# Some thoughts on exploiting CFD for turbomachinery design

M. B. Giles
*Oxford University Computing Laboratory*
*Wolfson Building, Parks Road, Oxford OX1 3QD, UK*

In the last twenty years, CFD has evolved enormously and it is now used extensively by all turbomachinery companies in designing specific components. In the next twenty years, the emphasis will be on automating the design process, speeding up design iterations, considering more radical design changes, and maximising the benefits from multidisciplinary trade-offs.

This does not reduce the role of the designer; on the contrary, the aim should be to increase the designers' productivity by allowing them to more easily investigate the possibilities of new designs. In some limited areas, this might include black-box optimisation under the control of the designer who will specify the design space and verify the acceptability of the final design.

This paper addresses some of the possibilities and the issues which will need to be faced. It also presents some work on the use of direct sensitivity calculations for the optimisation of outlet guide vanes in a turbofan bypass duct.

(This paper was presented at the IMeche Symposium on Exploiting CFD for Turbomachinery Design, on March 19th, 1998.)

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

May, 1998

Table 1: Hierarchical definition of a turbine vane

| | |
|---|---|
| Level 1 | number of blades, hub/tip radius, throat area, inflow/outflow angles, mass flow |
| Level 2 | camber/thickness distribution, cooling mass flow |
| Level 3 | geometry of fillets at hub and tip junctions |
| Level 4 | details of film cooling holes and slots, temperature of inflow and cooling flow |
| Level 5 | alloy type and thermal properties |

# 1 Hierarchical design

Aeroengines and other large turbomachines are very complex engineering systems. Viewed as a single entity, there may be hundreds of thousands of components, and well over a million important design parameters. This is obviously far too large a number to be handled by a single designer. The computational cost of analysing the entire system in detail is also prohibitive, which immediately rules out any thought of global 'black-box' optimisation.

Even when considering an individual component within the engine, the complete electronic product definition (EPD) often contains a level of detail which is unnecessary and even undesirable for much engineering analysis. An example is the cooling holes in a high pressure turbine. The geometry of these must be contained in the EPD database for manufacturing purposes, but when computing the viscous flow in the blade passage and the resulting heat transfer to the blade it is usual to ignore the details of the cooling holes and simply model the coolant injection through a transpiration boundary condition.

The solution to these problems is to use a hierarchical representation in which every component is defined at a number of different levels of detail. Table 1 presents a number of levels for a high pressure turbine vane; the level 1 representation is the most basic, with higher levels adding successively more detail. An engineering analysis tool will interface to whichever level of the EPD is most appropriate. In the case of the turbine vane, a CFD code might interface at level 2, treating the blade/hub and blade/tip junction as sharp corners, and modelling the film cooling as a distributed mass source. A stress analysis package would need to interface at level 3 or higher since the fillet geometry is needed to calculate the correct stresses in the corners.

To avoid the huge computational cost of analysing the entire engine requires a hierarchical approach to the design process as well. The design of all aeroengines is carried out at two levels, preliminary design and detailed component design. The preliminary design group considers the engine as an entire system, thinking about the customers' requirements, sizing the major components, deciding which subsystems to retain from previous products, and aiming to maximise profit over the lifetime of the entire project. When trying to optimise the overall configuration during preliminary design, the system is modelled very approximately using a considerable amount of empiricism based on past experience. This approximate modelling, working with the lowest levels in the hierarchical EPD database with a limited number of fundamental design parameters means that

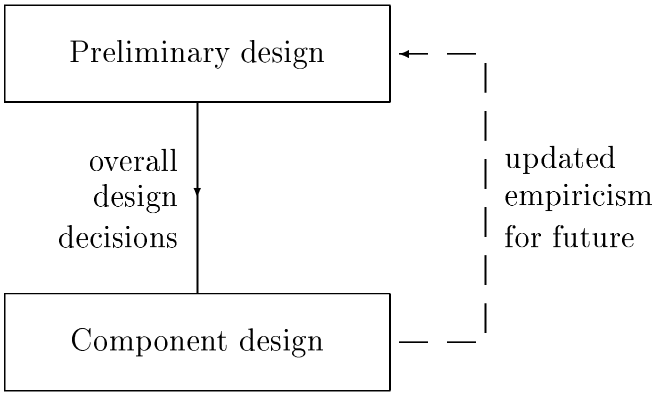



Figure 1: Current sequential two-level design process

the cost of simulating the entire system is reduced to a few minutes at most. This allows a thorough investigation of the global trade-offs influencing the overall system configuration, and makes it feasible to use robust black-box optimisation methods such as genetic algorithms which are well-suited to global optimisation and integer design parameters.

At the conclusion of the preliminary design process, many crucial design decisions have been made, such as engine thrust, mass flow and fan radius. The second level of the design hierarchy is the design of individual components within each subsystem, such as the HP turbine. The design intent for each component has been fairly tightly specified in preliminary design, and many constraints have been imposed. The task of the component design team is to fulfil the design intent as well as possible (good aerodynamic performance, good structural integrity, low weight, etc.) subject to the constraints. To a large extent, this is a matter of shape optimisation, the non-geometric design parameters having been set in preliminary design.

As described above, and illustrated in Figure 1, the current hierarchical design approach is sequential, preliminary design followed by component design. Except in exceptional circumstances, the decisions made in preliminary design are not changed during component design. This is due to preliminary design being firmly based on empiricism from past experience, so major surprises are unlikely to arise during the component design process.

There are two drawbacks to this sequential design process. The first is that its success depends on the new design not being too different from past designs, so that the empiricism in the modelling remains valid. This makes it very difficult to develop radically new designs. The second drawback is that the empiricism in the preliminary design system represents the collective experience of past projects, but no two projects are ever identical. Even if the customer requirements are identical, technological advances mean that the best engine or aircraft of today would be different from that designed twenty years ago. To some extent this technological progress can be accounted for in the empiricism, but inevitably preliminary design is based on only an approximate model of the system.

In the future, there may be a shift to a more tightly-coupled two-level design system, as illustrated in Figure 2. The overall system design will begin, as now, with a preliminary design based on past empiricism. This will provide the starting point for the detailed

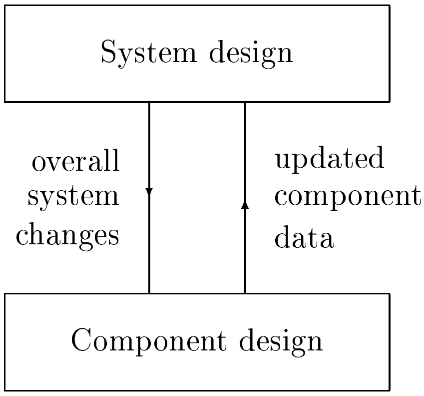



Figure 2: Future tightly-coupled two-level design process

component design. The change from the current sequential design process is that at this point data will be fed back into the overall system design, updating its empiricism based on the results of the detailed engineering analyses performed during the component design. This will allow further refinement of the overall system performance by fine-tuning some of the global trade-offs. Ideally, this design cycle would be repeated a number of times, with the component design responsible for the shape optimisation of specific components from a 'local' viewpoint, while the system design is concerned with overall optimisation of the major sub-systems based on a global viewpoint.

The main reason a tightly coupled design system is not used today is time. The design time for an engine or aircraft project is strictly limited. There are very strong commercial pressures to bring a product to market as quickly as possible, even if this involves sacrificing a certain amount of performance because of the lack of time to investigate all design options. Spending more time on refining a design also has manpower and experimental testing costs; these have to be weighed against the possible benefits to be gained. The key to the successful adoption of a tightly coupled design system in the future lies in software engineering and ever increasing computational power. Good software engineering will minimise the time spent by designers in the coupling between system design and component design. The continuing doubling of computational power every 18-24 months will ensure decreasing execution times for the more expensive analysis tools, allowing more cycles of the coupled design process to be completed within a given time.

# 2 Parametric CAD systems

The CAD system lies at the heart of engineering design. In the past it has been common for different disciplines such as aerodynamics and structural analysis to have different representations of components such as blades. Thus, the aerodynamicists might perfect their design with one representation, perhaps using a number of sectional profiles, and then the geometry would be mapped onto another representation for structural analysis. To some extent, this can be viewed as different levels in the hierarchical description discussed in the last section. However, it is a cleaner solution for these different representations to co-exist within a single CAD system. Only then can one easily perform multi-disciplinary analyses (such as fan untwist due to aerodynamic loads) and design tradeoffs.

Thus, one requirement for a CAD system is that it should support a hierarchical definition of all components and sub-systems. The second requirement is that this should be a parametric definition, in which one can easily change any geometric design parameter to obtain a new geometry.

This may seem any easy task, but it is not. Consider a turbine rotor as an example. For simplicity, we will neglect the important details of internal cooling and the fir-tree root. Some CAD systems would define the rotor in terms of its surface geometry, dividing the surface into a number of NURB spline patches which would cover the blade itself, the hub annulus, and the fillets at the junction between the blade and hub. If the designer wishes to change the camber of the rotor, how would the CAD representation change? Manual adjustment of the NURB surfaces would be incredibly laborious, especially in handling the details such as the fillets.

Most current CAD systems solve this problem by using at their core a solids modelling package (e.g. ParaSolids, Pro Engineer [4, 12]) which defines each object as a composite built from simpler solids using rules of union, intersection and exclusion. For example, the turbine rotor would be defined as a union of an axisymmetric solid, whose surface is the hub annulus, and an extended blade object whose definition would extend well within the hub annulus. The union operation would automatically compute the line of intersection between the rotor and the hub, and would create and add the fillets of the desired internal radius. As output, the CAD system may supply the same NURB surface definition as older CAD systems, but the key strength of a solids-based CAD system is the ability to vary design parameters easily. In the case of the turbine rotor, a change in the camber would produce a change in the blade object; the CAD system would then re-apply the union operation creating a new line of intersection between the rotor and the hub and therefore new fillets.

The final step in building this aspect of a design system is to couple the parametric CAD system to the grid generators needed for engineering analysis. For example, when performing an aerodynamic design of the rotor, a parametric investigation of the consequences of camber variations would require the generation of a sequence of CFD grids corresponding to different camber values. Although a baseline grid may be generated manually to ensure it is of good quality, to speed the design process it is essential that the other grids are generated automatically.

One option is to run the standard grid generator to create a new grid from scratch each time, using relevant grid spacing information from the baseline grid. Alternatively, the grid generator could determine from the CAD system the perturbations to the surfaces and lines of intersection, and use these to perturb the surface grid points of the baseline grid. Having done so, the grid generator can then perturb the interior grid points to produce a valid perturbed grid with the same topology as the original grid. This second approach is preferable in many cases since it produces a continuous perturbation to the flow field computed by the CFD code, which can then be differenced to obtain the flow-field sensitivity to the design change.

# 3 Optimisation and the role of the designer

At the outset, the designer must specify the design space, the parameters which are to be varied with the objective of improving the design. Since the computational cost of direct sensitivity methods is proportional to the number of design parameters, it is very important that the designer uses his expert judgement to limit the number of design parameters to those which are most important. The designer must also specify the design constraints. Some of these will be inequality constraints (e.g. minimum blade thickness, etc.) while some will be equalities (e.g. specified pressure ratio for a compressor).

There are then three possible design scenarios. In the first, the designer is able to define a single scalar function $I(\boldsymbol{U}, \boldsymbol{\alpha})$ (known as the *objective* function) to be optimised subject to all of the constraints. In general, the objective function depends on both the design variables $\boldsymbol{\alpha}$ and the flow field $\boldsymbol{U}$, which in turn also depends on $\boldsymbol{\alpha}$. In preliminary design this objective function may be overall fuel efficiency, or even the financial return on investment of the operating airline [9]. The computational design system will then attempt to find the optimal solution to the problem, subject to the constraints, using the most appropriate optimisation technique. For preliminary design, this may involve the use of genetic algorithms which are very good at finding the global optimum amongst many local optima, and in treating integer design parameters.

In component design, the definition of a suitable objective function can be trickier. Ideally, the designer might wish to minimise the loss in a compressor. However, due to limitations in turbulence and transition modelling, the time-averaged treatment of unsteadiness such as vortex shedding and wake/rotor interaction, and numerical effects due to grids which do not yet fully resolve all features in three-dimensional flows, CFD methods are often not able to predict loss with sufficient accuracy for the purposes of design optimisation. Therefore, it is more common for the designer to choose an alternative objective function, such as the deviation from a target pressure distribution, which if optimised will lead to an improved design with lower loss. This relies on the designer's ability to specify a target pressure distribution which will lead to low boundary layer growth and/or reduce the secondary flow. Thus, the designer plays a critical role in formulating a well-behaved objective function which can be reliably optimised by the available computational analysis tools.

During the optimisation process, the designer's task is to monitor the evolution of the design parameters, making sure that the design remains sensible. This may prove to be a much harder task than it appears. Aerospace design is very multidisciplinary and highly constrained. Initially, one might ignore a large number of inequality constraints, believing them to be unimportant because they will not be active in the final design, and wishing to minimise the computational cost of each step in the design process. One may even forget a large number of 'obvious' constraints (such as minimum blade thicknesses). The designer must therefore watch the evolution of the design to see if new constraints should be added [16]. In component design, it may also be necessary to examine the detailed results from the engineering analyses to ensure that the design does not produce flow fields or other features which violate basic modelling assumptions inherent in the analyses. For example, the use of potential flow modelling would no longer be appropriate if the design led to the presence of strong shocks or a separated boundary layer.

In the second design scenario, it is appropriate to work with more than one objective

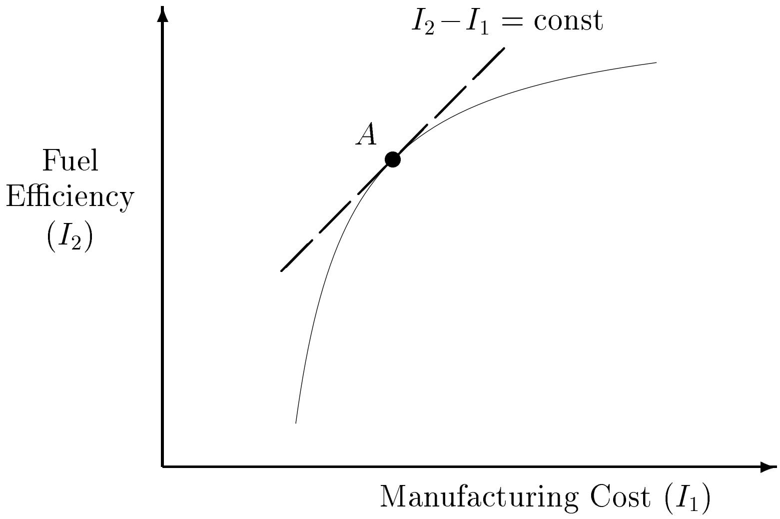



Figure 3: An example of a tradeoff between two different objective functions

function. For example, the thickness of a compressor blade is a tradeoff between aerodynamic performance, which decreases with increasing thickness, and structural integrity which improves with increasing thickness. Rather than fixing one and optimising the other, the designer may prefer to study the tradeoff between the two before making a judgement about the best compromise. This need to assess multidisciplinary tradeoffs is emphasised in Reference [16] in the context of aircraft design. If the relative importance of the two objective functions is known beforehand, then a single composite objective function of the form $I_2 + \lambda I_1$ can be created. Referring to Figure 3, optimising $I_2 - I_1$ corresponds to finding the point A on the curve which has the maximum value of $I_2 - I_1$, and for which there is a tangent line of the form $I_2 - I_1 = \text{const}$. The drawback of this approach is that the designer may not have a good idea of the appropriate value of $\lambda$, and optimising in this fashion would give no information about how the optimum would change if the value of $\lambda$ were changed.

In the first two design scenarios, the engineering design system was responsible for some, or all, of the optimisation of the design, with the designer monitoring the design evolution in the first, and making some critical design decisions in the second. In the third approach, the designer performs the optimisation, with the design system supplying the designer with sensitivity information about the consequences of design changes. This assumes that there is an existing design and the objective is to improve upon it. The designer specifies the active design parameters and the constraint functions and objective functions he considers important. The design system returns the sensitivity of each of the functions to changes in each of the parameters, and invites the designer to decide upon suitable parameter changes. The design system may also aid the designer by ensuring that the changes are compatible with the constraints in the problem.

This approach gives the designer the greatest flexibility, allowing the designer to take into account other factors and constraints which may be hard to specify in a computerised design system [16]. In particular, during the development of an integrated design system, when not all of the analysis modules have been developed or integrated into the system, this third approach may be the only feasible option.

Sensitivity analysis is a crucial component of the tightly coupled two-level design system described earlier. It is unlikely that at the component level one would simultaneously optimise the size and shape of different blade rows in an engine. However, sensitivity analysis at the component level could determine, for example, the change in the aerodynamic efficiency due to a change in chord length. With this information, the overall system level design could consider tradeoffs, increasing the chord of one blade row while simultaneously decreasing the chord of another to retain a fixed overall engine size.

# 4 Sensitivity analysis

In nonlinear sensitivity analysis, one obtains approximate linear sensitivities by simple finite differencing of the solutions from a number of nonlinear computations [19, 20, 21]. For each set of design parameters $\boldsymbol{\alpha}$, the discrete flow equations

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

can be solved to implicitly obtain $\boldsymbol{U}$ as a function of $\boldsymbol{\alpha}$. Using simple one-sided differencing, we can define the approximate sensitivity of the flow solution to variations in the $k^{th}$ design parameter as

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

where $\boldsymbol{e}_k$ is a unit vector in the $k^{th}$ direction and $\epsilon_k$ is an appropriately small perturbation [10, 11].

The main advantage of the nonlinear sensitivity approach is its simplicity. There are no major new analysis codes to be written, just a small amount of programming to evaluate the objective and constraint functions. With the appropriate design software to manage the construction of the approximate sensitivities it is then possible to assemble the analysis codes into a design system very rapidly.

Once the individual sensitivities have been computed, the linear response of the flow field to an arbitrary set of design variable perturbations is given by

$$\boldsymbol{U}(\boldsymbol{\alpha}+\boldsymbol{\epsilon}) \approx \boldsymbol{U}(\boldsymbol{\alpha}) + \sum_k \epsilon_k \frac{d\boldsymbol{U}}{d\alpha_k}.$$

It is then possible for a designer to interactively vary the values of $\epsilon_k$ to examine the consequences of changes in each design variable, and choose an optimum combination.

The direct sensitivity approach also has a big advantage when the objective function comes from a least-squares minimisation problem. In this case, the linear approximation to the perturbed flow field leads naturally to a quadratic approximation for the objective function, which can be minimised analytically [10].

The main disadvantage of the nonlinear approach is its cost when the number of design parameters is large. This is why it is important that the designer exercises good judgement in limiting the number of active design parameters. However, there is also the possibility of computing the different sensitivities in parallel, performing the large number of nonlinear flow computations overnight on a large group of workstations.

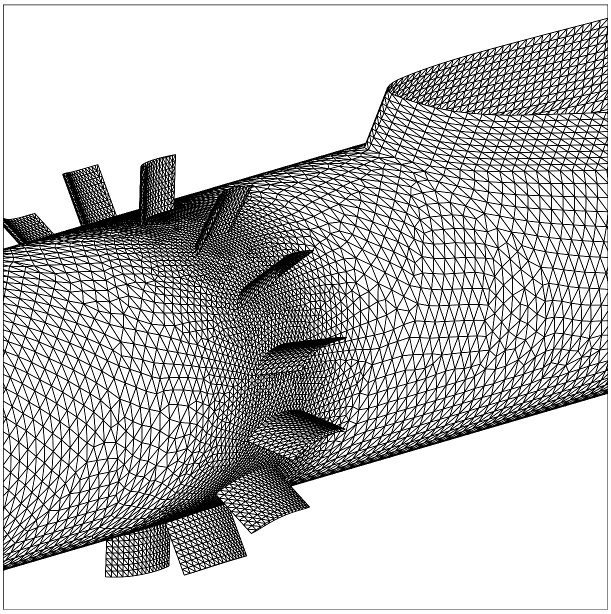


Figure 4: surface of tetrahedral grid for OGV design (outer annulus not plotted)

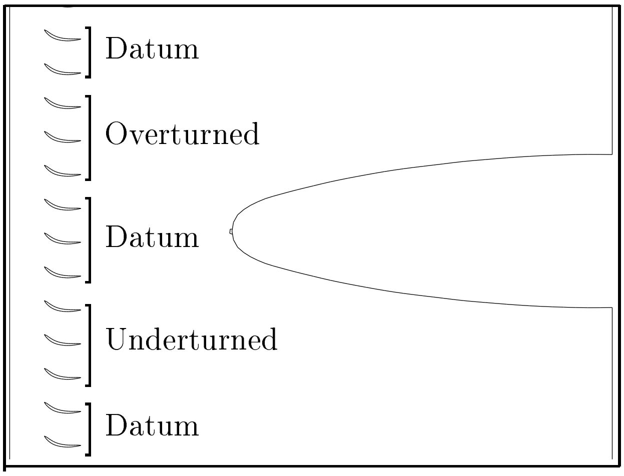
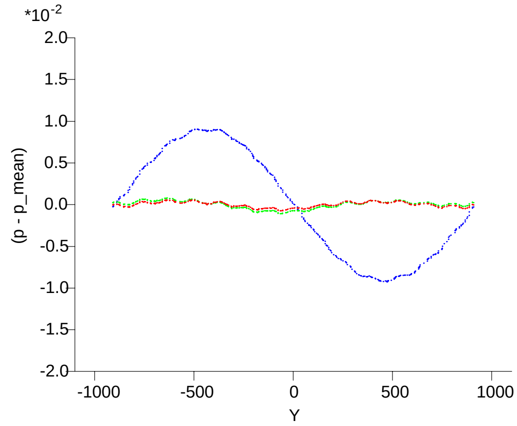



Figure 5: grouping of OGVs for 3 blade design

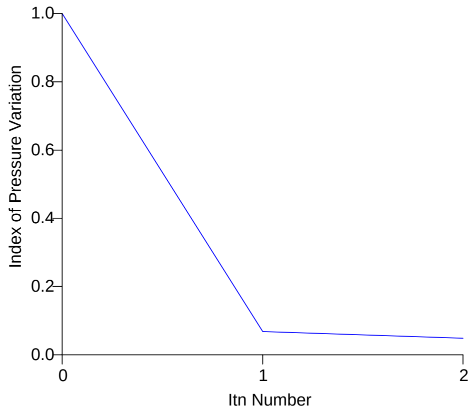
## 4.1 Outlet guide vane optimisation

Shrinivas has used nonlinear approximate sensitivities for a 3D design application concerning the bypass duct of a turbofan aeroengine [19]; this is an extension of earlier research by Shrinivas and Giles using 2D modelling [20, 21].

Figure 4 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 5 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) $Cp$ at mid radius




b) convergence of $I$

Figure 6: Optimisation using sinusoidal camber variation

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 OGVs 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.

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

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

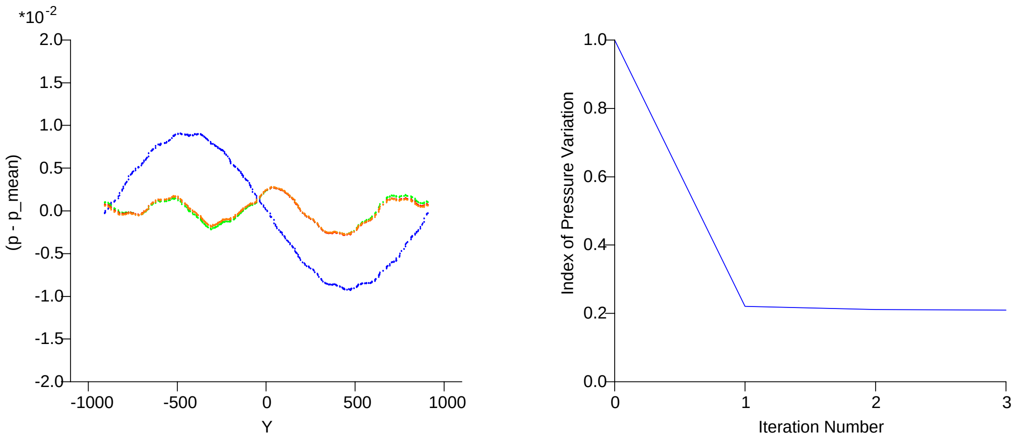
The inviscid flow code that was used in this work was developed by Crumpton [5]. It uses an edge-based discretisation of the Euler equations and a standard Runge-Kutta time-marching algorithm. Edge-collapsing is used to generate the coarser grids for the multigrid algorithm. The execution speed is further improved through parallel execution on distributed-memory machines such as the IBM SP2 using the OPlus parallel library [6].

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.




a) $C_p$ at mid radius b) convergence of $I$

Figure 7: Optimisation using 3 blade types

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 6 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 5 shows the chosen grouping of these blades. There are still just two 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 7 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.

# 5 Adjoint sensitivity analysis

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

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

This can be solved, directly or iteratively, to obtain $\widetilde{\boldsymbol{U}}_k$ for each design parameter. The total derivative of an objective function with respect to the $k^{th}$ design parameter is then given by

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

In CFD applications, the cost of solving the linear system of equations is comparable to the cost of solving the nonlinear system, so there are no computational savings from using the direct linear sensitivity approach. However, this is the starting point for the discrete version of the inverse (adjoint) sensitivity analysis.

Eliminating $\widetilde{\boldsymbol{U}}$ gives [7, 8]

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

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 adjoint 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 main 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 would be little to be gained from the adjoint approach.

A second weakness of the adjoint approach is that there is no simple way in which to compute the Hessian matrix $\partial^2 I / \partial \alpha_i \partial \alpha_j$ even when the objective function comes from a least-squares minimisation problem. Instead, the gradient-based optimisation methods must construct an approximation to the Hessian matrix using information about the variation in the gradient at different points in the design space. In addition, such methods usually determine a search direction and then find the optimum along this direction using a line search algorithm. Both of these aspects result in more steps in the optimisation procedure than are required when for the direct sensitivity approach using its approximate Hessian.

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 [17]. This can then be discretised and solved numerically [1, 2, 3, 13, 14, 15, 18, 22].

This is a very active research in the aeronautical research community, particularly in the US. Research on its application to turbomachinery design is only now beginning.

# 6 Conclusions

This paper has put forward the following four ideas:

- a hierarchical approach to turbomachinery design will remain essential, and there may be considerable scope for improvement through a tighter coupling between preliminary design and detailed component design;
- the underlying CAD system needs to support a hierarchical representation of the engine components, and be based on parametric solids to facilitate parametric design;
- approximate nonlinear sensitivity analysis is a straightforward approach to building a design system which can be steered interactively by a designer and coupled to an optimisation procedure;
- looking to the future, adjoint analysis provides a computationally efficient way of determining sensitivities when there are many design parameters.

Further discussion of these ideas and other design issues, in particular the use of unstructured grid methods, is contained in References [10, 11].

# 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] O. Baysal and M. Eleshaky. Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. *Journal of the American Institute on Aeronautics and Astronautics*, 30(3):718–725, 1992.

[3] O. Baysal and M.E. Eleshaky. Aerodynamic sensitivity analysis methods for the compressible Euler equations. *Journal of Fluids Engineering*, 113:681–688, 1991.

[4] S. Chen and D. Tortorelli. Three-dimensional shape optimization with variational geometry. AIAA Paper 96-3992-CP, 1996. Proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

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

[6] 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.

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

[8] J. Elliott 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.

[9] M. Ewing and K. Downs. Conceptual aircraft design with genetic search based on financial return on investment. AIAA Paper 96-4106-CP, 1996. Proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

[10] M.B. Giles. Adjoint equations in CFD: duality, boundary conditions and solution behaviour. AIAA Paper 97-1850, 1997.

[11] M.B. Giles. Aerodynamic design optimisation for complex geometries using unstructured grids (lecture notes for VKI lecture course on inverse design). Technical Report NA97/08, Oxford University Computing Laboratory, 1997.

[12] E. Hardee, K-H. Chang, K. Choi, X. Yu, and I. Grindeanu. A CAD-based design sensitivity analysis and optimization for structural shape design applications. AIAA Paper 96-3990-CP, 1996. Proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.

[13] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[14] A. Jameson. Optimum aerodynamic design using the control theory. pages 495–528, 1995.

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

[16] W. Jou, W. Huffmann, D. Young, R. Melvin, M. Bieterman, C. Hilmes, and F. Johnson. Practical considerations in aerodynamic design optimization. AIAA Paper 95-1730, 1995.

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

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

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

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

[21] 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.

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