Abstract—About forty years ago Rechenberg and Schwefel (Rechenberg, 1994) came up with the idea of evolution strategies for flow optimization. Since then advances in computer architectures and numerical algorithms have greatly decreased computational costs of realistic flow simulations, and today Computational Fluid Dynamics (CFD) is complementing flow experiments as a key guiding tool for aerodynamic design. Of particular interest are designs with active devices controlling the inherently unsteady flow fields, promising potentially drastic performance leaps. Here we demonstrate that CFD-based design of active control strategies can benefit from evolutionary computation. We optimize the flow past an actively controlled circular cylinder, a fundamental prototypical configuration. The flow is controlled using surface-mounted vortex generators; evolutionary algorithms are used to optimize actuator placement and operating parameters. We achieve drag reduction of up to 60 percent, outperforming the best methods previously reported in the fluid dynamics literature on this benchmark problem.

1 Introduction

The flow past a circular cylinder is a well established prototypical configuration of bluff body flows (Roshko, 1955). Here, for the first time, we use evolutionary algorithms in conjunction with CFD to achieve drag reduction in active flow control through modification of the tangential velocity on the cylinder surface.

Several control methodologies have been suggested to modify vortex shedding behind a circular cylinder and to effectuate drag reduction, either with passive geometrical modifications and/or with an open loop steady forcing. A non-exhaustive list includes studies of the effect of a splitter plate attached to the cylinder studied, among others, by (Kwon and Choi, 1996; Unal and Rockwell, 1987; Apelt and West, 1975; Apelt et al., 1973). These studies achieved drag reductions from 20 to 40% depending on the arrangement of the control devices. A free rotating cylinder attached to a splitter plate was considered in (Cimbala and Garg, 1991). The placement of a small secondary cylinder in the wake was shown to suppress the vortex shedding while reducing the drag coefficient (Strykowski and Sreenivasan, 1990). Another possible action is base suction/blowing (Shumm et al., 1994). Rotational oscillations were studied by (Tokumaru and Dimotakis, 1991; Chen et al., 1993). At Reynolds numbers $Re = 15000$ ($Re$ indicates the degree of flow turbulence) and for certain parameters, found after extensive experimentation, this control action was found to induce dramatic changes in the wake and a significant drag reduction of about 60%. Due to experimental limitations, however, it was not possible to reveal the underlying governing mechanisms.

While a wealth of different possible geometry modifications and/or open loop actions have been devised and studied, little effort has been devoted to algorithms optimizing the various control parameters. Recent CFD studies of suboptimal control (Lee et al., 1998) have tried to remedy this situation for simulation of flows using mass traspiration as an actuation mechanism. Drag reduction of up to 20% was achieved using this optimization strategy for idealized blowing/suction actuators.

In practice, however, it remains an almost impossible task even to cast the optimization problem in a tractable explicit analytical form for all different types of sensor and actuator devices. The dependence of the control on the type of actuators also minimizes the generalization of certain successful control strategies. Hence, most present day design strategies rely mainly on trial and error experiments, driven by human intuition derived from physical understanding of the phenomenon. This approach has the fundamental limitation that most of this understanding is based on simulations and experiments of the uncontrolled flow. Moreover when the placement of the actuators needs to be optimized the cost function can be evaluated only for a certain actuator arrangement and hence no analytical objective function can be pre-
scribed \textit{a priori} in order to be optimized with gradient based techniques. This motivates our interest in automatic optimizers of the \textit{controlled} flow that compactly encode information about desired flow behavior in their objective function.

We conduct two-dimensional simulations for Reynolds numbers $Re = 500$ and $Re = 1000$.

Although the uncontrolled flow past a circular cylinder at these $Re$ is three-dimensional, experiments by (Tokumaru and Dimotakis, 1991) for a cylinder undergoing rotating oscillations indicate that the controlled flows and the underlying physical mechanisms are strongly two-dimensional.

The paper is organized as follows: Section 2 presents flow equations, related numerical parameters of the flow solver and the calculation of the drag coefficient which constitutes the objective function to be minimized. In Section 3 we consider an array of actuators placed on the cylinder surface and a genetic algorithm that incorporates a mutation probability dependent on the distance from a prescribed target value of the objective function. Section 4 describes an alternative heuristic approach which generates drag-reducing control parameters based on previously observed drag values. Our methods will yield unprecedented drag reduction.

2 Governing equations and numerical method

We consider two-dimensional incompressible viscous flow past a circular cylinder. Governing equations are the Navier-Stokes equations in the velocity-pressure formulation

$$\frac{dv}{dt} + (v \cdot \nabla)v = -\frac{1}{\rho} \nabla P + v \nabla^2 v$$

$$\nabla \cdot v = 0$$

where $v$ is the velocity vector, $P$ is the pressure and $\rho$ is the density of the flow and $\nu$ denotes the kinematic viscosity.

The boundary conditions are defined as:

$$v(x, t) = V_{ext} \quad \text{on the cylinder surface}$$

$$v(x, t) = U_{\infty} e_x \quad \text{as } |x| \to \infty$$

where $V_{ext}$ is the surface velocity induced by the actuators, and $e_x$ is the unit vector in the stream-wise direction. $U_{\infty}$ denotes the free-stream velocity. The Reynolds number and Strouhal frequency of the flow are defined as:

$$Re = \frac{U_{\infty} D}{\nu} \quad St = f D \frac{1}{U_{\infty}}$$

where $D = 2R$ is the diameter of the cylinder and $f$ the shedding frequency of the flow.

We also define a normalized time, by scaling time as follows:

$$t^* = \frac{t}{t_a} = \frac{t}{(D/U_{\infty})}$$

where $t_a$ is the time taken by a fluid particle in the free stream to be advected past the cylinder. The definition of such normalized quantities allows a ready generalization of the results to differently scaled geometries.

3 Tangential actuation optimization

3.1 Problem setup

We consider flow at $Re = 500$, past a cylinder whose surface is subdivided in $n = 16$ equal size segments that are allowed to move tangentially to the cylinder surface, each with a different but steady velocity.

This configuration can be physically realized by using a set of moving belts covering the surface of the cylinder. It has been found experimentally by (Bechert et al., 1996) and (Modi, 1997) that such actuators are very efficient in achieving significant drag reduction. Fig.1 shows the location of the belts on the cylinder surface for the case study considered here.

An off-line optimization is performed here for the functional in Eq. 5; what we optimize is the time average of the drag in a fixed time interval. The average drag is computed at each simulation step, starting from a steady uncontrolled situation and averaging over 4 Strouhal periods.

We use a genetic algorithm (GA) (Holland, 1975; Goldberg, 1989) that was specifically designed to find a basin yielding fitness values below a given threshold (Milano and Barone, 1999).

We study whether this GA can automatically identify critical points such as the separation points of the uncontrolled flow. For $Re = 500$ these points are encompassed by segments 4 and 13.

3.2 GA Description

The GA used in this paper is a modification of Controlled Random Search (CRS) (Price, 1976). The algorithm is outlined in the following paragraphs.

Let $G(\lambda)$ the function to be minimized, $\lambda \in \mathbb{R}^n$ the parameter vector. In a first phase $S$ population points are initially randomly chosen according to a uniform distribution.
Figure 1: Location of the actuator belts on the cylinder surface. The belts are numbered as specified in the figure; also a snapshot of the uncontrolled vorticity is reported: the flow is from left to right.

within a defined search volume of dimension $n$. The inequality $S \gg n$ must hold for the algorithm to work properly. Then the algorithm proceeds as follows:

1) Choose the population point $\lambda_{max}$ in which $G$ reaches the maximum value:
   
   $\lambda_{max} = arg \{ \max_{i=1,\ldots,M} G(\lambda_i) \};$
   
   $G_{max} = G(\lambda_{max});$

2) Randomly choose $n + 1$ different population points:
   
   $\lambda_1, \ldots, \lambda_{n+1}$ (breeding set). All subsequent operations are performed on this set;

3) Mutation step: for all breeding set points, with probability
   
   $P_i = (1 - \alpha') \cdot \left(1 - \beta \frac{G(\lambda_i) - G_{max}}{G_{max}} \right) \cdot \gamma,$
   
   replace the point $\lambda_i$ with one randomly chosen within the search volume limits;

4) Recombination step: for each of the $n + 1$ points determine the centroid, $\lambda_\bar{x}$, of the other $n$ points, i.e.
   
   $\lambda_\bar{x} = \frac{1}{n} \sum_{j=1}^{n} \lambda_j;$

4.a) Generate offspring $\lambda_{si} = 2\lambda_\bar{x} - \lambda_{n+1}$; if $\lambda_{si}$ is not contained in the search volume, process next point in breeding set;

4.b) Calculate $G(\lambda_{si})$: if $G(\lambda_{si}) < G_{max}$ then purge $\lambda_{max}$ from population, and substitute by offspring $\lambda_{si};$

5) Compute the new $G_{max}$, if necessary;

6) Iterate steps 4 and 5 on the whole breeding set;

7) If convergence test fails, return to step 1.

There are 4 parameters, i.e. $\alpha, \beta, \gamma, G_T$ and 2 variables: $I$ and $G$ to be defined in this scheme. The variable $I$ is the number of consecutive iterations in which the population has not changed, i.e., no offspring has substituted any population member. It provides an empirical measure of the need for fresh information through increased mutation probability. The variable $G$ is the average population fitness, used as a scaling factor.

The most important parameter is $G_T$, a threshold $C_D$ value used for the convergence test: we declare convergence once all of the population's fitness values are below it. Given this criterion, population points will be clustered inside the domain defined as: $\{ \lambda | G(\lambda) < G_T \}$. The final cluster provides a sampling of the minimum basin. This allows to retrieve parameter correlations and other meaningful quantities.

According to the formula defining mutation probability, the parameter $0 <= \alpha <= 1$ modulates the mutation rate during the course of the optimization process, and the parameter $0 <= \gamma <= 1$ enforces its upper bound, since $0 <= \beta <= 1$. The term containing the parameter $\beta$ causes population members far from convergence (with fitness > $G_T$) to mutate more frequently. (CRS can be regarded as a GA with zero mutation probability.)

3.3 Results with the GA

For a GA population of 50 elements the parameters $\alpha$ and $\beta$ were fixed to 0.25, the upper bound $\gamma$ to 0.02 and the threshold $G_T$ to 0.7. This value is about 50% of the drag coefficient reported for and uncontrolled flow around a cylinder at $Re = 500$ (Panton, 1996). The optimization lasted for 1500 iterations, corresponding to about 30 hours of CPU-time on a NEC SX-4 supercomputer.

On a grid twice as fine as the one used for optimization,
a validation run was performed for the best candidate in the population. Fig. 2 plots the behavior of the drag coefficient during the transition from the uncontrolled and controlled phase. The transition phase is quite short, and the flow appears to settle quickly to the minimal drag configuration. The shedding frequency is drastically modified while the fluctuating amplitude in the drag coefficient is significantly reduced.

Fig. 3 shows a histogram of the converged population. We observe that the GA discovered that segments on opposite sides of the cylinder must rotate in opposite directions, in order to delay separation by allowing the flow to “slide” further on the cylinder surface.

This separation delay also becomes obvious by comparing the time averages of the controlled and uncontrolled vorticity contours, shown in Figs 4 and 5, respectively. The controlled wake is more elongated, a clear evidence of a later separation, resulting in a smaller drag.

4 Cylinder in rotary oscillations

A second set of CFD experiments was performed to optimize the parameters (amplitude, frequency) of rotary oscillations of a circular cylinder. This may be viewed as a special case of the previous experiments: all segments on the cylinder move with equal velocity.

In this case optimization was performed online, based on the same function in Eq. 5.

4.1 Problem setup - Parametrization

We consider flow at $Re = 1000$, undergoing rotary oscillations, with the tangential velocity on the surface of the cylin-
Figure 6: Evolution of the control parameters and of the drag, plotted from the onset of the algorithm.

\[ v_{slip} = A \sin(8 \pi \omega t / St) \]

where \( A, \omega \) denote amplitude and frequency of the rotary oscillations and \( St \) denotes the Strouhal frequency of the uncontrolled flow. The grid size used in these experiments is 80 x 160 (radial and azimuthal directions). The non-dimensional time step used in the numerical simulations is 0.003. A typical optimisation run takes about one hour of CPU time on a NEC SX-4 supercomputer.

4.2 Empirical online optimisation

One of the biggest problems with online optimisation of bluff body flow control are the large transients after control initiation. During the flow transient the drag can fluctuate a lot before the flow stabilises. This implies that any online algorithm building on past experience has to trade off the two goals of reactivity and insensitivity to transients.

To achieve this trade-off we make an analogy between optimisation in parameter space and dynamics due to a force field. To generate a force field based on the results given by the process so far, the general guideline is that good examples attract further points while bad examples push them away.

In our experiments, the force is following the line between the two parameter points, and it scales like \( ||1/\Delta \theta|| \), where \( \theta \) denotes the vector \((A, \omega)\). We define the force to be

\[ F(\theta, \{j\}) = \frac{1}{||j||} \sum_j F_j(\theta) \]

and

\[ F_i(\theta) = \frac{(C_{D,i} - \bar{C}_D)(\theta - \theta_i)}{||\theta - \theta_i||^2} \]

where the sum is taken over the set of values \( \{j\} \). For instance \( \{j\} = 1, \ldots, T \) means all the drag values observed so far. \( \bar{C}_D \)

is the mean of the drag values, \( \theta \) and \( \theta_i \) are base point and past point at instant \( i \) in parameter space, respectively.

Choices that have to be made include the mappings between force and values of the optimised function, the determination of a base point for update and the update rate. Here we take as coefficient the drag minus its average, as calculated by \((C_{D,i} - \bar{C}_D)\) in Eq. 8. This choice ensures that the long range force field component is zero. After a coarse experimental search the rate \( \alpha \) was set to 0.001. To summarize, the parameters are updated as

\[ \theta_{i+1} = \theta_i + \alpha F(\theta_i, \{1, \ldots, i-1\}) \]

Figure 7: Vorticity contour plots at \( Re = 1000 \), uncontrolled case. (a) Average, (b) Snapshot from the animation.

4.3 Results of online optimization

Parameters were randomly initialized in a large area suspected to contain parameter combinations corresponding to drag reduction, following previous studies (Koumoutsakos, 1993).

The algorithm indeed discovers parameters leading to significant drag reduction (see Fig. 6). In the end of the simula-
This mechanism is responsible for drastic drag reduction.

5 Conclusion

Two automatic optimizers have been applied to the problem of active flow control past a cylinder, both yielding unprecedented drag reduction of up to 60%.

Drag was reduced by two important mechanisms, namely: (a) delayed flow separation (discovered by off-line GA optimization – Section 3) and (b) drastic modification of vortical structures through formation and ejection of vortex dipoles at separation points (discovered by on-line heuristic optimization – Section 4).

An important aspect of our particular GA is the dependence of the mutation probability on the distance from a desired minimum.

Our results show for the first time that excellent solutions to benchmark problems of Computational Fluid Dynamics (CFD) can be obtained through evolutionary computation (EC). Since CFD avoids costly real-world experiments, CFD/EC combinations may become an important tool for aerodynamic designs employing on active control devices. Future work will focus on speeding up convergence of the optimizers and on three dimensional flow simulations.

Acknowledgements

This research is funded by the Swiss National Science Foundation grant No 2100-054093.98. Computational support was provided by the Swiss supercomputing center CSCS in Manno.

Bibliography


