

# Design search and optimization in aerospace engineering

BY A. J. KEANE\* AND J. P. SCANLAN

*University of Southampton, Highfield, Southampton SO17 1BJ, UK*

In this paper, we take a design-led perspective on the use of computational tools in the aerospace sector. We briefly review the current state-of-the-art in design search and optimization (DSO) as applied to problems from aerospace engineering, focusing on those problems that make heavy use of computational fluid dynamics (CFD). This ranges over issues of representation, optimization problem formulation and computational modelling. We then follow this with a multi-objective, multi-disciplinary example of DSO applied to civil aircraft wing design, an area where this kind of approach is becoming essential for companies to maintain their competitive edge.

Our example considers the structure and weight of a transonic civil transport wing, its aerodynamic performance at cruise speed and its manufacturing costs. The goals are low drag and cost while holding weight and structural performance at acceptable levels. The constraints and performance metrics are modelled by a linked series of analysis codes, the most expensive of which is a CFD analysis of the aerodynamics using an Euler code with coupled boundary layer model. Structural strength and weight are assessed using semi-empirical schemes based on typical airframe company practice. Costing is carried out using a newly developed generative approach based on a hierarchical decomposition of the key structural elements of a typical machined and bolted wing-box assembly.

To carry out the DSO process in the face of multiple competing goals, a recently developed multi-objective probability of improvement formulation is invoked along with stochastic process response surface models (Krigs). This approach both mitigates the significant run times involved in CFD computation and also provides an elegant way of balancing competing goals while still allowing the deployment of the whole range of single objective optimizers commonly available to design teams.

---

**Keywords:** multi-objective; Pareto; improvement; cost; drag; Krig

---

## 1. Introduction: the design process, design search and optimization

Design search and optimization (DSO) is the term used to describe the application of formal optimization software to the problem of engineering design. The inclusion of the word ‘search’ in this term indicates that alongside the desire for optimal designs is the recognition that the design activity is often an exploratory process where there are no fixed endpoints and no obviously optimal solutions to the complex problems being dealt with. Rather, an aerospace design

---

\* Author for correspondence (andy.keane@soton.ac.uk).

One contribution of 9 to a Theme Issue ‘Computational fluid dynamics in aerospace engineering’.

team is typically faced with multiple and competing goals, thousands of constraints and imperfect design tools that are almost never neatly integrated with each other. Moreover, the goals and constraints being dealt with will often not be defined with crisp precision, but instead need interpreting with a great deal of hard won experience. Any of the well-known texts on aircraft synthesis will make such difficulties abundantly clear; see, for example, those by [Stinton \(1983\)](#) or [Torenbeek \(1984\)](#).

Much progress has, of course, been made in the field of aerospace design since the first flight by the Wright brothers a little over a century ago. A great deal of this progress has been achieved by the use of computational tools, primarily in the prediction of the performance of new configurations before they have been built. The two fields of computational structural mechanics (CSM) and computational fluid dynamics (CFD) have been at the forefront of this progress (and this theme issue is dedicated to the latter area). In fact, so important has the use of computational modelling become in engineering design and analysis that a new field, which has been labelled ‘computational engineering’, has begun to emerge ([Keane & Nair 2005](#)). This field spans more than just the modelling of structures or fluid flows: it also tackles the way that high-quality analysis tools can be integrated into the design process, allowing for geometry manipulation and meshing, the application of optimization tools, the interpretation of results coming from varying domains and at differing levels of fidelity and the best use of the (often distributed) computing resources available to the design team. The use of knowledge stores and knowledge application tools is also beginning to impact on this world, so that the large quantities of corporate information accumulated by the world’s aerospace companies can be effectively captured and reused during design.

The current state-of-the-art in aerospace computational design typically makes use of<sup>1</sup>

- a geometry manipulation engine that may be process specific or make use of one of the sophisticated parametric CAD engines now available (e.g. CATIA, UNIGRAPHICS, PROENGINEER);
- a meshing process that accepts data from the geometry system (using standards such as STEP or IGES, or by direct file transfer) and which produces discretized analysis domains suitable for use by CSM or CFD codes (e.g. GRIDGEN, GAMBIT);
- a collection of analysis codes that work at varying levels of fidelity and deal with various models of the physics encountered (in CFD, these range from simple two-dimensional panel codes, through Euler and RANS solvers up to direct Navier–Stokes codes applied in three dimensions; e.g. VGK, VSAERO, FLUENT);
- post-processing tools that allow useful engineering metrics to be extracted from the results of analysis, perhaps combining results coming from multiple disciplines as in aero-elasticity (often highly customized and calibrated by the design team, rarely publicly available);

<sup>1</sup>N.B., here we mention a number of commercial codes by name, all of which are subject to copyright, etc., and all of which may readily be located from a web search—we make no recommendation as to the capabilities of any of them and of course there are many others besides those noted—they do, however, typify current practice in this area.

- exploration and optimization tools that can be used to modify designs formally in the search for improvement against numerically specified targets, either commercially available such as iSIGHT or DESIGNEXPLORER or in-house such as OPTIONS (Keane 2004);
- process integration tools that allow the preceding components to be linked together and run on heterogeneous and distributed computing platforms, taking into account the very large variations in run times that will be encountered and the team-like nature of those working on the design (e.g. FIPER, MODELCENTER, MATLAB or PYTHON); and
- database tools that allow the collected results of design studies to be stored and efficiently browsed at will (such as ORACLE).

Each of these tool types will require significant expertise to use to the standards now expected in aerospace companies. They will thus all need augmenting by large amounts of knowledge—currently, this mostly resides in the heads of the design team, although progress is being made in knowledge-based technologies across all these topics, whether it be in the construction of suitable meshes for accurate CFD solves or the correct choice of optimization strategy to allow for reasonable outcomes in affordable computing time. The linking together of these components into an automated design system to tackle a specific design challenge, preferably spanning multiple domains, is still very difficult. Although much progress has been made, almost all of the steps noted will require manual intervention, often for substantial periods. For example, the production of a high-quality mesh with good boundary layer modelling over a complete aircraft configuration remains a daunting task if accurate drag results are to be recovered—most aerospace companies still insist on full wind tunnel validation of any critical computations.

If the design process envisaged is to allow for a wide-ranging exploration over many possible configurations, the integration of all the relevant modules becomes even more difficult. It is therefore common practice to build the tool chain up for each major configuration as and when it is needed, reusing previous components where possible and customizing where necessary. This ‘workflow’ construction process can be dealt with in many ways, but typically feels a lot like computer programming whatever system is used: sequences of processes must be assembled, looped over and tested as the design progresses. Moreover, the workflow itself will be adapted as work proceeds and design options become more clearly understood. For example, the baseline geometric model being worked on will be modified in different ways at different points in the design. This may even involve the transition through completely different ways of holding the geometric data from bespoke concept exploration tools that feel much like spreadsheets to full product life-cycle modelling (PLM) systems in which even parametric CAD is just a sub-activity. It is also not uncommon to hold several versions of the same geometry, each specifically tailored to the process in hand: the level of detail needed in a manufacturing drawing is clearly unnecessary for CFD; the mesh used for CSM will differ from that used in CFD; even within CFD, differing resolutions will typically be used for drag recovery work when compared with aero-elastic studies.

A number of further factors combine to make work in this field particularly demanding. First and foremost, the run times needed for the most accurate studies in the most complex regimes can be fearsome. The study of a bird strike on a running aero-engine using large deflection finite element-based CSM codes

can take many days on a major parallel computer. A direct Navier–Stokes solve at even modest Reynolds numbers is similarly difficult. Second, design teams will wish to study multiple options, either to trade various desirable and competing goals against each other, or just to allow for a simple design optimization study. Lastly, no engineering product operates in its nominal geometry, under nominal conditions, unchanged throughout its life. Allowance must be made for uncertainty in shape and boundary conditions when carrying out analysis and the consequent stochastic analysis is always more time consuming than a simple deterministic calculation. All these issues point in essentially one direction— attempts must be made to mitigate analysis run time. Sometimes, this is possible through improved solution schemes, sometimes via the use of more powerful processors and sometimes through the deployment of large-scale computing clusters, although such parallel computation brings its own difficulties. Aerospace companies seek to exploit all these avenues.

In this paper, we will concentrate on software methods that enable multiple goals to be studied in an efficient way. We will combine classical drag reduction calculations with modern cost evaluation methods and show how these competing issues can be reconciled via computational models. Before treating specific examples, however, we briefly review a number of the approaches commonly adopted when applying optimization tools to CFD calculations. We then go on to use the Southampton multi-level wing design environment (Keane & Petruzzelli 2000) to study the merits of multi-objective, multi-disciplinary design when applied to three-dimensional CFD solvers over a transonic wing system combined with a cost modelling tool. Here, the aim is to build a multi-objective response surface model (RSM) using CFD data to model variations in drag at fixed lift and a new generative code to describe manufacturing costs as gross changes are made to the overall wing parameters. Currently, such changes are usually assessed using empirical concept design tools that make no attempt to solve the flow conditions over the wings being studied and that use past data to estimate probable costs. Although these concept tools can be extremely accurately calibrated to deal with familiar geometries and processes, they experience difficulties whenever extreme or even moderately novel configurations are considered. They are, however, very easy to use and capable of giving rapid estimates of probable drag levels (Cousin & Metcalfe 1990) and probable costs.

The work reported here attempts to overcome some of these limitations by fusing together data coming from CFD-based drag routines and the costing code using design of experiment (DoE) techniques (Mackay *et al.* 1979; Mead 1988) and multi-objective Kriging (Jones *et al.* 1998) to build RSMs (Myers & Montgomery 1995). Variants on these methods have been used in aerospace design for some time. However, so far, they have mostly been used to accelerate single domain, single goal optimization approaches using expensive codes (Keane 2003). It is only relatively recently it has been proposed that they might be helpful in multi-objective analysis (Hutchinson *et al.* 1994; Liu *et al.* 1999; Malone *et al.* 1999; Vitali *et al.* 1999; Zang & Green 1999). The main aim here is to use the DoE and Krigs to produce a combined RSM that combines multiple goals in a new way.

The techniques described here have recently been incorporated within the OPTIONS design exploration system (OPTIONS 2006) and the results presented have been produced using that system to drive the Southampton wing design environment (Keane & Petruzzelli 2000).

## 2. CFD-based optimization

For as long as people have been able to predict the behaviour of aerospace systems via modelling or computation, there have been attempts to use formal schemes to improve their performance. In the world of aerospace CFD, this has often focused on drag reduction or the improvement of lift/drag ratios in some way. In all cases, this has involved trying to specify the geometry in some codified or formalized way that will allow variations in shape to be systematically generated in preparation for performance assessment. To begin with, attempts were made to explore the behaviour of two-dimensional airfoil sections systematically using wind tunnels—leading, for example, to such things as the famous NACA four digit airfoil series (Ladson & Brooks 1975). The first computer-based applications of optimization in design appear to be for structural applications; see, for example, the early work of Schmidt (1960) on structural synthesis. One of the earliest studies to make use of computational approaches to airfoil design was that by Liebeck & Ormsbeef (1970) some 10 years later, using second-order airfoil theory and the calculus of variations. This early work has been followed by a veritable avalanche of publications that have attempted to use ever higher-quality CFD solvers over ever wider ranges of geometry operated over broader ranges of operating conditions.

### (a) *Parameterization*

In each case, a mapping is required between the (preferably few) variables controlled by the designer and the (commonly many tens of thousands of) surface coordinates needed to define the boundaries in the flow. Building such a mapping or *parameterization* of the geometry can be critical in achieving an efficient design search—if there are too many variables or these interact too strongly, the search landscape can become intractably complex to explore. Conversely, if too few variables or too restricted a mapping is used, it may be impossible to make worthwhile improvements in a design.

For purely local searches, there are some strategies that can overcome this dilemma by allowing an efficient search using all the coordinates of the surface directly. These make use of adjoint schemes (Jameson 1988) that allow one to trade the number of variables being changed against the number of outputs being studied. In a conventional search, one varies a few parameters and then analysis gives flow details at possibly millions of locations in the flow field. In adjoint approaches, all the grid points may be variables but one accepts only overall outputs such as the gradient of drag with respect to these variables. For a local search, this is very attractive since typical downhill methods work very well in such circumstances (Kim *et al.* 2000). Impressive results have been reported using adjoint schemes to carry out geometry improvements using local and thus limited range searches. The approach is not very useful for large-scale geometric changes and is difficult to use with CAD-based geometries since it requires invasive changes to *all* the codes being used (or at least the way they are used). This has prevented routine adoption of such approaches by most aerospace companies.

*(b) Meshing*

Given a parameterization scheme capable of generating the geometries required, the next hurdle faced in building an automated analysis scheme (and such automation is virtually a prerequisite for optimization) is in generating a suitable analysis mesh. Almost all the CFD codes in use today require the fluid domain to be sub-divided (discretized) into meshes consisting of simple shapes such as triangles, tetrahedra, quadrilaterals or bricks. Although various mesh-free schemes have been proposed, these currently remain experimental codes and at the time of writing almost all practical design work is carried out with meshes of some form or another. The meshes may be structured or unstructured, body-fitted or Cartesian and all combinations of these have their pros and cons. The work described later uses non-body fitted structured meshes which are easy to set up over wide ranges of body shapes but which result in rather less efficient solver run times than more complex, but more difficult to create, body-fitted methods.

*(c) Solving*

Even when a suitable geometry and mesh have been created, the analyst's difficulties are not completely over. Even the most well established of commercial CFD codes have a variety of controls that must be set governing things such as domain division over CPUs, order of solution, number of iterations, turbulence models, etc. It is quite common to find that a certain amount of experimentation is needed to establish the most efficient combination of controls, allocation of processors, usage of memory and settings for convergence parameters. For example, if parallel execution is desired over a given number of processors, the user may need to choose between studying several geometries at one time, each with a few CPUs or instead dedicating all the available processors to each calculation, one at a time. In the work reported here, each solve is handled by its own dedicated CPU, but an optimization scheme allowing 10 simultaneous calculations is adopted to make best use of the 10 available licences for the commercial code used. If commercial codes are being used, total licence numbers can often influence such decisions as much as total available computing power.

*(d) Post-processing*

Having run a suitable CFD (or any other) solver to acceptable convergence, the designer will wish to decide on how to assess the resulting data. When dealing with the results of CFD, this will commonly involve dealing with integrated quantities such as lift or drag as well as with specific flow features such as the locations of any shocks or separations. Extracting such information automatically and routinely from the flow solution is also non-trivial, particularly if a quantity such as drag is to be calculated with precision and sub-divided into the components of interest to the engineer such as wave drag and induced drag. Some of the most commercially sensitive aspects of aerodynamic analysis can occur in this phase as various calibration coefficients are brought to bear which may encapsulate the results of many experiments and trials carried out over extended periods.



*(e) Optimization schemes*

Once an automated design analysis capability has been assembled, it is then possible to consider the use of optimization tools. Such methods adjust design variables, seeking to improve goal functions while meeting any specified constraints. They can be as simple as the steepest descent schemes that date back to Cauchy and Newton or complex hybrids making use of surrogate models, evolutionary searches and a whole range of heuristics. All aim to move quickly to improved goals, although some specifically sacrifice global coverage in the aim to quickly converge to a local optimum while others specialize in wide-ranging exploration. There are some fundamental limitations that must be recalled whenever working in this area, however. In particular, the so-called ‘no free lunch’ theorems show that when averaged over all possible problems, all search methods are equally good (Wolpert & Macready 1997). Thus, if one seeks to make a search faster, it will inevitably become more specialized and thus prone to misbehave on problems for which it has not been carefully set up. Most design teams like to have a range of search tools to hand which they can bring to bear depending on the number of variables, goals and constraints being dealt with, the speed of the analysis process and the nature of the variations in the functions being studied. This topic is considered in more detail in subsequent sections of this paper.

*(f) Databases*

Since so much analysis is computationally expensive and online storage increasingly cheap, it is now sensible to store as many design calculation results as possible online, so that they can be browsed and reused. Such reuse is not simply a case of caching results to avoid duplication, rather it is often possible to interpolate and regress among existing data to guide future search processes better. This can not only speed up the design process but can also improve its quality. To do this often requires that additional descriptive ‘meta-data’ be stored with each computational result, so that its provenance can be readily searched by the tools available to the designer. The automated integration of such data stores is currently a very active topic of research.

**3. A brief history of optimization methods**

The development of nonlinear numerical optimization methods is a very large field, but may be grouped into three main themes: the classical gradient-based methods and hill climbers of the 1960s; the evolutionary approaches that began to appear in the late 1970s; and the adaptation of DoE and response surface methods to computer simulations in the 1980s and 1990s. Classical hill climbers all seek to move uphill (or downhill) from some given starting point in the direction of improved designs. Sometimes, they use simple heuristics to do this (as in the Hooke & Jeeves (1960) search) and sometimes they use the local gradient to establish the best direction to move in (as in quadratic programming Boggs & Tolle 1995). Where needed, such gradients are commonly obtained by finite differencing but can sometimes be obtained directly (e.g. by using an adjoint code). Few new heuristic approaches seem to have found favour in the literature since the early work of the 1960s. Instead, most effort since that time

has been focused on improving the speed of convergence of quadratic methods and getting them to deal efficiently with constraints. Perhaps, the best known of these is the FSQP method developed by Panier & Tits (1993).

The so-called evolutionary methods are linked by the common thread of a stochastic approach to optimization. That is, the sequence of designs tested while carrying out the search has some random element—the main benefit is that such methods can avoid becoming stuck in local basins of attraction and can thus explore more globally. The downside is that they are often slower to converge to any optima they do identify. The most well-known methods in this class are the genetic algorithms (GAs), which seek to mimic the processes of natural selection that have proved so successful in adapting living organisms to their environments (Goldberg 1989). A variety of alternative stochastic global search methods have also been developed over the same period and these methods are now much interwoven in their development. For example, a series of developments made in Germany at around the same time led to the so-called evolutionary algorithms (Schwefel 1995) or the work on simulated annealing (Kirkpatrick *et al.* 1983). Some of these techniques work with populations of designs and are thus efficiently handled by clusters of computers: they are thus increasingly popular owing to developments in modern computing grid architectures where hundreds of processors may be used simultaneously.

Design of experiment and response surface methods are not really optimizers *per se*: rather, they are techniques that allow complex and computationally expensive optimization problems to be transformed into simpler tasks that can be tackled by the methods outlined in the previous two paragraphs. Essentially, these are curve fitting techniques that allow the designer to replace calls to an expensive analysis code by calls to a curve fit that aims to mimic the real code. Such methods all work in two phases: first, data are gathered on the nature of the function being represented by making a judiciously selected series of calls to the full code, usually in parallel—the placing of these calls in the design space is often best achieved using formal DoE methods, which aim to cover the search space in some statistically acceptable fashion (Montgomery 1997). Then, when the resulting data are available, the second phase consists of constructing a curve fit through or near the data—such curve fits are often termed ‘response surfaces’ (Myers & Montgomery 1995). The choice of RSMs that may be used is quite wide and will depend on the nature of the problem being tackled and the quantities of data available. Usually, the initial RSM will not be sufficiently accurate in all the areas of interest and so an iterative updating scheme may then be used where fresh calls to the full analysis code are used to provide additional information—various updating schemes have been proposed (Jones *et al.* 1998), see figure 1. The use of DoE and RSM methods in optimization is relatively recent but, even so, has yielded some impressive results (Sacks *et al.* 1989; Buck & Wynn 1993).

As probably goes without saying, very many hybrid approaches that try and combine methods from these classes have been experimented with. So much so that current practice is often to use a collection of methods in some complicated workflow to try and achieve results tailored to the problem in hand. Most commercial DSO toolkits support such an approach, often providing powerful graphical workflow editors to set up and control such hybrid searches. At the time of writing, optimization methods from each of these three classes are now sufficiently mature that they are available in commercial optimization packages



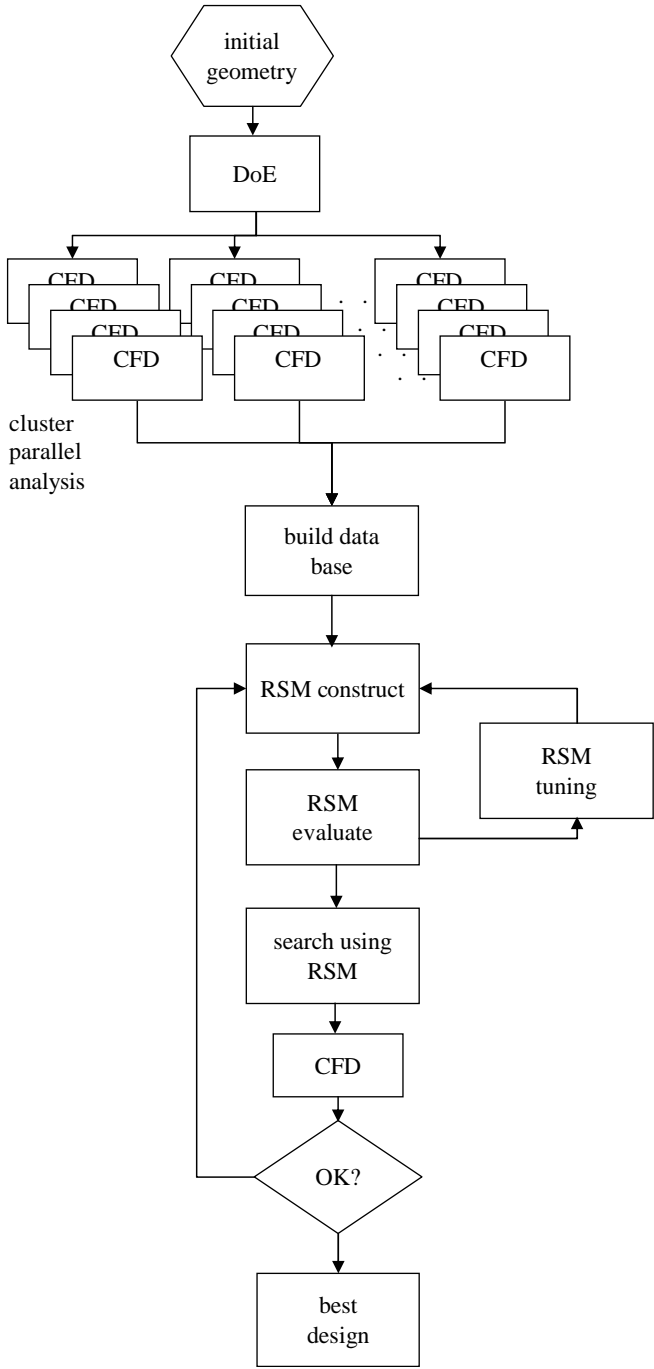


Figure 1. RSM-based optimization strategy.

and are in routine use by designers—academic research is currently more focused on how to deploy optimizers most effectively, given the desire to minimize the use of time-consuming and complex analysis codes, or to search very large design spaces,

also to exploit emerging features of analysis methods such as the sensitivities coming from CFD adjoint codes. For example, the published academic research on CFD optimization can be categorized as

- work that aims to make modest but detailed changes to refine an existing design via some kind of gradient descent approach on a single, deterministic goal function such as drag—this can involve impressively large CFD jobs using adjoint codes;
- inverse schemes that seek to modify a geometry so as to achieve a pre-specified flow behaviour, typically defined as a pressure distribution over the surface of a wing or airfoil—again commonly over very large meshes;
- wide-ranging searches over gross dimensions such as span and sweep so as to identify promising wing planforms, etc.—most commonly on modest cost panel or Euler codes;
- optimization using multiple operating point schemes or multiple objectives to allow for the fact that more complex design goals are the norm in real design work—less common and rarely carried out with full fidelity calculations; and
- models where uncertainty is specifically allowed for, such as where geometry may be uncertain or operating conditions unknown—relatively little work published to date but beginning to emerge.

Here, by way of example, we study planform design using the RSM approach—this technique is currently very popular when carrying out CFD-based optimization, particularly if wide-ranging searches are required.

#### 4. Kriging and probability of improvement

It is no surprise that there are a number of variations and refinements that may be applied to the basic RSM approach—the literature offers many possible alternatives. In this paper, an LP $\tau$  DoE sequence (Statnikov & Matusov 1995) is used to generate the initial set of points and a Kriging model applied to build the RSM. The Kriging approach allows the user to control the amount of regression as well as providing a theoretically sound basis for judging the degree of curvature needed to model the user's data adequately. Additionally, Kriging provides measures of probable errors in the model being built that can be used when assessing where to place any further design points. It also allows for the relative importance of variables to be judged.

In Kriging, the inputs  $\mathbf{x}$  are assumed to be related to the output (response)  $y$  by an expensive function  $f_e(\mathbf{x})$ . The response of the code is then evaluated for combinations of inputs generated by the DoE and used to construct an approximation

$$\hat{y} = \hat{f}_e(\mathbf{x}). \quad (4.1)$$

The response at any  $\mathbf{x}$  is then approximated by

$$y = \mu + \varepsilon(\mathbf{x}), \quad (4.2)$$

where  $\mu$  is the mean of the responses and  $\varepsilon(\mathbf{x})$  is a Gaussian random function with zero mean and variance  $\sigma^2$ . In Kriging,  $\varepsilon$  is taken to depend on the distance between corresponding points. The distance measure used here is

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{h=1}^k \theta_h \left( \mathbf{x}_h^{(i)} - \mathbf{x}_h^{(j)} \right)^{p_h}, \quad (4.3)$$

where  $\theta_h$  and  $p_h$  are hyper-parameters tuned to the data in hand. The correlation between points  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$  is given by

$$\mathbf{R}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp[-d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})] + \lambda \delta_{ij}, \quad (4.4)$$

where  $\lambda$  is a regularization constant that governs the degree of regression in the model (when set to zero the Krig strictly interpolates the data supplied) and  $\delta_{ij}$  is the Dirac delta function. When the response at a new point  $\mathbf{x}$  is required, a vector of correlations between the point and those used in the DoE is formed,  $\mathbf{r}(\mathbf{x}) = \mathbf{R}(\mathbf{x}, \mathbf{x}^{(i)})$ . The prediction is then given by

$$\mathbf{y}(\mathbf{x}) = \mu + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu), \quad (4.5)$$

where the mean  $\mu$  is found from

$$\mu = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}}. \quad (4.6)$$

The hyper-parameters  $\theta_h$ ,  $p_h$  and regularization constant  $\lambda$  are all obtained by maximizing the likelihood, defined as

$$\frac{1}{(2\pi)^{N/2} (\sigma^2)^{N/2} |\mathbf{R}|^{1/2}} \exp \left[ \frac{-(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2} \right], \quad (4.7)$$

where the variance  $\sigma^2$  is given by

$$\sigma^2 = \frac{(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{N}. \quad (4.8)$$

$N$  is the number of points used in the DoE. The mean squared error of the prediction is

$$s^2(\mathbf{x}) = \sigma^2 \left[ 1 + \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \quad (4.9)$$

which gives a measure of the accuracy of the Krig at  $\mathbf{x}$ . Another of the attractions of Kriging is that the  $\theta$  hyper-parameters produced may be used to screen the variables in the data for relative importance if the input variables are normalized to a unit range before the Krig is tuned. Once tuned, the hyper-parameters simply rank the significance of the variables they represent.

Having built an initial RSM, some thought must be given to how well it models the data used and if this modelling is fit for purpose. Commonly, it is not and some kind of improvement strategy is employed to help refine the process. Suppose that an initial set of training data is made available by running the high-fidelity model at the points generated by a space-filling DoE technique, i.e.  $D^0 \equiv \{\mathbf{x}^{(i)}, y(\mathbf{x}^{(i)})\}$ ,  $i=1, 2, \dots, N_0$ . This dataset is then used to construct a Krig model to approximate the input–output relationship where the mean squared error of the prediction gives an estimate of the uncertainty involved in making predictions using a finite set of input–output data. From the viewpoint of DoE, to improve the accuracy of the baseline surrogate, it is sensible to augment the dataset  $D^0$  with additional points where this error is high. However, from the perspective of finding iterates that lead to reductions in the objective function, the aim is to minimize  $\hat{y}(\mathbf{x})$ . Statistical improvement criteria attempt to balance these aims in a rational way. Here, we detail the so-called ‘probability of improvement’ although there are several alternatives (Keane & Nair 2005).

Since a Krig model is a Gaussian process, the probability of any newly calculated design point  $y(\mathbf{x}^{(N_0+1)})$  representing an improvement over the current best design,  $f_e^{\min}(\mathbf{x}) = \min(f_e^{(1)}(\mathbf{x}^{(1)}), f_e^{(2)}(\mathbf{x}^{(2)}), \dots, f_e^{(N_0)}(\mathbf{x}^{(N_0)}))$ , is readily calculated from

$$\begin{aligned} P[I] &= P\left[y(\mathbf{x}^{(N_0+1)}) \leq f_e^{\min}(\mathbf{x})\right] \\ &= \int_{-\infty}^{f_e^{\min}(\mathbf{x})} \frac{1}{s(\mathbf{x}^{(N_0+1)})\sqrt{2\pi}} \exp\left(-\frac{1}{2}(\hat{y} - \mu(\mathbf{x}^{(N_0+1)}))^2 / s^2(\mathbf{x}^{(N_0+1)})\right) d\hat{y} \\ &= \Phi\left(\frac{f_e^{\min}(\mathbf{x}) - \mu(\mathbf{x}^{(N_0+1)})}{s(\mathbf{x}^{(N_0+1)})}\right), \end{aligned} \quad (4.10)$$

where  $[\cdot]$  denotes the expectation operator and  $\Phi(\cdot)$  is the normalized Gaussian distribution function. This quantity is just the area in the tail of the Gaussian distribution below the current best function value  $f_e^{\min}(\mathbf{x})$  and indicates the probability that any new design will represent an improvement over those in the existing training data, but does not indicate how much of an improvement will be obtained. Therefore, to minimize the original objective function  $f_e(\mathbf{x})$ , the procedure set out in figure 1 can be followed: first, a baseline Krig model is constructed using  $N_0$  points generated by applying a space-filling DoE technique. Subsequently, a new iterate is generated by maximizing the probability of improvement criteria, i.e.  $\mathbf{x}^{(N_0+1)} = \arg \max_{\mathbf{x}} P[y(\mathbf{x}^{(N_0+1)}) \leq f_e^{\min}(\mathbf{x})]$ . The high-fidelity model is then evaluated at  $\mathbf{x}^{(N_0+1)}$  and the resulting exact function value,  $f_e(\mathbf{x}^{(N_0+1)})$ , is added to the baseline training dataset to give  $D^1$ . The augmented dataset is used to update the Krig and its hyper-parameters, which is then used to solve the previous equation for the next iterate. This process is continued until specified convergence criteria are met.

Kriging is not a panacea for all evils, however. It is commonly found that it is difficult to set up such models for more than 20 or so variables and also that the approach is numerically expensive if there are more than a few hundred data points, since the set-up (hyper-parameter tuning) process requires the repetitive LU decomposition of the correlation matrix  $\mathbf{R}$ , which has the same dimensions as the number of points used. Moreover, the number of such LU steps is strongly dependent on the number of variables and the likelihood is commonly highly multi-modal.

## 5. Multi-objective design: Pareto fronts

Most real design problems have more than one goal that the designer is trying to improve. In aerospace design, it is common to be aiming for light weight, low-cost, robust, high-performance systems. These aspirations are clearly in tension with each other and so compromise solutions have to be sought. Such compromises inevitably involve deciding on some form of weighting between the desired goals. However, before this stage is reached, it is possible to study design problems from the perspective of Pareto sets. A Pareto set of designs is one whose members are all optimal in some sense, but where the relative weighting

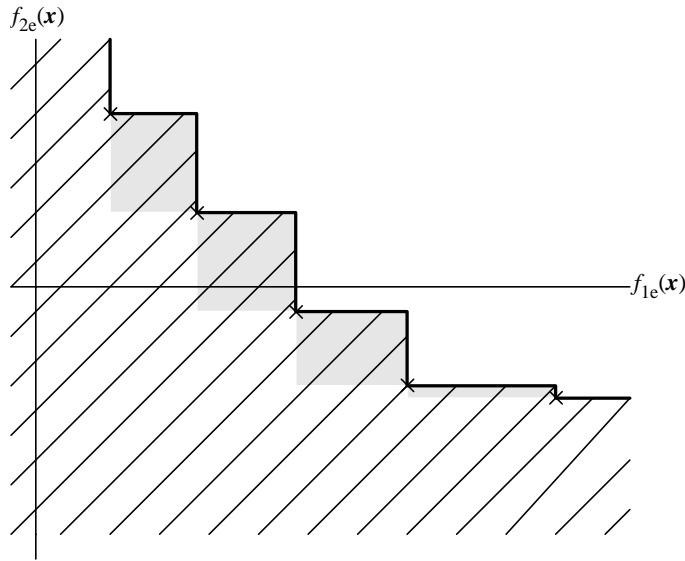


Figure 2. A typical Pareto front (all points added in the hatched area add to the front, those in the hatched and shaded area augment only, while those in the rest of the hatched area dominate, and thus replace, existing set members).

between the competing goals is yet to be finally fixed (Fonseca & Fleming 1995). More formally, a Pareto set of designs contains systems that are sufficiently optimized that, to improve the performance of any set member in any one goal function, its performance in at least one of the other functions must be made worse. Moreover, the designs in the set are said to be ‘non-dominated’ in that no other set member exceeds a given design’s performance in *all* goals. It is customary to illustrate a Pareto set by plotting the performance of its members against each goal function (figure 2). The series of horizontal and vertical lines joining the set members is referred to as the ‘Pareto front’—any design lying above and to the right of this line is dominated by members of the set.

There are a number of technical difficulties associated with constructing Pareto sets. First, the set members need to be optimal in some sense—since it is desirable to have a good range of designs in the set, this means that an order of magnitude more optimization effort is usually required to produce such a set than to find a single design that is optimal against just one goal. Second, it is usually necessary to provide a wide and even coverage in the set in terms of the goal function space—since the mapping between design parameters and goal functions is usually highly nonlinear, gaining such coverage is far from simple. Finally, and in common with single objective design, many problems of practical interest involve the use of expensive computer simulations to evaluate the performance of each candidate, and this means that only a limited number of such simulations can usually be afforded.

Currently, there appear to be two popular ways of constructing Pareto sets (Keane & Nair 2005). First, and most simply, one chooses a (possibly nonlinear) weighting function to combine all the goals in the problem of interest into a single quantity and carries out a single objective optimization. The weighting

function is then changed and the process repeated. By slowly working through a range of weightings, it is possible to build up a Pareto set of designs. Alternatively, one can try and use a population-based search to locate a whole series of points on the front in one go. Perhaps, the most well known of these schemes is the NSGA-II method introduced by Deb *et al.* (2000). In that approach, a GA is used to carry out the search, but the goal function used to drive the genetic process is based on the relative ranking and spacing of the designs in the set rather than their combined weighted performance. More specifically, at each generation, all the designs are compared and the non-dominated designs set to one side. These are assigned rank one. The remaining designs are compared and those that now dominate are assigned rank two and so on. Thus, the whole population is sorted into rank order based on dominance. This sorting into rank-order dominance can be carried out irrespective of the relative importance of the objectives being dealt with or the relative magnitudes and scaling of these quantities.

Having sorted the population of designs into ranks, they are next rewarded or penalized depending on how close they are to each other in goal space (and sometimes also in design variable space). This provides pressure to cause the search to fan out and explore the whole design space, but does require that the competing objectives be suitably scaled—an important issue that arises in many aspects of dealing with multi-objective approaches to design. When combined with the traditional GA operators of selection, crossover and mutation, the NSGA-II scheme is remarkably successful in evolving high-quality Pareto sets. As originally described, however, no means were provided for mitigating run-time issues arising from using expensive computer simulations in assessing competing designs.

To overcome the problem of long run times, a number of workers have advocated the use of RSM approaches, including Kriging within Pareto front frameworks (Wilson *et al.* 2001; Knowles & Hughes 2005). It is also possible to combine tools such as NSGA-II with Kriging (Voutchkov & Keane 2006). In such schemes, an initial DoE is carried out and RSMs built as per the single objective case, but now there is one RSM for each goal function. In the NSGA-II approach, the search is simply applied to the resulting RSMs and used to produce a Pareto set of designs. These designs are then used to form an update set, and after running full computations, the RSMs are refined and the approach continued. Although sometimes quite successful, this approach suffers from an inability to balance exploration and exploitation explicitly in the RSM construction in just the same way as when using such models in single objective search and greedily seeking for the best designs, although the crowding or niching measures normally used help mitigate these problems to some extent. The central thrust of the probabilistic improvement approach used here is to explicitly tackle this problem.

## 6. Multi-objective probability of improvement

Assuming that we have an expensive multi-objective search problem that is being tackled using a combined DoE and Kriging approach, it is possible to revisit the ideas of improvement and devise appropriate metrics. To begin with,



consider a problem with two expensive goal functions  $f_{1e}(\mathbf{x})$  and  $f_{2e}(\mathbf{x})$  with outputs (responses)  $y_1$  and  $y_2$  that must both be minimized. Moreover, for simplicity, assume that  $\mathbf{x}$  consists of just one design variable  $x$ . By constructing a DoE as before, it will be possible to construct a set of training data. This will allow us to identify the initial Pareto set of  $M_0$  designs that dominate all the others in the training set

$$\mathbf{f}_{1,2}^* = \left( \left( f_{1e}^{(1)*}(x^{(1)}), f_{2e}^{(1)*}(x^{(1)}) \right), \left( f_{1e}^{(2)*}(x^{(2)}), f_{2e}^{(2)*}(x^{(2)}) \right), \dots, \left( f_{1e}^{(M_0)*}(x^{(M_0)}), f_{2e}^{(M_0)*}(x^{(M_0)}) \right) \right).$$

In this set, the superscript  $*$  indicates that the designs are non-dominated. We may plot these results on the Pareto front axes as per figure 2 discussed in §5. In figure 2, the solid line is the Pareto front and the hatched area represents locations where new designs would need to lie if they are to become members of the Pareto set.

Given the training set, it is also possible to build a pair of Krig meta-models. Here, it is assumed that these models are independent though it is also possible to build correlated models by adopting the formalism known as co-Kriging (Cressie 1993). The pair of meta-models  $\hat{y}_1(x)$  and  $\hat{y}_2(x)$  will be Gaussian processes as before and each term in the two models will be identified by suffixes 1 and 2, respectively. Given a proposed new design point  $x$ , this pair of models will provide a prediction of the two goal function values and also their standard errors, here taken to be uncorrelated. These values may then be used to construct a two-dimensional Gaussian probability density function for the predicted responses that accords with the predicted mean and errors coming from the two Krig models at  $x$ .

When seeking to add a new point to the training data, we wish to know the likelihood that any newly calculated point will be good enough to become a member of the current Pareto set and, when comparing competing potential designs, which will improve the Pareto set most. There are a number of metrics that can be considered in these circumstances (see Keane (2006) for a full discussion) here we outline just the simplest; the probability that a new design will be good enough to at least augment the existing Pareto set. This is given by integrating the joint probability density function over the hatched area in figure 2,

$$\begin{aligned} P\left[\hat{y}_1(x^{(N_0+1)}) \leq \mathbf{f}_{1e}^*(x) \cup \hat{y}_2(x^{(N_0+1)}) \leq \mathbf{f}_{2e}^*(x)\right]_{\text{aug}} &= \int_{-\infty}^{\mathbf{f}_{1e}^{*(1)}} \int_{-\infty}^{\infty} \phi(\hat{y}_2, \hat{y}_1) d\hat{y}_2 d\hat{y}_1 \\ &+ \sum_{i=1}^{M_0-1} \int_{\mathbf{f}_{1e}^{(i)}}^{\mathbf{f}_{1e}^{*(i+1)}} \int_{-\infty}^{\mathbf{f}_{2e}^{*(i)}} \phi(\hat{y}_2, \hat{y}_1) d\hat{y}_2 d\hat{y}_1 \\ &+ \int_{\mathbf{f}_{1e}^{*(M_0)}}^{\infty} \int_{-\infty}^{\mathbf{f}_{2e}^{*(M_0)}} \phi(\hat{y}_2, \hat{y}_1) d\hat{y}_2 d\hat{y}_1, \end{aligned} \quad (6.1)$$

or

$$\begin{aligned}
 P[I]_{\text{aug}} = & \Phi \left( \frac{f_{1e}^{*(1)}(x) - \mu_1(x^{(N_0+1)})}{s_1(x^{(N_0+1)})} \right) \\
 & + \sum_{i=1}^{M_0-1} \left\{ \Phi \left( \frac{f_{1e}^{*(i+1)}(x) - \mu_1(x^{(N_0+1)})}{s_1(x^{(N_0+1)})} \right) - \Phi \left( \frac{f_{1e}^{*(i)}(x) - \mu_1(x^{(N_0+1)})}{s_1(x^{(N_0+1)})} \right) \right\} \\
 & \times \Phi \left( \frac{f_{2e}^{*(i)}(x) - \mu_2(x^{(N_0+1)})}{s_2(x^{(N_0+1)})} \right) \\
 & + \left\{ 1 - \Phi \left( \frac{f_{1e}^{*(M_0)}(x) - \mu_1(x^{(N_0+1)})}{s_1(x^{(N_0+1)})} \right) \right\} \Phi \left( \frac{f_{2e}^{*(M_0)}(x) - \mu_2(x^{(N_0+1)})}{s_2(x^{(N_0+1)})} \right).
 \end{aligned} \tag{6.2}$$

This metric is, of course, non-dimensional and is the multi-objective equivalent of equation (4.10). Moreover, it works irrespective of the relative scaling of the objectives being dealt with.<sup>2</sup>

## 7. A typical aircraft concept design problem: cost versus performance

Having set out the basic tools needed for our multi-level approach, we next illustrate their use in wing design using an Euler-based CFD solver and a generative cost model. Here, the responses being studied are the drag and cost of a transonic civil transport wing. A simple test problem has been constructed with the aim of optimizing the wing for operation at Mach 0.785 and a Reynolds number of 7.3 million. The objective is minimization of wing drag area, also known as drag per unit dynamic pressure,  $D/q$  (*drag coefficient*  $\times$  *wetted area*) as calculated by the CFD solver and manufacturing cost in £m with target lift, wing weight, volume, pitchup margin and root triangle layout chosen to be representative of a 220-seat wide-body airliner. Limits are placed on the design variables that are typical of work in this area (although they still admit designs that would be considered radical in practice—it is not common to use sweep angles as high as 45° in a civil aircraft, for example). To make this study sensibly realistic, and also to provide data for the costing process, a relatively detailed structural and weight estimation process is incorporated into the calculation.

### (a) Weight estimation

The most common way to calculate the weight of the load-carrying part of the wing is to split the structure into the parts that resist bending and those that resist shear. For the aircraft category under consideration (subsonic transport and executive aircraft), the wing is generally built up from the subparts depicted in figure 3. Here, the wing weight is estimated using an analytical–empirical

<sup>2</sup> N.B., the subscript aug is used to indicate that this is the probability of augmenting (as opposed to dominating) members in the existing front.

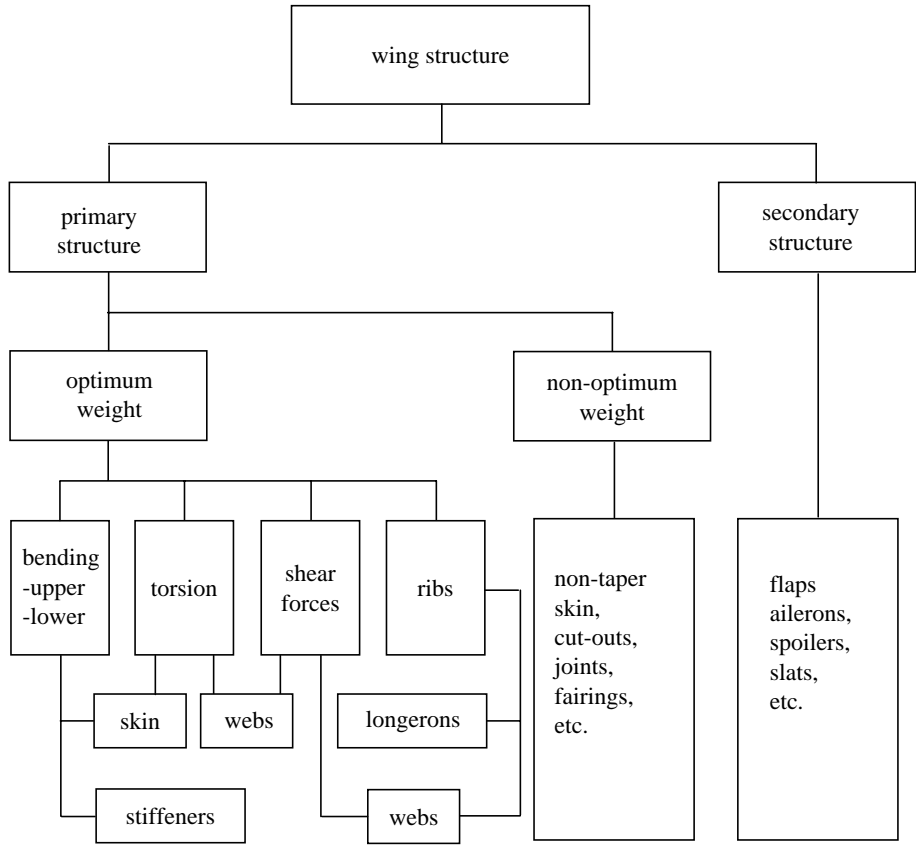


Figure 3. Subdivision of wing and weight distributions.

approach that is a development of [Torenbeek's \(1992\)](#) method developed by [Cousin & Metcalfe \(1990\)](#). In such methods, the wing weight is computed as the sum of several functional components, each of which is estimated via a rational and/or statistical approach. The methods are especially suitable for the preliminary design stage, when sensitivity studies are required on the effects of geometric and other variations on the design characteristics.

Once the wing weight has been evaluated, this has to be included in the aircraft weight build-up. The design aircraft weight can be broken into crew weight, payload weight, fuel weight and operational empty weight (structure, engines, landing gear, fixed equipment, avionics and anything else not considered part of crew, payload or fuel). The crew and payload weights are both known since they are given in the design requirement. The only unknowns are the fuel weight and the operational empty weight, both of which are dependent on the total aircraft weight. Thus, an iterative process must also be used for aircraft sizing. For the purposes of this study, it has been assumed that part of the operational empty weight (i.e. excluding the weight of the wing structure) is fixed (in the range of our search) and that the maximum fuel weight is dependent on the size of the wing box (which houses the fuel tanks). [Table 1](#) gives typical structural design parameters and [table 2](#) the resulting weight estimates.

Table 1. Wing structure parameters.

quantity	value
wing LE as a fraction of local chord	0.153
wing TE as a fraction of local chord	0.284
wing LE as a fraction of centerline chord	0.153
wing TE as a fraction of centerline chord	0.350
no. of wing-mounted engines	2
position of the engine as a fraction of semi-span	0.598
mass of one wing-mounted engine (kg)	9000
inner tank boundary limit as a fraction of semi-span	0.0
inner/outer tank boundary position as a fraction of semi-span	0.58
outer tank boundary limit as a fraction of semi-span	0.82
chordwise fraction of TE that the flap occupies	0.7555
chordwise fraction of TE that the slat occupies	0.9364
spanwise extent of outboard flap as a fraction of semi-span	0.7714
rib pitch (m)	0.75

Table 2. Aircraft weight build-up.

item	weight (kg)
lower panel	3325.7
upper panel	2621.1
spars	1445.0
light rib	322.9
U/C pickup structure	625.4
trailing edge	2350.2
leading edge	1055.0
miscellaneous	1305.0
total wing structure	13 050.3
wingless weight	78 515.3
maximum fuel mass	33 507.5
maximum take-off weight	125 073.1
cruise fuel mass (fuel fraction=0.37)	12 397.8
cruise weight	103 963.4

(b) Drag prediction

Here, drag is computed using the commercial CFD code MGAERO, which is a viscous coupled multigrid Cartesian Euler solver (Epstein *et al.* 1989). A series of drag recovery routines are incorporated with this code to assess the various drag components in a fashion compatible with typical concept design tools (Squire & Young 1937; Lock 1986; van Dam & Nikfetrat 1992). The input geometries to the CFD solver are created using a set of orthogonal functions derived from NACA transonic foils (Harris 1990; Robinson & Keane 2001). When using MGAERO, the inviscid analysis is coupled with a boundary layer model via surface transpiration. The viscous drag is then calculated using Cooke’s (1973) implementation of the Squire & Young (1937) method. The wave drag is computed via Lock’s (1986)

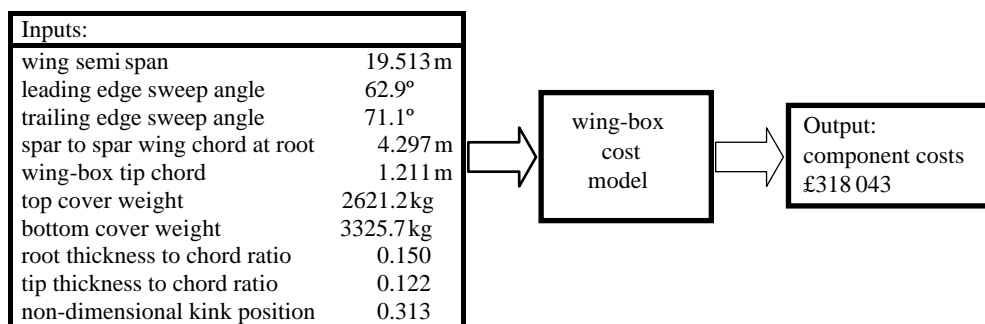


Figure 4. Top-level view of cost model.

second-order method. The induced drag is obtained from the integration of vorticity immediately downstream of the trailing edge as dissipation due to the presence of numerical viscosity precludes integration in the Trefftz plane. Of course, such drag values are subject to the usual inaccuracies associated with drag recovery from Euler codes ([van Dam & Nikfetrat 1992](#)), and by the standards of the other papers presented in this theme issue, this calculation is somewhat pedestrian; it nevertheless represents typical commercial design office practice. The CFD analysis requires around 30 min on a 3 GHz processor.

### (c) Cost modelling

The cost model uses a generic hierarchical modelling tool called DECISIONPRO developed by the Vanguard Software Corp., Cary NC, USA. The significance of this approach is that it allows libraries of reusable cost objects to be created. This leads to a very efficient and extensible model that is relatively easy to navigate and deploy through a standard web browser. Further details on this methodology applied to gas turbine design can be found in [Scanlan \*et al.\* \(2006\)](#).

For the purposes of optimization, the cost model can be viewed as a 'black-box' as illustrated in [figure 4](#). The internal structure of the cost model comprises cost-object libraries of

- part types (stringers, spars, ribs and skins);
- processes; and
- materials.

The model includes an aggregation tree to sum the component hierarchy that is produced by a wing-structure model ([figure 5](#)). The wing-structure model generates the component hierarchy by instantiating the part-type cost-objects and populating their geometrical properties. Each part-type cost-object calls a set of process functions to determine manufacturing costs for a set of process steps. A particularly useful feature within the DECISIONPRO software is the ability to define variables in terms of their relevant physical units. The tool is able to undertake dimensional analysis (using SI units) to ensure consistency in any calculations. This eliminates many errors that could be difficult to identify otherwise. A sample search created an initial set of results of varying cost. [Figure 6](#) gives an example of the response of the cost model to two different inputs.

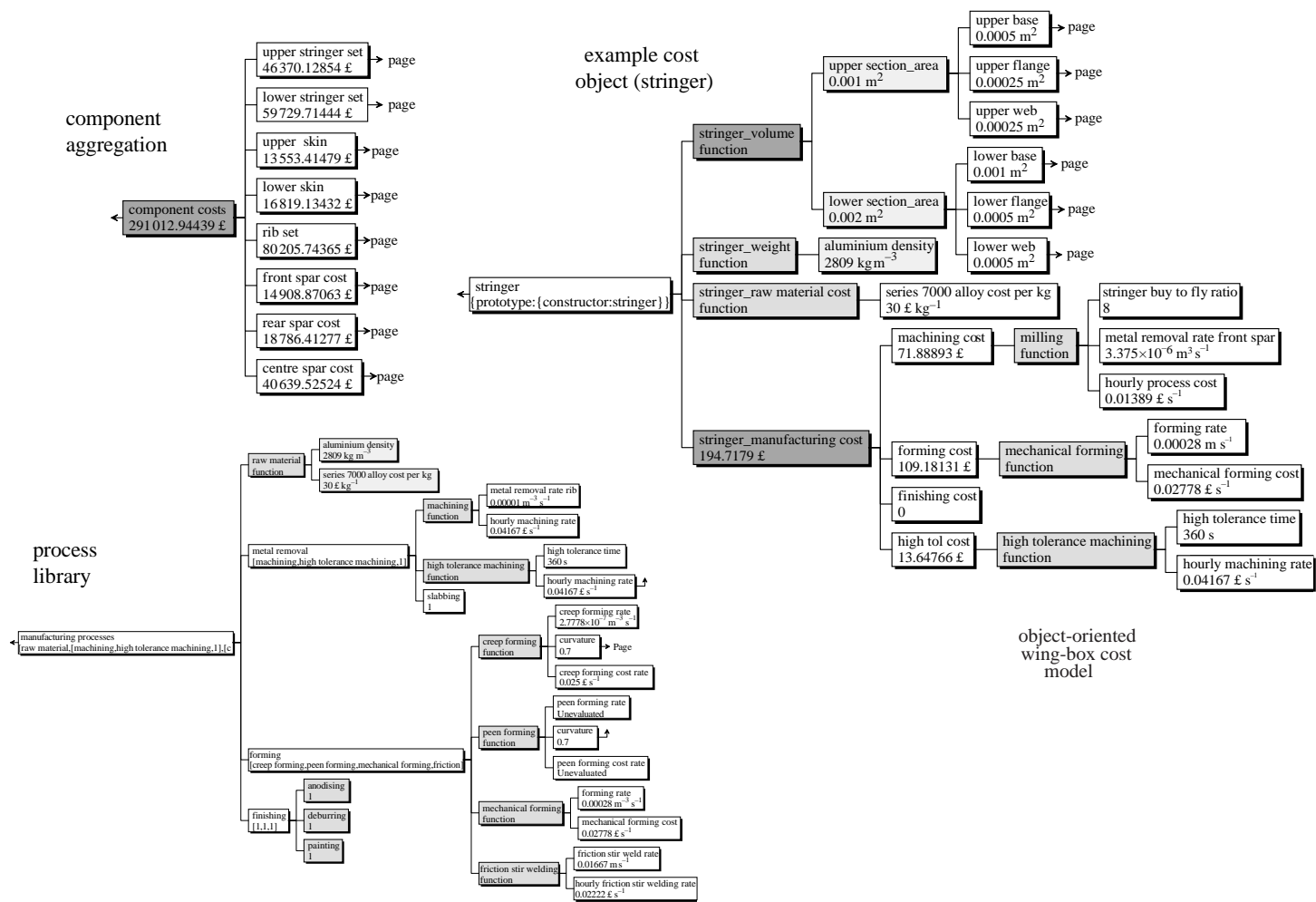


Figure 5. Illustration of cost model structure.



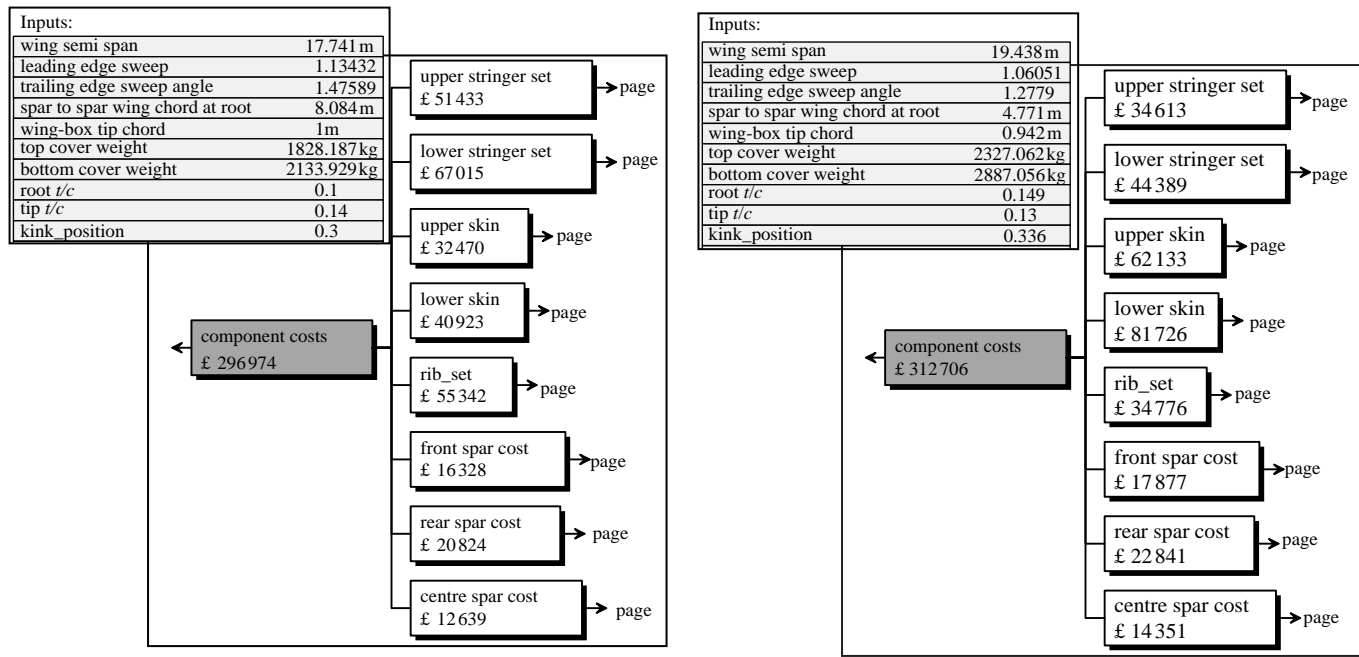


Figure 6. Example outputs for two different wing designs.

Table 3. Design parameters, constraint values and objective function values—direct CFD search.

lower limit	initial value	direct CFD	upper limit	quantity (units)
100	168	177.5	250	wing area (m <sup>2</sup> )
6	9.07	9.30	12	aspect ratio
0.2	0.313	0.406	0.45	kink position
25	27.1	25.2	45	sweep angle (degrees)
0.4	0.598	0.683	0.7	inboard taper ratio
0.2	0.506	0.259	0.6	outboard taper ratio
0.1	0.150	0.143	0.18	root $t/c$
0.06	0.122	0.096	0.14	kink $t/c$
0.06	0.122	0.069	0.14	tip $t/c$
4.0	4.5	4.5	5.0	tip washout (degrees)
0.65	0.75	0.67	0.84	kink washout fraction
	127 984	130 166	135 000	wing weight (N)
40.0	41.73	41.6		wing volume (m <sup>3</sup> )
	4.179	3.67	5.4	pitch-up margin
2.5	2.693	2.56		undercarriage bay length (m)
	2.922	2.524		$D/q$ (m <sup>2</sup> )
	0.502	0.453		cost (£ m)

(d) *Sample calculation*

A typical result from this combined weight, CFD and costing analysis is detailed in the initial value column of [table 3](#) while [figure 7](#) illustrates the equivalent geometry and overall mesh. Note that in this case, the wing is defined by 11 parameters and also that constraints are placed on the wing volume, undercarriage bay length, pitch-up margin and weight. At all times, the angle of attack is set to generate the required lift and the wing weight changes in a realistic fashion, allowing for necessary structural modifications as its dimensions alter. Given this analysis capability, it is possible to carry out a direct search: column 3 of [table 3](#) gives the results of such a study, using a direct GA optimization of 20 generations and a population size of 100 aimed solely at reducing drag and allowing the cost to float. This search represents some 75 *days* of computing effort; here, carried out on a cluster of PCs running in parallel over a week. The extreme cost of such searches makes them infeasible for everyday use—but they do provide benchmarks against which to compare other results. Here, the design variables are limited by the bounds also shown in the table. Note that the drag is reduced by some 14%, the cost by 11% and that the optimized wing has a significantly larger area. This demonstrates that the initial design can be improved on in both aerodynamic and cost terms, but does not indicate how any further trades might be made between these two aspects.

(e) *Multi-objective design optimization*

As will be obvious from the previous two calculations, optimizing a design for minimum drag does not always increase cost, particularly if the starting point for the search has not been specifically optimized for minimum cost or weight. To examine what interactions there might be between these two aspects, we

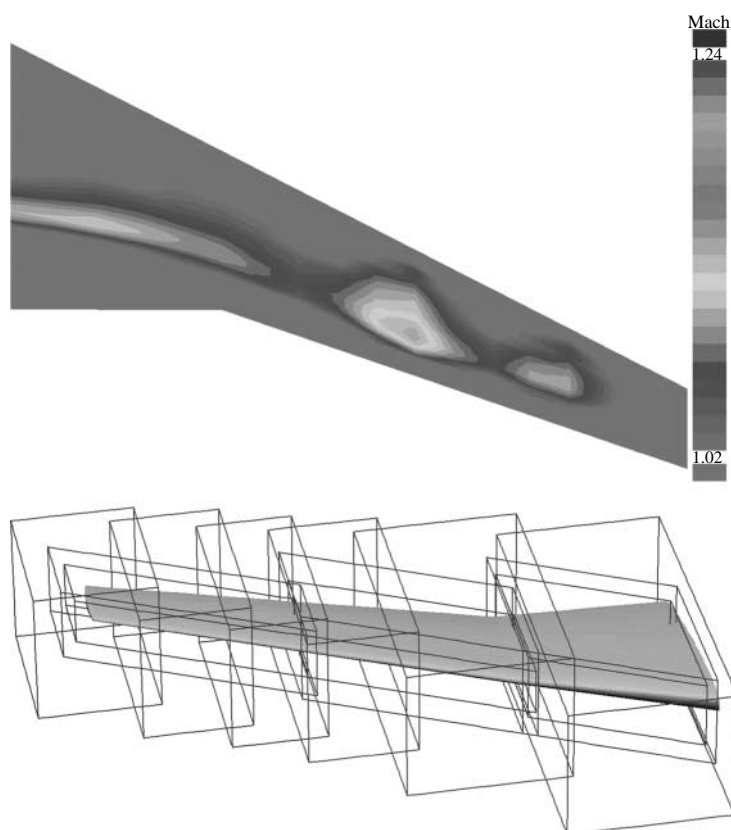


Figure 7. Initial wing geometry and overall CFD meshing (plan view shows upper surface supersonic Mach contours).

therefore next use the probability of the improvement metric and update scheme outlined earlier to produce a Pareto front of nine trade-offs possible between cost and drag (figure 8). This front has been produced using an initial set of 250 analysis runs to build the first Krig database, followed by a further 250 runs in batches of 10 found by searching equation (6.2) for good candidate solutions (the fact that this function is highly multi-modal means that it is always relatively easy to find such batches of good candidates). Of these 500 runs, some 45% fail to yield results either because the CFD meshing scheme being used cannot build an acceptable mesh or because the resulting CFD run does not converge correctly in either the inviscid or viscous coupled solves. These failed points are not included when constructing the Krig, but their presence is allowed for in the update scheme using the concept of imputation (Forrester *et al.* 2005), which allows the process to be warned that samples have already been taken in a given location. This is necessary to prevent the update process continually revisiting areas of the design space where solutions cannot be generated. Since these 500 runs can be handled in parallel, exploiting the 10 available licences, this search process requires around 5 days of computing effort. Note also that the design previously found by simply optimizing for low drag is not located by the multi-objective search, being slightly less than 1% lower in drag than the lowest drag design

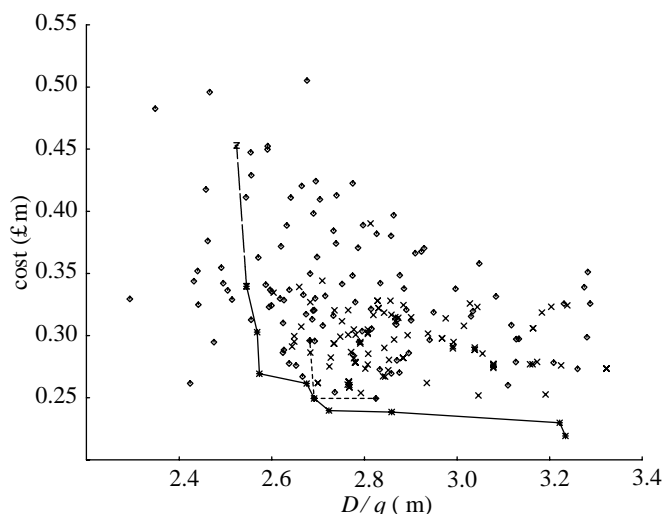


Figure 8. Pareto front of cost versus  $D/q$  showing the front after the initial DoE (dashed) and that after 250 updates (solid, with dashed extension to the design found by pure CFD optimization), together with the successful initial DoE points (open diamond) and the successful update points (times).

found on the front—it could thus be added to make a 10th point in the Pareto set—this is indicated in the figure by the dashed line linking the points mark with Z.

Also shown in figure 8 are the results from the initial 250 point DoE, all the successful update points and the position of the front after the initial DoE. Note that those design points that lie below and to the left of the front are not considered for membership of the front because they violate one or more of the constraints—in this case, either the wing weight or volume constraints are the active aspects in the design. It is clear that the update scheme being used here not only rapidly identifies good designs but also that these are nicely spread along the front, providing designers with a wide range of options from which to choose. Interestingly, one design (the 157th) from the initial DoE remains on the Pareto front even after the update sequence, moreover this also is quite a well-balanced design—its presence demonstrates the benefit of using a good quality space filling DoE at the outset.

Since the designs that make up the final front span a wide range of costs and aerodynamic performance, it is interesting to examine three cases taken from the front in more detail (table 4). Here, information is provided for the designs at either end of the front found by the update search, to illustrate the range of variation possible while still providing some degree of optimality, while the third design represents a reasonable trade-off position of intermediate cost with fairly low drag. Note that these three designs have notably different aspect ratios and weights with the low drag design having longer more slender and more swept back wings which are structurally less efficient and thus heavier and more costly, while the low-cost design has strongly tapered thick and deep inboard sections that lead to a very structurally efficient design of low weight and cost, but at the expense of 27% more drag, though being 35% cheaper. The intermediate design seems a much better compromise with drag only 6% higher than the low drag design, while being nearly 27% cheaper.

Table 4. Design parameters, constraint values and objective function values—multi-objective search.

lower limit	initial	low drag	DoE point 157	low cost	upper limit	quantity (units)
100	168	158.1	161.1	164.1	250	wing area (m <sup>2</sup> )
6	9.07	10.06	8.457	8.000	12	aspect ratio
0.2	0.313	0.3906	0.3465	0.300	0.45	kink position
25	27.1	29.62	25.53	25.82	45	sweep angle (degrees)
0.4	0.598	0.5385	0.5184	0.7000	0.7	inboard taper ratio
0.2	0.506	0.2678	0.2516	0.4044	0.6	outboard taper ratio
0.1	0.150	0.1735	0.1216	0.1800	0.18	root <i>t/c</i>
0.06	0.122	0.09758	0.1059	0.1400	0.14	kink <i>t/c</i>
0.06	0.122	0.1182	0.07031	0.0600	0.14	tip <i>t/c</i>
4.0	4.5	4.090	4.816	4.441	5.0	tip washout (degrees)
0.65	0.75	0.7344	0.7961	0.8500	0.85	kink washout fraction
	127 984	134 744	114 922	112 146	135 000	wing weight (N)
40.0	41.73	43.59	40.63	46.9656		wing volume (m <sup>3</sup> )
	4.179	5.001	3.313	3.34306	5.4	pitch-up margin
2.5	2.693	2.729	3.216	2.64626		undercarriage bay length (m)
	2.922	2.545	2.691	3.234		<i>D/q</i> (m <sup>2</sup> )
	0.502	0.338	0.248	0.218		cost (£m)

All three new designs are significantly cheaper than the baseline geometry or the purely CFD-optimized design (see figure 9 for a breakdown of these costs). Two of the three also offer lower drag than the baseline, showing how far from optimal this initial geometry was. The major cost increase in the low drag design when compared with the other two wings is primarily due to the large increase in skin costs due to the increase in skin thicknesses driven by the cover-weight estimates. The other differences are more minor and are driven by differences in geometry and number of features such as cut-outs, etc. Figure 10 shows the planform shape and upper surface Mach contours for DoE point 157 and it is clear that an enlarged inboard root section that is shock-free has been obtained at the expense of a slight shock in the reduced chord tip areas.

While there is no suggestion that any of these designs should be used in practice, it is hoped that this brief demonstration of what is possible using the latest generation DSO tools will be of value to practising designers, indicating that significant cost savings can often be made at the concept level in exchange for relatively modest penalties in aerodynamic performance.

8. Conclusions

This paper briefly reviews the current state-of-the-art of design optimization as it applies to aerospace engineering. It notes that if design improvements are to be made in a realistic industrial setting, a significant suit of capabilities must be assembled and managed together and that such tasks span CAD, classical engineering and modern computing grids. It then goes on to demonstrate what may be achieved in multi-discipline, multi-objective design by using a three-dimensional Euler CFD code, a modern generative costing approach and the sophisticated use of DoE and response

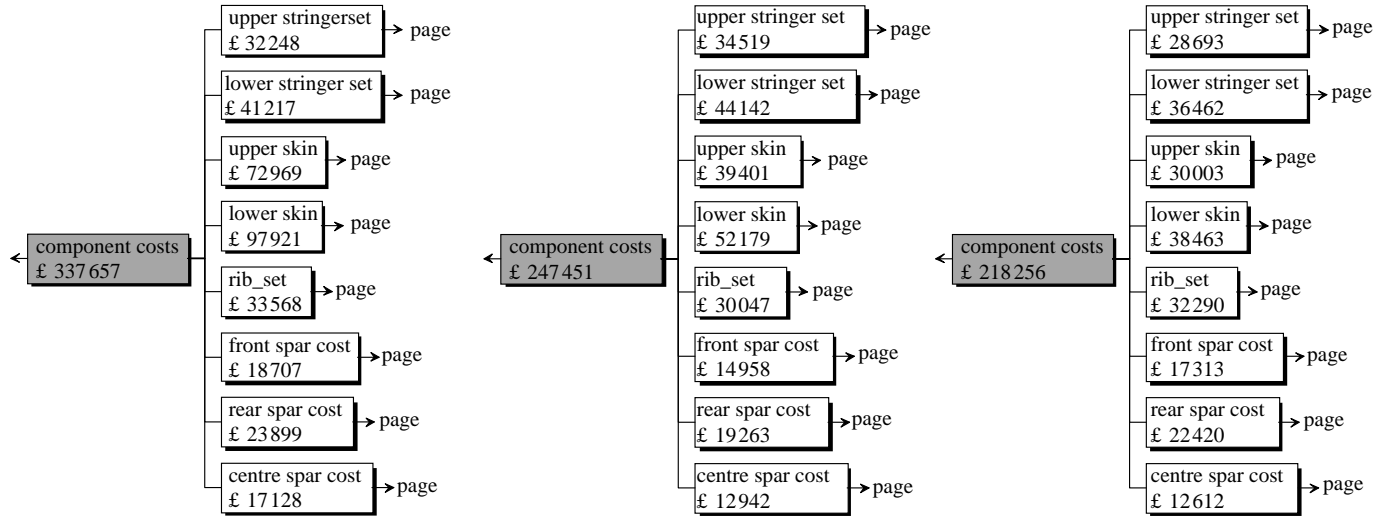


Figure 9. Wing cost breakdowns for the three designs detailed in table 4 (low drag, DoE point 157 and low-cost designs, respectively).



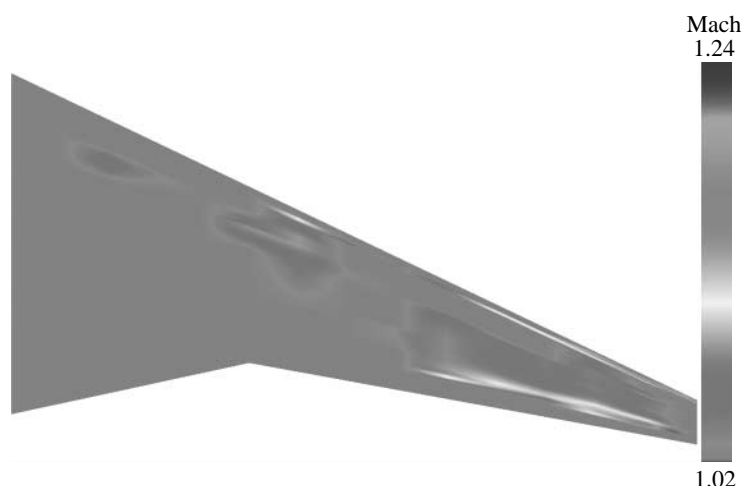


Figure 10. Wing geometry for DoE point 157 (plan view shows upper surface supersonic Mach contours to the same scale as in figure 7).

surface optimization techniques formulated using multi-objective expected improvement schemes. This allows physics-based predictions to be brought to bear when attempting to trade-off the conflicting desires for low-cost, high-performance designs.

It is fully anticipated that upcoming developments in RSM, systems integration and parallel CFD methods, coupled with the reducing cost of computational facilities will allow the routine deployment of such multi-objective approaches with tens of goals and hundreds of variables by design staff within the next 10 years.

Development of the Southampton wing design environment used here was supported by the UK Engineering and Physical Sciences Research Council under grant GR/L04733 and by BAE SYSTEMS. Their support is gratefully acknowledged.

## References

- Boggs, P. T. & Tolle, J. W. 1995 Sequential quadratic programming. *Acta Numerica* **4**, 1–51.
- Buck, R. J. & Wynn, H. P. 1993 Optimization strategies in robust engineering design and computer-aided design. *Qual. Reliab. Eng. Int.* **9**, 39–48.
- Cooke, T. A. 1973 Measurements of the boundary layer and wake of two aerofoil sections at high Reynolds numbers and subsonic mach numbers, Technical Report R&M 3722, Aircraft Research Centre.
- Cousin, J. & Metcalfe, M. 1990 The BAE Ltd Transport Aircraft Synthesis and Optimization Program, *AIAA 903 295*.
- Cressie, N. A. C. 1993 *Statistics for spatial data*. New York, NY: Wiley.
- Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. 2000 A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-ii. In *Lecture notes in computer science*, vol. 1917, pp. 848–849.
- Epstein, B., Luntz, A. & Nachson, A. 1989 Multigrid Euler solver about aircraft configurations, with Cartesian grids and local refinement, *AIAA Paper 89-1960*.
- Fonseca, C. M. & Fleming, P. J. 1995 An overview of evolutionary algorithms in multiobjective optimization. *Evol. Comput.* **3**, 1–16.
- Forrester, A. I. J., Sobester, A. & Keane, A. J. 2005 Optimization with missing data. *Proc. R. Soc. A* **462**, 935–945. (doi:10.1098/rspa.2005.1608)

- Goldberg, D. E. 1989 *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
- Harris, C. D. 1990 NASA Supercritical airfoils: a matrix of family-related airfoils, *NASA TP 2969*.
- Hooke, R. & Jeeves, T. A. 1960 *Direct search solution of numerical and statistical problems*. Westinghouse Research Labs, Scientific Paper.
- Hutchinson, M. G., Unger, E. R., Mason, W. H. & Haftka, R. T. 1994 Variable complexity aerodynamic optimization of a high speed civil transport wing. *J. Aircraft* **31**, 110–116.
- Jameson, A. 1988 Aerodynamic design via control theory. *J. Sci. Comput.* **3**, 233–260. (doi:10.1007/BF01061285)
- Jones, D. R., Schonlau, M. & Welch, W. J. 1998 Efficient global optimisation of expensive black-box functions. *J. Global Opt.* **13**, 455–492. (doi:10.1023/A:1008306431147)
- Keane, A. J. 2003 Wing optimization using design of experiment, response surface, and data fusion methods. *J. Aircraft* **40**, 741–750.
- Keane, A. J. 2004 The options design exploration system reference manual and user guide. See <http://www.soton.ac.uk/~ajk/options/welcome.html>.
- Keane, A. J. 2006 Statistical improvement criteria for use in multiobjective design optimization. *AIAA J.* **44**, 879–891.
- Keane, A. J. & Nair, P. B. 2005 *Computational approaches for aerospace design: the pursuit of excellence*. Chichester, UK: Wiley.
- Keane, A. J. & Petruzzelli, N. 2000 Aircraft Wing Design Using GA-based Multi-level Strategies, AIAA-2000-4937. In *8th AIAA/USAF/NASA/ISSMO Symp. Multidisciplinary Analysis and Optimization, Long Beach, 6–8 September*.
- Kim, S. K., Alonso, J. J. & Jameson, A. 2000 Two-dimensional high-lift aerodynamic optimization using the continuous adjoint method. In *8th AIAA/USAF/NASA/ISSMO Symp. Multidisciplinary Analysis and Optimization, AIAA Paper 2000-4741, Long Beach, CA, September*.
- Kirkpatrick, S., Gelatt Jr, C. D. & Vecchi, M. P. 1983 Optimization by simulated annealing. *Science* **220**, 671–680. (doi:10.1126/science.220.4598.671)
- Knowles, J. D. & Hughes, E. J. 2005 Multiobjective optimization on a budget of 250 evaluations. In *Evolutionary multi-criterion optimization, Third Int. Conf. EMO, Guanajuato*, vol. 3410 (eds C. A. Coello Coello, A. Hernandez Aguirre & E. Zitzler). Lecture notes in computer science, pp. 176–190. Berlin, Germany: Springer. ISBN 3-540-24983-4
- Ladson, C. L. & Brooks Jr, C. W. 1975 *Development of a computer program to obtain ordinates for naca 4-digit, 4-digit modified, 5-digit, and 16-series airfoils*. NASA TM X-3284.
- Liebeck, R. H. & Ormsbeef, A. I. 1970 Optimization of airfoils for maximum. *J. Aircraft* **7**, 409–416.
- Liu, B., Haftka, R. T., & Akgün, M. A. 1999 Two-level composite wing structural optimization using response surfaces. In *Proc. First ASMO UK/ISSMO conference on Engineering Design Optimization, Ilkley, West Yorkshire, UK, July 8–9*, pp. 19–27.
- Lock, R. C. 1986 Prediction of the drag of aerofoils and wings at high subsonic speeds. *Aeronaut. J.* **90**, 207–226.
- Mackay, M. D., Beckman, R. J. & Conover, W. J. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245. (doi:10.2307/1268522)
- Malone, J. B., Housner, J. M. & Lytle, J. K. 1999 The design of future airbreathing engine systems within an intelligent synthesis environments, ISABE Paper 99-7173. In *Proc. 14th Int. Symp. Air Breathing Engines (XIV ISABE), Florence, September* (ed. P. J. Waltrup), pp. 87.
- Mead, R. 1988 *The design of experiments*. Cambridge, UK: Cambridge University Press.
- Montgomery, D. C. 1997 *Design and analysis of experiment*, 4th edn. New York, NY: Wiley.
- Myers, R. H. & Montgomery, D. C. 1995 *Response surface methodology: process and product optimization using design of experiments*. New York, NY: Wiley.
- Panier, E. & Tits, A. L. 1993 On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Math. Program.* **59**, 261–276. (doi:10.1007/BF01581247)
- Robinson, G. M. & Keane, A. J. 2001 Concise orthogonal representation of supercritical airfoils. *J. Aircraft* **38**, 580–583.

- Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. P. 1989 Design and analysis of computer experiments (with discussion). *Stat. Sci.* **4**, 409–435.
- Scanlan, J., Rao, A., Bru, C., Hale, P. & Marshk, R. 2006 DATUM Project: cost estimating environment for support of aerospace design decision making. *J. Aircraft* **43**, 1022–1028. (doi:10.2514/1.17362)
- Schmidt, L. A. 1960 Structural synthesis. In *Proc. ASCE Structural Engineering Conf. ED-23*, 648.
- Schwefel, H. P. 1995 *Evolution and optimum seeking*. New York, NY: Wiley.
- Squire, H. B. & Young, A. D. 1937 The calculation of profile drag of aerofoils, *ARC R&M 1838*.
- Statnikov, R. B. & Matusov, J. B. 1995 *Multicriteria optimization and engineering*. London, UK: Chapman and Hall.
- Stinton, D. 1983 *The design of the aeroplane*. Oxford, UK: Blackwell Sciences.
- Torenbeek, E. 1984 *Synthesis of subsonic airplane design*. Delft; Dordrecht, The Netherlands: Delft University Press; Kluwer Academic Publishers.
- Torenbeek, E. 1992 *Development and application of a comprehensive, design sensitive weight prediction method for wing structures of transport category aircraft*. Report LR 693, Delft University.
- van Dam, C. P. & Nikfetrat, K. 1992 Accurate prediction of drag using Euler methods. *J. Aircraft* **29**, 516–519.
- Vitali, R., Haftka, R. T. & Sankar, B. V. 1999 Multifidelity design of stiffened composite panel with a crack. In *4th World Cong. Structural and Multidisciplinary Optimization. Paper 51-AAM2-1*, Buffalo, New York, NY, May 17–21.
- Voutchkov, I. I. & Keane, A. J. 2006 Multiobjective optimization using surrogates. In *Proc. 7th Int. Conf. Adaptive Computing in Design and Manufacture*, (ACDM 2006, ISBN 0-9552885-0-9), pp. 167–175. Bristol.
- Wilson, B., Cappelleri, D., Simpson, W. & Frecker, M. 2001 Efficient pareto frontier exploration using surrogate approximations. *Opt. Eng.* **2**, 31–50. (doi:10.1023/A:1011818803494)
- Wolpert, D. H. & Macready, W. G. 1997 No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82. (doi:10.1109/4235.585893)
- Zang, A. T. & Green, L. L. 1999 Multi disciplinary optimization techniques: implications and opportunities for fluid dynamic research, *AIAA 99-3798*. In *Proc. 30th AIAA Fluid Dynamics Conf.* Norfolk.