Application of generic algorithms in aerodynamic optimisation design procedures

R. P. Clayton and R. F. Martinez-Botas
Department of Mechanical Engineering
Imperial College London
London, UK

ABSTRACT

Direct optimisation techniques using different methods are presented and compared for the solution of two common flows: a two dimensional diffuser and a drag minimisation problem of a fixed area body. The methods studied are a truncated Newton algorithm (gradient method), a simplex approach (direct search method) and a genetic algorithm (stochastic method). The diffuser problem has a known solution supported by experimental data, it has one design performance measure (the pressure coefficient) and two design variables. The fixed area body also has one performance measure (the drag coefficient), but this time there are four design variables; no experimental data is available, this computation is performed to assess the speed/progression of solution.

In all cases the direct search approach (simplex method) required significantly smaller number of evaluations than the generic algorithm method. The simplest approach, the gradient method (Newton) performed equally to the simplex approach for the diffuser problem but it was unable to provide a solution to the four-variable problem of a fixed area body drag minimisation. The level of robustness obtained by the use of generic algorithm is in principle superior to the other methods, but a large price in terms of evaluations has to be paid.

NOMENCLATURE

A	area
B	diffuser blockage
C_D	drag coefficient
C_p	pressure coefficient
f_x	horizontal force (x direction)
GA	genetic algorithm
H	fixed area body half height
I	mixing length
L	diffuser length, fixed area body length
N	number of evaluations
p	pressure
R	robustness
SA	simulated annealing
T	temperature
U	velocity
W	diffuser width
$\mathbf{z}$	design vector
Π	Fitness or cost function

1.0 INTRODUCTION

Many design problems in engineering can be reduced to the specification of a vector of design variables, $\mathbf{z}$, such that a measure of the fitness or cost of the design, Π , is maximised or minimised. This is known as design optimisation, for which two distinct approaches are recognised: direct design and inverse (or indirect) design. Direct design is essentially a search method, in that designs are repeatedly proposed and assessed, via a relationship of the form $\Pi = f(\mathbf{z})$ until one is obtained which achieves the required goal. At worst this is a trial-and-error or enumerative search, while more complex search algorithms can improve convergence considerably. Indirect design is perhaps more elegant: a single step leads to the required design vector, via the inverse relationship $\Pi = f^{-1}(\mathbf{z})$. The non-existence, or at least unavailability, of such inverse relationships for many problems means that direct design is usually the only feasible approach to an optimisation problem.

In aerodynamics, $\mathbf{z}$ may describe the geometry of the problem or its boundary conditions, while Π may be a fitness function to be maximised, such as efficiency, lift or power output, or else a cost func-

Paper No. 2837. Manuscript received 31 March 2003, revised version received 19 August 2004, accepted 2 September 2004.

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

tion to be minimised, such as economic cost, weight or drag. The calculation of Π given $\mathbf{z}$ involves a flow model, perhaps the Navier-Stokes, Euler or Stokes equations, and usually calls upon a numerical technique, such as the finite volume or finite element method. The optimisation of Π is another matter, and will be discussed presently. The inverse approach, that is the calculation of $\mathbf{z}$ given Π, is possible when Π and $\mathbf{z}$ have certain physical meanings. For example, if $\mathbf{z}$ describes an aerofoil profile, and Π is the error from a prescribed pressure distribution, $\Pi = \int_A |p - p_T| \mathrm{d}A$, then a 2D potential method such as Lighthill's[1] or a 3D inviscid method such as the porous wall method[2] may be applied to minimise Π. Many inverse methods can be regarded as direct methods since, within their algorithm, they involve a series of direct solutions of the flow equations, each followed by an update of the estimated design vector. A general approach to formulating such algorithms, based on optimal control theory, has been developed by Jameson[3] and applied to inviscid external flows in two and three dimensions. The drawback to this approach is that each formulation is specific to a particular geometry and fitness or cost function.

The inverse approach may be impractical for some optimisations, for example because the problem is intractable by that route or because there is insufficient time in the design cycle to derive and code an algorithm. In this case, direct algorithms offer the only approach to the problem. In this paper, the scope for application of these methods to aerodynamic problems is discussed, which essentially means the coupling of a flow model, in this case a CFD solver, to a black-box optimiser. Two test cases are considered, one the maximisation of the pressure coefficient in a two-dimensional diffuser, the other the minimisation of profile drag on a fixed-area body. A number of direct optimisation algorithms are compared in terms of their ability to find the optimum design and their relative cost.

2.0 DIRECT OPTIMISATION ALGORITHMS

2.1 Overview

A wide variety of optimisation algorithms exist which may, in principle, couple with a CFD code. In practice, some effort may be required to achieve this end. In the following discussion, all algorithms are assumed to be equally simple to implement; they will be judged only on their potential to satisfy some performance requirements.

The simplest search algorithm is the enumerative search — a predetermined set of design vectors is evaluated and the one which minimises or maximises Π is the chosen design. The method is guaranteed to find the global optimum, provided the initial set of vectors is large enough to cover the whole design space in sufficient detail; unfortunately, it is woefully inefficient. Nevertheless, the positive characteristics of the enumerative search provide excellent criteria by which search algorithms can be judged:

- the final design should not be sensitive to the initial estimate of $\mathbf{z}$.
- the global optimum should be found; the algorithm should not be trapped by local optima.
- there should be no theoretical limit to the number of design variables.
- the design variables may be real- or integer-valued, discrete or continuous.
- equality and inequality constraints may be applied to limit the set of admissible designs.
- Π and derivatives of Π should not be required to exist at all points in the search space.

Since aerodynamic computations are usually relatively expensive, a final criterion must be added, the one which is not satisfied by the enumerative search:

- the number of evaluations of Π must be kept within practical limits.

No current search algorithm satisfies all these criteria, and some theorems suggest that it is a vain hope that one might[4,5]. The selection of a search algorithm will therefore involve a compromise, usually between speed of convergence and robustness.

2.2 Gradient methods

Classical methods[6,7], often referred to as gradient methods, offer speed of convergence. By requiring derivatives of Π to exist, and using the derivative information to direct the search up or down (depending on whether the aim is maximisation or minimisation), they can be relied upon to find local optima provided:

(a) the search space has the required degree of continuity.
(b) the dependence of Π on $\mathbf{z}$ is not excessively non-linear.
(c) the number of design variables is small.

The first and last of these violate our criteria, while the second one certainly presents a problem to aerodynamic optimisation. Chen *et al*[8] applied the GMRES algorithm, a quasi-Newton method, and a finite element Navier-Stokes solver to minimise the drag on an aerofoil in laminar flow. They found that the tendency for small perturbations of the leading edge profile to induce separation meant that the algorithm could not converge to a final design. Nevertheless, if it is believed that the search space satisfies the three provisos for gradient methods, then they may be expected to converge rapidly to a local optimum. This does not necessarily mean the nearest or best optimum in the search space, but simply the optimum within whose domain of attraction the initial design was located.

2.3 Direct search methods

Direct search methods[9,10] do not rely on gradient information, instead they seek the optimum by sampling points in the search space. Each algorithm has a unique method of choosing sampling points, but must rely on more than one sampling point, since it is otherwise impossible to judge whether the latest point is 'good' or 'bad'. Simplex methods[9] attempt to surround the optimum with a geometric entity, the simplex, whose vertices are previously sampled points. The simplex can expand, contract and translate, depending on where the optimum is judged to be. The algorithm converges when the simplex contracts to a specified size. In comparisons with gradient methods[11], simplex methods have been shown to be more robust (that is, more likely to converge on the global optimum), albeit at the expense of more fitness evaluations. They are not reliant on the continuity of Π, but require its existence. They tend to cover more of the search space than a gradient method would, hence the increased likelihood of finding the global optimum and also the increased expense, but may not perform as well as a gradient method when in the neighbourhood of the optimum. Chen *et al*[8] successfully used a simplex method to achieve a given pressure distribution over an aerofoil described by five design variables, with the flow field calculated by an incompressible Navier-Stokes solver.

At the other extreme, line searches[10] follow a single path through the search space, accepting only steps which lead to 'better' points. The algorithm converges when no further steps can be taken. These algorithms are close to gradient methods in performance, but have the advantage that they do not require gradient information. In general, direct search algorithms may be expected to show greater robustness than gradient methods, especially as the number of design variables increases, but their performance nevertheless deteriorates as more design variables are added and so they are unsuited to large-scale optimisation.

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

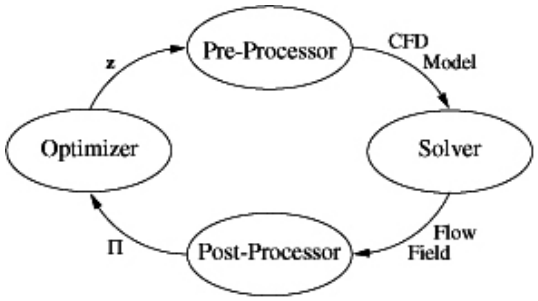


Figure 1. Example modules in a general aerodynamic optimisation system.

2.4 Stochastic methods

Stochastic optimisers — genetic algorithms (GAs) and simulated annealing (SA) — are a relatively recent development and, as well as their stochastic approach, they share the characteristic that they are developed by analogy with the physical world. Genetic algorithms[12,13] mimic natural selection in biological systems, in that the design vector **z** represents the chromosome of an 'individual'. The optimisation proceeds through stages known as 'generations', each of which involves the creation and evaluation of a population of individuals. New individuals are created by mating individuals from the preceding generation, which involves the genetic operators of selection, crossover and mutation acting on the population of design vectors. There is no proof that GAs are guaranteed to reach the global optimum but, compared to the optimisers discussed above, it appears that they are capable of finding, as Beasley *et al*[14] put it, 'acceptably good' solutions to problems 'acceptably quickly' in cases where the other optimisers fail. As one might expect, the price to be paid for this robustness is a higher number of design evaluations on average. However, the concept of generations is inherently suited to parallelisation.

Simulated annealing[15] is essentially a stochastic form of line search, obtained by analogy with the crystallisation of solids, a phenomenon chosen because it involves the minimisation of the energy of a system of molecules. The cost function is therefore analogous to energy, and the design vector analogous to the vector of molecule positions. Designs are assessed serially, the next being chosen randomly. A design closer to the optimum is accepted unconditionally, while a worse design is accepted with probability $P(T)$ where T is the 'temperature'. The probability begins near unity, but reduces as the temperature is systematically reduced. The optimisation ceases when $P(T) = 0$; a good T schedule will give the algorithm time to find the optimum before this 'freezing'. Although similarly robust to GAs, SA has not been applied as much in aerodynamic optimisation, probably because there is less scope for parallelisation and less certainty about optimum implementation, especially regarding the temperature schedule.

A number of workers have recently applied GAs to aerodynamic optimisation, almost exclusively to aerofoils in inviscid flow, either to achieve a target pressure distribution[8,16-21] or to minimise drag[16,21]. In a number of these examples, the GAs have been compared to gradient methods, with the consensus being that GAs lead to better optima, at the expense of more evaluations — typically an order of magnitude more evaluations. There do not appear to have been any applications to 3D flow, and optimisations in viscous flows are only just starting to appear, e.g. Ref. 22.

3.0 CFD-BASED BLACK-BOX OPTIMISATION

A general system for aerodynamic optimisation may consist of the following modules:

- An optimiser, such as any of the algorithms mentioned previously. This module expects to be able to obtain Π, and perhaps its gradient, for any design vector which it proposes. It is not necessary to know how Π is calculated in order to implement this module, hence the term 'black box'.
- A pre-processor, which can take a design vector and produce the corresponding CFD model.
- A flow solver, which obtains the flow field corresponding to a given CFD model.
- A post-processor, which calculates Π and ∇Π from a given flow solution.

The lines of communication between these modules are illustrated in Fig. 1. It will be seen from this that the pre- and post-processors act as a buffer between the optimiser and the solver.

Commercial CFD codes can provide the solver module and some or all of the functions of the pre- and post-processing modules; typically the user will have to write some code to link the optimiser to the CFD code's pre- and post-processors. The most significant shortcoming in a commercial CFD code is that ∇Π, where required, is not obtainable from a single flow solution. It can of course be calculated by finite differences from a perturbation of **z**. This method is simple to implement but potentially expensive if the number of design variables is high; it is also unreliable in regions of the design space where changes rapidly. Techniques for obtaining from a single flow solution are available[3,23,24], at least for some physical models, and Svenningsen *et al*[25] recently demonstrated the application of one of these techniques to a laminar flow model in the commercial code CFDS-FLOW3D.

In this paper, STAR-CD1 is used as the flow solver and pre/post-processor, and coupled with three of the aforementioned optimisation algorithms. This system is then applied to two test cases, the diffuser being a turbulent internal flow and the wedge a laminar external flow. The problems were chosen with the intention of comparing the performance of the optimisers on problems for which a solution could be obtained in reasonable time, which was taken to mean within one day on 1-5 workstations. This restriction meant that the problems were all trivial, to a certain degree, in that they were two-dimensional flows with few design variables. Larger scale optimisations are still thought to be beyond the scope of most engineering design facilities.

4.0 OPTIMISATION OF A TWO-DIMENSIONAL DIFFUSER

4.1 Problem specification

In the first test case, the aim is to maximise the static pressure recovery coefficient,

$$C_p = \frac{\overline{p_2} - \overline{p_1}}{\frac{1}{2}\rho \overline{U_1^2}}$$

of the generic two-dimensional planar diffuser illustrated in Fig. 2, for a given set of inlet conditions. The inlet duct and exit duct are assigned lengths of $1{\cdot}5W_1$ and $5W_2$ respectively, and so the geometry is completely described by any two of L/W_1, W_2/W_1 and θ. In other words, C_p is the fitness function and

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

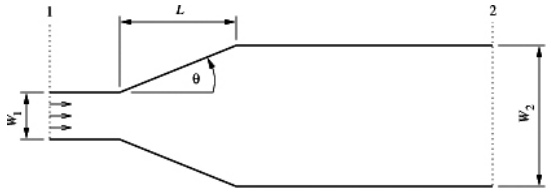


Figure 2. The geometry of a two-dimensional diffuser.

Table 1
Problem parameters for the two-dimensional diffuser

Geometry	$1{\cdot}25 \le W_2/W_1 \le 5{\cdot}00$ $2{\cdot}00 \le L_2/W_1 \le 30{\cdot}00$
Inlet conditions	$\mathrm{Re} = 10^5$ $B = 0{\cdot}003$ $I = 0{\cdot}01$

$$\mathbf{z} = \left\{ \begin{matrix} \dfrac{W_2}{W_1} \\ \dfrac{L}{W_1} \end{matrix} \right\},$$

for example, the design vector.

For the case when the inlet flow is turbulent, there is no analytical relation of the form $C_p = f(\mathbf{z})$ so the calculation of C_p for each $\mathbf{z}$ requires a numerical model. The intention of this work is to demonstrate the application of a general CFD code, but a specialised flow model such as that described by Bardina *et al*[26] could have been used. The range within which the optimum design lies, at least for realistic geometries, is known from experimental work carried out at Stanford[27] on diffusers without tailpipes. The performance maps for conical diffusers[28] suggest that the tailpipe does not move the optimum significantly; there is, however, a lack of data with which to construct the complete performance map for a planar diffuser with tailpipe, as noted in Ref. 29.

The parameters of this study are listed in Table 1. The range of geometries permitted is known to bound the optimum in reality, and the inlet conditions are intended to typify an incompressible, turbulent flow with thin boundary layers.

4.2 The CFD model

The calculations were carried out using the commercial CFD code STAR-CD2. A mesh of 400 × 25 cells (streamwise **x** normal) was created, taking advantage of the centre-line symmetry, and scaled at each design evaluation according to **z**. Since the flow was turbulent, the mesh density was increased close to the wall to resolve the boundary layer there, the nearest wall node being $W/100$ from the wall.

The inlet boundary conditions were prescribed using the data in Table 1. The velocity profile was specified using a 1/7th power law in the boundary layer, whose thickness was obtained from the given blockage, B. The outlet condition was one of zero gradient in the dependent variables, and the no-slip condition was applied at the walls. The RNG κ–ε turbulence model[30,31] was used, coupled with wall functions at the near-wall cells. Lai *et al*[32] compared the performance of a number of turbulence models in the prediction of the flow in a planar diffuser; they found that the standard κ–ε model, which is similar to the RNG variant used here, was capable of predicting the behaviour of C_p correctly, but could not reproduce all the features of the velocity field. This is in agreement with the findings of Ganesan and co-workers[33,34], whose experimental measurements were used to partially validate the CFD model used here.

Table 2
Parameters for the two-dimensional diffuser validation case

Geometry	$W_2/W_1 = 2{\cdot}1267$ $L/W_1 = 7{\cdot}9$
Inlet conditions	$\mathrm{Re} = 1{\cdot}5 \times 10^5$ $B = 0{\cdot}002$ (at throat) $I = 0{\cdot}005$

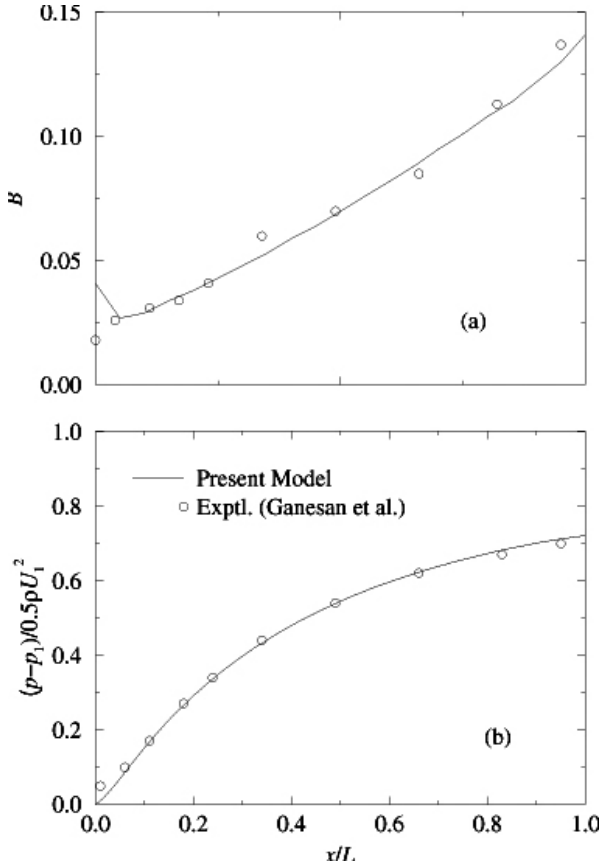


Figure 3. Comparison of predictions and measurements of (a) blockage and (b) centre-line pressure for the two-dimensional diffuser validation case.

The parameters of the diffuser reported in Refs 33 and 34 are listed in Table 2; the diffuser was intended to be without appreciable stall, and appeared to achieve this aim. The parameters listed were implemented in the parametric diffuser model described above and, in Fig. 3, the calculated blockage and centre-line pressure along the length of the diffuser are compared with the measurements. It appears that the agreement is sufficient to give confidence in the prediction of C_p which is, of course, strongly dependent on the accuracy of the pressure-field calculation. The abberation in the calculated blockage at the diffuser throat is thought to be a consequence

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

of the the square throat corner used in the current model.

However, it is difficult to validate a parametric model for the whole of its design space, and it must be said that the diffuser of Ganesan *et al* provides favourable conditions (moderate pressure gradient, no separation). A particular problem with diffuser flows is that transitory stall is seen over a large area of the design space, a feature which cannot be predicted with a high Reynolds number turbulence model. Moreover, the stall may be asymmetric, and so will not be reproduced using symmetric geometry and boundary conditions. In areas of the design space where transitory stall is expected to occur, the CFD model is likely to over-predict C_p. Since the error in C_p is not constant over the design space, it would be possible for the calculated optimum design not to coincide with the true optimum. This does not appear to be the case for the model used here, although this may be due in part to the fact that the optimum design is not known exactly.

Finally, the self-filtered central differencing scheme[35], a bounded blend of first- and second-order upwinding, was applied to both velocity and turbulence variables and central differencing applied to the pressure. The flow equations were solved using the SIMPLE algorithm. A typical run without utilising restart data required 30 minutes CPU time on a DEC3000/600 workstation.

4.3 The optimisation algorithms

Three optimisation algorithms were tested on this problem: a gradient method, a direct search method and a stochastic method. The gradient method was a truncated Newton algorithm[36], with gradient information obtained by finite differences around the current design point. In order to ensure success with this algorithm, it was found necessary to calculate the gradient using relatively large perturbations — of the order of a thousandth of the design space extent in each direction — and to solve the flow-field to an order of magnitude lower level of residual than with the other optimisers. These steps, together with an estimate of the repeatability of the C_p calculations, prevented the optimisation being disrupted by fine-scale noise.

The direct search algorithm was the simplex method from Press *et al*[37] with the bound constraints applied by a one-step interior penalty function[7].

The stochastic optimiser was an implementation of a genetic algorithm, the so-called micro-GA of Krishnakumar[38], which is designed to work with small populations (in this case, five individuals per generation). The micro-GA is attractive because the cost of each design evaluation is high, and hence the number of design evaluations must be minimised. The small population means that the population converges rapidly on sub-optimal designs. When this occurs, the population is reinitialised; the best design from the previous generation is unconditionally retained from one generation to the next. The probability of crossover and mutation were set at 0·5 and 0·0 respectively, and the selection of individuals for mating was by a deterministic tournament strategy. The design vector was coded as a binary string, in other words only a finite number of designs were considered. Both L/W_1 and W_1/W_2 were given 256 possible values each, giving 2^{16} possible designs. Other workers[16] have demonstrated aerodynamic optimisation using real-coded design variables (that is, an infinity of possible designs), but there is evidence[17] that local, fine-scale optimisation is better achieved using a gradient method from a converged coarse-scale GA solution. There is no indication when optimality has been attained with a micro-GA, since convergence to a sub-optimal design is highly likely. The algorithm must therefore run for as many generations as can be afforded. In this case, 50 were chosen, requiring a maximum of 201 design evaluations. Individuals carried over from a previous generation were not re-evaluated.

All three optimisers were started from a random starting point or points.

Table 3
Optimum design vectors for the two-dimensional diffuser

	W_2/W_1	L/W_1
Newton Algorithm	5·00	22·94
Simplex Algorithm	4·99	23·88
Genetic Algorithm	5·00	23·85
Reneau *et al*[27]	3·8-5·0	17-25

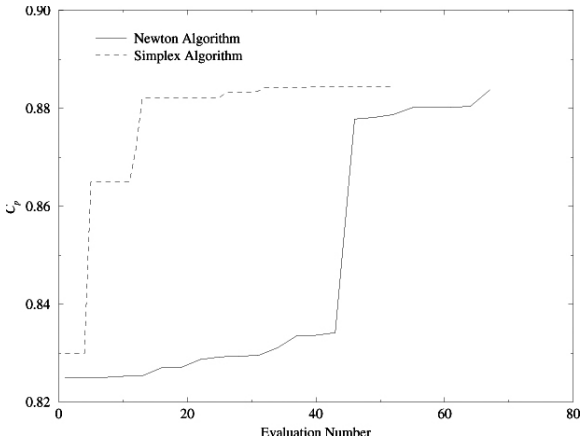


Figure 4. The evolution of C_p in the diffuser optimisation.

4.4 Results

All three optimisers achieved a maximum pressure-recovery coefficient of 0·884, although with slightly differing design vectors, as summarised in Table 3. The location of these computed optima is in line with the range indicated by Reneau *et al*[27].

The evolution of C_p for the Newton and simplex algorithms is plotted in Fig. 4. This shows that the simplex algorithm was within two significant figures of the optimum within 20 evaluations, but then found further improvement difficult, whereas the Newton algorithm produced more consistent progress; the sudden increase in C_p shown by the Newton algorithm coincided with a change of direction in the line search. The simplex algorithm required 52 iterations to converge on an optimum, whereas the Newton algorithm required 69, including gradient evaluations. However, since two out of every three evaluations for the Newton algorithm were for the gradient calculations, and were therefore only perturbations of previous designs, the Newton algorithm required less time to converge than the simplex, perhaps by 20%. The GA began with an initial guess of $C_p = 0{\cdot}88$, so little progress could be made from one generation to the next. In fact, $C_p > 0{\cdot}85$ in 30-40% of the design space, making it highly probably that one of the randomly-chosen initial designs would be close in performance to the optimum. This is a stringent test of the GA, since progress in the neighbourhood of an optimum is, as previously stated, difficult.

Figure 5 shows a scatter plot of designs evaluated during the run, clearly demonstrating that the GA biased attention towards the region surrounding the optimum, while coarsely exploring the rest of the design space. The simplex algorithm, on the other hand, followed a more direct route to the optimum, albeit with some designs out of bounds. The Newton algorithm followed the most direct route of all. In a case such as this one, when there is only a single optimum, such directness is an advantage as it minimises the number of design evaluations required.

As implemented here, the GA and the classical solvers are stochastic, the former inherently and the latter by their random starting point or points. For this reason, rigorous conclusions about

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

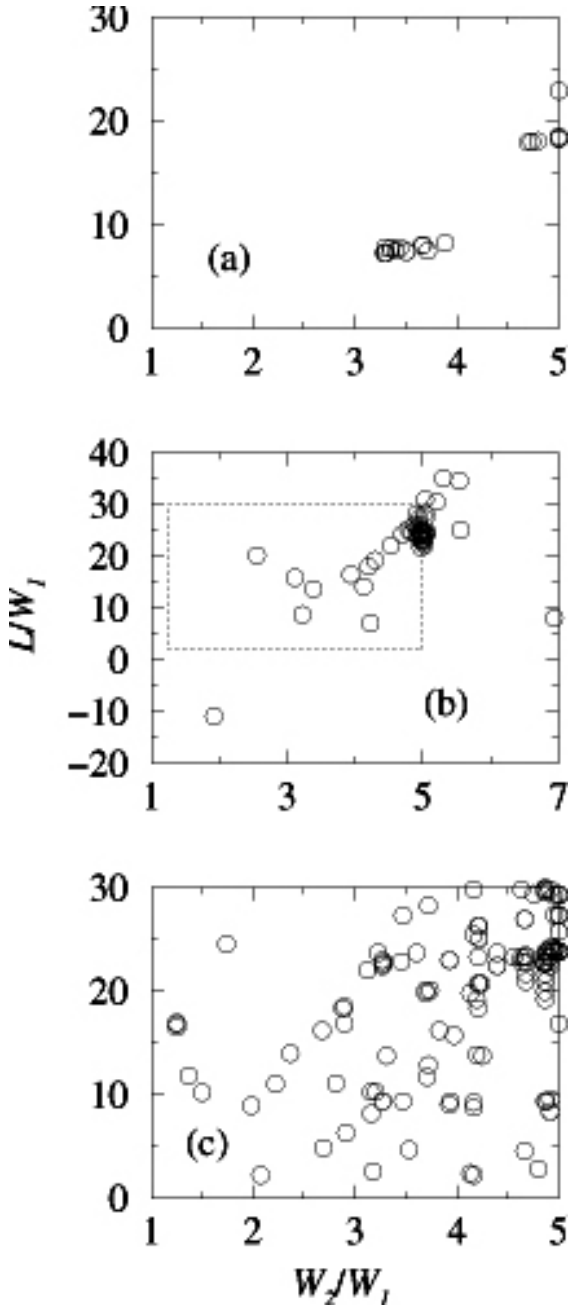


Figure 5. The distribution of designs evaluated by (a) the classical algorithms and (b) the genetic algorithm.

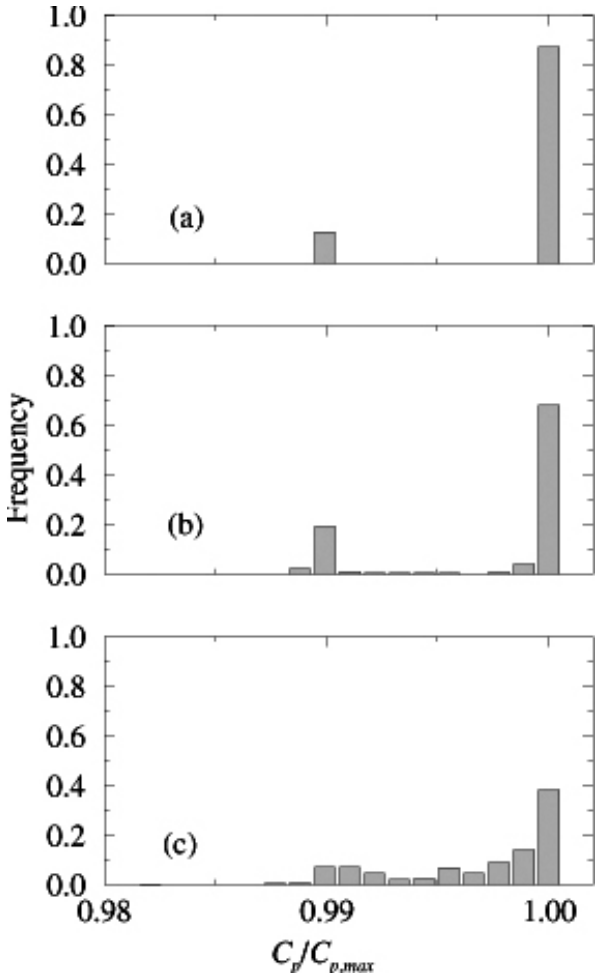


Figure 6. The frequency distribution of C_p calculated for the diffuser problem by (a) the GA, (b) the simplex method and (c) the gradient method.

their performance should be based on statistics from a sample of runs, rather than a single realisation. The results of the previous GA run were used for this purpose — a polynomial surface was fitted to C_p which could then be evaluated quickly rather than recalculated using the CFD model, thus allowing many realisations of the optimisation to be carried out. The frequency distribution of for 1,000 realisations of the GA and classical solvers is plotted in Fig. 6. All the optimisers converge with a probability greater than 0·99 on a design with C_p greater than 99% of the maximum, so they may be regarded as equally robust, at least from an engineering viewpoint. Van der Velden[11] attempted to quantify the robustness of optimisation schemes as

$$R = \frac{1}{N}\sum_{i=1}^{N}\frac{C_{p,i}}{C_{p,\max}}$$

for N realisations. In this case, R is 0·9967 for the GA, 0·9966 for the simplex method and 0·9988 for the gradient method. The similarity of these three values supports the preceding observation.

There is a little difference in the number of evaluations required by the two classical methods, as Fig. 7 demonstrates; the GA was allowed 50 generations in each realisation. Both the gradient method and the simplex method required a modal number of about 50 evaluations, compared to 150-200 for the GA. This is a consequence of the fact that the polynomial C_p is a continuous, unimodal function of two variables, an application in which, as mentioned earlier, classical methods have the advantage.

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

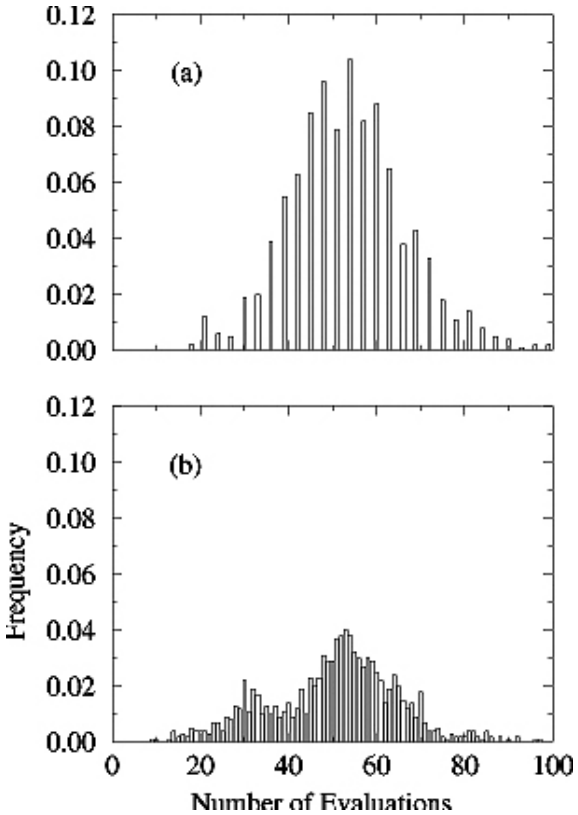


Figure 7. The frequency distribution of evaluations required for the diffuser problem by (a) the simplex method and (b) the gradient method.

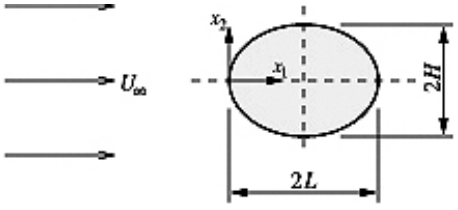


Figure 8. The geometry of the fixed-area body.

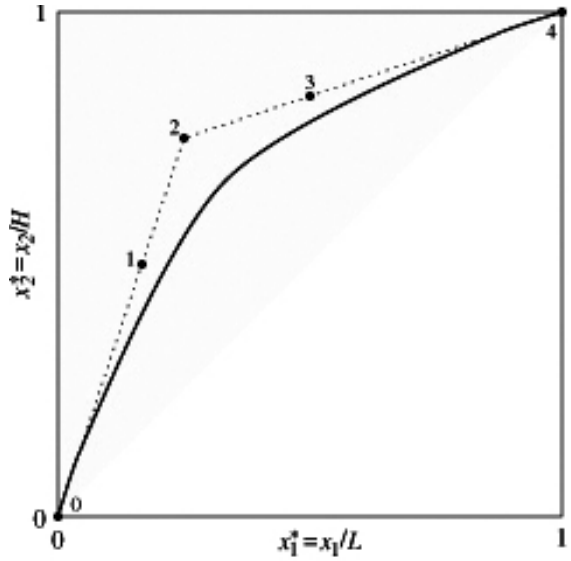


Figure 9. The fourth-order Bezier curve representation of the body profile.

5.0 DRAG MINIMISATION OF A FIXED-AREA BODY

5.1 Problem specification

The second test case concerns the minimisation of the profile drag of a two-dimensional body in laminar flow. The body is constrained to have a given length and area, and also has bilateral symmetry, as depicted in Fig. 8. The cost function to be minimised is defined as

$$C_D = \frac{f_x}{\frac{1}{2}\rho U_\infty^2 H}.$$

Here, f_x is the horizontal component of the force on the body, given by

$$f = \int_A (-pI + 2\mu D) \bullet n \mathrm{d}A$$

with D the deformation tensor. In order to optimise the shape of the body, it is necessary to describe its profile somehow; in fact only a quarter of it, say $0 \le x \le L$, $0 \le h \le H$, need be considered because its symmetry. It is impractical to have the infinity of possible profiles as the design space, so the space must be reduced by selecting a family of curves which is likely to encompass, or at least approximate, the optimum design. Such families of curves may include B-splines, Bezier curves, polynomials and trigonometric series, but the suitability of each depends on the requirements of the particular problem. It is always preferable to minimise the number of variables and constraints in an optimisation, as this makes the problem less stiff; this precludes the use of some curve families, with which it is difficult to exclude, for example, profiles with negative thickness or concavity (if convexity is thought to be essential). Bezier curves (e.g. Ref. 16) and B-splines (e.g. Ref. 21) are the most common choice, and a Bezier curve was chosen here.

It was assumed that a convex, continuous profile would be suitable to represent the optimum design, and that an assumption of zero curvature at the end-points would not be excessively restrictive. These assumptions allowed the Bezier representation illustrated in Fig. 9 to be used, which has four independent parameters ($\mathbf{z_1}$-$\mathbf{z_4}$) — two describing the position of point 2, restricted to the shaded area to ensure covexity, and one each for points 1 and 3. Since L and A are fixed, the value of H is obtained by integration of the profile. The complete curve description is

$$x^*(u) = \sum_{i=0}^{4} \binom{4}{i} u^i (1-u)^{4-i} x_i^*$$

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

Table 4
Problem parameters for the drag minimisation problem

Geometry	$0{\cdot}0 \le \mathbf{z_3} \le 1{\cdot}0$
	$0{\cdot}0 \le \mathbf{z_4} \le 1{\cdot}0$
	$L = 1\text{m}$
	$A = 2\text{m}^2$
Inlet conditions	$\text{Re} = 10$
	$M = 10^{-7}$

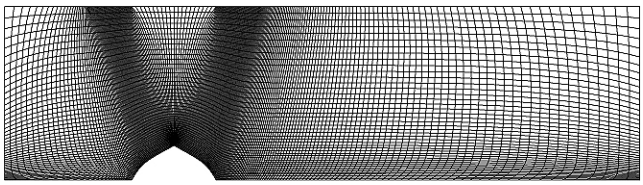


Figure 10. A typical CFD mesh for the drag-minimisation problem.

where $0 \le u \le 1$. The five control points x_i^* are given by

$$x_0^* = 0$$
$$x_1^* = z_3 x_2^*$$
$$x_2^* = \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix}$$
$$x_3^* = (1 - z_4)x_2^* + z_4 x_4^*$$
$$x_4^* = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}.$$

The flow conditions for the optimisation are summarised in Table 4. The Reynolds number is sufficiently small to preclude vortex shedding, and hence the non-existence of steady-state solutions, while high enough to make the inertia term of the momentum equations significant.

5.2 The CFD model

The calculations were, again, carried out using the commercial CFD code STAR-CD. The mesh, typified by the one in Fig. 10, consisted of 10,000 cells, of which a large number were packed close to the wall in order to resolve geometry changes — the boundary layer was, of course, laminar and would not have been difficult to resolve even with fewer cells in the near-wall region. The boundary conditions were as follows:

(a) Fixed velocity at inlet.
(b) Zero gradient of velocity and pressure at outlet.
(c) No-slip condition at the surface of the wedge.
(d) All other edges of the domain as symmetry axes.

The flow equations were solved using the SIMPLE algorithm, with the QUICK differencing scheme applied to the velocity variables and central differencing to the pressure.

5.3 The optimisation algorithms

Two optimisation algorithms were applied to this test case: the micro-GA and the simplex algorithm, both in essentially the same way as they were applied to the diffuser. The gradient method was not applied here because it proved unsuccessful on the first test case. For the GA, the only difference in implementation was in the coding of the design variables. Each of the four variables was coded as a six-bit binary string, giving 2^{24} possible designs. The simplex method was again implemented with an interior penalty function. Both optimisers started from the same random set of five designs.

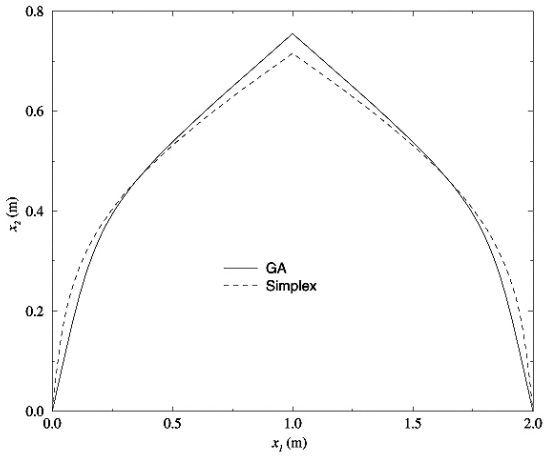


Figure 11. Final body profiles found by drag minimisation.

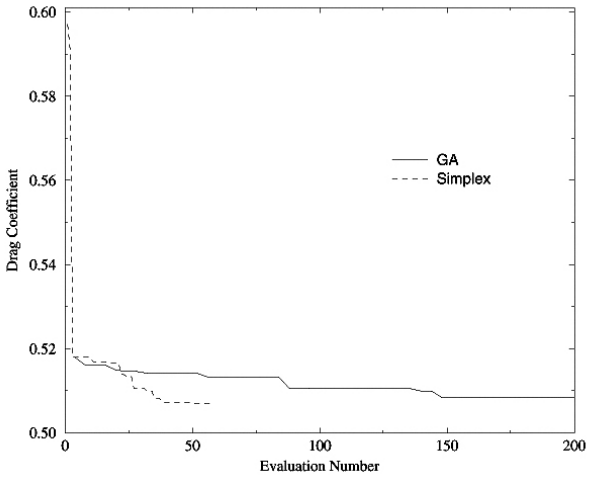


Figure 12. Optimisation history for the drag minimisation problem.

5.4 Results

Both the GA and the simplex algorithm managed to reduce C_D to approximately 0·5. In the case of the GA the final was 0·508, while the simplex improved this slightly to 0·507. In both cases, this represented an improvement of approximately 15% over the initial design. The two final designs differed as shown in Fig. 11. Although similar, the GA design has a more streamlined leading edge, and hence higher profile maintain constant area. The fact that both methods achieved similar final cost functions with different designs suggests that either there is a volume of the design space over which the drag coefficient varies little or the design space is multi-modal.

Figure 12 compares the rate of decrease in C_D achieved by the two optimisers. As with the diffuser case, the simplex method is found to need only 25-35% of the number of evaluations needed by the GA.

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

In this example, however, the GA produces a more constant improvement in the cost function than in the diffuser case, so that one may regard more of the evaluations as producing an improved design. A measure of efficiency based on this fact would suggest that the GA is more suited to this higher-dimensional problem than it was to the diffuser case. Nevertheless, the simplex algorithm outperforms the GA on this test case, having produced a better optimum in fewer evaluations. On a DEC3000/600 workstation, the optimum was found within one day, even though the initial simplex was a random one.

6.0 CONCLUSIONS

The two test cases presented here involved the optimisation of a single performance measure which was a function of two or four design variables. The flow calculations were carried out by a Navier-Stokes solver, with turbulent flow in the first case and laminar in the second. For both problems, a black-box optimiser was identified which could, on a single workstation, obtain the optimum design within a single day. Evidence from elsewhere suggests that no single optimiser is suited to all problems, the two principle factors affecting the performance of the optimiser being the number of design variables and the number of local optima contained in the design space. A global optimiser, such as the genetic algorithm, is likely to be better suited to large-scale problems, but computational requirements mean that such calculations are currently at the limit of practicability. In the small test cases presented here, the simplex algorithm was found to suffice, and indeed required less than 40% of the number of evaluations required by the GA.

In the first flow case, the aim is to maximise the static pressure recovery coefficient, all three optimisers achieved maximum value of 0·884. The evaluation plot demonstrated the GA biased attention towards the regions surrounding the optimum. The simplex method followed a more direct route to the maximum, while the simplest of all method was the most direct solution evolution. In terms of number of evaluations, the simplex and Newton methods converged in approximately the same number (50 evaluations), the GA on the other hand took over three times more evaluations. This is due to the fact that the polynomial C_p is a continuous unimodal function of two variables, in which classical methods have the advantage.

The second test (minimisation of the profile drag of a two dimensional body in laminar flow) both the simplex and GA approach managed to reduce the C_D by 15% of the initial design (the Newton gradient method was unable to provide a solution). The simplex method was found to need only 25-35% of the number of evaluation needed by the GA. Nevertheless, the GA produced a more constant improvement in the cost function than in the diffuser case; this suggest that GA is more suited to this higher dimensional problem.

The extension of these optimisation techniques to 3D turbulent flows is trivial, at least from the point of view of implementation. However, two facts should be borne in mind. Firstly, 3D calculations are expensive and tend to involve geometries described by more design variables than their 2D equivalents. Secondly, the optimisation is only as accurate as the underlying physical model which, for a complex turbulent flow modeled with, say, a κ–ε model, is not ideal. Any computation is only worthwhile when the usefulness of its results outweighs the expense in obtaining them.

REFERENCES

1. LIGHTHILL, J.M. A new method of two-dimensional aerodynamic design, 1945, ARC R&M 2112.
2. DEMEULENAERE, A. and VAN DEN BRAEMBUSSCHE, R. Three-dimensional inverse design method for turbomachinery blading design, 1996, ASME Paper No. 96-GT-39.
3. JAMESON, A. Optimum aerodynamic design via boundary control, Optimum design methods for aerodynamics, 1994, AGARD Rep 803.
4. CULBERSON, J.C. On the futility of blind search, 1996, Technical Report TR-96-18, Dept of Computer Science, University of Alberta.
5. WOLPERT, D.H. and MACREADY, W.G. No free lunch theorems for search, 1996, Technical Report SFI-TR-95-02-010, Santa Fe Institute.
6. ARORA, J.S. *Introduction to Optimum Design*, 1989, McGraw-Hill, New York.
7. VANDERPLAATS, G.N. *Numerical optimization techniques for engineering design: with applications*, 1984, McGraw-Hill, New York.
8. CHEN, H.Q., GLOWINSKI, R., HE, J.W., KEARSLEY, A.J., PÉRIAUX, J. and PIRONNEAU, O. Remarks on optimal shape design problems, 1994, *Frontiers of Computational Fluid Dynamics*, CAUGHEY, D.A. and HAFEZ, M.M. (Eds), Ch 5, pp 67-80, Wiley, Chichester.
9. NELDER, J.A. and MEAD, R. Downhill simplex method in multidimensions, *Comp J*, 1965, **7**, pp 308-313.
10. HOOKE, R. and JEEVES, T.A. Direct search solution of numerical and statistical problems, *J ACM*, 1961, **8**, pp 212-229.
11. VAN DER VELDEN, A. Tools for applied engineering optimization, 1994, *Optimum Design Methods for Aerodynamics*, Ch 8, AGARD Rep 803.
12. HOLLAND, J.H. *Adaption in Natural and Artificial Systems*, 1975, Univ Michigan Press, Ann Arbor.
13. GOLDBERG, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1988, Addison-Wesley, Reading, MA.
14. BEASLEY, D., BULL, D.R. and MARTIN, R.R. An overview of genetic algorithms: Part 1, *Fundamentals Uni Comp*, 1993, **15**, (2), pp 58-69.
15. KIRKPATRICK, S., GELATT, C.D. and VECCHI, M.P. Optimization by simulated annealing, *Science*, 1983, **220**, (4598), pp 671-680.
16. PÉRIAUX, J., SEFRIOUI, M., STOUFFLET, B. and MANTEL, B. *Robust Genetic Algorithms for Optimization Problems in Aerodynamic Design*, 1995, Ch 19, pp 371-396. In WINTER *et al*[39].
17. POLONI, C. Hybrid GA for Multi Objective Aerodynamic Shape Optimization, 1995, chapter 20, pages 397-415. In WINTER *et al*[39].
18. DOORLY, D. *Parallel Genetic Algorithms for Optimization in CFD*, 1995, Ch 13, pp 251-270. In WINTER *et al*[39].
19. QUAGLIARELLA, D. and CIOPPA, A.D. Genetic algorithms applied to the aerodynamic design of transonic airfoils. *J Aircr*, 1995, **32**, pp 889-891.
20. OBAYASHI S. and TAKANASHI, S. Genetic optimization of target pressure distributions for inverse design methods, 1995, Paper AIAA-95-1649-CP.
21. QUAGLIARELLA, D. *Genetic Algorithms Applications in Computational Fluid Dynamics*, 1995, Ch 21, pp 417-442, In WINTER *et al*[39].
22. YAMAMOTO K. and INOUE, O. Applications of genetic algorithm to aerodynamic shape optimization, 1995, Paper AIAA-95-1650-CP.
23. BAYSAL O. and ELESHAKY, M.E. Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. *AIAA J*, 1992, **30**, pp 718-725.
24. BORGGAARD J. and BURNS, J. A sensitivity equation approach to shape optimization in fluid flows, *Flow Control*, 1995, GUNZBURGER, M. (Ed), pp 49-78, Springer-Verlag.
25. SVENNINGSEN, K.H., MADSEN, J.I. HASSING, N.H. and PÄUKER, W.H.G. Optimization of flow geometries applying quasianalytical sensitivity analysis, *Appl Math Modelling*, 1996, **20**, pp 214-224.
26. BARDINA, J., LYRIO, A., KLINE, S.J., FERZIGER, J.H. and JOHNSTON, J.P. A prediction method for planar diffuser flows, Trans ASME, *J Fluids Eng*, 1981, **103**, pp 315-321.
27. RENEAU, L.R., JOHNSTON, J.P. and KLINE, S.J. Performance and design of straight, two-dimensional diffusers. Trans ASME, *J Basic Eng*, 1967, pp 141-150.
28. WARD-SMITH, A.J. *Internal Fluid Flow: The Fluid Dynamics of Flow in Pipes and Ducts*, 1980, Clarendon, Oxford.
29. ESDU, London. Performance in incompressible flow of plane-walled diffusers with single-plane expansion, 1974, data sheet 74015 edition.
30. YAKHOT V. and ORSZAG, S.A. Renormalization group analysis of turbulence, I. Basic theory, *J Sci Comp*, 1986, **1**, (1), pp 3-51.
31. YAKHOT, V. ORSZAG, S.A. THANGAM, S. GATSKI, T.B. and SPEZIALE, C.G. Development of turbulence models for shear flows by a double expansion technique, *Phys Fluids A*, 1992, **4**, (7), pp 1510-1520.
32. LAI, Y.G., SO, R.M.C. and HWANG, B.C. Calculation of planar and conical diffuser flows, *AIAA J*, 1989, 27, (5), pp 542-548.
33. GANESAN, V., SUZUKI, K., NARAYANA, P.A.A. and SWAMY, N.V.C. Numerical prediction of boundary-layer development in a two-dimensional diffuser. *J Inst Energy*, 1988, **LXI**, pp 192-200.
34. GANESAN, V., SUZUKI, K., NARAYANA and CHITHAMBARAN, V.K. Investigations of mean and turbulent flow characteristics of a two dimensional plane diffuser, *Expts in Fluids*, 1991, **10**, (4), pp 205-212.
35. ZIMAN, H.J. Computer Prediction of Chemically Reacting Flows in Stirred Tanks, 1990, PhD thesis, University of London.

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press

36. NASH, S.G. Newton-type minimisation via the Lanczos method, 1984, *SIAM J Numer Anal*, **21**, pp 770-778.
37. PRESS, W.H. FLANNERY, B.P. TEUKOLSKY, S.A. and VETTERLING, W.T. *Numerical Recipes in C: The Art of Scientific Computing*, 1993, 2nd edition, CUP, Cambridge.
38. KRISHNAKUMAR, K. Micro-genetic algorithms for stationary and non-stationary function optimization, *SPIE*, 1989, **1196**, Intelligent Control and Adaptive Systems, pp 289-296.
39. WINTER, G., PÉRIAUX, J., GALÀN, M. and CUESTRA, P. (Eds), *Genetic Algorithms in Engineering and Computer Science*, 1995, Wiley, Chichester.

https://doi.org/10.1017/S0001924000000440 Published online by Cambridge University Press