Domain-Element Method for Aerodynamic Shape Optimization Applied to a Modern Transport Wing

A. M. Morris,[*] C. B. Allen,[†] and T. C. S. Rendall[‡]
University of Bristol, Bristol, England BS8 1TR, United Kingdom

DOI: 10.2514/1.39382

Generic wraparound aerodynamic shape optimization technology is presented and applied to a modern commercial aircraft wing in transonic cruise. The wing geometry is parameterized by a novel domain-element method, which uses efficient global interpolation functions to deform both the surface geometry and corresponding computational fluid dynamics volume mesh. The technique also provides a method that allows geometries to be parameterized at various levels, ranging from global three-dimensional planform alterations to detailed local surface changes. Combining all levels of parameterization allows for free-form design control with very few design variables. The method provides an efficient combined shape parameterization and high-quality mesh deformation technique that is totally independent of mesh type (structured or unstructured). Optimization independence from the flow solver is achieved by obtaining sensitivity information for an advanced gradient-based optimizer by finite differences. The entire optimization suite has also been parallelized to allow optimization with highly flexible parameterization in practical times. Results are presented for highly constrained optimizations of the modern aircraft wing in transonic cruise, using three levels of parameterization (number of design variables) to assess the effect of parameterization level on the optimization. The highest-level optimization results in a totally-shock-free geometry with an associated substantial reduction in drag.

I. Introduction

IMPROVED aerodynamic or fluid dynamic efficiency is sought constantly in many areas and in all aerospace industries, particularly with increasing fuel costs. Computational fluid dynamics (CFD) is at the forefront of analysis capabilities and provides a fast and effective method of predicting aerodynamic performance, and these methods are now commonplace in aerospace industries and are being used increasingly in design. However, with increasing complexity of geometries, designers can often struggle to interpret the simulation results sufficiently to be able to manually alter the geometry to improve performance. Hence, there has been an increase in demand for intelligent and automatic shape optimization schemes. Combining geometry control methods with a numerical optimization algorithm provides a mechanism to mathematically seek improved and optimum designs using CFD as the analysis tool.

Central to any aerodynamic shape optimization method is the geometry parameterization (i.e., a method of representing the design surface that defines the degrees of freedom in which the geometry can be altered). This must then be linked with an effective method of deforming the CFD surface and volume mesh in a corresponding fashion. However, parameterizing complex shapes is a problem that remains a serious obstacle to both manual and automatic CFD-based optimization. A wide variety of shape control and morphing methods have been developed [1–11], but these often do not allow sufficiently free-form design, can produce infeasible shapes, and do not allow the possibility of manual manipulation of the geometry [12,13]. Furthermore, most methods do not have a suitable method to deform the CFD mesh once the surface has been changed, and regeneration is often required. This may not be a problem for simple geometries and/or small meshes, but can make automation of the optimization process impossible in some cases. Those methods that do incorporate CFD-mesh deformation techniques are often of poor quality, hence restricting the size of the allowable deformation, and/or are computationally expensive and impractical for large CFD meshes. It is essential that designs can be deformed into arbitrary shapes, but an excessive number of deformation degrees of freedom (design variables) can often make optimization impractically expensive.

An efficient domain-element shape parameterization method has been developed by the authors, along with a high-quality and robust mesh deformation scheme, and presented recently for two-dimensional CFD-based shape optimization [14,15]. The parameterization technique, surface-mesh motion, and volume-mesh motion are all accomplished simultaneously through global interpolations using radial basis functions, such that when the positions of the domain element are altered, both the design surface and its corresponding CFD volume mesh are deformed in a high-quality fashion, and this allows automating the entire process. This interpolation has been developed such that the domain-element parameterization method has no computational memory overhead that may restrict the size of the CFD volume mesh that can be used. The domain-element parameterization technique also allows for geometry control at various scales, ranging from gross three-dimensional planform alterations to fine, detailed surface-geometry changes. Furthermore, it is totally independent of the CFD-mesh type, removing any grid generation or flow solver dependence.

To ensure that a totally wraparound tool could be developed (i.e., also independent of the flow solver), the sensitivities required for optimization are computed here via finite difference. This allows numerous options in terms of optimization approaches, and an advanced feasible sequential programming (FSQP) [16–18] gradient-based optimizer has been integrated into the framework. Significantly, this approach allows the application of numerous strict constraints to an optimization. This domain-element parameterization, global interpolation-based CFD-mesh motion, and advanced optimization approach have been proven in two dimensions, demonstrating drag reductions of up to 45% for highly constrained airfoil cases [14,15].

The research presented in this paper is the three-dimensional extension, including parallelization, of the wraparound shape parameterization and optimization method. The software suite has been parallelized to allow optimization of three-dimensional bodies in practical times. Optimization is applied here to the multidisciplinary optimization (MDO) wing (a large modern transport

Received 26 June 2008; revision received 6 February 2009; accepted for publication 8 February 2009. Copyright © 2009 by C. B. Allen. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/09 $10.00 in correspondence with the CCC.

[*] Postdoctoral Researcher, Department of Aerospace Engineering.

[†] Professor of Computational Aerodynamics, Department of Aerospace Engineering. Member AIAA.

[‡] Postdoctoral Researcher, Department of Aerospace Engineering. Student Member AIAA.

aircraft wing, the result of a previous Brite-Euram project [19,20]) in the economical transonic cruise condition. This geometry is not optimized for aerodynamic performance, and so it offers significant scope for aerodynamic improvements; of particular interest here is the effect of varying the optimization scheme in regard to the number (level) of design parameters included. Optimizations at three parameterization levels are performed, ranging from pure planform alterations to detailed surface-geometry changes, and the flexibility and effectiveness of the approach is analyzed in detail. In all cases, the objective is drag minimization.

It should be noted that although in-house CFD grid generation and flow solution codes are used here, the method is completely generic and can be wrapped around any appropriate tools. Furthermore, although an external aerodynamic design problem is presented, this is not a restriction, and the methods can be applied to any steady-state fluid dynamic design problem.

II. Domain-Element Parameterization

The shape parameterization method developed is critical to both the success and the computational cost of an optimization. The method must allow sufficient free-form design such that any likely optimum design that may exist is achievable, and inverse design methods are usually applied to gauge the range of design space achievable by any method. To keep computational costs to a minimum, it is essential to both use an efficient optimization strategy and to have as few design variables as possible, especially if sensitivity information is to be evaluated by finite difference.

Parameterization methods used for CFD shape optimization either parameterize the design geometry from which a mesh is generated or they parameterize the aerodynamic mesh itself. Geometry parameterization methods uniquely define a surface geometry by the values of the design variables. These methods are inherently linked with the grid generation package, and optimization of a design requires automatic grid generation tools. Examples of this method include partial differential equation methods [1,8], polynomial or spline [9] and shape-function/class-function transformation [10,11] methods. Methods that parameterize the grid are generally independent of the grid generation package, but a mesh deformation algorithm is required. Grid generation is still a major and time-consuming challenge in industry, and these types of methods allow the use of previously generated grids for optimization, which is a significant advantage. Parameterization methods of this nature include discrete [21–23], analytical, basis vector [4], free-form deformation algorithms [5], and domain-element methods.[§] A comprehensive review of available parameterization methods is presented by Samareh [24,25].

The novel parameterization method developed and applied in this research links the entire aerodynamic volume mesh to a domain element that controls the shape of the design. At the center of this parameterization technique is a multivariate interpolation using radial basis functions, and this provides a unique mapping between the domain element, the surface geometry, and the locations of grid points in the volume mesh. All points are treated as point clouds, and so the parameterization technique is totally independent from the grid type and generation package. The mapping is only required once for the initial design, as the values of the parametric coordinates of the grid points with respect to the domain element remain constant throughout the optimization. Updates to the geometry and the corresponding mesh are provided simultaneously by application of the multivariate interpolation, and this results in very-high-quality mesh deformation [26,27].

Figure 1 depicts the domain-element parameterization of the MDO wing. The method here is an extension of that presented by Morris et al. [15] for a two-dimensional airfoil, such that the three-dimensional element consists of an evenly distributed series of two-dimensional slices located according to local surface geometry.

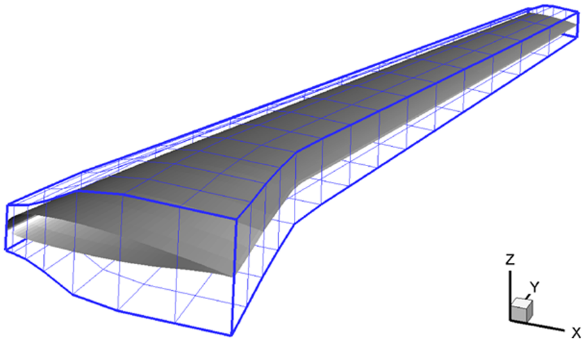


Fig. 1 MDO wing surface and parameterization.

Design variables are not chosen to be the individual location of each domain-element point. Instead, a hierarchy of intuitive shape deformation design variables has been developed to reduce the number of variables. Three levels of design variables have been established. At the highest level, design variables correspond to motions of all domain-element nodes simultaneously. For example, for an airfoil section, the parameters may be chord, camber, and thickness. For a wing, a single design variable may be defined to alter the positions of all the two-dimensional domain-element slices relative to each other, resulting in changing sweep or dihedral distributions. At the intermediate level, design variables may control the twist, chord, and thickness of each two-dimensional domain-element slice separately. At the lowest level, very small groups or individual domain-element nodes are altered to provide detailed and local shape changes.

A simple example is presented to demonstrate the approach. A cylindrical surface is taken as the design surface, and six square slices are used as the domain element to control this surface (see Fig. 2a). Figure 2b shows a local parameter (i.e., movement of a single domain-element point). Figures 2c and 2d show two global parameters: a sweep (in fact, this is actually a shear) parameter and a linear twist parameter. Figure 2e shows these two global parameters combined, and Fig. 2f shows these two global parameters and the local parameter combined. This shows both the flexibility of the parameterization scheme and the effectiveness of the surface deformation scheme; the volume mesh would be deformed by the same motion, and a mesh deformation example is demonstrated later for the real mesh.

Note that, as shown in Fig. 2, the parameterization is fully three-dimensional, such that deformations due to a movement of the domain element in the plane of a two-dimensional slice smoothly extend in the spanwise direction as well, and so no linear interpolation is required between slices. An example deformation of the CFD volume mesh to a change in a design variable is depicted in Fig. 3 (in this case, the eighth domain-element slice is perturbed in the local twist design variable). It is observed that the interpolation method provides a smooth but local deformation to the wing in all three dimensions. Selected planes of the volume mesh are shown, and the interpolation method is seen to provide a smooth and very-high-quality mesh deformation, with grid motion contained within the radial basis function (RBF) support radius (discussed subsequently).

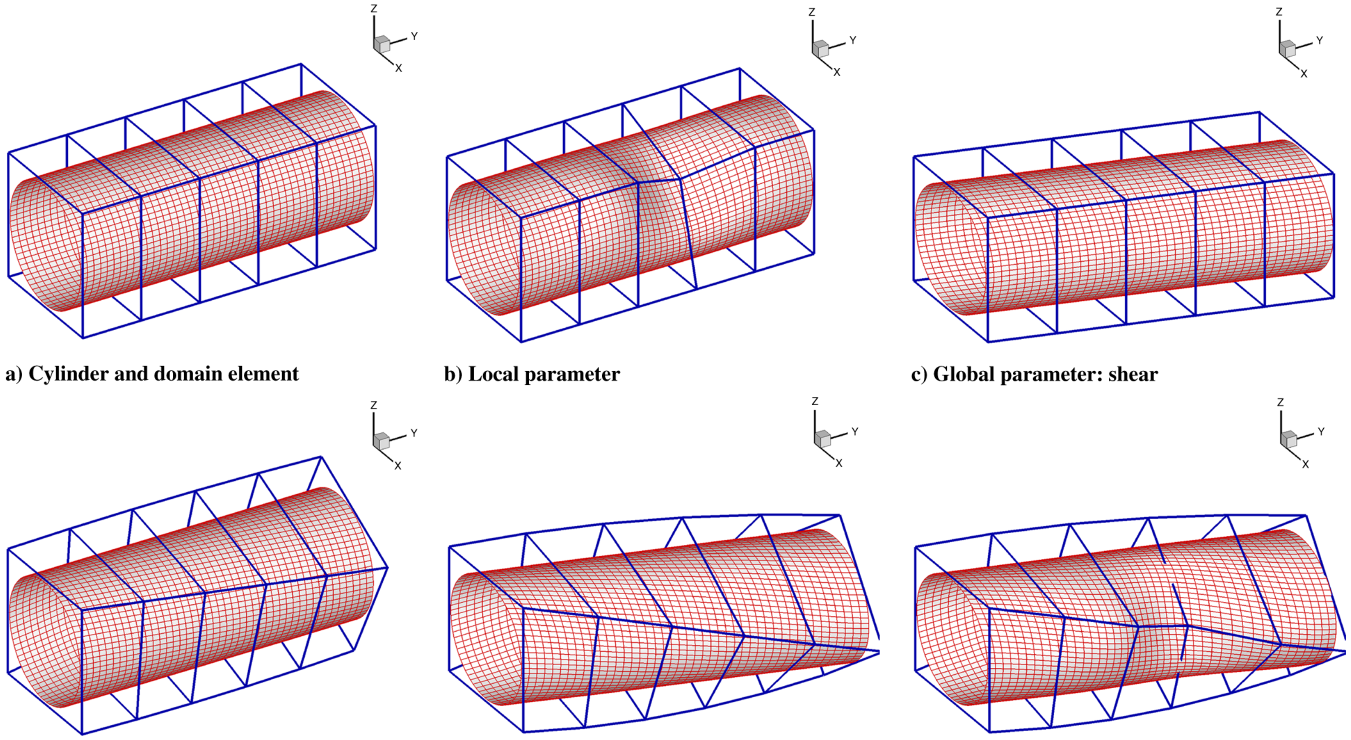
The results of three optimizations of the MDO are presented in this paper. Each optimization adopts identical domain elements, optimization algorithm, and constraints in the same cruise condition, however:

Optimization 1 corresponds to a low-cost optimization, with only 30 active design variables of the highest level. These control only truly-three-dimensional deformations of the wing planform.

Optimization 2 corresponds to an intermediate optimization with 104 active design variables. In addition to the three-dimensional design variables in optimization 1, each two-dimensional domain-element slice can deform.

Optimization 3 corresponds to the highest flexibility optimization. Each two-dimensional domain-element slice has a full set of 22 active design variables developed for free-form airfoil design

[§] Data available online at http://www.optimalsolutions.us/ [retrieved 10 June 2008].



a) Cylinder and domain element

b) Local parameter

c) Global parameter: shear

d) Global parameter: twist

e) Combined global parameters: twist and shear

f) Combined global and local parameters

Fig. 2 Cylinder parameterization example.

[14,15]. These are also combined with the some of design variables from optimizations 1 and 2 to create a total of 388 active design variables. (More details of the parameters will be given in a later section.)

The most flexible optimization here still has considerably fewer design variables than would be required by many other shape parameterization techniques to provide free-form surface design for such a geometry. A further advantage of the domain-element method is the truly-three-dimensional design-deformation degrees of freedom that cannot be easily replicated by many other parameterization techniques. The parameterization technique developed has several key advantages:

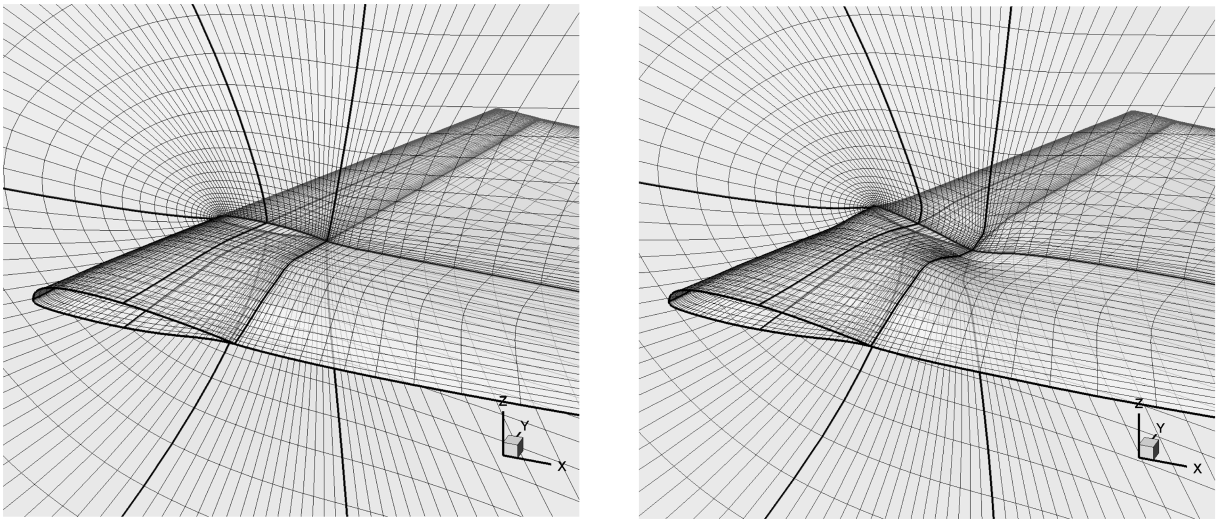
1) The number of design variables required to allow free-form design can be very low, when compared with other methods [15].

2) Design variables can range in scale from the coordinate of a single domain-element location, providing a detailed surface-geometry change, to a gross three-dimensional deformation such as sweep distribution. This can allow designers to choose the fidelity of the parameterization.

3) Global parameters (for example, angle of attack or wing sweep) can be included as design variables. This is impossible in many other optimization approaches, in which these quantities have to be adjusted externally to attempt to satisfy constraints.

4) The parameterization technique is independent of the initial geometry. This can allow very complex or multi-element geometries to be parameterized.

5) The parameterization method has a low number of control points (in this case, the number of domain-element points), and this



a) MDO with initial mesh planes

b) Perturbed MDO with perturbed mesh planes

Fig. 3 MDO mesh with RBF deformation.

allows incorporating a high-quality interpolation to deform the surface and mesh directly. This would be impractical if all the surface-mesh points were control points, for example.

6) Updates to the surface geometry and the required volume mesh are independent of the grid generation package. Grid generation is still a major and time-consuming challenge in industry, and the domain-element method allows the use of previously generated grids for optimization. Both structured and unstructured grids are equally applicable.

7) The interpolation dependence only needs to be computed once, before any simulation. Updates to the geometry and the corresponding volume mesh are then provided simultaneously by matrix-vector multiplication; this is extremely fast and efficient and results in high-quality mesh deformation.

8) There is no requirement for surface smoothness. For example, geometries that possess geometric excrescences or gaps can still be optimized with this technique. Because a smooth global interpolation method is used, if the initial surface is discontinuous, deformed geometries will still be discontinuous. If necessary, constraints can be placed around those excrescences (for example, extra control points can be placed there with zero deformation set), and so features remain within defined geometric parameters. This is an advantage and a novel aspect of this parameterization method.

III. Parameterization and Mesh Motion Formulation

A multivariate interpolation method using RBFs has been developed to provide a method of two- and three-dimensional geometry parameterization. The global dependence between the domain-element nodes and the aerodynamic mesh points is evaluated and translates a deformation of the element due to a design-variable change to smoothly alter the aerodynamic shape and its corresponding CFD volume mesh. Using the domain-element method, only an initial mesh of the original design is required to allow optimization. The interpolation method developed here also requires no connectivity information and can therefore be applied equally well to either structured- or unstructured-grid topologies. Domain-element points and volume-mesh points are simply treated as independent point clouds, with the dependence matrix computed only once. If desired, the surface deformation and volume deformations may be split. In this case, polynomial terms would be included in the surface deformation, but not in the volume deformation, which would best be done in terms of displacements, not positions [27].

The general theory of RBFs is presented by Buhmann [28] and Wendland [29], and the method used here is detailed by Allen and Rendall [26] and Rendall and Allen [27].

The solution of an interpolation problem using RBFs begins with the form of the required interpolation:

$$s(\mathbf{r}) = \sum_{i=1}^{i=N} \beta_i \phi(\|\mathbf{r} - \mathbf{r}_i\|) + p(\mathbf{r}) \tag{1}$$

where $s(\mathbf{r})$ is the function approximated, the index i identifies the centers for the RBFs (the domain-element nodes in this case), and $\mathbf{r}_i$ is the location of that center. For $p(\mathbf{r})$, a linear polynomial is used so that translation and rotation are recovered exactly. The coefficients β_i are found by requiring exact recovery of the original function.

Initially, a system needs to solved relating to the domain-element nodes, to evaluate the coefficients associated with them. Exact recovery of the centers gives (including linear polynomial terms)

$$\mathbf{X}_{\mathrm{DE}} = \mathbf{C}\mathbf{a}_x \tag{2}$$

$$\mathbf{Y}_{\mathrm{DE}} = \mathbf{C}\mathbf{a}_y \tag{3}$$

$$\mathbf{Z}_{\mathrm{DE}} = \mathbf{C}\mathbf{a}_z \tag{4}$$

where

$$\mathbf{X}_{\mathrm{DE}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{x}_{\mathrm{DE}} \end{pmatrix} \qquad \mathbf{x}_{\mathrm{DE}} = \begin{pmatrix} x_{\mathrm{DE}_1} \\ \vdots \\ x_{\mathrm{DE}_N} \end{pmatrix} \qquad \mathbf{a}_x = \begin{pmatrix} \gamma_0^x \\ \gamma_x^x \\ \gamma_y^x \\ \gamma_z^x \\ \beta_{\mathrm{DE}_1}^x \\ \vdots \\ \beta_{\mathrm{DE}_N}^x \end{pmatrix} \tag{5}$$

(analogous definitions hold for $\mathbf{Y}_{\mathrm{DE}}$ and $\mathbf{Z}_{\mathrm{DE}}$ and their $\mathbf{a}$ vectors) and

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & x_{\mathrm{DE}_1} & x_{\mathrm{DE}_2} & \cdots & x_{\mathrm{DE}_N} \\ 0 & 0 & 0 & 0 & y_{\mathrm{DE}_1} & y_{\mathrm{DE}_2} & \cdots & y_{\mathrm{DE}_N} \\ 0 & 0 & 0 & 0 & z_{\mathrm{DE}_1} & z_{\mathrm{DE}_2} & \cdots & z_{\mathrm{DE}_N} \\ 1 & x_{\mathrm{DE}_1} & y_{\mathrm{DE}_1} & z_{\mathrm{DE}_1} & \phi_{\mathrm{DE}_1\mathrm{DE}_1} & \phi_{\mathrm{DE}_1\mathrm{DE}_2} & \cdots & \phi_{\mathrm{DE}_1\mathrm{DE}_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{\mathrm{DE}_N} & y_{\mathrm{DE}_N} & z_{\mathrm{DE}_N} & \phi_{\mathrm{DE}_N\mathrm{DE}_1} & \phi_{\mathrm{DE}_N\mathrm{DE}_2} & \cdots & \phi_{\mathrm{DE}_N\mathrm{DE}_N} \end{pmatrix} \tag{6}$$

with

$$\phi_{\mathrm{DE}_1\mathrm{DE}_2} = \phi(\|\mathbf{r}_{\mathrm{DE}_1} - \mathbf{r}_{\mathrm{DE}_2}\|) \tag{7}$$

indicating the basis function evaluated on the distance between DE_1 and DE_2 (basis function defined later). Subscript DE represents a domain-element control point, and N is the number of control points. To locate the aerodynamic-mesh-point positions resulting from the domain-element positions, the following matrix must be formed, in which subscript a indicates an aerodynamic mesh point, and M is the number of mesh points:

$$\mathbf{A} = \begin{pmatrix} 1 & x_{a_1} & y_{a_1} & z_{a_1} & \phi_{a_1\mathrm{DE}_1} & \phi_{a_1\mathrm{DE}_2} & \cdots & \phi_{a_1\mathrm{DE}_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{a_M} & y_{a_M} & z_{a_M} & \phi_{a_M\mathrm{DE}_1} & \phi_{a_M\mathrm{DE}_2} & \cdots & \phi_{a_M\mathrm{DE}_N} \end{pmatrix} \tag{8}$$

The positions of the aerodynamic mesh points, given by the vectors $\mathbf{X}_a$, $\mathbf{Y}_a$, and $\mathbf{Z}_a$, are then computed by

$$\mathbf{X}_a = \mathbf{A}\mathbf{a}_x = \mathbf{A}\mathbf{C}^{-1}\mathbf{X}_{\mathrm{DE}} = \mathbf{H}\mathbf{X}_{\mathrm{DE}} \tag{9}$$

$$\mathbf{Y}_a = \mathbf{A}\mathbf{a}_y = \mathbf{A}\mathbf{C}^{-1}\mathbf{Y}_{\mathrm{DE}} = \mathbf{H}\mathbf{Y}_{\mathrm{DE}} \tag{10}$$

$$\mathbf{Z}_a = \mathbf{A}\mathbf{a}_z = \mathbf{A}\mathbf{C}^{-1}\mathbf{Z}_{\mathrm{DE}} = \mathbf{H}\mathbf{Z}_{\mathrm{DE}} \tag{11}$$

There are several methods to implement this approach: for example, computing $\mathbf{H}$ directly by inverting $\mathbf{C}$, inverting $\mathbf{C}$, storing $\mathbf{C}^{-1}$ and multiplying through by $\mathbf{A}$ when required, or evaluating $\mathbf{a}$ and multiplying through by $\mathbf{A}$ when required, but these are not considered here. Features of this interpolation worth noting are as follows:

1) The interpolation is independent of mesh type or structure, as the position vectors are simply grid positions in any order.

2) The interpolation is unique. This implies that no additional constraints need be applied.

3) The interpolation in all three coordinate directions is independent.

4) The interpolation is time-invariant and only needs to be computed once, before any simulation. The surface geometry and mesh are then deformed very efficiently by matrix-vector multiplication.

5) The interpolation is perfectly parallel, as the matrix and the position vectors can simply be split into required rows and elements.

Domain-element methods work well with this approach to mesh deformation, due to the low number of domain-element node points. However, if one were to use spline control points or surface-mesh points in three dimensions, then computational memory issues must be considered [30].

Because of the decaying behavior of the compact basis functions used here [27], the interpolation has a local character, and domain-element nodes located nearer the surface points have a greater influence. This would also be true for the other functions, including thin-plate spline functions; however, their global nature produces changes at the far field in the volume mesh as well. Therefore, domain-element nodes are located nearer and more clustered to the wing surface near the leading edge; this allows for finer control of the leading edge. This is in comparison with nodes over the main body of the wing, where smooth gross changes are generally more applicable. The exact basis function used and the support radius can be varied to provide smoother or more global changes to both the surface geometry and the volume mesh [27].

Of course, the function $\phi(\|\mathbf{r}\|)$ used is important. Any decaying function can be used, but the compact functions of Wendland [29] are used. The C2 function of Wendland was found to provide the best combination of deformation quality and matrix conditioning, and this is defined as

$$\begin{aligned} \phi(\|\mathbf{r}\|) &= (1 - \|\mathbf{r}\|)^4(4\|\mathbf{r}\| + 1) \qquad \|\mathbf{r}\| < 1.0 \\ \phi(\|\mathbf{r}\|) &= 0 \qquad \|\mathbf{r}\| \geq 1.0 \end{aligned} \tag{12}$$

where $\|\mathbf{r}\|$ is the Euclidean norm. In the preceding, $\|\mathbf{r}\|$ is actually $\|\mathbf{r}\|/\mathrm{SR}$, where SR is the support radius (i.e., the region of influence). Outside of this region, there will be no grid deformation.

IV. Optimization Method

Constrained gradient-based optimizers used to minimize an objective function are fast and efficient at providing solutions to local optimization problems [18,31]. When considering practical and relevant optimization of aerodynamic performance of a solid body (airfoil/wing/rotor blade), there are usually constraints that need to be imposed (minimum thickness, minimum volume, minimum lift, maximum moment, etc.). Unconstrained optimizations can incorporate constraints by using a penalty function for designs that are near or beyond the constraint boundary, but these methods are now considered to be inefficient and have been replaced by methods that focus on the solution of the Kuhn–Tucker equations.

The aim of an optimization process is to obtain the values of a set of design variables, $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$, that in some way can be defined as optimal. An objective function determined by the numeric values of the design variables $\underline{J}(\underline{\alpha})$ is to be minimized and can be subject to equality constraints, inequality constraints, and/or parameter bounds.

A general constrained-problem description is stated as

$$\text{Minimize } \underline{J}(\underline{\alpha}) \tag{13}$$

Subject to equality constraints $G_i(\underline{\alpha}) = 0$ (where $i = 1, \ldots, m_e$), inequality constraints $G_i(\underline{\alpha}) \leq 0$ (where $i = m_e + 1, \ldots, m$), and parameter bounds $\underline{\alpha}_l \leq \underline{\alpha} \leq \underline{\alpha}_u$, where the vector function $\underline{G}(\underline{\alpha})$ is a vector of length m containing the values of the equality and inequality constraints evaluated at $\underline{\alpha}$.

The Kuhn–Tucker equations are necessary conditions for optimality for a constrained optimization problem and can be stated as [in addition to the original constraints of Eq. (13)]:

$$\nabla \underline{J}(\underline{\alpha}^*) + \sum_{i=1}^{m} \lambda_i^* . \nabla \underline{G}_i(\underline{\alpha}^*) = 0 \tag{14}$$

$$\lambda_i^* \cdot \underline{G}_i(\underline{\alpha}^*) = 0 \qquad i = 1, \ldots, m \tag{15}$$

$$\lambda_i^* \geq 0 \qquad i = m_e + 1, \ldots, m \tag{16}$$

where $\underline{\alpha}^*$ refers to the optimum values of $\underline{\alpha}$.

The solution of these equations forms the basis of the nonlinear programming algorithm. The constrained quasi-Newton method guarantees superlinear convergence by accumulating second-order information relating to the Kuhn–Tucker equations using a quasi-Newton updating procedure; that is, at each major iteration, an approximation is made of the Hessian of the Lagrangian function. This is then used to generate a quadratic programming subproblem in which the solution is used to form a search direction for a line-search procedure. This forms the basis of the sequential quadratic programming (SQP) algorithm.

SQP methods represent the forefront of nonlinear programming techniques. Schittkowski [32] implemented and tested an SQP algorithm that, on average, outperformed other tested methods in terms of efficiency, accuracy, and percentage of successful solutions on a set of test problems. The FSQP algorithm used in the current research was originally developed in [16–18]. The feasibility aspect of the optimizer relates to a generated design satisfying all constraints; that is, if an initial design does not satisfy the specified constraints, the optimizer first achieves a satisfactory design, and then all subsequent iterates generated also satisfy all constraints simultaneously. This particular algorithm has been implemented across a wide range of optimization problems: most relevant and notable is the work of [33–36], in which the algorithm is used for CFD constrained optimization of a blended wing/body using an inviscid adjoint solver to obtain the sensitivities.

The development of generic optimization tools encompassing a wide range of applicability has been the principle aim of the current research. This has ultimately required the use of a finite difference technique for evaluating gradients to enable independence from the flow solver; the sensitivity for each design variable is easily obtained by the relative change in the value of steady-state objective function, due to a geometric perturbation. The cost of evaluating all the sensitivities with this method is approximately linear with the number of design variables. Hence, computation can prove to be expensive for a large number of design variables; however, significant speedup can be achieved by parallelization (in a data sense, rather than domain decomposition sense). To ensure no biasing toward one direction and to increase accuracy, a second-order central-difference finite difference stencil is used.

It is worth considering this approach further. It could be argued that gradient-based methods may not be ideal for the case in which the initial design is already well designed, particularly if this lies at or near a local minimum. In this case, something more global may be suitable. Reference [37] presented an interesting study related to this, comparing both optimization methods and flow solvers. Specifically, three optimization tools [SYN107 (gradient-based search technique using the continuous adjoint equation), MDOPT (a response-surface method), and OPTIMAS (a floating-point genetic algorithm)] were applied to the Navier–Stokes drag minimization of the DPW-W1 wing across a range of flight conditions. All three optimizers produced similar designs and when cross-checked by a different CFD flow solver yielded similar drag reductions and also reasonable offdesign performance.

For a case in which a more global approach is required, the optimization scheme used here could be used in a different mode. In the results here, it is applied such that every new evolution has to be an improvement on the previous. However, it can be run without this restriction, which allows a more global search space to be explored. Furthermore, it should be remembered that application of strict constraints can be extremely difficult with the alternative approaches mentioned previously.

V. Parallel Aspects

The entire optimization suite has been parallelized. The optimization algorithm comprises modules for objective and constraint function evaluations, gradient of objective and constraint function modules, and an SQP numerical optimizer for updates. Function evaluations and optimizer updates are performed on the master node; however, for gradient evaluation using a finite difference methodology, multiple independent steady-state flow calculations are required to evaluate the sensitivities. This gradient evaluation module has been parallelized to allow parallel evaluation of the required sensitivities, such that each CPU can control the

Table 1 Optimization results overview

	Level	Parameters	Evolutions	Drag reduction, %	CFD solutions	Time (CPUs)
Optimization 1	Low	30	15	8.13	930	1.0 (31)
Optimization 2	Medium	103	26	14.23	5410	4.3 (28)
Optimization 3	High	388	30	18.29	23,340	7.0 (66)

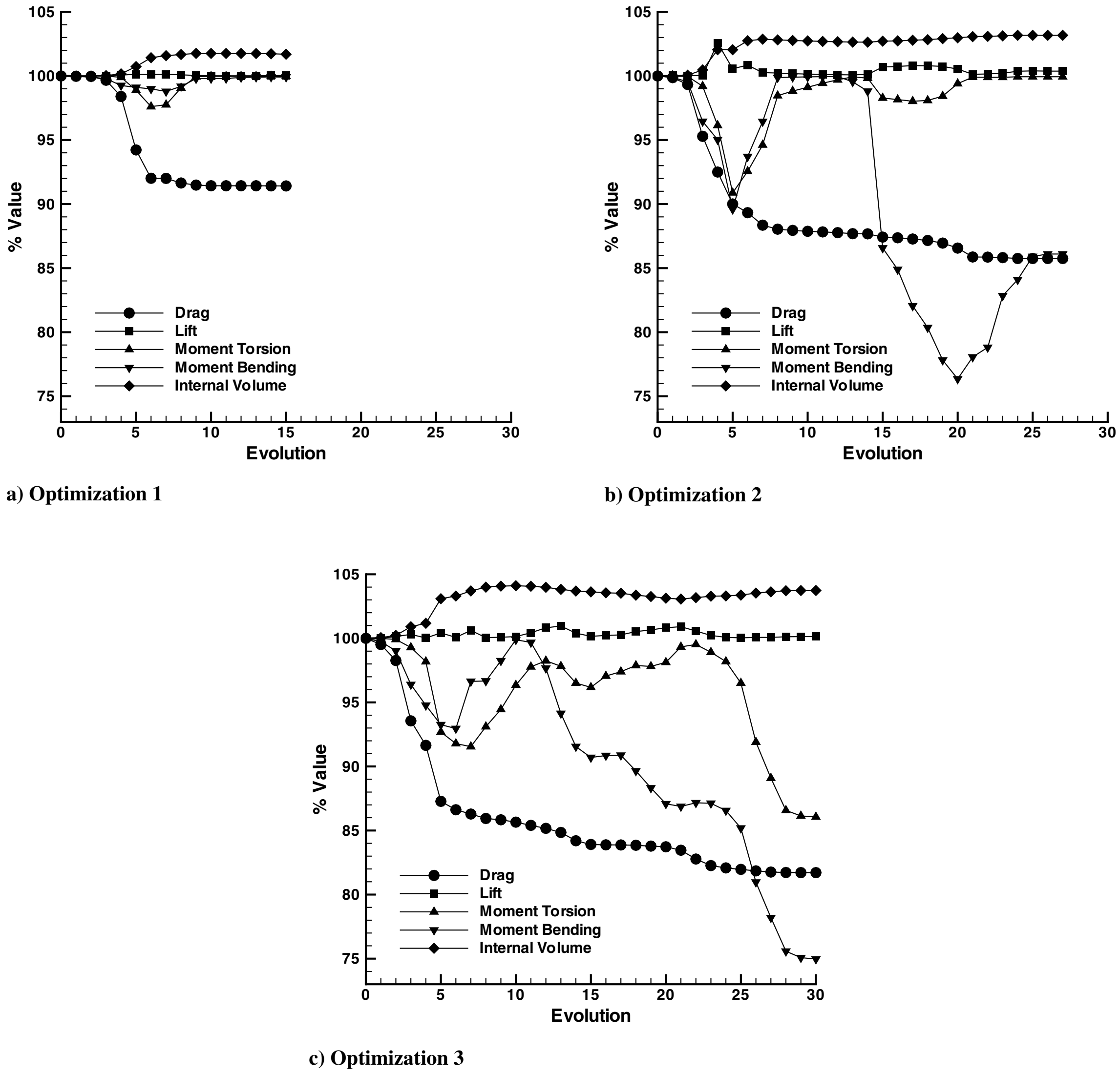
Table 2 Wing optimization results

	Initial	Optimization 1	% difference	Optimization 2	% diff	Optimization 3	% difference
Cl	0.4523	0.4525	+0.04	0.4541	+0.40	0.4530	+0.14
Cm_{bending}	0.1340	0.1338	−0.14	0.1153	−13.95	0.1004	−25.03
Cm_{torsion}	−0.0547	−0.0547	0.00	−0.0547	−0.18	−0.0471	−13.93
Volume	387.14	393.75	+1.71	399.430	+3.10	401.60	+3.73
Cd	0.02780	0.02554	−8.13	0.02401	−14.23	0.02287	−18.29

geometry (and CFD volume mesh) corresponding to a different design variable and call to the flow solver. Flowfield results are then returned to the master node for optimizer updates. This gradient evaluation module remains independent of the flow solver, and so either a serial or parallel version of the flow solver itself may be called. This approach scales almost linearly with processors, but with a small scalar overhead due to the line-search method, which must be sequential, and is carried out on the master node. The parallelization of the algorithm is key to enable optimization of three-dimensional bodies using a finite difference technique for gradient evaluation.

IV. Three-Dimensional Aerodynamic Optimization

The MDO wing corresponds to a typical traditional design of a large modern transport aircraft wing, with its primary design point being that of transonic cruise-flight efficiency. However, it should be



a) Optimization 1

b) Optimization 2

c) Optimization 3

Fig. 4 Optimization history.

stated here that this case offers significant scope for aerodynamic improvements, as the baseline case is by no means optimized with regard to aerodynamic performance. With this design point in mind, the objective of optimization is minimum drag; however, this must be achieved without detriment to other aerodynamic, structural, and geometric quantities. As such, four constraints are imposed on each optimization:

1) The total lift is greater than or equal to the total lift of the initial wing. This is an essential constraint. Crucially, the constraint involves total lift and not lift coefficient; this ensures that the optimization is independent of the aircraft reference area, which may alter with changing geometry.

2) The internal volume is greater than or equal to the internal volume of the initial wing. It has been demonstrated in previous works [15] that reductions in drag can be achieved by allowing internal volume to decrease, and so these designs do not truly represent design improvements.

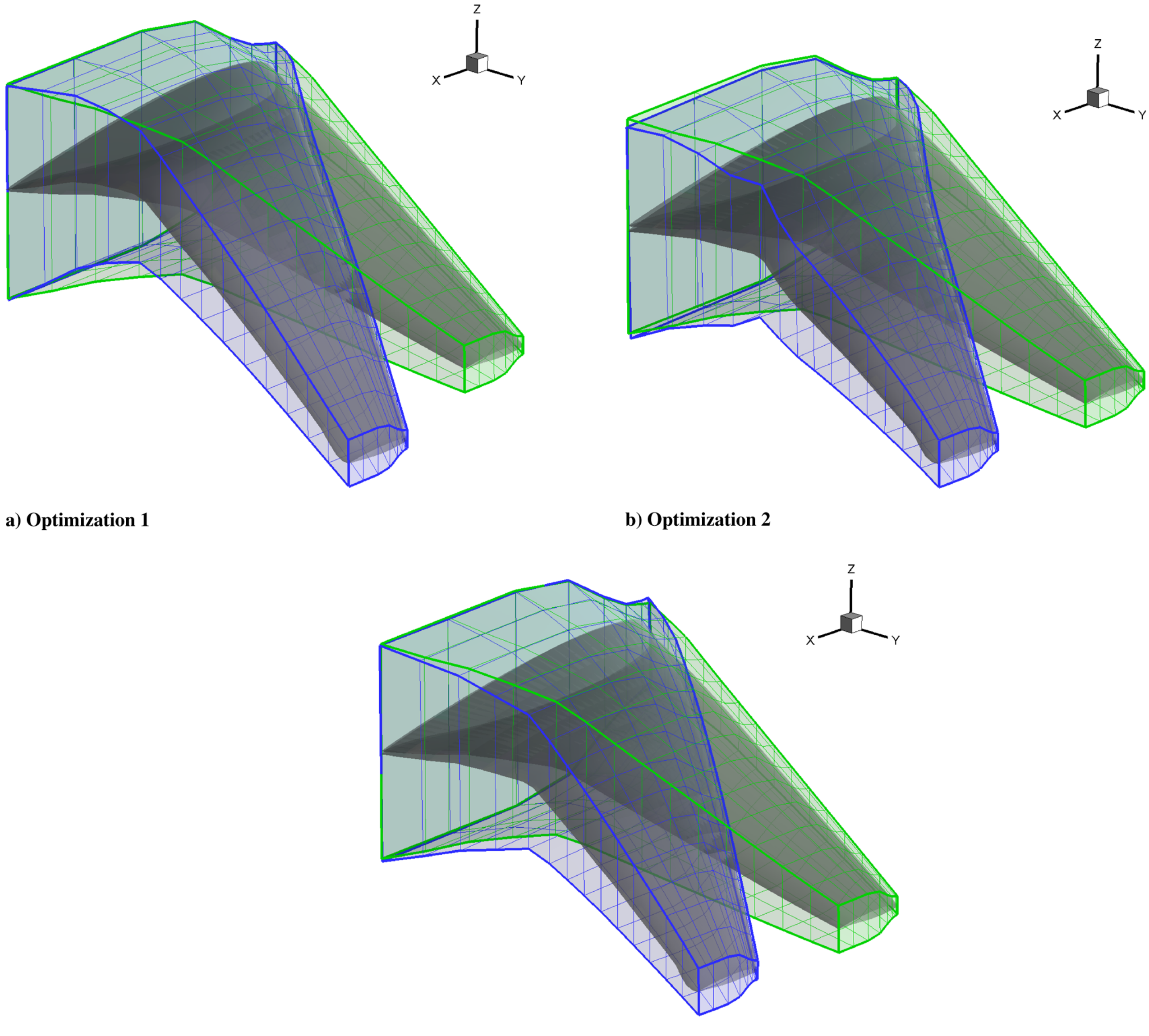
3) The root bending moment is less than or equal to the root bending moment of the initial wing. Major structural members of the wing are sized according to these loads and, as such, any increased moment will impact negatively on wing structural weight.

4) The root torsion moment is less than or equal to the root torsion moment of the initial wing, imposed for the same reasons given in the preceding constraint.

The preceding constraints ensure that the results of any optimization represent practical solutions and that any improvements achieved can be attributed solely to improvements in geometric design. Root moments are taken about the quarter-chord position of the root section. However, of course, the final result will depend on the constraints applied. For example, it may be more realistic to constrain the structural span rather than the aerodynamic span, but the results presented are suitable for demonstration of the method capability.

The economical cruise-flight Mach number for the MDO wing defined by Allwright [19] and Haase et al. [20] 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 by solution of the Euler equations, because induced and wave drag form a major part of the total drag. Furthermore, two-dimensional airfoil optimizations have shown previously that the improvements achieved through inviscid optimizations in transonic Mach numbers are also realized in viscous analysis [14,15]. The grid used in each optimization is a 330,000-point structured-multiblock mesh using the generation techniques of Allen [38]. Flow solutions are provided by an inviscid structured-multiblock finite volume upwind solver [39–42] using van Leer's flux vector splitting [43,44] and implementing multigrid convergence acceleration [45].

The results of three independent optimizations are presented. Each optimization of the MDO wing represents a different level of



a) Optimization 1

b) Optimization 2

c) Optimization 3

Fig. 5 Domain-element and wing geometries, initial and optimized.

parameterization, with a correspondingly increased number of active design variables. Optimization cost is directly linked with the number of active design variables for two reasons: first, gradient information is required for each design variable, and second, an optimization with an increased number of design variables frequently requires more evolutions to obtain an optimum solution.

The optimizations were run with the following design parameters:

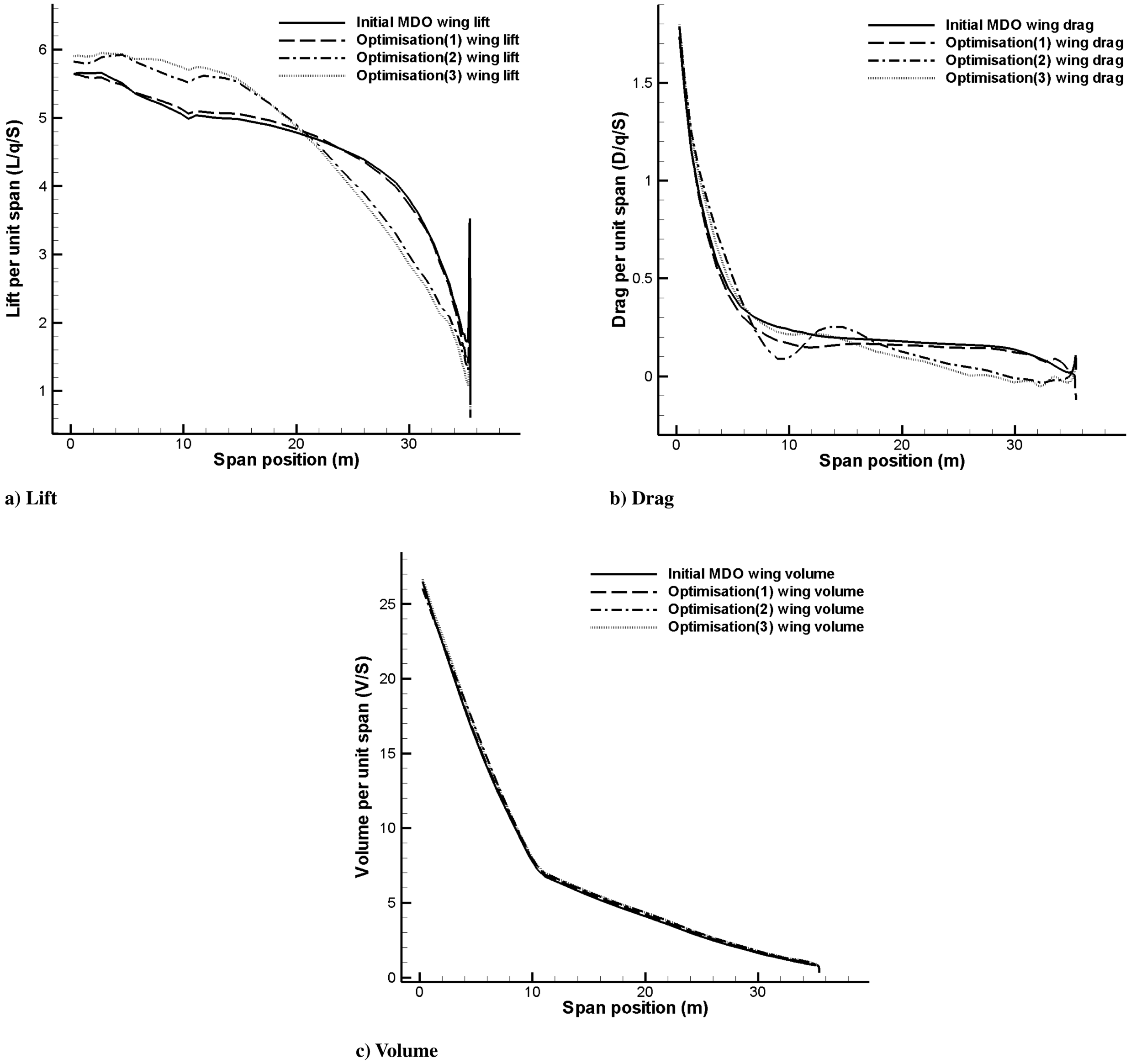
In optimization 1, only three issues are considered: twist, anhedral/dihedral, and sweep. There are 15 domain-element slices, and the root section is fixed in position; hence, there are 14 sweep design variables and 14 anhedral variables. A linear twist is added as a single parameter, and wing angle of attack is the final parameter, giving 30 parameters.

In optimization 2, each of the 15 domain-element slices now have five local design variables: angle of attack, chord, thickness, and (x, z) locations of the center of rotation. These are added to 28 of the variables from optimization 1; twist and angle of attack are is not required. Hence, there are 103 parameters.

In optimization 3, each of the 15 domain-element slices now have the full set of 22 active design variables developed for free-form airfoil design [14,15]. These are then combined with some of the design variables from optimizations 1 and 2. Only the (x, z) locations of the center of rotation are required for each slice, and angle of attack and twist are not required from the 30 parameters. This gives a total of 388 active design variables.

It should be stated here that the possible combinations of parameters defined are huge, and only representative parameters are used here. However, as has been shown in [15], a large proportion of the design space is available using these parameters in combination with the FSQP algorithm. The important issue here is the effect the parameter scales have on the optimization results. It should also be noted that the sweep parameter is not strictly a sweep, but a shear. However, it was decided to fix the wing semispan rather than the structural span.

A summary of the results is given in Table 1, with detailed results of the three optimizations given in Table 2. Optimization 1 achieves a drag reduction of over 8%, which is impressive, considering the extent to which the optimization is constrained and that only planform changes are allowed. As the level of the optimization increases, activating more design variables, which enables more detailed and subtle changes to the wing geometry, further improvements are achieved. Optimization 3 is of the highest level and corresponds to active design variables that enable fully free-form control over airfoil profile geometry combined with design variables that enable truly-three-dimensional planform alterations. This optimization achieves a reduction in drag coefficient of 18%, and this is a significant reduction.



a) Lift

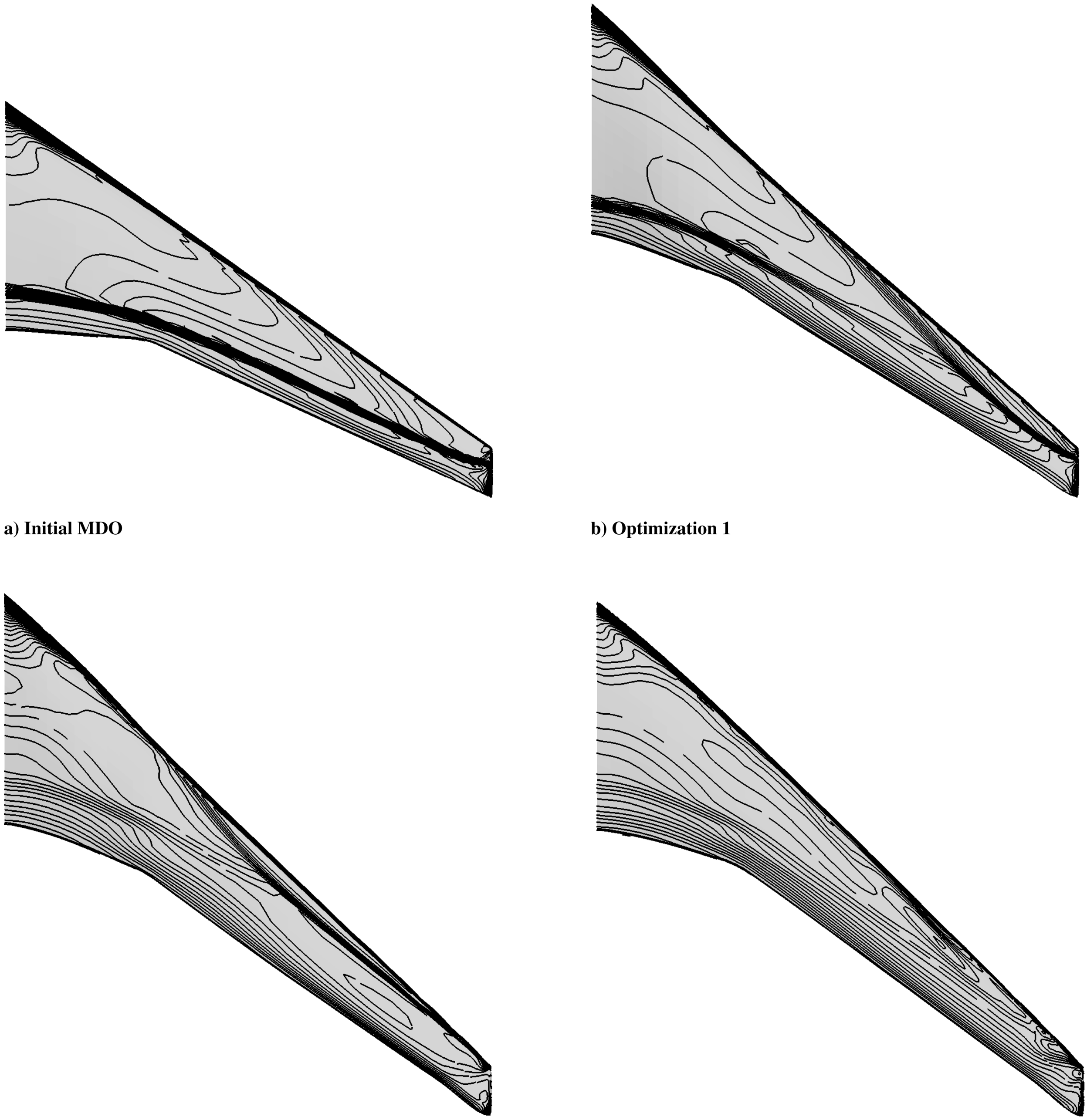
b) Drag

c) Volume

Fig. 6 Spanwise distributions.

Table 1 also shows the approximate number of flow solutions required, the relative runtime required, and the number of CPUs used for each case. The number of flow solutions is approximate, because the number required at the serial part of the code, computing the step size at each evolution, is not constant. Each processor can control its own parametric perturbation and corresponding mesh deformation and can spawn its own flow solver call. Hence, up to $2 \times N$ variables processors can be used (plus one required for the master process), using a serial call to the flow solver (the case here), as the positive and negative perturbations of each variable can be run on separate processors. The runtime speedup scales almost linearly with the number of processors, because there are no interprocessor communications required during the flow solutions. However, each flow solution may also be parallel (assuming that sufficient processors are available), and this would reduce the runtime further. Hence, this parallel approach has enabled high-level CFD shape optimization, even using a finite difference technique with serial calls to a flow solver, possible within practical runtimes.

Observation of the evolution histories in Figs. 4a–4c shows that optimizations could be halted even after only a few evolutions and significant improvements would still have been obtained. The 388-variable case could be stopped after only 15 evolutions, and a 16% drag reduction would still have been achieved. With only 30 design variables, optimization 1 finds its optimum quickly, requiring only 15 evolutions in total. This is a very interesting result, because an 8% reduction in drag has been achieved with only global changes (i.e., sections can move but not change profile). With only these select design variables, the optimization is constantly very close to three of the four constraint boundaries. When more design variables are activated (i.e., local surface changes are included), optimizations require more evolutions to reach optimal designs, but due to the additional degrees of freedom that the design can now deform in, optimizations start exploring the design space, surprisingly away from many of the constraint boundaries. The lift constraint is the only boundary for which all three optimizations find an optimum very close to the minimum-allowable value. Optimization has had the


a) Initial MDO

b) Optimization 1

c) Optimization 2

d) Optimization 3

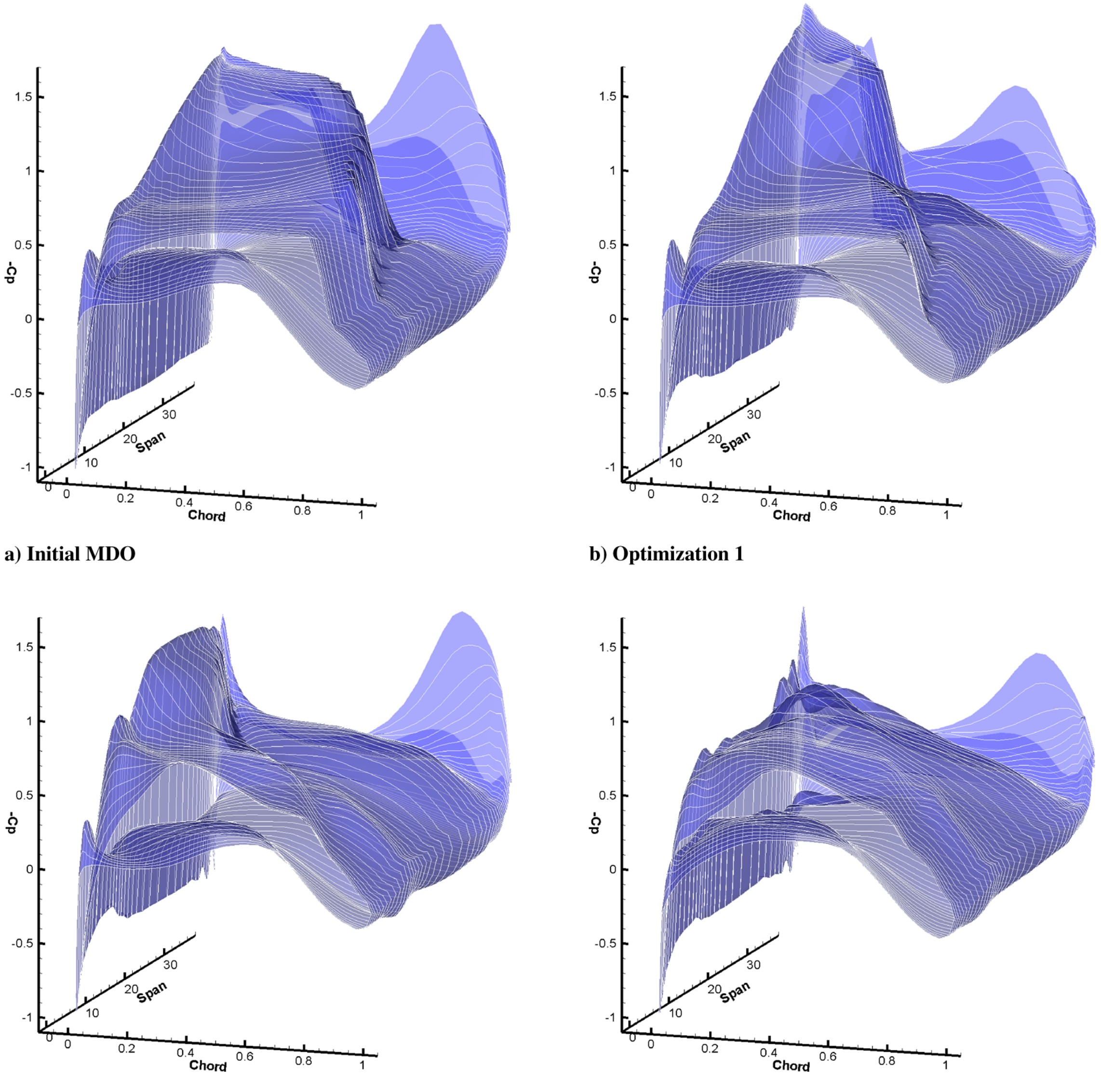
Fig. 7 *Cp* distributions.

beneficial effect of reducing root bending and root torsion moments, sometimes significantly. Wing internal volume has also increased slightly in all cases.

Initial and optimized domain-element and wing geometries are depicted in Figs. 5a–5c for optimizations 1, 2, and 3, respectively. The most notable changes to the optimized wings are to the sweep distribution; not only has sweep been increased in all cases to a similar extent, but the leading edges are no longer straight. This is surprising; root torsion moment is rigidly constrained and an increased sweep angle normally impacts negatively on this. Observation of Figs. 6 demonstrate that loading has moved significantly inboard, such that sweep angle can be increased in an effort to reduce drag with no penalty to root moments. Although cruise flight is not usually the determining case for structural wing-box design, reduced root aerodynamic moments could provide possible weight savings. Figures 6 also show that drag has increased slightly inboard due to its increased loading here, but with the exception of optimization 2, in which there is a small rise in drag around the 33% span region, the drag is significantly reduced at all outboard locations. It should be remembered here that even with the large amount of surface deformation shown in Figs. 5, the volume meshes are still computed as a deformation of the initial mesh and no mesh regeneration is required.

Figure 7 depicts contours of coefficient of pressure Cp for the initial MDO and optimized wings, and Fig. 8 depicts the sectional Cp distributions for the wings, transformed onto a rectangular wing of unit chord and original span length to enable comparison.

The MDO wing in cruise flight exhibits a strong shock along the entire length of the span. Optimization 1 has altered the pressure distribution significantly with now two clear shock patterns. The shock originating in the inboard sections is of similar strength and location to that of the original MDO; however, it starts to weaken significantly with span location. The second shock structure at the tip is comparatively nearer the leading edge than the original MDO wing and is slightly stronger. Without active design variables that can alter the airfoil profile geometry significantly, it was never anticipated that shock strength could be reduced significantly, but the reduction in drag by over 8% is considered to be substantial. Optimization 2, with active design variables controlling the three-dimensional planform (and, to a limited degree, airfoil profile changes), achieves over 14% reduction in drag. The main shock wave has been removed, but a small shock near the leading edge at outboard locations has been developed. Optimization 3, with planform and complete airfoil free-form control, achieves over 18% reduction in drag and results in a completely shock-free wing. This is a considerable result, considering the constraint on a high value of lift and at a high transonic



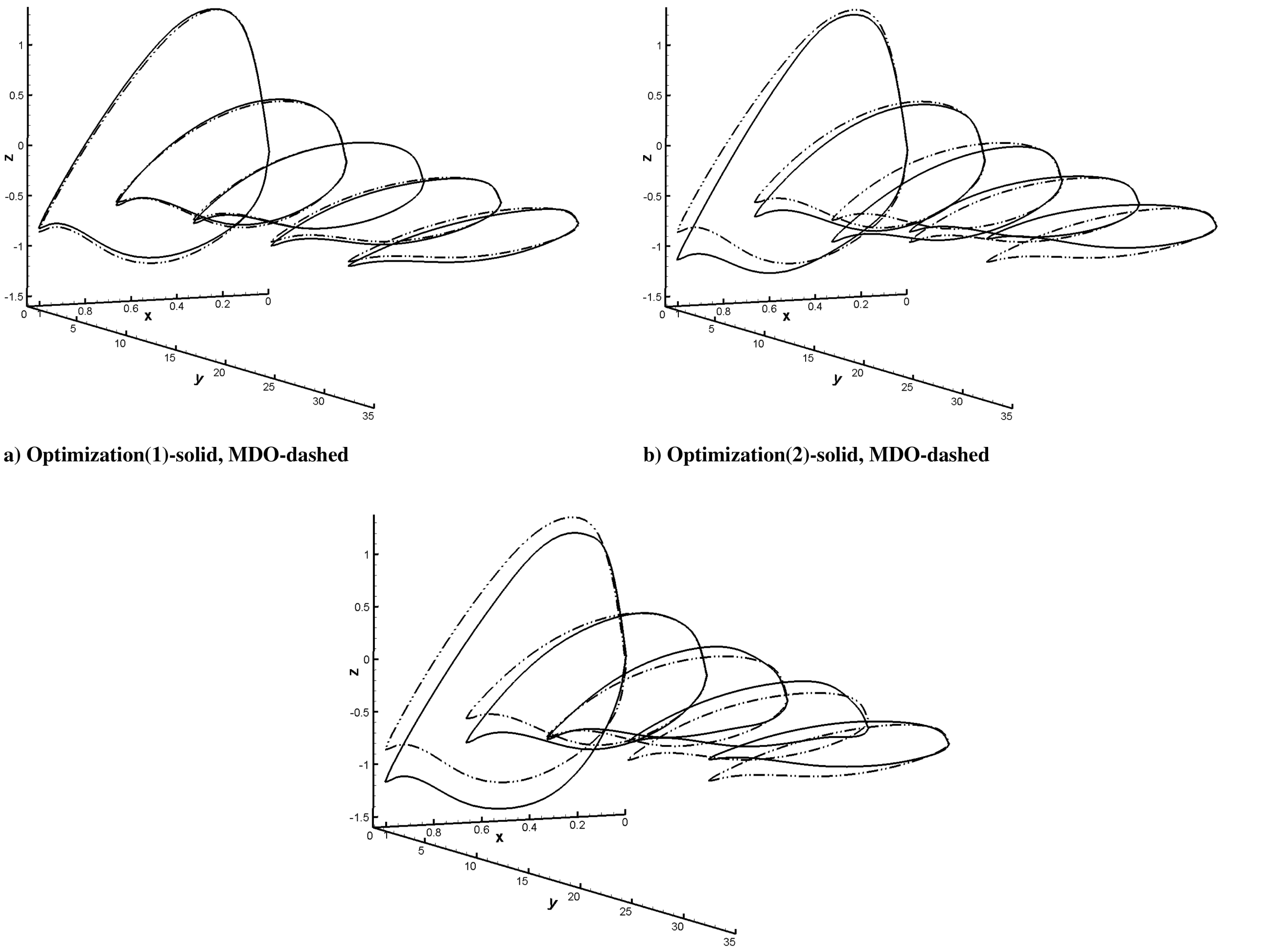
a) Initial MDO

b) Optimization 1

c) Optimization 2

d) Optimization 3

Fig. 8 Surface Cp distributions.



a) Optimization(1)-solid, MDO-dashed

b) Optimization(2)-solid, MDO-dashed

c) Optimization(3)-solid, MDO-dashed

Fig. 9 Airfoil profiles.

Mach number. From Fig. 8d, it is clear that the wing from optimization 3 has truly achieved an improvement in geometric design: a smooth Cp distribution that is completely shock-free along the entire span of the wing is obtained.

Sectional slices through the transformed wings are shown in Fig. 9 and compared with the initial MDO wing geometry. As anticipated, as the wing corresponding to optimization 1 is unable to change airfoil profile geometries, only a slight increase in wing angle of attack and twist are observed, in addition to the sweep change. However, significant airfoil section changes (and twist distribution) are clearly seen for optimizations 2 and 3. Root incidence is increased, but with a larger washout, highlighting that inboard sections are more highly loaded with relief toward the tip.

VII. Conclusions

A completely generic wraparound aerodynamic optimization tool has been developed and applied here to three-dimensional wing optimization. This comprises a new geometric parameterization technique for application to CFD-based aerodynamic optimization. The parameterization uses radial basis functions to interpolate positions of the domain element and the grid coordinates to provide simultaneous deformation of the design surface and its corresponding aerodynamic mesh. The interpolation and updates to the geometry and its associated CFD mesh are of extreme high quality, robust, fast, and efficient. This domain-element technique is independent of mesh topology and mesh generation package, requiring only an initial mesh.

Finite difference sensitivities are computed from the design parameter perturbations and fed into a feasible sequential quadratic programming gradient-based optimizer. Significantly, this approach allows application of multiple strict constraints. This optimization framework has been parallelized, such that each CPU can perform its own parametric perturbation and corresponding mesh deformation and can spawn its own flow solver calls, allowing optimization of three-dimensional bodies to be performed in practical runtimes. The entire optimization suite has been parallelized in a generic fashion such that the process is still independent of the flow solver, and using a serial flow solver, the speedup scales almost linearly with the number of CPUs. However, a parallel flow solution may also be used for each perturbation, and this would result in even faster computation.

The parameterization technique allows combination of variables of different scales and types with only a few parameterization nodes, and this leads to a significantly reduced number of design variables for three-dimensional applications when compared with many other types of shape parameterization method. The use of truly-three-dimensional planform design variables is novel and has led to significant planform adjustments in an effort to reduce drag. The effects of this allowed variable parameterization scale have been investigated here. Three aerodynamic optimizations of the MDO wing in transonic cruise have been presented, corresponding to different levels of active design variables ranging from 30 planform parameters to 388 global and local parameters. Of course, fewer than 30 parameters could have been used, but this would reduce the design space available, and it has been proven here that the space available increases with the number of parameters adopted. Alternative constraints could also have been applied, for example, constraining structural span rather than semispan. However, the important issue is demonstration of the parameterization approach and the effect of varying the number of parameters included. The lowest level adopted here still results in an 8% drag reduction, whereas the highest-level optimization results in a totally-shock-free wing with over 18% reduction in inviscid drag, combined with significantly reduced root aerodynamic moments.

Acknowledgments

The authors would like to thank AgustaWestland, Engineering and Physical Sciences Council (EPSRC), and the United Kingdom's Ministry of Defence Joint Grant Scheme funding under contract GR/S61294 for the partial funding of this research and the support of Asa Morris, and the University of Bristol, where Thomas Rendall was funded by a postgraduate research scholarship.

References

[1] Bloor, M. I. G., and Wilson, M. J., "Efficient Parameterization of Generic Aircraft Geometry," *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1269–1275.
doi:10.2514/3.46874

[2] Young, D. P., Huffman, W. P., Melvin, R. G., Bieterman, M. B., Hilmes, C. L., and Johnson, F. T., "Inexactness and Global Convergence in Design Optimization," 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, A/AA Paper 94-4286, Sept. 1994.

[3] Sobieczky, H., "Parametric Airfoils and Wings," *Notes on Numerical Fluid Mechanics*, Vol. 68, 1998, pp. 71–88.

[4] Pickett, R. M., Rubinstein, M. F., and Nelson, R. B., "Automated Structural Synthesis Using a Reduced Number of Design Coordinates," *AIAA Journal*, Vol. 11, No. 4, 1973, pp. 489–494.
doi:10.2514/3.50489

[5] Watt, A., and Watt, M., *Advanced Animation and Rendering Techniques*, Addison-Wesley, New York, 1992, Chap. 17.

[6] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.
doi:10.2514/3.58379

[7] Hicks, R. M., Murman, E. M., and Vanderplaats, G. N., "An Assessment of Aerofoil Design by Numerical Optimization," NASA Ames Research Center TMX-3092, Moffett Field, CA, July 1974.

[8] Smith, R. E., Bloor, M. I .G., Wilson, M. J., and Thomas, A. T., "Rapid Airplane Parametric Input Design (RAPID)" *12th AIAA Computational Fluid Dynamics Conference*, AIAA, Washington, D.C., 1995, pp. 452–462.

[9] Braibant, V., and Fleury, C., "Shape Optimal Design using B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, No. 3, Aug. 1984, pp. 247–267.
doi:10.1016/0045-7825(84)90132-4

[10] Kulfan, B. M., and Bussoletti, J. E., "Fundamental Parametric Geometry Representations for Aircraft Component Shapes," 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA Paper 2006-6948, Sept. 2006.

[11] Kulfan, B. M., "A Universal Parametric Geometry Representation Method—CST," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2007-62, Jan. 2007.

[12] Castonguay, P., and Nadarajah, S. K., "Effect of Shape Parameterization on Aerodynamic Shape Optimization," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2007-59, Jan. 2007.

[13] Nadarajah, S., Castonguay, P., and Mousavi, A., "Survey of Shape Parameterization Techniques and its Effect on Three-Dimensional Aerodynamic Shape Optimization," 18th AIAA Computational Fluid Dynamics Conference, Miami, AIAA Paper 2007-3837, June 2007.

[14] Morris, A. M., Allen, C. B., Rendall T. C. S., "Development of Generic CFD-Based Aerodynamic Optimization Tools for Helicopter Rotor Blades," 25th AIAA Applied Aerodynamics Conference, Miami, AIAA Paper 2007-3809, June 2007.

[15] Morris, A. M., Allen, C. B., Rendall, T. C. S., "CFD-based Optimization of Aerofoils using Radial Basis Functions for Domain Element Parameterization and Mesh Deformation," *International Journal for Numerical Methods in Fluids*, Vol. 58, No. 8, 2008, pp. 827–860.
doi:10.1002/fld.1769

[16] Zhou, J. L., Tits, A. L., and Lawrence, C. T., "User's Guide for FFSQP Version 3.7: A FORTRAN Code for Solving Optimization Programs, Possibly Minimax, with General Inequality Constraints and Linear Equality Constraints, Generating Feasible Iterates," Inst. for Systems Research, Univ. of Maryland, SRC-TR-92-107r5, College Park, MD, 1997.

[17] Zhou, J. L., and Tits, A. L., "Nonotone Line Search for Minimax Problems," *Journal of Optimization Theory and Applications*, Vol. 76, No. 3, 1993, pp. 455–476.
doi:10.1007/BF00939377

[18] Panier, E., and Tits, A. L., "On Combining Feasibility, Descent and Superlinear Convergence in Inequality Constrained Optimization," *Mathematical Programming*, Vol. 59, 1993, 261–276.

[19] Allwright, S., "Multi-Discipline Optimization in Preliminary Design of Commercial Transport Aircraft," *Computational Methods in Applied Sciences (ECCOMAS)*, Wiley, New York, Sept, 1996, pp. 523–526.

[20] Haase, D., Selmin, V., and Winzell, B., "Notes on Numerical Fluid Mechanics and Multidisciplinary Design," *Progress in Computational Flow-Structure Interaction*, Vol. 81, Springer, New York, 2002.

[21] Reuther, J., "Aerodynamic Shape Optimization Using Control Theory," NASA, CR-201064, 1996.

[22] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260.
doi:10.1007/BF01061285

[23] Jameson, A., "Automatic Design of Transonic Aerofoils to Reduce the Shock Induced Pressure Drag," *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics*, Feb. 1990, pp. 5–17.

[24] Samareh, J. A., "Status and future of geometry Modeling and Grid Generation for Design and Optimization," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 97–104.
doi:10.2514/2.2417

[25] 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.
doi:10.2514/2.1391

[26] Allen, C. B., and Rendall, T. C. S., "Unified CFD-CSD Interpolation and Mesh Motion Using Radial Basis Functions," 25th AIAA Applied Aerodynamics Conference, Miami, AIAA Paper 2007-3804, June 2007.

[27] Rendall, T. C. S., and Allen, C. B., "Unified Fluid-Structure Interpolation and Mesh Motion Using Radial Basis Functions," *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 10, 2008, pp. 1519–1559.
doi:10.1002/nme.2219

[28] Buhmann, H., *Radial Basis Functions*, 1st ed., Cambridge Univ. Press, New York, 2005.

[29] Wendland, H., *Scattered Data Approximation*, 1st ed., Cambridge Univ. Press, New York, 2005.

[30] Rendall, T. C. S., and Allen, C. B., "Efficient Mesh Motion Using Radial Basis Functions with Data Reduction Algorithms," Proceedings 46th AIAA Aerospace Sciences Meeting, Reno, NV, AIAA Paper 2008-305, Jan. 2008.

[31] Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: with Applications*, McGraw–Hill, New York, 1984.

[32] Schittkowski, K., "Test Examples for Nonlinear Programming Codes," Inst. for Information Technology, Univ. of Stuttgart, Stuttgart, Germany, 1984.

[33] Wong, W. S., Le Moigne, A., and Qin, N., "Parallel Adjoint-Based Optimization of a Blended Wing Body Aircraft with Shock Control Bumps," *The Aeronautical Journal*, Vol. 111, No. 1117, Mar. 2007, pp. 165–174.

[34] Qin, N., Vavalle, A., and Le Moigne, A., "Spanwise Lift Distribution for Blended Wing Body Aircraft," *Journal of Aircraft*, Vol. 42, No. 2, 2005, pp. 356–365.
doi:10.2514/1.4229

[35] Wong, W. S., Qin, N., Sellars, N., Holden. H., and Babinsky, H., "A Combined Experimental and Numerical Study of Flow Structures over Three-Dimensional Shock Control Bumps," *Aerospace Science and Technology*, Vol. 12, No. 6, 2008, pp. 436–447.
doi:10.1016/j.ast.2007.10.011

[36] Qin, N., Vavalle, A., Le Moigne, A., Laban, M., Huckett, K., and Weinerfelt, P., "Aerodynamic Studies of Blended Wing Body Aircraft," 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference, AIAA Paper 2002-5448, Sept. 2002.

[37] Epstein, B., Jameson, A., Peigin, S., Roman, D., Harrison, N., and Vassberg, J., "Comparative Study of 3D Wing Drag Minimization by Different Optimization Techniques," 46th AIAA Aerospace Sciences Meeting, Reno, NV, AIAA Paper 2008-326, 2008.

[38] Allen, C. B., "Towards Automatic Structured Multiblock Mesh Generation Using Improved Transfinite Interpolation," *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 5, 2008,

pp. 697–733.
doi:10.1002/nme.2170

[39] Allen, C. B., “Parallel Simulation of Unsteady Hovering Rotor Wakes,” *International Journal for Numerical Methods in Engineering*, Vol. 68, No. 6, 2006, pp. 632–649.
doi:10.1002/nme.1723

[40] Allen, C. B., “Convergence of Steady and Unsteady Formulations for Inviscid Hovering Rotor Solutions,” *International Journal for Numerical Methods in Fluids*, Vol. 41, No. 9, 2003, pp. 931–949.
doi:10.1002/fld.474

[41] Allen, C. B., “An Unsteady Multiblock Multigrid Scheme for Lifting Forward Flight Rotor Simulation,” *International Journal for Numerical Methods in Fluids*, Vol. 45, No. 9, 2004, pp. 973–984.
doi:10.1002/fld.711

[42] Allen, C. B., “Parallel Universal Approach to Mesh Motion and Application to Rotors in Forward Flight,” *International Journal for Numerical Methods in Engineering*, Vol. 69, No. 10, 2007, pp. 2126–2149.
doi:10.1002/nme.1846

[43] van Leer, B., “Flux Vector Splitting for the Euler Equations,” *Lecture Notes in Physics*, Vol. 170, 1982, pp. 507–512.

[44] Parpia, I. H., “Van-Leer Flux Vector Splitting in Moving Coordinates,” *AIAA Journal*, Vol. 26, No. 1, 1988, pp. 113–115.
doi:10.2514/3.9858

[45] Allen, C. B., “Multigrid Convergence of Inviscid Fixed- and Rotary-Wing Flows,” *International Journal for Numerical Methods in Fluids*, Vol. 39, No. 2, 2002, pp. 121–140.
doi:10.1002/fld.282

E. Livne
Associate Editor