Available online at www.sciencedirect.com

SCIENCE DIRECT®

Control Engineering Practice 12 (2004) 475–490

CONTROL ENGINEERING PRACTICE

www.elsevier.com/locate/conengprac

Global optimisation-based control algorithms applied to boundary layer transition problems

G.V. Veres[a], O.R. Tutty[a], E. Rogers[b,*], P.A. Nelson[c]

[a] *School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ, UK*
[b] *Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*
[c] *Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, UK*

Received 27 January 2003; received in revised form 29 September 2003; accepted 30 September 2003

Abstract

Turbulent flow has a significantly higher drag than the corresponding laminar flow at the same flow conditions. The presence of turbulent flow over a large part of an aircraft therefore incurs a significant penalty of increased fuel consumption due to the extra thrust required. One possible way of decreasing the drag is to apply surface suction to delay the transition from laminar to turbulent flow. However, in order for the gain from the reduction in drag to outweigh the extra costs associated with the suction system, the suction must be distributed in an optimum, or near optimum, manner. In this paper two practical cases are considered. In the first of these a flat plate with panels whose positions are adjustable but do not overlap is treated. Since the cost function in this case is multi-modal, non-smooth and non-convex, methods for solving the optimisation problems necessary to design multi-panel suction systems based on direct search techniques are developed. In the second case considered the problem is that of linear distributed suction over the front part of an aerofoil. For this case, the computational load increases so significantly that in some cases it is not really feasible to continue the investigation using a single processor code. To overcome this, three parallel global optimisation algorithms are developed for the design of multi-panel suction systems on an aerofoil and it is shown that good solutions can be found efficiently.

Keywords: Boundary layer transition control; Flow control; Aerofoils; Non-smooth direct optimisation; Parallel global optimisation

1. Introduction

In recent years a number of ways have been considered for drag reduction and hence reduced operating costs for civil aircraft due to a decrease in the fuel consumption (see Thiede (2001) for a summary of some recent work in this area). It is well established that suction applied to surface of a body can delay the onset of transition from laminar to turbulent flow and thereby reduce the overall drag. Consequently, surface suction is regarded as one of the most promising methods for drag reduction. However, in order to maximise the benefit from the applied suction, it is necessary to distribute the suction in an optimum, or near optimum, way.

In the current work, the approach consists, in effect, of monitoring the state of the flow together with the automatic control of suction applied through the surface of the body. This program involves both algorithm development and verification and wind tunnel based experiments. In the case of the latter, an essential element has been the development of techniques for monitoring the position of transition in a boundary layer. Experimentally, pressure measurements are used to determine the position of transition, and a control law is then used to maintain transition at a desired location with minimum power consumption by controlling the suction flow rates on suction panels.

Previous work has demonstrated that this overall strategy is technically feasible both in simulation studies (of which computational fluid dynamics is an essential part) (Tutty, Hackenberg, & Nelson, 2000a, b) and by wind tunnel-based experiments (Nelson, Wright, & Rioual, 1997; Wright & Nelson, 1999). This work employed gradient-based controllers, which used a local linear model constructed by standard linear system identification tools around selected operating points. For the case of flow over a flat plate these controllers

*Corresponding author. Tel.: +44-2380-592197; fax: +44-2380-594498.
E-mail address: etar@ecs.soton.ac.uk (E. Rogers).

0967-0661/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2003.09.009

performed well. In somewhat more complicated cases, however, they did not, e.g. when a non-zero pressure gradient is applied to a flat plate (Tutty et al., 2000b).

In MacCormack, Tutty, Rogers, and Nelson (2002) it was shown that the gradient-based approach failed when it was incompatible with the basic physics of the flow. An alternative approach, based on the use of Simulated Annealing and Genetic Algorithm methods to solve a non-linear optimisation problem, was presented in Dodd, Tutty, and Rogers (2001), MacCormack et al. (2002). However, although this approach was successful, computationally it is very demanding (and hence the time taken to obtain a satisfactory design may be (in relative terms) too long).

This paper first reports new results on the development of optimisation-based approaches for solving the design problem for the generic case of flow past a flat plate where the positions of the panels are calculated during the optimisation process, which for ease of terminology here we will term ‘moving panels’. The key point here is to include the panel positions as parameters in the optimisation task and hence obtain their best possible locations for minimising the cost function. This is a major advance on the previous work where we were optimising the suction flow rates for pre-positioned panels. A second new problem treated here is for an airfoil as opposed to panels where the computational cost will be (in relative terms) higher and the amount of time needed to complete the necessary design run may well be excessive (again in relative terms). Hence we also develop parallel versions of the algorithms for this case.

There is no claim that the proposed methods are always preferable to all other methods (in particular, the development of parallel algorithms should not be read as implying that the computations cannot be done sequentially but merely that they may release a designers time for other tasks). Rather, the choice of optimisation technique to be used will be determined by the demands of the application problem or/and the preferences of user (but this, of course, requires a range of algorithms with known properties to be available). Note however that the speed of the algorithm is a critical feature if such methods are to be incorporated into a design process with practical constraints, when it may be necessary to repeat the calculation many times. The optimisation of the suction distribution under fixed flow conditions is a problem of considerable practical significance as this reflects cruise conditions for transport aircraft.

In the next section, we give the necessary background results. To provide context and a validated test case, the following section gives a description and some typical results from experiments performed for a flat plate with two suction panels. This is followed by a description of the models, cost functions, and algorithms used in this work. Representative results are given and discussed. Previous work (Tutty et al., 2000a, b) has shown that the models developed below can be used to formulate a problem whose solution has good qualitative agreement with those from the experiments, including the dynamic behaviour under a control law.

2. Background

Consider a flat plate with a sharp leading edge aligned with a uniform incoming flow. Over most of the flow field, the effect of the viscosity of the fluid is negligible, and the flow will remain uniform. However, near the plate, the velocity must decrease from the free stream velocity to zero at the surface of the plate, and in this region, the ‘boundary layer’, viscous effects play a significant role. The width of the boundary layer is small compared with the length of the plate, but it increases along the plate, as $x^{1/2}$, where x is the distance along the plate from the leading edge, with the gradient of the velocity decreasing as $x^{-1/2}$.

Suppose now that small disturbances are introduced into the flow in the boundary layer. Near the leading edge, for $x < x_0$, where

$$x_0 \approx 6 \times 10^4 v / U_\infty \tag{1}$$

and U_∞ and v are the free stream velocity and kinematic viscosity of the fluid, respectively, these disturbances decay in amplitude, and the flow is stable. However, for $x > x_0$ the disturbances will grow, and the flow is unstable. These disturbances, known as Tollmien–Schlichting waves, initially grow relatively slowly, but further downstream will provoke much more vigorous disturbances which rapidly lead to transition and a fully turbulent flow.

One of the primary characteristics of turbulent flow is enhanced mixing due to the rotational character of the small scale motions in the flow. In particular, in a turbulent boundary layer, the turbulence transports momentum from (the higher mean velocity) outer flow towards the wall and increases the velocity near the wall compared to that found in a laminar (non-turbulent) flow in similar conditions. This increases the velocity gradient at the wall, and since the shear stress (the frictional force) is proportional to the velocity gradient, the shear stress increases drastically, by at least an order of magnitude, at transition.

The scenario presented above is for a flat plate, but will apply to any body which is basically aligned with the flow. Consequently, if the flow remains laminar, the drag on the body due to the skin friction will be much less than that if the flow is turbulent. Hence, there is great interest in the aerospace industry in delaying transition, or modifying the turbulent flow once transition has occurred, to decrease drag, and hence the fuel consumption of civil aircraft.

This paper is concerned with practical methods of delaying transition. There are a number of ways this can be achieved. For example, heating the surface will heat the fluid adjacent to it, and increase its kinematic viscosity, and enhance the stability of the flow: from (1) it can be seen that increasing v will move the point at which the flow becomes unstable downstream. However, in the aerospace industry there is a great deal of interest in using suction to delay transition. It has long been known that, in theory, withdrawing very small amounts of fluid from the boundary layer through the wall can greatly enhance the stability characteristics of the flow. In practice, suction flow rates are at least an order of magnitude greater than those suggested by the basic theory for a flat plate. However they are still small, and suction is regarded as one of the most viable means for practical drag reduction.

This paper considers distributed suction using discrete suction panels of the type commonly used in the aerospace field, but with the ability to vary the suction flow rate on each panel independently to optimise the effort required to achieve a desired objective. Different objectives and hence cost functions may be specified in different situations. For an aerofoil the aim will not be to simply delay transition as much as possible, but also to minimise the energy consumption of the system taking account of a number of factors. On a nacelle however, the design requirement is to fix transition at a particular point so that the benefits of laminar flow are obtained ahead of this point but the flow at the trailing edge of the nacelle is turbulent. Turbulence is desirable at the trailing edge of a nacelle as the contribution to the total drag from the pressure rather than the skin friction is substantially decreased in a body with a large blunt trailing edge if the flow is turbulent in this region. Both of these design requirements will be considered below.

Note that the transition scenario outlined above, through the amplification of linear Tollmien–Schlichting waves is not the only way that transition can occur. In particular, in a 'noisy' environment, a 'bypass' mechanism may operate where this linear stage may be absent completely, and transition may occur much more rapidly. However, in an environment with a low free stream turbulence level, there should be a relatively long region in which this linear mechanism is important, and where surface suction as considered in this paper can have a significant effect on the flow.

3. Experimental configuration

A simple experimental arrangement, for which an extensive series of tests has been performed (see e.g. Nelson et al., 1997) is shown in Fig. 1. A flat plate with, if required, a smooth 2D hump on the wall opposite to generate a pressure gradient, and two suction panels, one upstream and one downstream of the peak of the hump, is positioned in a wind tunnel with mean flow speeds up to 22 m/s. The suction panels are made of laser drilled titanium sheets, which provides a surface which is sufficiently smooth so as not to provoke transition through surface roughness. The holes have a diameter of 0.1 mm (approximately), and are randomly distributed with a density of 10^6 holes per square metre, giving a porosity (hole area to surface area) of around 0.78%. Suction flow rates of up to 2 cm/s are possible, where the suction flow rate is defined as the volume flow per unit surface area, and hence the flow in the holes is of the order of the suction flow rate divided by the porosity.

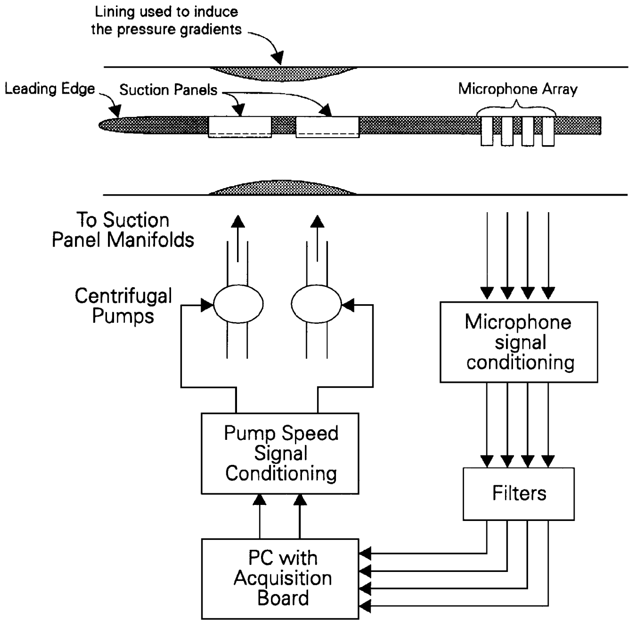
Lining used to induce the pressure gradients
Leading Edge
Suction Panels
Microphone Array
To Suction Panel Manifolds
Centrifugal Pumps
Microphone signal conditioning
Pump Speed Signal Conditioning
Filters
PC with Acquisition Board

Fig. 1. Experimental set-up.

The humps are circular arcs 0.36 m long with a radius of 0.555 m, giving a blockage of 10.3%. The suction panels are located such that, according to hot wire anemometry experiments, without suction transition occurs on the second panel. Each suction panel is connected to a pump which is controlled by a PC via an inverter. Downstream of the two panels an array of flush-mounted microphones is installed. There were eight microphones, positioned from 0.77 to 0.885 m from the leading edge. This covers the range of possible transition positions for the conditions in this wind tunnel. The signals of these microphones are high pass filtered at 800 Hz in order to remove the background noise due to the wind tunnel fan. The microphone signals are sampled at a sampling rate of 4 kHz for a given period, then the rms value is calculated. These rms values are then normalised by pre-measured rms values of the signals for a fully turbulent boundary layer, hence

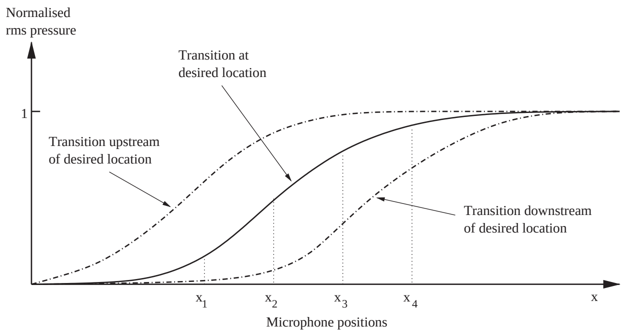

Fig. 2. Monitoring the transition position using surface mounted microphones.

yielding a value close to 0 for laminar flow and a value close to 1 for turbulent flow.

The location of the laminar-turbulent transition is determined by comparison of the rms pressures acquired by the microphones to four reference pressures. The resulting error signal can be described by

$$\begin{aligned} e(k) &= \sum_{m=1}^{4} [p_{ref}(x_m) - p(x_m, k)] \\ &= \sum_{m=1}^{4} p_{ref}(x_m) - \sum_{m=1}^{4} p(x_m, k) = r - y(k), \qquad (2) \end{aligned}$$

where k is the iteration index, r is the sum of the desired normalised pressures, and $y(k)$ is the sum of the measured normalised pressures. The desired normalised pressures are produced when transition is in the specified location. For the experiments r and $y(k)$ are referred to as the desired plant output and the actual plant output, respectively. These two values are also directly related to the desired transition location d and the actual transition location x_T, such that (1) is equivalent to

$$e(k) \propto [d - x_T(k)]. \qquad (3)$$

As illustrated in Fig. 2, an appropriate choice of the reference pressures enables the specification of the location of the transition region with respect to the microphone positions. Fig. 3 shows, for a constant mean flow speed, the position of transition as a function of the suction flow rates on the two panels. The contour lines indicate the position of transition. Fig. 3 also illustrates the purpose of the controller which will regulate the two suction flow rates in this case until the shaded point is reached. This point corresponds to zero error with the minimum possible suction effort.

Further details of this experimental work and the apparatus can be found in Rioual (1994).

4. Theoretical modelling

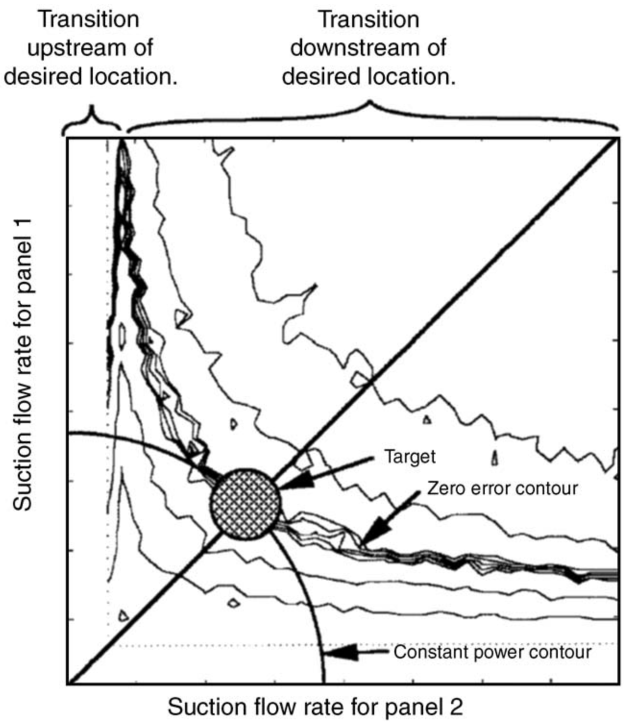

Fig. 3. Experimental relationship between transition position and suction flow rates for a two panel flat plate case.

When simulating viscous laminar flow it is possible to solve the full governing equations, the Navier–Stokes equations, numerically. However, with the type of high Reynolds numbers considered here it is more efficient to apply Prandtl's theory which assumes that the viscous part of the flow is restricted to a thin boundary layer, and solve a reduced set of equations which contain all the essential physics of the problem. The viscous part of the flow is described by the normalised boundary-layer equations:

$$\tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} + \tilde{v}\frac{\partial \tilde{u}}{\partial \tilde{y}} = \tilde{u}_e \frac{d\tilde{u}_e}{d\tilde{x}} + \frac{\partial^2 \tilde{u}}{\partial \tilde{y}^2}, \qquad (4)$$

$$\frac{\partial \tilde{u}}{\partial \tilde{x}} + \frac{\partial \tilde{v}}{\partial \tilde{y}} = 0 \qquad (5)$$

with the following boundary and initial conditions:

$$\tilde{y} = 0: \quad \tilde{u} = 0, \quad \tilde{v} = \tilde{v}_{wall}, \qquad (6)$$

$$\tilde{y} \to \infty: \quad \tilde{u}(\tilde{x}, \tilde{y}) \to \tilde{u}_e(\tilde{x}), \qquad (7)$$

$$\tilde{x} = \tilde{x}_0: \quad \text{Blasius profile}, \qquad (8)$$

where $\tilde{u}$ and $\tilde{v}$ are the axial and the transverse velocities, $\tilde{u}_e$ is the axial velocity at the edge of the boundary layer, and $\tilde{x}$ and $\tilde{y}$ are the streamwise and normal coordinates. The variables in (4)–(8) have been non-dimensionalised using the scalings

$$\tilde{u} = \frac{u}{U_\infty}, \quad \tilde{v} = \frac{v}{U_\infty}\sqrt{R}, \quad \tilde{x} = \frac{x}{L}, \quad \tilde{y} = \frac{y}{L}\sqrt{R},$$

where U_∞ is the free-stream velocity, L is the length of the plate, $R = \rho U_\infty L/\mu$ is the free stream Reynolds number, ρ is the density, and μ the viscosity of the fluid. The edge velocity $\tilde{u}_e$ in (7) comes from a first-order potential flow calculation for the flow past the body. A standard finite difference method is used to solve (4)–(8).

In this work it is assumed that the transition from laminar to turbulent flow is preceded by the linear amplification of small 2D travelling waves, the so-called Tollmien–Schlichting waves. Note that this is only one possible scenario for transition to turbulence, and that there are other, much more vigorous, mechanisms that can cause transition resulting in bypass transition (see e.g. Schmid & Henningson (2001) for discussion of this and other relevant details). However, the assumption of linear amplification of small disturbances, and transition prediction through the growth of these disturbance (the e^N method—see below), forms part of a standard method used in the aerospace industry for the design of wings, fins and nacelles, and will therefore be employed here.

Linear hydrodynamic stability theory describes the stability of the flow to this kind of disturbance. A detailed description of the linear stability theory is given, for example, by Mack (1977, 1984), therefore only the main features are discussed in this paper. Linear stability theory is based on the study of small perturbations super-imposed on a basic laminar flow. The purpose of this theory is to determine the conditions under which these perturbations are amplified or damped. Therefore the velocity components and the pressure are decomposed into an undisturbed mean part and a small disturbance. This decomposition is introduced into the unsteady 2D Navier–Stokes equations. The resulting equations are simplified by making the approximation that the products of disturbance quantities are negligible.

The result of the above analysis is a set of linear disturbance equations. The pressure is then eliminated by cross-differentiation. In the resulting equation the disturbance is expressed as a normal mode:

$$\varphi(x, y, t) = \Phi(y)\mathrm{e}^{\mathrm{i}(\alpha x - \omega t)}, \tag{9}$$

where φ is the (dimensionless) disturbance stream function, α is the wave number, ω the wave frequency of the disturbance, and Φ is the (complex) disturbance amplitude. This leads to the following disturbance velocities:

$$\hat{u} = \Phi'\mathrm{e}^{\mathrm{i}(\alpha x - \omega t)}, \tag{10}$$

$$\hat{v} = -\mathrm{i}\alpha\Phi\mathrm{e}^{\mathrm{i}(\alpha x - \omega t)}, \tag{11}$$

where the prime stands for differentiation with respect to y. Introducing these velocities into the disturbance equation yields

$$\left(u - \frac{\omega}{\alpha}\right)(\Phi'' - \alpha^2\Phi) - u''\Phi + \frac{\mathrm{i}}{\alpha\, Re}(\Phi'''' - 2\alpha^2\Phi'' + \alpha^4\Phi) = 0 \tag{12}$$

the well known Orr–Sommerfeld equation. The Reynolds number Re is now based on the local boundary-layer thickness, and u is the streamwise component of the mean-flow velocity, obtained by the solution of the boundary-layer equations. Note that there is an implicit re-scaling here between the variables used for the flow calculation and the stability calculation.

Eq. (12) defines an eigenvalue problem with homogeneous boundary conditions ($\hat{u}$ and $\hat{v}$ must vanish at the wall and in the free stream). For a given Reynolds number and real frequency, the solution yields an eigenfunction in the form of the complex disturbance amplitude and as an eigenvalue, the complex wave number $\alpha = \alpha_r + \mathrm{i}\alpha_i$. The imaginary part of the wave number, α_i gives the spatial growth rate of the disturbance, which is positive for damped waves and negative for amplified waves. The speed of propagation of the disturbance is given by α_r/ω.

Using linear stability theory, the onset of instability can be predicted, but instability is only a very early precursor of ultimate transition to turbulent flow. The e^N-method is a standard approach to relating the point of transition to the amplification of the disturbance waves. The total amplification rate is calculated by integrating the spatial amplification factors α_i at each single real frequency ω:

$$\ln\left(\frac{A}{A_0}\right) = -\int_{x_0}^{x} \alpha_i \,\mathrm{d}x, \tag{13}$$

where A is the wave amplitude and the index 0 refers to the streamwise position where this wave becomes unstable.

The e^N-method is based on the envelope of these curves of total amplification, where N is the maximum amplification factor at each point:

$$N = \max\left\{\ln\left(\frac{A}{A_0}\right)\right\}. \tag{14}$$

The position of transition is then predicted as the first point at which N exceeds a critical value N_T which can vary according to the operating conditions of the flow. Malik (1990) gives an extensive overview over the values of N_T found in various wind tunnel and flight experiments. For a low free-stream turbulence level, $N_T = 10$ proves to be a reasonable approximation. However, Mack (1977, 1984) shows that this critical amplification factor is strongly dependent on the turbulence level, and suggested the following relation:

$$N_T = -8.43 - 2.4\ln_e Tu, \tag{15}$$

where the turbulence level is calculated as

$$Tu = \sqrt{\frac{\overline{u'^2} + \overline{v'^2} + \overline{w'^2}}{3}}$$

with (u', v', w') denoting the dimensionless turbulent disturbance velocities and the overline averaging with respect to time. In the present case the value of N_T is determined by comparison with the experiments of Nelson et al. (1997). This gives $N_T = 4.3$, corresponding to a free-stream turbulence level of approximately 0.5%.

The calculations reported below use air at standard conditions, with kinematic viscosity of 1.5×10^{-5} m^2/s, and a free stream velocity of 20 m/s. All lengths will be in metres and velocities in m/s, and where convenient these units will be dropped.

5. Cost functions

One relatively simple cost function for this general problem area is obtained by fixing the transition position at a pre-determined location, and then minimising the effort by varying the suction flow rates and the positions of the panels to achieve the best result. This is clearly of practical interest as it comes directly from the design requirements for a suction control system on a nacelle. We formulate two (slightly) different optimisation problems for the two scenarios we consider in this paper: flow past a flat plate where the position of the panels are not fixed but are adjusted as part of the optimisation procedure (moving panels) and flow past an aerofoil with fixed panels and linearly distributed suction flow rates. The first of these scenarios reflects the fact that the minimum effort depends on not just the amount of suction, but also its distribution. The second, is a generalised version of a suction system consisting of a relatively large number of contiguous plenum chambers under the front part of the body providing a near continuous suction distribution. Both are based on systems of practical interest and have not been the subject of any detailed investigation in the literature to-date.

Note here that a transition location is some function of suction flow rates and there is an unknown or, at best, very complicated relationship linking the panel positions and flow rates, i.e. essentially ‘black box’ optimisation problems have to be treated here. Also there clearly must be bounds on the suction flow rates and here it is assumed that these are known.

5.1. Cost function for moving panels on a flat plate

The problem of constraining the transition position to some desired location by finding optimal locations for the suction panels on flat plate can be formulated as follows: Find optimal (u_i, p_i), $i = 1 \dots N$, which minimise

$$\phi_s = k_1 \sum_{i=1}^{N} u_i^2, \tag{16}$$

subject to

$$x_t - x_d = 0.$$

Here u_i denotes the suction flow rate through panel i which runs from $x = p_i$ to $x = p_i + H$, where p_i is the starting point of panel i and the panels are of equal length H, ϕ_s is an estimate of the suction effort, N is the number of panels, k_1 is a positive constant, u_i is the suction flow rate through panel i, x_t is the current transition position as predicted using the methods described in the previous section, and x_d is the desired transition position. A key point here is that panels are not fixed, and their position is found as part of the solution to the problem. There is however a further constraint in that the panels cannot overlap, and this limits the acceptable values of the p_i still further. This constrained minimisation problem can be rewritten as unconstrained minimum by solving the following problem:

Find optimal $(u_i, p_i), i = 1 \dots N$, which minimise:

$$\phi = \phi_s + k_2 |x_t - x_d|^h, \tag{17}$$

where k_2 is a positive constant, and h is a positive integer to be selected based on knowledge of the specific configuration being considered (here the case of $h = 1$ is treated). This last optimisation problem is nonlinear and, as experiments have shown, multi-modal. This is true even for the case with $N = 2$, which is investigated here numerically (Section 8) as a representative case. Note that ϕ_s is not a true measure of the suction effort, but it is a monotonic function of the power used by the suction system, and hence minimising ϕ_s will produce a valid solution.

The solution of the optimisation problem just defined is a two stage process. First we compute the optimal flow rates under (17) for a fixed non-overlapped pair of (p_1, p_2) of starting positions and then find optimal positions of the two panels. Each of the stages is detailed in Sections 6.1 and 6.2. For optimisation purposes, it is assumed that there are known bounds on the positions of the panels.

5.2. Cost function for linearly distributed suction flow rates on an aerofoil

The problem of constraining the transition position to some desired location on an aerofoil is considered here for non-overlapped panels without gaps and fixed in locations between the leading edge and 20% of the chord of the body—the region where suction will provide the greatest benefit. Then with suction flow rates linearly distributed over panels a practically relevant cost function is of the form:

$$\text{minimise}\ \phi_s = k_1 \sum_{i=1}^{N+1} u_i^2, \tag{18}$$

subject to $x_t - x_d = 0$. Here ϕ_s denotes the suction effort by panel i, N is the number of panels, k_1 is a positive constant, u_i is a suction flow rate at the beginning of panel i, or the end of panel i if $i = N+1$, x_t is the transition position and x_d is the desired transition position. Note that the suction flow rate at the beginning of panel i is equal to a suction flow rate at the end of panel $i-1$ for $1<i<N+1$. As in the previous problem, the constrained version can be rewritten as an unconstrained problem which here is of the form:

$$\text{minimise}\ \phi = \phi_s + k_2|x_t - x_d|^h, \tag{19}$$

where k_2 is a positive constant, and h is as in the previous case. Again this is a nonlinear problem and has been shown in experiments to be multi-modal.

To obtain a realistic problem in the case of an aerofoil, for example, can often require the number of panels N to be very large and thereby lead to an enormous computational burden. (Experiments have shown that it can take days to complete a design run.) Hence we develop and demonstrate the effectiveness of parallel algorithms for these problems. In this paper, the calculations are performed for an NACA 0012 aerofoil, with a chord of 1 m. In the numerical work here (Section 9) the representative cases of 5 and 10 panels are considered.

6. Optimisation algorithms

In this and the next section we detail in turn the algorithms developed for use in the work reported in this paper.

6.1. Optimisation of the flow rates with fixed panels

Consider a case when two panels are used and are in fixed positions and the design objective is to compute suction flow rates which minimise a cost function of form (17). This cost function is nonlinear, non-differential and multi-modal. Moreover, evaluation of the objective function requires the running of expensive analysis codes.

Since the cost function is non-smooth, non-convex, non-differential and multi-modal, appropriately modified versions of direct search methods are (potentially) the most appropriate here. Direct search methods of the form considered here have their origins in the work of Nelder and Mead (1965) but it is later modifications which are of particular interest. These are the so-called deformed configuration methods (Rykov, 1995) and the multi-directional search methods of Torczon (1991) (which can also be regarded as members of the deformed configuration class of methods). Here we will use an assortment of configuration types, i.e. a simplex, defined to have $c = n+1$ vertices in n-dimensional space, a complex, defined to $c \geqslant n+2$ vertices in n-dimensional space, of regular or irregular geometry, mapping functions, local optimality criteria, adaptation procedures and the rules for combining them, which together generate a class of deformed configuration methods. The major differences between the algorithm developed here from those of Torczon (1991) and Rykov (1995) are an additional preliminary procedure, the use of complex vertices defined by $c = 3^n - 1$ vertices in n-dimensional space, and the use of so-called accepted mapping directions to ensure that the global optimum of the multi-modal cost function is found.

Next we detail the so-called modified complex algorithm developed here for solving the optimisation problem under consideration. (In the numerical examples of Sections 8.1 and 8.2, only the case of two panels will be considered.) This algorithm consists of three stages—initialisation, exploratory moves and the modified direct search algorithm, respectively, which are now detailed in turn, and the cost function is denoted by $F(\cdot)$.

- *Initialisation*: Feasible bounds are defined and an inscribed complex is constructed within these bounds. The geometrical centre of the complex is calculated and the cost function values at the vertices of the complex and its centre are obtained through simulation.
- *Exploratory moves*: The purpose of these is to reduce the complex size and to attempt to locate the most likely regions for a global minimum. A line connected to each vertex of the complex and the complex centre is then divided into 10 intervals and cost function values obtained at these intervals. The point with minimum cost function value is taken as a new complex point, located in the direction of the old complex point and centre. After this operation a new complex with the same geometrical centre as the old one is constructed for further optimisation.
- *Modified direct search algorithm*: The basic operation of this algorithm is that a complex with $c = 3^n - 1$ vertices exists at each iteration. Then the geometrical centre of complex, together with the cost function values in each complex vertex and complex centre are obtained, and the correct enumeration of complex vertices is undertaken, i.e. such that the following chain of inequalities holds

  $$F(\mathbf{u}^{(k,1)}) \geqslant F(\mathbf{u}^{(k,2)}) \geqslant \cdots \geqslant F(\mathbf{u}^{(k,c)}),$$

  where k denotes the iteration number, and $\mathbf{u}$ is a vertex of the complex in n-dimensional space. Then the vertices with cost function values greater than the cost function value at the centre of the complex are mapped through the geometrical centre of non-mapped vertices. If the cost function of a reflected vertex is less than that for the complex centre, the step-size is increased and a vertex of the new complex is taken to be one with the minimum cost function value, otherwise the original vertex is brought near to the complex centre. These iterations are repeated until either the stopping rule is satisfied or a pre-specified maximum number of iterations has been undertaken. The stopping rule used in this case is that the algorithm is assumed to have converged if

$$F(\mathbf{u}^{(k,1)}) - F(\mathbf{u}^{(k,c)}) < \varepsilon_1$$

and

$$\max_{2<i<c} \|\mathbf{u}^{(k,1)} - \mathbf{u}^{(k,i)}\| < \varepsilon_2$$

hold, where ε_1 and ε_2 are user specified convergence measures (in the numerical work here both were set at 10^{-6}).

The modified direct search algorithm can now be detailed as follows.

Initial Data to be specified—an initial complex $\mathbf{u}^{(0,i)}$, $i = 1, \ldots, c$ (in the case of two panels $c = 8$), the centre of the initial complex $\mathbf{u}^{(k)}$; the step-parameters α, β, and γ (here $\alpha = 1.9$, $\beta = 0.25$, $\gamma = 1.1$); the maximum number of steps NN (here $NN = 50$)

```
for $k = 0, 1, \ldots NN$ do
    perform correct enumeration
    check the stopping rule
    $m \leftarrow 0$
    for $i = 1, \ldots, c$ do
        if $F(\mathbf{u}^{(k,i)}) > F(\mathbf{u}^{(k)})$ then
            $m \leftarrow m + 1$
    for $i = 1, \ldots, m$ do
        $\mathbf{s}^{(k,i)}(m) = \frac{1}{c-m} \sum_{j=m+1}^{c} \mathbf{u}^{(k,j)}$
    $\mathbf{u}^{(kr,i)} = \mathbf{s}^{(k,i)}(m) + \gamma(\mathbf{s}^{(k,i)}(m) - \mathbf{u}^{(k,i)})$
    calculate $F(\mathbf{u}^{(kr,i)})$
    if $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k)})$ then
        $\mathbf{u}^{(ke,i)} = \mathbf{s}^{(k,i)}(m) + \alpha(\mathbf{s}^{(k,i)}(m) - \mathbf{u}^{(k,i)})$
        calculate $F(\mathbf{u}^{(ke,i)})$
        if $F(\mathbf{u}^{(ke,i)}) < F(\mathbf{u}^{(kr,i)})$ then
            $\mathbf{u}^{(k+1,i)} = \mathbf{u}^{(ke,i)}$
            $F(\mathbf{u}^{(k+1,i)}) = F(\mathbf{u}^{(ke,i)})$
        else
            $\mathbf{u}^{(k+1,i)} = \mathbf{u}^{(kr,i)}$
            $F(\mathbf{u}^{(k+1,i)}) = F(\mathbf{u}^{(kr,i)})$
    else
        $\mathbf{u}^{(k+1,i)} = \mathbf{u}^{(k)} - \beta(\mathbf{u}^{(k)} - \mathbf{u}^{(k,i)})$
        calculate $F(\mathbf{u}^{(k+1,i)})$
for $i = m + 1, \ldots, c$ do
    $\mathbf{u}^{(k+1,i)} = \mathbf{u}^{(k,i)}$, $F(\mathbf{u}^{(k+1,i)}) = F(\mathbf{u}^{(k,i)})$
calculate new complex centre $\mathbf{u}^{(k+1)}$ and $F(\mathbf{u}^{(k+1)})$
```

The vertex which yields the minimum value of $F(\mathbf{u})$ is assumed to be the estimate of the minimum point.

Before proceeding further it should be noted that when the position of the panels is variable, different positions can result in different minimum values of the cost function.

6.2. Computing the optimal positions for two panels

When the positions of the panels (i.e. p_i) are allowed to vary, the complexity of the optimisation problem is considerably increased. Consider, however, the case of two panels. Then careful investigation of the final cost function shows that optimisation of the panel positions can be done independently of each other, i.e. fix one panel position whilst optimising the position of the other. Further, in this case the sub-cost function associated with the variable panel position is a unimodal 1D function. Hence the use of the golden section search method is proposed for the optimisation of each of panel position, i.e. first p_1 is chosen according to the golden section method and then the same method is run for p_2. This approach is faster than going through all possible combinations of panel positions and it also enables either a global minimum or, at worst, a near optimal solution to be computed. Details of the golden search method are omitted here since it can be found in many optimisation textbooks (see, for example, Fletcher, 1987).

Actual results (see Section 8) confirm that this algorithm can deliver good results but a major drawback is that its complexity will grow exponentially with dimension, i.e. with the number of panels used. Moreover, if the more realistic scenarios are considered (for example, continuous suction distributions over the front part of body with a realistic pressure gradient, etc.) and/or more panels are used, a much larger computational effort would be required. To reduce computational time in such cases, a parallel version of the algorithm detailed above has been developed and investigated for a case of linearly distributed flow rates on an aerofoil as detailed next.

7. Parallel global optimisation of linearly distributed flow rates on an aerofoil

In this section parallel global optimisation algorithms are developed for optimisation of linearly distributed flow rates on an aerofoil, which are then applied (see Section 9) to the particular cases of 5 and 10 non-overlapping panels without gaps fixed at 20% of chord with cost function (19). These are representative numerical cases in this particular aspect of the general problem area which in pure optimisation terms are nonlinear, non-differential, multi-modal and where it is required to run expensive analysis codes. (The analysis and results given here start from the preliminary findings in Veres, Tutty, Rogers, & Nelson, 2003b). We consider three different classes of algorithms and also combinations of them. We now detail each of these in turn.

7.1. Random search method

These methods directly aim to find the global minimum (Horst & Pardalos, 1995) and the version of them used here consists of the following steps.

- Choose the initial number of points N_1 and set initial feasible bounds on the flow rates.
- Select N_1 uniformly distributed points within the initial feasible bounds.
- Obtain the cost function values at the initial set of points concurrently and save the point with the minimum cost function value as the closest possible to an optimal solution.
- Taking into consideration cost function values, reduce the feasible bounds to areas more likely to contain optimal points and choose a number of points N_2 to be used within these new bounds.
- Select N_2 uniformly distributed points within the new feasible bounds and evaluate the cost functions at these points concurrently.
- Choose the point with the minimum cost function value from these new points and set the old near optimal solution equal to the new near optimal solution.

This approach can assist in refining the bounds on suction flow rates, but will require the use of a sufficient number of points to find the optimal or near optimal solution, i.e. to achieve a good coverage of the feasible region.

7.2. Parallel global optimisation method

Since the cost function (19) to be optimised is non-smooth, non-convex and multi-modal, we have already argued the case for the development of modifications of direct search methods to find the global minimum. Further support for these is provided by the fact that, for example, an asynchronous parallel pattern search (Kolda, Hough, & Torczon, 2001), was successfully implemented on parallel computing platforms for minimisation of objective functions which are expensive to evaluate.

Here we propose to use an algorithm which is one of a number of possible modifications of the deformed configuration methods (Rykov, 1995), which belong to the general class of direct search methods. In the method of deformed configurations, the basic feature of constructing the simplex and applying an optimisation procedure to the complex is augmented by introducing search control. This consists of choosing the locally optimal direction, mapping the configuration vertices, the centroid, and the step-size, and leads to reduced cost function values. Also, since we are mapping several configuration (simplex or complex) vertices the number of which is automatically adjusted from step to step, these methods have relatively fast convergence and are less sensitive to noise corruption than the computation of the cost function at any instant.

The major differences between the algorithm proposed here and an asynchronous parallel pattern search (Kolda et al., 2001) or the deformed configuration methods (Rykov, 1995) are the addition of the preliminary procedure and the (sophisticated) choice of mapping directions to ensure that we obtain the global optimum of multi-modal cost function. All the vertices are divided into three groups, i.e. mapped, reflected and the best vertex between best vertex of the complex and the complex centre. Mapped vertices are those whose cost function values are greater than the cost function value in the complex centre. Reflected vertices are those whose cost function values are less than or equal to those of the complex centre but also greater than cost function value at the best complex vertex, i.e. the complex vertex with the minimum cost function value in the complex.

After division of vertices the construction of a new complex is carried out as detailed below. All complex vertices are used here in the exploration of possible search directions and this gives better coverage of space in comparison to the original algorithm. This is a very important feature when there are several local optimum in the example under consideration.

In detail, the new algorithm for minimising the (19) consists of three steps—initialisation, exploratory moves, and the modified procedure for executing deformed configurations. These are described in turn next.

7.2.1. Initialisation

- Define feasible bounds for the suction flow rates.
- Define the step-size parameters α, β, and γ.
- Select stopping tolerances tol_1 and tol_2.
- Select the maximum number of iterations N_1 and maximum number of iterations allowed without changes in the minimum cost function values N_2.
- Construct an inscribed complex $\mathbf{u}^{(0,i)}$, $i = 1, \ldots, c$ within the bounds and compute the geometrical centre of the complex $\mathbf{u}^{(k)}$.
- Evaluate the cost function values at the vertices of complex $F(\mathbf{u}^{(0,i)})$ and the centre $F(\mathbf{u}^{(k)})$, concurrently.

7.2.2. Exploratory moves

(a) Divide a line connecting each vertex of the complex and the complex centre into ten intervals.
(b) Calculate the cost function values at the points defined in the last step concurrently.
(c) Take the point with minimum cost function value as a new complex point in the direction of the old complex point and centre.

7.2.3. Modified method of deformed configurations

1. If $k \leqslant N_1$, then set $k = k+1$ (k is the iteration counter) and go to Step 2. Else, exit.
2. Perform correct enumeration of complex vertices, i.e. such that the following chain of inequalities holds:

   $$F(\mathbf{u}^{(k,1)}) \geqslant F(\mathbf{u}^{(k,2)}) \geqslant \cdots \geqslant F(\mathbf{u}^{(k,c)}).$$

3. If

   $$(F(\mathbf{u}^{(k,1)}) - F(\mathbf{u}^{(k,c)}) < \varepsilon_1$$

   and

   $$\max_{2<i<c} \|\mathbf{u}^{(k,1)} - \mathbf{u}^{(k,i)}\| < \varepsilon_2)$$

   or the number of iterations without the minimum cost function value changing is greater than N_2, exit. Else, go to Step 4.
4. Divide all vertices into three groups: m mapped vertices with $F(\mathbf{u}^{(k,i)}) > F(\mathbf{u}^{(k)})$, $i = 1, \ldots, m$; q reflected vertices with $F(\mathbf{u}^{(k,i)}) \leqslant F(\mathbf{u}^{(k)})$ and $F(\mathbf{u}^{(k,i)}) > F(\mathbf{u}^{(k,c)})$, $i = 1, \ldots, q$; q here is the vertex with the minimum cost function value from the set whose elements are the best complex vertices and the complex centre.
5. Map m vertices through the centre of unmapped vertices

   $$\mathbf{s}^{(k,i)}(m) = \frac{1}{c-m} \sum_{j=m+1}^{c} \mathbf{u}^{(k,j)}$$

   according to

   $$\mathbf{u}^{(kr,i)} = \mathbf{s}^{(k,i)}(m) + \gamma(\mathbf{s}^{(k,i)}(m) - \mathbf{u}^{(k,i)}).$$

6. Reflect q vertices through the best complex vertex with the minimum cost function value according to

   $$\mathbf{u}^{(kr,i)} = \mathbf{u}^{(k,c)} + \gamma(\mathbf{u}^{(k,c)} - \mathbf{u}^{(k,i)}).$$

7. If $F(\mathbf{u}^{(k,c)}) > F(\mathbf{u}^{(k)})$, then reflect the best complex vertex through the complex centre

   $$\mathbf{u}^{(kr,c)} = \mathbf{u}^{(k)} + \gamma(\mathbf{u}^{(k)} - \mathbf{u}^{(k,c)}).$$

   Else, reflect the complex centre through the best complex vertex

   $$\mathbf{u}^{(kr,c)} = \mathbf{u}^{(k,c)} + \gamma(\mathbf{u}^{(k,c)} - \mathbf{u}^{(k)}).$$

8. Evaluate $F(\mathbf{u}^{(kr,i)})$, $i = 1, \ldots, c$, concurrently.
9. If $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k)})$, $i = 1, \ldots, m$, then repeat the mapping with a bigger step-size parameter α. Else, repeat the mapping with a smaller step-size parameter β.
10. If $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k,c)})$, $i = 1, \ldots, q$, then repeat the reflection with a bigger step-size parameter α. Else, repeat the reflection with a smaller step-size parameter β.
11. If a reflected vertex in group q has a smaller cost function value than a non-reflected vertex, repeat the reflection with a bigger step-size parameter α. Else, repeat the reflection with a smaller step-size parameter β.
12. Calculate cost function values at the new c vertices concurrently.
13. Compare the cost function values computed for the old set of c vertices with those at the new set of corresponding $2c$ vertices obtained by reflection/mapping with different step-size parameters. Choose c vertices with the minimum cost function values as the set of new complex vertices.
14. Calculate a new geometrical centre of the new complex and the cost function value in it.

The vertex which gives the minimum value of $F(\mathbf{u})$ is taken as the estimation of the minimum point.

In the above algorithm, parallelisation occurs only at the stage of computing the cost function values. Further parallelisation is, however, possible at the stage when the new complex vertices are being computed. Our experience has shown that this algorithm can find the global optimum quite accurately (Veres, Tutty, Rogers, & Nelson, 2003a), but the complexity of this problem will grow exponentially with dimension and in case of 10 panels it requires weeks of computational time even on parallel processors to achieve either the global minimum or a 'reasonable' estimate of it.

These last facts have led to the development of the multi-start parallel global optimisation algorithm detailed next for higher dimensions where, crucially, the problem complexity will grow linearly rather than exponentially.

7.3. Multi-start parallel global optimisation

To speed up the optimisation process for a large number of panels, we have developed the following algorithm which, in effect, combines the multi-start parallel global optimisation algorithm with the pattern search approach (Torczon, 1991) for local search tasks. The main idea here is to first find the points from where the local search starts and then apply this to selected points. Again, the resulting algorithm consists of three stages—initialisation, multi-start points selection, and local search which are detailed in turn next.

7.3.1. Initialisation

- Define feasible bounds for linear distributed flow rates.
- Define the step-size parameters α, β, and γ.
- Select the maximum number of iterations N_1 and maximum number of iterations allowed N_2 without changes in the minimum cost function value.
- Choose a number of complex vertices c for the local search.
- Select N uniformly distributed random points $\mathbf{u}_i$, $i = 1, \ldots, N$, within the feasible bounds—these points will form the set of initial points S_0.
- Evaluate the values of the cost functions $F(\mathbf{u}_i)$ concurrently.

7.3.2. Multi-start points selection

(a) Sort all points in ascending order with respect to the cost function values, i.e. $\mathbf{u}_1$ and $\mathbf{u}_N$ will be the points with minimum and maximum cost function values, respectively.

(b) Normalise all cost function values as

$$F_n(\mathbf{u}_i) = \frac{F(\mathbf{u}_i) - F(\mathbf{u}_1)}{F(\mathbf{u}_N) - F(\mathbf{u}_1)}, \quad i = 1, \ldots, N.$$

(c) Remove the points from the set S_0 with $F_n(\mathbf{u}_i) > tol_1$.

(d) Do the following until there are no points left in set S_0.

(e) Add the first point to the set of multi-start points S.

(f) Calculate the Euclidean distances from this point to all other points, normalise these distances, and then remove from the set S points with distances less or equal to the chosen threshold tol_2. Go to (d).

7.3.3. Local search

1. Let $\mathbf{u}_{min}$ be a point with the minimum cost function value $F(\mathbf{u}_{min})$ from the set S.
2. Choose the number of complex vertices c.
3. For each multi-start point in the set S do the following.
4. Construct an initial complex.
5. $min \leftarrow arg_i \min_{0 \leqslant i \leqslant c-1} F(\mathbf{u}^{(0,i)})$.
6. Swap $\mathbf{u}^{(0,min)}$ and $\mathbf{u}^{(0,0)}$.
7. If $k_1 \leqslant N_1$ and $k_2 \leqslant N_2$, then $k_1 = k_1 + 1$ (k_1 is—(as before) the iteration number and k_2 is the number of iterations for which there is no change in the minimum cost function value), go to Step 8. Else, go to Step 14.
8. Reflect $c - 1$ vertices through the best vertex $\mathbf{u}^{(k,min)}$ according to
$$\mathbf{u}^{(kr,i)} = \mathbf{u}^{(k,min)} + \gamma(\mathbf{u}^{(k,min)} - \mathbf{u}^{(k,i)}), = 1, \ldots, c-1.$$
9. Evaluate $F(\mathbf{u}^{(kr,i)})$, $i = 1, \ldots, c - 1$ concurrently.
10. If $F(\mathbf{u}^{(kr,i)}) < F(\mathbf{u}^{(k,min)})$, then repeat the mapping with a bigger step-size parameter α. Else, repeat the mapping with a smaller step-size parameter β.
11. Evaluate the cost function values at the new $c - 1$ vertices concurrently.
12. Choose $c - 1$ vertices from the old vertices and reflect them with different step-size parameters from the ones used to obtain the minimum cost function value.
13. If there is no change in the minimum cost function value set $k_2 = k_2 + 1$, else $k_2 = 0$.
14. The vertex $\mathbf{u}^{k,min}$ with minimum value of $F(\mathbf{u})$ is taken as the estimation of the minimum point.
15. If $F(\mathbf{u}^{k,min}) < F(\mathbf{u}_{min})$, replace $\mathbf{u}_{min}$ by $\mathbf{u}^{k,min}$, and $F(\mathbf{u}_{min})$ by $F(\mathbf{u}^{k,min})$. Go to Step 3.

This algorithm is faster than the parallel global optimisation algorithm based on the deformed configuration method, but its accuracy will be depend on the thresholds in the multi-start points selection stage and on the number of vertices in the initial complex for each multi-start point. To obtain a compromise between accuracy and the speed of global optimisation, a combination of a random search procedure and a multi-start parallel global optimisation is proposed below.

7.4. Combination of random search procedure and multi-start parallel global optimisation

If the most important criterion is accuracy, then it is possible to combine the random search method with multi-start parallel global optimisation. The idea here is simple—initial (ideally wide ranging) bounds for flow rates are defined and then a random search is used to narrow these bounds to regions more likely to contain the global optimum. Following this, the multi-start parallel global optimisation is applied with these new bounds in place. The steps in implementing this algorithm can be summarised as follows.

- Choose an initial number of points N_1 and initial feasible bounds for the flow rates.
- Select N_1 points uniformly distributed within the initial feasible bounds.
- Compute the cost function values for the initial set of points concurrently.
- Based on cost function values, reduce the feasible bounds to areas more likely to contain optimal points.

- Select N uniformly distributed random points $\mathbf{u}_i$, $i = 1, \ldots, N$, from within the feasible bounds, where these points will form the set of initial points S_0.
- Evaluate the cost function values $F(\mathbf{u}_i)$ concurrently.
- Perform multi-start points selection and the local search stages of the multi-start parallel global optimisation algorithm with the initial bounds for flow rates.

Note here that after choosing multi-start points, the bounds on flow rates are relaxed again. The purpose of this is to avoid missing the global minimum. (Since only a 'coarse' investigation of regions containing global solutions is undertaken (to avoid having an excessive number of points to evaluate), there is the possibility that a global minimum will be outside, but very close to, the reduced bounds.)

8. Results: optimisation of the flow rates with two moving panels

Consider a flat panel 1 m long with 2 non-overlapping suction panels, where the lengths of the panels are both equal to 0.1 m with a minimum gap of 0.05 m between them. To reduce the computational burden, the following bounds were imposed on the positions of the panels: $p_1 \in [0.18, 0.55]$, $p_2 \in [p_1 + 0.15, 0.7]$. The desired transition is taken as $x_d = 0.8$ m and also the following cost function parameters (see also MacCormack et al., 2002) were used $k_1 = 10^4$, $k_2 = 10$ and $h = 1$.

8.1. Optimisation of the flow rates with fixed panels

To check the robustness of this algorithm the following actions were taken. First, the simulations were run without optimisation and the desired transition point was achieved by changing the panel positions in steps of 0.01 m, subject to the constraint that they cannot overlap. The required suction flow rates were determined as follows. First, an initial guess of possible suction flow rates was made and then transition point x_t was computed by a suitable computational fluid dynamics (CFD) code for given flow rates. If the difference between desired transition point x_d and calculated transition point x_t was greater than a specified tolerance, the flow rates were adjusted using:

$$u_i(k+1) = u_i(k) - 0.001 * (x_d - x_t),$$

where $i = 1, \ldots, N$ (k is the iteration index). This procedure was then repeated until either convergence occurred or the maximum allowed number of iterations had been completed. Note also that in this case that flow rate will be the same for each panel.

The minimum cost function value achieved with the above approach is given in Table 1, where in this and all other tables in this paper, distances are given in metres, flow rates in metres per second, and the cost functions involved have been non-dimensionalised by suitable choices of units for the constants k_1 and k_2.

Now with fixed panels positions $(p_1, p_2) = (0.27, 0.51)$ the optimisation algorithm was applied. Different initial bounds were used to judge algorithm convergence from different starting regions. The results are given in Table 2. As can be seen from Table 2, applying the optimisation algorithm does reduce the overall cost, primarily by moving x_t closer to x_d, but in some cases reducing the suction cost by a small amount as well. Note also that the results in Table 2 were all obtained with the same values of the tuning parameters α, β and γ. We also found that the suction cost can be further reduced, but not to any significant extent, by varying the parameters α, β and γ. The running time of the algorithm for fixed panel positions depends on initial bounds and

Table 1
The minimum overall cost function without optimisation but with moving panels

p_1	p_2	u_1	u_2	x_t	ϕ_s	ϕ
0.27	0.51	−2.002e − 4	−2.002e − 4	0.799968	8.0160e − 4	8.3360e − 4

Table 2
The minimum overall cost function with optimisation and fixed panel locations (0.27, 0.51)

Bounds	u_1	u_2	x_t	ϕ_s	ϕ
[−2.5e − 4; −1.5e − 4]	−2.00125e − 4	−2.00288e − 4	0.8	8.0165e − 4	8.0165e − 4
[−3e − 4; −1e − 11]	−2.01656e − 4	−1.98695e − 4	0.8	8.0145e − 4	8.0145e − 4
[−4e − 4; −1e − 11]	−2.02957e − 4	−1.97355e − 4	0.8	8.0141e − 4	8.0141e − 4
[−5e − 4; −1e − 11]	−2.05762e − 4	−1.94510e − 4	0.8	8.0173e − 4	8.0173e − 4
[−6e − 4; −1e − 11]	−1.97535e − 4	−2.03024e − 4	0.8	8.0239e − 4	8.0239e − 4
[−7e − 4; −1e − 11]	−2.01636e − 4	−1.98715e − 4	0.8	8.0145e − 4	8.0145e − 4
[−1.5e − 2; −1e − 11]	−2.18090e − 4	−1.82672e − 4	0.8	8.0932e − 4	8.0932e − 4

Table 3
Global minimum solution when optimisation is applied and the panels are moving

p_1	p_2	u_1	u_2	x_t	ϕ_s	ϕ
0.27	0.48	$-1.995460e-4$	$-1.994041e-4$	0.800	$7.95806e-4$	$7.95806e-4$

Table 4
The optimal solution when both the golden section search method and modified complex method are applied

p_1	p_2	u_1	u_2	x_t	ϕ_s	ϕ
0.27	0.48	$-1.99546e-4$	$-1.994041e-4$	0.800	$7.95806e-4$	$7.96806e-4$

did not exceed 2 h for a chosen set of bounds on a 500 MHz PIII processor.

8.2. Finding the optimal positions of two panels

For this task, we first need to know where the actual global minimum located as this will allow us to estimate how good our optimisation actually is. For this purpose, the positions of the panels were changed by increments of 0.01 m and optimisation was applied for each non-overlapping pair (p_1, p_2). Table 3 gives the resulting global minimum solution (for this set-up).

Clearly this way of finding the global minimum is very heavy computationally and is useful only for comparison purposes. The results of applying the golden section search algorithm together with modified complex optimisation are given in Table 4 (where after each calculation the panel positions were rounded up to two digits after decimal point). The results in Table 4 are identical to those in Table 3, but the computational load is reduced.

Fig. 4 shows the variation of the cost function with the iteration count. Clearly, a good if not optimum solution is found relatively quickly, with further, slow, refinement later in the run. Similar plots were obtained in all other cases and are omitted here for brevity.

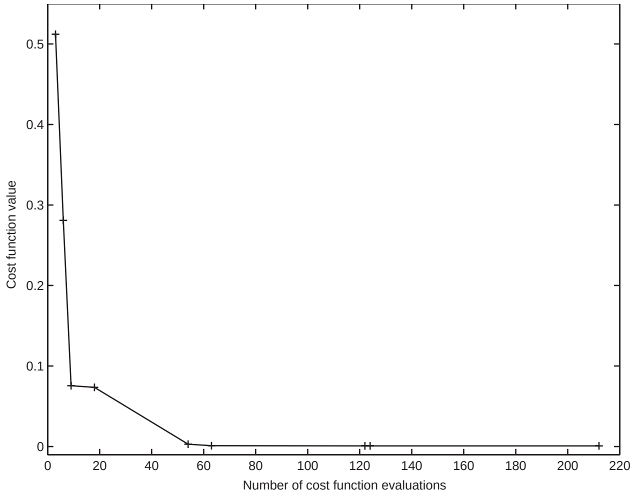

Fig. 4. Best value of the cost function as a function of the number of iterations.

9. Results: linear distributed suction flow rates on an aerofoil

In this section the different methods developed in Section 7 are compared for the representative cases of 5 and 10 non-overlapping panels fixed at 20% of the chord with linear continuous suction flow rate distributions for 2D flow. In particular, the three solution algorithms developed in Sections 7.2, 7.3 and 7.4 respectively above are compared with the random search approach (Section 7.1). The parameters α, β and γ were set at 1.9, 0.29, and 1.1, respectively, throughout these comparisons, and the thresholds $tol_1 = 0.15$ and $tol_2 = 0.5$ were set for multi-start points selection. Initial feasible sets for the flow rates were taken to be in the range $[-3.7e-4, 0]$.

9.1. 5 panels case

Here random search (RS), parallel global optimisation based on the modified deformed configuration method (PGO), multi-start parallel global optimisation with pattern search for local search (MPG), and a combination of random search and multi-start parallel global optimisation (CRS) are compared. For RS, $N_1 = 8190$ and $N_2 = 3567$ were chosen. Also the set S_0 contained 2900 initial points, and the set S contained 10 multi-start points for MPG. In the case of CRS, $N_1 = 3567$, S_0 contained 2900 points and S contained 13 multi-start points. The results obtained are given in Tables 5 and 6. Note that u_1 has been set to zero in this work since immediate vicinity of the leading edge of the aerofoil, where this suction effort would be applied, the flow is accelerating and stable to disturbances of the type considered here.

In Table 6, it is the elapsed rather than the total computational time which is given, where the former is

Table 5
The near optimal flow rates for 5 panels

Method	u_1	u_2	u_3	u_4	u_5	u_6
RS	0	−6.66254e−5	−3.444661e−4	−3.134907e−04	−8.161225e−5	−2.872861e−5
PGO	0	−1.5e−5	−1.1e−5	−9.0e−5	−3.6999e−4	−1.9683e−4
MPG	0	0	−4.32388e−5	0	−3.67273e−4	−1.98027e−4
CRS	0	−1.580755e−4	−4.168359e−5	9.0e−5	−3.442298e−4	−3.065454e−4

Table 6
The minimum overall cost functions obtained for 5 panels

Method	x_t	ϕ_s	ϕ	Time (h)
RS	0.799783	2.288582e−3	4.458582e−3	19.15
PGO	0.8	1.8408e−3	1.8408e−3	30.73
MPG	0.80003	1.75964e−3	2.05964e−3	19.47
CRS	0.7999722	2.472896e−3	2.750896e−3	19.81

much larger due to the parallel structure of the solution method. The number of processors used varied, and was adjusted to suit the problem and make best use of the resources available, by 'load balancing' the processor array so that there was very little idle time. For example, in the RS case, 22 processors where used for the first stage of the ($N_1 = 8190$—1 master processor and 21 workers) and 30 for the second stage ($N_2 = 3567$), while for CRS 15, 15 and 4 processors were used for the three stages, respectively.

As can be seen from the Table 6, PGO provides the best solution, but also requires almost twice as much time to obtain a solution compared to the RS, MPG and CRS methods. Moreover the complexity of PGO method, and execution time required as result of this, grows exponentially with increasing optimisation problem dimension. The MPG method suggests itself as a good compromise between accuracy and computational load, and in comparison CRS method is worse in both these respects. However, the performance of these two algorithms depend on the tuning parameters, the type of problem considered and the dimension of the resulting optimisation problem. Finally, it is to be expected that the CRS method will out-perform the MPG approach in terms of accuracy.

9.2. 10 panels case

Here the RS, MPG and the CRS methods are compared for a case with 10 panels. The reason why PGO is not included that it is too slow for 11D optimisation problem which has to be solved in this case. For RS $N_1 = 8190$ and $N_2 = 8961$ were chosen. The set S_0 contained 9984 initial points, and the set S contained 13 multi-start points for MPG. In the case of CRS $N_1 = 8190$, the set S_0 contained 9984 points, and the set S contained 20 multi-start points.

The results of the comparison are given in Tables 7–9, respectively. It can be seen from Table 9 that CRS method produces the best solution, as expected. However, it requires the longest computational time and hence the preferences of users or/and the type of the application problems considered will decide which one of the proposed methods to choose for use for the underlying global optimisation problem. Note also that, since all three methods have a degree of randomness associated with them, it is not guaranteed that the results will be the same if we start from different initial data sets. This is most relevant for the RS since the other two attempt to find the global or near optimal minimum.

The values given in Table 9 for the cost function values are larger than those given in Table 6 for the case of 5 panels. This is deceptive in that the cost function takes no account of the length of the panels. To obtain a direct comparison, the values in Table 9 should be divided by two. This shows that shorter panels produce solutions with lower total suction effort, as would be expected on physical grounds.

10. Conclusions

Turbulent flow has a significantly higher drag than the corresponding laminar flow at the same flow conditions. The presence of turbulent flow over a large part of an aircraft therefore incurs a significant penalty of increased fuel consumption due to the extra thrust required. One possible way of decreasing the drag is to apply surface suction to delay the transition from laminar to turbulent flow. However, in order for the gain from the reduction in drag to outweigh the extra costs associated with the suction system, the suction must be distributed in an optimum, or near optimum, manner. The results presented here show that the modified complex method developed in this paper applied with a golden section search method can be used to optimise the panel positions and suction distribution for a two panel system. Although this approach is relatively efficient, there is still a trade off between finding the optimal solution and the computational load. However, it is clear that if the aim is to find a 'good' solution, i.e. one that is close to the optimal

Table 7
The near optimal flow rates for 10 panels

Method	u_1	u_2	u_3	u_4	u_5	u_6
RS	0	−4.517587e − 5	−6.48977e−5	−6.626079e−05	−4.143352e − 5	−8.269108e−5
MPG	0	−4.517458e − 5	−8.147645e−5	−2.293512e−4	−1.702706e − 4	−6.660711e−5
CRS	0	−1.136449e − 4	0	0	−3.991312e − 5	0

Table 8
The near optimal flow rates for 10 panels—cont'd

Method	u_7	u_8	u_9	u_{10}	u_{11}
RS	−4.846197e − 5	−1.02092e−4	−1.354573e−4	−3.570386e−04	−2.767380e−4
MPG	−3.352883e − 4	−3.423307e−4	0	−7.614603e−5	0
CRS	−3.623112e − 4	−2.624833ee−4	0	0	0

Table 9
The minimum overall cost functions for 10 panels

Method	x_t	ϕ_s	ϕ	Time (h)
RS	0.8000698	2.54378e−3	3.24178e−3	16.80
MPG	0.7999865	3.301165e−3	3.4362e−3	18.20
CRS	0.8000892	2.146751e−3	3.03875e−3	31.53

solution with a significant reduction of the cost, rather than the optimal solution, then this can be done quickly.

In case of an aerofoil with a linear continuous suction distribution over its front part, the computational load increases significantly, and in some cases it is not possible to continue the investigation using a single processor. Therefore several parallel global optimisation solution algorithms have been developed in this paper. For the representative case of 5 panels, the parallel global optimisation method based on modified deformed configuration method (PGO) produced the best solution. However, this method requires 50% more computational time than random search method (RS) and multi-start parallel global optimisation method with pattern search for local search (MPG). Moreover, with an increase in dimensionality, the computational complexity of the PGO method will increase exponentially. Therefore for case of 10 panels we consider only RS, MPG and a combination of random search and multi-start parallel global optimisation (CRS). The results here showed that as expected CRS method gave the best solution in relative terms but it required more computational time. Therefore there is still a trade off between finding the global minimum and the computational load and choice of the global optimisation approach will depend on preferences of user and/or an application problem considered.

Finally we note that the methods developed here have major advantages over those described in earlier work in that they can obtain solutions in cases where simpler gradient based methods fail (Tutty et al., 2000b) but are much more efficient than methods based on stochastic optimisation (Dodd et al., 2001; MacCormack et al., 2002). Further, for the problems considered in MacCormack et al. (2002) which had a similar number of variables to be determined as part of the solution, it was necessary to severely restrict the values the suction flow rates could take in order to obtain a relatively crude solution in a reasonable time.

References

Dodd, T. O. R., Tutty, & Rogers, E. (2001). Genetic algorithm based constrained optimisation for laminar flow control. *Proceedings of the European control conference 2001* (*ECC'01*), Porto, September 2001, pp. 2970–2974.

Fletcher, R. (1987). *Practical methods of optimization*. Chichester: Wiley.

Horst, R., & Pardalos, P. M. (Eds.). (1995). *Handbook of global optimization*. Dordrecht/Boston/London: Kluwer Academic Publishers.

Kolda, T. G., Hough, P. D., & Torczon, V. J. (2001). Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing*, *23*, 134–156.

MacCormack, W., Tutty, O. R., Rogers, E., & Nelson, P. A. (2002). Stochastic optimisation based control of boundary layer transition. *Control Engineering Practice*, *10*, 243–260.

Mack, L. M. (1977). *Transition prediction and linear stability theory*. AGARD CP-224, 1.1.

Mack, L. M. (1984). *Boundary layer stability theory*. AGARD Report 709, 3.1.

Malik, M. R. (1990). Stability theory for laminar flow control design. *Progress in Aeronautics and Astronautics*, *123*(3).

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*(4), 308–313.

Nelson, P. A., Wright, M. C. M., & Rioual, J.-L. (1997). Automatic control of laminar boundary-layer transition. *AIAA Journal*, *35*, 85–90.

Rioual, J.-L. (1994). *The automatic control of boundary layer transition*. Ph.D. thesis, University of Southampton.

Rykov, A. (1995). Construction principles of deformed configurations methods. *Preprints of the summer school course on Identification and Optimization oriented for use in adaptive control*, *Prague*, *Czech Republic* (pp. 65–80).

Schmid, P. J., & Henningson, D. S. (2001). *Stability and Transition in Shear Flows*. New York: Springer.

Thiede, P. (Ed.). (2001). Aerodynamic Drag Reduction Technologies. *Proceedings of the CEAS/DragNet European drag reduction Conference*, Potsdam, Germany, June 2000. Berlin: Springer.

Torczon, V. (1991). On the convergence of the multidirectional search algorithm. *SIAM Journal on Optimization*, *1*(1), 123–145.

Tutty, O. R., Hackenberg, P., & Nelson, P. A. (2000a). Numerical optimisation of the suction distribution for laminar flow control. *AIAA Journal*, *38*, 370–372.

Tutty, O. R., Hackenberg, P., & Nelson, P. A. (2000b). Gradient projection based control and optimisation of the suction distribution for laminar flow control. *Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering*, *214*, 347–359.

Veres, G. V., Tutty, O. R., Rogers, E., & Nelson, P. A. (2003a). On design optimisation based control methods for distributed suctions for boundary layer transition control. *Proceedings of 2003 American control conference*, Denver (pp. 2187–2192).

Veres, G. V., Tutty, O. R., Rogers, E., & Nelson, P. A. (2003b). Parallel global optimisation based control of boundary layer transition. *2003 European control conference*, Cambridge, UK, CD-Rom proceedings.

Wright, M. C., & Nelson, P. A. (1999). Four-channel suction distribution optimization experiments for laminar flow control. *AIAA Journal*, *38*, 39–43.