used section-based domain element approach to deformations. An effective geometric deformation localization method is also presented, to ensure global modes can be reconstructed exactly by superposition of local modes. The modal approach is shown to be particularly efficient, with improved convergence over the domain element method, and only 10 modal design variables result in a 28% drag reduction.

keywords Constrained numerical optimization · Shape parameterization · Orthogonal modal design variables · Volume mesh deformation · Radial basis functions · Drag minimization · Aerodynamics

✉ Christian B. Allen
c.b.allen@bristol.ac.uk

Daniel J. Poole
d.j.poole@bristol.ac.uk

Thomas C. S. Rendall
thomas.rendall@bristol.ac.uk

¹ Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, UK

1 Introduction

Numerical simulation methods to model fluid flows are used routinely in industrial design, and increasing computer power has resulted in their integration into the optimization process to produce the aerodynamic shape optimization (ASO) framework. The aerodynamic model is used to evaluate some metric against which to optimize, which in the case of ASO is an aerodynamic quantity, most commonly drag or range, subject to a set of constraints which are usually aerodynamic or geometric. Along with the fluid flow model, the ASO framework requires a surface parameterization scheme, which describes mathematically the aerodynamic shape being optimized by a series of design variables; changes in the design variables, which are made by a numerical optimization algorithm, result in changes in the aerodynamic surface. Numerous advanced optimizations using compressible computational fluid dynamics (CFD) as the aerodynamic model have previously been performed (Hicks and Henne 1978; Qin et al. 2004; Nielsen et al. 2010; Lyu et al. 2015; Choi et al. 2014). The authors have also presented work in this area, having developed a modularised, generic optimization tool, that is flow-solver and mesh-type independent, and applicable to any aerodynamic problem (Morris et al. 2008, 2009; Allen and Rendall 2013).

The fidelity of results obtained by the optimization process are dependent on the fidelity and quality of each of the three individual components of the ASO process; optimization algorithm, shape parameterization and aerodynamic model. To facilitate optimum compatibility between these components, each is often designed in a modular manner such that, for example, the aerodynamic model is independent of the parameterization scheme used. A high-fidelity numerical aerodynamic model with good capture of the true physics is important in producing optimum aerodynamic designs, particularly at transonic conditions. The aerodynamic model also defines the parameter space of the problem, which is the definition of the aerodynamic outputs based on flow field inputs such as Mach number and angle of attack.

The quality of the optimization result obtained is driven, primarily, by the quality and type of numerical optimization algorithm used in the ASO framework, and the two primary types of optimization algorithms are local methods and global methods. The local methods are usually built around the gradient-based approach, which uses the local gradient of the design space as a basis around which to construct a search direction. The optimization algorithm therefore traces a movement path through the design space until the gradient values become very small where the result has converged. These approaches are the most common methods used in the ASO framework (e.g., Imiela 2012; Hicken and Zingg 2010; Lyu et al. 2015), driven primarily by the low cost associated with them compared to global methods (Chernukhin and Zingg 2013), and an efficient gradient-based optimizer is used here.

The aerodynamic model defines the parameter space of the problem, but the problem design space, which the optimization algorithm interrogates, is constructed by the definition of a surface parameterization scheme. The ability of the optimizer

to fully interrogate the true design space (which contains every possible design) is driven by the ability of the degrees of freedom of the parameterization scheme to represent any shape within the design space, and so this is a critical aspect of any optimization scheme. The level of flexibility generally increases with the number of design variables, but the use of a low number of design variables is advantageous, since good convergence of optimization algorithms tends to correlate with small numbers of design variables, and so there is a definite requirement for an efficient parameterization scheme.

The work presented in this paper considers aerodynamic shape optimization using a novel method of deriving design variables. The design variables used here are derived by a mathematical technique that is based on singular value decomposition (SVD), that extracts an orthogonal set of geometric ‘modes’. The method itself has been presented recently by the authors (Poole et al. 2015), and has been shown to outperform other commonly-used parameterization schemes (Masters et al. 2017b) when considering geometric inverse design in two dimensions, often requiring fewer than a dozen variables to represent a large design space (Poole et al. 2017).

1.1 Shape parameterization

A surface parameterization scheme defines a design space by a number of design variables. A separate problem to this, though often considered alongside, is the deformation of the subsequent surface during the optimization process, which is required to allow deformation of a body-fitted CFD mesh. An effective parameterization method is (i) flexible and robust enough to cover the design space, and (ii) efficient enough to represent a given shape with as few design variables as possible. Methods are classified as either constructive, deformative or unified. In-depth reviews have been presented by Samareh (2001b), Castonguay and Nadarajah (2007), Mousavi et al. (2007) and Masters et al. (2017b).

Constructive methods consider the definition of the surface and the deformation of the surface separately. Examples of these methods are CST (Kulfan 2008), PARSEC (Sobieczky 1998), PDEs (Bloor and Wilson 1997) and splines (Braubant and Fleury 1984). Other approaches that combine various parameterizations in a hybrid approach, such as that of Zhu and Qin (2014), can also be found. Because of the constructive nature of these approaches, perturbation of the base geometry through the optimization process requires that the new surface be reconstructed, which subsequently requires automatic mesh generation tools for production of a new surface and volume mesh. This extra difficulty can make it advantageous to consider approaches that manipulate an existing mesh.

An alternative to constructive methods are deformative methods which unify the geometry creation and perturbation. This tends to make them simpler to integrate with mesh deformation tools and allows the use of previously generated meshes; a considerably cheaper alternative to regeneration, although the mesh deformation scheme is a separate algorithm. Analytic (Hicks and Henne 1978) and discrete (Jameson 1988) methods are examples of deformative approaches.

A further refinement of unifying geometry creation and perturbation is the integration with a mesh deformation algorithm. Methods of this type typically have some interpolation that describes a link between the surface and volume, often via a set of control points that are independent of both, such that deformation of the control points results in deformation of the surface and CFD mesh. These approaches are commonly used in ASO, and the methods included in this unified category are free-form deformation (Samareh 2001a), domain elements (Morris et al. 2008) and direct manipulation (Yamazaki et al. 2010).

Surface parameterizations developed around the FFD and domain element approach are very popular in wing optimization as this type of approach allows the design space to be reduced from thousands of design parameters to hundreds of design parameters. Such techniques have been developed by Zingg and colleagues (Hicken and Zingg 2010; Leung and Zingg 2012), and have shown that these types of methods can be flexible enough to allow the moulding of a sphere into an aircraft-like shape under certain optimization conditions (Gagnon and Zingg 2012). Further work has been performed by Martins and others (Mader and Martins 2013; Lyu and Martins 2014) who showed results for blended-wing-body optimizations, and Yamazaki et al. (2010) who further reduced the number of design variables by considering the direct manipulation method for wing optimization.

A novel method, recently developed by the authors, is to extract aerofoil design variables using a mathematical approach. The approach utilises singular value decomposition in a manner that analyses an initial library of aerofoils and decomposes that library into a reduced set of optimum variables that are geometrically orthogonal to each another. The method is based on perturbations, so is independent of the initial geometry and can fit into any of the three categories outlined above; the deformative formulation is used in this work, to allow a unified application of the design variables to control both the surface and volume mesh within the ASO framework. Previous work has considered the method's ability to represent a wide-range of aerofoil shapes (Poole et al. 2015; Masters et al. 2017b), and their effectiveness in aerofoil optimisation (Poole et al. 2017; Masters et al. 2017a), wherein the efficiency of this modal approach was clearly demonstrated. Hence, the aim of the work presented here is to develop an effective method to apply these novel mathematically-extracted design variables, which have been extracted as two-dimensional quantities, in three dimensions, and determine their effectiveness when applied to aerodynamic optimization, in particular drag minimisation of wings in transonic flow.

2 Shape deformations by singular value decomposition

The derivation of aerofoil perturbation modes come from a singular value decomposition of a training library of aerofoils. The resulting modes, which form aerofoil design variables used in this work for ASO, are guaranteed to be orthogonal (scalar product of any two modes is zero), meaning a given aerofoil shape is described uniquely by a given set of input parameters. This alleviates some multimodality that can be introduced numerically by the given parameterization

scheme, and expands design space coverage (Castonguay and Nadarajah 2007). An alternative to deriving design variables by a direct decomposition approach is to manipulate already existing ones by Gram–Schmidt orthogonalisation. This can be used to force orthogonality (Chang et al. 1995; Robinson and Keane 2001), however, it is ideal to use the SVD method to guarantee orthogonal modes and provide a low-dimensional approximation (modal parameters) to a high-dimensional design space (full training library). Initial studies of using the SVD method to derive design variables were performed by Toal et al. (2010) and Ghoman et al. (2012), and further studies were carried out by the authors for geometric shape recovery (Poole et al. 2015; Masters et al. 2017b) and aerofoil optimisation (Poole et al. 2017; Masters et al. 2017a). The work presented here develops more fully the use of mathematically-derived modes for performing aerodynamic shape optimization in three dimensions.

It is worth considering the result of an SVD decomposition. A matrix is decomposed into constituent matrices where the dominant features of the input matrix are ordered. Hence, the SVD can be used to project a reduced-order basis approximation to produce a low-rank approximation to the original matrix. Eckart and Young (1936) showed that, given a low rank approximation found through SVD, $\mathbf{M}^{(k)}$, of a full rank input matrix, \mathbf{M} , the following is true:

$$\|\mathbf{M} - \mathbf{M}^{(k)}\|_F \leq \|\mathbf{M} - \mathbf{m}\|_F \quad (1)$$

where \mathbf{m} is any matrix of rank k and $\|\cdot\|_F$ is the Frobenius norm. Hence, the error in the low rank approximation (found from SVD) will always be at least as good as the error between any other rank k matrix and the full rank matrix. The SVD thus produces an optimal low order projection of the higher dimensional space into the lower dimensional one, which is significant for optimisation parameters.

The SVD method first requires a training library of N_a aerofoils to be collated from which the aerofoil deformation modes are extracted. Each aerofoil surface is parameterized by N surface points, where the i th surface point has a position in the space (x_i, z_i) . To ensure consistency of the surface description of the training data all aerofoils are parameterised with the same parametric distribution. The x distribution is often defined as the controlling parameterisation, with $z_i = f(x_i)$, but this is not the most flexible approach; instead all aerofoil surfaces are parameterised in terms of peripheral distance $s \in [0, 1]$, and then exactly the same s_i distribution is defined for all aerofoils. Following the surface point distribution, each aerofoil has a rigid body translation, scaling and then rotation applied to it to map the geometry into a consistent form where each section has unit chord and $z(0) = z(1) = 0$. A matrix is built from which SVD is performed, by evaluating the vector difference of the i th surface point between all aerofoils, producing $N_{def} = N_a(N_a - 1)/2$ aerofoil deformations. The x and z deformations are stacked into a single vector of length $2N$, for each aerofoil deformation, so a matrix is built of the aerofoil deformations which has $2N$ rows and N_{def} columns:

$$\mathbf{M} = \begin{pmatrix} \Delta x_{1,1} & \cdots & \Delta x_{1,N_{def}} \\ \vdots & \ddots & \vdots \\ \Delta x_{N,1} & \cdots & \Delta x_{N,N_{def}} \\ \Delta z_{1,1} & \cdots & \Delta z_{1,N_{def}} \\ \vdots & \ddots & \vdots \\ \Delta z_{N,1} & \cdots & \Delta z_{N,N_{def}} \end{pmatrix} \quad (2)$$

Performing a SVD decomposes the matrix into three constituent matrices:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3)$$

where \mathbf{U} is a matrix of vectors, each of length $2N$. The structure is analogous to the decomposed matrix, so the columns of this matrix are the aerofoil mode shapes. $\mathbf{\Sigma}$ is a diagonal matrix of the singular values, arranged in descending order. These can be considered the ‘relative energy’ of the modes, and represent the ‘importance’ of the mode shapes in the original library. The total number of possible mode shapes is governed by the number of singular values, which is the minimum of the number of columns and rows of the decomposed matrix. A truncation of the \mathbf{U} matrix, based on a certain total energy required, then gives the number of design variables used in the optimization. The training library is based on deformations, and this is an important choice such that design variables that result from the decomposition are also deformations, ensuring they are independent of the topology of the aerofoils that are used. This allows direct insertion into an aerodynamic shape optimization framework where deformation of the surface and mesh is important. If the constructive formulation is used, however, then the columns of the training matrix, \mathbf{M} , are absolute positions of the aerofoil surface points as opposed to deformations between surface points.

In this work, a generic, nonsymmetric training library is considered based on the optimization being performed. The library contains 100 different aerofoils, extracted from a larger library by quantifying their performances in the transonic regime using the Korn technology factor (Poole et al. 2015). The first six modes of the library are shown in Fig. 1; all modes are scaled up for illustration purposes, and have been added to a NACA0012 section. Also shown, in Fig. 2, is the relative

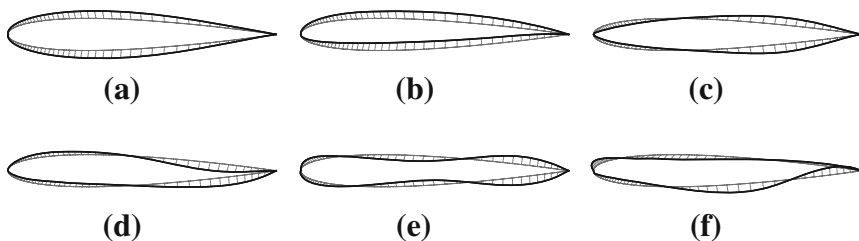
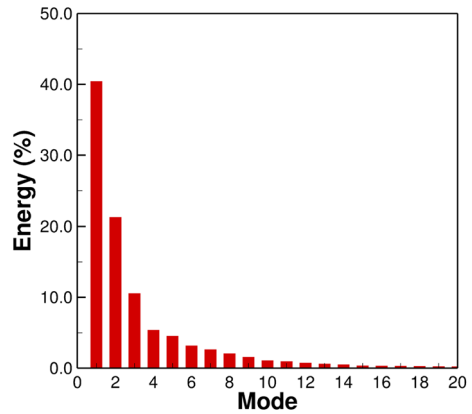


Fig. 1 Generic nonsymmetric aerofoil modes. **a** Mode 1. **b** Mode 2. **c** Mode 3. **d** Mode 4. **e** Mode 5. **f** Mode 6

Fig. 2 Modal energy of first 20 modes



‘energy’ of the first 20 modes, i.e. the singular values of each mode normalised by the sum of all singular values.

Once the design variables have been extracted and the total number of modes has been truncated, a new aerofoil can be formed by a weighted combination of m modal parameters, as shown in Eq. (4). The weighting vector, β , represents the magnitude of the modal deformations which are then the design variable values that the optimizer works with. The truncation of the total number of modes, which is often very large, down to a number which is useful for the optimization can either be user-specified or based on the requirement for a total amount of energy to be preserved, e.g., if 99.0% of the energy of the original library is required to be preserved then the first, say, six modes may cumulatively have 99.1% of the energy so six modes would be used. In this work, a number of modes is specified and those modes with the highest amount of energy are taken.

$$\mathbf{X}^{new} = \mathbf{X}^{old} + \sum_{n=1}^m \beta^n \mathbf{U}^n \quad (4)$$

The modes extracted here are two-dimensional deformations, based on a large database of aerodynamic surfaces, and so are effective in two dimensions, and this has been proven previously (Poole et al. 2017). Hence, it would make sense to consider a similar approach in three dimensions. However, this would require a database of wings, something that would not be easy to create, and with variable parameters such as taper and sweep, and surface discontinuities such as crank locations, would also require a complex parametric transformation to a normalised space. Furthermore, this would still not contain global variables, and so it is more sensible and flexible to consider a more local sectional approach. This is the approach considered here.

2.1 RBF coupling of point sets for aerofoil deformation

The aerofoil design variables must be coupled to a control point-based approach to allow flexible deformation of the CFD mesh. The control point method links

deformations of the CFD mesh to deformations of a small set of control points on or near the surface. At the centre of this technique is a multivariate interpolation using radial basis functions (RBFs), which provides a direct mapping between the control points, the surface geometry and the locations of grid points in the CFD volume mesh. The approach is meshless, so requires no connectivity and is applicable to any mesh type; control points and volume mesh points are simply treated as independent point clouds. The system is only the size of the number of control points, and so is not related to the mesh size.

The general theory of RBFs is presented by Buhmann (2005) and Wendland (2005), and the basis of the method used here is described in detail by Rendall and Allen (2008). If ϕ is the chosen basis function and $\|\cdot\|$ is used to denote the Euclidean norm, then a general volume interpolation model s has the form

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) \quad (5)$$

where $i = 1, \dots, n$ denotes the n control points, $\alpha_i, i = 1, \dots, n$ are model coefficients, \mathbf{x} is the vector coordinate, and $p(\mathbf{x})$ is an optional polynomial. The coefficients are found by requiring exact recovery of the original data, $\mathbf{s}_{\mathcal{X}} = \mathbf{f}$, for all points in the training data set \mathcal{X} . Hence the model is an interpolant, and all original solution information is preserved.

Control points (sometimes named domain element points) are used here to decouple the shape parameters from the surface mesh, and provide a flexible framework through which to control the shape of a base geometry. Setting up a global RBF volume interpolation for n_c control points then requires a solution to a linear system [see Morris et al. (2008)] for more details] to ensure exact recovery of the control point data, in this case deformations:

$$\Delta \mathbf{X}^c = \mathbf{C} \boldsymbol{\alpha}^x \quad (6)$$

$$\Delta \mathbf{Y}^c = \mathbf{C} \boldsymbol{\alpha}^y \quad (7)$$

$$\Delta \mathbf{Z}^c = \mathbf{C} \boldsymbol{\alpha}^z \quad (8)$$

Polynomials are not included here, due to their growing radial influence, and so (superscript c represents a control point):

$$\Delta \mathbf{X}^c = \begin{pmatrix} \Delta x_1^c \\ \vdots \\ \Delta x_{n_c}^c \end{pmatrix}, \quad \boldsymbol{\alpha}^x = \begin{pmatrix} \alpha_1^x \\ \vdots \\ \alpha_{n_c}^x \end{pmatrix} \quad (9)$$

(analogous definitions hold for y and z coordinates) and the control point dependence matrix, \mathbf{C} , takes the form

$$\mathbf{C} = \begin{pmatrix} \phi_{11} & \cdots & \phi_{1n_c} \\ \vdots & \ddots & \vdots \\ \phi_{n_c 1} & \cdots & \phi_{n_c n_c} \end{pmatrix} \quad (10)$$

where

$$\phi_{ij} = \phi(\|\mathbf{x}_i^c - \mathbf{x}_j^c\|) \quad (11)$$

For surface and volume mesh deformation, it is sensible to use decaying basis functions, to give the interpolation a local character and ensure deformation is contained in a region near the moving body, and Wendland's C^2 function (Wendland 2005) is used here. It is also sensible to omit polynomial terms, since these will transfer deformation throughout the entire mesh.

Hence, in the case considered here the global influence on any point in the aerodynamic mesh (denoted by superscript a) from the control points is determined by Eq. (5), which is applied as

$$\Delta x^a = \sum_{i=1}^{n_c} \alpha_i^x \phi(\|\mathbf{x}^a - \mathbf{x}_i^c\|) \quad (12)$$

$$\Delta y^a = \sum_{i=1}^{n_c} \alpha_i^y \phi(\|\mathbf{x}^a - \mathbf{x}_i^c\|) \quad (13)$$

$$\Delta z^a = \sum_{i=1}^{n_c} \alpha_i^z \phi(\|\mathbf{x}^a - \mathbf{x}_i^c\|) \quad (14)$$

Hence, the design variables are the modal deformations, which give control point perturbations, which hence are decoupled from the surface and volume meshes.

2.2 Control point deformations

The method for deriving surface design parameters and the methods for perturbing the CFD mesh have been presented. The derived parameters are, however, surface deformations whereas for the aerodynamic optimization process, control point parameters are required. Previous work has involved placing control points away from the surface, to form off-surface domain elements, and this has proven very effective, and is used again for the three-dimensional case later. In two dimensions, the control points to define the modal deformations are located on the surface of the aerofoil section. This ensures that there is direct coupling between the control point deformations and the surface deformations that derived them.

The deformation modes derived here by SVD are extracted from a training library of aerofoils. A complete library of aerofoils is quantified in terms of aerodynamic performance, using the Korn technology factor, and a form of library filtering applied to down-select the library; see Poole et al. (2015). A set of control points is used to control the aerodynamic surface; these points are independent of a

base geometry, and the surface deformations are defined in terms of perturbations so the modes can be applied to any geometry. A ‘shrink-wrapping’ method is used to map them onto the geometry being considered. Here, 24 control points are used; more than this are not necessary unless small wavelength changes are required. The modal deformations can be defined using a larger number of points than the N value in Eq. (2) and projected onto these 24 points, but here $N = 24$ is used in the SVD extraction process. Figure 3 shows the surface control points and an example deformation of the fourth mode for a NACA0012 mesh.

2.3 Computation of deformation field in two dimensions

The modal deformations can be applied to any geometry, and are extracted using a training library wherein all aerofoils have been normalised to unit chord and all have leading and trailing edges at $z(0) = z(1) = 0$. Hence, the modal perturbations are all extracted from these geometries, but since the surface that the modes are added to will not have leading and trailing edges at $z = 0$, each mode needs to be transformed to the local aerofoil axis system. A local rotation matrix is thus used to rotate each mode. All deformations are computed for the 24 control points, and added to the initial aerofoil defined at zero incidence, so there is a deformation due to rigid rotation and that due to the modal parameters. In two dimensions, deformation is computed at each control point, i , by:

$$\Delta \mathbf{x}_i^c = (\Delta x_i^c, 0, \Delta z_i^c)^T = (\mathbf{R} - \mathbf{I})(\mathbf{x}_i^c - \mathbf{x}^r) + \mathbf{R} \sum_{n=1}^m \beta^n \Delta \mathbf{x}_i^n \quad (15)$$

where \mathbf{R} is the rotation matrix which is computed using the total incidence, including the initial section incidence and any incidence change due to the pitch design variable, $\alpha^{total} = \alpha^0 + \alpha^{pitch}$, \mathbf{x}^r is the rotation centre, m is the number of modal design parameters, i.e. modes, β^n are the design parameters, and $\Delta \mathbf{x}_i^n$ is the modal deformation of point i for mode n . Once the deformation vector has been evaluated for every control, the $\Delta \mathbf{X}$ and $\Delta \mathbf{Z}$ vectors are known ($\Delta \mathbf{Y} = 0$ in two dimensions), Eqs. (6) and (8) can be solved, and the deformation of all mesh nodes, including those on the surface, is evaluated using Eqs. (12) and (14).

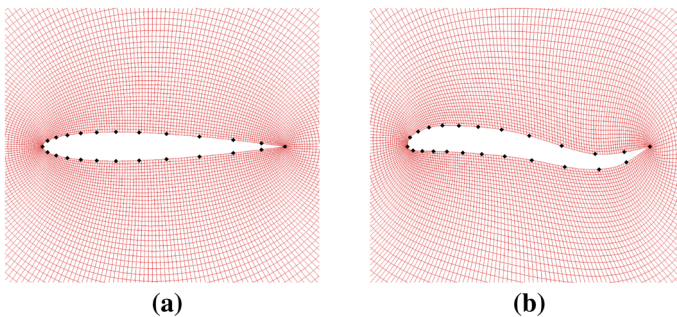


Fig. 3 Surface-based control points and example deformation. **a** Control points. **b** Example deformation

2.4 Computation of deformation field in three dimensions

In three dimensions, a set of n_s sectional slices of control points are applied to the surface at regular intervals. However, when these are deformed, the variation of the deformation field between the sections can either be defined explicitly or left to the global interpolation field. The latter is normally used, but this means that interpolation properties, for example the basis function chosen and the support radius set, will influence the deformed surface. That effect is undesirable, so it is eliminated here, as it can result in a more global influence of an effectively local deformation. Intermediate sections are thus defined between each deformed slice, and the deformation of these is controlled analytically. The spanwise region between each section is split into n_{int} intermediate regions, and so the total number of sections becomes $1 + (n_s - 1)n_{int}$.

The geometry considered here is the MDO wing (Allwright 1996; Haase et al. 2002) (see later). The surface is preprocessed to compute the local chord length at each section, i.e. c_j , and the initial rotation angle of each section, α_j^0 , where j is the section location. The control point sections are then applied to the surface by scaling by local chord, rotating by local incidence, and shrink-wrapping to the exact geometry using a local geometric intersection algorithm. Figure 4 shows the control points resulting from using $n_s = 10$ and $n_{int} = 4$, for the surface mesh used later. This means there are 37 control point sections but only 10 are deformed by the design parameters. The deformed points are shown in green, and the controlled points in black.

Consider first the deformation field for global application of the modal parameters. In this case the modal deformations are applied using a single global weighting, i.e. one design variable for each mode. As with the two-dimensional approach, all deformations are computed at each sectional set of 24 control points defined at zero incidence, and so there is deformation due to rigid rotation and that due to modal parameters. Global pitch and twist variables are used later. In three dimensions the modal deformations must be scaled by the local chord as well as being rotated by local section incidence, and so to compute the deformation field at the 24 control points, i , at section j :

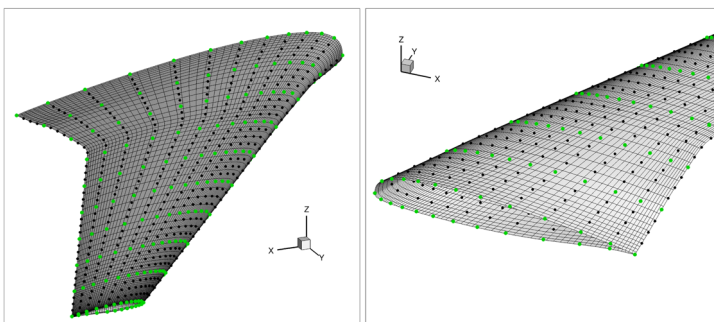


Fig. 4 Surface mesh and control points.

$$\Delta \mathbf{x}_{ij}^c = (\mathbf{R}_j - \mathbf{I}) (\mathbf{x}_{ij}^c - \mathbf{x}_j^r) + c_j \mathbf{R}_j \sum_{n=1}^m \beta^n \Delta \mathbf{x}_i^n \quad (16)$$

where $\mathbf{R}_j = \mathbf{R}(\alpha_j^0 + \alpha_j^{twist} + \alpha_j^{pitch})$ and \mathbf{x}_j^r is the local rotation centre. Hence, in this case there are m design parameters.

For local deformations, i.e. one design variable for each mode at each of the n_s sections, this can be formulated as:

$$\Delta \mathbf{x}_{ij}^c = (\mathbf{R}_j - \mathbf{I}) (\mathbf{x}_{ij}^c - \mathbf{x}_j^r) + c_j \mathbf{R}_j \sum_{s=1}^{n_s} \phi(j, s) \sum_{n=1}^m \beta_j^n \Delta \mathbf{x}_i^n \quad (17)$$

where $\phi(j, s)$ is a basis function. In this case there are $m \times n_s$ design parameters.

Hence, this basis function can be used to determine how the deformation of each of the n_s sections affects the other sections, i.e. controls the zone of influence. This can be left to the global interpolation, but is defined here to allow control of the decay. A basis function can be defined such that if the effect of the sectional deformation decays to zero at the neighbouring sections each side, a global modal deformation can be recovered exactly. In this case β^n would have a single value for all sections, and so it can be shown that if

$$\sum_{s=1}^{n_s} \phi(j, s) = 1 \quad (18)$$

at any spanwise point, Eqs. (15) and (16) are equivalent. Hence, a trigonometric function of (j, s) is used.

Figure 5 shows the control locations and resulting surface mesh for deformations using the first, third, and fifth modes; the upper row shows a global modal deformation, and the lower row shows local modal deformations of the fifth control point section. The modal deformation magnitude is exaggerated to 10% local chord for illustration purposes.

This improved localisation process is also adopted to improve the application of off-surface domain element perturbations, used previously by the authors. Figure 6 shows two views of the surface and domain element points, again using $n_s = 10$ and $n_{int} = 4$. An exaggerated movement of all the points on the fifth section is shown; magnitude 20% local chord. Figure 7 shows the resulting control points and surfaces with and without the intermediate points and basis function decay control; the initial surface is also shown in red, demonstrating clearly how the deformation has been localised by the improved method. Again this means global deformations can be recovered by a combination of local deformations.

3 Optimization approach

Typically, the two main types of numerical optimization algorithm that are chosen for aerodynamic optimization are gradient-based and global search. Gradient-based methods, such as conjugate gradient and sequential quadratic programming (SQP),

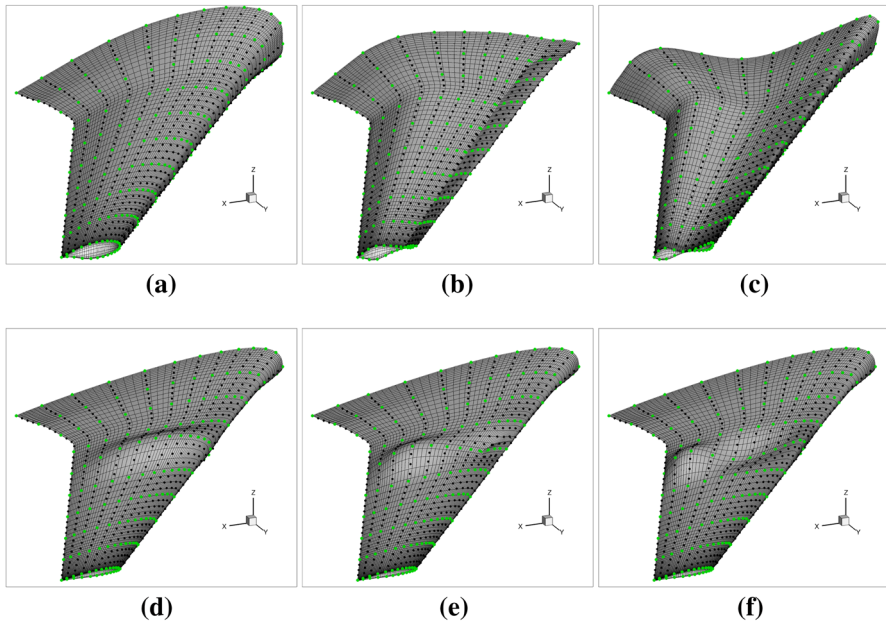


Fig. 5 Surface and control point modal deformations. **a** Mode 1 global. **b** Mode 3 global. **c** Mode 5 global. **d** Mode 1 local. **e** Mode 3 local. **f** Mode 5 local

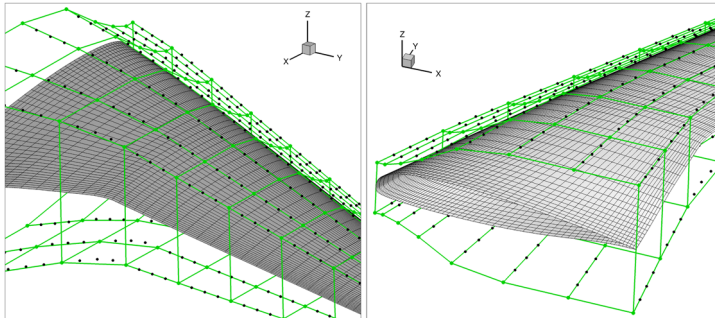


Fig. 6 Surface mesh and off-surface control points

use the local gradient as a basis from which to construct a search direction. The algorithm starts at an initial solution and marches towards the minimum solution. Global search methods, however, use a number of agents with different starting positions within the search space. These agents then cooperate and move by various, often nature-inspired, mechanisms towards the global optimum solution.

The selection of a gradient-based or global search algorithm for aerodynamic optimization is highly dependent on the optimization case analysed, specifically the degree of modality present in the situation. Multimodal problems are characterised by multiple local optima, where one or more of those local optima is the globally optimum solution. This can be particularly problematic for gradient-based

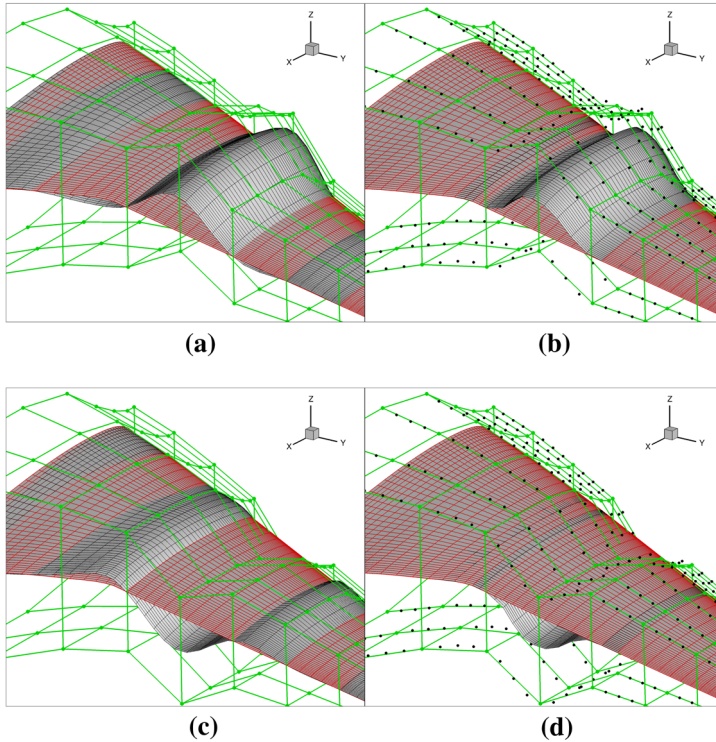


Fig. 7 Deformed and initial surface mesh and control points. **a** Positive dihedral mode, original. **b** Positive dihedral mode, localized. **c** Negative dihedral mode, original. **d** Negative dihedral mode, localized

optimizers due to premature convergence in a local minimum that is not necessarily close to the global optimum. Agent-based methods can alleviate this issue somewhat. Within the context of aerodynamic shape optimization, the presence of a multimodal search space is highly dependent on the extent of the surface representation and the fidelity of the flow analysis tool. The issue of degree of multimodality in aerodynamic optimization problems is an unanswered question, with work presented showing that multimodality exists in a number of cases, but unimodal cases also exist (Namgoong et al. 2002; Khurana et al. 2010; Buckley et al. 2010; Chernukhin and Zingg 2013). Chernukhin and Zingg (2013) have considered this issue by testing a number of different optimization problems and have shown that for a b-spline parameterization of the surface, viscous, compressible drag minimization of the RAE2822 aerofoil has one global optimum. They also showed multiple local optima for other three-dimensional problems.

For maximum flexibility and efficiency, a gradient-based method is used here, with a second-order finite-difference approach for gradient evaluation. This approach allows a ‘wrap-around’ approach, i.e. any flow-solver can be implemented within the framework.

3.1 Feasible sequential quadratic programming (FSQP)

The feasible sequential quadratic programming (FSQP) algorithm is used here as implemented in version 3.7 (Zhou et al. 1997). FSQP is based on an SQP method, which is an approach for constrained gradient-based optimization. It is constructed with a number of modifications to the conventional SQP method to avoid the so-called ‘Maratos’ effect (restriction of a step size due to the requirement of feasibility) (Maratos 1978). The modifications include a number of strategies, the primary one being combining a search along an arc (Mayne and Polack 1982) with a nonmonotone procedure for that search (Grippo et al. 1986). The FSQP algorithm is briefly outlined below, and fully described and analysed in Panier and Tits (1991) and Bonnans et al. (1992).

At every major iteration, t , the design vector, β , at the next iteration is given by:

$$\beta(t + 1) = \beta + a\Delta\beta + a^2\overline{\Delta\beta} \tag{19}$$

where a is the step length, $\Delta\beta$ is the line step direction and $\overline{\Delta\beta}$ is a correction direction used to create a search arc. To find the line step direction, FSQP solves a quadratic programming (QP) subproblem. Considering inequality constraints only, this QP subproblem at every major iteration is:

$$\begin{aligned} \underset{\Delta\beta_{sqp} \in \mathbb{R}^n}{\text{minimise}} & \frac{1}{2} \Delta\beta_{sqp}^T \mathbf{H} \Delta\beta_{sqp} + \nabla J(\beta)^T \Delta\beta_{sqp} \\ & \nabla g_i(\beta)^T \Delta\beta_{sqp} + g_i(\beta) \leq 0 \quad i = 1, \dots, G \end{aligned} \tag{20}$$

where J is the objective function and g_i is the i th inequality constraint of a total of G inequality constraints. FSQP augments the SQP descent direction by a feasible step direction, $\Delta\beta_f$, that is a fraction of either $\nabla J(\beta)$ or $\nabla g_i(\beta)$, depending on the constraint value. The overall step direction is then a blend of the SQP and feasible step directions:

$$\Delta\beta = (1 - \rho)\Delta\beta_{sqp} + \rho\Delta\beta_f \tag{21}$$

where $\rho \in [0, 1] = O(\|\Delta\beta_{sqp}\|^2)$ such that as a solution is approached, ρ tends very quickly to zero to enable the fast convergence of the pure SQP step direction to be inherited (Bonnans et al. 1992). The correction direction is found such that both descent and feasibility are ensured by solving a further quadratic programme while avoiding the need for further constraint and function evaluations. The exact implementation of the rules that govern the computation of the step size are given in Zhou et al. (1997).

The Hessian, or an approximation to the Hessian, at every major iteration is required, which in turn requires sensitivity of the objective function and constraints with respect to the design variables. The Hessian is approximated by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update scheme where the Hessian approximation is initialised as the identity matrix. The gradients are obtained by a second-

order central-difference scheme, so the number of objective function evaluations is proportional to the number of design variables.

Once the search arc has been determined, the nonmonotone line search proceeds (Zhou and Tits 1993). For this, the conventional backtracking line search requirement of requiring a suitable reduction in the objective function from iteration t to $t + 1$ is relaxed, such that a reduction in the objective function to iteration $t + 1$ is required against the maximum objective function from the last four iterations. This has been shown to be highly effective for unconstrained optimization when compared to a conventional backtracking search (Grippio et al. 1986), and when implemented for FSQP shows similar results for constrained optimization (Zhou and Tits 1993).

The algorithm iterates until the Kuhn–Tucker conditions are satisfied, which then represent a converged solution using a constrained gradient-based optimizer.

For computational efficiency, the sensitivity evaluation has been parallelised based on the number of design variables such that the evaluation of the sensitivity of the objective function and constraints with respect to the design variables is split between the number of CPUs available (Morris et al. 2008, 2009). This is necessary as within the ASO environment, an objective function evaluation represents a CFD solution, so this formulation allows parallel evaluation of the required sensitivities; second-order finite-differences are used for the sensitivities. It is well known that the accuracy of the gradient evaluation is a critical issue, and the authors have performed several studies on perturbation size for finite-differences; see for example Morris et al. (2008). A relative perturbation of 10^{-4} is adopted here, i.e. a deformation magnitude of 0.01% of local chord. Constraint and step-size evaluations and optimizer updates occur in the master process, and each CPU controls the geometry (and CFD volume mesh) perturbations corresponding to the different design variables, and calls the flow solver. Flow-solver results are then returned to the master for optimizer updates.

3.2 Flow solver

The flow-solver used is a structured multiblock finite-volume code, with upwind spatial discretisation, using the flux vector splitting of van Leer (1982), and multi-stage Runge–Kutta time-stepping. Convergence acceleration is achieved through multigrid (Allen 2002).

4 Application of modal design variables in three dimensions

Optimization is applied here to the MDO wing [a large modern transport aircraft wing, the result of a previous Brite–Euram project (Allwright 1996; Haase et al. 2002)] in the economical transonic cruise condition.

4.1 Problem definition

The economical cruise flight Mach number for the MDO wing defined by Allwright (Allwright 1996; Haase et al. 2002) is 0.85, with the wing trimmed to obtain a lift coefficient of 0.452. This design case is well-suited to inviscid flow analysis, since induced and wave drag are dominant here. Compressible transonic wing optimization for drag minimization subject to strict constraints is investigated, and so the problem definition is:

Objective :	Minimizedrag(C_D)
Constraint1(lift) :	$C_L \geq C_L^0$
Constraint2(moment) :	$C_{Mx} \leq C_{Mx}^0$
Constraint3(moment) :	$C_{My} \leq C_{My}^0$
Constraint4(internalvolume) :	$V \geq V^0$

A 688,000 cell, eight-block structured C-mesh was generated (Allen 2008); 129×81 surface mesh, 33 points on either side of the wake, 33 points in the tip-slit, and 33 points between inner and outer boundary. Figure 8 shows domain and boundaries boundaries and farfield mesh, and Fig. 9 gives two views of the surface mesh and chordwise planes.

In previous work the authors have applied a 16-point off-surface domain element for an aerofoil, and a set of section-based domain elements for a wing, which has been shown to be very effective (Morris et al. 2008, 2009). Hence, an improved version of this approach, implementing the localization method, is used here as a comparison with the new method; the 24-point on-surface set of control points used in two dimensions is again used here. The same evenly-distributed set of slices is used as above, but the points at each slice are ‘shrink-wrapped’ to the local surface. Figure 6 shows two views of the located control point spanwise locations.

Optimizations of the MDO wing were run using four sets of design variables, all with the parallel FQSP optimizer, and are detailed below. The drag comprises pressure, induced, and wave drag components, and it has been found to be most

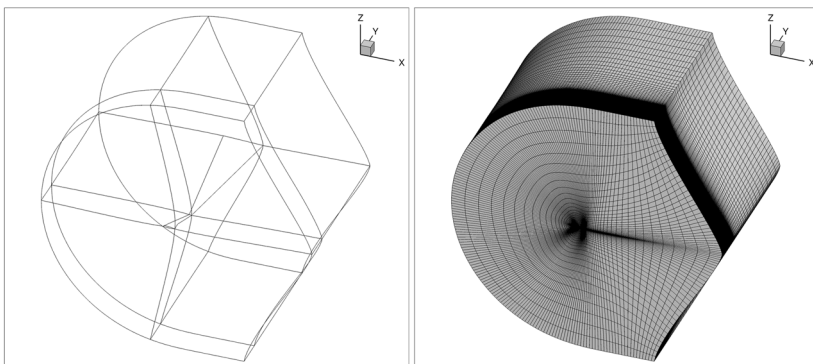


Fig. 8 Domain and block boundaries and farfield mesh

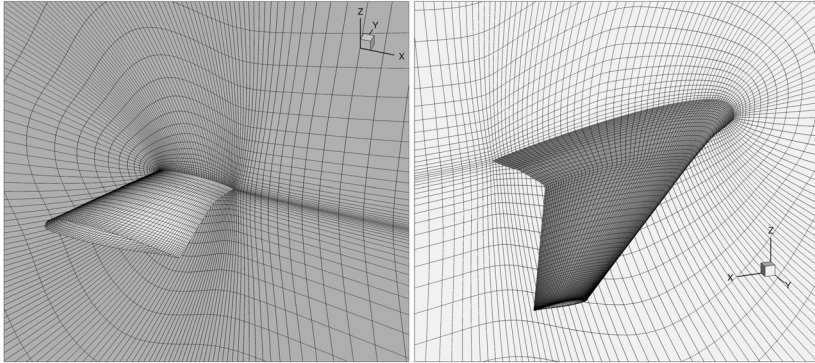


Fig. 9 Surface and selected mesh planes

efficient to address these separately since, with a gradient-based approach, the twist variable can dominate the sensitivities. Hence, the induced drag was considered by running a twist-only optimization first, and optimizations to minimize the remaining drag restarted from this geometry; the restarted cases still included the two twist variables.

1. *Twist case* To address the induced drag effect, a simple case was first run using a global linear twist variable, plus a global pitch variable to allow lift balancing. This results in two variables.
2. *Individual point deformation case* Conventional off-surface domain element, with individual deformations of each point, normal to the local chord, in each of the 10 slices, plus a global linear twist variable and a global pitch variable to allow lift balancing. This results in $10 \times 16 + 2 = 162$ variables.
3. *Global mode case* Global modal deformations of all 10 sectional slices using 6, 8, and 10 modes, a global linear twist variable, plus a global pitch variable to allow lift balancing. This results in $6 + 2, 8 + 2$ or $10 + 2 = 8, 10$ or 12 variables. A global mode is a single deformation of all control points, with the modes scaled and rotated according to the local geometry.
4. *Local mode case* Local modal deformation using 6, 8, and 10 modes at each of the 10 sectional locations, a global linear twist variable, plus a global pitch variable to allow lift balancing. This results in $10 \times 6 + 2, 10 \times 8 + 2$ or $10 \times 10 + 2 = 62, 82$ or 102 variables. Again at each section, the local modes are scaled and rotated according to the local geometry. Global modes are not included, since these can be recovered exactly from a combination of the local modes.

4.2 Results

Table 1 presents results for the four sets of variables. The twist variables are clearly effective at reducing the induced drag, and the finer surface deformations then

Table 1 Optimization results (C_D in counts)

Variables	Parameters	C_L	C_D	$\Delta C_D(\%)$	Iterations	CPU time/h	Run time/h (cores)
Baseline geometry		0.452	153.8	–	–	–	–
Twist	2	0.452	142.2	– 7.5	20	18.3	11.8 (3)
Domain element	162	0.452	108.6	– 29.4	114	6222.0	85.5 (163)
Modes 6 global	8	0.452	112.7	– 26.7	41	119.3	23.9 (9)
Modes 8 global	10	0.452	111.3	– 27.6	43	155.5	26.5 (11)
Modes 10 global	12	0.452	110.3	– 28.3	52	222.7	32.0 (13)
Modes 6 local	62	0.452	106.9	– 30.5	75	1572.5	47.5 (63)
Modes 8 local	82	0.452	105.8	– 31.2	71	1962.0	45.0 (83)
Modes 10 local	102	0.452	104.7	– 31.9	82	2912.6	51.9 (103)

reduce the pressure and wave drag. Figure 10 shows the upper surface pressure contours for the baseline case, and optimizations using 16-point domain element deformations at each section, 10 global modes, and 10 local modes. Sectional pressure coefficient variations are also presented in Fig. 11 (results for 10 global and local modes are shown). Hence, all three of the local mode cases have produced shock-free solutions. The slight increase in drag reduction with increasing local modes is then due to increased flexibility to improve the curvature at the leading and trailing edge to reduce pressure drag.

The convergence histories in terms of iterations and function evaluations are shown in Fig. 12 in terms of the objective function convergence. Figure 13 shows the convergence of the objective function and all the constraints, so it is clear which constraints are active through the optimization; this is the 10 local mode case, but the other cases show similar behaviour.

Also shown in Table 1 is the total CPU time, i.e. the number of objective evaluations (flow solutions) multiplied by run-time per solution. All cases were run on the University of Bristol HPC cluster, comprising Intel Sandy Bridge 2.6 GHz cores. Also shown in the table are optimization run times and the number of cores used for each. Note that the costs presented do not include the cost of the twist-only case run first. The optimizer adopts a second-order central finite-difference gradient evaluation, and so each iteration requires two flow solutions per variable, and a further one or two solutions for the step size evaluation, and the step size evaluation is always performed in serial on the master node. All cases were run with one core per design variable and one for the master process. Hence, the total cost scales with the number of design variables, but the parallel framework means the optimization run time scaling can be reduced to the number of iterations.

It is clear that the global modes are particularly efficient, requiring significantly fewer evaluations than the off-surface domain element, for similar drag reduction. However, neither of these approaches has eliminated the wave drag entirely,

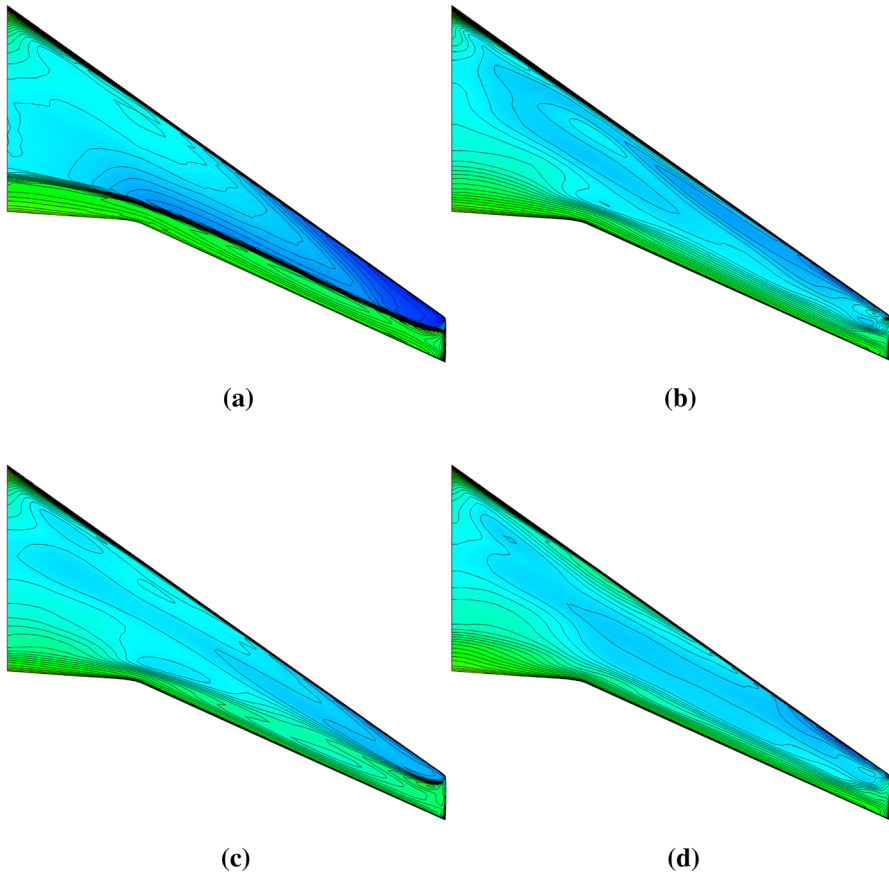


Fig. 10 Upper surface pressure coefficient. **a** Initial geometry. **b** Domain element. **c** 10 Global modes. **d** 10 Local modes

whereas the local modes have achieved this for significantly lower cost than the domain element approach.

Figure 14 shows the initial and optimized surface at various locations along the span from root to tip, for 10 local modes. The chord has been normalised and the figure shows the z coordinate scaled by two. The loading distributions are also presented, along with the elliptical value (10 local and global modes case shown), showing an improved distribution.

5 Conclusions

Aerodynamic shape optimization has been considered, using mathematically-derived design variables. Orthogonal design variables have been extracted by a singular value decomposition approach where a training library of aerofoils is analysed and decomposed to obtain an efficient and reduced set of design variables;

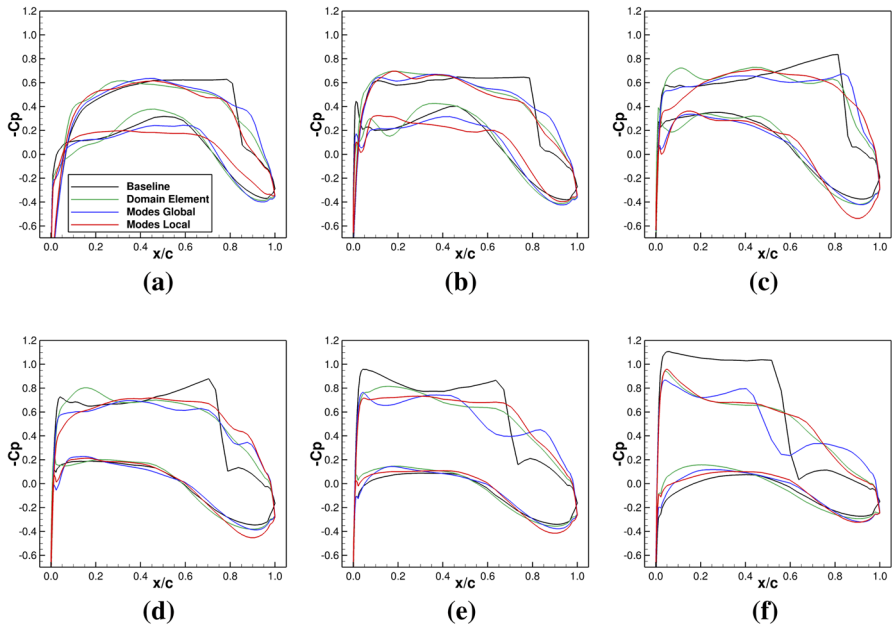


Fig. 11 Sectional pressure coefficient: 10 local and 10 global mode cases shown. **a** $y/s = 0.0$. **b** $y/s = 0.1$. **c** $y/s = 0.3$. **d** $y/s = 0.5$. **e** $y/s = 0.7$. **f** $y/s = 0.9$

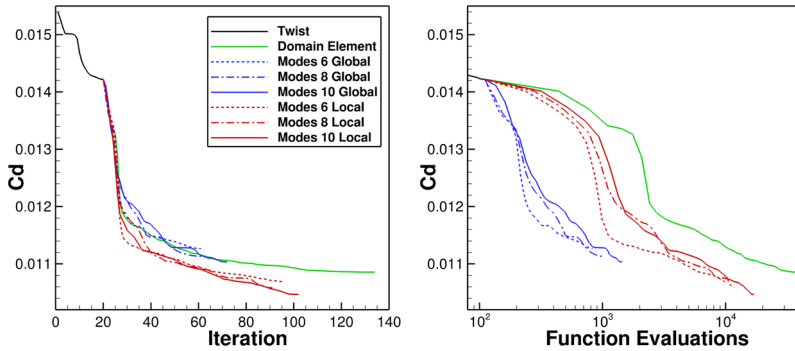


Fig. 12 Convergence histories

they are geometric ‘modes’ of the original library, representing typical airfoil design parameters. In the aerodynamic shape optimization framework a surface and mesh deformation algorithm is required, and a control point approach has been adopted. This adopts a small number of control points which are linked to the numerical mesh points by a global volume interpolation using radial basis functions to allow large, smooth deformations of the mesh.

The performance of the mathematical design variables has been demonstrated in three dimensions, with results of optimization of the MDO wing in transonic flow. The modal deformations have been applied as both local and global variables, and a

Fig. 13 Convergence history of objective and constraints: 10 local modes

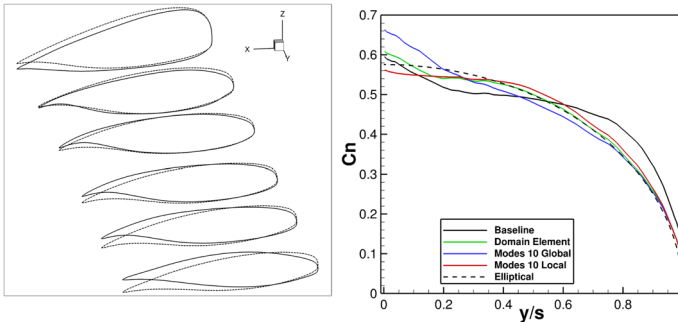
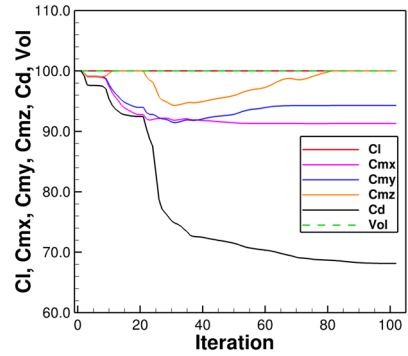


Fig. 14 Initial and optimised surface geometries and loadings

further case run with a previously-used off-surface domain element approach. An important aspect of effective geometric application of these two-dimensional variables is localization of the deformation field. A basis function deformation control approach has been developed and presented, allowing improved local control of deformations, and ensuring exact recovery of global modes from local modes. It has been demonstrated that the modal approach gives better results than the domain element approach, for significantly fewer design variables and, furthermore, using global modes, an impressive result is achieved with only $O(10)$ variables.

Acknowledgements The authors gratefully acknowledge the support of the EPSRC-supported University of Bristol EPSRC Doctoral Training Grant. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol: <http://www.bris.ac.uk/acrc/>.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Allen CB (2002) Multigrid convergence of inviscid fixed- and rotary-wing flows. *Int J Numer Meth Fluids* 39(2):121–140. <https://doi.org/10.1002/fld.282>
- Allen CB (2008) Towards automatic structured multiblock mesh generation using improved transfinite interpolation. *Int J Numer Meth Eng* 74(5):697–733. <https://doi.org/10.1002/nme.2170>
- Allen CB, Rendall TCS (2013) Computational-fluid-dynamics-based optimisation of hovering rotors using radial basis functions for shape parameterisation and mesh deformation. *Optim Eng* 14:97–118. <https://doi.org/10.1007/s11081-011-9179-6>
- Allwright S (1996) Multi-discipline optimisation in preliminary design of commercial transport aircraft. In: *Computational Methods in Applied Sciences*, ECCOMAS, pp 523–526
- Bloor MIG, Wilson MJ (1997) Generating parameterizations of wing geometries using partial differential equations. *Comput Methods Appl Mech Eng* 148:125–138. [https://doi.org/10.1016/S0045-7825\(97\)00033-9](https://doi.org/10.1016/S0045-7825(97)00033-9)
- Bonnans JF, Panier ER, Tits AL, Zhou JL (1992) Avoiding the Maratos effect by means of a nonmonotone line search II. Inequality constrained problems-feasible iterates. *SIAM J Numer Anal* 29(4):1187–1202. <https://doi.org/10.1137/0729072>
- Braibant V, Fleury C (1984) Shape optimal design using B-splines. *Comput Methods Appl Mech Eng* 44(3):247–267. [https://doi.org/10.1016/0045-7825\(84\)90132-4](https://doi.org/10.1016/0045-7825(84)90132-4)
- Buckley HP, Zhou BY, Zingg DW (2010) Airfoil optimization using practical aerodynamic design requirements. *J Aircr* 47(5):1707–1719. <https://doi.org/10.2514/1.C000256>
- Buhmann M (2005) Radial basis functions, 1st edn. Cambridge University Press, Cambridge
- Castonguay P, Nadarajah SK (2007) Effect of shape parameterization on aerodynamic shape optimization. In: 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 10.2514/6.2007-59, AIAA Paper 2007–59
- Chang IC, Torres FJ, Tung C (1995) Geometric analysis of wing sections. Technical report NASA Ames Research Centre, Moffett Field, California, NASA Technical Memorandum, p 110346
- Chernukhin O, Zingg DW (2013) Multimodality and global optimization in aerodynamic design. *AIAA J* 51(6):1342–1354. <https://doi.org/10.2514/1.J051835>
- Choi S, Lee KH, Potsdam M, Alonso JJ (2014) Helicopter rotor design using a time-spectral and adjoint based method. *J Aircr* 51(2):412–423. <https://doi.org/10.2514/1.C031975>
- Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. *Psychometrika* 1(3):211–218. <https://doi.org/10.1007/BF02288367>
- Gagnon H, Zingg DW (2012) Two-level free-form deformation for high-fidelity aerodynamic shape optimization. In: 12th AIAA aviation technology, integration and operations (ATIO) conference and 14th AIAA/ISSMO multidisciplinary analysis optimization conference, Indianapolis, Indiana, <https://doi.org/10.2514/6.2012-5447>, AIAA Paper 2012–5447
- Ghoman SS, Wang Z, Chen PC, Kapania RK (2012) A POD-based reduced order design scheme for shape optimization of air vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference and co-located events, Honolulu, Hawaii, <https://doi.org/10.2514/6.2012-1808>, AIAA Paper 2012–1808
- Grippo L, Lampariello F, Lucidi S (1986) A nonmonotone line search technique for Newton's method. *SIAM J Numer Anal* 23(4):707–716. <https://doi.org/10.1137/0723046>
- Haase D, Selmin V, Winzell B (2002) Progress in computational flow-structure interaction. Springer, New York
- Hicken JE, Zingg DW (2010) Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA J* 48(2):400–413. <https://doi.org/10.2514/1.44033>
- Hicks RM, Henne PA (1978) Wing design by numerical optimization. *J Aircr* 15(7):407–412. <https://doi.org/10.2514/6.1977-1247>
- Imiela M (2012) High-fidelity optimization framework for helicopter rotors. *Aerosp Sci Technol* 23:2–16. <https://doi.org/10.1016/j.ast.2011.12.011>
- Jameson A (1988) Aerodynamic design via control theory. *J Sci Comput* 3(3):233–260. <https://doi.org/10.1007/BF01061285>
- Khurana MS, Winarto H, Sinha AK (2010) Airfoil optimisation by swarm algorithm with mutation and artificial neural networks. In: 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace Exposition, Orlando, Florida, <https://doi.org/10.2514/6.2009-1278>, AIAA Paper 2009–1278

- Kulfan BM (2008) Universal parametric geometry representation method. *J Aircr* 45(1):142–158. <https://doi.org/10.2514/1.29958>
- Leung TM, Zingg DW (2012) Aerodynamic shape optimization of wings using a parallel newton-krylov approach. *AIAA J* 50(3):540–550. <https://doi.org/10.2514/1.J051192>
- Lyu Z, Martins JRRR (2014) Aerodynamic shape optimization studies of a blended-wing-body aircraft. *J Aircr* 51(5):1604–1617. <https://doi.org/10.2514/1.C032491>
- Lyu Z, Kenway GKW, Martins JRRR (2015) Aerodynamic shape optimization investigations of the common research model wing benchmark. *AIAA J* 53(4):968–985. <https://doi.org/10.2514/1.J053318>
- Mader CA, Martins JRRR (2013) Stability-constrained aerodynamic shape optimization of flying wings. *J Aircr* 50(5):1431–1449. <https://doi.org/10.2514/1.C031956>
- Maratos N (1978) Exact penalty function algorithms for finite dimensional and optimization problems. Ph.D. thesis, Imperial College
- Masters DA, Poole DJ, Taylor NJ, Rendall TCS, Allen CB (2017a) Influence of shape parameterisation on a benchmark aerodynamic optimization problem. *AIAA J Aircr* 10(2514/1):C034006
- Masters DA, Taylor NJ, Rendall TCS, Allen CB, Poole DJ (2017b) Geometric comparison of aerofoil shape parameterization methods. *AIAA J* 55(5):1575–1589. <https://doi.org/10.2514/1.J054943>
- Mayne DQ, Polack E (1982) A superlinearly convergent algorithm for constrained optimization problems. *Math Programm Stud* 4:45–61. <https://doi.org/10.1007/BFb0120947>
- Morris AM, Allen CB, Rendall TCS (2008) CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation. *Int J Numer Meth Fluids* 58(8):827–860. <https://doi.org/10.1002/flid.1769>
- Morris AM, Allen CB, Rendall TCS (2009) Domain-element method for aerodynamic shape optimization applied to a modern transport wing. *AIAA J* 47(7):1647–1659. <https://doi.org/10.2514/1.39382>
- Mousavi A, Castonguay P, Nadarajah SK (2007) Survey of shape parameterization techniques and its effect on three dimensional aerodynamic shape optimization. In: 18th AIAA computational fluid dynamics conference, Miami, Florida, <https://doi.org/10.2514/6.2007-3837>, AIAA Paper 2007–3837
- Namgoong H, Crossley W, Lyrintzis AS (2002) Global optimization issues for transonic airfoil design. In: 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Atlanta, Georgia, <https://doi.org/10.2514/6.2002-5641>, AIAA Paper 2002–5641
- Nielsen EJ, Lee-Rausch EM, Jones WT (2010) Adjoint based design of rotors in a noninertial frame. *J Aircr* 47(2):638–646. <https://doi.org/10.2514/1.46044>
- Panier ER, Tits AL (1991) Avoiding the Maratos effect by means of a nonmonotone line search I. General constrained problems. *SIAM J Numer Anal* 28(4):1183–1195. <https://doi.org/10.1137/0728063>
- Poole D, Allen C, Rendall T (2017) High-fidelity aerodynamic shape optimization using efficient orthogonal modal design variables with a constrained global optimizer. *Comput Fluids* 143:1–15. <https://doi.org/10.1016/j.compfluid.2016.11.002>
- Poole DJ, Allen CB, Rendall TCS (2015) Metric-based mathematical derivation of efficient airfoil design variables. *AIAA J* 53(5):1349–1361. <https://doi.org/10.2514/1.J053427>
- Qin N, Vavalle A, Le Moigne A, Laban M, Hackett K, Weinerfelt P (2004) Aerodynamic considerations of blended wing body aircraft. *Prog Aerosp Sci* 40(6):321–343. <https://doi.org/10.1016/j.paerosci.2004.08.001>
- Rendall TCS, Allen CB (2008) Unified fluid-structure interpolation and mesh motion using radial basis functions. *Int J Numer Meth Eng* 74(10):1519–1559. <https://doi.org/10.1002/nme.221>
- Robinson GM, Keane AJ (2001) Concise orthogonal representation of supercritical airfoils. *J Aircr* 38(3):580–583. <https://doi.org/10.2514/2.2803>
- Samareh JA (2001a) Novel multidisciplinary shape parameterization approach. *J Aircr* 38(6):1015–1024. <https://doi.org/10.2514/2.2888>
- Samareh JA (2001b) Survey of shape parameterization techniques for high fidelity multidisciplinary shape optimization. *AIAA J* 39(5):877–884. <https://doi.org/10.2514/2.1391>
- Sobieczky H (1998) Parametric airfoils and wings. *Notes Numer Fluid Mech* 68:71–88
- Toal DJJ, Bressloff NW, Keane AJ, Holden CME (2010) Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization. *AIAA J* 48(5):916–928. <https://doi.org/10.2514/1.41420>
- van Leer B (1982) Flux vector splitting for the Euler equation. *Lecture Notes in Physics* https://doi.org/10.1007/3-540-11948-5_66
- Wendland H (2005) Scattered data approximation, 1st edn. Cambridge University Press, Cambridge

- Yamazaki W, Mouton S, Carrier G (2010) Geometry parameterization and computational mesh deformation by physics-based direct manipulation approaches. *AIAA J* 48(8):1817–1832. <https://doi.org/10.2514/1.J050255>
- Zhou JL, Tits AL (1993) Nonmonotone line search for minimax problems. *J Optim Theory Appl* 76(3):455–476. <https://doi.org/10.1023/A:1020896407415>
- Zhou JL, Tits AL, Lawrence CT (1997) Users guide for FSQP version 3.7 : a Fortran code for solving optimization programs, possibly minimax, with general inequality constraints and linear equality constraints, generating feasible iterates. Technical report, Institute for Systems Research, University of Maryland, sRC-TR-92-107r5
- Zhu F, Qin N (2014) Intuitive class/shape function parameterization for airfoils. *AIAA J* 52(1):17–25. <https://doi.org/10.2514/1.J052610>