Report no. 97/08

Aerodynamic design optimisation for complex geometries using unstructured grids

M. B. Giles

These lecture notes, prepared for the 1997 VKI Lecture Course on Inverse Design, discuss the use of unstructured grid CFD methods in the design of complex aeronautical geometries. The emphasis is on gradient-based optimisation approaches. The evaluation of approximate and exact linear sensitivities is described, as are different ways of formulating the adjoint equations to greatly reduce the computational cost when dealing with large numbers of design parameters.

The current state-of-the-art is illustrated by two examples from turbomachinery and aircraft design.

Key words and phrases: design, optimisation, sensitivity analysis, adjoint methods, unstructured grids

This work was supported by Rolls-Royce plc and EPSRC.

Oxford University Computing Laboratory
Numerical Analysis Group
Wolfson Building
Parks Road
Oxford, England OX1 3QD
http: //www.comlab.ox.ac.uk
email: giles@comlab.oxford.ac.uk

February, 2000

1 Introduction

For simple aeronautical applications such as aircraft wing/fuselage analysis and turbomachinery cascade analysis, the standard CFD approach is to use a structured grid, often containing just a single structured block. In these cases, the grid generation is relatively easy and the structured grid CFD methods have the advantages of better accuracy and lower computational cost compared to unstructured grid methods.

However, in aeronautical design one is often concerned with much more complex geometries. With aircraft, in addition to the wing and fuselage one might wish to include the tail, winglets, the engines and their pylons, the underwing track fairings for the flaps and even the extended flaps in the high-lift configuration. In turbomachinery, there are complex geometries in the bypass ducts of turbofan engines, combustors, internal cooling passages within turbine blades, etc. For such applications, the generation of an appropriate multi-block structured grid becomes a difficult time-consuming task. Furthermore, the topological restrictions of such grids usually result in excessive grid resolution in some parts of the flow field in order to achieve adequate grid resolution in other areas.

It is for such applications with complex geometries that the use of unstructured grid CFD algorithms is most appropriate. Given a valid CAD/CAM definition of the complex object as a set of intersecting solids/surfaces, there are now grid generators which can produce a good quality grid with one million tetrahedra for inviscid flow analysis in ten minutes on a standard workstation. There is still considerable research on the issues of viscous grid generation and simplifying the user specification of the desired grid resolution in different parts of the grid. However, steady progress is being made in these areas so that it will soon be possible to create high quality unstructured grids for viscous flow analysis with a minimum of user intervention. The fact that one can ensure that all of the unstructured grid points are where they are needed to resolve the flow features, offsets the inherent poorer accuracy of these unstructured grid discretisation compared to those using smoothly varying structured grids. In addition, grid adaptation through insertion of additional grid points is very easily accomplished. Thus, for a given level of the accuracy, the computational cost of unstructured grid CFD algorithms is comparable to that of multi-block structured grid algorithms. This takes into account advances in multigrid acceleration algorithms which now give the same speedup benefits for unstructured grids as for structured grids.

It is not my intention here to present a comprehensive review of the substantial literature on unstructured grid flow algorithms; there are many good review articles on this subject. Instead, my objective is to discuss the use of such methods for the purposes of design. The next section briefly outlines the standard analysis approach, and the way in which it can be used within stochastic optimisation.

Thereafter, the focus is on gradient-based optimisation methods in which the optimisation of an objective function is accomplished using information about its gradient with respect to the design parameters. This gradient can be derived in a number of different ways. Finite differencing the results of nonlinear flow computations is the simplest. To obtain accurate approximate derivatives requires the use of consistently perturbed grids. Grid movement algorithms to accomplish this are described in the third section.

An alternative approach to obtain the linear sensitivities is exact linearisation leading to the discrete adjoint equations which is the subject of the fourth section. This section also discusses the analytic formulation of the adjoint equations, which can then be discretised to again obtain approximate linear sensitivities at low computational cost.

The final section presents some examples of the use of unstructured grid methods for design applications. These 3D examples are all very recent, illustrating the current state of research in this area.

2 Analysis methods and stochastic optimisation

An unstructured grid CFD analysis system has two main components, the grid generator and the flow code.

Grid generation for viscous flow analysis is usually accomplished in three phases, the second of which is omitted when the grid is to be used for inviscid flow analysis:

- Surface

  The gridding of the surface starts by placing grid points along key lines, such as the leading edge of the wing, the intersection of the fuselage with the symmetry plane, the intersection of a blade with the hub or tip annuli, etc. These lines divide the surface of the complex object into a number of pieces, each of which is then triangulated, often using an advancing front algorithm.

- B.l./wake

  This is still an active research area, but at present the most successful methods generate the grid for the boundary layer and wake by advancing outwards from the triangulated surface along lines which are approximately normal to the surface, creating new grid points and generating prisms which are subsequently cut into tetrahedra [5, 17].

- Interior

  Having also triangulated any domain boundary surfaces (far-field, symmetry, inflow/outflow) the final step is to create the grid points and tetrahedra in the remainder of the interior of the domain. The fastest methods for this are based on the Delauney algorithm [32], but advancing front methods are also used because they offer additional flexibility in controlling the quality of the grid that is generated

In each phase the user has to control the grid resolution which is desired; the surface grid density in the first phase, the boundary layer resolution in the second, and the volumetric grid density in the third. In 2D applications, the grid density is often controlled through the use of a 'background grid', a triangular grid used to define a piecewise linear spacing function $\delta(x,y)$ [22]. To ensure good resolution at the leading edge of a wing, one would place a grid point of the background grid inside the leading edge circle, and specify a small spacing value at that point. In 2D it is relatively easy to create this background grid, but in 3D it is much harder. One approach is to define the surface grid density based on the local curvature and then interpolate from this to define the interior grid spacing [33]. In principle, this is a good idea, but in practice there are problems with it producing too many grid points in some regions and too few in others. An alternative approach is to define a default spacing, and then use a combination of point, line and planar 'sources' to define increased grid resolution in certain regions [31, 33]. Using this technique, increased grid resolution near the leading edge of a 3D wing is accomplished by placing a spanwise line source inside the leading edge of the wing. The boundary layer resolution is controlled by specifying a boundary layer grid thickness function on the surface of the object (possibly using a 'background grid' on each piece of the surface) and specifying the desired number of grid points across the boundary layer.

Unstructured grid flow solvers are also the subject of much current research. There are now a number of well established algorithms for discretising the Euler equations on tetrahedral grids. There are also well-developed multigrid algorithms for accelerating the steady-state convergence of Euler computations, using non-nested grids [6, 21], cell agglomeration [28], edge-collapsing [7] and other methods. Methods for parallelising such flow codes have also been developed [8, 9], so that it is now possible to achieve four orders of magnitude convergence (sufficient for engineering accuracy) on a grid with 1 million tetrahedra in 1 hour on an 8-processor parallel computer. Thus, inviscid flow calculations for complex geometries using large grids are becoming a practical engineering tool.

It is the viscous algorithms on unstructured grids which require further development. The discretisations on the highly stretched tetrahedral grids necessary for high Reynolds number applications are still not as accurate as those for structured grids. It is possible the solution is the use of prism cells in the boundary layer [20]. There is scope for improving the multigrid algorithms as well, as so far they do not achieve the same speedup as the multigrid methods for viscous computations on structured grids. In the longer term, it is increasingly believed that the optimum solution will be a hybrid method, using a combination of structured and unstructured grids, benefitting from the accuracy and low computational cost of body-fitted structured grids in most of the flow domain, together with the flexibility of unstructured grids in regions of geometric complexity.

Whether or not the future lies with simple tetrahedral grids or more complicated hybrid grids, it seems likely that in the next 5 years Reynolds-averaged Navier-Stokes computations for complex geometries will start to become a routine engineering analysis tool. This then raises the question of how best to use these for design optimisation. Conceptually, the simplest approach is to use genetic algorithms [10] or some other form of stochastic optimisation. All these optimisation methods require is the ability to calculate the flow field given the values of a set of design parameters.

However, in practice, there are two major difficulties. One is due to the grid generation process described above. Suppose that the CAD/CAM system can automatically generate the new solid body geometry given the values of the design parameters. The problem is that the grid generator cannot automatically generate a new grid because at present user input is needed to define the required grid resolution, modifying the background grid or point/line sources as necessary. For example, if the sweep angle of the aircraft wing is changed (or lean of the turbine blade stacking) then the leading edge of the wing or blade will be in a different place, and so the line source placed inside the leading edge will need to be moved. One solution to this problem is the use of grid movement (to be described in the next section) to generate a deform an initial grid as the design parameters change. However, this forces the grid to remain topologically identical which in turn prevents consideration of design parameter values which change the topology of the aircraft. The other possible solution is an advance in grid generation techniques to automatically define the grid resolution (for a given overall number of grid points) based on the geometry of the aircraft or turbomachine, using information such as the surface curvature.

The second problem is the computational cost. Stochastic methods require several hundred flow calculations. If all of these are performed on fine grids with the resolution needed to produce answers with acceptable engineering accuracy, the total cost is prohibitive and looks likely to remain so for the next twenty years. A possible solution to this problem is to use coarser grids (or the Euler equations instead of the Navier-Stokes equations) during the initial phases of the optimisation, switching progressively to the finer grids (and more accurate modelling) as the design converges towards the optimum. In this way, the more accurate (and more expensive) results can be used to 'correct' the more approximate values. This is a promising line of research which could greatly reduce the cost of stochastic optimisation. However, at present, I believe the most cost-effective approach to optimising complex geometries lies with classical optimisation methods which use the gradient of the objective function with respect to the design parameters.

3 Approximate sensitivities and grid movement

One simple way in which to calculate the approximate sensitivity of the flow field to changes in each design parameters is through finite differencing of the solutions from a number of nonlinear computations [25, 26]. Thus, for each set of design parameters $\boldsymbol{\alpha}$, the discrete flow equations

$$\boldsymbol{F}(\boldsymbol{U}, \boldsymbol{\alpha}) = 0,$$

are solved to implicitly obtain the flow field $\boldsymbol{U}$ as a function of the design parameters $\boldsymbol{\alpha}$. To obtain the gradient of an objective function $I(\boldsymbol{U}, \boldsymbol{\alpha})$ when there are N design parameters requires at least $N+1$ calculations. The simplest approach is to use one-sided differencing so that the derivative of the objective function with respect to the k^{th} design parameter is approximated by

$$\frac{dI}{d\alpha_k} \approx \frac{I(\boldsymbol{U}(\boldsymbol{\alpha}+\epsilon_k \boldsymbol{e}_k), \boldsymbol{\alpha}+\epsilon_k \boldsymbol{e}_k) - I(\boldsymbol{U}(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\epsilon_k}$$

where $\boldsymbol{e}_k$ is a vector whose elements are zero except for the k^{th} which is unity, and ϵ_k is some suitably small perturbation.

Perturbing the design parameters changes the surface geometry of the aircraft or turbomachine, and hence perturbs the grid. If the new grid is created by the standard unstructured grid generator, there is the possibility that the new grid will be topologically different to the original. This could lead to significant errors in the approximate sensitivity that is computed. A change in the topology will lead to a small discontinuity in the value of the objective function; if ϵ_k is very small then in the worst case this discontinuity in objective function will lead to a very large error in the approximate gradient. The other problem with using the standard grid generator to create each grid is the one raised in the previous section, that significant user input is often required to ensure good grid resolution in the regions that require it.

A better approach is to create a perturbed grid with the same topology as the original, by perturbing the coordinates of the grid points of the original grid in a way which is consistent with the perturbation to the surface geometry of the aircraft. Referring back to the three phases of grid generation outlined in the last section, the first step is to define perturbations to the grid nodes lying on the surface. In practice, this is also the hardest step. The methods which are currently used [12, 25, 26] employ simple algebraic functions to perturb the surface points. These perturbations are not compatible with the standard solid body representations in CAD/CAM software and so the body which is designed would have to be modified to be stored within the CAD/CAM system.

A much better solution would be to interface to the CAD/CAM system directly. Maintaining the relative spacing along the key lines such as the leading edge line would define the movement of the grid points as the lines themselves move in response to changes in a design parameter. On each surface patch, the geometry is represented in parametric form as $(x(\xi,\eta), y(\xi,\eta), z(\xi,\eta))$ where ξ, η are general surface coordinates. Knowing the $\widetilde{\xi}, \widetilde{\eta}$ perturbations to the points along the bounding lines, corresponding perturbations to the other grid points on the surface could be constructed using either the method of 'springs' or the elliptic p.d.e. approach to be described shortly. Having determined the $\widetilde{\xi}, \widetilde{\eta}$ perturbations to the surface grid points, the corresponding Cartesian perturbations $\widetilde{x}, \widetilde{y}, \widetilde{z}$ are then easily evaluated.

Once the surface perturbations are constructed it is relatively easy to construct corresponding coordinate perturbations for all interior grid points, including those in the boundary layer and wake region. The method of springs considers all edges in the grid to be springs [2, 29]. The base grid is defined to be in equilibrium due to the addition of nodal forces. The displacement of the surface grid points upsets this equilibrium, and so the perturbation of the interior grid points is defined to re-establish an equilibrium of forces acting on each interior grid point. It is common to define the spring constant for each spring to be inversely proportional to the length of the edge. The effect of this is to maintain the grid spacing across a boundary layer and prevent the formation of a grid singularity leading to cells with negative volumes.

The alternative approach is to compute the coordinate perturbation $\widetilde{\boldsymbol{x}}(\boldsymbol{x})$ by solving an elliptic p.d.e. of the form

$$\nabla \cdot (k(\boldsymbol{x}) \nabla \widetilde{\boldsymbol{x}}) = 0,$$

subject to Dirichlet boundary conditions [7]. The choice of an elliptic p.d.e. ensures a smooth perturbation. Using a standard Galerkin finite element discretisation, the diffusivity factor k is chosen to be proportional to the cell volume of the perturbed grid. This ensures that the volume of the cells in the boundary layer is almost unchanged, maintaining the desired grid resolution across the boundary layer, and again prevents the creation of cells with negative volume.

4 Linearisation and adjoint equations methods

Mathematically, the simplest form of linear analysis is equivalent to the nonlinear analysis in the limit as $\epsilon_k \to 0$. If we define $\widetilde{\boldsymbol{U}}_k$ to be the sensitivity of $\boldsymbol{U}$ to changes in the k^{th} design parameter, then linearising the nonlinear discrete equations yields

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} \widetilde{\boldsymbol{U}}_k + \frac{\partial \boldsymbol{F}}{\partial \alpha_k} = 0.$$

The term $\frac{\partial \boldsymbol{F}}{\partial \alpha_k}$ arises from the fact that the the flux residuals depend on the coordinates of the grid points, represented collectively by the symbol $\boldsymbol{X}$, and the grid in turn depends on the design parameters. Hence, by the chain rule one obtains

$$\frac{\partial \boldsymbol{F}}{\partial \alpha_k} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{X}}{\partial \alpha_k}.$$

where $\frac{\partial \boldsymbol{X}}{\partial \alpha_k}$ represents the linearised grid sensitivity to perturbations in the k^{th} design parameter. In principle, one could define $\frac{\partial \boldsymbol{X}}{\partial \alpha_k}$ to be non-zero only at surface points. Such an approach was adopted in the first methods for linearised unsteady potential flow analysis [30, 34] but it was later discovered that using a continuously deforming grid gave improved numerical accuracy [13, 35]. Thus it is probable that here too one will obtain improved accuracy from using $\frac{\partial \boldsymbol{X}}{\partial \alpha_k}$ defined from a smooth grid perturbation.

The linear perturbation equation can be solved by the standard multigrid method to obtain $\widetilde{\boldsymbol{U}}_k$. The total derivative of an objective function with respect to the k^{th} design parameter is then given by

$$\frac{dI}{d\alpha_k} = \frac{\partial I}{\partial \boldsymbol{U}} \widetilde{\boldsymbol{U}}_k + \frac{\partial I}{\partial \alpha_k}.$$

The drawback with this approach is that it involves the development of an entirely new CFD code, and since it still requires a separate calculation for each design parameter it offers no benefits in computational cost compared to using nonlinear computations to obtain the approximate sensitivities. The one exception to this is an unusual turbomachinery application in which one designs a blade row with a sinusoidal circumferential variation in camber with the aim of producing a corresponding pressure variation cancelling that produced by a single large pylon. In this case the nonlinear analysis must be performed for the full annulus whereas the linear analysis can be performed using complex variables on a single blade passage with a complex phase shift between its two periodic boundaries [25, 26].

The desire to reduce the computational cost leads to the use of the adjoint equations. The discrete adjoint approach [12, 11, 19] starts from the linear equations above and eliminates $\widetilde{\boldsymbol{U}}_k$ to obtain

$$\frac{dI}{d\alpha_k} = -\frac{\partial I}{\partial \boldsymbol{U}} \left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} \right)^{-1} \frac{\partial \boldsymbol{F}}{\partial \alpha_k} + \frac{\partial I}{\partial \alpha_k}.$$

This can then be written as

$$\frac{dI}{d\alpha_k} = \boldsymbol{V}^T \frac{\partial \boldsymbol{F}}{\partial \alpha_k} + \frac{\partial I}{\partial \alpha_k},$$

where the vector $\boldsymbol{V}$ satisfies the equation

$$\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} \right)^T \boldsymbol{V} + \left(\frac{\partial I}{\partial \boldsymbol{U}} \right)^T = 0.$$

The great advantage of this approach is that one only needs to solve a single finite difference equation to get the sensitivities of I with respect to all of the design parameters. This is because the same solution $\boldsymbol{V}$ is used for each value of k. The only additional cost for each design parameter is the computation of $\frac{\partial \boldsymbol{F}}{\partial \alpha_k}$ and $\frac{\partial I}{\partial \alpha_k}$, which is inexpensive, and the dot product $\boldsymbol{V}^T \frac{\partial \boldsymbol{F}}{\partial \alpha_k}$ which is even cheaper.

The drawback of the adjoint approach is that a separate adjoint equation must be solved for each objective function or constraint function. Hence, in a highly constrained design in which the number of active constraints is comparable with the number of active design parameters, there is little to be gained from the adjoint approach.

The label 'adjoint' comes from the alternative treatment in which one starts with the linearised partial differential equation and converts the linear sensitivity of the objective function into an equivalent form involving the solution of the adjoint partial differential equation with appropriate boundary conditions [18]. This can then be discretised and solved numerically [1, 3, 4, 14, 15, 16, 23, 27]. The theoretical development of the adjoint p.d.e. will not be presented here. Instead we consider the formulation of the linearised equations. In almost all of the references listed above, the linearised equations are formulated using curvilinear coordinates (ξ, η, ζ) such that the airfoil is always a coordinate surface, such as $\eta = 0$. This approach is appropriate for developing CFD methods using a structured grid with a single grid block, but a different approach is necessary when using unstructured grids.

For simplicity, we consider the 2D Euler equations which may be written in conservation form are

$$\frac{\partial}{\partial x} F_x(U) + \frac{\partial}{\partial y} F_y(U) = 0,$$

where U is the vector of conservation variables and $F_x(U)$ and $F_y(U)$ are nonlinear flux functions,

$$U = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho E \end{pmatrix}, \quad F_x = \begin{pmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ \rho u_x H \end{pmatrix}, \quad F_y = \begin{pmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + p \\ \rho u_y H \end{pmatrix}.$$

Simply linearising about a given steady-state solution, $U(x, y)$, leads to the equation

$$Lu \equiv \frac{\partial}{\partial x}(A_x u) + \frac{\partial}{\partial y}(A_y u) = 0,$$

where u is defined to be the linear change in the flow solution at a point with fixed coordinates (x, y), and the spatially varying matrices A_x, A_y are defined by

$$A_x \equiv \left. \frac{\partial F_x}{\partial U} \right|_{U(x,y)}, \quad A_y \equiv \left. \frac{\partial F_y}{\partial U} \right|_{U(x,y)}.$$

This approach, extended to the Navier-Stokes equations, is the one used in Reference [1]. However, this definition of u leads to difficulties in approximating the boundary conditions on a perturbed surface, since a point which used to be on the surface may no longer be. For the Navier-Stokes equations, the no-slip boundary condition requires that

$$u_x = u_y = 0.$$

Linearising these conditions when the surface is displaced through distance $\widetilde{\boldsymbol{x}}$ gives

$$\widetilde{u}_x + \widetilde{\boldsymbol{x}} \cdot \nabla u_x = \widetilde{u}_y + \widetilde{\boldsymbol{x}} \cdot \nabla u_y = 0.$$

Thus, the boundary conditions for the linearised problem involve the evaluation of gradients of the original base solution. In the boundary layer these gradients are very large, resulting in significant errors. This approach, and its weaknesses, are similar to the classical treatment of linear unsteady flow analysis, as discussed earlier.

To avoid these problems, what is needed is a treatment which uses a grid which deforms linearly in a way which is consistent with the design changes to the surface geometry. Therefore, u is defined to be the linear perturbation in the flow solution taking into account a linear perturbation in the coordinates. The starting point for this formulation is the conservative form of the Euler equation using general curvilinear coordinates,

$$\frac{\partial}{\partial \xi}\left(F_x \frac{\partial y}{\partial \eta} - F_y \frac{\partial x}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(F_y \frac{\partial x}{\partial \xi} - F_x \frac{\partial y}{\partial \xi}\right) = 0.$$

We now define the perturbed coordinates as

$$x = \xi + \alpha X(\xi,\eta), \qquad y = \eta + \alpha Y(\xi,\eta),$$

where α is a design variable. $X(\xi,\eta)$ and $Y(\xi,\eta)$ are smooth functions which match the surface perturbations due to the design variable, so that a point (ξ,η) which is initially on a solid surface remains so as the design variable changes. Linearising with respect to α yields

$$\frac{\partial}{\partial \xi}(A_x u) + \frac{\partial}{\partial \eta}(A_y u) = -\frac{\partial}{\partial \xi}\left(F_x \frac{\partial Y}{\partial \eta} - F_y \frac{\partial X}{\partial \eta}\right) - \frac{\partial}{\partial \eta}\left(F_y \frac{\partial X}{\partial \xi} - F_x \frac{\partial Y}{\partial \xi}\right),$$

where u is now the perturbation in the flow variables for fixed (ξ,η) rather than fixed (x,y), and the fluxes F_x and F_y are based on the unperturbed flow variables U.

Switching notation from (ξ,η) back to (x,y) then produces an inhomogeneous p.d.e. of the form $Lu = f$, where L represents the differential operator. In essence, this treatment is very similar to that used by Jameson and others for single-block Euler and Navier-Stokes computations. The difference is that in Jameson's formulation the solid surface corresponds to part of the coordinate surface $\eta = 0$, whereas in this formulation the solid surface is the original surface defined in Cartesian coordinates. As a consequence, this new formulation can be used with an unstructured grid discretisation for complex geometries.

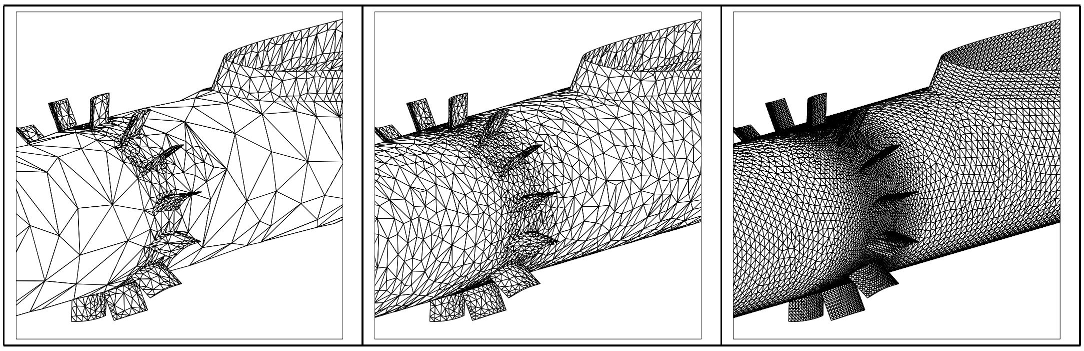


Figure 1: sequence of grids used by multigrid (outer annulus not plotted)

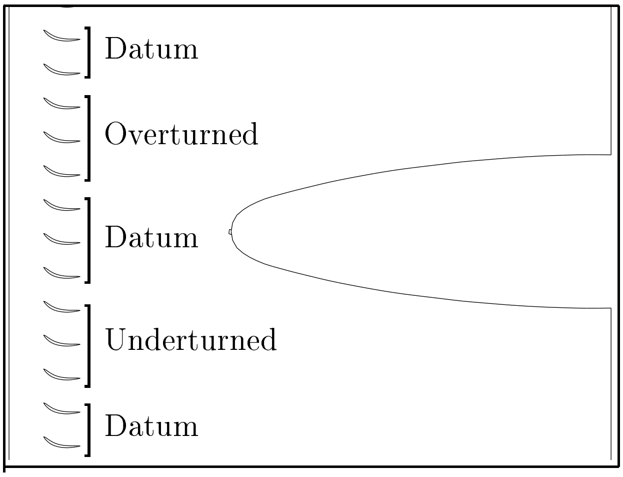


Figure 2: grouping of OGV's for 3 blade design

5 Examples

5.1 Outlet guide vane optimisation

Shrinivas has used nonlinear approximate sensitivities for a 3D design application concerning the bypass duct of a turbofan aeroengine [24]; this is an extension of earlier research by Shrinivas and Giles using 2D modelling [25, 26].

Figure 1 shows the geometry of the bypass duct and three of the grids used for the multigrid acceleration. For clarity, only the inner annulus of the duct is not plotted. Figure 2 displays an 'unwrapped' circumferential view of the mid-span geometry, halfway between the inner and outer annuli. There is a large pylon which is the main structural support for the engine core. Upstream of the pylon is a set of outlet guide vanes (OGVs) and upstream of these would be the rotating fan in the actual engine. The fan generates a circumferential component of flow velocity and the purpose of the OGVs is to turn the flow back in the axial direction. The design problem is that the very large pylon causes a blockage which produces a pressure field which decays very slowly in the axial direction. The OGVs shield it to some extent, but nevertheless there is a significant circumferential pressure variation upstream of the OGVs. In the engine this leads to an unsteady interaction with the rotating fan, producing higher stress levels and reduced aerodynamic efficiency.

The objective of the design process is to reduce this interaction to a minimum by re-designing the OGV's to counteract the pressure field created by the pylon. The objective function is a discrete approximation to the following integral of the circumferential pressure variation on a plane upstream of the OGVs.

$$I = \iint \left(p(r,\theta) - \overline{p}(r)\right)^2 d\theta\, dr$$

where $\overline{p}(r)$ represents the circumferentially averaged pressure at a particular radius.

The inviscid flow code that was used in this work was developed by Crumpton [7]. It uses an edge-based discretisation of the Euler equations and a standard Runge-Kutta time-marching algorithm. Edge-collapsing is used to generate the coarser grids for the multigrid algorithm. The execution speed is further improved through parallel computing on machines such as the IBM SP2 using the OPlus parallel library [8].

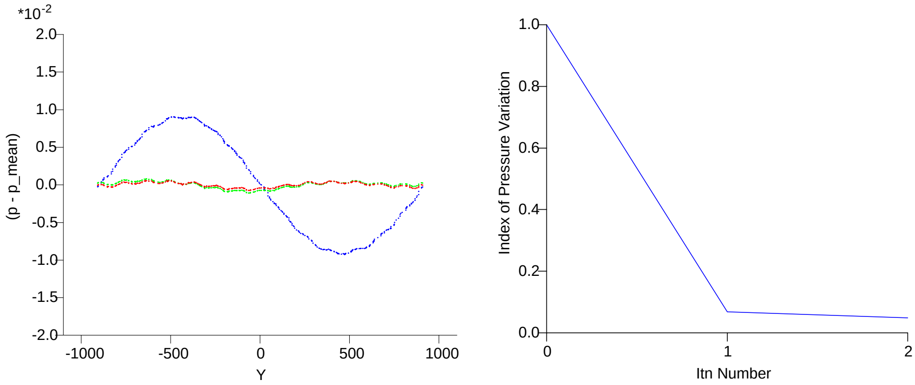
Two design exercises have been conducted. In each case, the camber of the OGVs is altered through a circumferential displacement $\Delta\theta$ which varies quadratically in the axial direction and linearly in the spanwise direction,

$$\Delta\theta = \left(x - x_{l.e.}(r)\right)^2 (ar + b)$$

with $x_{l.e.}(r)$ being the axial location of the leading edge.

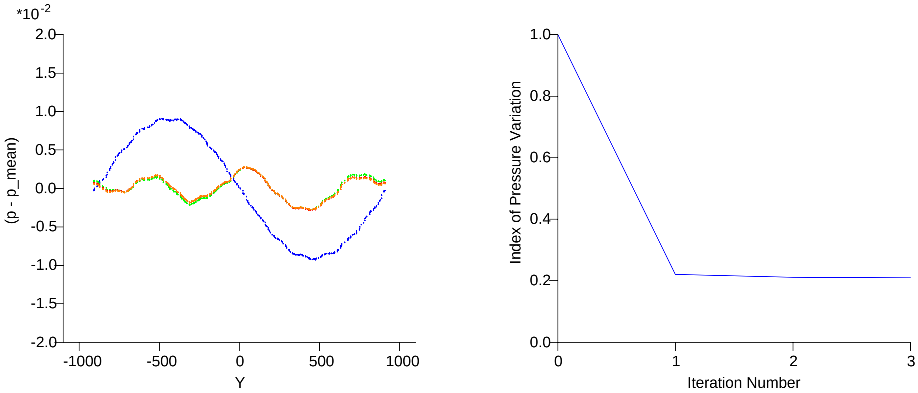
In the first design exercise, the constants a, b vary sinusoidally from one OGV to the next, with the OGVs nearest to the pylon and farthest from it having zero perturbations. This is appropriate because of the symmetry of the design problem. Thus, there are just 2 design parameters, the values for a and b for the blade with maximum displacement. Figure 3 shows the decrease in the level of circumferential pressure variation at mid-span, and the associated decrease in the value of the objective function. Because the objective function is approximately quadratic, and the method of direct sensitivities provides a very good estimate of the Hessian, the design optimum is almost achieved in one iteration.

From a practical engineering viewpoint, this design is far from ideal because it requires each OGV to be unique, increasing the cost of manufacture and the number of spare parts the airlines must keep. The second design exercise addresses this by allowing only 3 blade types, the original datum blade, an overturned blade with increased camber and an underturned blade with decreased camber. Figure 2 shows the chosen grouping of these blades. There are still just two



a) Cp at mid radius b) convergence of I

Figure 3: Optimisation using sinusoidal camber variation



a) C_p at mid radius b) convergence of I

Figure 4: Optimisation using 3 blade types

design parameters, the constants a, b for the over-turned blade; the underturned blade uses constants $-a, -b$ giving a camber perturbation of equal magnitude but opposite sign. Figure 4 shows that the design iteration still achieves near convergence in just one iteration. As one would expect, the restriction of using just 3 blade types means that the optimum solution has a larger remaining circumferential pressure variation than in the first design case.

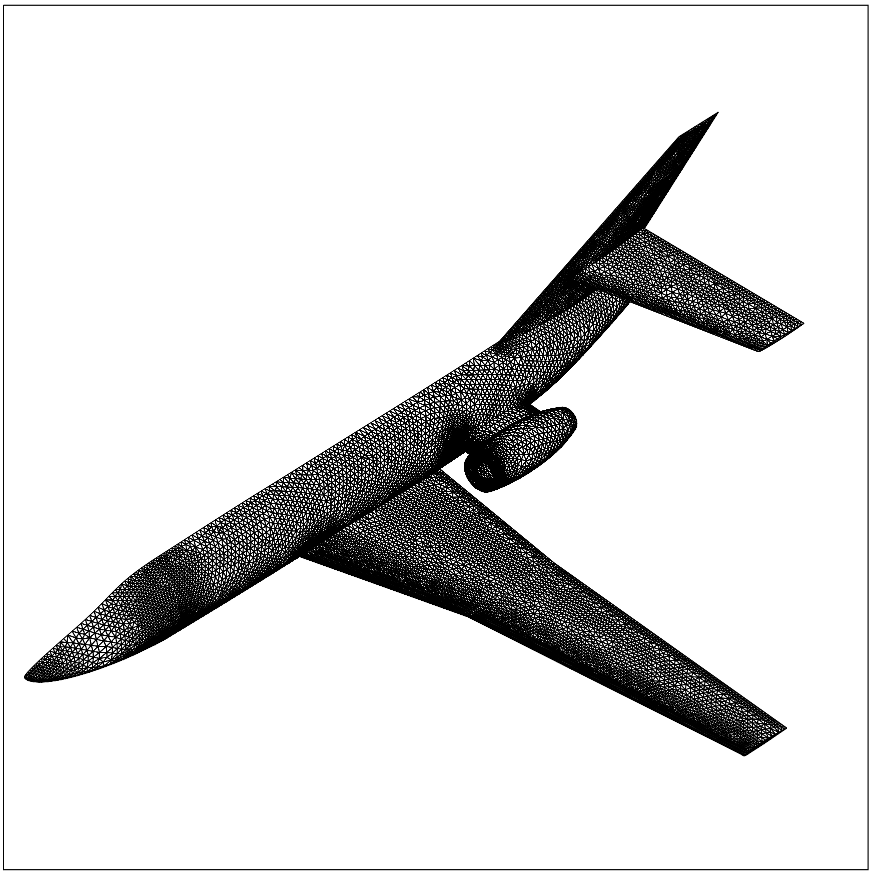


Figure 5: Initial surface grid for aircraft wing design [12]

5.2 Aircraft wing design

A recent paper by Elliot and Peraire [12] shows the state-of-the-art in the use of adjoint equations for design optimisation on unstructured grids. The main application considered is the wing optimisation for a business jet. The surface grid for the baseline configuration is shown in Figure 5.

Simple algebraic functions are used to define 6 design perturbation modes for the wing surface; care was taken to ensure compatible perturbations to grid points on the fuselage. The authors comment that a more ideal solution would involve coupling to a CAD/CAM system, as described earlier in these lecture notes. A linearised version of the method of springs is used to create the grid deformations in the interior.

The optimisation uses the discrete adjoint method, based on a linearisation of the discrete flow equations. Multigrid and parallel computing, using MPI or PVM on machines such the IBM SP2 or CRAY-T3D, are employed to reduce the execution time.

The objective function is the mean-square deviation from a target pressure

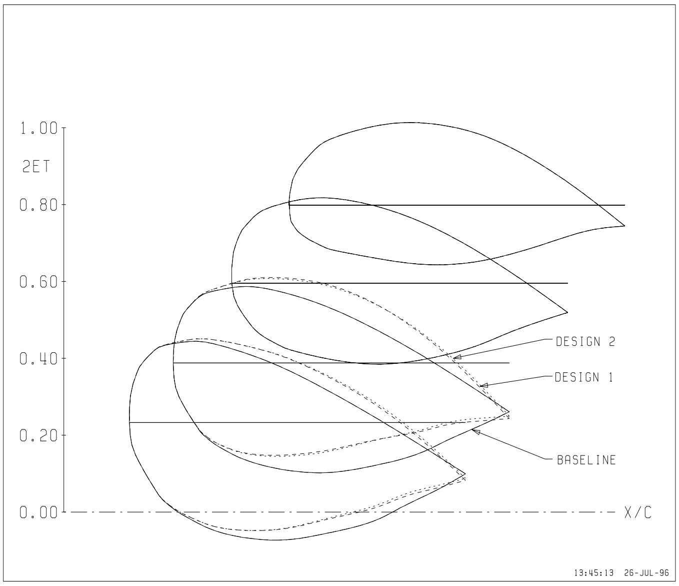


Figure 6: Evolution of the wing geometry during design [12]

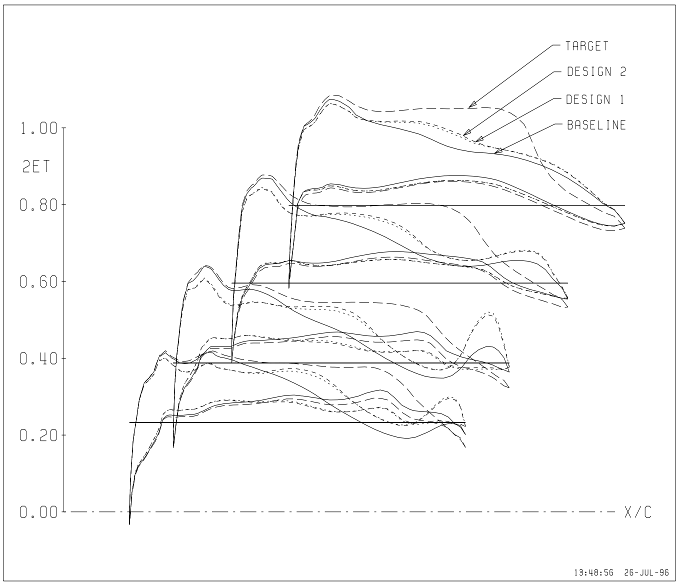


Figure 7: Evolution of the pressure distribution on the wing [12]

distribution corresponding to a 'clean wing' in the absence of the rear-mounted engine nacelle and pylon. Two design iterations are taken, decreasing the objective function by 75%. Figures 6 and 7 show the evolution of the wing geometry and pressure distributions, respectively.

6 Conclusions

The optimisation of complex aeronautical geometries using unstructured grid CFD methods is an active area of research. Most of the references in this paper are from the last 5 years, and it can be expected that there will be significant new developments in the next 5 years.

Increased computational power will make design optimisation using the Navier-Stokes equations feasible. A coupling of the design system to solid geometry CAD/CAM systems seems essential for industrial use; although there is no theoretical difficulty in accomplishing this, it may involve a substantial amount of software engineering.

Thereafter, the challenge will be to tackle increasingly difficult design applications, involving many more design variables and constraints. Only then will it be possible to properly assess the advantages and disadvantages of the different optimisation approaches.

References

[1] W.K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. AIAA Paper 97-0643, 1997.

[2] J. Batina. Implicit flux split Euler scheme for unsteady aerodynamic analysis involving unstructured dynamic meshes. *AIAA J.*, 29(11):1836–1843, 1991.

[3] O. Baysal and M. Eleshaky. Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. *AIAA J.*, 30(3):718–725, 1992.

[4] O. Baysal and M.E. Eleshaky. Aerodynamic sensitivity analysis methods for the compressible Euler equations. *J. Fluids. Engrg.*, 113:681–688, 1991.

[5] S. Connell and M. Braaten. Semi-structured mesh generation for 3D Navier-Stokes calculations. AIAA Paper 95-1679, 1995.

[6] P.I. Crumpton and M.B. Giles. Aircraft computations using multigrid and an unstructured parallel library. AIAA Paper 95-0210, 1995.

[7] P.I. Crumpton and M.B. Giles. Implicit time accurate solutions on unstructured dynamic grids. AIAA Paper 95-1671, 1995.

[8] P.I. Crumpton and M.B. Giles. Multigrid aircraft computations using the OPlus parallel library. In A. Ecer, J. Periaux, N. Satofuka, and S. Taylor, editors, *Parallel Computational Fluid Dynamics. Implementations and Results Using Parallel Computers*, pages 339–346. North-Holland, 1996.

[9] R. Das, D.J. Mavriplis, J. Saltz, S. Gupta, and R. Ponnusamy. Design and implementation of a parallel unstructured Euler solver using software primitives. *AIAA J.*, 32(3):489–496, 1994.

[10] D. Doorly, J. Peiro, and J-P. Oesterle. Optimization of aerodynamic-structural design using parallel genetic algorithms. AIAA Paper 96-4027-CP, 1996. Proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

[11] J. Elliot and J. Peraire. Practical 3D aerodynamic design and optimization using unstructured grids. AIAA Paper 96-4122-CP, 1996. Proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

[12] J. Elliott and J. Peraire. Aerodynamic design using unstructured meshes. AIAA Paper 96-1941, 1996.

[13] K.C. Hall. A deforming grid variational principle and finite element method for computing unsteady small disturbance flows in cascades. AIAA Paper 92-0665, 1992.

[14] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3:233–260, 1988.

[15] A. Jameson. Optimum aerodynamic design using the control theory. *Comput. Fluid Dynam. Rev.*, pages 495–528, 1995.

[16] A. Jameson, N.A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier–Stokes equations. AIAA Paper 97-0101, 1997.

[17] A. Khawaj, H. McMorris, and Y. Kallinderis. Hybrid grids for viscous flows around complex 3-D geometries including multiple bodies. AIAA Paper 95-1685, 1995.

[18] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer-Verlag, 1971. Translated by S.K Mitter.

[19] J. Newman, A. Taylor, and G. Burgeen. An unstructured grid approach to sensitivity analysis and shape optimization using the Euler equations. AIAA Paper 95-1646, 1995.

[20] V. Parthasarathy and Y. Kallinderis. Directional viscous multigrid using adaptive prismatic meshes. *AIAA J.*, 33(1):69–78, 1995.

[21] J. Peraire, J. Peiro, and K. Morgan. A 3D finite element multigrid solver for the Euler equations. AIAA Paper 92-0449, 1992.

[22] J. Peraire, M. Vahdati, K. Morgan, and O.C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comput. Phys.*, 72:449–466, 1987.

[23] J. Reuther and A. Jameson. Control based airfoil design using the Euler equations. AIAA Paper 94-4272-CP, 1994.

[24] G.N. Shrinivas. *Three-dimensional design methods for turbomachinery applications.* PhD thesis, Oxford University Computing Laboratory, 1996.

[25] G.N. Shrinivas and M.B. Giles. Application of sensitivity analysis to the redesign of OGV's. In *Proceedings of the IMECE Conference*, 1995.

[26] G.N. Shrinivas and M.B. Giles. OGV tailoring to alleviate pylon-OGV-fan interaction. In *Proceedings of the IGTI Turbo Expo*, 1995. ASME paper 95-GT-198.

[27] S. Ta'asan, G. Kuruvila, and M.D. Salas. Aerodynamic design and optimization in one shot. AIAA Paper 92-0025, 1992.

[28] V. Venkatakrishnan and D. Mavriplis. Agglomeration multigrid for the three-dimensional Euler equations. AIAA Paper 94-0069, 1994.

[29] V. Venkatakrishnan and D. Mavriplis. Implicit method for the computation of unsteady flows on unstructured grids. AIAA Paper 95-1705, 1995.

[30] J.M. Verdon and J.R. Caspar. Development of a linear unsteady aerodynamic analysis for finite-deflection subsonic cascades. *AIAA J.*, 20:1259–1267, 1982.

[31] N. Weatherill, O. Hassan, M. Marchant, and D. Marcum. Grid adaptation using a distribution of sources applied to inviscid compressible flow simulation. *Internat. J. Numer. Methods Fluids*, 19:739–764, 1994.

[32] N. Weatherill, O. Hassan, and D. Marcum. Compressible flowfield solutions with unstructured grids generated by Delauney triangulation. *AIAA J.*, 33(7):1196–1204, 1995.

[33] N.P. Weatherill, O. Hassan, and D.L. Marcum. Calculation of steady compressible flowfields with the finite element method. AIAA Paper 93-0341, 1993.

[34] D.S. Whitehead. The calculation of steady and unsteady transonic flow in cascades. University of Cambridge, Department of Engineering, 1982. Report CUED/A-Turbo/TR 118.

[35] D.S. Whitehead. A finite element solution of unsteady two-dimensional flow in cascades. *Internat. J. Numer. Methods Fluids*, 10:13–34, 1990.