Cylinder drag minimization through wall actuation: a Bayesian optimization approach

Anthony Larroque[a], Miguel Fosas de Pando[a], Luis Lafuente[b]

[a]*Departamento de Ingeniería Mecánica y Diseño Industrial, Escuela Superior de Ingeniería, Universidad de Cádiz, 11519 Puerto Real, Spain*
[b]*Departamento de Matemáticas, Escuela Superior de Ingeniería, Universidad de Cádiz, 11519 Puerto Real, Spain*

Abstract

Bayesian Optimization (BO) has recently gained popularity as an efficient derivative-free method for the global optimization of expensive noisy black-box objective functions. These characteristics render BO a promising tool to tackle optimization problems involving numerical simulations of complex unsteady flows at moderate-to-high Reynolds numbers. In this work, we assess the efficiency of Bayesian Optimization by considering two canonical flow problems: the drag reduction in the two-dimensional and three-dimensional flow around circular cylinders at, respectively, Re = 500 and Re = 3900, through tangential-velocity actuation at the cylinder wall. The root-mean-square of the drag coefficient with and without penalty terms is considered as the objective function. Several variants of Bayesian Optimization are assessed and compared against competing optimization algorithms such as Particle Swarm Optimization, CMA-ES, Nelder-Mead and the Explorative Gradient Method. Results show that in this case, the serial and the parallel BO techniques outperform other algorithms.

Keywords: Bayesian Optimization, Drag reduction, Cylinder, Optimization methods, Computational Fluid Dynamics

1. Introduction

Owing to the constantly increasing available computational power, High Performance Computing and high-fidelity simulations are progressively being integrated into the design and optimization of engineering devices. The improvement of a given design often resorts to the solution of an optimization problem. In particular, the simulations are parameterized and the set of all possible choices for these parameters, i.e. the design space, is defined. Then, the goal is to find the point in the design space that corresponds to the *best design*, as quantified by the extremum of an objective function. To find the solution, optimization algorithms (1) repeatedly evaluate different design choices, leading to a sequence of points that converges to this extremum.

In the case of devices involving flows, high-fidelity simulations of unsteady complex flows at moderate-to-high Reynolds numbers using Detached or Large Eddy Simulation are becoming commonplace. However, the increased level of detail that such simulations provide is associated with a dramatic increase in computational cost. This observation puts severe restrictions into optimization studies. First, the cost of solving an optimization problem is comparable to the product of the cost of a single simulation and the number of function evaluations. Typically, the available computational resources translate into a modest number of function evaluations that can be performed, i.e. a *budget*. Second, the flow features that are present in high-fidelity simulations often result into objective functions with complex landscapes where a large number of local extrema are present. For these reasons, solving optimization problems involving high-fidelity simulations requires the use of optimization methods that can escape from these local extrema and find the global extremum in few evaluations.

Email address: anthony.larroque@uca.es (Anthony Larroque)

Although several classifications exist for optimization methods (e.g. local/global, constrained/unconstrained, deterministic/stochastic, single-objective/multi-objective), we will classify them here into gradient-based and derivative-free methods.

Gradient-based methods use derivative information to determine the direction in the design space where the objective function has maximum local growth. These methods have the advantage of benefiting from fast convergence rates to an extremum. However, they may often be trapped in local optima and are difficult to parallelize efficiently. Furthermore, obtaining accurate gradients from sophisticated numerical solvers can be problematic. As the number of parameters increase, computing the gradients through the sensitivity equations or finite differences becomes increasingly more expensive. To tackle this problem, Jameson (2) pioneered the use of adjoint methods where the cost of obtaining the gradients is independent of the number of design parameters. A major advantage of such methods is that they provide sensitivity maps at no additional cost. For instance, Giannetti and Luchini (3) and Marquet *et al.* (4) developed theoretical frameworks that allow to determine regions in the flow that are more sensitive to external forcing and base-flow modifications, respectively. In the context of flow control, Camarri and Iollo (5) applied sensitivity analysis to design a feedback control to suppress vortex shedding in the wake of a square cylinder. However, adjoint methods have several drawbacks: they require a significant development effort, and in the case of unsteady computations, long integration times translate into large memory requirements, exceeding typically available storage capacity. Among these limitations, perhaps the most important one is that they run into complications with turbulent flows. As illustrated by Talnikar and Wang (6), the adjoint solution tends to diverge for long-term averaged quantities derived from flows with chaotic features—even if the statistics are well defined—resulting thus into large errors in the gradient.

On the other hand, derivative-free methods do not rely on gradient information, and instead, use only the information acquired through function evaluations. For example, the Nelder-Mead algorithm (7) relies on contraction, reflection, expansion and shrink of simplices in the design space. Recently, this algorithm has been combined with Latin Hypercube Sampling (LHS) and applied to the stabilization of the so-called fluidic pinball and drag reduction of the Ahmed body through steady blowing (8). Algorithms inspired by the observation of nature, such as natural selection (9), bird flocking (10), and metal annealing (11) have also been proposed. They generally introduce a population that will evolve towards the optimum. Although these methods suffer from lower convergence rates, they are often able to escape from local optima, are easier to parallelize and can be easily wrapped around general black-box functions. These properties render these methods very attractive compared to the gradient-based counterparts. Unfortunately, the large number of function evaluations that derivative-free methods typically require precludes them from being routinely used in optimization problems involving high-fidelity simulations. This is further aggravated when the number of design parameters increases. For this reason, when robust gradient information is available, gradient-based methods are often preferred. Examples of optimization problems involving numerical flow simulations are (12; 13; 14; 15) for genetic algorithms and (12; 16; 17) for Particle Swarm Optimization.

Typically applied within a derivative-free framework, the Response Surface Methodology (RSM) (18) is a widely used approach to optimize functions that are expensive to evaluate. In contrast to the methods mentioned above, RSM attempts to use the value of the objective function at selected design points to build a model that approximates the value of the objective function but is cheaper to evaluate. To build an accurate model over the entire design space, a careful sampling in the design space, known as the Design of Experiment (DOE), is performed first. Once the model is built, optimization methods rely onto this model and not directly onto the objective function, in order to determine the next design to evaluate. Then, this point is evaluated and added to the surrogate surface to improve its accuracy. Since the model is cheap to evaluate, the cost of finding the next candidate point is often negligible. Then, this process is repeated

until a stop criterion is fulfilled. The main drawback of using this framework is that the accuracy of the model is difficult to estimate *a priori*, and it degrades quickly as the number of design parameters increases.

Bayesian Optimization (BO) (19; 20) has recently gained popularity as an effective derivative-free optimization method for expensive objective functions. Once a DOE has been performed, a surrogate model typically based on Gaussian Process (GP) is introduced. Then, the sequence of points that leads to the optimum is guided by (i) the value of the objective function at the optimum of the surrogate model (ii) the uncertainty of the model. This can be interpreted as a trade-off between ensuring that (i) the optimum is found and (ii) that the model is accurate. Interesting features of the resulting strategy are the ability to deal with black-box functions, handle uncertainty and noise, and the possibility to reach a global optimum in few function evaluations. These characteristics render this method a valuable candidate for solving optimization problems involving high-fidelity numerical flow simulations.

The modern BO approach was probably pioneered by Kushner (21), who was interested in finding the maximum of a noisy function. With Brownian motion stochastic processes (or Wiener processes) selected as a model of the noisy function, an auxiliary (or acquisition) function called Probability of Improvement (PI) was used to determine the maximum of unconstrained one-dimensional problems. Later, Mockus *et al.* (22) extended the BO approach to multidimensional problems using an alternative acquisition function named Expected Improvement (EI). In 1951, Krige (23) developed a statistical technique for mine valuation. The idea of this technique is to model the objective function as the realization of a stochastic process. A mean function is then used to model the most probable value of our objective function at every point whereas the standard deviation acts as an estimation of the uncertainty of the model. Later, Sacks *et al.* (24) applied this model to the Design of Analysis and Computer Experiments (DACE) to approximate the deterministic output of numerical simulations and provide an efficient way of choosing the inputs for prediction purposes. In 1998, Jones (25) introduced a methodology named Efficient Global Optimization (EGO) that combined DACE with EI to deal with expensive black-box functions. Since then, several variants of this method have been developed to deal with constraints (26) or to address stochastic objective functions (27). A review of recent developments of this method can be found in (20) and in (28).

Regarding applications to numerical flow simulations, Jeong *et al.* (29) used the Kriging model and the Expected Improvement to maximize the lift-to-drag ratio of a two-dimensional airfoil with 10 design parameters. Later, Duvigneau and Chandrashekar (30) applied BO to drag minimization of flow around a three-dimensional rotating cylinder considering the amplitude and frequency of oscillations as the design parameters. In this study, the method was also applied to a two-dimensional airfoil where four design parameters determined the shape of a protrusion. The optimal solution obtained for the rotating cylinder was the same as in the literature, whereas in the case of the airfoil, a better optimum than the one reported in the literature was found. Contrarily to these studies, which used RANS simulations, Talnikar *et al.* (31) developed a parallel BO approach for Large-Eddy Simulations (LES). The EI criteria was modified to determine at each iteration several promising points, i.e. one per parallel process. The resulting algorithm was successfully applied to the drag reduction in a turbulent channel and the minimization of the heat transfer and pressure coefficient on a turbine blade. More recently, Mahfoze *et al.* (32) applied BO to reduce the skin-friction drag of spatially evolving turbulent boundary layers simulated through DNS. Lam *et al.* (33) reviewed two BO methods to deal with a finite budget and to include gradient information through the adjoint equations for aerospace engineering applications. There is thus an increasing interest in this methodology, mainly due to its increasing popularity in machine learning, and its application to unsteady flow simulations could provide an effective way to improve engineering devices and gain physical insight into the effect of external parameters. However, to the best of our knowledge this technique has not been compared to alternative techniques in the context of numerical flow simulations.

Reducing the vortex shedding behind bluff bodies and the drag exerted on these structures through active and passive

control methods is one of the many applications that can benefit from advances in optimization techniques. Here, we focus on the flow around a circular cylinder to assess the efficiency of Bayesian Optimization in numerical flow simulations. Among the many existing control methods applied to the cylinder (34), one of particular interest consists in introducing a velocity profile at the cylinder surface that mimics the effect of a distributed array of actuator jets. For instance, Li *et al.* (35) used a blowing–suction mechanism in a two-dimensional cylinder to reduce the vortex shedding for Reynolds numbers, Re, up to 110. The adjoint equations and the Davidon-Fletcher-Powell (DFP) quasi-Newton method were used to find the optimal 18 design parameters related to the blowing–suction mechanism. Milano and Koumoutsakos (13) used evolutionary algorithms to determine the optimal amplitude of 16 actuators for drag minimization of a two-dimensional circular cylinder at Re = 500. Catalano *et al.* (36) minimized the drag coefficient of a two dimensional cylinder at Re = 500 and Re = 3900 using an actuator jet. A response surface method was used to find the optimal frequency and position of the jet. More recently, Meliga *et al.* (37) also investigated the ability of the adjoint equations to reduce the drag in a two- and three-dimensional cylinder at Re = 100 and Re = 3900 using RANS models. Mao *et al.* investigated using DNS the physical mechanisms leading to two- and three-dimensional cylinder drag reduction at Re $\leq$ 1000 through a surface-normal wall transpiration (38), and a tangential motion of the surface (39). The sensitivity information computed through the adjoint equations was used to find the optimal configuration of the system.

The goal of this article is to assess the efficiency of Bayesian Optimization in high-fidelity simulations in canonical flows, such as the flow around a circular cylinder computed using high-order numerical solvers and Large Eddy Simulation, and compare its performance against alternative derivative-free optimization techniques. To this end, we consider the drag reduction problem through a series of tangential velocity actuators of unknown intensity along a cylinder wall in two and three dimensions.

This article is organized as follows: in Section 2, we give an overview of the optimization framework for CFD applications. In Section 3, we present the two-dimensional case at Re = 500 and discuss the influence of the number of actuators, the evolution of the drag coefficient and the resulting flow field. A parametric study of BO and a comparison against alternative optimization techniques are also presented. In Section 4, the results of the three-dimensional case at Re = 3900 are presented and discussed. Finally, we conclude in Section 5 discussing the main results and suggestions for future work.

2. Optimization framework

2.1. General outline

The starting point in this work is a numerical simulation of an unsteady flow that attempts to assess the effect of a given set of parameters $\mathbf{s}$, such as a configuration of flow actuators, shapes of obstacles, etc. In general, this requires solving numerically an advection–diffusion type of equation represented by

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{1}$$

on a domain $\Omega \times [0, T]$ subject to appropriate initial conditions $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ and boundary conditions on $\partial\Omega$. In the above, $\mathbf{u}$ is the vector of flow variables, $\mathbf{F}$ is the tensor of advective and diffusive fluxes, T is the final time of the simulation, $\mathbf{x} = (x, y, z)^\mathsf{T}$ is the vector of spatial coordinates and t is the time. In the case of incompressible flow, this system is augmented by the incompressibility constraint. We denote the solution of this problem by $\mathbf{u}(\mathbf{x}, t; \mathbf{s})$, where the dependence of the choice of parameters is indicated explicitly.

To quantify the performance of a specific choice of parameters $\mathbf{s}$, we consider an objective function f that depends functionally on the evolution of the flow field $\mathbf{u}(\mathbf{x}, t; \mathbf{s})$ and $\mathbf{s}$. Example of such functions are the time-averaged lift and

drag coefficients. As per our optimization problem, the goal is to find the optimum design $\mathbf{s}^*$ in a given design space $\mathcal{S}$ such that

$$\mathbf{s}^* = \arg\min_{\mathbf{s}\in\mathcal{S}} f(\mathbf{s}). \tag{2}$$

The general outline of the algorithm to solve this problem is shown in Algorithm 1. First, a Design of Experiments is performed to determine the points in the design space that will be used to initialize a surrogate model. As a surrogate model, we choose the Gaussian Process (GP), typically used in BO. Second, simulations are run at these design points and the values of the objective function are computed. Then, we repeatedly maximize an auxiliary function, known as acquisition function, to determine the next point to evaluate, perform the simulation at this point, evaluate the objective function, and update the Gaussian Process. Once the stop criterion is satisfied, the last candidate point is returned as the solution. In the following, we present the algorithmic details of each step.

Algorithm 1: General outline of Bayesian Optimization.

```
Initialization:
Design of Experiments (DOE);
Computational Fluid Dynamics;
Evaluation of the objective function;
while stop criterion not satisfied: do
    Update of the Gaussian Process;
    Optimization of the acquisition function;
    Computational Fluid Dynamics;
    Evaluation of the objective function;
end
```

2.2. Design of Experiments (DOE)

To initialize the optimization loop, n initial observations $\mathbf{q}_{1:n} = (q_1, q_2, \ldots, q_n)^\mathsf{T}$ are performed at n different design parameters $\mathbf{s}_{1:n} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n)^\mathsf{T}$. The purpose of this step, known as Design of Experiments (DOE), is to gather sufficient data $\mathcal{D}_{1:n} = \{\mathbf{s}_{1:n}, \mathbf{q}_{1:n}\}$ to build an initial surrogate model that will approximate the objective function.

The quality of such model greatly depends on how these initial points $\mathbf{s}_{1:n}$ are chosen, and a number of DOE techniques that attempt to distribute the initial points optimally have been proposed. The so-called full factorial sampling technique is perhaps the most intuitive. It consists in dividing the interval for each design variable in discrete values or *levels*. Then, a full factorial DOE is built by considering all possible combinations of these levels across all design variables. The main shortcoming of this technique is that it is difficult to achieve a good DOE for a given available budget, i.e. the number of initial observations n that is considered. Furthermore, when the initial sampling space is projected onto each axis, these points overlap.

Other space-filling techniques such as Latin Hypercube Sampling (LHS) or Sobol sequences have been developed. The first one consists in the subdivision of the design space into an orthogonal grid. To this end, each direction of the design space is divided into n elements of the same length. Then, we choose n sub-volumes such that we only have one element along each column or row of the grid and avoid correlations between the dimensions. On the other hand, Sobol sequences are quasi-random low-discrepancy sequences. The main idea is to subdivide the space into elements, set a sample in each element and then consider a finer grid. These two design techniques try to fill the space such that

points, for a given budget, are optimally distributed. For a given budget and modeling purposes, these techniques are often superior to alternatives (see (1)).

Regarding the number of initial points that is required, Jones *et al.* (25) suggested to use an initial sample size of $n = 10N$ where N is the dimension of the design space. However, in the case of unsteady flow simulations, this sample size can easily exceed available computational resources. Forrester *et al.* (40) considered different initial sample size estimates depending on the end purpose of the DOE. If the surrogate model is only meant to provide an accurate representation of the objective function, the total budget should be invested in the DOE. If the model is used for local optimization, then most of the budget should be spent on the DOE and a few points will be collected during the optimization process. Finally, for global optimization purposes, most of the points should come from the optimization procedure. For the Expected Improvement acquisition function (see Section 2.5), Sóboster *et al.* (41) showed that approximately one third of the points should be in the DOE and the remaining two thirds collected during the optimization procedure.

For further information, we refer the reader to Part 2 of Cavazzuti (1) and to Part 1 and Section 3.3.2 of Forrester *et al.* (40).

2.3. Computational Fluid Dynamics (CFD) and evaluation of the objective function

The numerical simulations presented in Section 3 and Section 4 are carried out using the open-source numerical flow solver PyFR (42). This code is written in Python and solves advection–diffusion equations such as Eq. (1) on streaming architectures using the flux reconstruction methodology from (43).

We can write the compressible Navier–Stokes equations in the form of Eq. (1), expressing the conservation laws of mass, momentum and energy. For that, we will take the state vector as $\mathbf{u} = (\rho, \rho v_x, \rho v_y, \rho v_z, \epsilon)^\mathsf{T}$, ρ being the density, $\mathbf{v} = (v_x, v_y, v_z)$ the velocity field in Cartesian coordinates, and ϵ the total energy density per unit volume. The flux tensor $\mathbf{F}$ will be given by

$$\mathbf{F} = \begin{pmatrix} \rho v_x & \rho v_y & \rho v_z \\ \rho v_x^2 + p - \tau_{xx} & \rho v_y v_x - \tau_{yx} & \rho v_z v_x - \tau_{zx} \\ \rho v_x v_y - \tau_{xy} & \rho v_y^2 + p - \tau_{yy} & \rho v_z v_y - \tau_{zy} \\ \rho v_x v_z - \tau_{xz} & \rho v_y v_z - \tau_{yz} & \rho v_z^2 + p - \tau_{zz} \\ v_x(\epsilon + p) - v_i \tau_{ix} - \Delta \partial_x T & v_y(\epsilon + p) - v_i \tau_{iy} - \Delta \partial_y T & v_z(\epsilon + p) - v_i \tau_{iz} - \Delta \partial_z T \end{pmatrix}, \tag{3}$$

where we have used Einstein notation and p represents the pressure, T the temperature, $\boldsymbol{\tau}$ is the stress tensor, and $\Delta = \mu c_\mathrm{p}/\mathrm{Pr}$ with μ being the dynamic viscosity, c_p the specific heat capacity at constant pressure and Pr the Prandtl number. For a Newtonian fluid, we have $\tau_{ij} = \mu\left(\partial_i v_j + \partial_j v_i\right) - \frac{2}{3}\mu\delta_{ij}\nabla \cdot \mathbf{v}$ where δ_{ij} is the Kronecker delta. Finally, we also have to consider the equation of state for the perfect gas

$$p = \rho \frac{\gamma - 1}{\gamma} c_\mathrm{p} T, \tag{4}$$

where γ is the adiabatic index, and the energy equation

$$\epsilon = \frac{1}{2}\rho \|\mathbf{v}\|^2 + \frac{p}{\gamma - 1}. \tag{5}$$

Once the simulations are performed, we can compute our objective function as in Eq. (2).

2.4. Gaussian Process

With the data $\mathcal{D}_{1:n}$, a surrogate model $\hat{q}$ is built to approximate the objective function. Following Bayes' theorem, we write

$$P(\hat{q}|\mathcal{D}_{1:n}) \propto P(\mathcal{D}_{1:n}|\hat{q})P(\hat{q}), \tag{6}$$

where $P(\hat{q}|\mathcal{D}_{1:n})$ is the posterior distribution of the model, $P(\mathcal{D}_{1:n}|\hat{q})$ the likelihood, and $P(\hat{q})$ the prior distribution. The posterior distribution represents the updated beliefs of the objective function, the likelihood is the agreement between the data and the model, and the prior corresponds to the initial beliefs on the objective function. After each observation of the objective function, the posterior distribution is updated. An acquisition function will then be used together with the posterior distribution to drive the optimization process, as it introduces a criterion to select the next point to evaluate. Typically these criteria allow for a trade-off between exploration, i.e. a design point $\mathbf{s}_{n+1}$ is chosen such that there is a high uncertainty of our model on the objective function, and exploitation, i.e. a design point $\mathbf{s}_{n+1}$ is chosen such that there is a high probability of reward, i.e. low value of the objective function.

2.4.1. Prior distribution

One widely used approach in Bayesian Optimization is to consider Gaussian Process (GP) as a surrogate model of the objective function. An extended description of the GP can be found in Rasmussen and Williams (44), and a tutorial can be found in Schulz *et al.* (45). At every design point, the distribution of the function will be described by a Gaussian distribution. The GP is the joint distribution of all the multivariate normal distributions, and the model of the function is then given by

$$f(\mathbf{s}) \sim \mathrm{GP}(\mu_0(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')). \tag{7}$$

In the above, the mean $\mu_0(\mathbf{s}) = \mathrm{E}[f(\mathbf{s})]$ represents the expected value of the objective function at $\mathbf{s}$ whereas the covariance function or kernel $k(\mathbf{s}, \mathbf{s}') = \mathrm{E}[(f(\mathbf{s}) - \mu(\mathbf{s}))(f(\mathbf{s}') - \mu(\mathbf{s}'))]$ defines the smoothness of the model by introducing a dependence between the designs $\mathbf{s}$ and $\mathbf{s}'$.

Initially, it is generally assumed that the prior mean $\mu_0(\mathbf{s}) = 0$. Regarding the kernel, popular choices are the Radial Basis Function (RBF), the Matérn52 and Matérn32 kernels

$$k_{\mathrm{RBF}}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2}\right), \tag{8}$$

$$k_{5/2}(r) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5r^2}{3}\right) \exp\left(-\sqrt{5}r\right), \tag{9}$$

and

$$k_{3/2}(r) = \sigma_f^2 \left(1 + \sqrt{3}r\right) \exp\left(-\sqrt{3}r\right), \tag{10}$$

respectively, where σ_f^2 is the variance, $r = (\mathbf{s} - \mathbf{s}')^{\mathsf{T}} \mathbf{\Lambda} (\mathbf{s} - \mathbf{s}')$ with $\mathbf{\Lambda}$ a square diagonal matrix whose entries are $1/\lambda_i^2$, λ_i being a characteristic length scale along the i-th direction. The RBF kernel is generally used for smooth processes and the Matérn32 kernel is used for rougher processes, whereas the Matérn52 represents an intermediate between these two cases.

Then, we will perform a number of observations of our objective function. Since these can be noisy, we formally write

$$q_i = f(\mathbf{s}_i) + \eta_i, \tag{11}$$

where q_i is the observation of the objective function and η_i the noise in the observation at the input point $\mathbf{s}_i$. Generally, the noise is considered to be normally distributed, i.e. $\eta = \mathcal{N}(0, \sigma_\eta^2)$, where σ_η^2 is the noise variance. After n observations of the objective function, the joint prior distribution at $\mathbf{s}_{n+1}$ is

$$\begin{bmatrix} \mathbf{q}_{1:n} \\ f_{n+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \mathbf{K} + \sigma_\eta^2 I & \mathbf{k} \\ \mathbf{k}^{\mathsf{T}} & k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) \end{pmatrix}\right), \tag{12}$$

where, $\mathbf{K} = [k_{ij}]$, $\mathbf{k} = [k_{i,n+1}]$ with $k_{ij} = k(\mathbf{s}_i, \mathbf{s}_j)$, and $1 \leq i, j \leq n$.

2.4.2. Posterior distribution

To update the model with new observations of the objective function, it is necessary to restrict or to *condition* the previous joint prior distribution to functions that agree with the gathered data $\mathcal{D}_{1:n}$. A posterior predictive distribution can then be obtained:

$$P(q_{n+1}|\mathcal{D}_{1:n}, \mathbf{s}_{n+1}) = \mathcal{N}(\mu_n(\mathbf{s}_{n+1}), \sigma_n^2(\mathbf{s}_{n+1})), \tag{13}$$

where

$$\mu_n(\mathbf{s}_{n+1}) = \mathbf{k}^\mathsf{T}[\mathbf{K} + \sigma_\eta I]^{-1}\mathbf{q}_{1:n}, \tag{14}$$

and

$$\sigma_n^2(\mathbf{s}_{n+1}) = k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) - \mathbf{k}^\mathsf{T}[\mathbf{K} + \sigma_\eta^2 I]^{-1}\mathbf{k}, \tag{15}$$

where μ_n and σ_n^2 are, respectively, the updated mean and variance functions after n observations.

2.4.3. Hyperparameters estimation

In the previous sections, we have introduced hyperparameters such as σ_f, λ_i and σ_η, which we denote here by the vector $\boldsymbol{\psi} = (\sigma_f, \lambda_i, \sigma_\eta)^\mathsf{T}$. The value of $\boldsymbol{\psi}$ is unknown and it is estimated from the data by maximizing the logarithm of the marginal likelihood of the model

$$\log P(\mathbf{q}_{1:n}|\mathbf{s}_{1:n}, \boldsymbol{\psi}) = -\frac{1}{2}\mathbf{q}_{1:n}^\mathsf{T}(\mathbf{K} + \sigma_\eta^2 I)^{-1}\mathbf{q}_{1:n} - \frac{1}{2}\log|\mathbf{K} + \sigma_\eta^2 I| - \frac{n}{2}\log(2\pi). \tag{16}$$

Eq. (16) is generally maximized using gradient-based optimization methods.

2.5. Acquisition functions

Once the posterior mean and variance functions are obtained, an acquisition function that will exploit the information on the mean and standard deviation will be used to determine the next candidate point to evaluate. The location of the maximum of this function will then correspond to the next point to evaluate. Since the acquisition function depends on the mean and standard deviation, the computational cost of maximizing this function is much cheaper than minimizing the objective function, which requires expensive simulations.

Probably, the simplest acquisition function is the negative lower confidence bound (NLCB) derived from the work of Cox (46) given by

$$\text{NLCB}(\mathbf{s}) = -\mu_n(\mathbf{s}) + \kappa\sigma_n(\mathbf{s}), \tag{17}$$

where κ is an exploration/exploitation trade-off parameter.

Kushner (21) used the Probability of Improvement (PI), also known as Maximum Probability of Improvement (MPI), as an acquisition function which is defined as

$$\text{PI}(\mathbf{s}) = \Phi\left(\frac{f(\mathbf{s}_n^*) - \mu_n(\mathbf{s}) - \kappa}{\sigma_n(\mathbf{s})}\right) \tag{18}$$

where Φ is the normal cumulative distribution function, and $\mathbf{s}_n^* = \arg\min_{\mathbf{s}\in\mathbf{s}_{1:n}} f(\mathbf{s})$. The Probability of Improvement represents the probability of obtaining a better value of the objective function than the best value that has been found so far, i.e. $P(f(\mathbf{s}) \leq f(\mathbf{s}^*))$.

It is also possible that instead of just quantifying the probability of improvement to calculate the amount of expected improvement. Moćkus *et al.* (22) defined the Expected Improvement (EI) as

$$\mathrm{EI}(\mathbf{s}) = \begin{cases} (f(\mathbf{s}_n^*) - \mu_n(\mathbf{s}) - \kappa)\Phi\left(\dfrac{f(\mathbf{s}_n^*) - \mu_n(\mathbf{s}) - \kappa}{\sigma_n(\mathbf{s})}\right) + \sigma_n(\mathbf{s})\phi\left(\dfrac{f(\mathbf{s}_n^*) - \mu_n(\mathbf{s}) - \kappa}{\sigma_n(\mathbf{s})}\right) & \text{if} \quad \sigma_n > 0 \\ 0 & \text{if} \quad \sigma_n = 0 \end{cases} \tag{19}$$

where ϕ is the Gaussian probability density function.

The use of PI is known to result in an aggressive exploitation (47). On the other hand, EI and NLCB are more explorative. EI remains the most used acquisition function. In Section 3.4, we will illustrate the behaviour of these acquisition functions through an application to a numerical flow simulation.

2.6. Parallel evaluations

Even if BO is normally a sequential process, it is also possible to run BO in parallel. In this article, the parallelization will be done through a local penalisation method developed by (48). When running in parallel, the maximum of the acquisition function will be evaluated, and a penalty term is introduced near this location. Then, the penalized acquisition function is maximized in order to find a second point. This process is repeated until the desired number of candidate points are determined. More precisely, we have

$$\mathbf{s}_{i,k} = \arg\max_{\mathbf{s}\in\mathcal{S}} \left\{ g(a(\mathbf{s}, \mathcal{D}_{1:i-1})) \prod_{j=0}^{k-1} \varphi(\mathbf{s}, \mathbf{s}_{i,j}) \right\}, \tag{20}$$

where $\mathbf{s}_{i,k}$ is the k-th point of the batch at the i-th iteration, $a(\mathbf{s}, \mathcal{D}_{1:i-1})$ is the acquisition function that depends on $\mathbf{s}$ and the data previously evaluated $\mathcal{D}_{1:i-1}$, and $g\colon \mathbb{R} \mapsto \mathbb{R}^+$ is a differentiable function that maps the acquisition function $a(\mathbf{s}, \mathcal{D}_{1:i-1})$ into the positive real numbers space without changing the locations of its extrema. Here, $g(z) = z$ in the zones where the acquisition function is positive and $g(z) = \log(1 + \exp z)$ elsewhere. $\varphi(\mathbf{s}, \mathbf{s}_{i,j})$ is a penalty term that depends on $\mathbf{s}$ and the previously suggested points of the batch at the i-th iteration, i.e. $\mathbf{s}_{i,j}$, and is defined by

$$\varphi(\mathbf{s}, \mathbf{s}_{i,j}) = \frac{1}{2}\mathrm{erfc}\left(-\frac{1}{\sqrt{2\sigma_i^2(\mathbf{s}_{i,j})}} \left(L\|\mathbf{s}_{i,j} - \mathbf{s}\| + M - \mu_i(\mathbf{s}_{i,j}) \right) \right), \tag{21}$$

where μ_i and σ_i are, respectively, the mean and the standard deviation at the i-th iteration, $\mathrm{erfc}(\cdot)$ is the complementary error function, $M = \min_{\mathbf{s}\in\mathcal{S}} f(\mathbf{x})$ and L is a valid Lipschitz constant.

Since the values of M and L are a priori unknown, the approximation $\hat{M} = \min \mu_i(\mathbf{s})$ and the so-called Gaussian Process Lipschitz Constant Approximation $\hat{L}_{GP-LCA} = \max \|\mu_\nabla(\mathbf{s})\|$ (with $\mu_\nabla(\mathbf{s})$ the gradient of the mean μ_i) are introduced. The resulting suggested points constitute a batch that can be evaluated in parallel.

2.7. Example

To conclude this section, a one-dimensional optimization problem that is solved using Bayesian Optimization is presented. We consider the flow in the incompressible limit around a circular cylinder at Reynolds number Re = 40 (based on the diameter D). For now, we omit the numerical details of the simulation as they will be discussed below in Section 3. The resulting flow is steady and the drag coefficient is $C_\mathrm{d} = 2f_x/(\rho_\infty U_\infty^2 D) \approx 1.58$, where U_∞ and ρ_∞ are, respectively, the free-stream speed and density, and f_x is the streamwise component of the force exerted on the cylinder.

To reduce the drag coefficient, a tangential velocity profile is introduced at the cylinder surface as illustrated in Fig. 1. In this example, the tangential velocity profile is given by

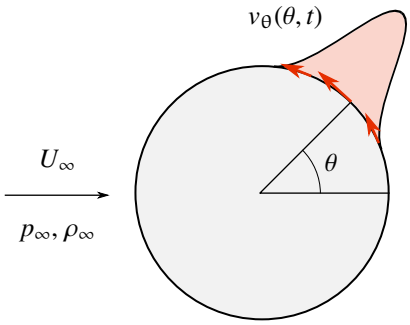
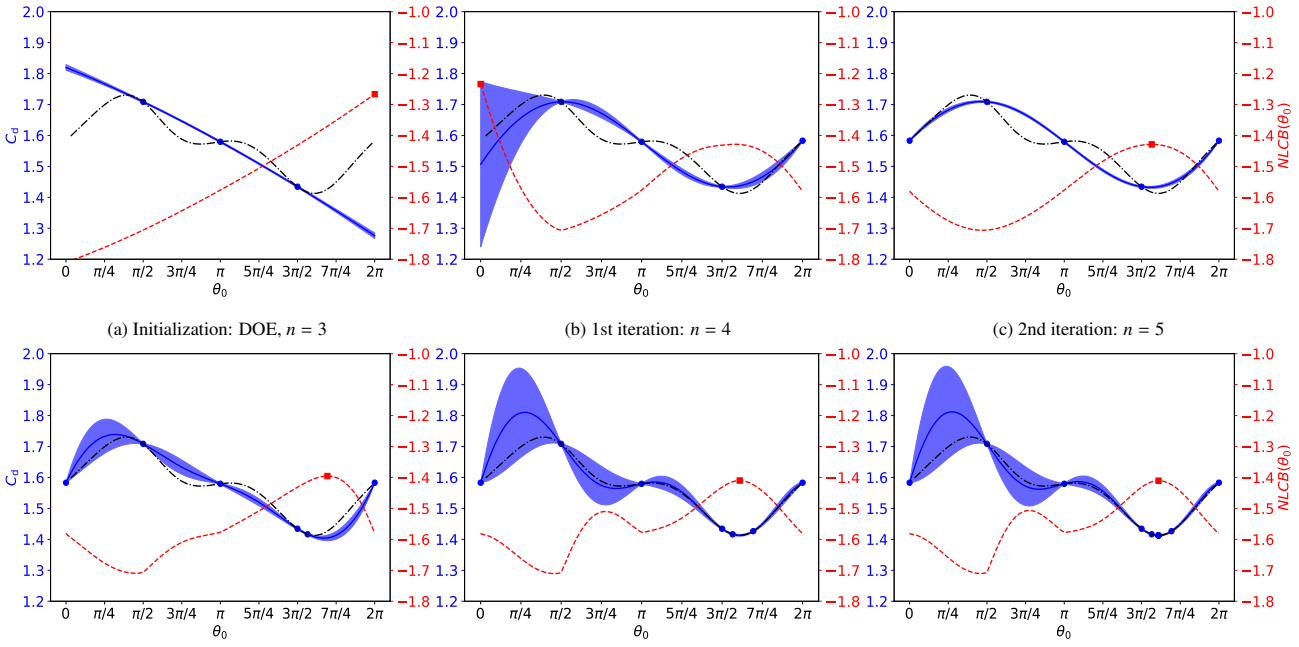


Figure 1: Uniform flow around a cylinder with tangential flow actuation at the surface.



(a) Initialization: DOE, $n = 3$

(b) 1st iteration: $n = 4$

(c) 2nd iteration: $n = 5$

(d) 3rd iteration: $n = 6$

(e) 4th iteration: $n = 7$

(f) 10th iteration: $n = 13$

Figure 2: Example of BO on a canonical case. The blue line represents the mean of the Gaussian Process, the blue shaded area corresponds to the 95% confidence interval, and the blue circles are the design points already evaluated. The red dashed line is the acquisition function and the square marker indicates the next candidate point. The dash-dotted line shown in black is the true objective function obtained from a 100-point Sobol DOE.

$$v_\theta(\theta; \theta_0, \sigma) = \sum_{k=-\infty}^{+\infty} \exp\left(\frac{-(\theta - \theta_0 + 2\pi k)^2}{2\sigma^2}\right), \tag{22}$$

where θ is the angle measured from the aft of the cylinder, θ_0 is the angle of maximum amplitude and σ is the standard deviation. This profile corresponds to a counterclockwise tangential component localized around θ_0.

In Eq. 22, we set $\sigma = 0.5$. The goal is then to find the value of $\theta_0 \in [0, 2\pi]$ that minimizes the drag coefficient. A Gaussian Process is initialized using a 3-point Sobol DOE and the RBF kernel. In the following, Bayesian Optimization is performed in sequential mode, and the NLCB acquisition function given in Eq. (17) with $\kappa = 2$ is chosen. Throughout this article, optimization problems are solved using the open-source software suite GPyOpt (49).

The first 10 iterations of the algorithm are illustrated in Fig. 2. At the beginning of the algorithm, the GP is built using the initial data, i.e. $\theta_0 \approx \pi/2$, $\theta_0 \approx \pi$ and $\theta_0 \approx 3\pi/2$ as well as the corresponding values of the objective function (Fig. 2a). To determine the next candidate point, the acquisition function is maximized; the point $\theta_0 = 2\pi$ is chosen due to the low value of the estimated mean. At this point, the acquisition function reaches its maximum, which indicates that this point corresponds to the best trade-off between exploration and exploitation. Once this point is evaluated, the Gaussian Process is updated. At the first iteration, a low mean and a high uncertainty is observed at $\theta_0 = 0$ (Fig. 2b). A function evaluation

will then be performed at this point and the Gaussian process is updated again. At the second iteration, the algorithm will find a promising design at $\theta_0 \approx 4.91$ (Fig. 2c). Once the Gaussian Process is updated, the best trade-off between exploitation and exploration is found at $\theta_0 \approx 5.32$ at the third iteration (Fig. 2d). At the fourth iteration, the algorithm will evaluate f at $\theta_0 \approx 5.06$ due to the low mean (Fig. 2e). Finally, the algorithm keeps evaluating points that are increasingly closer to $\theta_0 \approx 5.06$ (Fig. 2f).

3. Drag reduction in the two-dimensional unsteady flow around a cylinder

3.1. Problem description

Once the details of the optimization scheme have been presented, we now turn the attention to the drag reduction in the unsteady flow around a cylinder.

The goal is to reduce the root-mean-square (RMS) value of the drag coefficient over a given time interval $[T, T + \Delta T]$ through a tangential velocity actuation at the cylinder wall $v_\theta(\theta, t)$, where $\theta \in [0, 2\pi]$ is again the positive angle measured from the aft of the cylinder, as depicted in Fig. 1. The velocity profile is then determined by linear interpolation from the value of the tangential velocity $v_{\theta,j}$ at a number $\hat{N}$ of equally-spaced control points $\theta_j = j(2\pi/\hat{N}) = j\Delta\theta$ where $j = 0, 1, \ldots, \hat{N} - 1$. In particular, we have

$$v_\theta(\theta, t) = \sum_{j=0}^{\hat{N}-1} v_{\theta,j} \Lambda\left(\frac{\theta - \theta_j}{\Delta\theta}\right) \tanh t, \qquad \text{with } \Lambda(x) = \max\left(1 - |x|, 0\right). \tag{23}$$

In addition, we set $v_{\theta,j} = -v_{\theta,\hat{N}-j}$ from symmetry considerations. Then, $v_{\theta,0} = 0$, and $v_{\theta,\hat{N}/2} = 0$ if $\hat{N}$ is even. The vector of design variables is $\mathbf{s} = [v_{\theta,i}/U_\infty]$ with $i = 1, \ldots, N$ and the number of independent design variables is $N = \left\lfloor (\hat{N} - 2)/2 \right\rfloor$.

An objective function is then defined by considering the sum of the square of the drag coefficient over the time window $T \le t \le T + \Delta T$ (the initial transients for $0 \le t \le T$ are discarded), and a penalization term that mimics the energetic cost of such actuation. More precisely, we have

$$f(\mathbf{s}) = \sqrt{\frac{1}{\Delta T} \int_T^{T+\Delta T} \left(C_\mathrm{d}^2(t; \mathbf{s}) + \frac{\alpha}{N} \mathbf{s}^\mathsf{T} \mathbf{s} \right) \mathrm{d}t}, \tag{24}$$

where the instantaneous drag coefficient is $C_\mathrm{d}(t; \mathbf{s}) = 2/(\rho_\infty U_\infty^2 D) \int_{\partial\Omega_c} (-P n_x + \tau_{xj} n_j) \, \mathrm{d}\mathbf{x}$, where n_j is the component of the wall normal unit vector along the j-th direction, and α is a scaling constant for the penalization term. Once a suitable design space $\mathcal{S}$ is defined, we arrive at an optimization problem as given in Eq. (2), which is solved using the framework presented in Section 2.

In the remaining of this section, a two-dimensional case at Reynolds number Re = 500 and Mach number Ma = 0.2 is chosen. The Prandtl number Pr is set to 0.71 and γ is 1.4. Even though the flow is expected to display three-dimensional structures in this regime, a two-dimensional configuration is selected as the reduced computational cost allows for the assessment of the influence of various parameters and a comparison against other optimization techniques. A three-dimensional case is presented in Section 4.

3.2. Numerical set-up

The numerical simulations have been performed using PyFR. The extent of the computational domain is $[-9D, 25D] \times [-9D, 9D]$, and it is discretized using a C-H grid topology. This mesh is refined in the region close to the cylinder $[-4D, 15D] \times [-4D, 4D]$, to improve the spatial resolution of the boundary layers and the wake. The results of the grid independence study are given in Appendix A. The cylinder surface is discretized along the tangential direction using

32 points, and the height of the first cell is $\Delta y/D = 0.0678$. A third-order discretization is also performed by the solver on the elements using the Gauss-Legendre quadrature for quadrilateral elements and the Williams-Shunn quadrature for triangular elements.

At the cylinder surface $\partial\Omega_c$, the wall-normal velocity component is set to zero, the wall-tangential component is given by Eq. (23) and the temperature is set to the free-stream value. The boundary conditions at the far-field boundary $\partial\Omega_\infty$ are specified using Riemann invariants to avoid the reflection of spurious acoustic waves (50).

For each design point $\mathbf{s}$, the equations are integrated in time starting from a snapshot obtained from the long-time integration ($100D/U_\infty$ time units) of the uncontrolled simulation, i.e. $\mathbf{s} = 0$, for $15D/U_\infty$ time units. The cost function is evaluated using Eq. (24), where we have taken $T = 10D/U_\infty$ and $\Delta T = 5D/U_\infty$ to discard the initial transients, $0 \leq t \leq 10D/U_\infty$, and consider only contributions from $10D/U_\infty \leq t \leq 15D/U_\infty$. This choice of ΔT corresponds to approximately one shedding cycle, and it leads to a reasonable accuracy while keeping a reduced computational cost. For longer integration times, i.e. larger values of ΔT, the differences in the value of the cost functions are smaller than 4% and 2% for $\alpha = 0$ and $\alpha \neq 0$, respectively.

Regarding the actuation at the wall, three configurations with an increasing number of control points are considered, namely 8, 16 and 32 control points, which correspond to 3, 7 and 15 design variables, respectively. The design space is defined by the box constraints $-1 \leq v_{\theta,i}/U_\infty \leq 1$. Note that the case with 8 actuators is included in the configuration with 16 actuators, which in turn is included in the setting with 32 actuators.

For each of these configurations, four different values of the penalization term are considered, i.e. $\alpha = 0, 2, 4$, and 8, leading to a total of 12 optimization problems.

3.3. Optimal solutions and flow-field features

In this section, the optimal solutions and the corresponding features of the flow fields are reported. We first focus on the cases with 32 actuators, i.e. $N = 15$, and varying α.

The optimal tangential velocity profiles are depicted in Fig. 3a. For $\alpha \neq 0$, the actuation is concentrated at the rear of the cylinder near $\theta = 3\pi/8$ ($\approx 68°$) and $\theta = 13\pi/8$ ($\approx 292°$), which is located downstream the boundary layer separation point. As the penalty term is decreased, the amplitudes increase progressively before reaching the boundary of the design space, i.e. $v_{\theta,i}/U_\infty = \pm 1$ for some i. At $\alpha = 0$, the optimum is located at one of the corners of the domain. For reference, we display also the optimal tangential velocity profile for $\alpha = 8$ as determined by an alternative numerical flow solver that features the adjoint equations (51). In this particular case, the optimal solution that is found using adjoint methods and 242 control points is very similar to the one found with BO and 32 control points. These velocity profiles are also qualitatively similar to the ones obtained by Mao *et al.* (39) at Reynolds number Re = 100. To confirm that a significant drag reduction persists for long integration times, additional simulations corresponding to the optimal solutions have been run for $100D/U_\infty$ time units; the results are shown in Fig. 3b.

To assess the changes that the actuators introduce in the flow, we now present in Fig. 4 the instantaneous vorticity field at $tU_\infty/D = 80$ as α is increased, as well as the average mean streamwise velocity component and the spectrum of the vertical velocity component v_y recorded at $x/D = 3$ and $y/D = 0$. The contour levels of the instantaneous vorticity component $\omega_z D/U_\infty$ (Fig. 4c, 4e, 4g, 4i) reveal that the maximum of the optimal tangential velocity profile is located downstream the boundary-layer separation point, and it counteracts the vorticity component of the vortices that are shed into the von Kármán street. This results in delayed flow separation and a weakened vortex formation that now occurs further downstream. These effects become increasingly more prominent as α decreases, i.e. the higher overall amplitude of the tangential velocity profile increases (Fig. 3a). The average streamwise velocity component $\overline{v}_x/U_\infty$ from

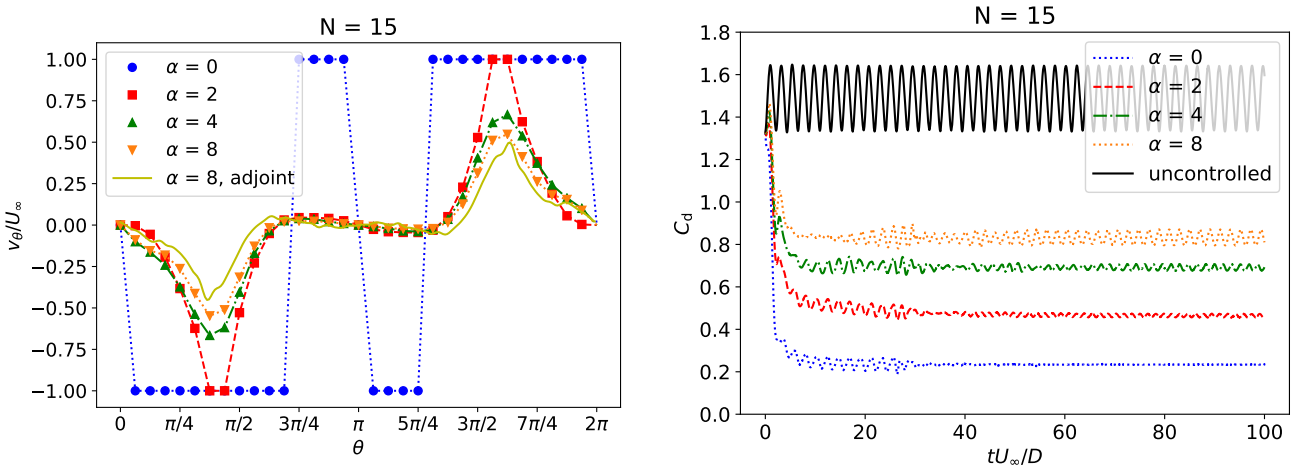


Figure 3: Optimal solution for 32 actuators ($N = 15$) and varying α. a) Optimal tangential velocity profiles as a function of θ. The markers indicate the location of the actuators. b) Temporal evolution of the drag coefficient for long integration times.

Case	$f(\mathbf{s})$	$\overline{C_d}$	C_l'	a/D	L_d/D	St
$\alpha = 0$	0.25	0.23	-	-	-	-
$\alpha = 2$	0.79	0.47	1.1×10^{-2}	0.5	1.13	0.311
$\alpha = 4$	0.93	0.69	8.8×10^{-2}	0.35	0.77	0.280
$\alpha = 8$	1.08	0.83	3.1×10^{-1}	0.13	0.71	0.245
Uncontrolled	1.50	1.50	8.8×10^{-1}	0.08	0.39	0.228

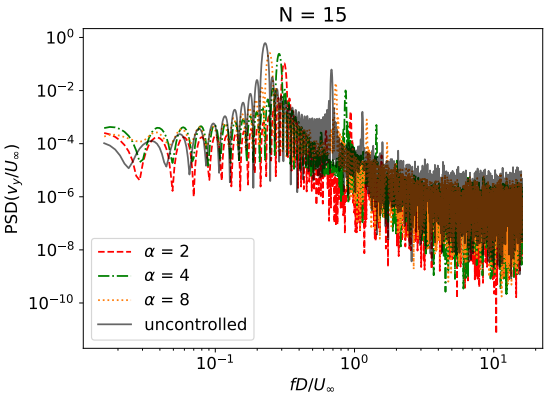
Table 1: Objective function $f(\mathbf{s})$, mean drag coefficient $\overline{C_d}$, RMS of the lift coefficient C_l', distance between the aft of the cylinder and the beginning of the recirculation zone a/D, length of the recirculation zone calculated between its two horizontal extremities L_d/D and Strouhal number St at the optimal designs $\mathbf{s}^*$ for varying α as well as the uncontrolled case $\mathbf{s} = 0$ ($N = 15$). a/D and L_d/D were calculated on the averaged solutions from $tU_\infty/D = 50$ to $tU_\infty/D = 100$ at $y = 0$

$tU_\infty/D = 50$ to $tU_\infty/D = 100$, depicted in Fig. 4 (right column), shows that as the strength of the actuation is increased, the wake narrows behind the cylinder and the length of the separation bubble is also increased, until finally disappearing for $\alpha = 0$. In this case, the wake is stabilized and the flow becomes steady. Finally, in Fig. 4a, the spectrum of v_y/U_∞ confirms that for decreasing $\alpha \neq 0$ the frequency of the vortex shedding increases.

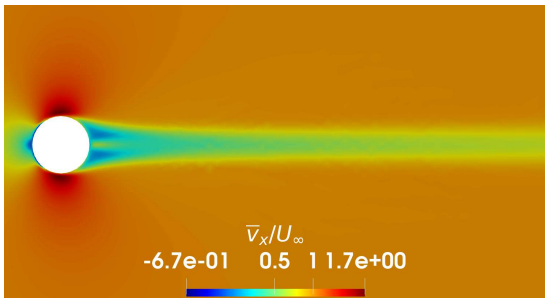
The force coefficients and the numerical values of the separation length and Strouhal number are summarized in Table 1. Starting from the uncontrolled case, the average drag coefficient and the RMS of the lift coefficient decrease as the penalization constant α is decreased; on the other hand, the length of the separation bubble and its distance from the cylinder increase whereas the vortex shedding frequency increases before the separation bubble is suppressed for $\alpha = 0$, leading to a stable steady flow field.

3.4. Influence of Bayesian Optimization parameters

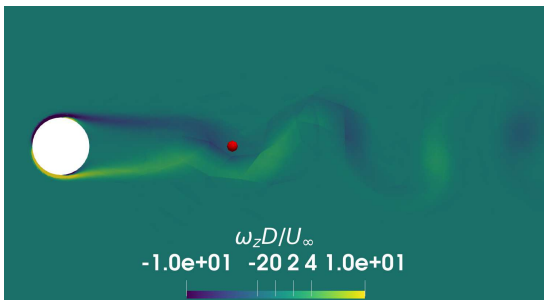
Prior to assessing the efficiency of BO, we investigate the influence of the number of design parameters, the initial number of points in the DOE, the choice of kernel, the acquisition functions and the optimizer for the several values of the penalty term. As mentioned before, 8, 16 and 32 actuators were considered, leading to 3, 7 and 15 design parameters, respectively, after symmetry considerations. The initial spaces were built using Sobol sequences and they consisted of 5, 10 and 15 observations. The isotropic Radial Basis Function (RBF), Matérn32 and Matérn52 kernels were used. We also considered the model without noise and fix $\sigma_\eta = 1 \times 10^{-6}$. Regarding the acquisition functions, Expected Improvement



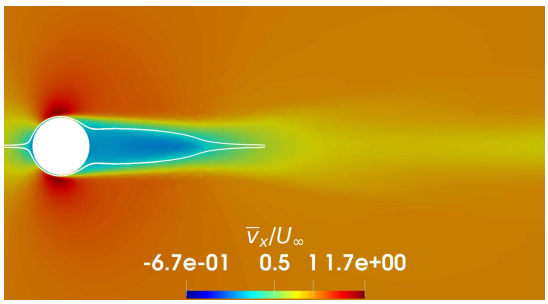
(a)



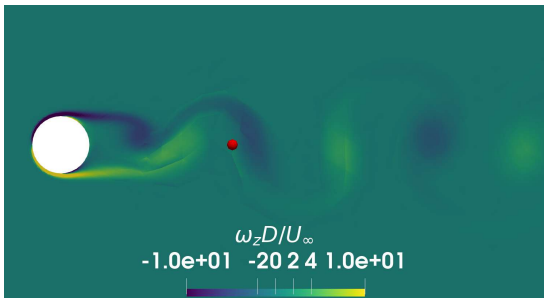
(b)



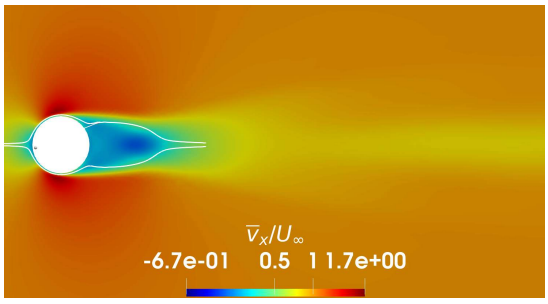
(c)



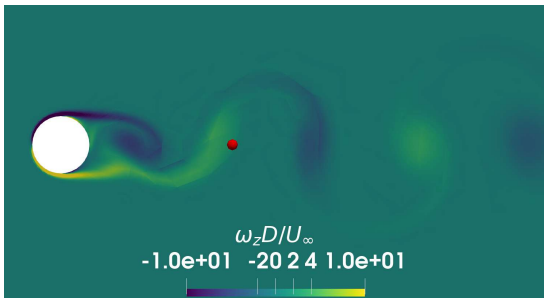
(d)



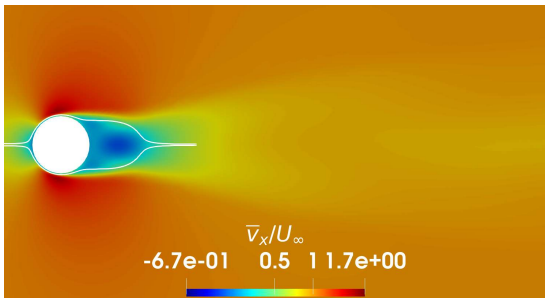
(e)



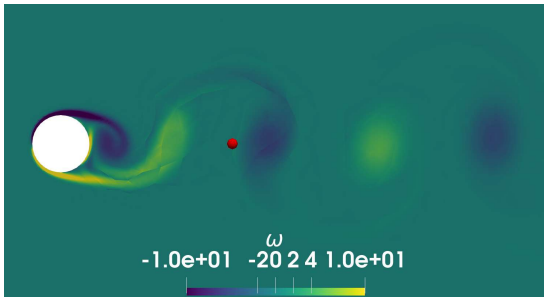
(f)



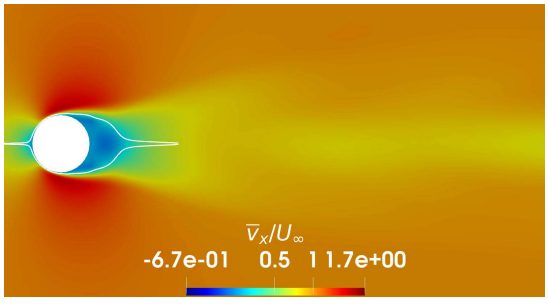
(g)



(h)

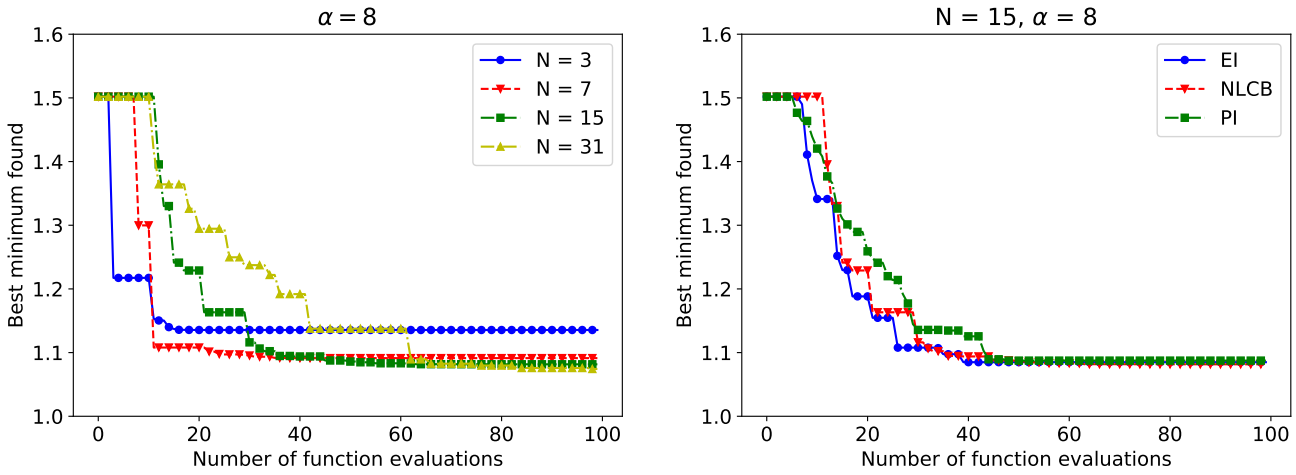


(i)



(j)

Figure 4: Flow-field characteristics of the optimum cases for $N = 15$ and the uncontrolled case. (a) Power spectral densities of the vertical velocity component v_y/U_∞ recorded at $x/D = 3$ and $y/D = 0$ (red dot in (c), (e), (g) and (i)). Left column: instantaneous vorticity component $\omega_z D/U_\infty$ at $tU_\infty/D = 80$ for (c) $\alpha = 2$, (e) $\alpha = 4$ (g) $\alpha = 8$ and (i) uncontrolled case. Right column: average streamwise velocity component $\overline{v}_x/U_\infty$ and selected streamlines (in white) for (b) $\alpha = 0$, (d) $\alpha = 2$, (f) $\alpha = 4$, (h) $\alpha = 8$ and (j) uncontrolled case.



(a) (b)

Figure 5: Best minimum found as a function of the number of function evaluations. Influence of (a) the number of design parameters with an initial DOE of 5 points, the NLCB acquisition function, RBF kernel, the L-BFGS-B optimizer and $\alpha = 8$ and (b) the acquisition function with an initial DOE of 5 points, 15 design parameters, the RBF kernel, the L-BFGS-B optimizer and $\alpha = 8$.

(EI), Negative Lower Confidence Bound (NLCB) and Probability of Improvement (PI) were compared. In the case of EI and PI, $\kappa = 0.01$ was chosen, whereas in the case of NLCB, $\kappa = 2$. To optimize the acquisition functions, the L-BFGS-B and CMA-ES optimization algorithms were compared. In the case of L-BFGS-B, the acquisition function is first sampled at 1000 randomly generated points in the design space and the 5 points with the highest value are retained. Then, the optimum is determined as the best optimum found from the maximization of the acquisition function using these 5 points as initial guesses. In the case of CMA-ES, the mean is initially specified at the center of domain, and the variance is set to 1/4 of the domain extent. For each setting, the cases $\alpha = 0, 2, 4, 8$ were considered. In total, 216 optimization problems were solved and we present below some observations.

In Fig. 5a, we depict a typical evolution of the best minimum found as a function of the number of function evaluations for different numbers of design parameters, where an additional case with 31 design variables is presented. Since the penalty term is normalized with the number of design variables N, we can compare the optimum value for different cases. As expected, a lower value at the optimum is obtained for increasing N. Beyond $N = 7$, the improvements are comparatively smaller. Even though Bayesian Optimization is typically restricted to problems with a moderate number of design variables, i.e. up to 15, in this case the algorithm showed good performance as the number of design variables was increased. For 3, 7, 15 and 31 design variables, we reach a reasonable optimum after 15, 23, 46 and 67 function evaluations, respectively. Further function evaluations lead to improvements that are within 1% of the previously computed optima. A possible explanation of the good performance of BO for an increasing number of design variables is that the objective function is low-dimensional due to the strong correlation between neighbouring control points.

We now turn the attention to the influence of the choice of acquisition function (see Fig. 5b). Here, EI, PI and NLCB provide a good approximation of the optimum after 40, 46 and 45 function evaluations, respectively. Again, further iterations are within 1% of the previously computed optima. EI is typically the one that performs best since it is the fastest to reduce the objective function. We can also notice the greedy behaviour of PI: a significant reduction of the objective function reduction is observed at selected function evaluations for EI and NLCB, whereas in the case of PI, the value of the objective function is decreased gradually. It is because during the 60 first function evaluations, this acquisition function has a maximum at points that are closer to the previously evaluated point. This acquisition function is known to be more aggressive than EI and NLCB, which are comparatively more exploratory. Even if EI was often the most efficient

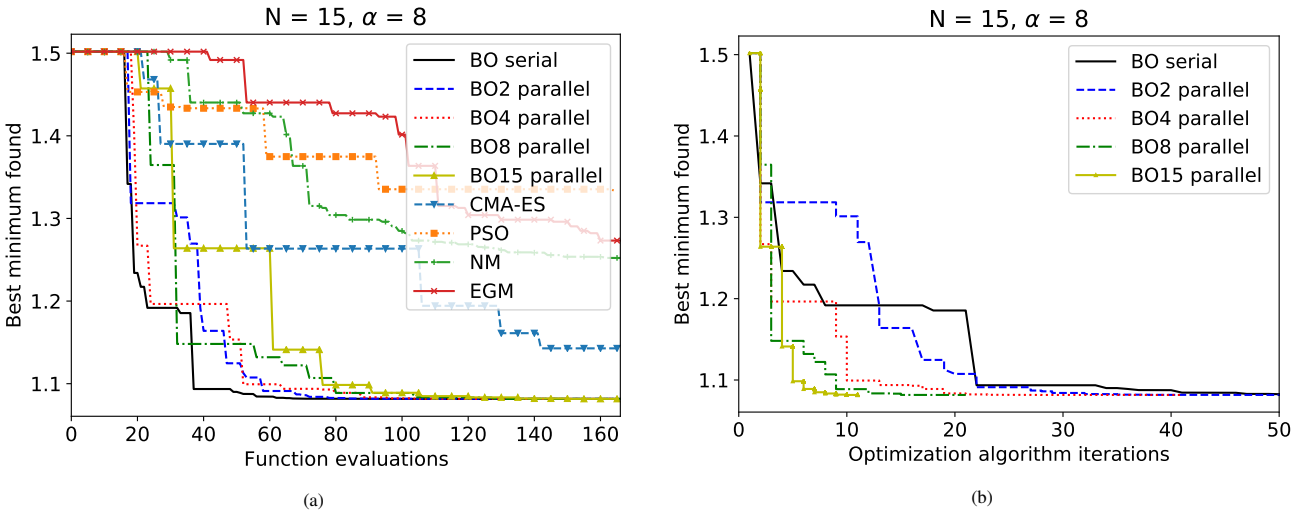


Figure 6: (a) Best minimum found as a function of the number of function evaluations for several optimization algorithms. (b) Comparison between the best minimum found using BO serial and BO parallel as a function of the number of iterations. BO# refer to parallel Bayesian Optimization with # parallel function evaluations per iteration. Case $N = 15$ and $\alpha = 8$.

one in reducing the value of the objective function, NLCB was the most accurate for the given budget and the one that performed the best for 31 design parameters.

Finally, regarding the number of samples in the DOE, and choices of kernel, optimizer and penalty term, we did not observe significant differences (not shown here). As a minor remark, the algorithm required fewer iterations to reach the optimum in the case $\alpha = 0$, which we recall is located at the corners of the domain (see Fig. 3a). This location corresponds to regions in the domain with high levels of uncertainty, which translates into large values of the acquisition function.

3.5. Comparison against other derivative-free techniques

To conclude this section, we present a comparison between Bayesian Optimization and several derivative-free optimization techniques, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (52), Particle Swarm Optimization (PSO) (10), Nelder–Mead (7; 53) and Explorative Gradient Method (EGM)(8). In the following, we briefly discuss details regarding the usage of these algorithms in the case under consideration.

Regarding CMA-ES, a population of $n > 1$ designs is sampled in the design space according to the multivariate normal distribution

$$\mathbf{s}_i^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(0, \mathbf{C}^{(g)}), \quad \text{for} \quad i = 1, \ldots, n, \tag{25}$$

where $\mathbf{m}^{(g)} \in \mathbb{R}^N$ is the mean, $\sigma^{(g)}$ the step size, and $\mathbf{C}^{(g)}$ a $N \times N$ positive definite matrix. The superscript (g) denotes that these quantities are taken at the g-th generation (or iteration). We also have $\mathbf{C}^{(0)} = \mathbf{I}_N$ where $\mathbf{I}_N$ is the $N \times N$ identity matrix. The individuals are then sorted, i.e. $f(\mathbf{s}_{1:n}^{(g+1)}) \leq f(\mathbf{s}_{2:n}^{(g+1)}) \leq \cdots \leq f(\mathbf{s}_{n:n}^{(g+1)})$, the best ζ designs are then selected and they are weighted such that $\sum_{i=1}^{\zeta} w_i = 1$ and $w_1 \geq w_2 \geq \cdots \geq w_\zeta$. The mean and covariance are then updated according to the selected individuals, the previous mean $m^{(g)}$ and the weights. Finally, a new population with the new mean and new covariance matrix is sampled. From the user perspective, CMA-ES requires two parameters: the initial mean $m^{(0)}$ and the initial standard deviation $\sigma^{(0)}$. We set $m^{(0)}$ at the center of the optimization domain and $\sigma^{(0)} = 0.5 U_\infty$, corresponding to $1/4$ of the domain length in each direction.

Regarding PSO, a population of particles is initialized in the design space. Each particle is defined by its location and

velocity, which is drawn from a uniform distribution at the first iteration, and subsequently updated as follows:

$$\mathbf{v}_{i+1} = w_i \mathbf{v}_i + c_1 r_1 (\mathbf{s}_p - \mathbf{s}_i) + c_2 r_2 (\mathbf{s}_g - \mathbf{s}_i), \tag{26}$$

$$\mathbf{s}_{i+1} = \mathbf{s}_i + \mathbf{v}_{i+1}, \tag{27}$$

where $\mathbf{v}_i$ is the velocity of the particles in the design space at the i-th iteration, w_i is an iteration-dependent inertial term, c_1 and c_2 are scalar parameters, r_1 and r_2 are random numbers uniformly distributed in $[0, 1]$, $\mathbf{s}_p$ is the personal best minimum position found by each particle, $\mathbf{s}_i$ the position of the particle at the iteration i, and $\mathbf{s}_g$ the best minimum position found during the optimization process. We set $\mathbf{v}_0 = 0$. The performance of PSO is highly dependent on the specific choice of the parameters w, c_1 and c_2. Here, the numerical values from (16) were used, i.e. c_1 and $c_2 = 2$ and w given by $w_i = (w_{\text{ini}} - w_{\text{end}})(i_{\max} - i)/i_{\max} + w_{\text{end}}$, where w_{ini} is the initial inertia term (here 0.9), w_{end} the final inertia term (here 0.4), $i_{\max}$ the maximum number of iterations of the algorithm and i the current iteration. Since we deal with a maximum number of function evaluations and not a maximum number of iterations, w_i decreases linearly from w_{ini} to w_{end} at the end of the budget.

The Nelder–Mead (NM) method is based on the construction of an initial simplex of dimension $N + 1$ where N is the dimension of the design space. We sort the vertices such as $f(\mathbf{s}_1) \leq f(\mathbf{s}_2) \leq ... \leq f(\mathbf{s}_{N+1})$. The centroid $\bar{\mathbf{s}}$ of all the points except $\mathbf{s}_{N+1}$ is then calculated:

$$\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i. \tag{28}$$

With this centroid, various operations can be performed on the simplex, namely reflection, expansion, contraction and a shrink step. For this algorithm, we keep the default parameters $\{\tau, \beta, \nu, \delta\} = \{1, 2, 1/2, 1/2\}$ where τ, β, ν and δ are respectively the parameters associated with the reflection, expansion, contraction and shrink step.

The Explorative Gradient Method (EGM) is based on the Nelder–Mead algorithm. Since the Nelder–Mead method may get trapped in a local minimum, Li *et al.* (8) added an extra-step. At the end of each iteration of the Nelder–Mead algorithm, a space-filling technique LHS is utilized to improve the exploration of the design space. Indeed, the point that maximizes the minimal distance with the points already evaluated is selected as the next point to evaluate

$$\mathbf{s}_{\text{LHS}} = \arg\max \min_{i=1,2,...n} \|\mathbf{s} - \mathbf{s}_i\|, \tag{29}$$

where $\mathbf{s}_{\text{LHS}}$ is the next point to evaluate, $\mathbf{s}$ is a point in the design space and $\mathbf{s}_i$ the i point evaluated.

A comparison between these algorithms and Bayesian Optimization is presented in Fig. 6.

In addition to the serial case, BO with parallel evaluations was performed according to the procedure sketched in Section 2.6. An initial space of 16 points is considered for all cases. This initial space design is chosen as the initial population for PSO and CMA-ES. The best $N + 1$ points are chosen to build the initial simplex of the NM and EGM algorithm. In the case of serial and parallel BO, the isotropic RBF kernel, the NLCB acquisition function and the L-BFGS-B optimizer were chosen.

The best minimum found for $N = 15$ and $\alpha = 8$ in terms of function evaluations is depicted in Fig. 6a. It is observed that serial BO followed by parallel BO are the most efficient algorithms in terms of function evaluations. Aside from BO, CMA-ES performed better than the remaining techniques. In this case, NM outperforms EGM, which indicates that the exploration step translates into a slower convergence rate. One explanation of this could be that the objective function do not have local optima but only one global optima. Thus, using a more explorative strategy is inefficient compared to using pure exploitation.

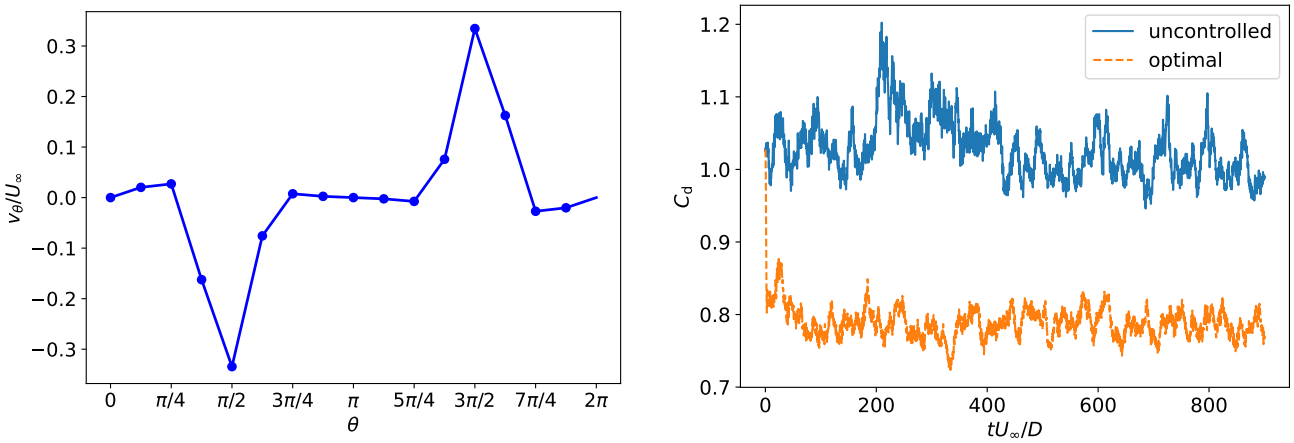


Figure 7: Results for the three-dimensional flow around a cylinder at Re = 3900, showing (a) optimal velocity profile around the cylinder, and (b) drag coefficient as a function of time for the uncontrolled and optimal cases.

In Fig. 6b results are presented for serial and parallel BO in terms of the iteration count. Even though serial BO requires fewer function evaluations, the time to solution can be greatly reduced if function evaluations can be performed in parallel.

4. Drag reduction in the three-dimensional flow around a cylinder

4.1. Problem description and numerical set-up

In this section, we apply the procedure previously outlined to the drag reduction in the three-dimensional flow around a cylinder at Re = 3900. In this case, the flow is solved using the implicit LES approach and 16 actuators, i.e. 7 design variables, were chosen to impose a streamwise invariant tangential velocity profile as described in Section 3.1; the penalty term constant is $\alpha = 8$. The optimization problem is again solved following the approach outlined in Section 2 using an initial sampling space of 5 points, an isotropic RBF kernel and the NLCB acquisition function (which is maximized using the L-BFGS-B technique). The Gaussian Process is considered noise-free ($\sigma_\eta = 1 \times 10^{-6}$).

The case without actuation is described in Vermeire *et al.* (54) and is provided as supplementary material. This case was also investigated by Lehmkuhl *et al.* (55), although with a different numerical solver. We use the numerical grid provided in Vermeire *et al.* (54). Details of the simulation and comparison with the literature are presented in Appendix B.

To accelerate the computations during the optimization process, the flow is solved using 3rd order elements. A first simulation is performed without actuation for $100D/U_\infty$ time units. Then, the subsequent flow evolution with tangential actuators is computed over $15D/U_\infty$ time units. The objective function is given in Eq. (24) with $T = 10D/U_\infty$ and $\Delta T = 5D/U_\infty$, and now the aerodynamic coefficients are referred to the frontal area $S = \pi D^2$.

During the BO process, each 3rd order simulation was run on 2 nodes on the supercomputer MinoTauro. Each node was equipped with 2 Intel Xeon E5-2630 v3 (Haswell) 8-core processors and 2 K80 NVIDIA GPU Cards. One simulation required approximately 96 CPU hours, i.e. 3 hours (wall time).

4.2. Optimization results

Once the optimal actuation is determined, the flow evolution is computed using 4-th order elements for $900D/U_\infty$ time units. On MinoTauro, this simulation needed around 20,000 CPU hours with GPUs, i.e approximately 417 hours

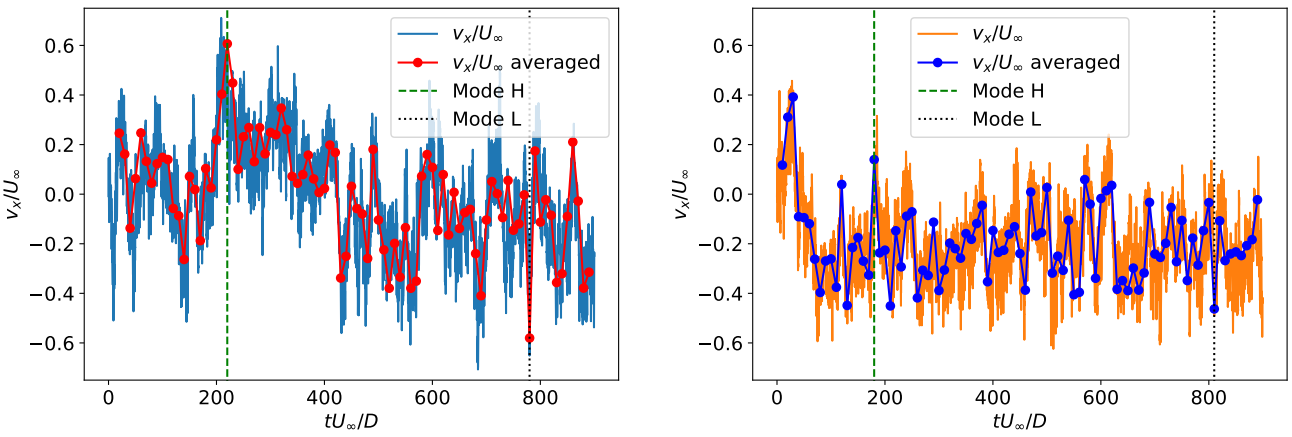


Figure 8: Temporal evolution of the streamwise velocity component averaged in the streamwise direction at (a) P_3 for the uncontrolled case and (b) for the optimal case. The dots represent the time average on a $10D/U_\infty$ time-unit window. The presence of modes L and H are represented by vertical dashed and dotted lines, respectively.

(wall time) on 3 nodes. In the remaining of this section, we compare the flow statistics in the cases with and without actuation. It should be noted that despite considering a spanwise invariant tangential velocity profile, no synchronization effects have been observed in the wake.

In Fig. 7a, we present the optimal velocity profile. As in the two-dimensional counterpart, the control points with the highest tangential velocity are located immediately downstream the boundary-layer separation point. A 23% reduction of the drag coefficient can be observed in Fig. 7b.

In Fig. 8, we present instantaneous streamwise averaged horizontal velocity component at $P_3 = (2.0, 0.0)$ for the uncontrolled case and the optimal solution. The streamwise velocity profiles were averaged using $10D/U_\infty$ time-unit windows (red line). Lehmkuhl *et al.* (55) pointed out that the uncontrolled case oscillates between a mode of high energy (mode H) and a mode of low energy (mode L). The modes H and L correspond, respectively, to a shorter and larger separation bubble. Thus, the mode L is associated to negative streamwise velocities whereas the mode H corresponds to positive streamwise velocities. In the uncontrolled case (Fig. 8a), it can readily be observed that the minimum of these averaged profiles (mode L) occurs at $tU_\infty/D = 220$ and its maximum (mode H) occurs at $tU_\infty/D = 780$. These results are in agreement with the frequency of oscillation between modes H and L $f_m D/U_\infty = 0.0064$ reported by (55). In the optimum case (Fig. 8b), a minimum occurs at $tU_\infty/D = 180$ and the maximum occurs at $tU_\infty/D = 810$. The results for the uncontrolled case have been compared against previous results; see Fig. B.12 and Fig. B.13. Since the simulation time is comparable to the oscillation time between modes L and H, some differences are observed in the long-time averaged velocity profiles shown in Vermeire *el al.* (54). As discussed in Parnaudeau *et al.* (56), the average length of the recirculation zone does not converge before $1200D/U_\infty$ time units. Nonetheless, the results are in good agreement with the experimental results of Parnaudeau *et al.* (56) and lay between the profiles from modes H and L computed by Witherden *et al.* (57). Finally, some differences are observed between the computed modes and the ones reported in Witherden *et al.* (57). The features of these modes are dependent on the window width and the start time of each window. Here, a shorter time window has been chosen, i.e. $10D/U_\infty$ time units, whereas in Witherden *et al.*, a window of $100D/U_\infty$ time units was used.

Following Lehmkuhl *et al.* (55), the temporal evolution of the velocity components at probes located at $P_1 = (0.71, 0.66)$, $P_2 = (1.3, 0.69)$, and $P_3 = (2.0, 0.0)$ is registered. The power spectra of the streamwise and cross-flow components for all the probes are computed using a Lomb periodogram technique (58). The power spectra of the cross-

flow component at P_1 is depicted in Fig. 9a.

For the uncontrolled case, three distinct frequencies can be readily noticed and correspond to the vortex shedding frequency at $f_{vs}D/U_\infty = 0.208$, the second harmonic of the vortex shedding $f_{sp}D/U_\infty = 0.419$ and the Kelvin–Helmholtz instability appearing in the separated shear layers at $f_{kh}D/U_\infty = 1.57$. For the optimal case, the amplitudes of the power spectra are reduced until $x/D \approx 2.0$ (not shown here), i.e. probe P_3, where we can observe that the spectra are very similar to the uncontrolled case, meaning that the actuators do not have much influence on the shedding downstream this point. It should be noted, however, that the vortex shedding frequencies are shifted towards slightly faster oscillations. Indeed, for the optimal case, we obtain $f^*_{vs}D/U_\infty = 0.244$ and $f^*_{sp}D/U_\infty = 0.511$. The Kelvin–Helmholtz instabilities were not noticed at the P_1 and P_2 probes. For the uncontrolled case, these probes are located in the limits of the wake. However, for the optimal case, the wake amplitude is reduced and the probes P_1 and P_2 are then located outside the wake.

The pressure coefficients for the long-term averaged uncontrolled and optimal case as well as for modes H and L are depicted in Fig. 9b. For the optimal solutions, fewer differences are observed between modes H and L and the long-time average. A comparison between the profiles for the optimal and uncontrolled case reveals that the most important area to reduce the drag is the one located behind the boundary layer detachment zone. A lower pressure value is observed for the optimal cases just before the boundary layer detachment zone for the optimal solutions.

In Fig. 9c and 9d, the averaged streamwise velocity and its fluctuations at $x/D = 1.06$ for the long term averaged and modes for the uncontrolled and optimal cases are presented. At $x/D = 1.06$, the difference between the averaged streamwise velocity profiles for the uncontrolled and optimal case is not significant. However, in the uncontrolled case, the mode H features a comparatively larger separation bubble and higher levels of fluctuations. At this point, the mode L is very similar to the optimal solution. The fluctuations of the optimal solution are always lower than the uncontrolled case. At $x/D = 1.54$ (not shown here), it can be noticed that the averaged streamwise velocity profile associated with mode H of the uncontrolled case is higher at $y/D = 0$ compared with the other solutions. Smaller averaged velocities are also obtained for this mode at $y/D = -1$ and $y/D = 1$. At $x/D = 2.02$ (not shown here), the gap between the mode H (uncontrolled case) and the other solutions is more noticeable. A separation zone can be observed for both the mode L of the uncontrolled case and the optimal solutions. It is also surprising that at this point, the mode H has the lowest streamwise fluctuations.

In Fig. 9e and 9f, the averaged cross-flow velocity and its fluctuations at $x/D = 1.06$ are presented. The averaged cross-flow optimal solution has a similar profile to the uncontrolled case. However, smaller amplitude oscillations behind the cylinder, between $y/D = -0.5$ and $y/D = 0.5$, are observed. High amplitude oscillations are visible for the mode H. At $x/D = 1.54$ (not shown here), the optimal solutions and the mode L of the uncontrolled case behave similarly. Indeed, between $y/D = -0.5$ and $y/D = 0.5$, the averaged cross-stream velocity profile diminishes, before rising and decreasing again. This behaviour is not observed for the long term averaged and mode H of the uncontrolled case. We can also observe that the fluctuations of the cross-flow are reduced downstream the cylinder, between $y/D = -1.0$ and $y/D = 1.0$. The lowest cross-stream fluctuations are associated with the lowest drag coefficient values.

No significant differences are observed between the long term averaged optimal solution and its modes on both the streamwise and cross-stream velocities. Interestingly enough as well, the optimal solutions found were close to the mode L of the uncontrolled case. These observations suggest that the optimal actuation technique drives the flow closer to the L mode and further away from the H mode.

We can also observe in Fig. 9g and 9h the averaged streamwise velocity according to x/D at $y/D = 0$ and the associated fluctuations. For the uncontrolled case, we notice a minimum at $x/D = 1.11$ for the mode H, at $x/D = 1.66$ for the mode L, and at $x/D = 1.55$ for the time-averaged solution. For the optimal case, the minimum of the mode H, mode

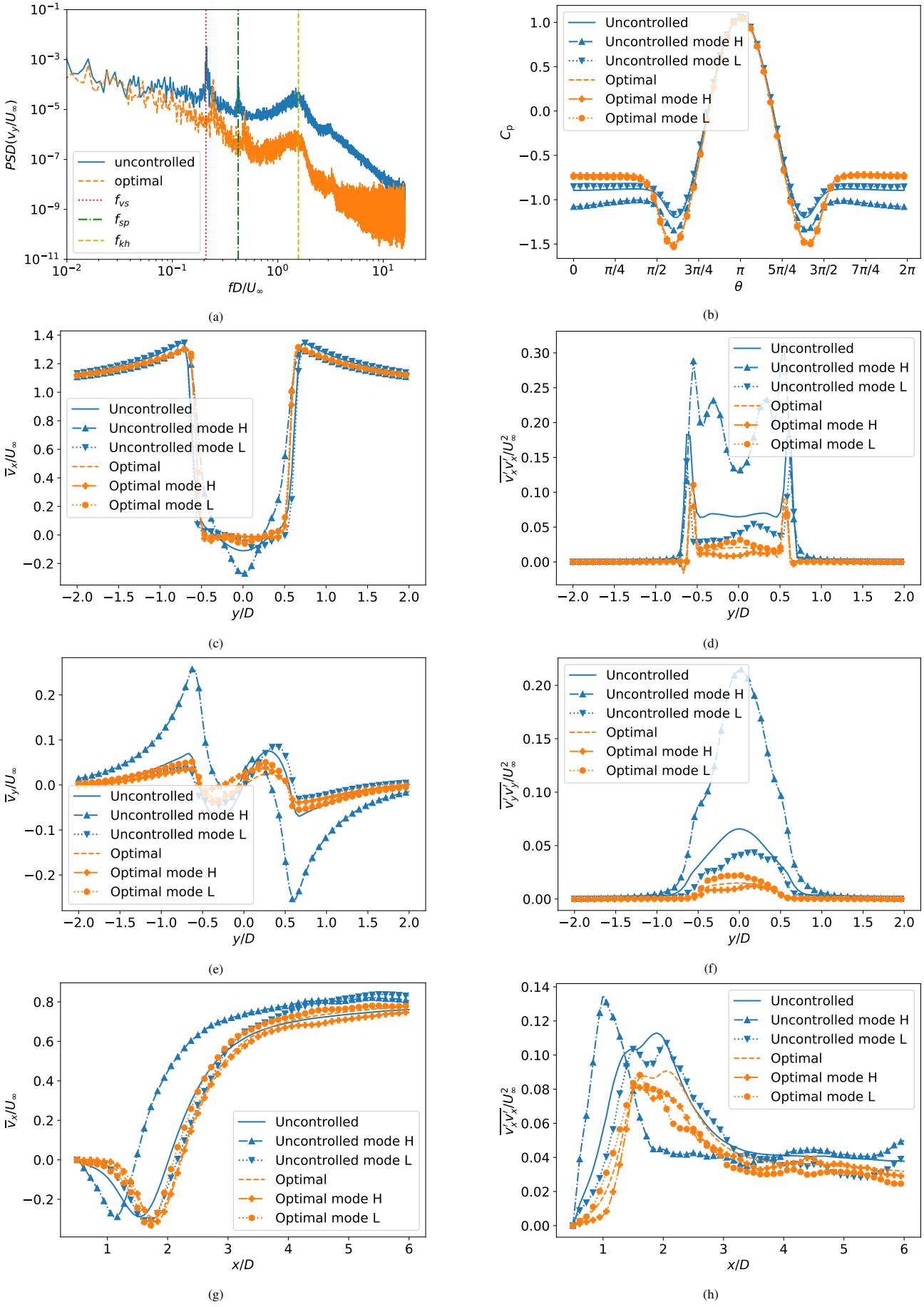
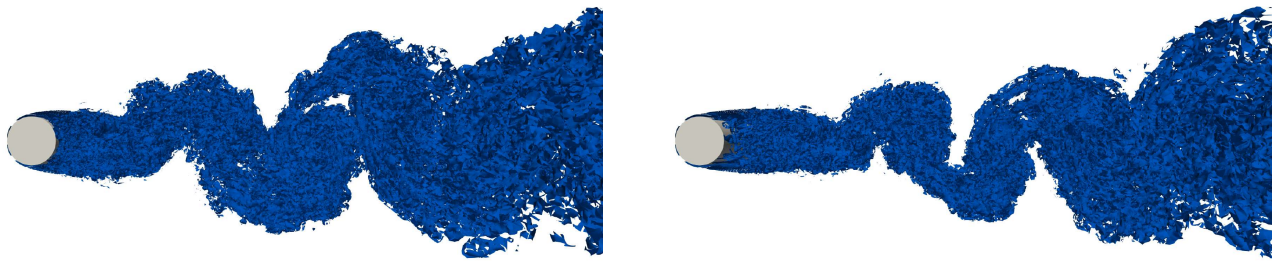
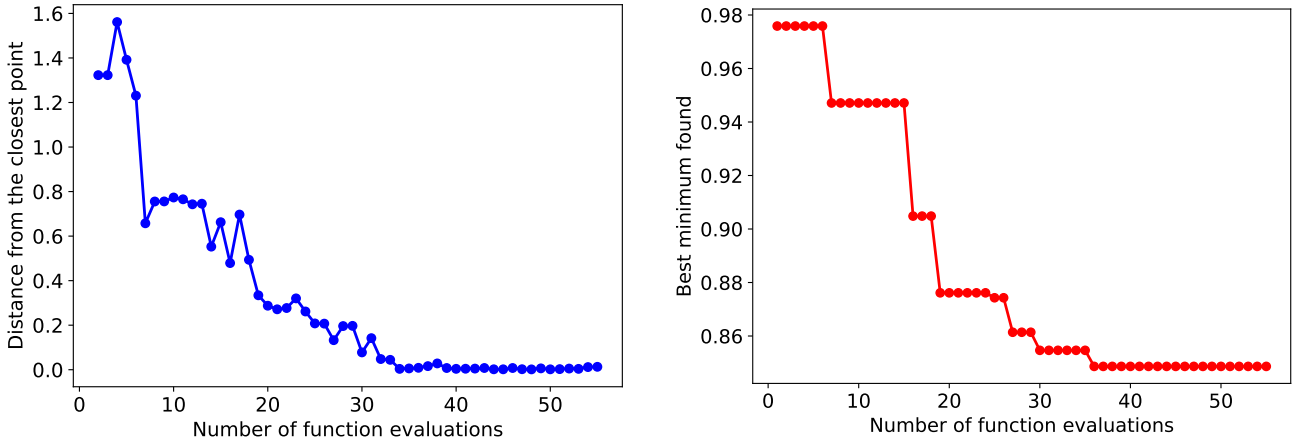


Figure 9: Three-dimensional cylinder at Re = 3900. (a) Power spectral densities of the cross-flow velocity component at the probe P_1. (b) Pressure coefficient averaged in the streamwise direction. (c) Time-averaged streamwise velocity profiles and (d) fluctuations at $x/D = 1.06$. (e) Time-averaged cross-flow velocity profiles and (f) fluctuations at $x/D = 1.06$. (g) averaged streamwise velocity profile at $y/D = 0$ and (h) fluctuations.



(a) (b)

Figure 10: Isosurface of the Q-criterion $Q = 0.1$ for (a) the uncontrolled case and (b) optimal case at $tU_\infty/D = 400$.



(a) (b)

Figure 11: Bayesian Optimization efficiency on the cylinder at Re = 3900. (a) Distance from the closest point already evaluated as a function of the number of function evaluations. (b) Best minimum found as a function of the number of iterations.

L and time-averaged solution are respectively located at $x/D = 1.78$ and $x/D = 1.72$, and $x/D = 1.78$. Regarding the streamwise fluctuations in the wake, we can deduce that the higher the drag the lower and further from the cylinder are the maximum of these fluctuations.

In Fig. 10, the isosurface of the Q-criterion $Q = 0.1$ for the uncontrolled and the optimal instantaneous solutions at $tU_\infty/D = 400$ are represented. In the optimal case, the vortices are delayed further away from the cylinder and thus reduce the drag force on the cylinder.

Finally, we present in Fig. 11 the BO efficiency on the three-dimensional cylinder control case at Re=3900. Fig. 11b represents the best minimum found according to the number of function evaluations. The algorithm was able to find the optimum in 36 iterations (DOE included). In Fig. 11a is represented the distance in the design space from the point currently evaluated to the closest point already evaluated. We can observe that after approximately 40 iterations, the algorithm evaluates points that are very close to previous ones.

5. Conclusions

In this work, the efficiency of Bayesian Optimization (BO) was investigated in the context of drag reduction in the two- and three-dimensional unsteady flow around circular cylinder at Re = 500 and Re = 3900, respectively.

A tangential velocity profile was set at the cylinder surface and an objective function consisting of the root-mean-square of the drag coefficient with and without penalization was minimized. The design parameters were the amplitudes of the tangential velocity set at equidistant points on the cylinder.

In two dimensions, the influence of the number of design parameters, the acquisition functions, the optimizer of the acquisition functions, the kernels, the number of initial points in the DOE and the penalty term was examined. No significant differences were observed on the performance of BO regarding the choice of the optimizer of the acquisition functions, the kernels, the number of initial points in the DOE and the penalty term (except for $\alpha = 0$ where we observed a faster convergence). Regarding the choice of the acquisition functions, NLCB and EI performed well whereas PI resulted into a more aggressive optimization strategy. BO also showed robustness as the number of design variables was increased. Indeed, approximately twice the number of function evaluations were necessary when the number of design parameters was more than doubled. Increasing the penalty term revealed that the most important areas of the cylinder to decrease the drag coefficient were close to the boundary layer separation point. With $\alpha = 0$, we observed no fluctuations downstream the cylinder, and the Von Kármán street was suppressed.

BO was also compared to other optimization algorithms such as CMA-ES, PSO, Nelder-Mead and EGM. BO performed significantly better both in serial and in parallel. The parallel BO approach was superior to the serial BO in terms of iterations. However, the more we increase the number of the parallel evaluations during one iteration, the more function evaluations are required to converge towards the minimum.

Finally, we applied BO on the three-dimensional cylinder. BO found the minimum with 7 design parameters after 36 iterations (5 initial samples included). A 23% drag reduction was observed and the time averaged quantities were examined. The fluctuations of the velocities were reduced and a mode close to the mode L was obtained. The vortices were also delayed further in the wake.

As future work, this technique can be used to tune parameters in more sophisticated control techniques to reduce drag in bluff bodies. Also, the surrogate model can be improved by incorporating gradient information, which might accelerate convergence in some cases. Furthermore, multi-fidelity Gaussian Processes can be used to reduce the cost of the optimization problems.

Acknowledgments

The authors thankfully acknowledge financial support from MINECO through the Programa Estatal de I+D+i Orientada a los Retos de la Sociedad, Grant No. DPI2016-75777-R, MINECO AEI/FEDER, UE. as well as the computer resources at MinoTauro and the technical support provided by Barcelona Supercomputing Center (RES-IM-2019-3-0013).

References

[1] M. Cavazzuti, Optimization Methods: From Theory to Design, Springer, 2013.

[2] A. Jameson, Aerodynamic design via control theory, Journal of Scientific Computing 3 (3) (1988) 233–260.

[3] F. Giannetti, P. Luchini, Structural sensitivity of the first instability of the cylinder wake, Journal of Fluid Mechanics 581 (2007) 167–197.

[4] O. Marquet, D. Sipp, L. Jacquin, Sensitivity analysis and passive control of cylinder flow, Journal of Fluid Mechanics 615 (2008) 221–252.

[5] S. Camarri, A. Iollo, Feedback control of the vortex-shedding instability based on sensitivity analysis, Physics of Fluids 22 (9) (2010) 094102.

[6] C. Talnikar, Q. Wang, Adjoint-based trailing edge shape optimization of a transonic turbine vane using large eddy simulations (2020). arXiv:2011.06744.

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

[8] Y. Li, W. Cui, Q. Jia, Q. Li, Z. Yang, M. Morzyński, B. R. Noack, Explorative gradient method for active drag reduction of the fluidic pinball and slanted Ahmed body (2019). arXiv:1905.12036.

[9] D. B. Fogel, An introduction to simulated evolutionary optimization, IEEE transactions on neural networks 5 (1) (1994) 3–14.

[10] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[11] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[12] M. A. A. Chikh, I. Belaidi, S. Khelladi, J. Paris, M. Deligant, F. Bakir, Efficiency of bio-and socio-inspired optimization algorithms for axial turbomachinery design, Applied Soft Computing 64 (2018) 282–306.

[13] M. Milano, P. Koumoutsakos, A clustering genetic algorithm for cylinder drag optimization, Journal of Computational Physics 175 (1) (2002) 79–107.

[14] T. K. Sengupta, K. Deb, S. B. Talla, Control of flow using genetic algorithm for a circular cylinder executing rotary oscillation, Computers & fluids 36 (3) (2007) 578–600.

[15] T. Mengistu, W. Ghaly, Single and multipoint shape optimization of gas turbine blade cascades, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004, p. 4446.

[16] D. Yanhui, W. Wenhua, F. Zhaolin, C. Ti, An introduction of aerodynamic shape optimization platform for compressor blade, in: Turbo Expo: Power for Land, Sea, and Air, Vol. 49712, American Society of Mechanical Engineers, 2016, p. V02CT39A031.

[17] A. Safari, K. H. Hajikolaei, H. Lemu, G. Wang, A high-dimensional model representation guided pso methodology with application on compressor airfoil shape optimization, in: Turbo Expo: Power for Land, Sea, and Air, Vol. 49712, American Society of Mechanical Engineers, 2016, p. V02CT45A013.

[18] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. K. Tucker, Surrogate-based analysis and optimization, Progress in Aerospace Sciences 41 (1) (2005) 1–28.

[19] E. Brochu, V. M. Cora, N. de Freitas, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning (2009). arXiv:1012.2599.

[20] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE 104 (1) (2015) 148–175.

[21] H. J. Kushner, A new method of locating the maximal point of an arbitrary multipeak curve in the presence of noise, Journal of Basic Engineering 86 (1) (1964) 97–106.

[22] J. Moćkus, V. Tiesis, A. Źilinskas, The Application of Bayesian Methods for Seeking the Extremum. Vol. 2, in: Towards Global Optimization, 1978, pp. 117–128.

[23] Krige, D. G., A statistical approach to some, basic mine valuation problems on the witwatersand., Journal of the Chemical Metallurgical & Mining Society of South Africa 52 (6) (1951) 119–139.

[24] J. Sacks, W. J. Welch, J. M. Toby, H. P. Wynn, Design and Analysis of Computer Experiments, Statistical Science 4 (4) (1989) 409–435.

[25] D. R. Jones, M. Schonlau, W. J. Welch, Efficient Global Optimization of Expensive Black - Box Functions, Journal of Global Optimization 13 (4) (1998) 455–492.

[26] C. Audet, J. E. Dennis Jr, D. W. Moore, A surrogate-model-based method for constrained optimization, in: 8th symposium on multidisciplinary analysis and optimization, 2000, p. 4891.

[27] D. Huang, T. T. Allen, W. I. Notz, N. Zheng, Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models D. Huang, Journal of Global Optimization 34 (3) (2006) 441–466.

[28] P. I. Frazier, A Tutorial on Bayesian Optimization (2018). arXiv:1807.02811.

[29] S. Jeong, M. Murayama, K. Yamamoto, Efficient optimization design method using Kriging model., Journal of Aircraft 42 (2) (2005) 413–420.

[30] R. Duvigneau, P. Chandrashekar, Kriging-based optimization applied to flow control, International Journal for Numerical Methods in Fluids 69 (11) (2012) 1701–1714.

[31] C. Talnikar, P. Blonigan, J. Bodart, Q. Wang, Parallel optimization for large eddy simulations (2014). arXiv:1410.8859.

[32] O. Mahfoze, S. Laizet, A. Wynn, Bayesian optimisation of intermittent wall blowing for drag reduction of a spatially evolving turbulent boundary layer, in: Proceedings of the Tenth International Conference on Computational Fluid Dynamics, Barcelona, Spain, 2018, pp. 1–17.

[33] R. Lam, M. Poloczek, P. Frazier, K. E. Willcox, Advances in Bayesian Optimization with Applications in Aerospace Engineering, 2018 AIAA Non-Deterministic Approaches Conference (2018) 1656.

[34] S. Rashidi, M. Hayatdavoodi, J. A. Esfahani, Vortex shedding suppression and wake control: A review, Ocean Engineering 126 (2016) 57–80.

[35] Z. Li, I. M. Navon, M. Y. Hussaini, F. X. Le Dimet, Optimal control of cylinder wakes via suction and blowing, Computers and Fluids 32 (2) (2003) 149–171.

[36] P. Catalano, M. Wang, G. Iaccarino, I. F. Sbalzarini, Optimization of cylinder flow control via actuators with zero net mass flux, in: Proceedings of the Summer Program, 2002, p. 297.

[37] P. Meliga, E. Boujo, M. Meldi, F. Gallaire, Revisiting the drag reduction problem using adjoint-based distributed forcing of laminar and turbulent flows over a circular cylinder, European Journal of Mechanics, B/Fluids 72 (2018) 123–134.

[38] X. Mao, H. M. Blackburn, S. J. Sherwin, Nonlinear optimal suppression of vortex shedding from a circular cylinder, Journal of Fluid Mechanics 775 (4) (2015) 241–265.

[39] X. Mao, E. Pearson, Drag reduction and thrust generation by tangential surface motion in flow past a cylinder, Theoretical and Computational Fluid Dynamics 32 (3) (2018) 307–323.

[40] A. Forrester, A. Sobester, A. Keane, Engineering design via surrogate modelling: a practical guide, John Wiley & Sons, 2008.

[41] A. Sóbester, S. J. Leary, A. J. Keane, On the design of optimization strategies based on global response surface approximation models, Journal of Global Optimization 33 (1) (2005) 31–59.

[42] F. D. Witherden, A. M. Farrington, P. E. Vincent, PyFR: An open source framework for solving advection-diffusion type problems on streaming architectures using the flux reconstruction approach, Computer Physics Communications 185 (11) (2014) 3028–3040.

[43] H. T. Huynh, A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods, in: 18th AIAA Computational Fluid Dynamics Conference, 2007, p. 4079.

[44] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for machine Learning, MIT Press, 2006.

[45] E. Schulz, M. Speekenbrink, A. Krause, A tutorial on Gaussian process regression with a focus on exploration-exploitation scenarios, Journal of Mathematical Psychology 85 (2018) 1 – 16.

[46] D. D. Cox, S. John, A statistical method for global optimization, in: [Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 1992, pp. 1241–1246.

[47] Jones D.R., A Taxonomy of Global Optimization Methods Based on Response Surfaces, Journal of Global Optimization 21 (4) (2001) 345–383.

[48] J. González, Z. Dai, P. Hennig, N. Lawrence, Batch bayesian optimization via local penalization, in: Artificial intelligence and statistics, PMLR, 2016, pp. 648–657.

[49] The GPyOpt authors, GPyOpt: A bayesian optimization framework in python, `http://github.com/SheffieldML/GPyOpt`, accessed: 2020-09-17 (2016).

[50] C. Hirsch, Numerical Computation of Internal and External Flows, volume 2, John Wiley & Sons, 1990.

[51] M. Fosas de Pando, miguelfp/ibmos: Initial release, `https://doi.org/10.5281/zenodo.3757783`, accessed: 2020-09-17 (Apr. 2020). doi:10.5281/zenodo.3757783.

[52] N. Hansen, The CMA Evolution Strategy: A Tutorial (2016). arXiv:1604.00772.

[53] F. Gao, L. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, Computational Optimization and Applications 51 (1) (2012) 259–277.

[54] B. C. Vermeire, F. D. Witherden, P. E. Vincent, On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools, Journal of Computational Physics 334 (2017) 497–521.

[55] O. Lehmkuhl, I. Rodríguez, R. Borrell, A. Oliva, Low-frequency unsteadiness in the vortex formation region of a circular cylinder, Physics of Fluids 25 (8) (2013) 085109.

[56] P. Parnaudeau, J. Carlier, D. Heitz, E. Lamballais, Experimental and numerical studies of the flow over a circular cylinder at reynolds number 3900, Physics of Fluids 20 (8) (2008) 085101.

[57] F. D. Witherden, B. C. Vermeire, P. E. Vincent, Heterogeneous computing on mixed unstructured grids with PyFR, Computers and Fluids 120 (2015) 173–186.

[58] N. R. Lomb, Least-squares frequency analysis of unequally spaced data, Astrophysics and Space Science 39 (2) (1976) 447–462.

[59] M. Zhao, L. Cheng, B. Teng, D. Liang, Numerical simulation of viscous flow past two circular cylinders of different diameters, Applied Ocean Research 27 (1) (2005) 39–55.

[60] G. X. Wu, Z. Z. Hu, Numerical simulation of viscous flow around unrestrained cylinders, Journal of Fluids and Structures 22 (3) (2006) 371–390.

[61] H. Baek, G. E. Karniadakis, Suppressing vortex-induced vibrations via passive means, Journal of Fluids and Structures 25 (5) (2009) 848–866.

[62] Y. G. Liu, L. H. Feng, Suppression of lift fluctuations on a circular cylinder by inducing the symmetric vortex shedding mode, Journal of Fluids and Structures 54 (2015) 743–759.

[63] M. Breuer, Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modeling aspects, International journal for numerical methods in fluids 28 (9) (1998) 1281–1302.

[64] A. G. Kravchenko, P. Moin, Numerical studies of flow over a circular cylinder at $Re_D = 3900$, Physics of fluids 12 (2) (2000) 403–417.

[65] J. Franke, W. Frank, Large eddy simulation of the flow past a circular cylinder at $Re_D = 3900$, Journal of wind engineering and industrial aerodynamics 90 (10) (2002) 1191–1206.

[66] K. Mahesh, G. Constantinescu, P. Moin, A numerical method for large-eddy simulation in complex geometries, Journal of Computational Physics 197 (1) (2004) 215–240.

[67] A. Mani, P. Moin, M. Wang, Computational study of optical distortions by separated shear layers and turbulent wakes, Journal of Fluid Mechanics 625 (2009) 273–298.

Appendix A. Grid independence study and validation of the two-dimensional case

We present here a grid independence study of the two-dimensional case presented in Section 3. A finer and coarser grid consisting of a C-H grid topology centred around the cylinder is considered. In both cases, the computational domain is given by $[-9D, 25D] \times [-9D, 9D]$, and it is refined in the region $[-4D, 15D] \times [-4D, 4D]$. The main parameters of the meshes are presented in Table A.2. It should be noted that depending on the order of the elements p that is chosen, the number of solution points will vary as well as the distance of the first solution point to the cylinder surface.

Mesh	# points cylinder	$\Delta y/D$	#steps extrusion	# Quads	# Tri
coarse	32	0.0678	7	224	3020
fine	65	0.041	14	910	9901

Table A.2: Meshes used for validation. The # points cylinder is the number of points along the cylinder, Δy is the distance of the first mesh point to the cylinder, #steps extrusion corresponds to the number of the steps performed for the extrusion around the cylinder in order to build the boundary layer, # Quads is the number of quads used (all in the boundary layer), and # Tri the number of triangles.

Simulations based on these two meshes were performed for three different element orders p, namely second, third and fourth order, and four different time steps $U_\infty \Delta t/D$: 10^{-3}, $5 \cdot 10^{-4}$, $2.5 \cdot 10^{-4}$ and 10^{-3} with adaptive time-stepping. In Table A.3, the Strouhal number St, the time-averaged drag coefficient $\overline{C_\mathrm{d}}$ and the RMS value of the lift coefficient C_l' for the uncontrolled case and the optimal solution for the case with 32 actuators and $\alpha = 8$ are presented. The uncontrolled case was run from $t = 0$ to $t = 100D/U_\infty$ and the comparative statistics were calculated from $t = 40D/U_\infty$ to $t = 100D/U_\infty$. We ran the optimal case starting from the long-time integration of the uncontrolled case ($100D/U_\infty$) during $100D/U_\infty$ additional time units. In that case, we computed the averaged statistics from $t = 50D/U_\infty$ to $t = 100D/U_\infty$.

A comparison between the uncontrolled simulations and results reported in the literature is also given. The configuration highlighted in bold is selected for computational efficiency reasons.

Mesh	Order	$U_\infty \Delta t/D$	St	$\overline{C_\mathrm{d}}$	C_l'	St*	$\overline{C_\mathrm{d}}^*$	$C_l^{*\prime}$	CPU time (s)
coarse	2	10^{-3}	0.232	1.444	0.847	0.261	0.764	0.205	739
coarse	2	$5 \cdot 10^{-4}$	0.232	1.444	0.847	0.261	0.764	0.205	1385
coarse	2	$2.5 \cdot 10^{-4}$	0.232	1.444	0.847	0.261	0.764	0.205	2733
coarse	2	adaptive	0.232	1.444	0.847	0.261	0.764	0.205	270
coarse	3	10^{-3}	0.228	1.492	0.872	0.245	0.832	0.317	805
coarse	3	$5 \cdot 10^{-4}$	0.228	1.492	0.872	0.245	0.832	0.317	1647
coarse	3	$2.5 \cdot 10^{-4}$	0.228	1.492	0.872	0.245	0.832	0.317	3276
coarse	**3**	**adaptive**	**0.228**	**1.492**	**0.872**	**0.245**	**0.832**	**0.317**	**574**
coarse	4	10^{-3}	0.228	1.490	0.865	0.245	0.836	0.310	1115
coarse	4	$5 \cdot 10^{-4}$	0.228	1.490	0.865	0.245	0.836	0.310	2226
coarse	4	$2.5 \cdot 10^{-4}$	0.228	1.490	0.865	0.245	0.836	0.310	4580
coarse	4	adaptive	0.228	1.490	0.865	0.245	0.836	0.310	1228
fine	2	10^{-3}	0.228	1.487	0.878	0.245	0.828	0.314	1248
fine	2	$5 \cdot 10^{-4}$	0.228	1.487	0.878	0.245	0.828	0.314	2551
fine	2	$2.5 \cdot 10^{-4}$	0.228	1.487	0.878	0.245	0.828	0.314	5178
fine	2	adaptive	0.228	1.487	0.878	0.245	0.828	0.314	941
fine	3	$5 \cdot 10^{-4}$	0.228	1.492	0.868	0.244	0.838	0.314	3501
fine	3	$2.5 \cdot 10^{-4}$	0.228	1.492	0.868	0.244	0.838	0.314	7027
fine	3	adaptive	0.228	1.492	0.868	0.244	0.838	0.314	2462
fine	4	$2.5 \cdot 10^{-4}$	0.228	1.484	0.848	0.244	0.842	0.316	20208
fine	4	adaptive	0.228	1.484	0.848	0.244	0.842	0.316	12734
Ref. (59)	-	-	0.23	1.463	0.837	-	-	-	-
Ref. (60)	-	-	-	1.28	0.622	-	-	-	-
Ref. (61)	-	-	0.235	1.518	0.876	-	-	-	-
Ref. (62)	-	-	0.225	1.440	0.818	-	-	-	-

Table A.3: Grid independence study and validation of the two-dimensional case at Re = 500. St, $\overline{C}_\mathrm{d}$, C_l' are respectively, the Strouhal number, the average drag coefficient and the RMS value of the lift coefficient for the uncontrolled case. St*, $\overline{C}_\mathrm{d}^*$, and C_l^* refer to the quantities for the optimal solution found in Section 3 with 32 actuators and $\alpha = 8$. In the last column, the computational cost of each simulation is given in seconds. Computations were performed on a cluster composed of 48 nodes, where each node has two Intel Xeon E5 2670. Since these CPUs are octo core, 16 cores are available on each node. Each simulation was run using all the cores on one node.

Appendix B. Validation of the three-dimensional case

We present here the details of the simulation for the three-dimensional cylinder at Re = 3900. The extent of the computational domain is $[-9D, 25D] \times [-9D, 9D] \times [0, \pi]$ in respectively the streamwise, cross-flow and spanwise directions. The cylinder is centered at $(0, 0, 0)$. The mesh is composed of 79 344 prismatic elements and 227 298 tetrahedral elements. Again, the Mach number is set to Ma = 0.2, the Prandtl number is Pr = 0.71 and $\gamma = 1.4$. Riemann invariant boundary conditions are set on the far-field boundary conditions whereas periodicity is imposed in the spanwise direction. Finally, at the wall of the cylinder, the tangential velocity profile is given by Eq. (23), the normal velocity is set to zero and the temperature is set to the free-stream value. A comparison against previously reported results is presented in Table B.4.

Case	$f_{vs}D/U_\infty$	$\overline{C}_\mathrm{d}$	L_d/D	$-\overline{C}_\mathrm{pb}$
Ref. (55)	0.215	1.015	1.36	0.935
Ref. (63)	0.215	1.016	1.372	0.941
Ref. (64)	0.21	1.04	1.35	0.94
Ref. (65)	0.209	0.978	1.64	0.85
Ref. (66)	0.218	1.0	1.35	-
Ref. (67)	0.206	0.99	-	0.86
Ref. (56)	0.208	-	1.51	-
Present study	**0.208**	**1.027**	**1.51**	**0.895**

Table B.4: Comparison of the three-dimensional long-time averaged uncontrolled case at Re = 3900 with the literature. $f_{vs}D/U_\infty$ is the non-dimensional vortex shedding frequency, $\overline{C}_\mathrm{d}$ the mean drag coefficient, L_d/D the length of the recirculation zone measured from the aft of the cylinder and $\overline{C}_\mathrm{pb}$ the base pressure coefficient.

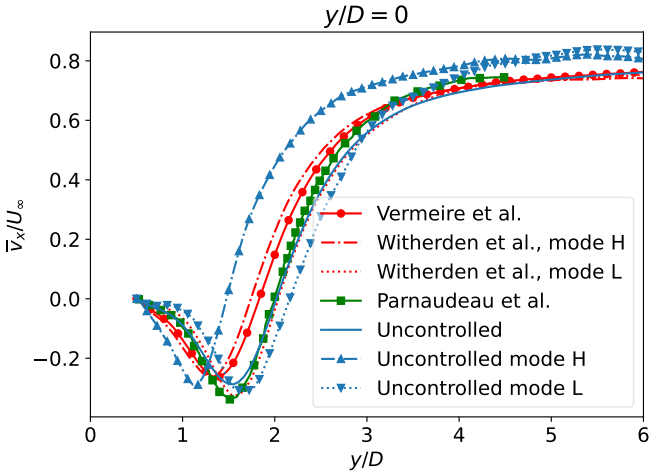


Figure B.12: Three-dimensional cylinder at Re = 3900. Time-averaged streamwise velocity profiles. Comparison with the long time-averaged quantities of Vermeire *et al.* (54), with the modes H and L of Witherden *et al.* (57) and the experimental results of Parnaudeau *et al.* (56).

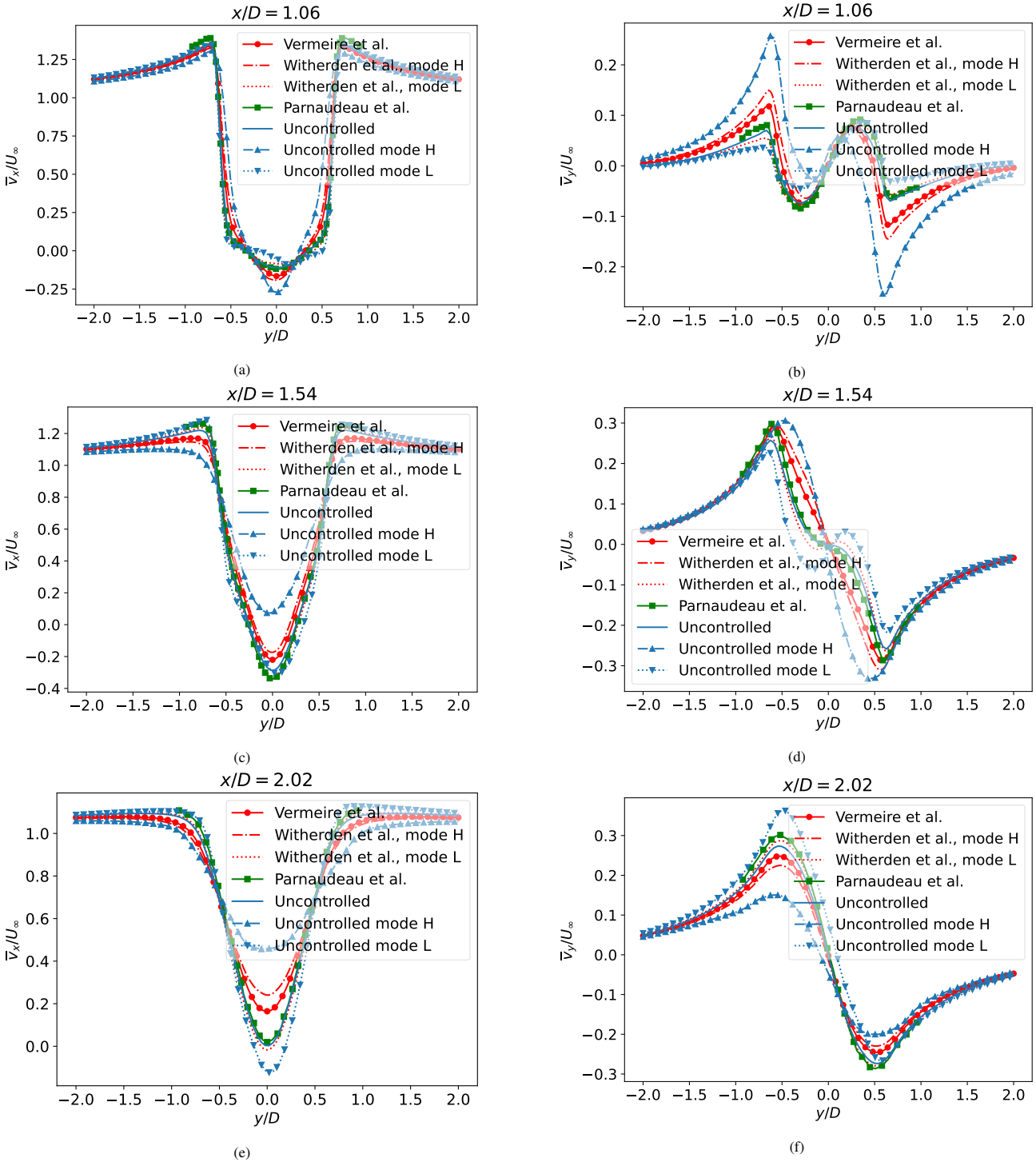


Figure B.13: Three-dimensional cylinder at Re = 3900. Time-averaged streamwise (left) and cross-flow (right) velocity profiles. Top row: $x/D = 1.06$, middle row: $x/D = 1.54$, bottom row: $x/D = 2.02$. Comparison with the long time-averaged quantities of Vermeire *et al.* (54), with the modes H and L of Witherden *et al.* (57) and the experimental results of Parnaudeau *et al.* (56).