Universidad de Cádiz

Departamento de Ingeniería Mecánica y Diseño Industrial

Variations on Bayesian Optimization Applied to Numerical Flow Simulations

Author:
Anthony Larroque

Advisors:
Miguel Fosas de Pando
Luis Lafuente Molinero

March 9, 2023

A thesis presented for the degree of
Doctor of Philosophy

Variations on Bayesian Optimization Applied to Numerical Flow Simulations

Anthony Larroque

Abstract

Bayesian Optimization (BO) has recently regained interest in optimization problems involving expensive black-box objective functions. Several variants have been proposed in the literature, such as including gradient and/or multi-fidelity information, and it has been extended to multi-objective optimization problems. Despite its recent applications to numerical flow simulations, the efficiency of this method and its variants remains to be characterized in typical applications involving canonical flows.

In this work, the efficiency of classical BO and alternative derivative-free methods is compared on a simplified flow case, i.e. drag reduction in the two-dimensional flow around a cylinder. The application of BO to complex flows is then showcased by considering a three-dimensional case at Reynolds number Re = 3900. Next, the performance of BO with gradient and/or multi-fidelity information is investigated for global modelling and optimization on typical benchmark objective functions and on the cylinder case at Re = 200. Finally, an algorithm combining dimension reduction and Multi-objective Bayesian Optimization (MOBO) is proposed.

It is found that BO was more efficient than other derivative-free alternatives and showed promising results on the three-dimensional cylinder at Re = 3900 by reducing drag by 23 %. The performance of the algorithm was further improved when multi-fidelity and/or gradient information was included, both for modelling and optimization. Including gradient information on the low-fidelity model was useful for global modelling and to decrease rapidly the objective function in a BO framework. On the contrary, adding derivative information on the high-fidelity model generally gave the most accurate approximation of the minimum but was inefficient for global modelling when the computational cost of the gradient was high. Finally, the developed algorithm combining dimension reduction and MOBO enabled us to obtain more precise and diverse minima.

Acknowledgements

Quand j'ai commencé cette thèse, je ne pensais pas qu'elle serait si longue et difficile. Pour moi qui aime normalement les défis et m'améliorer, j'ai trouvé en la thèse un challenge inimaginable. J'ai notamment pensé à abandonner plusieurs fois tant le défi était immense mais chaque fois me retenaient le désir de ne pas baisser les bras et la soif d'apprendre et comprendre. Quand je vois le résultat final, je suis heureux d'avoir pu aller au bout.

Cette thèse, je la dois à beaucoup de personnes puisque je n'aurais jamais pu la réaliser seul. Et je crois que c'est un nouveau défi qui se présente à moi que de remercier toute ces personnes qui ont contribué directement ou indirectement à ce que cette thèse voit le jour. Je dois avouer que je ne sais même pas en quelle langue l'écrire : français, español, english ? Creo que, ya que hice el doctorado en España, tiene sentido de empezar en español.

Así que mis primeros agradecimientos van a mi primer director de tesis: Miguel Fosas de Pando. Gracias por tu paciencia, tu generosidad, tu sentido del humor y los consejos que me has dado a lo largo de la tesis. Creo que fuiste bastante exigente con el nivel de la escritura, pero creo también que me hacía falta para superar mis límites y mejorar, ya que a veces puedo ser un poco flojo. ¡Gracias también por venir buscarme con el coche a El Palmar (creo que ese era el nombre del pueblo) cuando no presté atención al horario del autobús! ¡No lo olvido!

Los siguientes agradecimientos van a mi segundo director de tesis: Luis Lafuente Molinero. Aunque quedamos pocas veces por culpa de tu muy apretada agenda, siempre me diste consejos pertinentes y/o originales en los momentos importantes. ¡Así que gracias por eso, por tu sentido del humor y tu bondad!

La experiencia de esta tesis tampoco la hubiera vivido igual sin las personas que conocí en Cádiz. Creo que la lista es demasiado larga para que quepa en una página, especialmente con los nombres tan largos que existen en España. Así que a todas las personas con quienes compartí unas palabras, un café, un despacho, una comida en la ESI, unas pachangas de baloncesto o fútbol, comida en un restaurante, bebidas en un bar o en discotecas o con quién compartí piso. ¡Os lo agradezco! ¡Gracias por haber acogido un guiri como yo y haber intentado hacer de él un gaditano! Tengo que admitiros, sin embargo, que todavía no entiendo todas las chirigotas, así que quizás me hace falta algo más de tiempo para completar mi integración.

Retour au français ! Je remercie mes amis français de toujours (ou presque) qui m'ont offert une bulle d'oxygène pendant cette thèse. Là encore, la liste serait certainement trop longue. Merci de m'avoir permis de m'évader ou me réconforter durant l'orage par des paroles ou des gestes. Votre amitié m'est précieuse et bien que les histoires avec vous soient souvent rocambolesques et arrosées, je ne les changerais pour rien au monde.

Last but not least (un peu d'anglais dans ces remerciements, il n'y en avait pas encore !), je tiens à remercier toute ma famille qui eux aussi m'ont apporté cette précieuse bulle d'oxygène à travers des paroles, des gestes et des bons petits repas agrémentés de vins. Plus particulièrement, merci à mes parents qui m'ont toujours soutenu quand je souhaitais abandonner, m'ont permis de réaliser mes études afin d'avoir un futur prometteur et d'avoir largement participé à l'homme que je suis devenu (bon, je ne sais pas si c'est une bonne chose mais bon vous y avez participé !).

A toutes ces personnes, cette thèse je vous la dois et puisque toutes ces relations m'ont façonné, il y a forcément un peu de vous dans cette thèse !

Este trabajo ha sido financiado por el Programa Estatal de I+D+i Orientada a los Retos de la Sociedad, Ref. DPI2016-75777-R, MINECO AEI/FEDER, UE.

Contents

List of Figures

Chapter 1

Introduction

This thesis deals with Bayesian Optimization (BO) in the context of Computational Fluid Dynamics (CFD). More specifically, BO and some of the recently proposed variants are assessed in the context of numerical flow simulations to determine the range of applications and their computational performance.

The organization of this chapter is as follows. In Section 1.1 the state of the art is presented. The different optimization methods, the BO framework, the improvement of BO with gradients and/or multi-fidelity models as well as the methods that have been proposed to apply BO to high dimensional problems are discussed. In Section 1.2, the goals of the thesis are detailed. Finally, the general outline of the thesis is presented in Section 1.3.

1.1 State of the art

1.1.1 Optimization methods

With the constantly increasing computing power, high-fidelity simulations are routinely been used to predict the behaviour and performance of technological devices. Once a design is proposed, it is not unusual to explore several designs in order to minimize (or maximize) a desired objective function. This process may sometimes be carried out by brute force and intuition since in the most simple cases, the objective functions in engineering are monotonic [13]. However, in more complex configurations stemming from high-fidelity simulations, finding the optimum design becomes thornier.

Optimization methods [11] can assist in this process by automating the choices of the designs to test. Several classifications of optimization methods exist in the literature: e.g. deterministic/stochastic, local/global, gradient-based/derivative-free. The so-called no-free lunch theorem [120] states that if we compute the average efficiency of all the methods on all the existing optimization problems, all algorithms perform equal. Nonetheless, depending on the features of a given problem, an algorithm may be preferable to alternatives [11]. In the following, the characteristics of gradient-based and derivative-free methods are presented and compared.

Gradient-based methods rely on derivative information of the objective function to determine the next design to evaluate. Indeed, the gradient provides information about the local changes of the objective function. A search direction where the objective function has maximum local growth (for maximization problems) or

decrease (for minimization problems) can then be obtained. An adequate step in this direction is set and the next design to evaluate is obtained. Gradient-based methods have the advantage of being accurate and benefit from fast convergence rates. However, they may often be trapped in local optima and are difficult to parallelize. An additional drawback is that obtaining the gradient of the objective function requires, in typical engineering problems, a significant implementation effort and a non-negligible computational cost. This cost increases dramatically for high dimensional spaces when these gradients are computed with finite differences as it requires $N + 1$ objective functions evaluations, N being the dimension of the design space. Fortunately, Lions [68] developed the adjoint method where the cost of obtaining the gradient is independent of the number of design variables. This approach was used by Pironneau [92] in fluid mechanics and later by Jameson [40] in aerodynamics. Still, the cost of solving the adjoint equations is often equal or higher than the cost of evaluating the objective function. Implementing the adjoint method requires an important development effort and it is memory intensive for unsteady simulations. Further to this, large errors on the gradient can also be observed in the case of turbulent flows as illustrated by Talnikar and Wang [111], rendering the application of gradient-based methods nontrivial. Arguably, the most popular gradient-based methods are the quasi-Newton methods where the Hessian is approximated to find the minimum, such as the L-BFGS-B algorithm [7].

On the other hand, the derivative-free optimization methods do not require gradient information. The advantages of this type of methods are that they can more easily escape from local optima and several of these optimization methods can be parallelized. These methods only require objective function evaluations and can then be more easily wrapped around a black-box system than gradient-based optimization when the gradients are obtained with the adjoint method. However, derivative-free methods suffer from lower convergence rates and are less accurate than gradient-based optimization when gradients can be numerically computed with high accuracy. There are numerous algorithms falling into the category of the derivative-free methods. One can cite for example the Nelder-Mead algorithm [86] that relies on contraction, reflection, expansion and shrink of design simplices. Also many algorithms inspired by the nature came to live as derivative-free methods such as genetic algorithms [19] or particle swarm optimization [47]. They generally involve an initial population that will be attracted towards the optimum through a weighting system.

Even if it cannot be directly classified as an optimization method, the Response Surface Methodology (RSM) [94] is a widely used framework for the optimization of black-box functions. The idea is to firstly build a surrogate model with the observations of the objective function at carefully selected design points. This step is known as the Design of Experiments (DOE). Then the surrogate model is optimized with gradient-based or derivative-free methods to determine the most promising design. The observation of the objective function at this design point is then added to the model and a new iteration begins. The benefit of this approach is that the optimization is performed on the surrogate model and not directly on the objective function. Finding the optimum of the surrogate model becomes then much cheaper than finding the optimum of the true underlying objective function. The drawbacks of this framework are that it may be difficult to estimate the accuracy of the model and building a reliable surrogate model in high dimensions requires a significant number of samples.

1.1.2 Bayesian Optimization

Bayesian Optimization [6, 103, 25] has recently emerged as a promising optimization technique through its applications in machine learning. This technique relies on a surrogate model, typically a Gaussian Process (GP) [96, 101]. With the GP (also called Kriging), the objective function is considered as the realization of a stochastic process

$$f(\mathbf{s}) \sim \mathrm{GP}(\mu_0(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')), \tag{1.1}$$

where f is the objective function, $\mathbf{s}$ is a design point, and GP is defined by its mean function $\mu_0(\mathbf{s})$ and its covariance function $k(\mathbf{s}, \mathbf{s}')$.

The GP is a function distribution over a continuous domain. With this model, if we consider n observations of the objective function $\mathbf{f}_{1:n} = (f(\mathbf{s}_1), f(\mathbf{s}_2), \ldots, f(\mathbf{s}_n))^\mathsf{T}$, at n design points $\mathbf{s}_{1:n} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n)^\mathsf{T}$, these objective function values are then initially modelled as the realizations of a multivariate normal distribution given by

$$\mathbf{f}_{1:n}|\mathbf{s}_{1:n} \sim \mathcal{N}(\boldsymbol{\mu}_0(\mathbf{s}_{1:n}), \mathbf{K}) = P(\mathbf{f}_{1:n}|\mathbf{s}_{1:n}), \tag{1.2}$$

where $\mathbf{f}_{1:n}|\mathbf{s}_{1:n}$ should be interpreted as $\mathbf{f}_{1:n}$ given $\mathbf{s}_{1:n}$, $\mathcal{N}$ is the multivariate normal distribution, $\boldsymbol{\mu}_0(\mathbf{s}_{1:n})$ the mean function at $\mathbf{s}_{1:n}$ and $\mathbf{K} = [k_{ij}]$ with $k_{ij} = k(\mathbf{s}_i, \mathbf{s}_j)$ and $1 \leq i, j \leq n$. $P(\mathbf{f}_{1:n}|\mathbf{s}_{1:n})$ is known as the prior distribution, and more specifically, we have

$$P(\mathbf{f}_{1:n}|\mathbf{s}_{1:n}) = \frac{1}{(2\pi)^{n/2}|\mathbf{K}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{f}_{1:n} - \boldsymbol{\mu}_0(\mathbf{s}_{1:n}))^\mathsf{T}\mathbf{K}^{-1}(\mathbf{f}_{1:n} - \boldsymbol{\mu}_0(\mathbf{s}_{1:n}))\right), \tag{1.3}$$

where $|\cdot|$ denotes the determinant.

It may happen in practice that we do not have access to the true objective function values $\mathbf{f}_{1:n}$ but to noisy observations $\mathbf{q}_{1:n}$. Typically, such observations are modelled by

$$\mathbf{q}_{1:n} = \mathbf{f}_{1:n} + \boldsymbol{\eta}_{1:n}, \tag{1.4}$$

where $\boldsymbol{\eta}_{1:n} = (\eta_1, \eta_2, \ldots, \eta_n)^\mathsf{T}$. Each η_i is considered to be drawn from a normal distribution with zero mean:

$$\eta_i \sim \mathcal{N}(0, \sigma_\eta^2) = \frac{1}{\sigma_\eta\sqrt{2\pi}} \exp\left(-\frac{\eta_i^2}{2\sigma_\eta^2}\right). \tag{1.5}$$

With these considerations, the prior probability of the observations $\mathbf{q}_{1:n}$ reads

$$\begin{aligned} P(\mathbf{q}_{1:n}|\mathbf{s}_{1:n}) &= \mathcal{N}(\boldsymbol{\mu}_0(\mathbf{s}_{1:n}), \mathbf{K} + \sigma_\eta^2\mathbf{I}_n) \\ &= \frac{1}{(2\pi)^{n/2}|\mathbf{K} + \sigma_\eta^2\mathbf{I}_n|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{q}_{1:n} - \boldsymbol{\mu}_0(\mathbf{s}_{1:n}))^\mathsf{T}[\mathbf{K} + \sigma_\eta^2\mathbf{I}_n]^{-1}(\mathbf{q}_{1:n} - \boldsymbol{\mu}_0(\mathbf{s}_{1:n}))\right), \end{aligned} \tag{1.6}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix.

If we consider now the objective function value f_{n+1} at a design point $\mathbf{s}_{n+1}$, the joint probability of the noisy observations $\mathbf{q}_{1:n}$ and f_{n+1} is given by

$$P(\mathbf{q}_{1:n}, f_{n+1}|\mathbf{s}_{1:n}, \mathbf{s}_{n+1}) = \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_0(\mathbf{s}_{1:n}) \\ \mu_0(\mathbf{s}_{n+1})\end{pmatrix}, \begin{pmatrix}\mathbf{K} + \sigma_\eta^2\mathbf{I}_n & \mathbf{k} \\ \mathbf{k}^\mathsf{T} & k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1})\end{pmatrix}\right), \tag{1.7}$$

where $\mathbf{k} = [k_{i,n+1}]$.

The name Bayesian Optimization comes from the famous Bayes' theorem that states in that case:

$$P(f_{n+1}|\mathbf{q}_{1:n}, \mathbf{s}_{1:n}, \mathbf{s}_{n+1}) = \frac{P(\mathbf{q}_{1:n}|f_{n+1}, \mathbf{s}_{1:n}, \mathbf{s}_{n+1})P(f_{n+1}|\mathbf{s}_{n+1})}{P(\mathbf{q}_{1:n}|\mathbf{s}_{1:n})}, \tag{1.8}$$

where $P(f_{n+1}|\mathbf{q}_{1:n}, \mathbf{s}_{1:n}, \mathbf{s}_{n+1})$ is the posterior distribution of f_{n+1}, $P(\mathbf{q}_{n+1}|f_{n+1}, \mathbf{s}_{1:n}, \mathbf{s}_{n+1})$ the likelihood, $P(f_{n+1}|\mathbf{s}_{n+1})$ the prior distribution of f_{n+1} and $P(\mathbf{q}_{1:n}|\mathbf{s}_{1:n})$ the marginal likelihood. In other words, with Eq. 1.8, we want to describe the probability of f_{n+1} once the noisy observations $\mathbf{q}_{1:n}$ have been gathered.

Fortunately, the right hand side of Eq. 1.8 is easy to compute with the multivariate normal distribution (see Appendix A.2 of [96]):

$$\begin{aligned} &P(a, b) = \mathcal{N}\left(\begin{pmatrix}\mu_a \\ \mu_b\end{pmatrix}, \begin{pmatrix}\mathbf{A} & \mathbf{C} \\ \mathbf{C}^\mathsf{T} & \mathbf{B}\end{pmatrix}\right) \\ &\implies P(b|a) = \mathcal{N}(\mu_b + \mathbf{C}^\mathsf{T}\mathbf{A}^{-1}(a - \mu_a), \mathbf{B} - \mathbf{C}^\mathsf{T}\mathbf{A}^{-1}\mathbf{C}). \end{aligned} \tag{1.9}$$

It then naturally follows with Eq. 1.7 and Eq. 1.9, that we can rewrite Eq. 1.8 as a normal distribution:

$$P(f_{n+1}|\mathbf{q}_{1:n}, \mathbf{s}_{1:n}, \mathbf{s}_{n+1}) = \mathcal{N}(\mu_n(\mathbf{s}_{n+1}), \sigma_n^2(\mathbf{s}_{n+1})), \tag{1.10}$$

where

$$\mu_n(\mathbf{s}_{n+1}) = \mu_0(\mathbf{s}_{n+1}) + \mathbf{k}^\mathsf{T}[\mathbf{K} + \sigma_\eta^2\mathbf{I}_n]^{-1}(\mathbf{q}_{1:n} - \boldsymbol{\mu}_0(\mathbf{s}_{1:n})), \tag{1.11}$$

and

$$\sigma_n^2(\mathbf{s}_{n+1}) = k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) - \mathbf{k}^\mathsf{T}[\mathbf{K} + \sigma_\eta^2\mathbf{I}_n]^{-1}\mathbf{k}. \tag{1.12}$$

The mean $\mu_n(\mathbf{s}_{n+1})$ represents the expected value of f_{n+1} given the observations $\mathbf{q}_{1:n}$ whereas $\sigma_n(\mathbf{s}_{n+1})$ represents the standard deviation and is in itself a metric of the uncertainty predicted by the model.

Generally, in BO, an acquisition function is defined in order to determine the next candidate design point. This acquisition function generally trades off exploitation (sampling in promising areas where an optimum could be found) and exploration (evaluate design points where the uncertainty is high). Maximizing the acquisition function is then generally much cheaper than optimizing the true objective function. One of simplest acquisition function is probably the negative lower confidence bound derived from the work of Cox [15]:

$$\text{NLCB}(\mathbf{s}) = -\mu_n(\mathbf{s}) + \kappa\sigma_n(\mathbf{s}), \tag{1.13}$$

that we want to maximize to select the next design point $\mathbf{s}_{n+1}$. Maximizing the term $-\mu_n(\mathbf{s})$ is equivalent to perform exploitation of the model whereas maximizing the term $\sigma_n(\mathbf{s})$ corresponds to exploration. The trade-off between exploitation and exploration is controlled by the parameter κ whose choice is left to the user. This parameter can be fixed or changed dynamically with the number of observations as in [106].

Other popular choices of acquisition functions include the Probability of Improvement [54] and the Expected Improvement [85]. As their names suggest, they respectively quantify the probability of improvement and expected improvement over a minimum that is generally taken as the minimum observed value. Recently, more elaborated acquisition functions have been proposed such as the Knowledge Gradient [24] that quantifies the gain over the current model minimum if one additional observation was possible. Hennig and Schuler [34] developed the Entropy Search algorithm where the mutual information between the minimum design point of the objective function and the objective function evaluation at a design point $\mathbf{s}$ is chosen as an acquisition function. Capitalizing on this idea, Hernández-Lobatto *et al.* [35] developed the Predictive Entropy Search where the acquisition function is the same as the Entropy Search algorithm but computed differently. More precisely, the symmetry properties of the mutual information are used to simplify the numerical methods employed to compute the acquisition function. Finally, Wang and Jegelka [117] developed the Max-Value Entropy Search where the acquisition function is the mutual information between the objective function minimum value and the objective function at a design point. The main benefit of this method is that compared to the Entropy Search and the Predictive Entropy Search, the Max-Value Entropy Search has a closed form and is faster to compute.

Depending on the choice of the acquisition function, different designs will be selected. Still, for all these methods, the process remains the same: the acquisition function is maximized in order to select the next design point to evaluate, the objective function is observed at this design point and the posterior distribution of the GP is then updated. This process is repeated until a stop criterion is met, e.g. a maximum number of iterations, a value below a certain threshold, proximity between consecutive design points, etc.

1.1.3 Bayesian Optimization with gradients and/or multi-fidelity

The performance of BO is therefore dependent on two criteria: the model and the acquisition function chosen. Whereas as mentioned earlier various acquisitions functions were created, Shahriari *et al.* [103] argued that: "There has been a great deal of work that has focused heavily on designing acquisition functions; however, we have taken the perspective that the importance of this plays a secondary role to the choice of the underlying surrogate model". Several authors took the direction of developing the models for GP. Among the different possibilities, one feature that appears particularly suited to expensive numerical simulations, such as the ones that appear in fluid mechanics, is the usage of various sources of information.

For example in CFD, it is possible to obtain the derivative information of the objective function at a reasonable cost with the adjoint method. This gradient information can then be included in the prior distribution and a new posterior distribution can be deduced. The method to do so is detailed in Section 9.4 of [96] and Section 7 of Forrester *et al.* [21]. Lizotte [71] showed that when the derivative information was included, BO outperformed the gradient-based algorithm L-BFGS-B [7]. Later, Wu *et al.* [122] repeated the experiment with the derivative Knowledge Gradient acquisition function. As the Knowledge Gradient mentioned earlier, this acquisition function evaluates the gain over the model optimum according to future

observations. However, in addition to the future objective function observations that use the Knowledge Gradient, the derivative Knowledge Gradient acquisition function also takes into account future derivative observations. Once again, the authors showed that BO with gradients was more efficient than L-BFGS-B. Recently, Talnikar and Wang [111] used a GP with derivative information to perform optimization in Large Eddy Simulation (LES). The gradients are typically noisy in these conditions and can diverge, making difficult the use of gradient-based methods. They compared BO with and without gradients and demonstrated that adding the derivative information improved the efficiency.

Another possibility is to include various fidelity levels of the objective function into the GP and build what is called a multi-fidelity model. For example, we can consider two fidelity levels: one that is defined by a fast realization of the objective function but imprecise (referred to as the low-fidelity model) and another one that is slower to obtain but accurate (referred to as the high-fidelity model). In CFD, low/high-fidelity models can include (but is not restricted to): coarse/fine mesh, Reynolds Averaged Navier Stokes (RANS)/Large Eddy Simulations (LES), loosely/tightly converged objective function. As with the derivative information, the information brought by the low and high-fidelity models are included in the prior distribution through the covariance matrix (and also possibly in the prior mean). The model of reference for the multi-fidelity setting is the model of Kennedy and O'Hagan [48]. The high-fidelity function is then modelled as the sum of a scaled low-fidelity objective function plus an error term known as the bridge function. Both the low-fidelity objective function and bridge functions are considered independent processes. More details on the implementation of these various fidelity models can be found in Section 8 of Forrester *et al.* [21]. On the Hartmann 6 test function, for the same final accuracy, Park *et al.* [88] showed that multi-fidelity models could save up to 86% of the cost compared to the single high-fidelity model. On the same test function and for the same cost, the multi-fidelity model was able to improve the accuracy up to 51% compared to the single high-fidelity model.

Some authors have explored the combination of derivative information and multi-fidelity models. For example, Han *et al.* [31] enhanced a Kriging multi-fidelity model with the gradient information. In this article, they also proposed a new generalised hybrid bridge function for the multi-fidelity model. The high-fidelity function is then modelled as the product between a low order polynomial and the low-fidelity objective function more a bridge function. Compared to the multi-fidelity model and gradient enhanced Kriging, the model combining both muti-fidelity and gradient information was the most accurate, efficient and robust in the context of an aerodynamic application. The same year, Yamazaki and Mavriplis [123] relied on the same idea and proposed a model named derivative-enhanced variable-fidelity surrogate model. This model was investigated for both modelling and optimization. They showed that the multi-fidelity model with gradient information was more accurate than when these models were used separately and lead to fast objective function reduction on an aerodynamics case. Later, Ulaganathan *et al.* [114] also worked with multi-fidelity and gradient information. They introduced the Gradient Enhanced recursive CoKriging model (GECoK), a multi-fidelity model enhanced by the gradient information on both objective functions. This model showed superior modelling accuracy than using multi-fidelity and gradient information separately in the GP model.

1.1.4 Bayesian Optimization in high-dimensional spaces

Still, despite the progress made in the GP modelling, one question remains: how do we apply BO to high dimensional problems ? Indeed, since the performance of BO lays on the accuracy of the surrogate model, its performance decreases as the dimension of the design space increases. As mentioned earlier, a significantly higher number of design points is required in high dimensions to build a reliable surrogate model. Thus, BO remains generally limited to problems of moderate dimension (up to 10 design parameters). As mentioned in Lam [57], two strategies can be used to tackle high dimensional problems with BO: collect more information per iteration in order to build a more accurate model, or reduce the dimension of the optimization problem. The first strategy could be applied with the GP with gradients or with the multi-fidelity model with gradients mentioned earlier in this introduction. However, the GP scales cubically with the number of observations. Thus, for a design space of dimension N, the cost of building a single-fidelity GP with gradient information would be $(N+1)^3$ times the cost of fitting a single-fidelity GP without derivative information. The cost of building a multi-fidelity model with gradient information would even be higher since low-fidelity samples are added. This cost rapidly makes the usage of models with gradient information impractical in the context of high dimensional BO or would require further investigation in order to reduce dramatically this cost.

We are then left with the other strategy: reducing the dimension of the design space. In BO, various methods have already been employed to decrease the dimension of a high dimensional space. To tackle this problem, Chen *et al.* [12] developed a two-stage algorithm. In the first phase, Hierchical Diagonal Sampling (HDS) is applied in order to determine the active variables. Then, BO is applied on this set of active variables. Hutter *et al.* [38] used Random forests instead of a GP as a surrogate model since random forests can naturally determine the most active variables [103]. Wang *et al.* [116] create the Random EMbedding Bayesian Optimization (REMBO). Firstly, the design space is projected into a lower dimensional space through a random generated matrix. BO is then applied into this lower dimensional space. In the article, this method was successfully employed to optimize a two-dimensional function embedded in a one billion dimensional space. Constantine [13] and his team developed the Active Subspaces (AS) algorithm. With an initial set of sampling points, a linear combination of the variables is determined in order to reduce the dimension. Then BO can be applied in the subdimensional space. Whereas this method normally requires the gradient information, Constantine [13] also proposed two other methods to build this linear combination if the gradients are not available: approximating the gradient information or trying to fit a global linear model. Lam [57] further developed the AS in order to estimate the lower dimensional space with multi-fidelity samples. Results showed that it was possible to estimate the AS with less computational cost using multi-fidelity samples. Finally, Kusner *et al.* [55] used a variational autoencoder to project the initial design space into a lower dimensional space. A Gaussian Process latent variable model [62] was employed in order to consider the uncertainty of the input.

1.2 Goals of the thesis

In this thesis, three main objectives were pursued:

1. Evaluate the performance of Bayesian Optimization in problems involving numerical flow simulations.

2. Compare the different Gaussian Process models with derivative information and/or multi-fidelity for modelling and Bayesian Optimization.

3. Develop a framework to tackle multi-objective Bayesian Optimization in high-dimensional spaces.

The motivation and scope of these goals are discussed in the remaining of this section.

Evaluate the performance of Bayesian Optimization in problems involving numerical flow simulations

This thesis focuses on the efficiency of BO and its variants for aerodynamics improvements in flows around cylinders and airfoils. More specifically, BO is applied on numerical simulations of these two cases in order to reduce the drag or increase the lift through tangential wall actuation.

Improving the aerodynamics performance through numerical simulations and optimization algorithms have been widely investigated in the literature. For example, Li *et al.* [65] used a quasi-Newton method to suppress the vortex shedding of a cylinder up to a Reynolds number $\mathrm{Re} = 110$ through a blowing-suction mechanism. Meliga *et al.* [81] addressed the drag reduction on two and three-dimensional cylinders up to $\mathrm{Re} = 3900$ through wall-transpiration actuation. The RANS equations were used to model turbulence. Mao *et al.* [77] also investigated the drag reduction problem of a two- and three-dimensional cylinder up to $\mathrm{Re} = 1000$ using Direct Numerical Simulation (DNS). They considered a surface-normal wall transpiration. Later, the same problem was investigated with a tangential motion of the surface substituting the transpiration mechanism in [78]. In both studies, the adjoint method was used to compute gradients.

Derivative-free optimization methods were not forgotten either for aerodynamics optimization. Li *et al.* [64] added an additional step to the Nelder-Mead algorithm by performing at each iteration an explorative step through a Latin Hypercube sampling. They applied this algorithm named Explorative Gradient Method (EGM) to the drag reduction of the fluidic pinball and the Ahmed body. Milano and Koumoutsakos [83] used evolutionary algorithms to tackle the drag reduction of a two-dimensional cylinder at $\mathrm{Re} = 500$ where the design parameters were the tangential velocity amplitude of actuators set around the cylinder. Sengupta *et al.* [102] used genetic algorithms to reduce the drag of a rotating two-dimensional cylinder at $\mathrm{Re} = 15000$. The genetic algorithms were also used to improve the aerodynamic performance of turbine blades in [82]. Duan *et al.* [124] preferred to use Particle Swarm Optimization (PSO) to improve the peak efficiency of the Rotor 37. PSO was also employed in [100] with a metamodelling methodology for the optimization

of the airfoil shape of a compressor. Both PSO and genetic algorithms (and 3 additional optimization algorithms) were also used in [1] to optimize the design of an axial pump.

The RSM has also been widely applied to aerodynamics cases. Catalano *et al.* [10] used a surrogate model to minimize the drag of a two-dimensional cylinder at $\mathrm{Re} = 500$ and $\mathrm{Re} = 3900$ with a synthetic jet. Jeong *et al.* [41] relied on the Kriging model with the Expected Improvement acquisition function to maximize the lift-to-drag ratio of a two-dimensional airfoil with the Spalart-Allmaras turbulence model. More recently, Duvigneau and Chandrashekar [17] also used a Kriging model to minimize the drag around an oscillating cylinder at $\mathrm{Re} = 200$. They also applied this method to reduce the intensity of a shock wave for a transonic airfoil using RANS models. Later Talnikar *et al.* [110] used a parallel BO approach to minimize the drag in a turbulent channel with LES. They also applied their method to optimize the heat transfer and pressure coefficient of a turbine blade. Later, Mahforze *et al.* [75] also used the BO to minimize the skin-friction drag of spatially evolving turbulent boundary layers simulated through DNS.

However, despite the important number of publications related to aerodynamics optimization, application of BO in numerical flow simulations are uncommon and the performance of this optimization algorithm needs to be further investigated. Moreover, the literature would benefit from a comparison of derivative-free optimization algorithms in the context of numerical simulations in fluid mechanics. A comparison between some of them was already done in [1] but did not include BO and some recent algorithms such as CMA-ES [33] and explorative gradient method [64].

Compare the different Gaussian Process models with derivative information and/or multi-fidelity for modelling and Bayesian Optimization

As mentioned earlier, the GP has been developed to include lower-fidelity models and/or derivative information. All these approaches showed significant improvement over the classical GP. Still, in front of all the possible models and depending on the computational cost associated with each source of information, the user may benefit from guidelines on which model to choose for either modelling or optimization purposes. Indeed, the cost of the source of information has often been neglected in the studies. For example, Yamazaki *et al.* [123] compared the models neglecting the costs of the gradient information and low-fidelity levels when they examined the modelling performances. When they applied optimization, the gradients were not included in the DOE, but the single-fidelity and multi-fidelity models were not initialized with the same cost. Han *et al.* [31] also neglected the costs of the lower fidelity model and gradient information in their studies. Finally, Ulaganathan *et al.* [114] studied the performance of the multi-fidelity gradient enhanced Kriging as a function of the number of high-fidelity sample points and the behaviour of the model depending on which fidelity the gradients were included. Some results on the number of low- and high-fidelity samples required to achieve the same level of accuracy between the different were also given. Still, despite being included in this paper, the notion of information cost and effects of the gradients on one fidelity level for the multi-fidelity setting were not assessed in detail. Also, the performance study of the models for varying number of high-fidelity samples was presented with

a constant number of low-fidelity sample points and not for various configurations of the DOE and a fixed budget. Even if these articles provide useful information regarding the GP models with multi-fidelity and/or derivative information, the BO user would also benefit from a performance study of the GP models that takes into account the cost of all sources of information and the configuration of the DOE to make an adequate choice of the model and DOE for global modelling or optimization of an objective function. The cost of information becomes even more a critical factor when the user can only perform modelling or BO with a limited budget (under 10 times the dimension of the design space for example).

Develop a framework to tackle multi-objective Bayesian Optimization in high-dimensional spaces

Finally, as argued before, the gradients can not be readily used to build a reliable GP model in high dimensions since its cost dramatically increases. Thus, the other approach left is to reduce the dimension of the design space. This approach has been notably used in BO as described in 1.1.4. However, this approach has rarely been used in the context of multi-objective Bayesian Optimization (MOBO). In contrast with BO in the single-objective case, MOBO optimizes several objective functions at the same time. This process requires more samples than BO, as the goal is not to find only one optimal design point but several ones that will compose what is called a Pareto front. A Pareto front consists of optimal design points where an objective function can not be improved without making another one worse. MOBO could really benefit from dimension reduction in order to decrease the optimization cost.

Ling *et al.* [66] designed a MOBO algorithm for high dimensional space and applied it to solve an engineering problem composed of 37 design parameters. But no dimension reduction method was employed. Lukaczyk *et al.* [73] used AS to project a design space of dimension 50 into a design space of dimension 2. They then applied BO in this lower dimensional space to optimize the shape of the ONERA-M6 transonic wing in order to reduce the drag subject to a lift constraint. A link between the lower dimensional space of the drag and the lower dimensional space of the lift was made but no proper MOBO was performed. Later, Grey and Constantine [30] applied the AS to reduce the dimension of two airfoil shape parametrizations. Both lift and drag coefficients were reduced from a maximum of 11 design parameters to a two-dimensional space. Through visualisations, they were able to find the Pareto front of the drag and lift coefficients. Still, to find the Pareto front, no MOBO algorithm was applied. Thus, an algorithm combining dimension reduction and MOBO in order to automatize the optimization process could be useful to the scientific community.

1.3 Outline of the thesis

This thesis is divided in five chapters. Each objective is investigated in a different chapter. Thus, in Chapter 2, the BO performance is assessed on a canonical fluid case. In Chapter 3, the possible GP models with gradient information and/or multi-fidelity are studied for global modelling and optimization purposes. In Chapter 4, a method combining MOBO and dimension reduction is developed and tested on

a benchmark objective function and two aerodynamics applications. Finally, in Chapter 5, conclusions are summarized and future lines of research are suggested.

Chapters 2, 3 and 4 are self-contained. Chapter 2 has been published as a journal paper [61] whereas Chapter 3 and Chapter 4 are begin prepared for submission. Below a summary of each chapter is provided.

Chapter 2

In Chapter 2, the performance of BO on numerical flow simulations is investigated. Firstly, BO is presented, including details about the kernel functions and some of the most used acquisition functions. Then, BO is applied to the drag reduction of a two-dimensional cylinder at Re = 500. Several points, called actuators, are placed around the cylinder, each actuator being able to set a tangential velocity. These tangential velocities are the design parameters. The objective function is the root mean square of the sum between the drag coefficient and a penalty term proportional to the kinetic energy set by the actuators:

$$f(\mathbf{s}) = \sqrt{\frac{1}{\Delta T}\int_{T}^{T+\Delta T}\left(C_{\mathrm{d}}^{2}(t;\mathbf{s}) + \frac{\alpha}{N}\mathbf{s}^{\mathsf{T}}\mathbf{s}\right)\,\mathrm{d}t}, \tag{1.14}$$

where f is the objective function, $\mathbf{s}$ the design parameters, T the starting time of the computation of f, ΔT the time period considered for the integration, C_{d} the drag coefficient, t the time of the simulation, α a scaling constant for the penalty term and N the number of actuators set.

The resulting flow fields found with BO for various α values are described. The influence of the BO parameters such as the kernel function, the dimension of the design space, the acquisition function and the size of the DOE is then investigated. On the same case, the performance of serial and parallel BO is assessed and compared against other derivative-free optimization algorithms such as the Nelder-Mead algorithm, the Explorative Gradient Method (EGM), the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and the Particle Swarm Optimization (PSO). Finally, BO is applied to the drag reduction of a three-dimensional cylinder at Re = 3900. The same objective function and design parametrization as for the two-dimensional cylinder at Re = 500 are chosen. The optimal solution found with BO for $\alpha = 8$ is compared with the uncontrolled case and the resulting behaviour is examined.

Chapter 3

In Chapter 3, the performance of the GP models with gradients and/or multi-fidelity is explored for global modelling and BO. In the first place, the different GP models and the posterior distributions are developed. These models are then tested on the two-dimensional Styblinski-Tang objective function, the six-dimensional Hartmann-6 objective function, and on the drag reduction of a two-dimensional cylinder at Re = 200. For the last case, a tangential velocity is set around the cylinder through 3 design parameters. As in Chapter 2, an objective function consisting of the sum of the root mean square drag coefficient and a penalty term proportional to the kinetic energy set by the tangential wall motion is chosen.

For the three test cases described, the global accuracy and the minimum prediction of the models are firstly investigated with the DOE. Various gradient costs, ratio between the low and high-fidelity samples for the multi-fidelity models, and various total budgets are studied. Then, for each case and for a given initial budget, the configuration with the lowest initial minimum prediction is chosen as the starting point of the BO. The minimum obtained at each objective function observation is compared between the different models for different gradient costs. Finally, the main observations on the models and the advantages and drawbacks of each one are highlighted in the conclusions of Chapter 3.

Chapter 4

In Chapter 4, a framework combining MOBO and dimension reduction is developed and presented. Definitions of the multi-objective optimization and the hypervolume metric as well as the MOBO and some dimension reduction methods are introduced first. The framework combining MOBO and dimension reduction is then presented and showcased on three different problems: the Fonseca-Fleming problem, the two-dimensional cylinder at Re = 40 and the two-dimensional NACA0012 profile at Re = 1000 with an angle of attack $AoA = 10°$. For each case, the Pareto front obtained as well as the hypervolume metric are investigated and compared for the MOBO algorithm with and without dimension reduction.

For the Fonseca-Fleming problem, a quadratic dimension reduction is used for both objective functions. The MOBO algorithms with and without dimension reduction are tested with $N = 2, 5, 10, 20, 50, 100$ design parameters.

For the two-dimensional cylinder at Re = 40, 79 actuators are placed around the cylinder, each actuator being able to set a tangential velocity. These tangential velocities are the design parameters. The two objective functions to minimize are the squared drag coefficient and an objective function proportional to the cost of the actuation. The squared drag coefficient is reduced to one dimension with the AS method whereas a straightforward dimension reduction is used for the second objective function. In addition to the performances study of the optimization algorithms, the optimal velocity profiles around the cylinder and the fluid results are also presented.

Finally, for the two-dimensional NACA0012 profile at Re = 1000, N tangential actuators are set around the profile as with the cylinder case. The two objective functions to minimize are the time-averaged drag coefficient and the negative lift objective function. Both objective functions are penalized by a term proportional to the kinetic energy set by the actuators. The dimension reduction on both resulting objective functions is performed through a quadratic dimension reduction method. The performances of the MOBO algorithm with dimension reduction and the MOBO algorithm without dimension reduction are investigated with $N = 10, 20, 40, 80$. The optimal velocity profiles for each case are presented and the fluid solutions of the optimal solutions for $N = 80$ are also introduced.

Chapter 5

Finally, in Chapter 5, the main results and observations are summarized. Future lines of research related with the work performed are also suggested.

Chapter 2

Cylinder drag minimization through wall actuation: a Bayesian optimization approach

2.1 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 [11] 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.

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 [40] 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 [27] and Marquet *et al.* [80] 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 [8] 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 [111], 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 [86] 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 [64]. Algorithms inspired by the observation of nature, such as natural selection [19] or bird flocking [47], 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 [1, 83, 102, 82] for genetic algorithms and [1, 124, 100] for Particle Swarm Optimization.

Typically applied within a derivative-free framework, the Response Surface Methodology (RSM) [94] 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 surrogate 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) [6, 103] 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, and (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 [54], 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.* [85] extended the BO approach to multidimensional problems using an alternative acquisition function named Expected Improvement (EI). In 1951, Krige [52] 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.* [99] 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 [43] 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 [2] or to address stochastic objective functions [37]. A review of recent developments of this method can be found in [103] and in [25].

Regarding applications to numerical flow simulations, Jeong *et al.* [41] 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 [17] 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.* [110] 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.* [75] applied BO to reduce the skin-friction drag of spatially evolving turbulent boundary layers simulated through DNS. Lam *et al.* [59] 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 [95], 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.* [65] 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 [83] used evolutionary algorithms to determine the optimal amplitude of 16 actuators for drag minimization of a two-dimensional circular cylinder at $\mathrm{Re} = 500$. Catalano *et al.* [10] minimized the drag coefficient of a two dimensional cylinder at $\mathrm{Re} = 500$ and $\mathrm{Re} = 3900$ 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.* [81] also investigated the ability of the adjoint equations to reduce the drag in a two- and three-dimensional cylinder at $\mathrm{Re} = 100$ and $\mathrm{Re} = 3900$ using RANS models. Mao *et al.* investigated using DNS the physical mechanisms leading to two- and three-dimensional cylinder drag reduction at $\mathrm{Re} \leq 1000$ through a surface-normal wall transpiration [77], and a tangential motion of the surface [78]. The sensitivity information computed through the adjoint equations was used to find the optimal configuration of the system.

The goal of this chapter 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 chapter is organized as follows: in Section 2.2, we give an overview of the optimization framework for CFD applications. In Section 2.3, we present the two-dimensional case at $\mathrm{Re} = 500$ 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 2.4, the results of the three-dimensional case at Re = 3900 are presented and discussed. Finally, we conclude in Section 2.5 discussing the main results and suggestions for future work.

2.2 Optimization framework

2.2.1 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

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = 0 \tag{2.1}$$

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

$$\mathbf{s}^* = \arg\min_{\mathbf{s} \in \mathcal{S}} f(\mathbf{s}). \tag{2.2}$$

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.

2.2.2 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

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
```

$\mathcal{D}_{1:n} = \{\mathbf{s}_{1:n}, \mathbf{q}_{1:n}\}$ 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 modelling purposes, these techniques are often superior to alternatives (see [11]).

Regarding the number of initial points that is required, Jones *et al.* [43] 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.* [21] 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 2.2.5), Sóboster *et al.* [105] 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 [11] and to Part 1 and Section 3.3.2 of Forrester *et al.* [21].

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

The numerical simulations presented in Section 2.3 and Section 2.4 are carried out using the open-source numerical flow solver PyFR [118]. This code is written in Python and solves advection–diffusion equations such as Eq. (2.1) on streaming architectures using the flux reconstruction methodology from [39].

We can write the compressible Navier–Stokes equations in the form of Eq. (2.1), 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

$$\mathbf{F} = \begin{pmatrix} \rho v_x & \rho v_y & \rho v_z \\ \rho v_x^2 + p - \tau_{xx} & \rho v_y v_x - \tau_{yx} & \rho v_z v_x - \tau_{zx} \\ \rho v_x v_y - \tau_{xy} & \rho v_y^2 + p - \tau_{yy} & \rho v_z v_y - \tau_{zy} \\ \rho v_x v_z - \tau_{xz} & \rho v_y v_z - \tau_{yz} & \rho v_z^2 + p - \tau_{zz} \\ v_x(\epsilon + p) - v_i \tau_{ix} - \Delta \partial_x T & v_y(\epsilon + p) - v_i \tau_{iy} - \Delta \partial_y T & v_z(\epsilon + p) - v_i \tau_{iz} - \Delta \partial_z T \end{pmatrix}, \tag{2.3}$$

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

$$p = \rho \frac{\gamma - 1}{\gamma} c_\mathrm{p} T, \tag{2.4}$$

where γ is the adiabatic index, and the energy equation

$$\epsilon = \frac{1}{2}\rho \|\mathbf{v}\|^2 + \frac{p}{\gamma - 1}. \tag{2.5}$$

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

$$P(\hat{q}|\mathcal{D}_{1:n}) \propto P(\mathcal{D}_{1:n}|\hat{q})P(\hat{q}), \tag{2.6}$$

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.

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 [96], and a tutorial can be found in Schulz *et al.* [101]. 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

$$f(\mathbf{s}) \sim \mathrm{GP}(\mu_0(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')). \tag{2.7}$$

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

$$k_{\mathrm{RBF}}(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2}\right), \tag{2.8}$$

$$k_{5/2}(r) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5r^2}{3}\right) \exp\left(-\sqrt{5}r\right), \tag{2.9}$$

and

$$k_{3/2}(r) = \sigma_f^2 \left(1 + \sqrt{3}r\right) \exp\left(-\sqrt{3}r\right), \tag{2.10}$$

respectively, where σ_f^2 is the variance, $r^2 = (\mathbf{s} - \mathbf{s}')^{\mathsf{T}} \mathbf{\Lambda} (\mathbf{s} - \mathbf{s}')$ 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

$$q_i = f(\mathbf{s}_i) + \eta_i, \tag{2.11}$$

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

$$\begin{bmatrix} \mathbf{q}_{1:n} \\ f_{n+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \mathbf{K} + \sigma_\eta^2 \mathbf{I}_n & \mathbf{k} \\ \mathbf{k}^{\mathsf{T}} & k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) \end{pmatrix}\right), \tag{2.12}$$

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

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:

$$P(f_{n+1}|\mathcal{D}_{1:n}, \mathbf{s}_{n+1}) = \mathcal{N}(\mu_n(\mathbf{s}_{n+1}), \sigma_n^2(\mathbf{s}_{n+1})), \tag{2.13}$$

where

$$\mu_n(\mathbf{s}_{n+1}) = \mathbf{k}^\mathsf{T}[\mathbf{K} + \sigma_\eta^2 \mathbf{I}_n]^{-1}\mathbf{q}_{1:n}, \tag{2.14}$$

and

$$\sigma_n^2(\mathbf{s}_{n+1}) = k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) - \mathbf{k}^\mathsf{T}[\mathbf{K} + \sigma_\eta^2 \mathbf{I}_n]^{-1}\mathbf{k}, \tag{2.15}$$

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

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

$$\log P(\mathbf{q}_{1:n}|\mathbf{s}_{1:n}, \boldsymbol{\psi}) = -\frac{1}{2}\mathbf{q}_{1:n}^\mathsf{T}(\mathbf{K} + \sigma_\eta^2 I)^{-1}\mathbf{q}_{1:n} - \frac{1}{2}\log|\mathbf{K} + \sigma_\eta^2 I| - \frac{n}{2}\log(2\pi). \tag{2.16}$$

Eq. (2.16) is generally maximized using gradient-based optimization methods.

2.2.5 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 [15] given by

$$\text{NLCB}(\mathbf{s}) = -\mu_n(\mathbf{s}) + \kappa\sigma_n(\mathbf{s}), \tag{2.17}$$

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

Kushner [54] used the Probability of Improvement (PI), also known as Maximum Probability of Improvement (MPI), as an acquisition function which is defined as

$$\text{PI}(\mathbf{s}) = \Phi\left(\frac{f(\mathbf{s}_n^*) - \mu_n(\mathbf{s}) - \kappa}{\sigma_n(\mathbf{s})}\right), \tag{2.18}$$

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.* [85] defined the Expected Improvement (EI) as

$$\mathrm{EI}(\mathbf{s}) = \begin{cases} (f(\mathbf{s}_n^*) - \mu_n(\mathbf{s}) - \kappa)\Phi\left(\frac{f(\mathbf{s}_n^*)-\mu_n(\mathbf{s})-\kappa}{\sigma_n(\mathbf{s})}\right) & \\ \qquad\qquad +\sigma_n(\mathbf{s})\phi\left(\frac{f(\mathbf{s}_n^*)-\mu_n(\mathbf{s})-\kappa}{\sigma_n(\mathbf{s})}\right) & \text{if} \quad \sigma_n > 0, \\ 0 & \text{if} \quad \sigma_n = 0, \end{cases} \tag{2.19}$$

where ϕ is the Gaussian probability density function.

The use of PI is known to result in an aggressive exploitation [44]. On the other hand, EI and NLCB are more explorative. EI remains the most used acquisition function. In Section 2.3.4, we will illustrate the behaviour of these acquisition functions through an application to a numerical flow simulation.

2.2.6 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 [29]. 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

$$\mathbf{s}_{i,k} = \arg\max_{\mathbf{s}\in\mathcal{S}} \left\{ g(a(\mathbf{s}, \mathcal{D}_{1:i-1})) \prod_{j=0}^{k-1} \varphi(\mathbf{s}, \mathbf{s}_{i,j}) \right\}, \tag{2.20}$$

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

$$\varphi(\mathbf{s}, \mathbf{s}_{i,j}) = \frac{1}{2}\mathrm{erfc}\left(-\frac{1}{\sqrt{2\sigma_i^2(\mathbf{s}_{i,j})}} \left(L\|\mathbf{s}_{i,j} - \mathbf{s}\| + M - \mu_i(\mathbf{s}_{i,j})\right) \right), \tag{2.21}$$

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

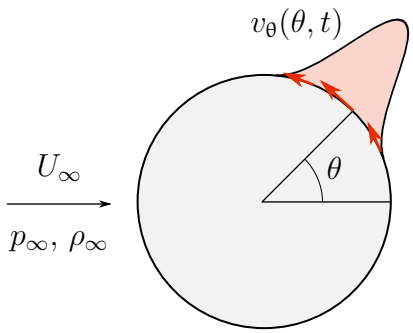


Figure 2.1: Uniform flow around a cylinder with tangential flow actuation at the surface.

$\hat{L}_{GP-LCA} = \max \|\mu_\nabla(\mathbf{s})\|$ (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.

2.2.7 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 $\mathrm{Re} = 40$ (based on the diameter D). For now, we omit the numerical details of the simulation as they will be discussed below in Section 2.3. 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.

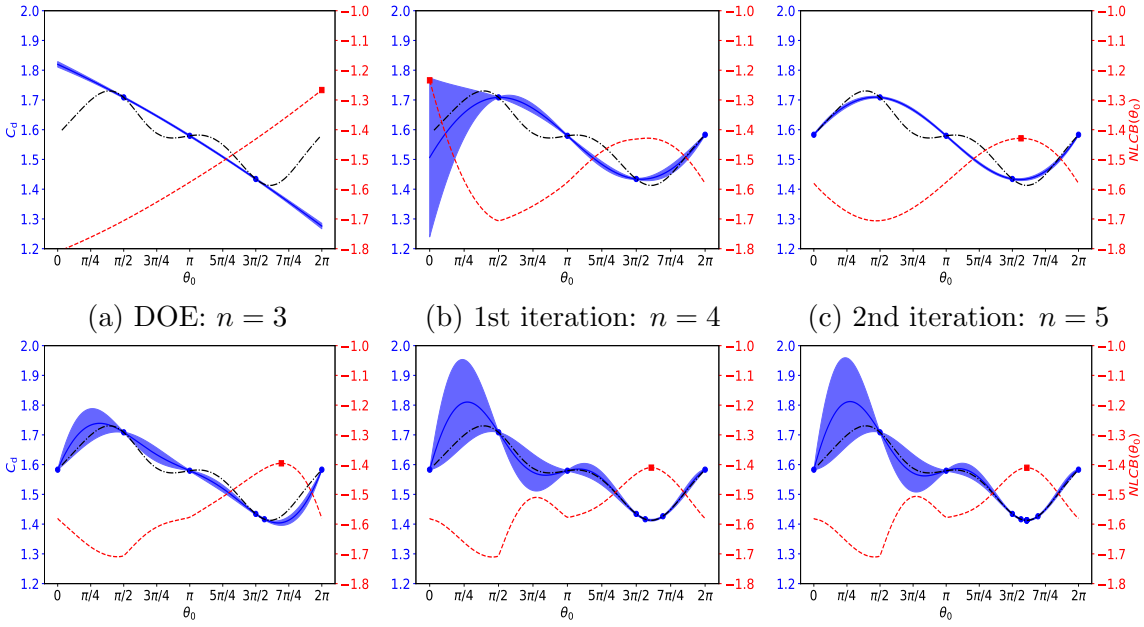
To reduce the drag coefficient, a tangential velocity profile is introduced at the cylinder surface as illustrated in Fig. 2.1. In this example, the tangential velocity profile is given by

$$v_\theta(\theta; \theta_0, \sigma) = \sum_{k=-\infty}^{+\infty} \exp\left(\frac{-(\theta - \theta_0 + 2\pi k)^2}{2\sigma^2}\right), \tag{2.22}$$

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. 2.22, 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. (2.17) with $\kappa = 2$ is chosen. Throughout this chapter, optimization problems are solved using the open-source software suite GPyOpt [112].

The first 10 iterations of the algorithm are illustrated in Fig. 2.2. 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. 2.2a). 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



(a) 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 2.2: 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

.

and a high uncertainty is observed at $\theta_0 = 0$ (Fig. 2.2b). 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. 2.2c). 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. 2.2d). At the fourth iteration, the algorithm will evaluate f at $\theta_0 \approx 5.06$ due to the low mean (Fig. 2.2e). Finally, the algorithm keeps evaluating points that are increasingly closer to $\theta_0 \approx 5.06$ (Fig. 2.2f).

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

2.3.1 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. 2.1. 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

$$v_\theta(\theta, t) = \sum_{j=0}^{\hat{N}-1} v_{\theta,j} \Lambda\left(\frac{\theta - \theta_j}{\Delta\theta}\right) \tanh t, \qquad \text{with } \Lambda(x) = \max\left(1 - |x|, 0\right). \tag{2.23}$$

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 $T \leq t \leq T + \Delta T$ (the initial transients for $0 \leq t \leq T$ are discarded), and a penalization term that mimics the energetic cost of such actuation. More precisely, we have

$$f(\mathbf{s}) = \sqrt{\frac{1}{\Delta T} \int_T^{T+\Delta T} \left(C_\mathrm{d}^2(t; \mathbf{s}) + \frac{\alpha}{N} \mathbf{s}^\mathsf{T} \mathbf{s} \right) \mathrm{d}t}, \tag{2.24}$$

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. (2.2), which is solved using the framework presented in Section 2.2.

In the remaining of this section, a two-dimensional case at Reynolds number $\mathrm{Re} = 500$ and Mach number $\mathrm{Ma} = 0.2$ 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 2.4.

2.3.2 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 A.1. 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. (2.23) 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 [36].

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. (2.24), 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.

2.3.3 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. 2.3a. 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

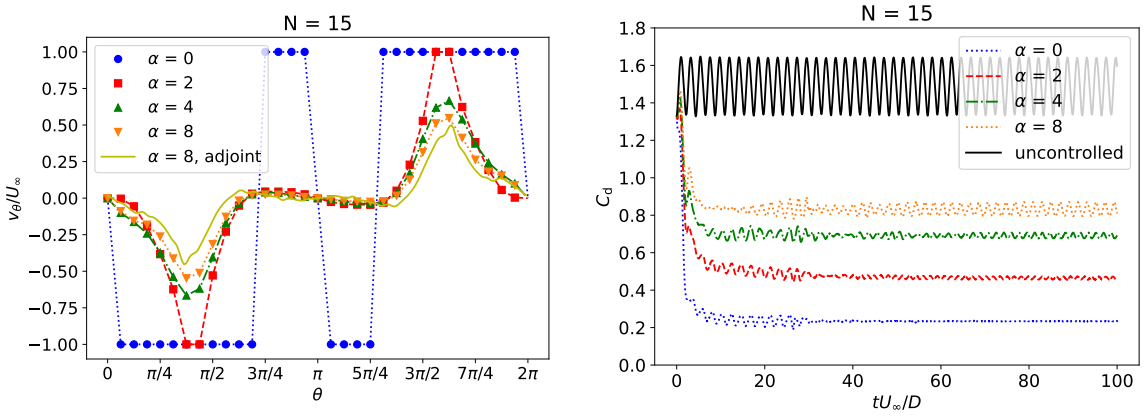


Figure 2.3: Optimal solution for 32 actuators ($N = 15$) and varying α. a) Optimal tangential velocity profiles as a function of θ. The markers indicate the location of the actuators. b) Temporal evolution of the drag coefficient for long integration times.

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 [22]. 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.* [78] at Reynolds number $\mathrm{Re} = 100$. 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. 2.3b.

To assess the changes that the actuators introduce in the flow, we now present in Fig. 2.4 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. 2.4c, 2.4e, 2.4g, 2.4i) 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. 2.3a). The average streamwise velocity component $\overline{v}_x/U_\infty$ from $tU_\infty/D = 50$ to $tU_\infty/D = 100$, depicted in Fig. 2.4 (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. 2.4a, 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 2.1. Starting from the uncontrolled case, the average drag coefficient and the RMS of the lift coefficient decrease as the pe-

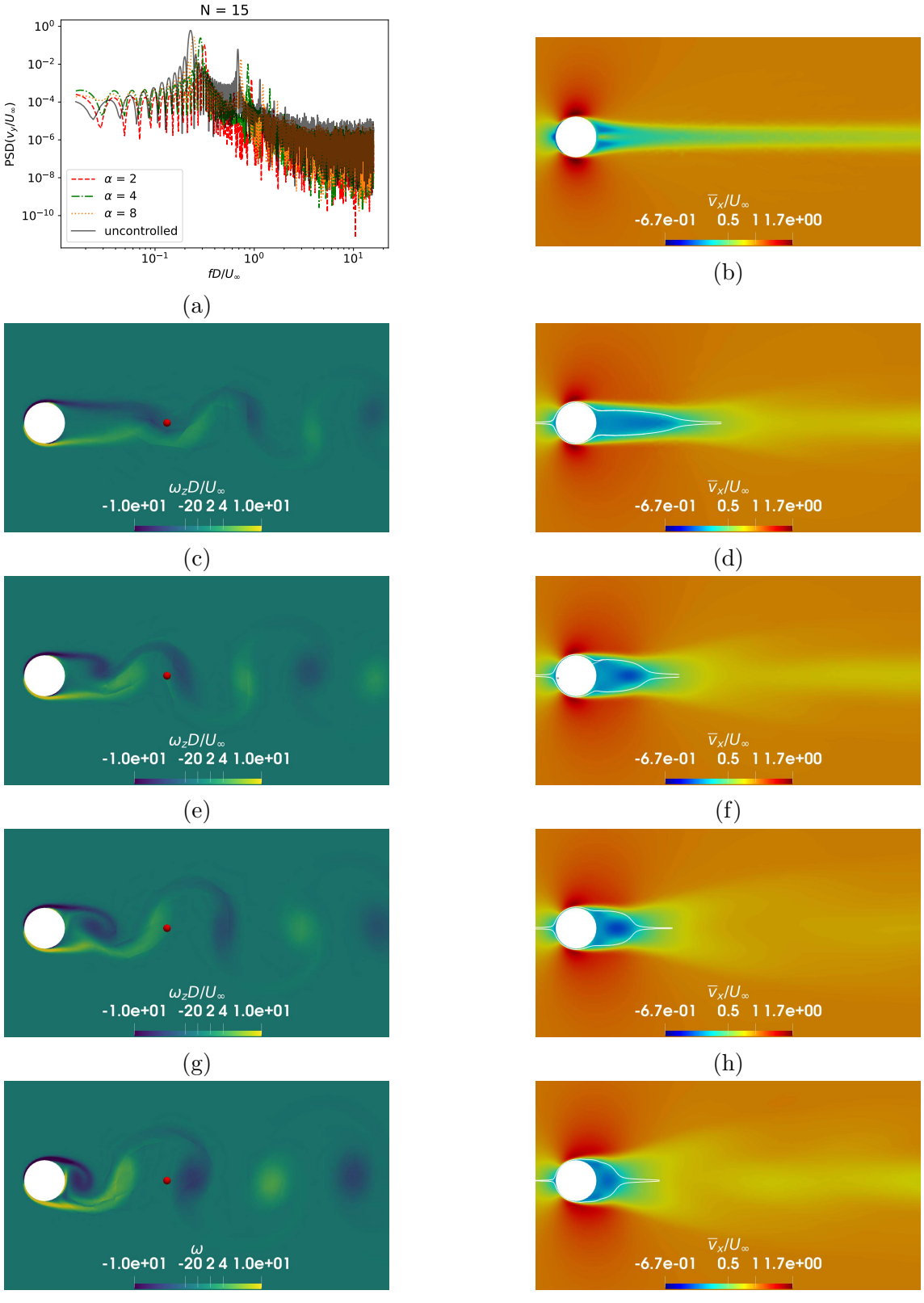


Figure 2.4: Flow-field characteristics of the optimum cases for $N = 15$ and the uncontrolled case. (a) Power spectral densities of the vertical velocity component v_y/U_∞ recorded at $x/D = 3$ and $y/D = 0$ (red dot in (c), (e), (g) and (i)). Left column: instantaneous vorticity component $\omega_z D/U_\infty$ 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.

Case	$f(\mathbf{s})$	$\overline{C_\mathrm{d}}$	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

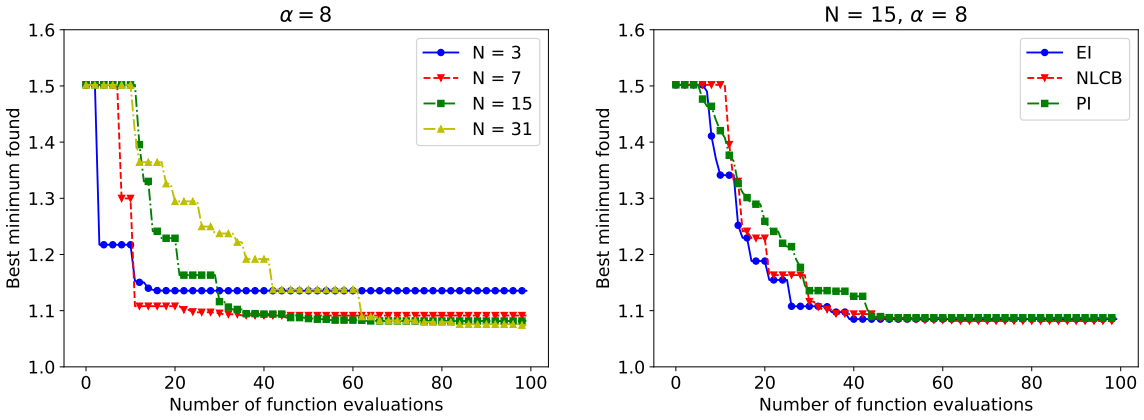
Table 2.1: Objective function $f(\mathbf{s})$, mean drag coefficient $\overline{C_\mathrm{d}}$, RMS of the lift coefficient C_l', distance between the aft of the cylinder and the beginning of the recirculation zone a/D, length of the recirculation zone calculated between its two horizontal extremities 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 $y = 0$

nalization 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.

2.3.4 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 (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. 2.5a, 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



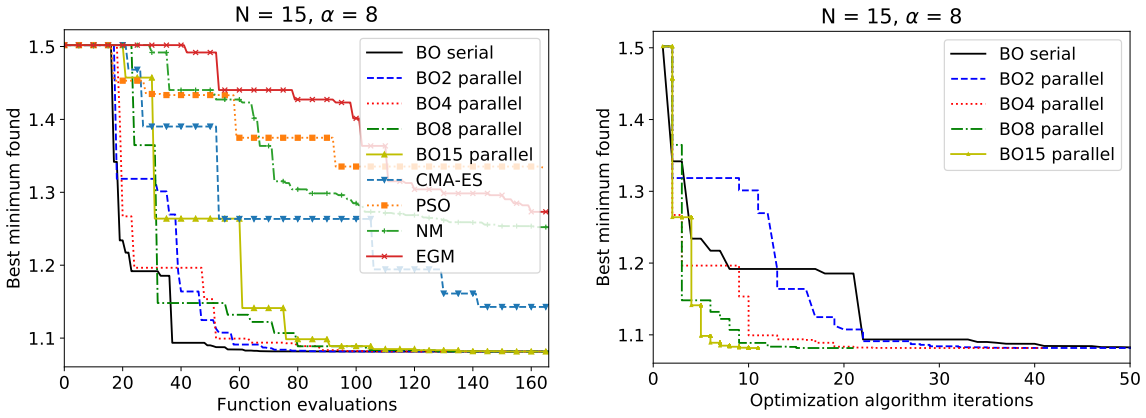
(a) (b)

Figure 2.5: 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 $\alpha = 8$.

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. 2.5b). 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 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. 2.3a). This location corresponds to regions in the domain with high levels of uncertainty, which translates into large values of the acquisition function.



(a) (b)

Figure 2.6: (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 $\alpha = 8$.

2.3.5 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) [33], Particle Swarm Optimization (PSO) [47], Nelder–Mead [86, 26] and Explorative Gradient Method (EGM)[64]. 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

$$\mathbf{s}_i^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(0, \mathbf{C}^{(g)}), \quad \text{for} \quad i = 1, \dots, n, \tag{2.25}$$

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 $\sigma^{(0)} = 0.5U_\infty$, 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:

$$\mathbf{v}_{i+1} = w_i \mathbf{v}_i + c_1 r_1 (\mathbf{s}_p - \mathbf{s}_i) + c_2 r_2 (\mathbf{s}_g - \mathbf{s}_i), \tag{2.26}$$

$$\mathbf{s}_{i+1} = \mathbf{s}_i + \mathbf{v}_{i+1}, \tag{2.27}$$

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 [124] 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:

$$\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i. \tag{2.28}$$

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.* [64] 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

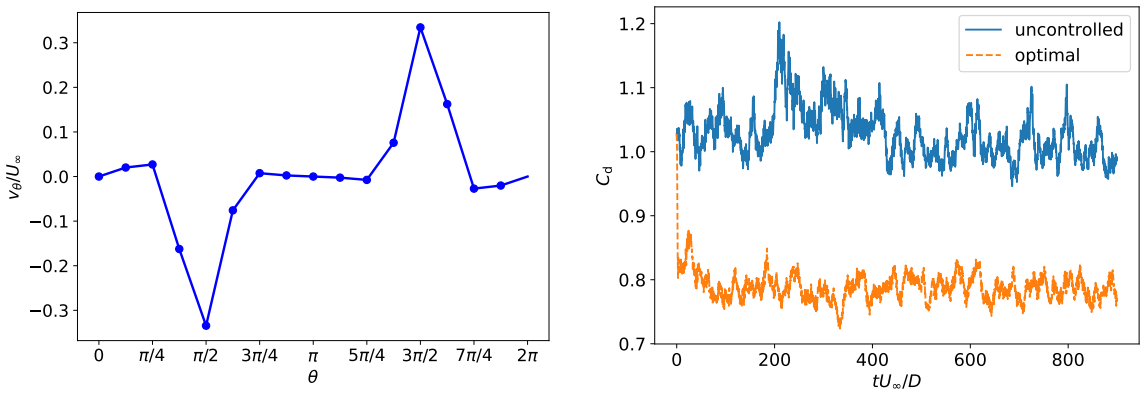
$$\mathbf{s}_{\text{LHS}} = \arg\max \min_{i=1,2,...n} \|\mathbf{s} - \mathbf{s}_i\|, \tag{2.29}$$

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. 2.6.

In addition to the serial case, BO with parallel evaluations was performed according to the procedure sketched in Section 2.2.6. 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. 2.6a. 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



(a) (b)

Figure 2.7: 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.

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.

In Fig. 2.6b 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.

2.4 Drag reduction in the three-dimensional flow around a cylinder

2.4.1 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 2.3.1; the penalty term constant is $\alpha = 8$. The optimization problem is again solved following the approach outlined in Section 2.2 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.* [115] and is provided therein as supplementary material. This case was also investigated by Lehmkuhl *et al.* [63], although with a different numerical solver. We use the numerical grid provided in Vermeire *et al.* [115]. Details of the simulation and comparison with the literature are presented in A.2.

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. (2.24)

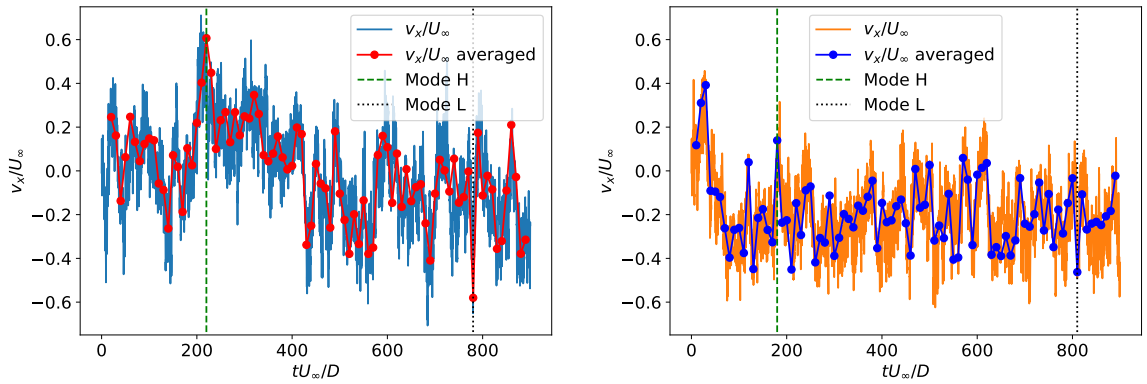


Figure 2.8: Temporal evolution of the streamwise velocity component averaged in the streamwise direction at (a) P_3 for the uncontrolled case and (b) for the optimal case. The dots represent the time average on a $10D/U_\infty$ time-unit window. The presence of modes L and H are represented by vertical dashed and dotted lines, respectively.

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).

2.4.2 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 (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. 2.7a, 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. 2.7b.

In Fig. 2.8, 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.* [63] 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. 2.8a), 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 [63]. In the optimum case (Fig. 2.8b), 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. A.1 and Fig. A.2. 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.* [115]. As discussed in Parnaudeau *et al.* [89], 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.* [89] and lay between the profiles from modes H and L computed by Witherden *et al.* [119]. Finally, some differences are observed between the computed modes and the ones reported in Witherden *et al.* [119]. 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.* [63], 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 [72]. The power spectra of the cross-flow component at P_1 is depicted in Fig. 2.9a.

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. 2.9b. 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. 2.9c and 2.9d, 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

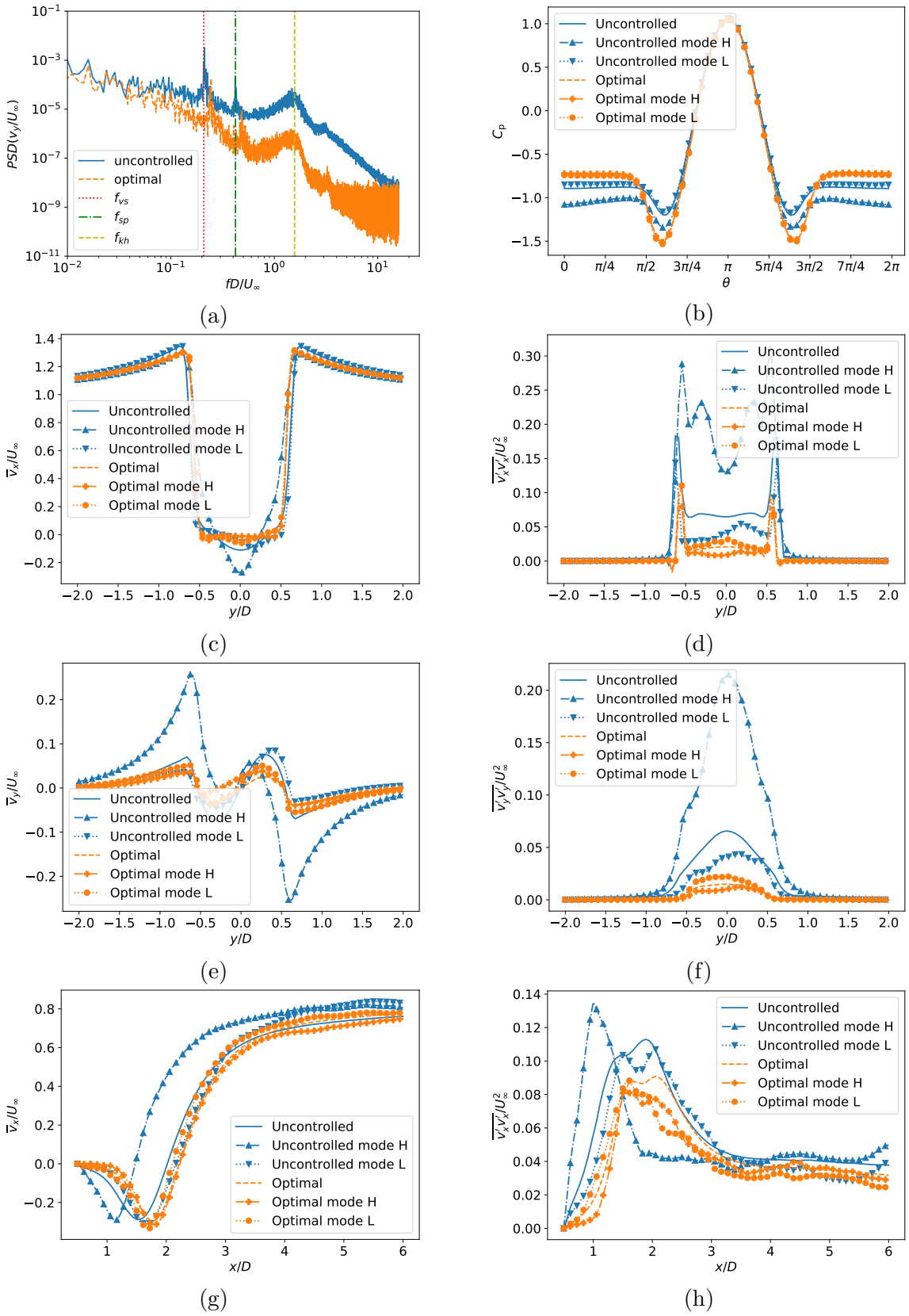
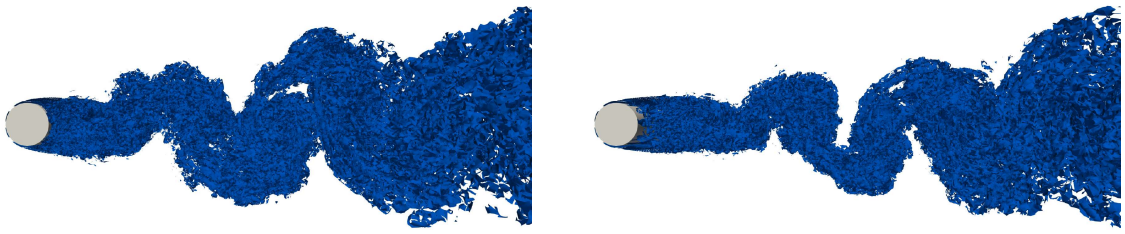


Figure 2.9: Three-dimensional cylinder at Re = 3900. (a) Power spectral densities of the cross-flow velocity component at the probe P_1. (b) Pressure coefficient averaged in the streamwise direction. (c) Time-averaged streamwise velocity profiles and (d) fluctuations at $x/D = 1.06$. (e) Time-averaged 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 2.10: Isosurface of the Q-criterion $Q = 0.1$ for (a) the uncontrolled case and (b) optimal case at $tU_\infty/D = 400$.

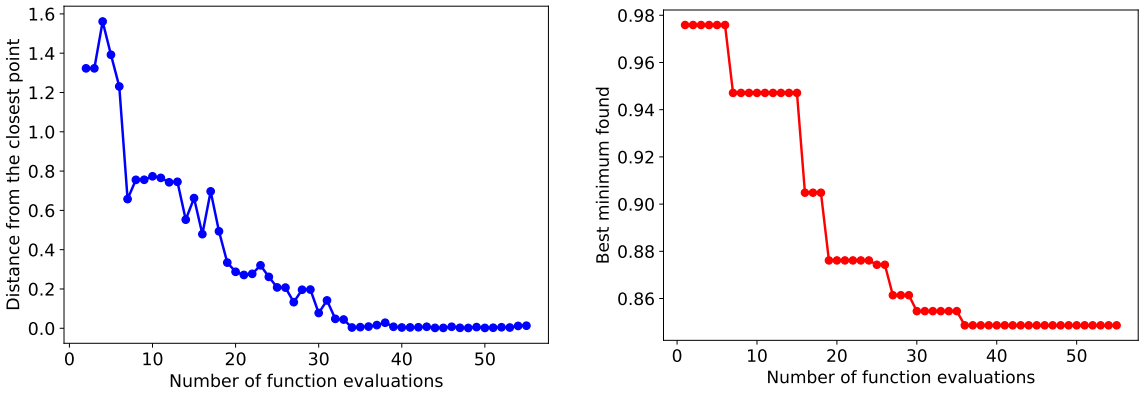
averaged streamwise velocity profile associated with mode H of the uncontrolled case is higher at $y/D = 0$ compared with the other solutions. Smaller averaged velocities are also obtained for this mode at $y/D = -1$ and $y/D = 1$. At $x/D = 2.02$ (not shown here), the gap between the mode H (uncontrolled 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. 2.9e and 2.9f, 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. 2.9g and 2.9h 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 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. 2.10, 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.



(a) (b)

Figure 2.11: 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.

Finally, we present in Fig. 2.11 the BO efficiency on the three-dimensional cylinder control case at Re=3900. Fig. 2.11b 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. 2.11a 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.

2.5 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.

Chapter 3

Gaussian Process, gradients and multi-fidelity: a parametric study

3.1 Introduction

As seen in Chapter 2, Bayesian Optimization (BO) often relies on a Gaussian Process (GP) model. The GP consists in modelling the objective function as the realization of a stochastic process defined by a prior mean and a covariance function, also called the kernel. Once the objective function has been evaluated at design points, the posterior mean and variance can be obtained and predictions with an uncertainty value can be performed at every point of the design space.

An additional feature of the GP is that it can be extended when various sources of information are available. Possible sources of information include the gradients or a low-fidelity model that is less accurate than the initial high-fidelity objective function but faster to evaluate. Such features can easily be implemented in the GP through the covariance matrix.

Derivative information can be added to the model as described in Section 9.4 of Rasmussen and Williams [96] or in Section 7 of Forrester *et al.* [21]. By including the gradient information in the covariance matrix, predictions on both the objective function and gradients values can be obtained. For numerical simulations, the gradients can be calculated through finite differences or through the adjoint methods [40, 28]. The advantage of the latter over the former is that the cost of computing the gradients is independent of the number of design parameters. Lizotte [71] compared BO when the derivative information was included in the Gaussian Process, with the gradient-based method L-BFGS-B. Results showed that BO outperformed L-BFGS-B when the gradients were used. Later, Wu *et al.* [122] developed an acquisition function named derivative Knowledge Gradient in order to dynamically choose whether or not to evaluate the derivatives and in which direction. This acquisition function quantifies the possible gain on the current posterior mean optimum if future objective function and gradient observations were added to the model at a design $\mathbf{s}$. With this acquisition function, they also showed that the BO with gradients could outperform the L-BFGS-B algorithm. Talnikar and Wang [111] also used a GP with gradient information for the BO in Large Eddy Simulations (LES) since the GP can handle the noisy observations of both the objective function and gradients. Adding the derivative information to the GP enabled them to find a better minimum.

Another possible source of information is a model of lower fidelity, i.e. a less accu-

rate model. Practical examples where low/high-fidelity models might be encountered in Computational Fluid Dynamics (CFD) are simulations based on coarse and fine meshes, RANS and LES simulations or loosely and tightly converged simulations, among others. The lower fidelity models are often cheaper to evaluate than their higher fidelity counterparts. Under some circumstances, it might be preferable to perform several function evaluations using a low accuracy model than one function evaluation with high accuracy, at the same computational budget. A model that combines low/high-fidelity sources of information is typically referred to as a multi-fidelity model, and such model can be built, as it will be discussed later, by adjusting the covariance matrix accordingly (see Section 8 in Forrester *et al.* [21]).

Park *et al.* [88] showed, on a 6-dimensional test function, that for the same accuracy we can save through the multi-fidelity model up to 86% of the cost for the same accuracy, whereas for the same computational budget, we can improve the model accuracy up to 51% compared to the single-fidelity GP. For optimization purposes, the multi-fidelity model also showed promising results. For example, Marco *et al.* [79] developed the multi-fidelity Entropy Search acquisition function. They used entropy [34] to trade-off the information content brought by an objective function evaluation on the fidelity j and the cost of sampling on this fidelity. With this acquisition criterion, they were able to get a better minimum and with fewer evaluations of the high-fidelity objective function than with the single-fidelity set up. The same year, Poloczek *et al.* [93] developed the cost-sensitive Knowledge Gradient acquisition function that is the ratio between how the posterior minimum would change with an additional evaluation on a certain fidelity and the cost of evaluating the objective function at this fidelity. Results showed that this method predicted better optimum designs at a lower cost than other existing multi-fidelity optimization methods. Later, Takeno *et al.* [109] created the multi-fidelity Max-Value Entropy Search. Similar to the multi-fidelity Entropy search, the mutual information between the minimum value of the objective function and the objective function value on a certain fidelity at a design point is considered. By dividing this mutual information by the cost of obtaining the objective function at this fidelity, they were also able to outperform other multi-fidelity optimization algorithms with a significant speed up in the computation of the acquisition function compared to the multi-fidelity Entropy Search acquisition function. For a general and broader look of the multi-fidelity models and its applications, we refer to the review of Peherstorfer *et al.* [90].

Finally, multi-fidelity and gradient information can be combined. For example, Ulaganathan *et al.* [114] developed the so-called Gradient-Enhanced co-Kriging (GECoK) method and showed that including gradient information of both the low and high-fidelity models can improve the accuracy of the Kriging model. Han *et al.* [31] also developed the multi-fidelity model with gradients. They used a new generalized hybrid bridge function (the function quantifying the difference between the low and high-fidelity objective functions) to evaluate the accuracy of this model. They showed that combining gradients and multi-fidelity improved the accuracy and robustness of the multi-fidelity model in aerodynamic applications. Yamazaki and Mavriplis [123] also exploited this idea under the name of derivative-enhanced variable-fidelity surrogate model, and applied it to the aerodynamic shape optimization of a two-dimensional airfoil. They found that combining gradient information with variable-fidelity model led to a faster reduction of the objective function.

However, in front of all the possible models if the multi-fidelity is feasible and gradient information available, one may wonder which one to choose for modelling and/or optimization. Indeed, at the best knowledge of the authors, clear indications on which source of information should be prioritized according to the available budget and the different costs of the source of information are missing in the literature. Until now, most studies neglect the gradient or low-fidelity objective function evaluation costs or compare the different models with a different initial budget.

Thus, in this chapter, we compare all the possible models that use the derivative information and/or multi-fidelity for global modelling and optimization purposes when similar computational costs are considered. The global accuracy, the minimum prediction and the minimum obtained in a BO framework are investigated for different budgets and various sources of information costs on three different test cases.

This chapter is organized as follows. The mathematical formulation models with gradient information and/or multi-fidelity sources of information is described in 3.2. Then, in 3.3, the accuracy when we only consider the cost of the evaluation of the high-fidelity objective function is discussed to gain further insight into how these sources of information can improve the model. Since practically speaking, evaluating the gradients or low-fidelity objective function comes with a cost, a parametric study with various budgets, gradients costs, and sample ratio between the high and low-fidelity objective functions is performed on three different test functions of different dimensions in 3.4. The performance of the models is investigated in the context of global modelling and in an optimization framework. Finally, we list our main observations and conclusions in Section 3.5.

3.2 Methodology

3.2.1 Gaussian Process

As already discussed in Chapter 2, one of the central ideas of BO is to optimize a surrogate model instead of the true underlying objective function. Generally, BO relies on a Gaussian Process (GP) [96, 101] that can be seen as a function distribution. The objective function is then considered as the realization of a stochastic process

$$f_2(\mathbf{s}) \sim \mathrm{GP}(\mu_{2,0}(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')), \tag{3.1}$$

where f_2 is the objective function, $\mathbf{s}$ and $\mathbf{s}'$ two points in the design space $\mathcal{S}$, $\mu_{2,0}$ is the mean of the GP and k the covariance function (or kernel). A widely used covariance function is the Radial Basis Function (RBF) kernel

$$k(\mathbf{s}, \mathbf{s}') = \sigma_f^2 \exp\left(-\frac{(\mathbf{s}-\mathbf{s}')^\intercal \mathbf{\Lambda} (\mathbf{s}-\mathbf{s}')}{2}\right), \tag{3.2}$$

where σ_f^2 is the variance and $\mathbf{\Lambda}$ a diagonal squared matrix whose entries are $1/\lambda_i^2$, λ_i being a characteristic length scale along the i-th direction in the design space $\mathcal{S}$.

The subscript $_2$ is used to indicate references to the high-fidelity objective function in the multi-fidelity setting (see Section 3.2.3), and it is also kept in single-fidelity models (Sections 3.2.1 and 3.2.2).

Let us now consider a list of n_2 design points $\mathbf{s}_{2,1:n_2} = (\mathbf{s}_{2,1}, \mathbf{s}_{2,2}, \ldots, \mathbf{s}_{2,n_2})^\mathsf{T}$, all of these designs being included in $\mathcal{S}$. Suppose that we are normally able to observe the objective function at $\mathbf{s}_{2,1:n_2}$ with possibly additive noise, i.e. $\mathbf{q}_{2,1:n_2} = (f_2(\mathbf{s}_{2,1}) + \eta_1, f_2(\mathbf{s}_{2,2}) + \eta_2, \ldots, f_2(\mathbf{s}_{2,n_2}) + \eta_{n_2})^\mathsf{T}$ where

$$\eta_i \sim \mathcal{N}(0, \sigma_\eta^2) = \frac{1}{\sigma_\eta \sqrt{2\pi}} \exp\left(-\frac{\eta_i^2}{2\sigma_\eta^2}\right), \tag{3.3}$$

σ_η^2 being the noise variance and $\mathcal{N}$ the Gaussian distribution. Prior to the evaluation of the objective function, the probability of these observations at the design points $\mathbf{s}_{2,1:n_2}$ is given by

$$P(\mathbf{q}_{2,1:n_2} | \mathbf{s}_{2,1:n_2}, \boldsymbol{\psi}) = \mathcal{N}(\boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2}), \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \sigma_\eta^2 \mathbf{I}_{n_2}), \tag{3.4}$$

where $\mathcal{N}$ is the multivariate Gaussian distribution, $\mathbf{I}_{n_2}$ is the $n_2 \times n_2$ identity matrix, $\boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2}) = (\mu_{2,0}(\mathbf{s}_{2,1}) \ldots \mu_{2,0}(\mathbf{s}_{2,n_2}))^\mathsf{T}$, $\mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = [k_{ij}]$ with $k_{ij} = k(\mathbf{s}_{2,i}, \mathbf{s}_{2,j})$ and $1 \leq i, j \leq n_2$ and $\boldsymbol{\psi} = (\sigma_f, \lambda_i, \sigma_\eta)^\mathsf{T}$ contains the hyperparameters of the model. Eq. 3.4 is the prior distribution of our objective function, and it represents our initial beliefs about the objective function.

We now consider an additional design $\mathbf{s}_{2,n_2+1}$ and the corresponding objective function value f_{2,n_2+1}. The joint distribution is given by

$$\begin{aligned} &P(\mathbf{q}_{2,1:n_2}, f_{2,n_2+1} | \mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}, \boldsymbol{\psi}) = \\ &\mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2}) \\ \mu_{2,0}(\mathbf{s}_{2,n_2+1}) \end{pmatrix}, \begin{pmatrix} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \sigma_\eta^2 \mathbf{I}_{n_2} & \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}) \\ \mathbf{K}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,1:n_2}) & k(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,n_2+1}) \end{pmatrix}\right). \end{aligned} \tag{3.5}$$

The prior distribution can now be conditioned such as this distribution goes through the observed values $\mathbf{q}_{2,1:n_2}$ and write our prediction on the objective function f_{2,n_2+1} at the design $\mathbf{s}_{2,n_2+1}$ as

$$P(f_{2,n_2+1} | \mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}, \mathbf{q}_{2,1:n_2}, \boldsymbol{\psi}) = \mathcal{N}(\mu_{2,n}(\mathbf{s}_{2,n_2+1}), \sigma_{2,n}^2(\mathbf{s}_{2,n_2+1})), \tag{3.6}$$

where

$$\begin{aligned} \mu_{2,n}(\mathbf{s}_{2,n_2+1}) =& \mu_{2,0}(\mathbf{s}_{2,n_2+1}) + \\ & \mathbf{K}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,1:n_2})[\mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \sigma_\eta^2 \mathbf{I}_{n_2}]^{-1}(\mathbf{q}_{2,1:n_2} - \boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2})), \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} \sigma_{2,n}^2(\mathbf{s}_{2,n_2+1}) =& k(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,n_2+1}) - \\ & \mathbf{K}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,1:n_2})[\mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \sigma_\eta^2 \mathbf{I}_{n_2}]^{-1} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}). \end{aligned} \tag{3.8}$$

3.2.2 Gaussian Process with gradient information

The gradient information obtained through adjoint equations or finite differences can also be included in the Gaussian Process (see Section 7 of [21] and Section 9.4 of [96]). The objective function and the observations of its gradient at $\mathbf{s}_{2,1:n_2}$ are considered in the following composite vector:

$$\tilde{\mathbf{q}}_{2,1:n_2} = \begin{pmatrix} \mathbf{q}_{2,1:n_2} \\ \frac{\partial \mathbf{q}_{2,1:n_2}}{\partial s^{(1)}} \\ \frac{\partial \mathbf{q}_{2,1:n_2}}{\partial s^{(2)}} \\ \vdots \\ \frac{\partial \mathbf{q}_{2,1:n_2}}{\partial s^{(N)}} \end{pmatrix},$$

where $s^{(i)}$ is the i^{th} design variable in a N-dimensional space.

Analogously (see Eq. 3.4), a prior can be set on both the objective function and its derivative as follows:

$$P(\tilde{\mathbf{q}}_{2,1:n_2}|\mathbf{s}_{2,1:n_2}, \boldsymbol{\psi}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2}), \tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \tilde{\boldsymbol{\Sigma}}^2), \tag{3.9}$$

where $\tilde{\boldsymbol{\Sigma}}^2 = \text{diag}(\sigma_\eta^2 \mathbf{1}_{1,n_2}, (\sigma_\eta^{(1)})^2 \mathbf{1}_{1,n_2} \ldots, (\sigma_\eta^{(N)})^2 \mathbf{1}_{1,n_2})$, with $\mathbf{1}_{1,n_2}$ being the $1 \times n_2$ ones matrix, $(\sigma_\eta^{(i)})^2$ the noise variance on the derivative observations in the i^{th} direction and $\boldsymbol{\psi} = (\sigma_f, \lambda_i, \sigma_\eta, \sigma_\eta^{(i)})^\mathsf{T}$. In the above,

$$\tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2}) \\ \frac{\partial \boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2})}{\partial s^{(1)}} \\ \frac{\partial \boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2})}{\partial s^{(2)}} \\ \vdots \\ \frac{\partial \boldsymbol{\mu}_{2,0}(\mathbf{s}_{2,1:n_2})}{\partial s^{(N)}} \end{pmatrix}, \tag{3.10}$$

and

$$\tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \mathbf{J}_{\mathbf{s}'}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \\ \mathbf{J}_{\mathbf{s}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \mathbf{H}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.11}$$

where

$$\mathbf{J}_{\mathbf{s}'}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \frac{\partial}{\partial s'^{(1)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \ldots & \frac{\partial}{\partial s'^{(N)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.12}$$

$$\mathbf{J}_{\mathbf{s}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \frac{\partial}{\partial s^{(1)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \\ \vdots \\ \frac{\partial}{\partial s^{(N)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.13}$$

and

$$\mathbf{H}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \frac{\partial^2}{\partial s^{(1)} \partial s'^{(1)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \ldots & \frac{\partial^2}{\partial s^{(1)} \partial s'^{(N)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \\ \vdots & & \\ \frac{\partial^2}{\partial s^{(N)} \partial s'^{(1)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \ldots & \frac{\partial^2}{\partial s^{(N)} \partial s'^{(N)}} \mathbf{K}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}. \tag{3.14}$$

Matrices $\mathbf{J}_{\mathbf{s}'}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2})$, $\mathbf{J}_{\mathbf{s}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2})$ and $\mathbf{H}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2})$ are obtained by applying the following covariance rules

$$\text{cov}\left(f_{2,i}, \frac{\partial f_{2,j}}{\partial s^{(l)}}\right) = \left.\frac{\partial k(\mathbf{s}, \mathbf{s}')}{\partial s'^{(l)}}\right|_{(\mathbf{s},\mathbf{s}')=(\mathbf{s}_{2,i},\mathbf{s}_{2,j})}, \tag{3.15}$$

$$\text{cov}\left(\frac{\partial f_{2,i}}{\partial s^{(l)}}, f_{2,j}\right) = \left.\frac{\partial k(\mathbf{s}, \mathbf{s}')}{\partial s^{(l)}}\right|_{(\mathbf{s},\mathbf{s}')=(\mathbf{s}_{2,i},\mathbf{s}_{2,j})}, \tag{3.16}$$

$$\operatorname{cov}\left(\frac{\partial f_{2,i}}{\partial s^{(l)}}, \frac{\partial f_{2,j}}{\partial s^{(m)}}\right) = \left.\frac{\partial^2 k(\mathbf{s}, \mathbf{s}')}{\partial s^{(l)} \partial s'^{(m)}}\right|_{(\mathbf{s},\mathbf{s}')=(\mathbf{s}_{2,i},\mathbf{s}_{2,j})}, \tag{3.17}$$

where $1 \leq l, m \leq N$ and $1 \leq i, j \leq n_2$.

If we consider an additional design $\mathbf{s}_{2,n_2+1}$ and the vector containing the objective function value and its gradient $\tilde{\mathbf{f}}_{2,n_2+1}$, the joint prior distribution can be written as

$$\begin{aligned} &P(\tilde{\mathbf{q}}_{2,1:n_2}, \tilde{\mathbf{f}}_{2,n_2+1} | \mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}, \boldsymbol{\psi}) = \\ &\mathcal{N}\left(\begin{pmatrix} \tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2}) \\ \tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,n_2+1}) \end{pmatrix}, \begin{pmatrix} \tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \tilde{\boldsymbol{\Sigma}}^2 & \tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}) \\ \tilde{\mathbf{K}}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,1:n_2}) & \tilde{\mathbf{K}}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,n_2+1}) \end{pmatrix}\right). \end{aligned} \tag{3.18}$$

Then, as with Eq. 3.6, by conditioning the prior, we arrive at

$$P(\tilde{\mathbf{f}}_{2,n_2+1} | \mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}, \tilde{\mathbf{q}}_{2,1:n_2}, \boldsymbol{\psi}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{2,n}(\mathbf{s}_{2,n_2+1}), \tilde{\mathbf{K}}_{2,n}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,n_2+1})) \tag{3.19}$$

with

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{2,n}(\mathbf{s}_{2,n_2+1}) =& \tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,n_2+1}) + \\ & \tilde{\mathbf{K}}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,1:n_2})[\tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \tilde{\boldsymbol{\Sigma}}^2]^{-1}(\tilde{\mathbf{q}}_{2,1:n_2} - \tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2})), \end{aligned} \tag{3.20}$$

and

$$\begin{aligned} \tilde{\mathbf{K}}_{2,n}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,n_2+1}) =& \tilde{\mathbf{K}}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,n_2+1}) - \\ & \tilde{\mathbf{K}}(\mathbf{s}_{2,n_2+1}, \mathbf{s}_{2,1:n_2})[\tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \tilde{\boldsymbol{\Sigma}}^2]^{-1}\tilde{\mathbf{K}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,n_2+1}). \end{aligned} \tag{3.21}$$

3.2.3 Multi-fidelity Gaussian Process

As previously discussed, it is also possible to use various fidelity levels, i.e. one level where the objective function is cheap to evaluate but inaccurate (the low-fidelity model) and another one that is more expensive to sample but more accurate (the high-fidelity model).

Following Kennedy and O'Hagan [48], the high-fidelity objective function can be modelled as

$$f_2(\mathbf{s}) = \zeta f_1(\mathbf{s}) + f_{err}(\mathbf{s}), \tag{3.22}$$

where f_2 is the high-fidelity objective function (e.g. in CFD: fine mesh, LES, tightly converged objective function), f_1 is the low-fidelity objective function (e.g. in CFD: coarse mesh, RANS, loosely converged objective function), ζ is a scale parameter and f_{err} is the so-called bridge function. In practice, f_1 and f_{err} are modelled as two independent Gaussian Processes:

$$f_1(\mathbf{s}) \sim \mathrm{GP}(\mu_{1,0}(\mathbf{s}), k_1(\mathbf{s}, \mathbf{s}')), \tag{3.23}$$

$$f_{err}(\mathbf{s}) \sim \mathrm{GP}(\mu_{err,0}(\mathbf{s}), k_{err}(\mathbf{s}, \mathbf{s}')), \tag{3.24}$$

where $\mu_{1,0}$ and $k_1(\mathbf{s}, \mathbf{s}')$ (respectively $\mu_{err,0}$ and $k_{err}(\mathbf{s}, \mathbf{s}')$) are respectively the prior mean function and the covariance function of f_1 (respectively f_{err}). In this study, RBF kernels were chosen for both k_1 and k_{err}:

$$k_1(\mathbf{s},\mathbf{s}') = \sigma_{1,f}^2 \exp\left(-\frac{(\mathbf{s}-\mathbf{s}')^\mathsf{T}\boldsymbol{\Lambda}_1(\mathbf{s}-\mathbf{s}')}{2}\right), \tag{3.25}$$

$$k_{err}(\mathbf{s},\mathbf{s}') = \sigma_{err,f}^2 \exp\left(-\frac{(\mathbf{s}-\mathbf{s}')^\mathsf{T}\boldsymbol{\Lambda}_{err}(\mathbf{s}-\mathbf{s}')}{2}\right), \tag{3.26}$$

where $\sigma_{1,f}^2$ and $\sigma_{err,f}^2$ are the variances, and $\boldsymbol{\Lambda}_1$ (respectively $\boldsymbol{\Lambda}_{err}$) is a diagonal squared matrix whose entries are $1/\lambda_{1,i}^2$ (respectively $1/\lambda_{err,i}^2$) where $\lambda_{1,i}$ and $\lambda_{err,i}$ are characteristic lengthscales along the i-th direction in the design space.

Let us now consider n designs. The first n_1 designs are to be evaluated using the low-fidelity model and the remaining n_2 designs are to be evaluated using the high-fidelity model. Then, we have

$$\mathbf{s}_{1,1:n_1} = \left(\mathbf{s}_{1,1}, \mathbf{s}_{1,2} \dots \mathbf{s}_{1,n_1}\right)^\mathsf{T}, \quad \mathbf{s}_{2,1:n_2} = \left(\mathbf{s}_{2,1}, \mathbf{s}_{2,2} \dots \mathbf{s}_{2,n_2}\right)^\mathsf{T},$$

and

$$\mathbf{q}_{1,1:n_1} = \begin{pmatrix} \mathbf{q}_{1,1} \\ \mathbf{q}_{1,2} \\ \vdots \\ \mathbf{q}_{1,n_1} \end{pmatrix}, \quad \mathbf{q}_{2,1:n_2} = \begin{pmatrix} \mathbf{q}_{2,1} \\ \mathbf{q}_{2,2} \\ \vdots \\ \mathbf{q}_{2,n_2} \end{pmatrix},$$

which we write in compact form as $\mathbf{s}_{mf,1:n} = (\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2})^\mathsf{T}$, and $\mathbf{q}_{mf,1:n} = (\mathbf{q}_{1,1:n_1}, \mathbf{q}_{2,1:n_2})^\mathsf{T}$. The prior distribution is given by

$$P(\mathbf{q}_{mf,1:n}|\mathbf{s}_{mf,1:n}, \boldsymbol{\psi}) = \mathcal{N}(\boldsymbol{\mu}_{mf,0}(\mathbf{s}_{mf,1:n}), \mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \boldsymbol{\Sigma}_{mf}^2), \tag{3.27}$$

where by construction,

$$\boldsymbol{\mu}_{mf,0}(\mathbf{s}_{mf,1:n}) = \begin{pmatrix} \boldsymbol{\mu}_{1,0}(\mathbf{s}_{1,1:n_1}) \\ \zeta\boldsymbol{\mu}_{1,0}(\mathbf{s}_{2,1:n_2}) + \boldsymbol{\mu}_{err,0}(\mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.28}$$

and

$$\begin{aligned} &\mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) = \\ &\begin{pmatrix} \mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{1,1:n_1}) & \zeta\mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}) \\ \zeta\mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}) & \zeta^2\mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \mathbf{K}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \end{aligned} \tag{3.29}$$

with $\mathbf{K}_1$ and $\mathbf{K}_{err}$ being the covariance matrices built with k_1 and k_{err}, respectively, $\boldsymbol{\Sigma}_{mf}^2 = \mathrm{diag}(\sigma_{1,\eta}^2\mathbf{1}_{1,n_1}, \zeta^2\sigma_{1,\eta}^2\mathbf{1}_{1,n_2} + \sigma_{err,\eta}^2\mathbf{1}_{1,n_2})$ with $\sigma_{1,\eta}^2$ being the noise variance of the low-fidelity model, $\sigma_{err,\eta}^2$ being the noise variance of the model of the bridge function, and $\boldsymbol{\psi} = (\sigma_{1,f}, \lambda_{1,i}, \sigma_{err,f}, \lambda_{err,i}, \zeta, \sigma_{1,\eta}, \sigma_{err,\eta})^\mathsf{T}$ contains the hyperparameters of the model.

Eq. 3.29 was obtained by applying the covariance rules

$$\mathrm{cov}(\mathbf{f}_{1,1:n_1}, \mathbf{f}_{1,1:n_1}) = \mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{1,1:n_1}), \tag{3.30}$$

$$\begin{aligned} \mathrm{cov}(\mathbf{f}_{1,1:n_1}, \mathbf{f}_{2,1:n_2}) &= \mathrm{cov}(\mathbf{f}_{1,1:n_1}, \zeta f_1(\mathbf{s}_{2,1:n_2}) + f_{err}(\mathbf{s}_{2,1:n_2})) \\ &= \zeta\mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}), \end{aligned} \tag{3.31}$$

$$\begin{aligned} \mathrm{cov}(\mathbf{f}_{2,1:n_2}, \mathbf{f}_{1,1:n_1}) &= \mathrm{cov}(\zeta f_1(\mathbf{s}_{2,1:n_2}) + f_{err}(\mathbf{s}_{2,1:n_2}), \mathbf{f}_{1,1:n_1}) \\ &= \zeta\mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}), \end{aligned} \tag{3.32}$$

$$\begin{aligned}\operatorname{cov}(\mathbf{f}_{2,1:n_2}, \mathbf{f}_{2,1:n_2}) &= \operatorname{cov}(\zeta f_1(\mathbf{s}_{2,1:n_2}) + f_{err}(\mathbf{s}_{2,1:n_2}), \zeta f_1(\mathbf{s}_{2,1:n_2}) + f_{err}(\mathbf{s}_{2,1:n_2})) \\ &= \zeta^2 \mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \mathbf{K}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}).\end{aligned} \tag{3.33}$$

If we consider a set of test inputs $\mathbf{s}_{mf,n+2} = (\mathbf{s}_{1,n_1+1}, \mathbf{s}_{2,n_2+1})^\mathsf{T}$, where $\mathbf{s}_{1,n_1+1}$ and $\mathbf{s}_{2,n_2+1}$ are designs respectively on the low and high-fidelity objective functions, then the joint prior prediction will be given by

$$\begin{aligned}&P(\mathbf{q}_{mf,1:n}, \mathbf{f}_{mf,n+2} | \mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}, \boldsymbol{\psi}) = \\ &\mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{mf,0}(\mathbf{s}_{mf,1:n}) \\ \boldsymbol{\mu}_{mf,0}(\mathbf{s}_{mf,n+2}) \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \boldsymbol{\Sigma}^2_{mf} & \mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}) \\ \mathbf{K}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) & \mathbf{K}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) \end{pmatrix} \right),\end{aligned} \tag{3.34}$$

where $\mathbf{f}_{mf,n+2} = (f_{1,n_1+1}, f_{2,n_2+1})^\mathsf{T}$, f_{1,n_1+1} and f_{2,n_2+1} being respectively the low and high-fidelity objective functions at respectively $\mathbf{s}_{1,n_1+1}$ and $\mathbf{s}_{2,n_2+1}$.

Finally, the posterior distribution is given by

$$P(\mathbf{f}_{mf,n+2} | \mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}, \mathbf{q}_{mf,1:n}, \boldsymbol{\psi}) = \mathcal{N}(\boldsymbol{\mu}_{mf,n}(\mathbf{s}_{mf,n+2}), \mathbf{K}_{mf,n}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2})) \tag{3.35}$$

with

$$\begin{aligned}&\boldsymbol{\mu}_{mf,n}(\mathbf{s}_{mf,n+2}) = \boldsymbol{\mu}_{mf,0}(\mathbf{s}_{mf,n+2}) \\ &+ \mathbf{K}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,1:n})[\mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \boldsymbol{\Sigma}^2_{mf}]^{-1}(\mathbf{q}_{mf,1:n} - \boldsymbol{\mu}_{mf,0}(\mathbf{s}_{mf,1:n})),\end{aligned} \tag{3.36}$$

$$\begin{aligned}&\mathbf{K}_{mf,n}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) = \mathbf{K}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) \\ &- \mathbf{K}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,1:n})[\mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \boldsymbol{\Sigma}^2_{mf}]^{-1}\mathbf{K}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}).\end{aligned} \tag{3.37}$$

3.2.4 Multi-fidelity Gaussian process with gradient information

We now turn the attention to the case where gradient information is available on one or the two fidelity models. The objective functions and its derivatives at low-fidelity $\mathbf{s}_{1,1:n_1}$ and high-fidelity $\mathbf{s}_{2,1:n_2}$ designs are respectively written as

$$\tilde{\mathbf{q}}_{1,1:n_1} = \begin{pmatrix} \mathbf{q}_{1,1:n_1} \\ \frac{\partial \mathbf{q}_{1,1:n_1}}{\partial s^{(1)}} \\ \frac{\partial \mathbf{q}_{1,1:n_1}}{\partial s^{(2)}} \\ \vdots \\ \frac{\partial \mathbf{q}_{1,1:n_1}}{\partial s^{(N)}} \end{pmatrix}, \quad \tilde{\mathbf{q}}_{2,1:n_2} = \begin{pmatrix} \mathbf{q}_{2,1:n_2} \\ \frac{\partial \mathbf{q}_{2,1:n_2}}{\partial s^{(1)}} \\ \frac{\partial \mathbf{q}_{2,1:n_2}}{\partial s^{(2)}} \\ \vdots \\ \frac{\partial \mathbf{q}_{2,1:n_2}}{\partial s^{(N)}} \end{pmatrix}.$$

Introducing

$$\mathbf{s}_{mf,1:n} = \begin{pmatrix} \mathbf{s}_{1,1:n_1} \\ \mathbf{s}_{2,1:n_2} \end{pmatrix}, \quad \text{and} \quad \tilde{\mathbf{q}}_{mf,1:n} = \begin{pmatrix} \tilde{\mathbf{q}}_{1,1:n_1} \\ \tilde{\mathbf{q}}_{2,1:n_2} \end{pmatrix},$$

where $n = n_1 + n_2$ as before, the prior distribution reads

$$P(\tilde{\mathbf{q}}_{mf,1:n} | \mathbf{s}_{mf,1:n}, \boldsymbol{\psi}) = \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_{mf,0}(\mathbf{s}_{mf,1:n}), \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \tilde{\boldsymbol{\Sigma}}^2_{mf}\right). \tag{3.38}$$

In the above,

$$\tilde{\boldsymbol{\mu}}_{mf,0}(\mathbf{s}_{mf,1:n}) = \begin{pmatrix} \tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s}_{1,1:n_1}) \\ \zeta\tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s}_{2,1:n_2}) + \tilde{\boldsymbol{\mu}}_{err,0}(\mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.39}$$

$$\begin{aligned} &\tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) = \\ &\begin{pmatrix} \tilde{\mathbf{K}}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{1,1:n_1}) & \zeta\tilde{\mathbf{K}}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}) \\ \zeta\tilde{\mathbf{K}}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}) & \zeta^2\tilde{\mathbf{K}}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) + \tilde{\mathbf{K}}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \end{aligned} \tag{3.40}$$

where $(\tilde{\boldsymbol{\mu}}_{1,0}, \tilde{\mathbf{K}}_1)$, and $(\tilde{\boldsymbol{\mu}}_{err,0}, \tilde{\mathbf{K}}_{err})$ are calculated as in Section 3.2.2:

$$\tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s}_{j,1:n_j}) = \begin{pmatrix} \boldsymbol{\mu}_{1,0}(\mathbf{s}_{j,1:n_j}) \\ \frac{\partial \boldsymbol{\mu}_{1,0}(\mathbf{s}_{j,1:n_j})}{\partial s^{(1)}} \\ \frac{\partial \boldsymbol{\mu}_{1,0}(\mathbf{s}_{j,1:n_j})}{\partial s^{(2)}} \\ \vdots \\ \frac{\partial \boldsymbol{\mu}_{1,0}(\mathbf{s}_{j,1:n_j})}{\partial s^{(N)}} \end{pmatrix}, \quad \tilde{\boldsymbol{\mu}}_{err,0}(\mathbf{s}_{j,1:n_j}) = \begin{pmatrix} \boldsymbol{\mu}_{err,0}(\mathbf{s}_{j,1:n_j}) \\ \frac{\partial \boldsymbol{\mu}_{err,0}(\mathbf{s}_{j,1:n_j})}{\partial s^{(1)}} \\ \frac{\partial \boldsymbol{\mu}_{err,0}(\mathbf{s}_{j,1:n_j})}{\partial s^{(2)}} \\ \vdots \\ \frac{\partial \boldsymbol{\mu}_{err,0}(\mathbf{s}_{j,1:n_j})}{\partial s^{(N)}} \end{pmatrix}, \tag{3.41}$$

and

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{j,1:n_j}, \mathbf{s}_{j',1:n_{j'}}) = \begin{pmatrix} \mathbf{K}_1(\mathbf{s}_{j,1:n_j}, \mathbf{s}_{j',1:n_{j'}}) & \mathbf{J}_{1,\mathbf{s}'}(\mathbf{s}_{j,1:n_j}, \mathbf{s}_{j',1:n_{j'}}) \\ \mathbf{J}_{1,\mathbf{s}}(\mathbf{s}_{j,1:n_j}, \mathbf{s}_{j',1:n_{j'}}) & \mathbf{H}_1(\mathbf{s}_{j,1:n_j}, \mathbf{s}_{j',1:n_{j'}}) \end{pmatrix}, \tag{3.42}$$

$$\tilde{\mathbf{K}}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \mathbf{K}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \mathbf{J}_{err,\mathbf{s}'}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \\ \mathbf{J}_{err,\mathbf{s}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) & \mathbf{H}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.43}$$

with j and j' being equal to 1 or 2. $\mathbf{J}_{1,\mathbf{s}'}$, $\mathbf{J}_{1,\mathbf{s}}$ and $\mathbf{H}_1$ (respectively $\mathbf{J}_{err,\mathbf{s}'}$, $\mathbf{J}_{err,\mathbf{s}}$ and $\mathbf{H}_{err}$) are respectively the matrix of Eq. 3.12, Eq. 3.13 and Eq. 3.14 calculated with $\mathbf{K}_1$ (respectively $\mathbf{K}_{err}$) instead of $\mathbf{K}$. We also have by construction, $\tilde{\boldsymbol{\Sigma}}^2_{mf} = \text{diag}(\tilde{\boldsymbol{\Sigma}}^2_1, \tilde{\boldsymbol{\Sigma}}^2_2)$ with

$$\tilde{\boldsymbol{\Sigma}}^2_1 = (\sigma^2_{1,\eta}\mathbf{1}_{1,n_1}, (\sigma^{(1)}_{1,\eta})^2\mathbf{1}_{1,n_1}, \ldots, (\sigma^{(N)}_{1,\eta})^2\mathbf{1}_{1,n_1}), \tag{3.44}$$

and

$$\begin{aligned} &\tilde{\boldsymbol{\Sigma}}^2_2 = \\ &([\zeta^2\sigma^2_{1,\eta} + \sigma^2_{err,\eta}]\mathbf{1}_{1,n_2}, [\zeta^2(\sigma^{(1)}_{1,\eta})^2 + (\sigma^{(1)}_{err,\eta})^2]\mathbf{1}_{1,n_2}, \ldots, [\zeta^2(\sigma^{(N)}_{1,\eta})^2 + (\sigma^{(N)}_{err,\eta})^2]\mathbf{1}_{1,n_2}), \end{aligned} \tag{3.45}$$

where $(\sigma^{(i)}_{1,\eta})^2$ (respectively $(\sigma^{(i)}_{err,\eta})^2$) is the noise variance on the derivative observations of the low-fidelity objective function (respectively the bridge function) in the i^{th} direction of the design space. $\boldsymbol{\psi} = (\sigma_{1,f}, \lambda_{1,i}, \sigma_{err,f}, \lambda_{err,i}, \zeta, \sigma_{1,\eta}, \sigma^{(i)}_{1,\eta}, \sigma_{err,\eta}, \sigma^{(i)}_{err,\eta})^\mathsf{T}$ contains the hyperparameters of the model.

Finally, we consider the cases where gradient information is only available for one of the models. If gradient information is only available for the low-fidelity model, the following substitutions apply:

$$\tilde{\mathbf{q}}_{mf,1:n} = \begin{pmatrix} \tilde{\mathbf{q}}_{1,1:n_1} \\ \mathbf{q}_{2,1:n_2} \end{pmatrix}, \tag{3.46}$$

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}) = \begin{pmatrix} \mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}) \\ \mathbf{J}_{1,\mathbf{s}}(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}) \end{pmatrix}, \tag{3.47}$$

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}) = (\mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}), \mathbf{J}_{1,\mathbf{s}'}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1})), \tag{3.48}$$

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}), \tag{3.49}$$

$$\tilde{\mathbf{K}}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}) = \mathbf{K}_{err}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{2,1:n_2}), \tag{3.50}$$

$$\tilde{\boldsymbol{\Sigma}}_2^2 = [\zeta^2 \sigma_{1,\eta}^2 + \sigma_{err,\eta}^2] \mathbf{1}_{1,n_2}, \tag{3.51}$$

and

$$\boldsymbol{\psi} = (\sigma_{1,f}, \lambda_{1,i}, \sigma_{err,f}, \lambda_{err,i}, \zeta, \sigma_{1,\eta}, \sigma_{1,\eta}^{(i)}, \sigma_{err,\eta})^{\mathsf{T}}. \tag{3.52}$$

Conversely, when the gradient is only available for the high-fidelity model, the following substitutions apply:

$$\tilde{\mathbf{q}}_{mf,1:n} = \begin{pmatrix} \mathbf{q}_{1,1:n_1} \\ \tilde{\mathbf{q}}_{2,1:n_2} \end{pmatrix}, \tag{3.53}$$

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{1,1:n_1}) = \mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{1,1:n_1}), \tag{3.54}$$

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}) = (\mathbf{K}_1(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}), \mathbf{J}_{1,\mathbf{s}'}(\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2})), \tag{3.55}$$

$$\tilde{\mathbf{K}}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}) = \begin{pmatrix} \mathbf{K}_1(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}) \\ \mathbf{J}_{1,\mathbf{s}}(\mathbf{s}_{2,1:n_2}, \mathbf{s}_{1,1:n_1}) \end{pmatrix} \tag{3.56}$$

and

$$\tilde{\boldsymbol{\Sigma}}_1^2 = \sigma_{1,\eta}^2 \mathbf{1}_{1,n_1}. \tag{3.57}$$

If we consider a set of test inputs $\mathbf{s}_{mf,n+2} = (\mathbf{s}_{1,n_1+1}, \mathbf{s}_{2,n_2+1})^{\mathsf{T}}$, where $\mathbf{s}_{1,n_1+1}$ and $\mathbf{s}_{2,n_2+1}$ are designs respectively on the low and high-fidelity objective functions, then the joint prior prediction will be

$$P(\tilde{\mathbf{q}}_{mf,1:n}, \tilde{\mathbf{f}}_{mf,n+2} | \mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}, \boldsymbol{\psi}) = \\ \mathcal{N}\left(\begin{pmatrix} \tilde{\boldsymbol{\mu}}_{mf,0}(\mathbf{s}_{mf,1:n}) \\ \tilde{\boldsymbol{\mu}}_{mf,0}(\mathbf{s}_{mf,n+2}) \end{pmatrix}, \begin{pmatrix} \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \tilde{\boldsymbol{\Sigma}}_{mf}^2 & \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}) \\ \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,1:n}) & \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) \end{pmatrix} \right), \tag{3.58}$$

where $\tilde{\mathbf{f}}_{mf,n+2} = (\tilde{\mathbf{f}}_{1,n_1+1}, \tilde{\mathbf{f}}_{2,n_2+1})^{\mathsf{T}}$, $\tilde{\mathbf{f}}_{1,n_1+1}$ and $\tilde{\mathbf{f}}_{2,n_2+1}$ being respectively the low and high-fidelity objective functions and derivatives at respectively $\mathbf{s}_{1,n_1+1}$ and $\mathbf{s}_{2,n_2+1}$.

Then, the posterior distribution is given by

$$P(\tilde{\mathbf{f}}_{mf,n+2} | \mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}, \tilde{\mathbf{q}}_{mf,1:n}, \boldsymbol{\psi}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{mf,n}(\mathbf{s}_{mf,n+2}), \tilde{\mathbf{K}}_{mf,n}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2})) \tag{3.59}$$

with

$$\tilde{\boldsymbol{\mu}}_{mf,n}(\mathbf{s}_{mf,n+2}) = \tilde{\boldsymbol{\mu}}_{mf,0}(\mathbf{s}_{mf,n+2}) \\ + \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,1:n})[\tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \tilde{\boldsymbol{\Sigma}}^2_{mf}]^{-1}(\tilde{\mathbf{q}}_{mf,1:n} - \tilde{\boldsymbol{\mu}}_{mf,0}(\mathbf{s}_{mf,1:n})), \tag{3.60}$$

$$\tilde{\mathbf{K}}_{mf,n}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) = \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,n+2}) \\ - \tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,n+2}, \mathbf{s}_{mf,1:n})[\tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,1:n}) + \tilde{\boldsymbol{\Sigma}}^2_{mf}]^{-1}\tilde{\mathbf{K}}_{mf}(\mathbf{s}_{mf,1:n}, \mathbf{s}_{mf,n+2}). \tag{3.61}$$

3.2.5 Estimation of hyperparameters

All the models aforementioned depend on the hyperparameters contained in vector $\boldsymbol{\psi}$. For single-fidelity models, if we generically define current observations $\tilde{\mathbf{q}}_{2,1:n_2}$ (that may or not include derivatives) on the high-fidelity objective function at design points $\mathbf{s}_{2,1:n_2}$, $\tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2})$ the prior mean of the GP evaluated at $\mathbf{s}_{2,1:n_2}$, $\tilde{\mathbf{K}}$ the covariance matrix and $\tilde{\boldsymbol{\Sigma}}^2$ the noise variance matrix, the value of $\boldsymbol{\psi}$ is determined by maximizing the logarithm of the marginal likelihood of the model [96]

$$\log P(\tilde{\mathbf{q}}_{2,1:n_2}|\mathbf{s}_{2,1:n_2}, \boldsymbol{\psi}) = \\ -\frac{1}{2}(\tilde{\mathbf{q}}_{2,1:n_2} - \tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2}))^\mathsf{T}(\tilde{\mathbf{K}} + \tilde{\boldsymbol{\Sigma}}^2)^{-1}(\tilde{\mathbf{q}}_{2,1:n_2} - \tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s}_{2,1:n_2})) \\ -\frac{1}{2}\log|\tilde{\mathbf{K}} + \tilde{\boldsymbol{\Sigma}}^2| - \frac{n}{2}\log(2\pi). \tag{3.62}$$

For multi-fidelity models, the hyperparameters treatment is done similarly to [46] and [114]. The first step is to maximize the logarithm of the marginal likelihood of the model on the low-fidelity data given by

$$\log P(\tilde{\mathbf{q}}_{1,1:n_1}|\mathbf{s}_{1,1:n_1}, \boldsymbol{\psi}_1) = \\ -\frac{1}{2}(\tilde{\mathbf{q}}_{1,1:n_1} - \tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s}_{1,1:n_1}))^\mathsf{T}(\tilde{\mathbf{K}}_1 + \tilde{\boldsymbol{\Sigma}}^2_1)^{-1}(\tilde{\mathbf{q}}_{1,1:n_1} - \tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s}_{1,1:n_1})) \\ -\frac{1}{2}\log|\tilde{\mathbf{K}}_1 + \tilde{\boldsymbol{\Sigma}}^2_1| - \frac{n}{2}\log(2\pi), \tag{3.63}$$

where $\tilde{\mathbf{q}}_{1,1:n_1}$ and $\tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s}_{1,1:n_1})$ are respectively the observations and predictions on the low-fidelity objective function (that may or not include gradients information) at $\mathbf{s}_{1,1:n_1}$, $\tilde{\mathbf{K}}_1$ and $\tilde{\boldsymbol{\Sigma}}^2_1$ are respectively the associated covariance and noise variance matrices. $\boldsymbol{\psi}_1$ contains $\sigma_{1,f}$, $\lambda_{1,i}$, $\sigma_{1,\eta}$ and if the derivatives are considered, $\sigma^{(i)}_{1,\eta}$.

Then, the logarithm of the marginal likelihood of the bridge function is maximized

$$\log P(\tilde{\mathbf{d}}_{1:n2}|\mathbf{s}_{2,1:n_2}, \boldsymbol{\psi}_{err}, \zeta) = \\ -\frac{1}{2}(\tilde{\mathbf{d}}_{1:n2} - \tilde{\boldsymbol{\mu}}_{err,0}(\mathbf{s}_{2,1:n_2}))^\mathsf{T}(\tilde{\mathbf{K}}_{err} + \tilde{\boldsymbol{\Sigma}}^2_{err})^{-1}(\tilde{\mathbf{d}}_{1:n2} - \tilde{\boldsymbol{\mu}}_{err,0}(\mathbf{s}_{2,1:n_2})) \\ -\frac{1}{2}\log|\tilde{\mathbf{K}}_{err} + \tilde{\boldsymbol{\Sigma}}^2_{err}| - \frac{n}{2}\log(2\pi), \tag{3.64}$$

where $\tilde{\mathbf{d}}_{1:n2} = \tilde{\mathbf{q}}_{2,1:n_2} - \zeta\tilde{\mathbf{q}}_{1,1:n_1}(\mathbf{s}_{2,1:n_2})$. $\tilde{\mathbf{q}}_{2,1:n_2}$, $\tilde{\mathbf{q}}_{1,1:n_1}(\mathbf{s}_{2,1:n_2})$ are the observations of respectively the high-fidelity and low-fidelity objective functions (and possibly derivatives) at $\mathbf{s}_{2,1:n_2}$. If observations on the low-fidelity objective function (and possibly gradients) are not available at some points $\mathbf{s}_{2,i:j}$ of $\mathbf{s}_{2,1:n_2}$, the low-fidelity prediction $\tilde{\boldsymbol{\mu}}_{1,n_1}(\mathbf{s}_{2,i:j})$ is used instead (see Section 8 of [21] and [114]). If derivative observations are available only on one fidelity, only the objective functions observations are used in $\tilde{\mathbf{q}}_{2,1:n_2}$ and $\tilde{\mathbf{q}}_{1,1:n_1}(\mathbf{s}_{2,1:n_2})$. $\tilde{\boldsymbol{\mu}}_{err,0}(\mathbf{s}_{2,1:n_2})$ is the prior mean prediction of the bridge function at $\mathbf{s}_{2,1:n_2}$, $\tilde{\mathbf{K}}_{err}$ and $\tilde{\boldsymbol{\Sigma}}^2_{err}$ represent respectively the covariance and noise variance matrices of the bridge function whereas $\boldsymbol{\psi}_{err}$ contains $\sigma_{err,f}$, $\lambda_{err,i}$, $\sigma_{err,\eta}$ and for the multi-fidelity model with derivative information on both fidelities $\sigma^{(i)}_{1,f}$ and $\sigma^{(i)}_{err,f}$. Note that Eq. 3.63 and Eq. 3.64 are independent and can be maximized in parallel.

In this study, Eq. 3.62, Eq. 3.63 and Eq. 3.64 are maximized with a quasi-Newton method with 10 different starting points chosen randomly.

3.2.6 Scaling

As advised in Forrester *et al.* [21], it is good practice to normalize the design parameters onto the unit cube $[0,1]^N$ to avoid scaling issues. We adhere to this recommendation and this step is performed through:

$$s^{(i)} = \frac{s^{(i)}_{ini} - s^{(i)}_{ini,low}}{s^{(i)}_{ini,up} - s^{(i)}_{ini,low}}, \quad i = 1, 2, \ldots, N \tag{3.65}$$

where $s^{(i)}$ are the new design variables in the unit cube, $s^{(i)}_{ini,low}$ and $s^{(i)}_{ini,up}$ are, respectively, the lower and upper bounds of the design variables $s^{(i)}_{ini}$ in the original design space.

It is also habitual to set the prior means $\boldsymbol{\mu}_{2,0}(\mathbf{s})$, $\boldsymbol{\mu}_{1,0}(\mathbf{s})$ and $\boldsymbol{\mu}_{err,0}(\mathbf{s})$ equal to 0 in the whole design space $\mathcal{S}$. By construction, $\tilde{\boldsymbol{\mu}}_{2,0}(\mathbf{s})$, $\tilde{\boldsymbol{\mu}}_{1,0}(\mathbf{s})$ and $\tilde{\boldsymbol{\mu}}_{err,0}(\mathbf{s})$ are also equal to 0. We also adhere to this common practice for the rest of the chapter.

As done in Takeno *et al.* [109], the GP will use normalized observations with zero mean and standard deviation equal to one:

$$\mathbf{q}_{j,1:n_j} = \frac{\mathbf{q}_{j,1:n_j,ini} - \mathrm{ME}(\mathbf{q}_{j,1:n_j,ini})}{\mathrm{SD}(\mathbf{q}_{j,1:n_j,ini})}, \quad j = 1, 2, \tag{3.66}$$

where $\mathbf{q}_{j,1:n_j}$ are the n_j normalized training observations of $\mathbf{q}_{j,1:n_j,ini}$, $\mathrm{ME}(\mathbf{q}_{j,1:n_j,ini})$ and $\mathrm{SD}(\mathbf{q}_{j,1:n_j,ini})$ are respectively the mean and the standard deviation of the set $\mathbf{q}_{j,1:n_j,ini}$. The normalized observations enable to be more coherent with the assumption of the prior means $\boldsymbol{\mu}_{2,0}(\mathbf{s})$, $\boldsymbol{\mu}_{1,0}(\mathbf{s})$ and $\boldsymbol{\mu}_{err,0}(\mathbf{s})$ equal to 0. In numerical experiments involving the one-dimensional example presented in Section 3.3.1, it was observed that the optimization of Eq. 3.62, Eq. 3.63 and Eq. 3.64 was more robust using normalized observations.

Owing to the normalization of both the design variables and observations, the derivative observations of the GP are also scaled:

$$\frac{\partial \mathbf{q}^{(i)}_{j,1:n_j}}{\partial s^{(i)}} = \frac{s^{(i)}_{ini,up} - s^{(i)}_{ini,low}}{\mathrm{SD}(\mathbf{q}_{j,1:n_j,ini})} \frac{\partial \mathbf{q}^{(i)}_{j,1:n_j,ini}}{\partial s^{(i)}_{ini}}, \quad j = 1, 2, \tag{3.67}$$

where $\frac{\partial \mathbf{q}^{(i)}_{j,1:n_j}}{\partial s^{(i)}}$ and $\frac{\partial \mathbf{q}^{(i)}_{j,1:n_j,ini}}{\partial s^{(i)}_{ini}}$ are respectively the normalized and non-normalized derivative observations in respectively the unit cube and original design space of the objective function j in the i^{th} direction.

The predicted value $\mu_{j,n,ini}$ of the objective function f_j at the original design $\mathbf{s}_{ini}$ after n total samples is found with

$$\mu_{j,n,ini}(\mathbf{s}_{ini}) = \mathrm{SD}(\mathbf{q}_{j,1:n_j,ini}) \times \mu_{j,n}(\mathbf{s}) + \mathrm{ME}(\mathbf{q}_{j,1:n_j,ini}), \quad j = 1, 2, \tag{3.68}$$

where $\mu_{j,n}(\mathbf{s})$ is the mean of the GP at $\mathbf{s}$ of the normalized objective function f_j.

3.3 Validation of the models

We show in this section how the developments in the previous section can be used to improve the accuracy of the surrogate model when the costs of evaluating the gradients and the low-fidelity model are not considered. In the remaining of the chapter, we adopt the following conventions when referring to the characteristics of the different models: SF and MF stand for, respectively, the single- and multi-fidelity models, and in the models where gradient information is used, it is followed by G, G_1 and/or G_2, to indicate the usage of gradients, gradients from the low-fidelity model and gradients from the high-fidelity models, respectively. These variants have been described in Section 3.2

In the following, validation and results are performed using the emukit software [87]. An in-house kernel version was developed for SFG, $\mathrm{MFG_1G_2}$, $\mathrm{MFG_1}$ and $\mathrm{MFG_2}$ and implemented in the software.

To assess the accuracy of each model, we use the normalized root mean squared error (NRMSE) defined as

$$\mathrm{NRMSE} = \frac{\sqrt{\sum_{j=1}^{M} \frac{(\mu_{2,n,ini}(\mathbf{s}_{j,ini}) - f_2(\mathbf{s}_{j,ini}))^2}{M}}}{\max f_2(\mathbf{s}_{1:M,ini}) - \min f_2(\mathbf{s}_{1:M,ini})} \tag{3.69}$$

where $\mathbf{s}_{j,ini}$ is the j^{th} design of M sample points $\mathbf{s}_{1:M,ini}$ drawn from a uniform Cartesian grid when $N = 2$ or a Sobol design [11] when $N > 2$. Low values of the NRMSE are desired for an adequate approximation of the high-fidelity function. For the computation of the NRMSE, we set the value $M = 10000$.

Also, for the multi-fidelity models, to determine whether the difference between the true underlying function and the model comes from the low-fidelity function or the bridge function, we define the following coefficient of determinations:

$$R_1^2 = 1 - \frac{\sum_{j=1}^{M} [\overline{f}_1(\mathbf{s}_j) - \mu_{1,n}(\mathbf{s}_j)]^2}{\sum_{j=1}^{M} [\overline{f}_1(\mathbf{s}_j) - \mathrm{ME}(\overline{f}_1(\mathbf{s}_j))]^2}, \tag{3.70}$$

$$R_{err}^2 = 1 - \frac{\sum_{j=1}^{M} [\overline{f}_{err}(\mathbf{s}_j) - \mu_{err,n}(\mathbf{s}_j)]^2}{\sum_{j=1}^{M} [\overline{f}_{err}(\mathbf{s}_j) - \mathrm{ME}(\overline{f}_{err}(\mathbf{s}_j))]^2}, \tag{3.71}$$

where

$$\overline{f}_1(\mathbf{s}_j) = \frac{f_1(\mathbf{s}_{j,ini}) - \mathrm{ME}(\mathbf{q}_{1,1:n_1,ini})}{\mathrm{SD}(\mathbf{q}_{1,1:n_1,ini})}, \tag{3.72}$$

$$\overline{f}_{err}(\mathbf{s}_j) = \frac{f_2(\mathbf{s}_{j,ini}) - \mathrm{ME}(\mathbf{q}_{2,1:n_2,ini})}{\mathrm{SD}(\mathbf{q}_{2,1:n_2,ini})} - \zeta \frac{f_1(\mathbf{s}_{j,ini}) - \mathrm{ME}(\mathbf{q}_{1,1:n_1,ini})}{\mathrm{SD}(\mathbf{q}_{1,1:n_1,ini})}, \tag{3.73}$$

with ME and SD being the mean and standard deviation of the set between the parenthesis. As the NRMSE, the coefficients of correlation quantify the precision of the models where values equal to 1 indicate that the model is accurate.

3.3.1 One-dimensional example

To illustrate the models presented in Section 3.2, we now turn the attention to the one-dimensional function considered in Forrester [21]. The high-fidelity model is given by

$$f_2(s_{ini}) = (6s_{ini} - 2)^2 \sin(12s_{ini} - 4), \tag{3.74}$$

whereas the low-fidelity model reads

$$f_1(s_{ini}) = Af_2(s_{ini}) + B(s_{ini} - 0.5) - C, \tag{3.75}$$

Following Chapter 8 in [21], the design space is already the unit cube $s = s_{ini} \in [0, 1]$ and the following numerical values are considered: $A = 0.5$, $B = 10$ and $C = -5$.

Four high-fidelity objective function evaluations $\mathbf{q}_{2,1:n_2,ini}$ are performed at the points $\mathbf{s}_{2,1:n_2,ini} = (0, 0.4, 0.6, 1)^\mathsf{T}$. With the multi-fidelity models, seven low-fidelity objective function evaluations $\mathbf{q}_{1,1:n_1,ini}$ at $\mathbf{s}_{1,1:n,ini} = (0, 0.2, 0.4, 0.6, 0.7, 0.8, 1.0)^\mathsf{T}$ are also included. Gradient information is evaluated at every point and included into the respective models.

The mean $\mu_{n,2,ini}$ of f_2 for all the models described previously is depicted in Fig. 3.1. On Fig. 3.1a, the mean of the high-fidelity objective function for SF, SFG and MF are represented with the low and high-fidelity objective functions. As it can be observed, including gradient information or using a multi-fidelity model leads to improvements in accuracy with respect to the SF, as it is able to detect the dip around $s_{ini} = 0.75$. However, both SFG and MF exhibit areas where the model can be further improved. In Fig. 3.1b, the mean of the high-fidelity objective function for $\mathrm{MFG_1G_2}$, $\mathrm{MFG_1}$ and $\mathrm{MFG_2}$ is shown. In comparison to SFG and MF, combining gradients and multi-fidelity results into a model with higher accuracy.

3.3.2 Two-dimensional validation

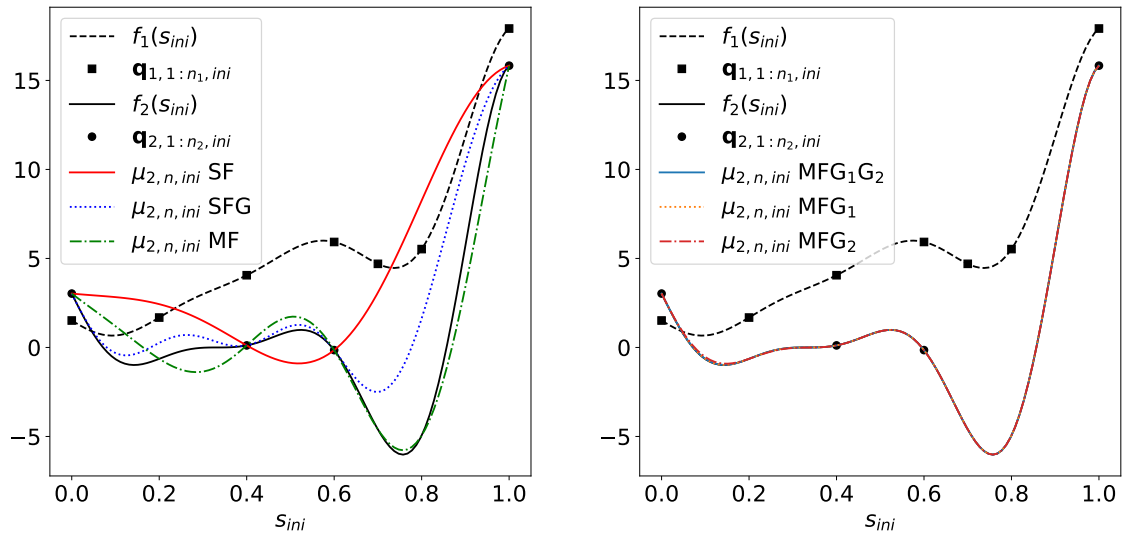
As a two-dimensional example, we use the same objective functions as in Yamazaki *et al.* [123]. The high-fidelity function is defined in $\mathbf{s}_{ini} \in [-2, 2]^N$, $N = 2$, and it reads

$$f_2(\mathbf{s}_{ini}) = \cos(\sum_{k=1}^{N} s_{ini}^{(i)}). \tag{3.76}$$

Several low-fidelity models are considered:

$$f_1^{Multi}(\mathbf{s}_{ini}) = 0.1 f_2(\mathbf{s}_{ini}), \tag{3.77}$$

$$f_1^{Shift}(\mathbf{s}_{ini}) = f_2(\mathbf{s}_{ini}) - 1, \tag{3.78}$$



(a) (b)

Figure 3.1: Mean $\mu_{2,n,ini}$ of f_2 for the different models and the Forrester function. The solid black line is the high-fidelity objective function f_2 and the dashed black line represents the low-fidelity objective function f_1. The circle markers are the observations of f_2 whereas the square markers are for f_1. (a) When the sources of information are used separately: SF, SFG and MF. (b) When the gradient information and multi-fidelity are combined: MFG$_1$G$_2$, MFG$_1$ and MFG$_2$.

$$f_1^{Xshift}(\mathbf{s}_{ini}) = f_2(\mathbf{s}_{ini} + 0.1), \tag{3.79}$$

and

$$f_1^{Lin}(\mathbf{s}_{ini}) = f_2(\mathbf{s}_{ini}) + 0.1(s_{ini}^{(1)} + s_{ini}^{(2)}). \tag{3.80}$$

For the multi-fidelity models, we use $n_1 = 50$ low-fidelity samples drawn from a Sobol design. The gradients are evaluated at each sample. We focus the attention to the evolution of the NRMSE as high-fidelity samples n_2 are included in the model. We sample $n_2 = 5i$, $i = [1, 2, \ldots, 10]$, high-fidelity samples. The samples are again drawn from a Sobol design, and they coincide with the first n_2 points of the low-fidelity sample points. In the case of models that include gradient information into the high-fidelity model, the gradients are also evaluated at each new sample.

The hyperparameters of the models are determined by maximizing the marginal likelihood using a gradient-based method starting from ten different guesses. In this process it is thus possible that a local minimum of the marginal likelihood is found or that the marginal likelihood presents several minima corresponding then to different values of the hyperparameters. With the aim of obtaining results that are statistically relevant, for each choice of n_2, the experiment is repeated ten times and the median of the results are considered. The median of a quantity is the intermediate value between the worst half and the best half quantity values obtained on all the runs. It is an indicator of the general central tendency of the results and is less sensitive to extreme results obtained during a run than the average.

The median NRMSE as a function of the number of high-fidelity samples n_2 for the four low-fidelity models defined previously is depicted in Fig. 3.2.

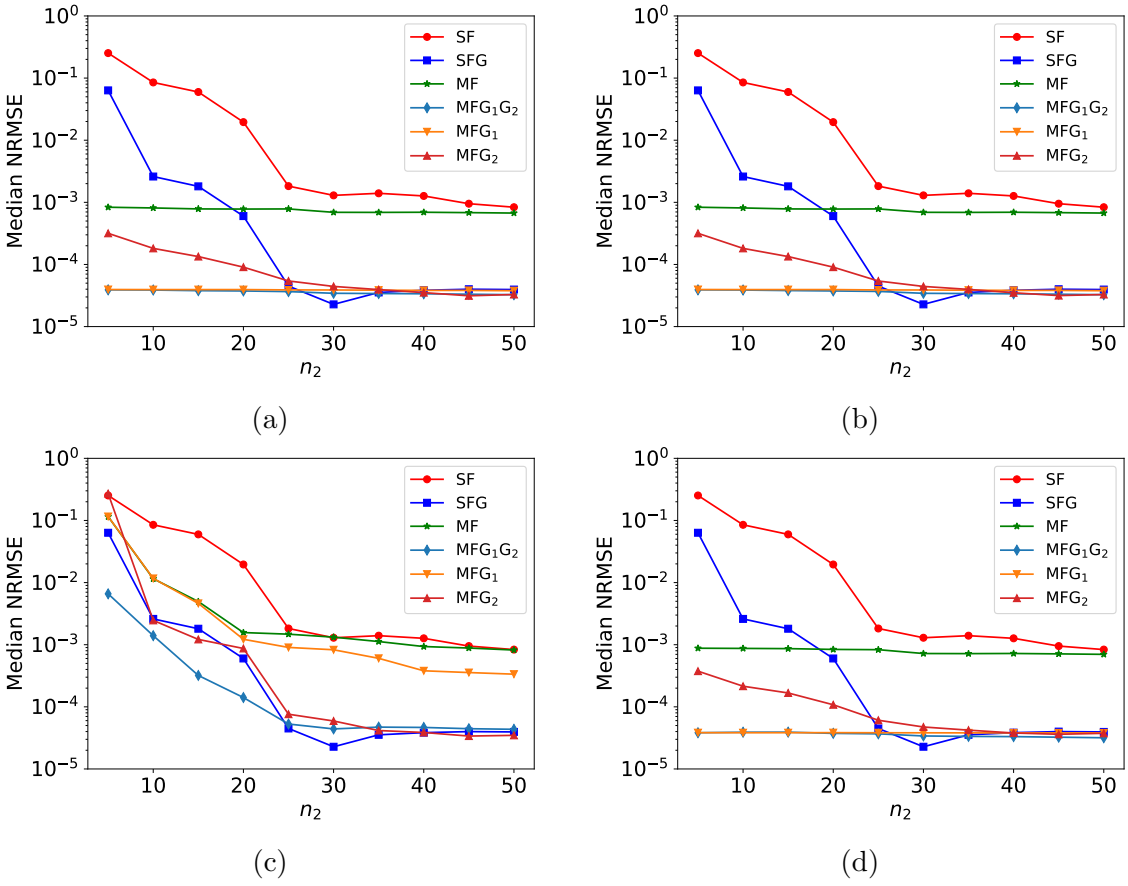


Figure 3.2: Median NRMSE of the different models for the high-fidelity cosine function Eq. 3.76 as a function of the number of high-fidelity sample points. The low-fidelity objective function is taken as: a) f_1^{Multi}, b) f_1^{Shift}, c) f_1^{Xshift}, d) f_1^{Lin}. For the multi-fidelity models, $n_1 = 50$ low-fidelity samples are used.

As it can be observed, SFG and MF outperform SF. Generally, with few high-fidelity samples ($n_2 \leq 15$), MF is more accurate than SFG but as n_2 increases the opposite is observed. The only exception in that case is for f_1^{Xshift} where for each n_2 considered SFG has a lower NRMSE than MF. One explanation is due to the linear model considered for MF in Eq. 3.22. Whereas f_1^{Multi}, f_1^{Shift} and f_1^{Lin} can be easily expressed as in Eq. 3.22, f_1^{Xshift} can not. Yamazaki and Mavriplis [123] also observed worse performances for this test function and concluded that the most important aspect for the MF model was not that the values of the low and high-fidelity functions were close but rather than these two functions had similar trends.

In almost all the test cases considered, and for all the considered n_2, MFG_1G_2, MFG_1 and MFG_2 perform better than MF. The only exception is for f_1^{Xshift} with the MFG_2 model when $n_2 = 5$. With a high number of high-fidelity samples, we do not observe however a significant improvement of the MFG_1G_2, MFG_1 and MFG_2 models over SFG. Also, when the MF model is accurate, as for f_1^{Multi}, f_1^{Shift} and f_1^{Lin}, adding the gradient information on a high number of low-fidelity samples performs better than including the gradient information on few high-fidelity samples (*cf.* the NRMSE of MFG_1 and MFG_2 for $n_2 \leq 10$ in Fig. 3.2a, Fig. 3.2b and Fig. 3.2d).

Even if for some configurations MFG_1 and MFG_2 have slightly lower NRMSE than MFG_1G_2, the latter model is robust and accurate even when the high-fidelity objective function is not a linear function of the low-fidelity objective function (Fig. 3.2c).

The absolute error $\text{AE}(\mathbf{s}_{ini}) = |\mu_{2,n,ini}(\mathbf{s}_{ini}) - f_2(\mathbf{s}_{ini})|$ is represented in Fig. 3.3 for f_1^{Multi} and $n_2 = 5$. In the case of SF, the model is highly inaccurate far away from the high-fidelity samples, as can be observed, for example, in the bottom left corner of Fig. 3.3b.

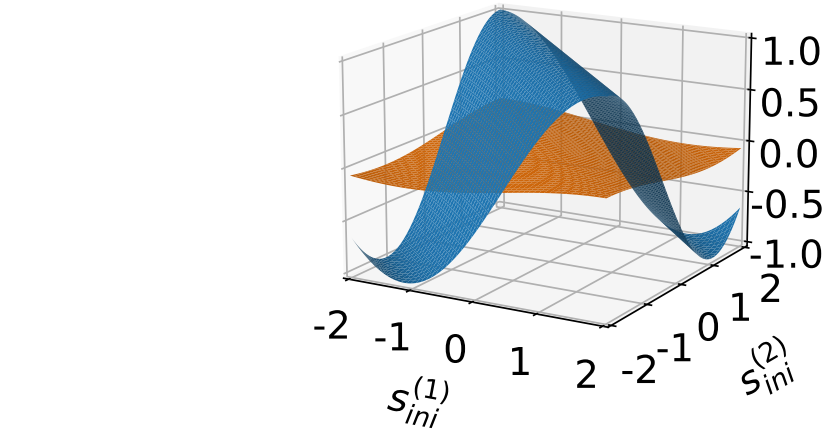
Adding gradient information (Fig. 3.3c) enables us to get a better estimation in this area and leads to a reduction of the maximum AE by a factor of two, approximately.

The MF model (Fig. 3.3d) approximates areas located far away from high-fidelity samples better than SFG since low-fidelity samples are set in these areas. Note that the highest AE of SFG is reduced by approximately a factor of 22 with MF.

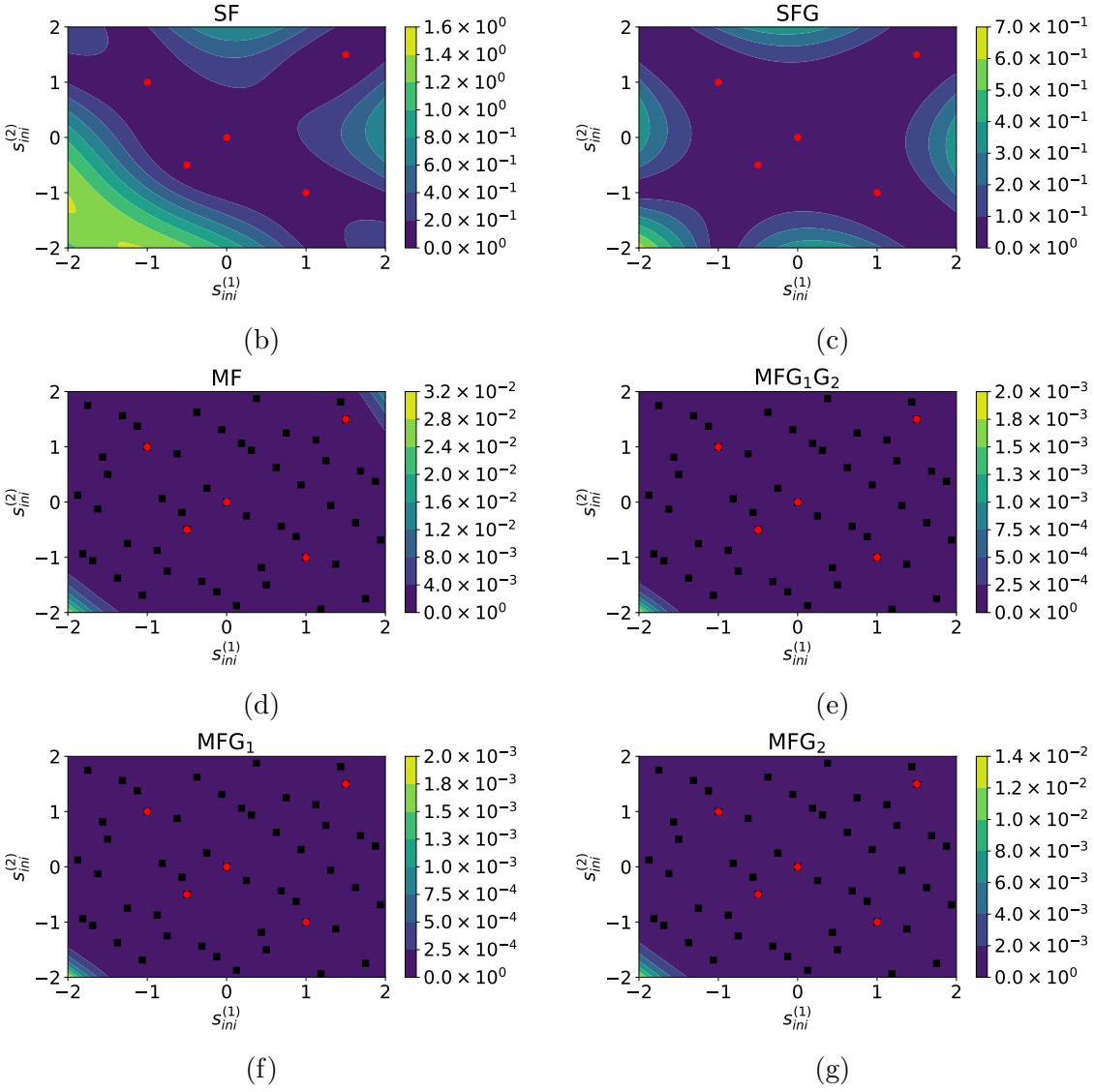
Finally, combining gradients and multi-fidelity gives the most accurate representation of f_2. Indeed, even if MF performs fairly well in this example as the AE does not exceed 2.8×10^{-2} elsewhere, using the models MFG_1G_2, MFG_1 and MFG_2 reduces the maximum AE of MF approximately by a factor of 14, 14 and 2, respectively. Indeed, the uncertainties in the top right and bottom left corner of Fig. 3.3d are reduced as the gradients yield a better approximation in these locations.

3.4 Parametric study

Until now, we have considered that the computational cost associated with the evaluation of the gradient and the low-fidelity model were negligible compared to that of the high-fidelity objective function. In practice, this is rarely the case as even if the low-fidelity function is usually cheaper to evaluate, its computational cost might represent a significant fraction of the total available computational budget. Furthermore, obtaining gradient information has a computational cost that is comparable to that of the evaluation of the objective function. This is typically the case when the gradients are obtained, for instance, using the adjoint equations.



(a)



(b) (c) (d) (e) (f) (g)

Figure 3.3: a) f_2 (blue) and f_1^{Multi} (dark orange) in the initial design space. AE with f_1^{Multi} and $n_2 = 5$: b) SF, c) SFG, d) MF, e) MFG_1G_2, f) MFG_1 and g) MFG_2. Red dots and black squares are respectively the high and low-fidelity design points.

In the following, the different models are compared for modelling and optimization purposes when the cost of each source of information is considered. Different budgets and gradient costs are used to evaluate the performances of the models on two benchmark objective functions and one CFD application.

In order to assess the efficiency of the optimization algorithms, the normalized inference regret (NIR) and the normalized simple regret (NSR) are defined as follows:

$$\text{NIR} = \frac{f_2(\arg\min \mu_{2,n,ini}(\mathbf{s}_{ini})) - \min f_2(\mathbf{s}_{ini})}{\max f_2(\mathbf{s}_{1:M,ini}) - \min f_2(\mathbf{s}_{1:M,ini})}, \tag{3.81}$$

and

$$\text{NSR} = \frac{\min \mathbf{q}_{2,1:n_2,ini} - \min f_2(\mathbf{s}_{ini})}{\max f_2(\mathbf{s}_{1:M,ini}) - \min f_2(\mathbf{s}_{1:M,ini})}, \tag{3.82}$$

where $\min f_2(\mathbf{s}_{ini})$ is obtained with the L-BFGS-B algorithm and represents the true minimum. The inference regret and simple regret are commonly used in the multi-fidelity setting; see, for instance, [109]. The inference regret (respectively the simple regret) represents the gap between the objective function evaluated at the point with the lowest mean (respectively the current minimum obtained) on the high-fidelity level and the true minimum. The inference regret is necessary in the multi-fidelity setting since adding low-fidelity samples to the model does not change the simple regret but can improve the minimum prediction. On the other hand, the simple regret quantifies the current gain that we observed on the high-fidelity objective function. As the NRMSE, the inference and simple regret are normalized by the range as in Eq. 3.81 and Eq 3.82.

We also introduce the computational cost factor (CCF) as the total cost of the objective function and gradients observations. We define c_1 and c_2 as the costs of observing respectively the low and high-fidelity objective functions. For simplicity and since it is often the case in CFD, we consider that the ratio between the computational cost of the gradient observation and the objective function evaluation on the same fidelity is the same for both fidelity. We denote this ratio by c_∇. Thus, for the different models, we have

$$\text{CCF(SF)} = n_2 \times c_2, \tag{3.83}$$

$$\text{CCF(SFG)} = n_2 \times c_2(1 + c_\nabla), \tag{3.84}$$

$$\text{CCF(MF)} = n_2 \times c_2 + n_1 \times c_1, \tag{3.85}$$

$$\text{CCF(MFG}_1\text{G}_2\text{)} = n_2 \times c_2(1 + c_\nabla) + n_1 \times c_1(1 + c_\nabla), \tag{3.86}$$

$$\text{CCF(MFG}_1\text{)} = n_2 \times c_2 + n_1 \times c_1(1 + c_\nabla), \tag{3.87}$$

and

$$\text{CCF(MFG}_2\text{)} = n_2 \times c_2(1 + c_\nabla) + n_1 \times c_1. \tag{3.88}$$

For optimization purposes, we will use the Algorithm 2, described for the MFG_1G_2 model. For all the other models, the initial data set $\mathcal{D}$ has to be adapted according to the data possible and obviously, if the gradients are not available on the high-fidelity model, they are not evaluated.

In Algorithm 2, the CCF_{max} is the maximum CCF used for the optimization. It also should be noted that the next design $\mathbf{s}_{2,n_2+1}$ is not selected using Expected Improvement [85] or the Probability of Improvement [54] acquisition criteria, as it is

Algorithm 2: Optimization algorithm

Initialization:
$\mathcal{D}_{ini} = \{\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}, \mathbf{q}_{1,1:n_1,ini}, \nabla\mathbf{q}_{1,1:n_1,ini}, \mathbf{q}_{2,1:n_2,ini}, \nabla\mathbf{q}_{2,1:n_2,ini}\}$;
Compute the CCF ;
while $CCF < CCF_{max}$ **do**
  Rescale $\mathcal{D}_{ini}$ in order to obtain
  $\mathcal{D} = \{\mathbf{s}_{1,1:n_1}, \mathbf{s}_{2,1:n_2}, \mathbf{q}_{1,1:n_1}, \nabla\mathbf{q}_{1,1:n_1}, \mathbf{q}_{2,1:n_2}, \nabla\mathbf{q}_{2,1:n_2}\}$ (Eq. 3.66 and Eq. 3.67) ;
  Update the Gaussian Process with $\mathcal{D}$;
  Find the design $\mathbf{s}_{2,n_2+1} = \arg\min \mu_{2,n}(\mathbf{s})$;
  Retrieve the original design $\mathbf{s}_{2,n_2+1,ini}$ (with Eq. 3.65);
  Observe the high-fidelity objective function $\mathbf{q}_{2,n_2+1,ini} = f_2(\mathbf{s}_{2,n_2+1,ini})$ and the gradient $\nabla\mathbf{q}_{2,n_2+1,ini} = \nabla f_2(\mathbf{s}_{2,n_2+1,ini})$;
  $\mathcal{D}_{ini} = \mathcal{D}_{ini} \cup \{\mathbf{s}_{2,n_2+1}, \mathbf{q}_{2,n_2+1,ini}, \nabla\mathbf{q}_{2,n_2+1,ini}\}$;
  $n_2 = n_2 + 1$;
  Update the CCF ;
end
Find the minimum $q^*_{2,n^*_2,ini}$ and the corresponding design $\mathbf{s}^*_{2,n^*_2}$ in $\mathcal{D}_{ini}$;
return $\{\mathbf{s}^*_{2,n^*_2}, q^*_{2,n^*_2,ini}\}$

often done in Bayesian Optimization. The reason is the same as mentioned in Lam *et al.* [58]: after a large number of high-fidelity objective functions n_2, optimizing these acquisition functions is equivalent to find an isolated peak in a flat zone. This can become challenging and can introduce error in the results when the optimum is not found. We also tried to determine the next design with the following acquisition criterion:

$$\mathbf{s}_{2,n_2+1} = \arg\min[\mu_{2,n}(\mathbf{s}) - \beta\sigma_{2,n}(\mathbf{s})], \tag{3.89}$$

where $\sigma_{2,n}(\mathbf{s})$ is the standard deviation of the high-fidelity model at $\mathbf{s}$ and $\beta = 2$. This acquisition criterion was intended in order to balance the exploitation (sampling in promising areas) and exploration (sampling in zones where the uncertainty is high in order to improve the model) as is normally performed in Bayesian Optimization. However, the lowest NSR values were obtained when the exploitation term (second right-hand side term of Eq. 3.89) was not included. Thus, the next design is determined with

$$\mathbf{s}_{2,n_2+1} = \arg\min \mu_{2,n}(\mathbf{s}). \tag{3.90}$$

This acquisition criterion was then chosen for all the models in order to avoid that the difference of performances between two models could be related to a different choice of acquisition functions.

In the following, numerical experiments have been carried out ten times in order to compute the median of the NRMSE, NIR and NSR due to the hyperparameters treatment (see 3.3.2).

3.4.1 Two-dimensional test function

The first example is the Styblinski-Tang function. Both low and high-fidelity objective functions are introduced in Takeno *et al.* [109] and are defined as

$$f_1(\mathbf{s}_{ini}) = \frac{1}{2}\sum_{i=1}^{2}[0.9(s_{ini}^{(i)})^4 - 15(s_{ini}^{(i)})^2 + 6s_{ini}^{(i)}], \tag{3.91}$$

and

$$f_2(\mathbf{s}_{ini}) = \frac{1}{2}\sum_{i=1}^{2}[(s_{ini}^{(i)})^4 - 16(s_{ini}^{(i)})^2 + 5s_{ini}^{(i)}], \tag{3.92}$$

where $\mathbf{s}_{ini} \in [-5, 5]^2$. As in Takeno *et al.* [109], we set the costs of the low and high-fidelity objective function observations to $c_1 = 1$ and $c_2 = 5$. The costs of evaluating gradient information on the low and high-fidelity models are, respectively, $c_\nabla c_1$ and $c_\nabla c_2$. We consider three different cases, $c_\nabla = 0.2, 1, 2$. Two total budgets corresponding to 12 (CCF/$c_2 = 12$) and 24 (CCF/$c_2 = 24$) high-fidelity sameples are used. All the models are initialized with the same budget and the DOE are drawn from a Sobol sequence. For the models with derivative information, the gradients are added at every point in the DOE. For the multi-fidelity models, we test all the combinations of n_1 and n_2 that satisfy the prescribed total budget. However, we do not consider the cases $n_2 = 1$ and $n_2 > n_1$. Note that all the n_2 high-fidelity points are a subset of the n_1 low-fidelity points. The noise variance for the objective function evaluations and its derivatives is set to 10^{-4} .

Modelling

The median NRMSE as a function of n_2 is represented for each case in Fig. 3.4. For all the cases, SF is never the model with the lowest NRMSE.

SFG performs better than SF when $c_\nabla = 0.2$ as shown in Fig. 3.4a and Fig. 3.4b. However, the opposite is observed when $c_\nabla = 2$ (Fig. 3.4e and Fig. 3.4f). When $c_\nabla = 1$, the NRMSE of SFG is lower than the NRMSE of SF with few high-fidelity objective function evaluations (Fig. 3.4c) but higher with a more significant budget (Fig. 3.4d). This means that SFG is more useful for modelling than SF only if enough samples can be provided.

For all the cases, MF has a lower NRMSE than SF. However, some combinations of n_1 and n_2 lead to smaller NRMSE than others for the MF model (e.g. the right column of Fig 3.4). By looking at the coefficients of determination defined in Eq. 3.70 and Eq. 3.71 (not shown here), it is observed that enough low and high-fidelity samples are necessary to have a high accuracy on respectively the low and the error bridge functions. Obviously, for a given budget, these two conditions are competitive as a higher number of low-fidelity samples n_1 implies a lower number of high-fidelity samples n_2 with the consequence of deteriorating the quality of the bridge function model. On the contrary, if there is a higher number of high-fidelity samples, fewer low-fidelity samples can be used and the accuracy of the low-fidelity model decreases.

Except for the case CCF/$c_2 = 12$ and $c_\nabla = 2$ (Fig. 3.4e), we note that MFG$_1$G$_2$ requires fewer high-fidelity samples than MF to decrease the NRMSE, since the approximation of the bridge function is better due to the derivative information

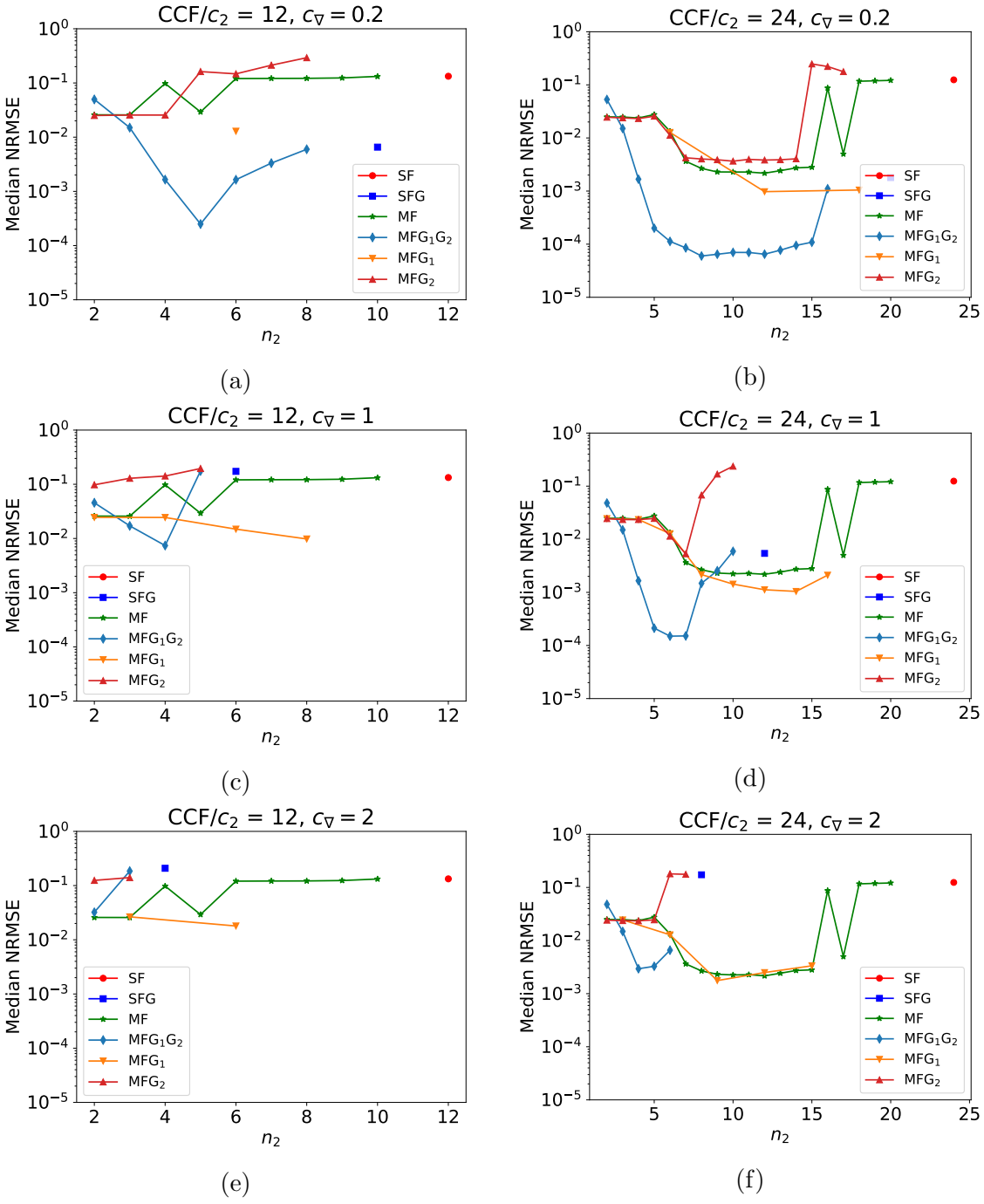


Figure 3.4: Median NRMSE as a function of the number of high-fidelity samples n_2 for all the models considered on the Styblinski-Tang objective function. From top to bottom: $c_\nabla = 0.2$, $c_\nabla = 1$ and $c_\nabla = 2$. Left column: for a total budget of $12c_2$, right column: for a total budget of $24c_2$.

provided (observed with the coefficients of determination). In addition, since the gradient information is also included in the low-fidelity model, the low-fidelity objective function is better modelled with fewer low-fidelity samples than with MF. Thus, for $c_\nabla = 0.2$ and $n_2 \geq 4$ (number of high-fidelity samples to start to approximate correctly the bridge function), the median NRSME obtained with all the possible configurations of n_1 and n_2 with MFG_1G_2 is lower than the best NRMSE possible with the MF model (Fig. 3.4a and Fig. 3.4b). Also, in both cases, the lowest NRMSE obtained with MFG_1G_2 was the best among all the possible models. However, as with the MF model, the NRMSE of the MFG_1G_2 model increases when very few low-fidelity samples are included in the model, since the low-fidelity objective function becomes poorly modelled (as in Fig. 3.4d for $n_2 > 7$). The performance of the model also deteriorates rapidly for a low budget and high gradient cost as fewer low and high-fidelity samples can be included in the model. As an example, when $c_\nabla = 2$ and $\text{CCF}/c_2 = 12$ (Fig. 3.4e), MFG_1G_2 becomes worse than MF since there are not enough samples to accurately approximate the low-fidelity and bridge functions.

The modelling performance of MFG_1 is generally equal or better than that of MF for all the budgets and c_∇ considered. Its performance is generally the same as MF with few high-fidelity samples since the bridge function is poorly approximated. However, with a higher number of high-fidelity samples, its performance does not deteriorate as quickly as MF, since fewer low-fidelity samples are required to approximate the low-fidelity function than with MF due to the gradient information (as in Fig. 3.4b). The modelling performance of MFG_1 can even be increased with a higher number of high-fidelity samples since the bridge function is better modelled and there are enough low-fidelity samples for the low-fidelity function (as in Fig. 3.4c).

Finally, we note that the NRMSE of MFG_2 is generally higher than MF for the same number of high-fidelity samples. Note that by the construction employed for the multi-fidelity model, when the gradient information is only available on the high-fidelity model, it is not employed to build either the low-fidelity model or the bridge function model. Thus, it does not increase the accuracy of these two functions and less low-fidelity samples are employed compared to MF due to the cost of the gradient information.

Note also that the relationship between the two Styblinski-Tang functions is not linear and the multi-fidelity models used struggle more to correctly approximate the high-fidelity objective function.

Initial minimum prediction

The median NIR for the Styblinski-Tang objective function as a function of n_2 is represented in Fig. 3.5. Globally, the NIR follows the same trend as the NRMSE presented in Fig. 3.4. However, we can still observe some differences.

For example, the NIR of the SFG model is lower than the NIR of the SF model for all the gradient costs and budgets considered. Thus, even if for high gradient cost, SFG was not efficient for global modelling purposes as the number of sampled points largely decreased (see Fig. 3.4e and Fig. 3.4f), it can still be useful in local modelling for the determination of the minimum of an objective function. However, except for a small budget of 12 high-fidelity functions and $c_\nabla = 0.2$ (Fig. 3.5a), the median NIR of SFG was higher the one found with the best configuration for MF.

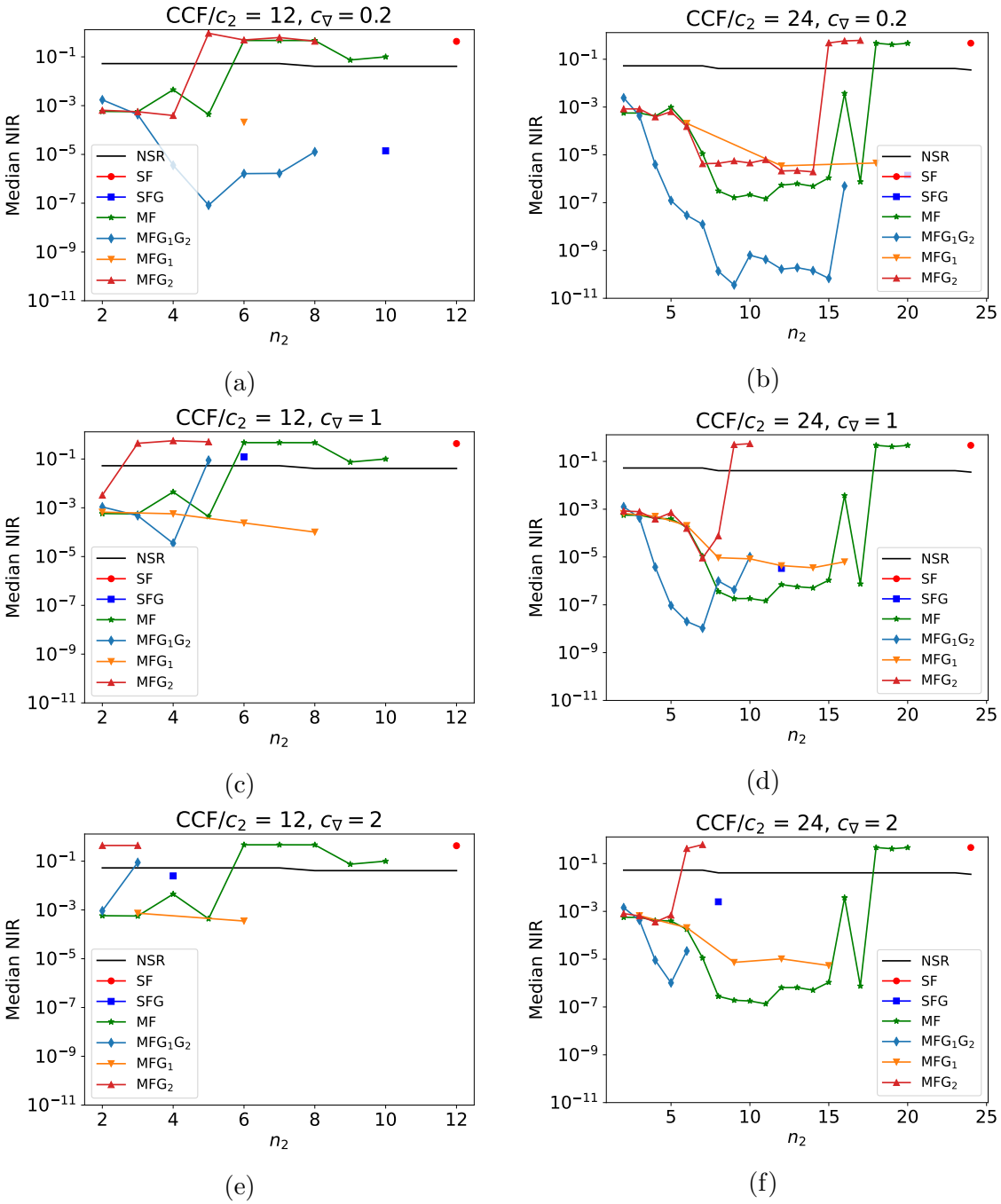


Figure 3.5: Median NIR as a function of the number of high-fidelity samples n_2 for all the models considered on the Styblinski-Tang objective function. From top to bottom: $c_\nabla = 0.2$, $c_\nabla = 1$ and $c_\nabla = 2$. Left: for a total budget of $12c_2$, right: for a total budget of $24c_2$. The black solid line represents the NSR as a function of n_2.

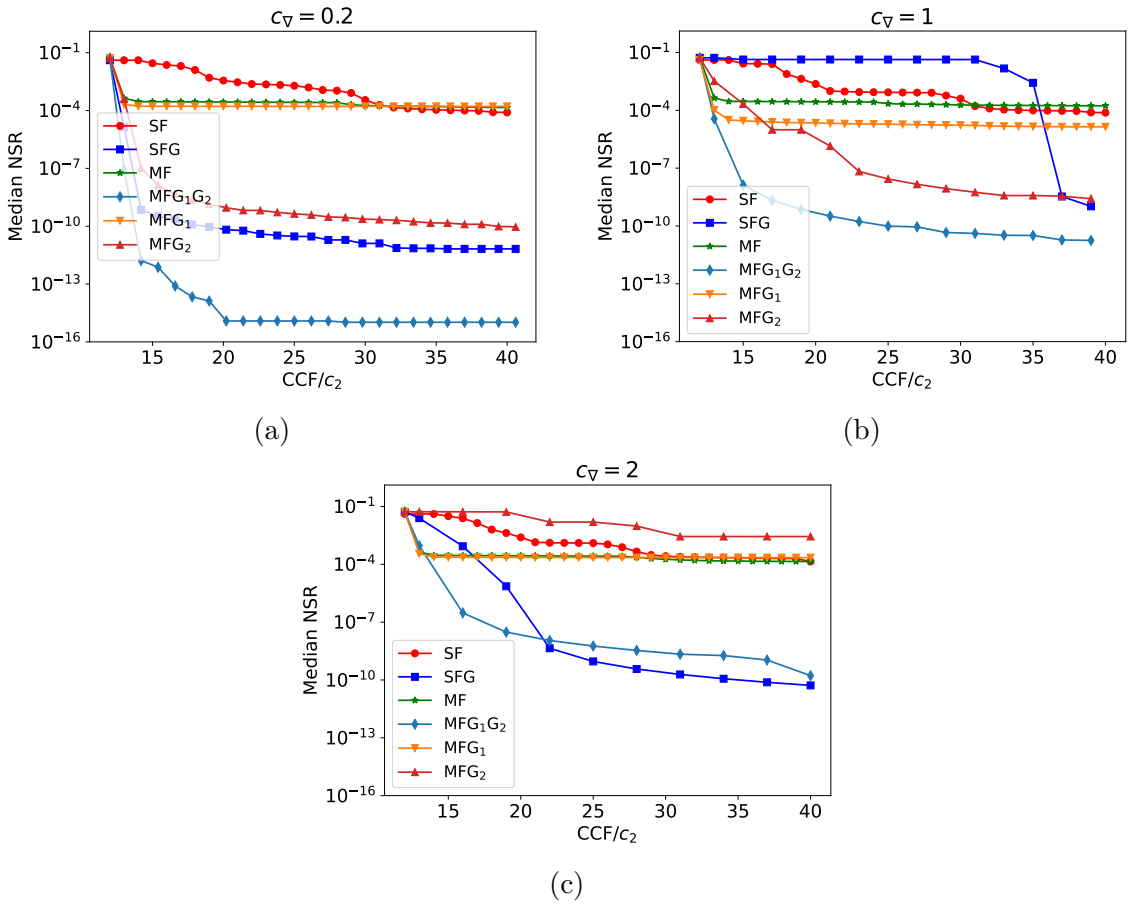


Figure 3.6: Median NSR as a function of CCF/c_2 for all the models considered on the Styblinski-Tang objective function. a) $c_\nabla = 0.2$, b) $c_\nabla = 1$ and c) $c_\nabla = 2$. The CCF and the NSR are updated at each new objective function observation.

Another difference between the NRMSE and the NIR is that for CCF/$c_2 = 24$, the NIR of MFG$_1$ is generally higher than the one found with MF (e.g. Fig. 3.5b) whereas the opposite was observed with the NRMSE (e.g. Fig. 3.7a).

Once again, despite the gradient information added on the high-fidelity objective function, we generally observe that the MFG$_2$ model performs worse in the prediction of the minimum than the MF model, except for few configurations of n_2 and n_1. We also can observe that the best median NIR is obtained with the MFG$_1$G$_2$ model when $c_\nabla = 0.2$ and CCF/c_2 and is equal to 3.60×10^{-11}.

In most cases, the median NIR is below the lowest NSR: even with few high-fidelity samples, the minimum predicted by the models is generally better than the minimum obtained if all the budget was spent in high-fidelity samples. The only exceptions are with the SF model where the median NIR is higher than the NSR for the two budgets considered and for the multi-fidelity models MF, MFG$_2$ and MFG$_1$G$_2$ when very few low-fidelity samples are used (corresponding to high n_2 and/or high c_∇). However, for all the cases, the median NIR of the MFG$_1$ model is lower than the best NSR.

Optimization

We now investigate the performance of the different models in an optimization framework. As before, $c_\nabla = 0.2, 1, 2$. All the models are initialized with $\mathrm{CCF}/c_2 = 12$ objective functions and gradients observations. For the multi-fidelity models, n_1 and n_2 are chosen such it returns the best initial NIR (cf Fig. 3.5a, Fig. 3.5c and Fig. 3.5e). The Algorithm 2 is then applied with $\mathrm{CCF}_{max}/c_2 = 40$.

The median NSR as a function of CCF/c_2 is depicted in Fig. 3.6. When $c_\nabla = 0.2$ (Fig. 3.6a), we can note that including the gradients on the high-fidelity objective function improves the NSR. MF and MFG_1 perform similarly even if the NSR is slightly lower with MFG_1 for low CCF/c_2 than with MF. Finally, SF initially performs the worst but reaches a lower median NSR than MF and MFG_1 at the end of the budget.

When $c_\nabla = 1$, $\mathrm{MFG}_1\mathrm{G}_2$ is the one with the lowest median NSR at each iteration. MFG_1 diminishes faster the NSR than MFG_2 but the latter reaches in the end of the optimization budget a lower NSR than the first one. The MFG_1 and MFG_2 models perform better than the MF model. In that case, SFG is slow to diminish the NSR but reaches a NSR lower than all the models except $\mathrm{MFG}_1\mathrm{G}_2$ at the end of the optimization budget.

However, note that when $c_\nabla = 2$, SFG performs better than when $c_\nabla = 1$, whereas it is initialized with fewer points. At each iteration, the NSR of SFG is lower than the NSR of the SF model and requires $\mathrm{CCF}/c_2 = 19$ to reach a lower NSR than the MF model. It is also the model with the lowest median NSR at the end of the optimization budget. $\mathrm{MFG}_1\mathrm{G}_2$ also performs well and a median NSR lower than 10^{-7} is obtained for $\mathrm{CCF}/c_2 = 19$. In that case however, MFG_2 is the worst model to use. Also, the best initial NIR of MFG_2 was more than 100 times higher than for the MF, MFG_1 and $\mathrm{MFG}_1\mathrm{G}_2$ models (Fig. 3.5e). Again, the MF and MFG_1 models perform in a similar way. We can also note that despite the aggressive optimization strategy used and more high-fidelity samples are added, the MF model did not reach a NSR value lower than the best NIR value predicted for the DOE study with $\mathrm{CCF}/c_2 = 24$ (Fig. 3.5b for example). Thus, a more explorative acquisition strategy and/or adding more low-fidelity sample points during the optimization process could improve the results of the MF model.

3.4.2 Six-dimensional test function

The second example is the Hartmann-6 test function. Various forms of the low and high-fidelity objective functions have been designed. The objective functions defined in Takeno *et al.* [109] are used in this study:

$$f_1(\mathbf{s}_{ini}) = -\sum_{i=1}^{4}(\alpha_i - 0.2)\exp\left(-\sum_{j=1}^{6} A_{ij}(s_{ini}^{(j)} - P_{ij})^2\right), \tag{3.93}$$

$$f_2(\mathbf{s}_{ini}) = -\sum_{i=1}^{4}\alpha_i\exp\left(-\sum_{j=1}^{6} A_{ij}(s_{ini}^{(j)} - P_{ij})^2\right), \tag{3.94}$$

with $\alpha = (1.0, 1.2, 3.0, 3.2)^\mathsf{T}$,

$$\mathbf{A} = \begin{pmatrix} 10 & 3 & 17 & 3.50 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \tag{3.95}$$

$$\mathbf{P} = 10^{-4} \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}, \tag{3.96}$$

and $\mathbf{s}_{ini} \in [0, 1]^6$. Again, we set $c_1 = 1$, $c_2 = 5$ and consider the cases $c_\nabla = 0.2, 1, 2$ for two budgets corresponding to 24 and 48 high-fidelity samples. All the models are initialized with the same budget, the DOE drawn from a Sobol design and the derivative information for the corresponding models are added to every point of the DOE. The noise variance is set to 10^{-6} for the objective functions and derivative observations.

Modelling

The median NRMSE as a function of n_2 is depicted in Fig. 3.7. Instead of a Cartesian grid to compute the NRMSE, we used 10000 points drawn from a Sobol Design. Among all the possibilities, SF is never the best model to use.

In this case, SFG is better than SF for all the cases except for a budget of 24 high-fidelity samples and $c_\nabla = 2$ (Fig. 3.7e). For this case, the number of samples (8) is too low to get an approximation of quality of the high-fidelity objective function with SFG.

For a low number of high-fidelity samples, MF performs better than SF and SFG for all the budgets and gradient costs considered. However, as the number of high-fidelity samples increases, its performance deteriorates as the low-fidelity objective function is no longer well approximated (observed with the coefficients of determination not shown here).

When $c_\nabla = 0.2$ and for the same number of high-fidelity samples, MFG_1G_2 has a lower NRMSE than MF for the two budgets considered (Fig. 3.7a and Fig. 3.7b). Indeed, due to the gradient information provided, both the low-fidelity and bridge functions are better approximated with MFG_1G_2 than with MF. When $c_\nabla = 1$ (Fig. 3.7c and Fig. 3.7d), for a low number of high-fidelity samples, MFG_1G_2 still performs better than MF. However, its efficiency decreases as n_2 increases since fewer low-fidelity samples are evaluated, reducing then the accuracy of the low-fidelity objective function. The worse performance of MFG_1G_2 over MF becomes more obvious when $c_\nabla = 2$ (Fig. 3.7e and Fig. 3.7f).

Regarding MFG_1, this model shows again an overall better efficiency than MF for all the cases considered. For this test function and for the same number of high-fidelity samples, MFG_1 has also an overall better performance than MFG_1G_2.

Finally, MF is again more efficient than MFG_2 for modelling purposes than MF.

As a final note on this test function, all the MF models fail to completely describe the low-fidelity objective function or the bridge function as maximum values of $R_1 \approx 0.97$ and $R_{err} \approx 0.71$ are found in the best cases. Thus, in order to approximate accurately the high-fidelity objective function, a higher budget or a different DOE would be required.

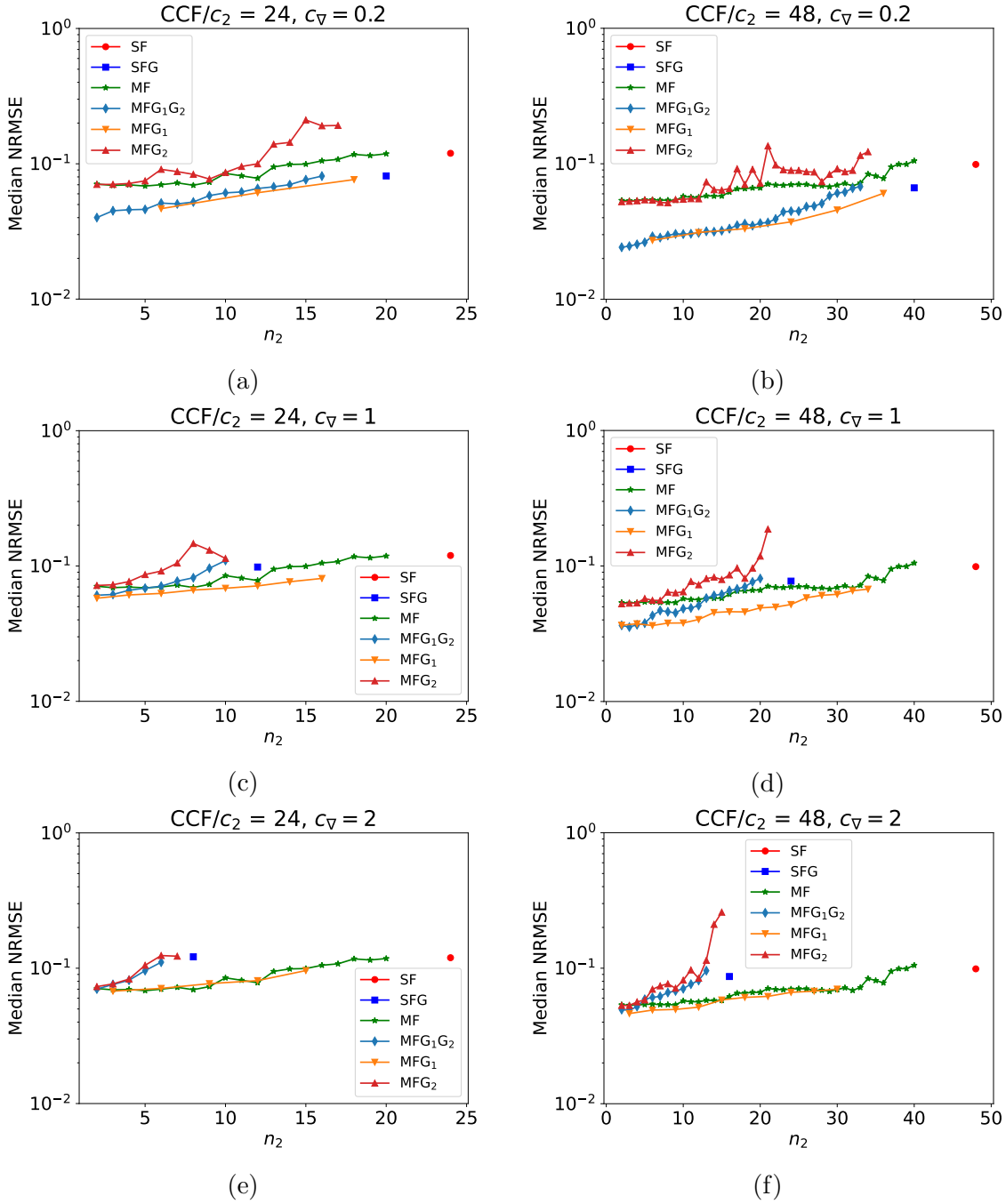


Figure 3.7: Median NRMSE as a function of the number of high-fidelity samples n_2 for all the models considered on the Hartmann-6 objective function. From top to bottom: $c_\nabla = 0.2$, $c_\nabla = 1$ and $c_\nabla = 2$. Left: for a total budget of $24c_2$, right: for a total budget of $48c_2$.

Initial minimum prediction

The median NIR as a function of n_2 for the Hartmann-6 objective function is depicted in Fig. 3.8. Once again, as with the Styblinski-Tang objective function, the median NIR of the SFG model is lower than the one found with SF for all the cases considered.

When the number of high-fidelity samples n_2 is sufficient (as in Fig. 3.8a, Fig. 3.8b, Fig. 3.8d and Fig. 3.8f), SFG performs better than MF for all possible configurations of n_1 and n_2. However, when too few high-fidelity objective functions are used for SFG (Fig. 3.8c and Fig. 3.8e), there exists some n_1 and n_2 such that the median NIR of the MF model is lower than the median NIR of SFG.

Also, MFG_1G_2 performs generally better in the minimum prediction than MF for low n_2, even with a limited budget and computationally expensive gradient observations (as in Fig. 3.8c and Fig. 3.8e). However, when very few low-fidelity samples n_1 are added to the model, for the same number of high-fidelity samples n_2, the median NIR of MF becomes lower than the one of MFG_1G_2.

In this case, the model MFG_1 exhibits a lower median NIR than the MF model for all the gradients costs, budget and number of high-fidelity samples considered.

Once again, we do not observe a major improvement of the MFG_2 model over the MF model for the minimum prediction.

Again, we note that the NIR is higher than the NSR with the SF model, with the multi-fidelity models when very few low-fidelity samples are added or with the SFG model when n_2 is low. Conversely, in most cases, the MFG_1 and MFG_1G_2 models have a lower NIR than the NSR.

Optimization

The performance of all the models is again investigated in an optimization framework for $c_\nabla = 0.2, 1, 2$ for the Hartmann-6 objective function. All the models are initialized with the same initial computational cost factor $\text{CCF}/c_2 = 24$. For the multi-fidelity models, n_1 and n_2 are chosen such it has the lowest initial NIR (cf Fig. 3.8a, Fig. 3.8c and Fig. 3.8e). The Algorithm 2 is then applied for $\text{CCF}_{max}/c_2 = 60$.

The median NSR as a function of CCF/c_2 is displayed in Fig. 3.9. As can be seen for all the gradient costs considered, MFG_1G_2 and SFG are the models that reach the lowest median NSR once the computational budget is consumed. For all the gradient costs considered, both of these models reach final NSR values below 2×10^{-10}. The best median NSR value is found with MFG_1G_2 when $c_\nabla = 0.2$ and is equal to 1.16×10^{-11}. As the gradient cost increases, we note however, that MFG_1G_2 and MFG_1 diminish the NSR faster than SFG for low CCF/c_2 (as in Fig. 3.9c). MFG_1 performs better than both MF and MFG_2 for all the gradient costs considered. Indeed, at each iteration, the NSR of MFG_1 is lower than the ones obtained with MF and MFG_2. Note also, that the NSR of the SF model is slower to decrease but reaches lower NSR values than MF and MFG_2 at the end of the optimization budget for all the gradient costs considered. The SF model also obtains a better final minimum than MFG_1 when $c_\nabla = 2$.

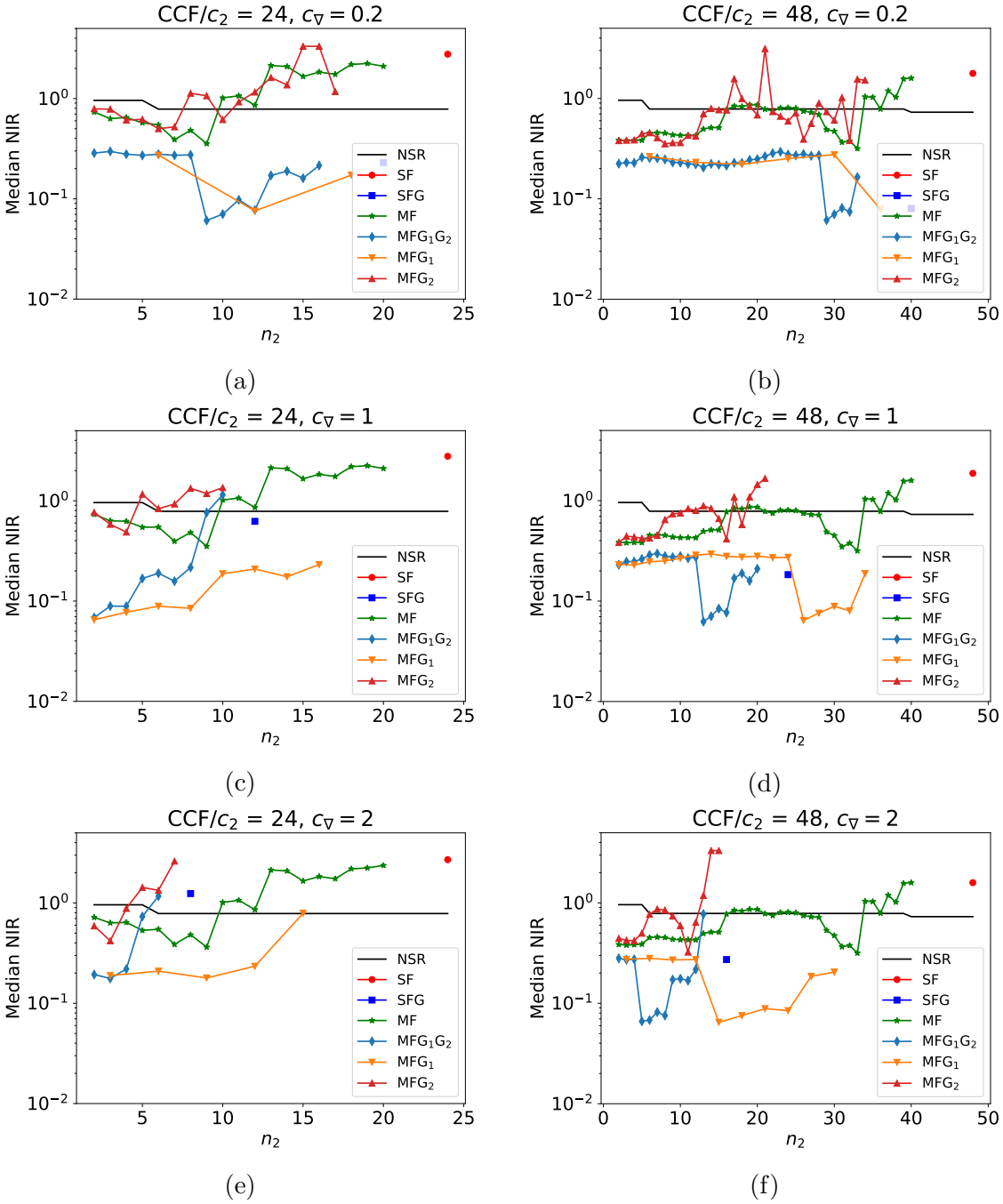
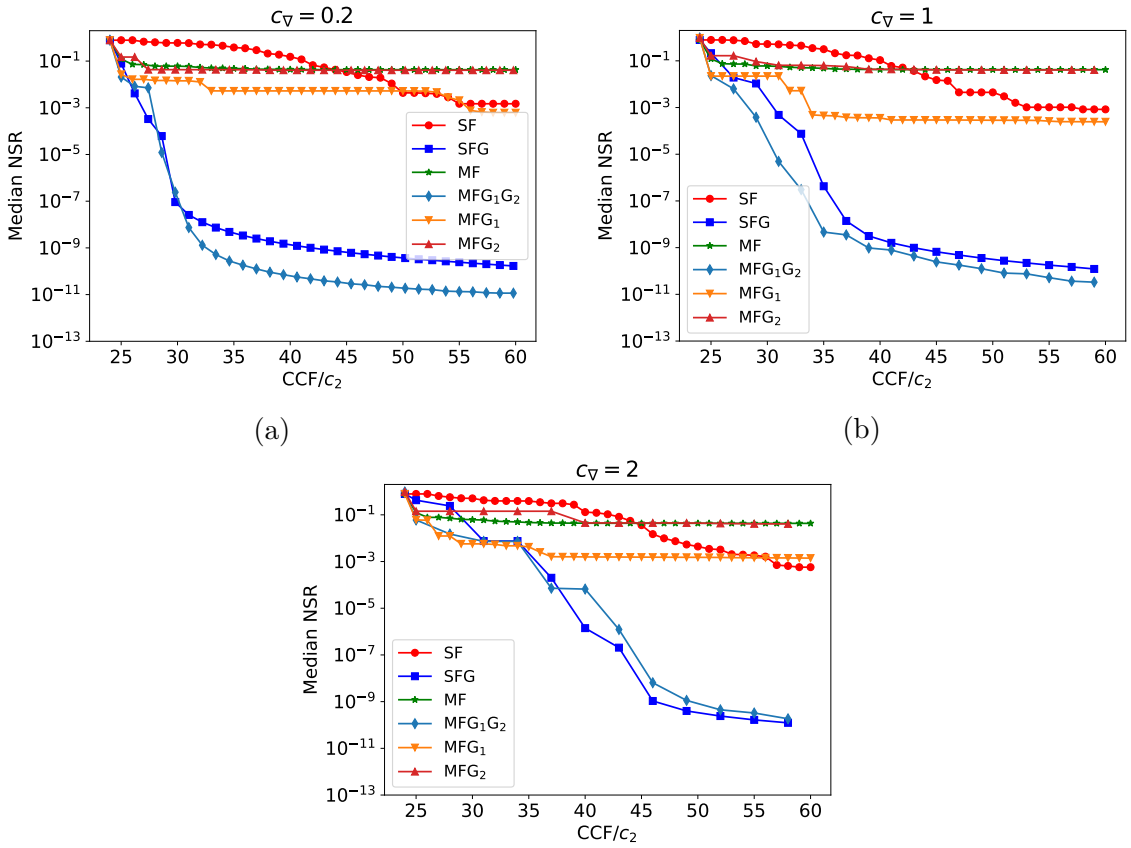


Figure 3.8: Median NIR as a function of the number of high-fidelity samples n_2 for all the models considered on the Hartmann-6 objective function. From top to bottom: $c_\nabla = 0.2$, $c_\nabla = 1$ and $c_\nabla = 2$. Left: for a total budget of $24c_2$, right: for a total budget of $48c_2$. The black solid line represents the NSR as a function of n_2.



(a) (b)

(c)

Figure 3.9: Median NSR as a function of CCF/c_2 for all the models considered on the Hartmann-6 objective function. a) $c_\nabla = 0.2$, b) $c_\nabla = 1$ and c) $c_\nabla = 2$. The CCF and the NSR are updated at each new objective function observation.

	St	$\overline{C_\mathrm{d}}$	C_l
Ref. [4]	0.193	1.19 ± 0.042	± 0.64
Ref. [69]	0.192	1.31 ± 0.049	± 0.69
Ref. [56]	0.190	-	-
Ref. [98]	0.19	-	-
Ref. [67]	0.197	1.34 ± 0.044	± 0.69
Ref. [108]	0.196	1.35 ± 0.048	± 0.68
Present study	**0.195**	**1.33**	$\mathbf{\pm 0.67}$

Table 3.1: Comparison with the literature of the Strouhal number St, the time averaged drag coefficient $\overline{C_\mathrm{d}}$ and the maximum and minimum of the lift coefficient C_l at $\mathrm{Re} = 200$.

3.4.3 Cylinder drag at $\mathrm{Re} = 200$

Problem description

We now consider the numerical simulation of an external flow around a two-dimensional cylinder at $\mathrm{Re} = 200$. 96 equally-spaced Lagrangian points are set on a cylinder of diameter $D = 1$. Once the transients have vanished, at each of these points j, we prescribe a tangential velocity

$$\frac{v_{\theta,j}(\mathbf{s}_{ini})}{U_\infty} = s_{ini}^{(1)} \sum_{k=-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{\theta_j - s_{ini}^{(2)} - 2\pi k}{s_{ini}^{(3)}}\right)^2}, \tag{3.97}$$

where U_∞ is the free stream velocity, and θ_j is the angular position from the aft of the cylinder of the point j. We define as boundaries, $s_{ini}^{(1)} \in [-1, 1]$, $s_{ini}^{(2)} \in [-\pi, \pi]$ and $s_{ini}^{(3)} \in [0.1, \pi/4]$.

The simulation is then run during an additional time $\Delta T_1 = 2.5D/U_\infty$ or $\Delta T_2 = 10D/U_\infty$, respectively, for the low and high-fidelity models. The following low and high-fidelity functions are then defined

$$f_1(\mathbf{s}_{ini}) = \frac{1}{\Delta T_1}\int_0^{\Delta T_1} C_\mathrm{d}^2(t; \mathbf{s}_{ini})\,\mathrm{d}t + \frac{\alpha}{96}\sum_{j=1}^{96} v_{\theta,j}^2(\mathbf{s}_{ini}), \tag{3.98}$$

$$f_2(\mathbf{s}_{ini}) = \frac{1}{\Delta T_2}\int_0^{\Delta T_2} C_\mathrm{d}^2(t; \mathbf{s}_{ini})\,\mathrm{d}t + \frac{\alpha}{96}\sum_{j=1}^{96} v_{\theta,j}^2(\mathbf{s}_{ini}), \tag{3.99}$$

where α is a penalty term set here to 2, and $C_\mathrm{d}(t; \mathbf{s}_{ini}) = 2f_x/(\rho_\infty U_\infty^2 D)$ is the drag coefficient with ρ_∞ being the free stream density, and f_x is the force exerted on the cylinder in the streamwise direction.

Numerical set-up

The domain is rectangular and is included in $[-15.46, 41.11]$ in the streamwise direction x and $[-28.29, 28.29]$ in the cross-flow direction y. 384 cells were used both in the streamwise and cross-flow directions. The mesh size around the cylinder is $\Delta x = 0.033$ whereas far away, we have $\Delta x = 0.20$. The cylinder is composed of

96 Lagrangian markers corresponding to the points aforementioned and is centered at $(0, 0)$. A first simulation without any control is run for $t = 100D/U_\infty$ with the in-house software IBMOS [22].

The Strouhal number St, the time averaged drag coefficient $\overline{C_\mathrm{d}}$ and the maximum and minimum of the lift coefficient C_l are compared with the literature in Table 3.1. These quantities are calculated from $t = 50D/U_\infty$ to $t = 100D/U_\infty$. We can observe that the present case is in good agreement with previous studies.

From the snapshot obtained at $t = 100D/U_\infty$, the velocity prescribed in Eq. 3.97 is applied and the simulation is run during ΔT_1 or ΔT_2 accordingly. Due to the values of ΔT_1 and ΔT_2 proposed, we set $c_1 = 1$ and $c_2 = 4$. We normally have $c_\nabla \approx 1$ but in order to study the influence of the cost of gradient evaluations, we consider $c_\nabla = 0.2, 1, 2$. As before, two budgets of 18 and 24 high-fidelity evaluations are considered. The noise variance is set to 10^{-6}.

Modelling

The median NRMSE of the different models for the cylinder at $\mathrm{Re} = 200$ is displayed in Fig. 3.10. 10000 points drawn from a Sobol design were used for the computation of the NRMSE.

When $c_\nabla = 0.2$ (Fig. 3.10a and Fig. 3.10b) or $c_\nabla = 1$ (Fig. 3.10c and Fig. 3.10d), SFG performs better than SF.

MF also performs generally better than SF for the two budgets considered. MF is only less efficient than SF for a high n_2 since the low-fidelity function is poorly modelled.

$\mathrm{MFG_1G_2}$ is generally more accurate than MF when $c_\nabla = 0.2$ (Fig. 3.10a and Fig. 3.10b). However, when $c_\nabla \geq 1$, (Fig. 3.10c and Fig. 3.10d), the NRMSE of $\mathrm{MFG_1G_2}$ becomes lower than the NRMSE of MF for most cases due to an insufficient number of samples (Fig. 3.10c, Fig. 3.10d, Fig. 3.10e and Fig. 3.10f).

When $c_\nabla = 0.2$ (Fig. 3.10a and Fig. 3.10b) or $c_\nabla = 1$ (Fig. 3.10c and Fig. 3.10d), $\mathrm{MFG_1}$ is always better than MF for the same number of high-fidelity samples. However, when $c_\nabla = 2$ (Fig. 3.10c and Fig. 3.10d), the modelling performance of $\mathrm{MFG_1}$ starts to deteriorate and reaches similar NRMSE values as the MF model since very few low-fidelity samples are included in the model.

Once again, it is observed that in most cases $\mathrm{MFG_2}$ performs worse than MF.

Initial minimum prediction

The median NIR for the cylinder problem is depicted in Fig. 3.11.

When $c_\nabla = 0.2$ (Fig. 3.11a and Fig. 3.11b), or $c_\nabla = 1$, (Fig. 3.11c and Fig. 3.11d), SFG has a lower NIR than SF. For a low budget of 18 high-fidelity samples and $c_\nabla = 2$ (Fig. 3.11e) the NIR of the SFG model is however slightly higher than the SF model which is not the case with a higher budget (Fig. 3.11f).

For a low budget of 18 high-fidelity samples, the NIR of SFG is also lower than MF for every possible configuration of n_2 and n_1 when $c_\nabla = 0.2$ (Fig. 3.11a and $c_\nabla = 1$ Fig. 3.11c). For a higher budget of 24 high-fidelity samples (Fig. 3.11b and Fig. 3.11d), some sampling schemes of n_1 and n_2 can lead to a better NIR with MF than with SFG.

We also observe that, for the same number of high-fidelity samples, the $\mathrm{MFG_1G_2}$ model has a lower NIR than the MF model in all the cases. Also, compared to MF,

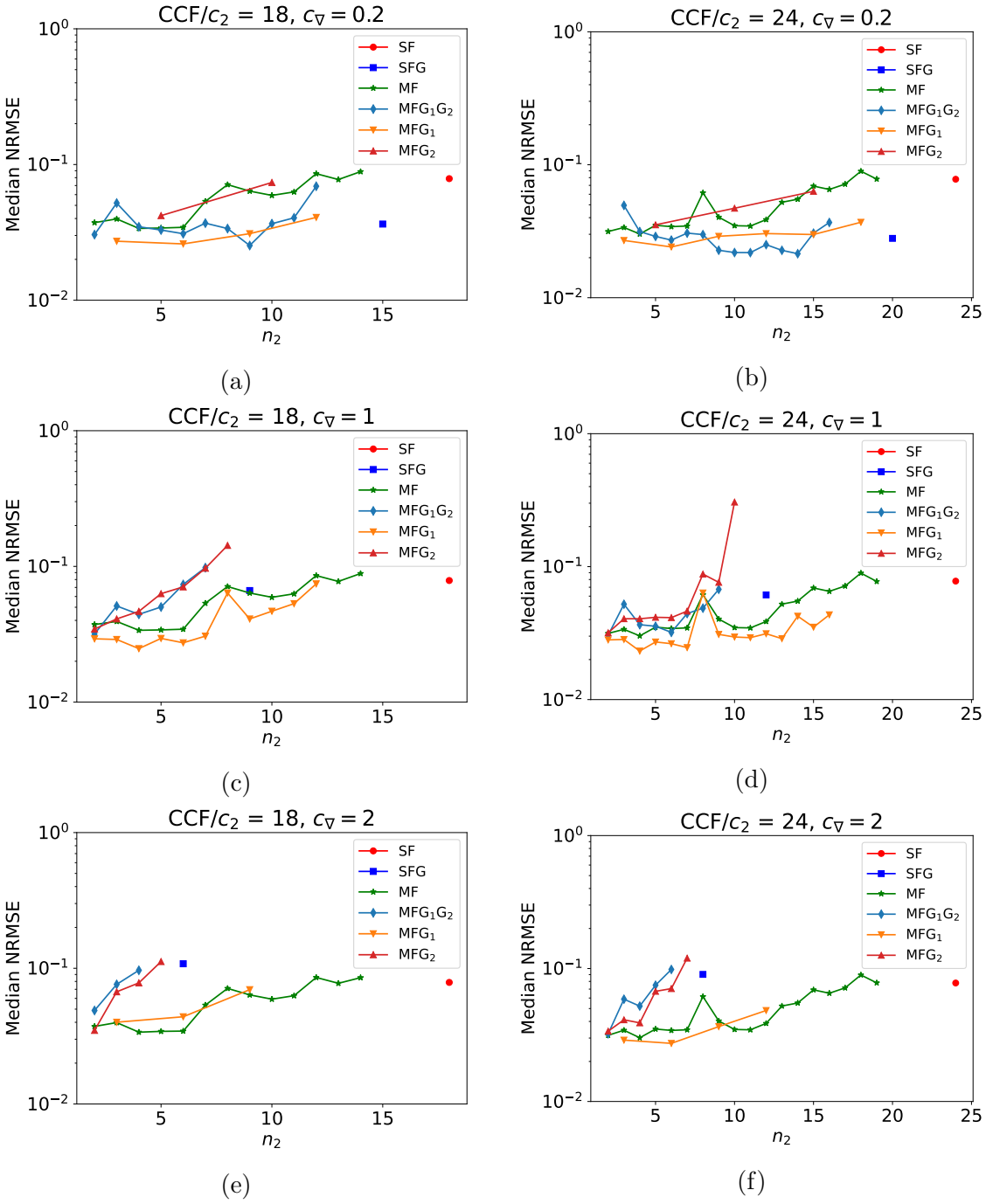


Figure 3.10: Median NRMSE as a function of the number of high-fidelity samples n_2 for all the models considered on the cylinder problem at Re = 200. From top to bottom: $c_\nabla = 0.2$, $c_\nabla = 1$ and $c_\nabla = 2$. Left: for a total budget of $18c_2$, right: for a total budget of $24c_2$.

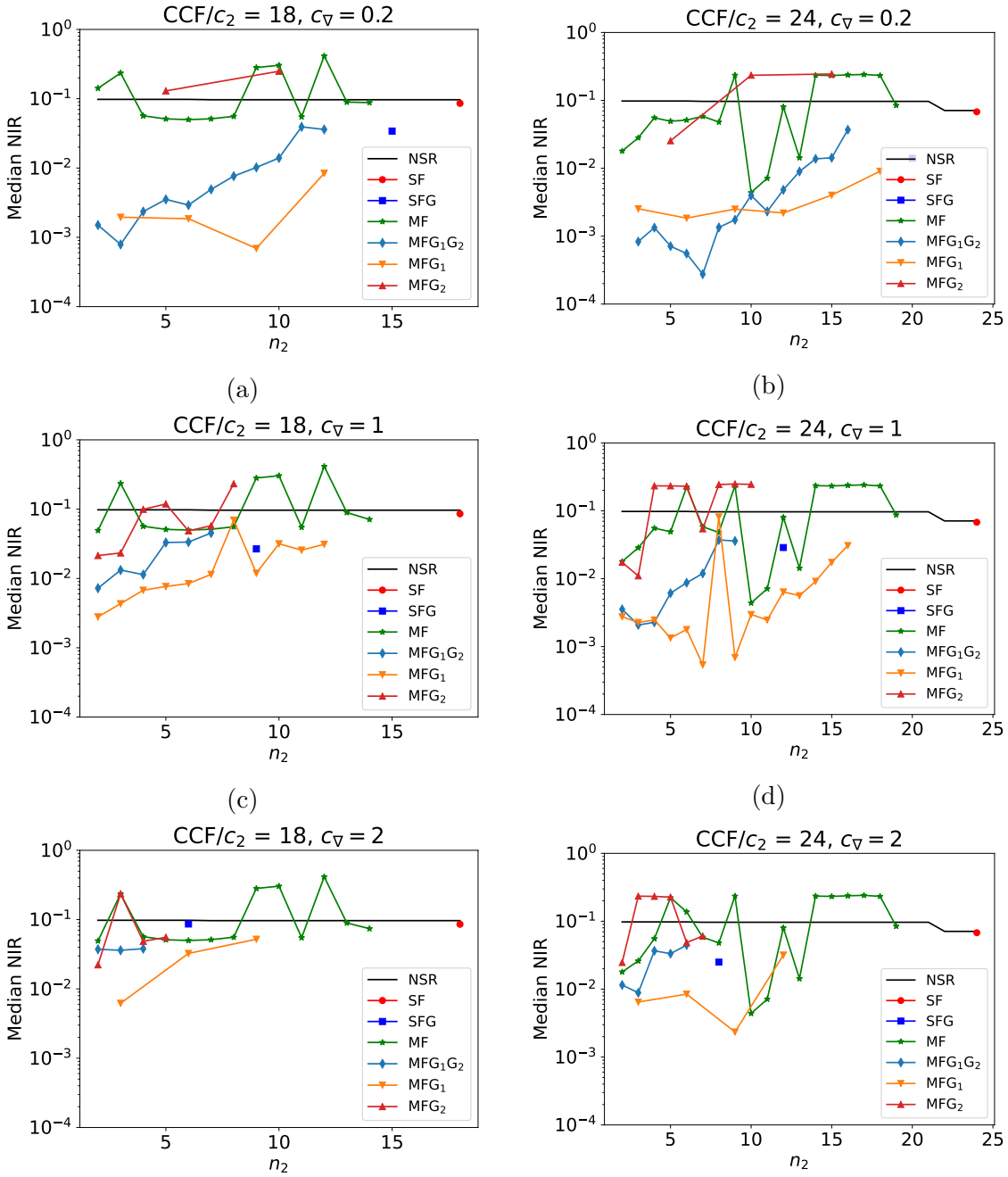


Figure 3.11: Median NIR as a function of the number of high-fidelity samples n_2 for all the models considered on the cylinder problem at Re = 200. From top to bottom: $c_\nabla = 0.2$, $c_\nabla = 1$ and $c_\nabla = 2$. Left: for a total budget of $18c_2$, right: for a total budget of $24c_2$. The black solid line represents the NSR as a function of n_2.

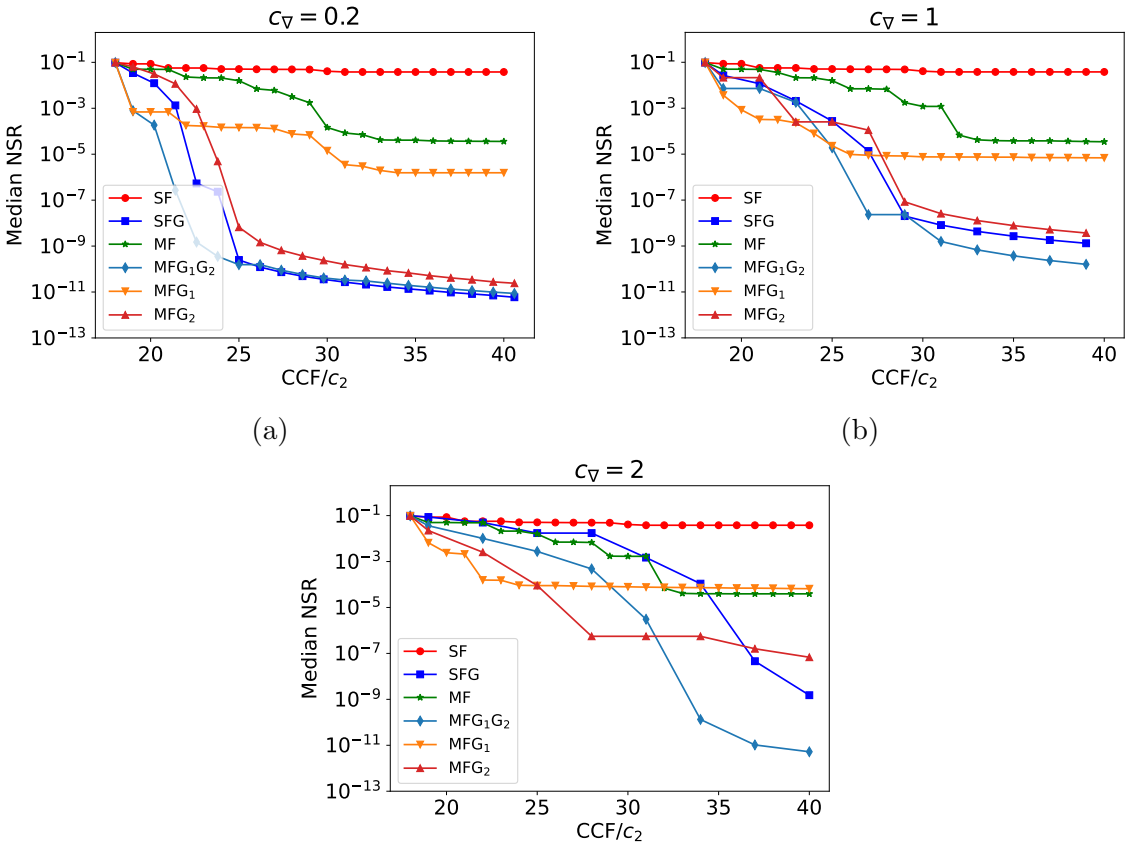


Figure 3.12: Median NSR and standard deviation as a function of CCF/c_2 for all the models considered on the cylinder problem at Re = 200. a) $c_\nabla = 0.2$, b) $c_\nabla = 1$ and c) $c_\nabla = 2$. The CCF and the NSR are updated at each new objective function observation.

the MFG$_1$G$_2$ model reached the lowest NIR for all the cases except for a budget of 24 high-fidelity samples and $c_\nabla = 2$ (Fig. 3.11f).

Generally, the NIR found with the MFG$_1$ model is lower than the one found with the MF model for the same number of high-fidelity samples n_2. There is only one exception when $c_\nabla = 1$ and $n_2 = 8$ (Fig. 3.11c and Fig. 3.11d). The best NIR found with the MFG$_1$ model was also lower than with the MF model or the SFG model for all the cases, and than with the MFG$_1$G$_2$ model when $c_\nabla = 1$ (Fig. 3.11c and Fig. 3.11d) and $c_\nabla = 2$ (Fig. 3.11e and Fig. 3.11f).

Finally, we observe that for a low number of high-fidelity samples n_2, the MFG$_2$ model can sometimes perform better than the MF model as in Fig. 3.11c but should not be established as a general conclusion.

In that case, the median NIR of the SF, SFG, MFG$_1$G$_2$ and MFG$_1$ models were in every case and for all the configurations of n_1 and n_2 lower than the NSR. Again, the minimum predicted with the MF or MFG$_2$ models were generally worse than the minimum of the DOE when n_1 was low.

Optimization

The performance of the models is again investigated in the optimization framework described in Algorithm 2 for $c_\nabla = 0.2, 1, 2$. The models are initialized with a computational cost factor $\mathrm{CCF}/c_2 = 18$. n_1 and n_2 are chosen such that all the models have the lowest initial NIR (cf Fig. 3.11a, Fig. 3.11c and Fig. 3.11e). The optimization algorithm is run for $\mathrm{CCF}_{max}/c_2 = 40$.

The median NSR as a function of CCF/c_2 is represented in Fig. 3.12. For this test case, the models with the gradient information on the high-fidelity objective function (SFG, MFG_2 and $\mathrm{MFG}_1\mathrm{G}_2$) are the ones with the lowest NSR at the end of the budget. However, MFG_1 is generally the faster to decrease the NSR and to reach NSR values lower than 10^{-3}. All the multi-fidelity models with gradients ($\mathrm{MFG}_1\mathrm{G}_2$, MFG_1 and MFG_2) perform generally better than the MF model, the exception being the end of the optimization process for MFG_1 and $c_\nabla = 2$. Finally, note that the median NSR of the SF model decreases extremely slowly.

Flow fields

The optimal tangential velocity profile around the cylinder is represented in Fig. 3.13a. It corresponds to the optimal design $\mathbf{s}^* \approx (1, 0.33, 0.42)^\mathsf{T}$ equivalent to $\mathbf{s}^*_{ini} \approx (1, 5.19, 0.39)^\mathsf{T}$. Thus, the maximal amplitude of the actuator is located at the bottom back part of the cylinder. The drag coefficient as a function of time for the uncontrolled case and optimal solution is depicted in Fig. 3.13b. From $tU_\infty/D = 50$ to $tU_\infty/D = 100$, we obtain a time averaged drag coefficient of $\overline{C_\mathrm{d}} = 0.97$ for the optimal solution, equivalent to a 27% drag reduction over the uncontrolled case. Note also that even if they are not shown here, the lift fluctuations are also reduced as they are equal to ± 0.20 around the time averaged lift coefficient $\overline{C_\mathrm{l}} = -0.62$ (also averaged from $tU_\infty/D = 50$ to $tU_\infty/D = 100$) whereas for the uncontrolled case these fluctuations are ± 0.67. The Strouhal numbers remain relatively close however, as we obtain $\mathrm{St} = 0.205$ for the optimal solution against $\mathrm{St} = 0.195$ for the uncontrolled case.

The averaged streamwise velocity with the streamlines is displayed in Fig. 3.13c for the uncontrolled case and Fig. 3.13d for the optimal solution. A snapshot was stored each $0.24tU_\infty/D$ time units during $100tU_\infty/D$ time units. The averaged streamwise velocity was derived from the instantaneous snapshots obtained from $tU_\infty/D = 48$ to $tU_\infty/D = 99.84$. Note that with optimal solution, the recirculation zone is increased and the wake amplitude reduced but slightly shifted upwards. The actuation set does not enable to suppress vortices but delay further downstream their detachments. Finally, the velocity profiles v_x/U_∞ and v_y/U_∞ as a function of y/D at $x/D = 1.73$ are respectively represented in Fig. 3.13e and Fig. 3.13f. At this location, the streamwise velocity v_x/U_∞ is positive for each y/D for the uncontrolled case whereas the minimum of v_x/U_∞ for the optimal solution is negative, confirming the recirculation zone. Also compared to the uncontrolled case, the cross-flow velocity v_y/U_∞ amplitudes are reduced at $y/D = \pm 0.5$. Note that due to the asymmetric actuation, assymetric velocity profiles are obtained in Fig. 3.13e and Fig. 3.13f for the optimal solution. Indeed, the maximal amplitudes of v_x/U_∞ and v_y/U_∞ are higher when $y/D < 0$ than when $y/D > 0$ due to the actuation set on the bottom back part of the cylinder. Also, the minimum of v_x/U_∞ and v_y/U_∞ are obtained when y/D is slighly greater than 0, confirming that the wake is shifted upwards.

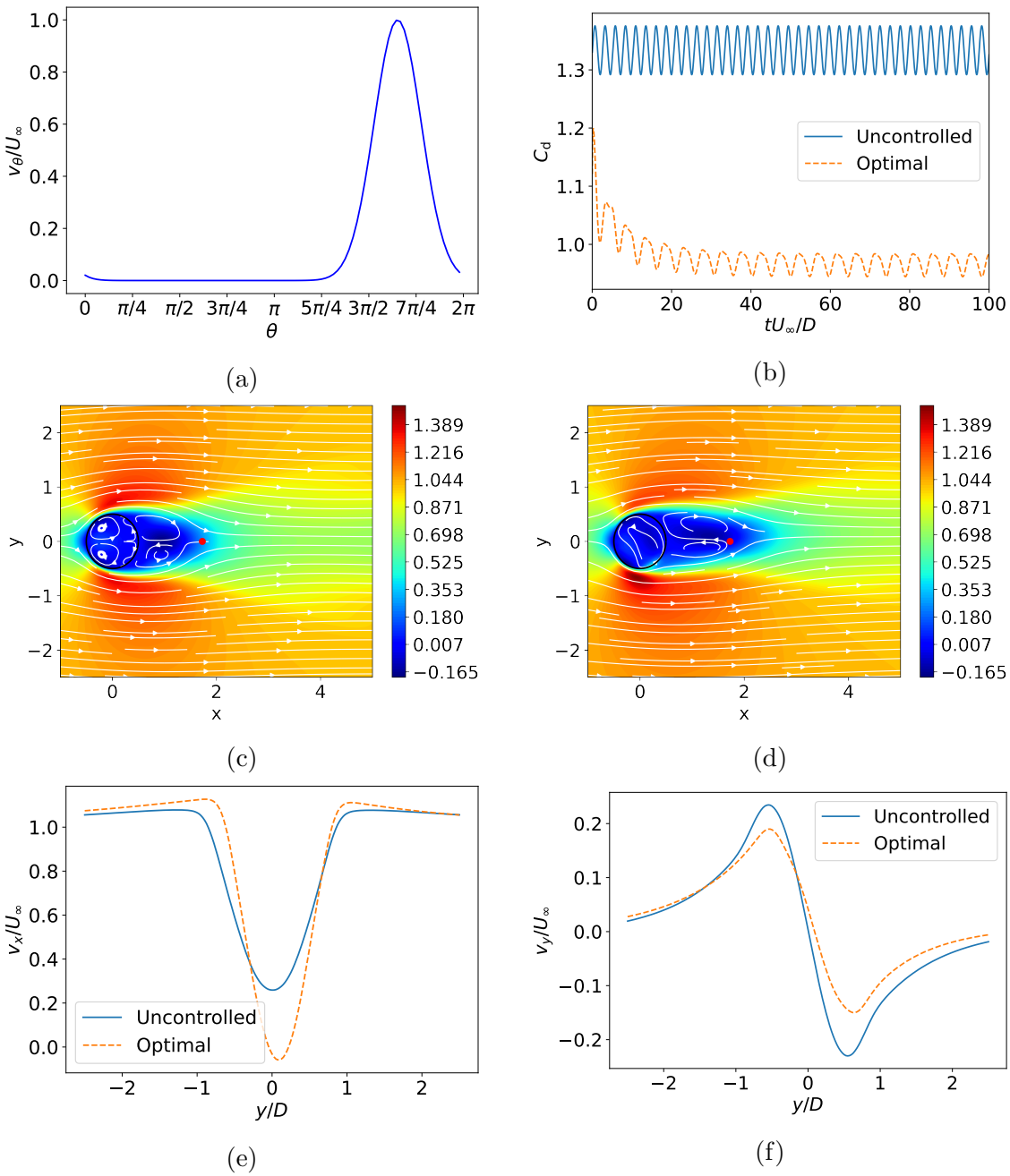


Figure 3.13: a) Optimal velocity profile around the cylinder. b) Drag coefficient as a function of time for the uncontrolled case and optimal solution. Averaged streamwise velocity with streamlines (white lines and arrows) for c) the uncontrolled case and d) the optimal solution. For the uncontrolled case and optimal solution, as a function of the cross flow direction y/D: e) v_x/U_∞ and f) v_y/U_∞. These two last quantities are taken at $x/D = 1.73$. As an illustration of this position, the red dot in c) and d) is located at $(x/D, y/D) = (1.73, 0)$.

3.5 Conclusions

In this work, we compared the different possible models with Gaussian Processes when gradient information and a lower fidelity objective function are available. The global models accuracies as well as their minimum prediction were investigated on three different objective functions of various dimensions for different budgets, gradient costs and for the multi-fidelity models, the sample ratio between the low and high-fidelity objective functions.

Below we summarize the main observations on the test cases presented in this chapter:

- Among all the possible models and design of experiments initializations for the multi-fidelity models, SF was never the best model to use whether for modelling or optimization. However, when very few low-fidelity samples are added in MF, SF can outperform the latter for modelling. Similarly, when the gradient cost was high and for a small total budget, SF was more accurate for global modelling than SFG since too few samples were possible with the latter model. Also, since SF is generally the one with the least information provided, the optimization of the hyperparameters is computationally cheaper than for the other models. Thus, it is generally the fastest model to build.

- Including the derivative information in the Gaussian Process was efficient for modelling when the cost of obtaining the gradient was low or for a large computational budget. SFG appears being more useful for optimization since it was able to find a better minimum than SF even when the cost of evaluating the gradient was equal to twice the objective function cost.

- Except for some sample ratios between the low and high-fidelity objective functions, MF generally showed better results than SF for both modelling and optimization purposes. Also, this model showed generally better results than SFG for modelling purposes when the gradient cost was higher or equal to the cost of evaluating the high-fidelity objective function. However, the difference in computational cost between the low and high-fidelity function was much higher in that case than the difference in cost between the objective function observation and gradient evaluation. When the cost of evaluating the gradient was the same as evaluating the low-fidelity sample, the results were approximately similar for global modelling. Note also, that MF depends on the precision of the low-fidelity model and the bridge function. This means that it requires a moderate number of low-fidelity samples to approximate correctly the low-fidelity objective function, and also a reasonable number of high-fidelity objective functions to approximate correctly the bridge function. Still, the number of low-fidelity samples must be higher than the number of the high-fidelity samples.

- Adding derivative information to both fidelity in the multi-fidelity setting gave generally the best results for both modelling and optimization purposes when the computational cost associated with the gradient was negligible. However, for global modelling purposes, the precision of $\mathrm{MFG_1G_2}$ rapidly decreases as the cost of evaluating the gradient cost is increased. Still, for optimization

purposes, even when the gradient cost was twice the cost of evaluating the objective function, this model gave generally better results.

- Including the gradients only to the low-fidelity model in the multi-fidelity setting showed promising results for both modelling and the minimum prediction. Indeed, the NRMSE of MFG_1 was often lower than for MF for the same number of high-fidelity samples. Also, MFG_1 required a lower CCF than all the other models to reach a NSR value below 10^{-3} during the optimization of the cylinder problem. This model appears particularly interesting in the context of RANS/LES simulations where the gradients can easily be obtained for RANS simulations but are unavailable in LES.

- For a given budget, we generally observed no real interest in adding the gradient information only on the high-fidelity objective function in the multi-fidelity setting. Indeed, the global modelling accuracy was generally worse than for MF. MFG_2 gave clearly better results than the MF for modelling when the gradient cost was negligible. However, compared to MF, the results were generally improved for optimization purposes with MFG_2. Still, compared to SFG, the results were generally worse for the optimization. Also, note that compared to the other multi-fidelity models with gradient information, and with the multi-fidelity formulation used in this chapter, the gradient information of MFG_2 is not included in the hyperparameters optimization. Indeed, the gradient information is just added in the final covariance matrix. Thus, its accuracy heavily depends on the accuracy of MF. This model may prove to be more useful if the gradient information could be included in the optimization of the hyperparameters.

- Generally speaking, the gradient information on the high-fidelity function was useful for modelling when its cost was low or for a high budget available. Including the gradient information on the low-fidelity model enabled to get a better initial modelling and provided better initial minimum predictions. However, within an optimization framework, including the gradients on the high-fidelity function gave the most accurate results at the end of the optimization algorithm.

These conclusions are valid for the three objective functions studied in this chapter. Thus, additional experiments with different objective functions would be needed to confirm our findings. Nonetheless, even with different objective functions, we expect similar conclusions.

As a final note, in this study the cost of building the model has been neglected as we considered the cost of evaluating the objective functions and gradients significantly higher. However, the cost of building a Gaussian Process grows cubically with the number of observations provided [57]. This rapidly makes the models with gradient information impractical for problems where the cost of observing the objective function is moderate or in high dimensions. Improving the efficiency of building single and multi-fidelity Gaussian Process with derivative information can lead to important gains in this regard. Among the possibilities, we can cite adding the gradients only at certain points as in Yamazaki and Mavriplis[123], providing to the Gaussian Process the derivative information only on some design variables as in Wu

et al. [122], or decomposing the model into a series of submodels with smaller correlation and weighing them in order to decrease the cost associated with the initial large correlation matrix as in Han *et al.* [32].

Another improvement can also be to use non-linear multi-fidelity models such as the ones developed in [91]. Indeed, the performance of the linear multi-fidelity model decreased when the relationship between the high and low-fidelity model was not linear. Finally, the last improvement that we propose is to develop acquisition functions for optimization and/or modelling purposes that can take into account the cost of the various sources of information. Arguably, the most adequate acquisition functions are the ones relying on the Value Of Information (VOI). These acquisition functions aim to sample at design points that bring the more information about the objective function or its minimum. For optimization purposes, the most promising ones are probably the ones relying on the Max-Value Entropy Search [117, 109] and the Knowledge Gradient class of acquisition functions [24, 122, 93]. The first ones are cheap to compute but would require extensions to deal with derivative information whereas the second one has been tested for derivative information and multi-fidelity but are computationally expensive.

Chapter 4

Multi-objective Bayesian Optimization and dimension reduction: applications to numerical flow simulations

4.1 Introduction

The main drawback of Bayesian Optimization (BO) and other methods who rely on surrogate model is directly related to the curse of dimensionality. Indeed, as the dimension of the design space increases, the computational resources to build a reliable surrogate model and find the minimum of the objective function also increase. For this reason, even if in Chapter 2, BO was successfully applied to optimization problems with up to 31 design parameters, typical guidelines restrict the usage of BO to problems with a moderate number of dimensions, i.e. $N \leq 10$. As stated in Lam [57], two strategies can be adopted to overcome this drawback: including the gradient information (cf paragraph 9.4 of Rasmussen and Williams [96] for example) or reduce the dimension of the design space. However, the cost of building the surrogate model increases cubically with the number of observations [57]. Thus, it rapidly becomes inefficient to include the gradient information in the GP model, as in Chapter 3, for high-dimensional problems since $N + 1$ observations are included in the model for each objective function and gradient evaluation. The alternative approach is to reduce the dimension of the design space. For example, Carpentier and Munos [9] combined compressed sensing with the linear bandit problem to decrease the regret for high dimensional problems. Chen *et al.* [12] proposed a two-staged algorithm. First, the active variables are determined using hierarchical diagonal sampling (HDS). Then, BO is applied on the active variables. Hutter *et al.* [38] used random forests instead of a GP as a surrogate model. As mentioned in Shahriari [103], random forests can naturally determine the most active variables but are poor extrapolators. Wang *et al.* [116] developed the Random EMbedding Bayesian Optimization (REMBO) algorithm. The idea is to generate a random matrix and then project the initial design through this matrix in an embedded space of lower dimension. BO is then applied in this lower dimensional space. This method was successfully applied to a two-dimensional function embedded in a space of one billion of dimensions. A similar idea is to employ Active Subspaces (AS) [13]. AS

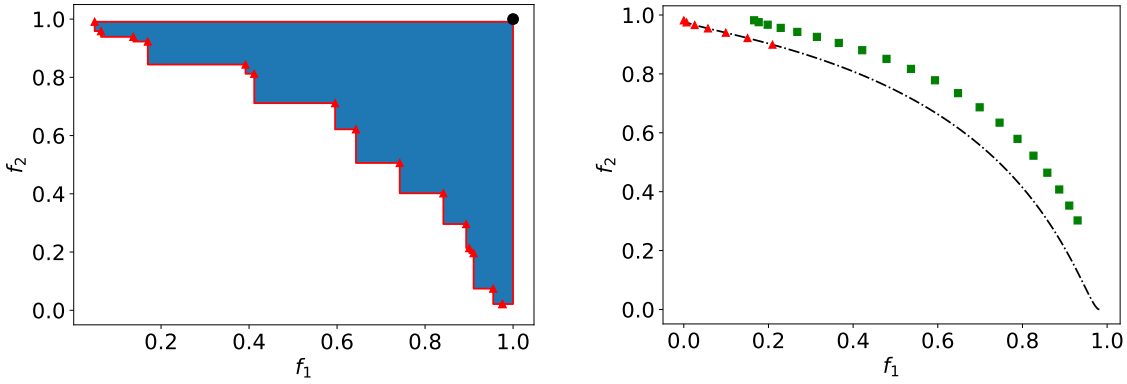
aims to build linear combinations of the parameters to find a design space of reduced dimension. In order to reduce the cost of this operation, Lam [57] followed this approach in the context of multi-fidelity information. Recently, Kusner *et al.* [55] used a variational autoenconder to reduce the dimension and a Gaussian Proccess latent variable model (GPVLM) [62] that includes the uncertainty in the input.

All the aforementioned methods showed promising results in the context of BO in high dimensional space. However, to the best knowledge of the author, there are few studies that address multi-objective BO (MOBO) in high dimensional problems. Multi-objective optimization aims at optimizing several objective functions concurrently. From the initial work of Jones [43], Knowles developed the ParEGO algorithm [50]. The idea is to represent all the objective functions by a unique objective function and set random weights from a predetermined set at each iteration on each objective function. The Expected Improvement (EI) criterion is then applied to determine the next design point to evaluate. Instead of building a unique objective function depending on the objective functions considered, it is also possible to develop acquisition criteria. For example, Keane [45] developed the Euclidean EI (EEI) criterion, Emerich *et al.* [18] proposed the expected hypervolume criterion (EHVI), Svenson and Santner [107] used the truncated maximum fitness function. However, these methods consider optimization problems with a dimension up to 10.

Some authors developed methods to combine multi-outputs and dimension reduction for high dimensional problems. For example, Lamboni *et al.* [60] developed an algorithm that combines Principal Component Analysis (PCA) and a global sensitivity analysis for multivariate and function outputs. PCA is firstly applied to identify the dominant dynamics in a time dependant signal. A sensitivity analysis is then performed on the most informative components. Ji *et al.* [42] developed AS for multiple outputs in the context of uncertainty quantification. Finally, Zahm *et al.* [125] generalized the concept of AS through ridge functions for uncertainty quantification. However, none of these studies have been applied in the context of MOBO.

Ling *et al.* [66] designed a MOBO algorithm for high dimensional space. The algorithm was successfully applied to the solution of an engineering problem with 37 design parameters. However, no dimension reduction technique was employed there. Lukaczyk *et al.* [73] applied BO to the shape optimization of the ONERA-M6 transonic wing. 50 design parameters were used and the design space dimension was finally reduced to 2 using Active Subspaces. As an objective function, they considered the drag that was subject to a lift constraint. They were thus able to link the active subspaces associated with the drag and the lift forces to proceed the optimization. However, even if several outputs in the context of BO were considered, this approach differs from MOBO in the sense that only one objective function was optimized subject to a constraint on the other one. Later, Grey and Constantine [30], applied the Active Subspaces algorithm to decrease the dimension of two different airfoil shape parameterizations. The lift and drag coefficient were both reduced from a maximum number of 11 design parameters to a two-dimensional space. Through visualizations of the drag and lift coefficients in the reduced space, they were able to find the Pareto front (solutions where an objective function can not be improved without deteriorating another one). However, no optimization algorithm was applied in that case and relied exclusively on visual inspections.

The goal of this chapter is to develop a method that combines dimension reduc-



(a) (b)

Figure 4.1: (a) Example of the Pareto front (indicated by the red triangles) and the hypervolume (blue shaded area) calculated from the reference point $\mathbf{r} = (1, 1)$ depicted by the black dot. (b) Illustrations of the proximity and diversity concepts. The red triangles exhibit a good proximity but a poor diversity compared to the green squares that have a higher diversity but a lower proximity. The true Pareto front is indicated by the black dash-dotted line.

tion techniques and MOBO. The proposed method is independent of the dimension reduction algorithm. This article is organized as follows: first, the methodology is presented in Section 4.2. Then, we introduce definitions, the initial MOBO algorithm and the dimension reduction techniques and the proposed technique. In Section 4.3, the developed algorithm is applied to a test function and onto two Computational Fluid Dynamics (CFD) applications. The flow fields and the performance of the proposed algorithm are then presented. Finally, conclusions are given in Section 4.4.

4.2 Methodology

Multi-objective optimization deals with the minimization of l objective functions f_1, f_2, ..., f_l concurrently. More precisely, this problem can be written as

$$\mathbf{s}^* = \arg\min_{\mathbf{s}\in\mathcal{S}}(f_1(\mathbf{s}), f_2(\mathbf{s}), \dots f_l(\mathbf{s})). \tag{4.1}$$

Optimizing some of these functions may be conflicting in the sense that minimizing one could maximize another one. Thus, the concept of optimum in the single-objective function case differs from the one in the multi-objective setting. In this case , a solution $\mathbf{s}_i$ is said to dominate another solution $\mathbf{s}_k$ if it fulfills the following two conditions:

$$\begin{cases} \forall j \in [1, 2, \dots, l], & f_j(\mathbf{s}_i) \leq f_j(\mathbf{s}_k), \\ \exists j \in [1, 2, \dots, l], & f_j(\mathbf{s}_i) < f_j(\mathbf{s}_k). \end{cases} \tag{4.2}$$

If the design $\mathbf{s}_i$ is not dominated by any other $\mathbf{s} \in \mathcal{S}$, then $\mathbf{s}_i$ is said to be a Pareto optimum. The set of all the Pareto optima define the Pareto front.

To compare the different algorithms that will be introduced later, we consider the hypervolume metric (HV) (see [50], for example). If the Pareto front, composed

of n^* solutions, is denoted by $\mathbf{s}^*_{1:n^*}$, then the HV corresponds to the volume in the objective functions space that is dominated by the solutions $\mathbf{s}^*_{1:n^*}$:

$$\mathrm{HV}(\mathbf{s}^*_{1:n^*}, \mathbf{r}) = \int_{\mathbb{Q}} \mathrm{d}\mathbf{q}, \tag{4.3}$$

where:

$$\mathbb{Q} = \{\mathbf{q} \in \mathbb{R}^l | \exists i \in [1, 2, \ldots, n^*], \forall j \in [1, 2, \ldots, l] : f_j(\mathbf{s}^*_i) \leq q_j \leq r_j\} \tag{4.4}$$

with $\mathbf{r} = (r_1, r_2, \ldots, r_l)^\mathsf{T}$ a reference point, which must be chosen such as it is dominated by every point in $\mathbf{s}^*_{1:n^*}$. As a reference point, we will take the anti-ideal point, i.e. the point with the worst value on all the objective functions, shifted by a small value:

$$r_j = \max{}_j + \delta_r(\max{}_j - \min{}_j), \quad \forall j \in [1, 2, \ldots, l], \tag{4.5}$$

where $\max_j$ and $\min_j$ are, respectively, the maximum and minimum of the objective function j. If $\max_j$ and $\min_j$ are not known, we will instead take respectively the maximum and minimum objective function j found among all the solutions evaluated of all the algorithms tested. Here, we set $\delta_r = 0.01$.

A graphical representation of the Pareto front and the HV for two objective functions is shown in Fig 4.1a. The HV enables us to quantify both the proximity, i.e. closeness to the true Pareto front, and the diversity, i.e. coverage of the Pareto front. When the proximity and diversity raise, the HV also increases. Both proximity and diversity are desirable in the multi-optimization process and are illustrated in Fig. 4.1b. Thus, a high HV is an indicator of the efficiency of the multi-optimization algorithm. Note that there are metrics to measure independently the proximity and the diversity as in Zuhal *et al.* [127] but will not be considered here.

4.2.1 Multi-objective Bayesian Optimization

In this work, Multi-objective Bayesian Optimization (MOBO) will be based on the ParEGO algorithm [50]. In the following, we describe how this algorithm works in the original design space $\mathcal{S}$.

Gaussian Process

As a first step, the augmented Tchebycheff function is introduced to deal with the l objective functions:

$$f_{tot}(\mathbf{s}) = \max_{j=1}^{l}(\omega_j \overline{f_j}(\mathbf{s})) + \xi \sum_{j=1}^{l} \omega_j \overline{f_j}(\mathbf{s}), \tag{4.6}$$

where $\mathbf{s}$ is a point in the design space $\mathcal{S}$, f_{tot} is the resulting objective function, $\overline{f_j}$ the j^{th} objective function f_j normalized between 0 and 1, and ξ a small positive value set to 0.05 as in the original implementation [50]. The first term on the right hand side of Eq. 4.6 ensures that points on non-convex regions of the Pareto front are also explored. The weights ω_j are drawn uniformly, at each iteration, from the set Ω:

$$\Omega = \left\{ \omega = (\omega_1, \omega_2, ..., \omega_l) | \sum_{j=1}^{l} \omega_j = 1 \wedge \forall j, \omega_j = \frac{\psi_1}{\psi_2}, \psi_1 \in \{0, ..., \psi_2\} \right\}. \tag{4.7}$$

In the remaining of this chapter, we set $\psi_2 = 10$, resulting in 11 weight factors as in the original implementation [50] for two objective functions.

Next, f_{tot} is approximated by a surrogate model $\hat{f}$

$$f_{tot}(\mathbf{s}) \approx \hat{f}(\mathbf{s}), \tag{4.8}$$

In this work, the GP model is used. The function f_{tot} is considered as the realization of a stochastic process

$$f_{tot}(\mathbf{s}) \sim \mathrm{GP}(\mu_0(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')). \tag{4.9}$$

We set $\mu_0(\mathbf{s}) = 0$ and we commit here to the Radial Basis Function (RBF) kernel:

$$k(r) = \sigma_f^2 \exp\left(-\frac{r^2}{2}\right), \tag{4.10}$$

where σ_f^2 is the variance, $r = (\mathbf{s} - \mathbf{s}')^\mathsf{T} \mathbf{\Lambda} (\mathbf{s} - \mathbf{s}')$ with $\mathbf{\Lambda}$ a diagonal squared matrix whose entries are $1/\lambda_i^2$, λ_i being a characteristic length scale along the i-th direction.

For each design $\mathbf{s}_k$ in the original design space, we can observe the objective function f_{tot} with possibly some noise as:

$$\begin{aligned} q_{tot,k} &= \max_{j=1}^{l} (\omega_j \overline{q_{j,k}}(\mathbf{s}_k)) + \xi \sum_{j=1}^{l} \omega_j \overline{q_{j,k}}(\mathbf{s}_k) \\ &= f_{tot}(\mathbf{s}_k) + \eta_k, \end{aligned} \tag{4.11}$$

where $\overline{q_{j,k}}$ is the observation $q_{j,k}$ of the objective function f_j at $\mathbf{s}_k$ normalized between 0 and 1, and η_k is the noise assumed to be drawn from a normal distribution $\mathcal{N}(0, \sigma_\eta^2)$ in the observations at the input $\mathbf{s}_k$.

After n observations of the objective function $\mathbf{q}_{tot,1:n} = (q_{tot,1}, q_{tot,2}, \ldots, q_{tot,n})^\mathsf{T}$ at designs $\mathbf{s}_{1:n} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n)^\mathsf{T}$, the joint prior distribution at $\mathbf{s}_{n+1}$ is

$$\begin{bmatrix} \mathbf{q}_{tot,1:n} \\ f_{tot,n+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \mathbf{K} + \sigma_\eta^2 \mathbf{I}_n & \mathbf{k} \\ \mathbf{k}^\mathsf{T} & k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) \end{pmatrix}\right), \tag{4.12}$$

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

As in Chapter 2, a posterior predictive distribution can then be obtained:

$$P(f_{tot,n+1} | \mathcal{D}_{1:n}, \mathbf{s}_{n+1}) = \mathcal{N}(\mu_n(\mathbf{s}_{n+1}), \sigma_n^2(\mathbf{s}_{n+1})), \tag{4.13}$$

where

$$\mu_n(\mathbf{s}_{n+1}) = \mathbf{k}^\mathsf{T} [\mathbf{K} + \sigma_\eta^2 \mathbf{I}_n]^{-1} \mathbf{q}_{tot,1:n}, \tag{4.14}$$

and

$$\sigma_n^2(\mathbf{s}_{n+1}) = k(\mathbf{s}_{n+1}, \mathbf{s}_{n+1}) - \mathbf{k}^\mathsf{T} [\mathbf{K} + \sigma_\eta^2 \mathbf{I}_n]^{-1} \mathbf{k}, \tag{4.15}$$

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

Again, as in Chapter 2, the vector of hyperparameters $\boldsymbol{\psi} = (\sigma_f, \lambda_i, \sigma_\eta)^\mathsf{T}$ is found by maximizing the logarithm of the marginal likelihood of the model

$$\log P(\mathbf{q}_{tot,1:n}|\mathbf{s}_{1:n}, \boldsymbol{\psi}) = -\frac{1}{2}\mathbf{q}_{tot,1:n}^\mathsf{T}(\mathbf{K} + \sigma_\eta^2\mathbf{I}_n)^{-1}\mathbf{q}_{tot,1:n} - \frac{1}{2}\log|\mathbf{K} + \sigma_\eta^2\mathbf{I}_n| - \frac{n}{2}\log(2\pi) \tag{4.16}$$

using gradient-based optimization.

Acquisition function

Once the posterior mean and variance are obtained and the hyperparameters optimized, we rely on the Expected Improvement (EI) acquisition function to determine the next design sample:

$$\mathrm{EI}_n(\mathbf{s}) = \begin{cases} \sigma_n(\mathbf{s})Z\Phi(Z) + \sigma_n(\mathbf{s})\phi(Z) & \text{if} \quad \sigma_n > 0, \\ 0 & \text{if} \quad \sigma_n = 0, \end{cases} \tag{4.17}$$

where EI_n is the EI criterion after n objective function observations, $Z = (q_{tot}^* - \mu_n(\mathbf{s}) - \kappa)/\sigma_n(\mathbf{s})$, q_{tot}^* being the best objective observation minimum found so far with the augmented Tchebycheff function, κ is a trade-off parameter, Φ and ϕ are respectively the cumulative density function and the probability density function. The combination of the augmented Tchebycheff function and EI_n is known as the ParEGO algorithm [50].

Finally, as mentioned in Lam [58], optimizing EI_n is difficult especially at the end of the optimization process. Thus, for this reason, if the Expected Improvement at the point suggested by the algorithm is lower than a threshold set here at 10^{-6}, the next candidate point is determined by

$$\mathbf{s}_{n+1} = \arg\min \mu_n(\mathbf{s}). \tag{4.18}$$

MOBO algorithm without dimension reduction

The MOBO algorithm without dimension reduction is detailed in Algorithm 3. For GP modelling and the optimization of the acquisition function EI_n and μ_n, we use the software emukit [87]. An in-house python program was developed to interface between this software and the numerical flow solver.

4.2.2 Dimension reduction

In this chapter, two different dimension reduction techniques are used: the Active Subspaces (AS) estimated with gradients (see [13]), and a method based on a quadratic model. The AS estimation with gradients was used when the gradients were available and cheap to evaluate. We also developed a method based on a quadratic model to easily tackle non-linear dimension reduction since the AS with gradients is a linear dimension reduction.

Algorithm 3: MOBO algorithm without dimension reduction

Initialization: $\mathcal{D} = \{\mathbf{s}_{1:n}, \mathbf{q}_{1,1:n}, \mathbf{q}_{2,1:n}, \dots \mathbf{q}_{l,1:n}\}$;
Initialize iteration counter: i = 0 ;
while $i < i_{max}$ **do**
  Compute the augmented Tchebycheff observations $\mathbf{q}_{tot,1:n+i}$;
  Update the Gaussian Process with $\{\mathbf{s}_{1:n+i}, \mathbf{q}_{tot,1:n+i}\}$;
  Find the design $\mathbf{s}_{n+i+1} = \arg\max \mathrm{EI}_{n+i}(\mathbf{s})$;
  if $\mathrm{EI}_{n+i}(\mathbf{s}_{n+i+1}) < 10^{-6}$ **then**
    $\mathbf{s}_{n+i+1} = \arg\min \mu_{n+i}(\mathbf{s})$;
  end
  for $j \in [1, 2, \dots, l]$ **do**
    Observe the objective function $q_{j,n+i+1}(\mathbf{s}_{n+i+1})$;
  end
  $\mathcal{D} = \mathcal{D} \cup \{\mathbf{s}_{n+i+1}, q_{1,n+i+1}(\mathbf{s}_{n+i+1}), q_{2,n+i+1}(\mathbf{s}_{n+i+1}), \dots, q_{l,n+i+1}(\mathbf{s}_{n+i+1})\}$;
  i = i+1 ;
end
Find the non-dominated solutions $\mathbf{s}^*_{1:n^*}$ and the corresponding objective functions observations $\mathbf{q}^*_{1:n^*}$ in $\mathcal{D}$;
return $\{\mathbf{s}^*_{1:n^*}, \mathbf{q}^*_{1:n^*}\}$

Active Subspace estimated with the gradients

Active Subspaces is a method aiming to obtain a reduced design variable from a linear combination of the initial design variables. First, the inputs are normalized into the design space $[-1, 1]^N$. The design in the initial design space can be found with

$$\mathbf{s}_{ini} = \frac{1}{2}((\mathbf{s}_{ini,up} - \mathbf{s}_{ini,low})\mathbf{s} + (\mathbf{s}_{ini,up} + \mathbf{s}_{ini,low})), \tag{4.19}$$

where $\mathbf{s}$ is the normalized input, $\mathbf{s}_{ini,up}$ and $\mathbf{s}_{ini,low}$ represent respectively the upper and lower bounds of the original design space.

A sampling density is then chosen to build an initial data set in the normalized design space $[-1, 1]^N$. When the gradients are available and cheap to compute, for each sample $\mathbf{s}_k$, we observe the objective function j, $q_{j,k} = q_j(\mathbf{s}_k)$ and its gradient $\nabla_{\mathbf{s}} q_{j,k} = \nabla q_j(\mathbf{s}_k)$. For n samples drawn from the sampling density, we can calculate the following matrix and its eigenvalue decomposition:

$$\hat{\mathbf{C}}_j = \frac{1}{n}\sum_{k=1}^{n} \nabla_{\mathbf{s}} q_{j,k} \nabla_{\mathbf{s}} q_{j,k}^{\mathsf{T}} = \hat{\mathbf{W}}_j \hat{\mathbf{\Lambda}}_j \hat{\mathbf{W}}_j^{\mathsf{T}}, \tag{4.20}$$

where $\hat{\mathbf{W}}_j$ is the eigenvectors matrix and $\hat{\mathbf{\Lambda}}_j$ is the diagonal eigenvalues matrix. Since the matrix $\hat{\mathbf{C}}_j$ is symmetric, all its eigenvalues are real. Both $\hat{\mathbf{W}}_j$ and $\hat{\mathbf{\Lambda}}_j$ are sorted by decreasing eigenvalues. If a large gap is observed between the m_j and $m_j + 1$ eigenvalues of $\hat{\mathbf{\Lambda}}_j$, a reduced variable of dimension m_j can be obtained and $\hat{\mathbf{W}}_j$ and $\hat{\mathbf{\Lambda}}_j$ can be recast in the following form

$$\hat{\mathbf{W}}_j = \begin{pmatrix} \mathbf{U}_j & \mathbf{V}_j \end{pmatrix}, \quad \hat{\mathbf{\Lambda}} = \begin{pmatrix} \mathbf{\Lambda}_{j,1} & \mathbf{\Lambda}_{j,2} \end{pmatrix}, \tag{4.21}$$

where $\mathbf{U}_j$ and $\mathbf{\Lambda}_{j,1}$ contain the m_j first eigenvectors and eigenvalues, respectively, whereas $\mathbf{V}_j$ and $\mathbf{\Lambda}_{j,2}$ contain the remaining $N - m_j$ eigenvectors and eigenvalues, respectively. A reduced variable can then be obtained as follows:

$$h_j(\mathbf{s}) = \mathbf{U}_j^\mathsf{T}\mathbf{s}. \tag{4.22}$$

Method based on a quadratic model

Since the mapping provided by AS between the reduced variable and the original design space is a linear relationship, this method is not really adapted to the case when the objective function can be written as a quadratic sum of the initial design variables. We then develop a method to alleviate this drawback.

A quadratic model is fitted to the design variables such as

$$h_j(\mathbf{s}) = \sum_{i=1}^{N}(a_{j,i}s^{(i)2} + b_{j,i}s^{(i)}) + c_j, \tag{4.23}$$

where $h_j(\mathbf{s})$ is a one-dimensional variable, $s^{(i)}$ is the i^{th} variable of the design $\mathbf{s}$ in the design space $\mathcal{S} = [-1, 1]^N$. If we have n observations $\mathbf{q}_{j,1:n}$ at n initial design points $\mathbf{s}_{1:n}$, we can then introduce the corresponding $h_j(\mathbf{s}_{1:n})$ and fit a GP on the data $\mathcal{D}_n = \{h_j(\mathbf{s}_{1:n}), \mathbf{q}_{j,1:n}\}$. We then choose the coefficients $a_{j,i}$, $b_{j,i}$ and c_j that maximize the marginal likelihood of the GP through gradient-based optimization with the L-BFGS-B algorithm. The gradients can be obtained with the chain rule:

$$\begin{cases} \frac{\partial P(\mathbf{q}_{j,1:n}|h_j(\mathbf{s}_{1:n}),\boldsymbol{\psi})}{\partial a_{j,i}} = \frac{\partial P(\mathbf{q}_{j,1:n}|h_j(\mathbf{s}_{1:n}),\boldsymbol{\psi})}{\partial h_j(\mathbf{s}_{1:n})}\frac{\partial h_j(\mathbf{s}_{1:n})}{\partial a_{j,i}}, \\ \frac{\partial P(\mathbf{q}_{j,1:n}|h_j(\mathbf{s}_{1:n}),\boldsymbol{\psi})}{\partial b_{j,i}} = \frac{\partial P(\mathbf{q}_{j,1:n}|h_j(\mathbf{s}_{1:n}),\boldsymbol{\psi})}{\partial h_j(\mathbf{s}_{1:n})}\frac{\partial h_j(\mathbf{s}_{1:n})}{\partial b_{j,i}}, \\ \frac{\partial P(\mathbf{q}_{j,1:n}|h_j(\mathbf{s}_{1:n}),\boldsymbol{\psi})}{\partial c_j} = \frac{\partial P(\mathbf{q}_{j,1:n}|h_j(\mathbf{s}_{1:n}),\boldsymbol{\psi})}{\partial h_j(\mathbf{s}_{1:n})}\frac{\partial h_j(\mathbf{s}_{1:n})}{\partial c_j}. \end{cases} \tag{4.24}$$

Once a dimension reduction h_j is obtained, we can express the objective function f_j as:

$$f_j(\mathbf{s}) \approx g_j(h_j(\mathbf{s})). \tag{4.25}$$

4.2.3 Multi-objective Bayesian Optimization and dimension reduction

Dimension reduction

In the case of MOBO, the dimension reduction can be performed on each of the objective functions that we are seeking to minimize. For each f_j, n initial observations of the objective function $\mathbf{q}_{j,1:n}$ are performed at n designs $\mathbf{s}_{1:n}$ in the original space. A dimension reduction $h_j(\mathbf{s})$ is then performed on the data pair $\{\mathbf{s}_{1:n}, \mathbf{q}_{j,1:n}\}$.

We can then build a variable $\hat{\mathbf{s}} = (h_1(\mathbf{s}), h_2(\mathbf{s}), \ldots, h_l(\mathbf{s}))^\mathsf{T}$ of dimension $\hat{N} = \sum_{j=1}^{l} N_j$, where N_j is the dimension of the reduced space. For example, let us consider two objective functions f_1 and f_2 we wish to minimize that can be expressed in reduced dimensional spaces as $h_1(\mathbf{s})$ in one dimension and $h_2(\mathbf{s})$ in two dimensions, respectively. We then build a three dimensional variable where the first coordinate is $h_1(\mathbf{s})$ and the two others are associated with $h_2(\mathbf{s})$.

For each dimension reduction $h_j(\mathbf{s})$, the boundaries of the i-th coordinate $h_j^{(i)}(\mathbf{s})$ are indicated by

$$\begin{aligned} h_{j,low}^{(i)} &= \min_{\mathbf{s}\in\mathcal{S}} h_j^{(i)}(\mathbf{s}), \\ h_{j,up}^{(i)} &= \max_{\mathbf{s}\in\mathcal{S}} h_j^{(i)}(\mathbf{s}), \end{aligned} \tag{4.26}$$

where $h_{j,low}$ and $h_{j,up}$ are respectively the lower and upper bounds of $h_j^{(i)}(\mathbf{s})$. Eq. 4.26 can be solved either with an optimization scheme or analytically.

Also, a constraint is required on the subspace for optimization. Indeed, in order to perform the mapping from the subspace towards the original design space, for each point in the subspace $\hat{\mathbf{s}} = (\hat{s}^{(1)}, \hat{s}^{(2)}, \ldots, \hat{s}^{(\hat{N})})^\mathsf{T}$, at least one design point $\mathbf{s}$ must be find in $\mathcal{S}$ such that $\hat{\mathbf{s}} = (h_1(\mathbf{s}), h_2(\mathbf{s}), \ldots, h_l(\mathbf{s}))^\mathsf{T}$. For this reason, during the optimization process in the subspace, we only consider the points $\hat{\mathbf{s}}$ satisfying the constraint:

$$\min_{\mathbf{s}\in\mathcal{S}} \sum_{j=1}^{l} \|h_j(\mathbf{s}) - \hat{\mathbf{s}}^{(\sum_{i=1}^{j-1} N_i : \sum_{i=1}^{j-1} N_i + N_j)}\| \leq \epsilon_{des}, \tag{4.27}$$

with $\hat{\mathbf{s}}^{(\sum_{i=1}^{j-1} N_i : \sum_{i=1}^{j-1} N_i + N_j)}$ is a vector of dimension j containing the coordinates of $\hat{\mathbf{s}}$ from $\sum_{i=1}^{j-1} N_i$ to $\sum_{i=1}^{j-1} N_i + N_j$. The term on the left hand side of Eq. 4.27 is solved with a quasi-Newton method. Ideally, $\epsilon_{des} = 0$ but this value is numerically hard to obtain. For this reason, we set $\epsilon_{des} = 10^{-6}$.

Gaussian Process, acquisition function and constraint

The Gaussian Process and the acquisition function steps are done as in the case without dimension reduction. The difference is that the model and the acquisition functions are then designed in the subdimensional space of dimension $\hat{N}$ instead of N, and the variable $\hat{\mathbf{s}}$ is used instead of $\mathbf{s}$ for the design space. The boundaries of the design space are then set through Eq. 4.26 and the constraint 4.27 is applied when optimizing the acquisition function.

Mapping to the original space

Once a design $\hat{\mathbf{s}}_{n+1}$ has been selected through the acquisition function step, a mapping towards the original design space of dimension N is necessary to evaluate the objective functions. Thus, in order to retrieve the design $\mathbf{s}_{n+1}$ in the original design space, we have to solve:

$$\mathbf{s}_{n+1} = \arg\min_{\mathbf{s}\in\mathcal{S}} \sum_{j=1}^{l} \|h_j(\mathbf{s}) - \hat{\mathbf{s}}_{n+1}^{(\sum_{i=1}^{j-1} N_i : \sum_{i=1}^{j-1} N_i + N_j)}\|. \tag{4.28}$$

Eq. 4.28 is solved with a quasi-Newton method.

MOBO algorithm with dimension reduction

The MOBO algorithm with dimension reduction is described in Algorithm 4. Again, for Gaussian Process modelling and the optimization of the acquisition function EI_n

and μ_n, we use the software emukit [87]. All the dimension reduction algorithms as well as the mapping toward the design in the original design space were implemented in the python interface we developed between emukit and the CFD software.

Algorithm 4: MOBO algorithm with dimension reduction

Initialization: $\mathcal{D} = \{\mathbf{s}_{1:n}, \mathbf{q}_{1,1:n}, \mathbf{q}_{2,1:n}, \dots \mathbf{q}_{l,1:n}\}$;
Initialize iteration counter: i = 0 ;
while $i < i_{max}$ **do**
  for $j \in [1, 2, \dots, l]$ **do**
    Find a dimension reduction $h_j(\mathbf{s})$ with $\{\mathbf{s}_{1:n+i}, \mathbf{q}_{j,1:n+i}\}$ of dimension N_j ;
    for $i \in [1, 2, \dots, N_j]$ **do**
      Compute the boundaries of $h^{(i)}_{j,low}$ and $h^{(i)}_{j,up}$ (Eq. 4.26) ;
    end
  end
  Create the variable $\hat{\mathbf{s}} = (h_1(\mathbf{s}), h_2(\mathbf{s}), \dots, h_j(\mathbf{s}))^\mathsf{T}$;
  Compute the augmented Tchebycheff observations $\mathbf{q}_{tot,1:n+i}$;
  Update the Gaussian Process with $\{\hat{\mathbf{s}}_{1:n+i}, \mathbf{q}_{tot,1:n+i}\}$;
  Set the constraint defined in Eq. 4.27 ;
  Find the design in the subspace $\hat{\mathbf{s}}_{n+i+1} = \arg\max \mathrm{EI}_{n+i}(\hat{\mathbf{s}})$;
  if $\mathrm{EI}_{n+i}(\hat{\mathbf{s}}_{n+i+1}) < 10^{-6}$ **then**
    $\hat{\mathbf{s}}_{n+i+1} = \arg\min \mu_{n+i}(\hat{\mathbf{s}}_{n+i+1})$;
  end
  Find the original design $\mathbf{s}_{n+i+1}$ by solving Eq. 4.28 ;
  for $j \in [1, 2, \dots, l]$ **do**
    Observe the objective function $q_{j,n+i+1}(\mathbf{s}_{n+i+1})$;
  end
  $\mathcal{D} = \mathcal{D} \cup \{\mathbf{s}_{n+i+1}, q_{1,n+i+1}(\mathbf{s}_{n+i+1}), q_{2,n+i+1}(\mathbf{s}_{n+i+1}), \dots, q_{l,n+i+1}(\mathbf{s}_{n+i+1})\}$;
  i = i+1 ;
end
Find the non-dominated solutions $\mathbf{s}^*_{1:n^*}$ and the corresponding objective functions observations $\mathbf{q}^*_{1:n^*}$ in $\mathcal{D}$;
return $\{\mathbf{s}^*_{1:n^*}, \mathbf{q}^*_{1:n^*}\}$

4.3 Applications

4.3.1 Fonseca-Fleming problem

We test the algorithm on the Fonseca-Fleming problem introduced in [20]. The following objective functions are considered:

$$\begin{aligned} f_1(\mathbf{s}_{ini}) &= 1 - \exp\left(-\sum_{i=1}^{N}\left(s^{(i)}_{ini} - \frac{1}{\sqrt{N}}\right)^2\right) \\ f_2(\mathbf{s}_{ini}) &= 1 - \exp\left(-\sum_{i=1}^{N}\left(s^{(i)}_{ini} + \frac{1}{\sqrt{N}}\right)^2\right), \end{aligned} \tag{4.29}$$

where $s_{ini}^{(i)}$ is the i^{th} variable of $\mathbf{s}_{ini}$ and N is the dimension of the design space.

The Pareto front is composed of all the points $\mathbf{s}_{ini}$ satisfying

$$s_{ini}^{(1)} = s_{ini}^{(2)} = \cdots = s_{ini}^{(N)} \wedge -\frac{1}{\sqrt{N}} \leq s_{ini}^{(i)} \leq \frac{1}{\sqrt{N}}. \tag{4.30}$$

Initially, we set $-4 \leq s_{ini}^{(i)} \leq 4$ for the design space. However, when we increased the dimension N to large values, no valuable information was extracted from the DOE as f_1 and f_2 were equal to 1 for all the design points. For this reason, we decided to set $-2/\sqrt{N} \leq s_{ini}^{(i)} \leq 2/\sqrt{N}$. We consider the following cases with increasing dimension $N = 2, 5, 10, 20, 50$ and 100.

Latin Hypercube Sampling [11] was used to draw 7, 17, 24, 30, 40 and 47 initial samples for, respectively, $N = 2, 5, 10, 20, 50$ and 100. Then, the quadratic model dimension reduction discussed in in 4.2.2 was applied to the DOE for f_1 and f_2. Finally, the developed algorithm (Algorithm 4) was run for each case and compared with the case without dimension reduction (Algorithm 3). A maximum number of 100 iterations was chosen for each N considered. For consistency, the same ω in Eq. 4.7 was used at each iteration for both methods. For the optimization, we set the following boundaries in the reduced design space:

$$\begin{aligned} h_{j,low} &= -\sum_{i=1}^{N}(|a_{j,i}| + |b_{j,i}|) - |c_j|, \\ h_{j,up} &= \sum_{i=1}^{N}(|a_{j,i}| + |b_{j,i}|) + |c_j|, \end{aligned} \tag{4.31}$$

where $h_{j,low}$ and $h_{j,up}$ are respectively the lower and upper boundaries of the dimension reduction function h_j associated to the function f_j. The coefficients of the quadratic model $a_{j,i}$, $b_{j,i}$ and c_j (see Eq. 4.23) are related to the mapping h_j. $h_{j,low}$ and $h_{j,high}$ are lower and higher, respectively, than they should theoretically be with Eq. 4.31. However, setting them in that way avoids to solve Eq. 4.26 with an optimization algorithm. Moreover, since only the designs satisfying the constraint defined in Eq. 4.27 are considered, a larger subspace domain does not influence the optimal solution found.

The design space of the MOBO algorithm without dimension reduction is $\mathcal{S} = [-1, 1]^N$. The MOBO algorithm with dimension reduction performs the mapping from $\hat{\mathbf{s}}$ towards $\mathcal{S}$ through Eq. 4.28. Once a design point is selected in $\mathcal{S}$, the design point is found in the original design space by $\mathbf{s}_{ini} = 2\mathbf{s}/\sqrt{N}$.

On Fig. 4.2 is depicted the Pareto Front found with the MOBO algorithms with and without dimension reduction for 100 optimization iterations. As can be seen on this figure, for $N = 2$, both methods are able to find a reasonable number of points on the true Pareto front defined in Eq. 4.30. However, the performance of the MOBO algorithm without dimension reduction significantly decreases as the dimension space is increased since a poorer proximity to the true Pareto front is observed. On the other hand, with the dimension reduction, the algorithm is still able to find a significant number of points close to the true Pareto front even for $N = 100$. A better diversity than without dimension reduction can also be observed.

The hypervolume (HV) calculated from the reference point $\mathbf{r} = (1.01, 1.01)$ (computed with Eq. 4.5) as a function of the iteration algorithm is shown on Fig. 4.3. As

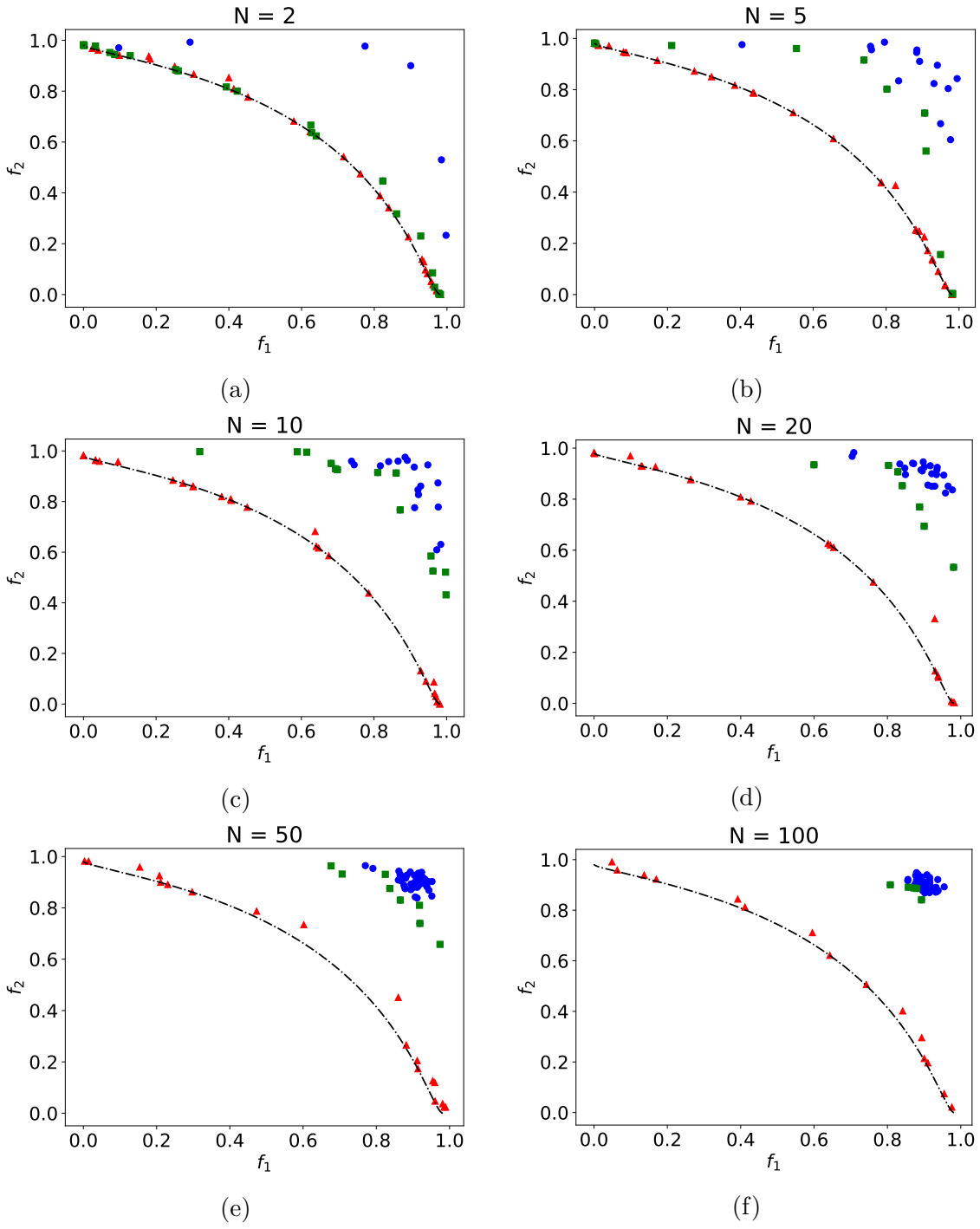


Figure 4.2: Pareto fronts found with the MOBO algorithm with dimension reduction (red triangles) and the MOBO algorithm without dimension reduction (green squares) for the Fonseca-Fleming problem with (a) $N = 2$, (b) $N = 5$, (c) $N = 10$, (d) $N = 20$, (e) $N = 50$ and (f) $N = 100$. The initial DOE is represented with blue dots and the black dash-dotted line is the true Pareto front defined in Eq. 4.30.

	C_d	L_d/D
Ref. [14]	-	2.13
Ref. [113]	1.59	-
Ref. [16]	1.52	2.35
Ref. [67]	1.54	2.28
Ref. [108]	1.55	2.33
Present study	**1.58**	**2.38**

Table 4.1: Comparison with the literature of the drag coefficient C_d and length of the recirculation zone from the aft of the cylinder L_d/D at $y = 0$ for $\mathrm{Re} = 40$.

can be seen, with the dimension reduction, the HV is lower than for the MOBO algorithm without dimension reduction at each iteration for all the cases. We can also note that changing the design space is even efficient for the case $N = 2$, whereas the dimension of the reduced design space is also equal to the dimension of the original design space. Finally, after 100 iterations, the HV computed with the algorithm with dimension reduction achieves respectively 92%, 90%, 85%, 83%, 73% and 83% of the HV of the Pareto front computed from 200 points linearly spaced on the line defined in Eq. 4.30 for respectively $N = 2, 5, 10, 20, 50$ and 100.

4.3.2 Cylinder at $Re = 40$

Problem description

An external flow around a two-dimensional cylinder at Re=40 is considered. The surface of the cylinder is discretized using 79 equispaced Lagrangian points. At each of these points j, a tangential velocity $v_{\theta,j}/U_\infty = g^{(j)}_{filter}(\mathbf{s})$ is specified, where U_∞ is the free stream velocity, $\mathbf{s} \in [-1, 1]^{79}$ and $g^{(j)}_{filter}$ is the j^{th} element of the filter function, given by

$$g_{filter}(\mathbf{s}) = \mathbf{G}^4\mathbf{s}, \tag{4.32}$$

with $\mathbf{G}$ a 79×79 sparse matrix defined by

$$G = \begin{pmatrix} 0.5 & 0.25 & 0 & \dots & 0.25 \\ 0.25 & 0.5 & 0.25 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.25 & 0 & \dots & 0.25 & 0.5 \end{pmatrix}. \tag{4.33}$$

The filter function smooths the final velocity profile by weighting at each point i the prescribed tangential velocity $s^{(i)}$ by the values around the point.

Our aim is to minimize both the drag coefficient and the actuation cost. Thus, we consider the following objective functions:

$$f_1(\mathbf{s}) = C_d^2(\mathbf{s}), \quad f_2(\mathbf{s}) = \mathbf{s}^\mathsf{T}\mathbf{s}, \tag{4.34}$$

where C_d is the drag coefficient once the steady flow is obtained.

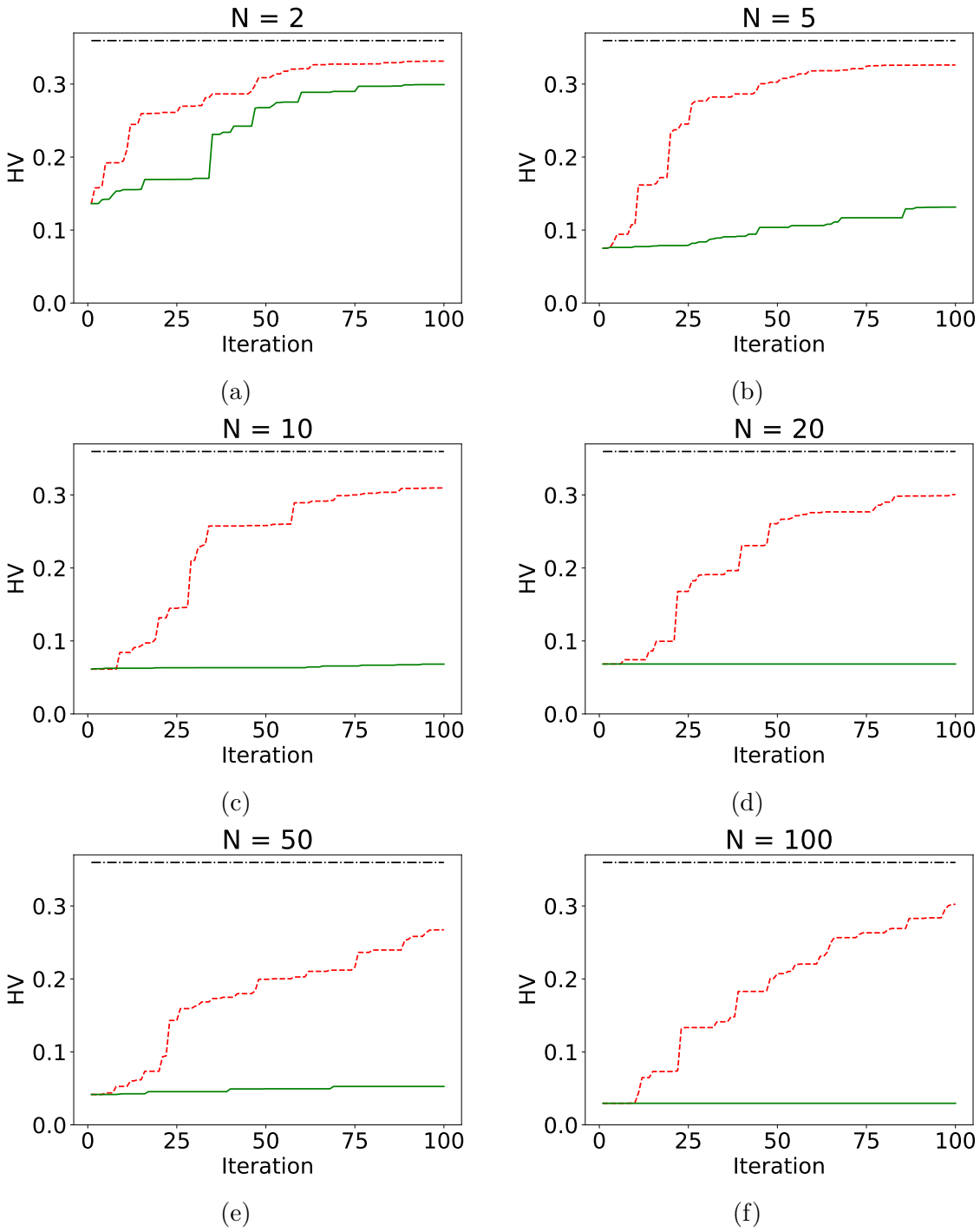
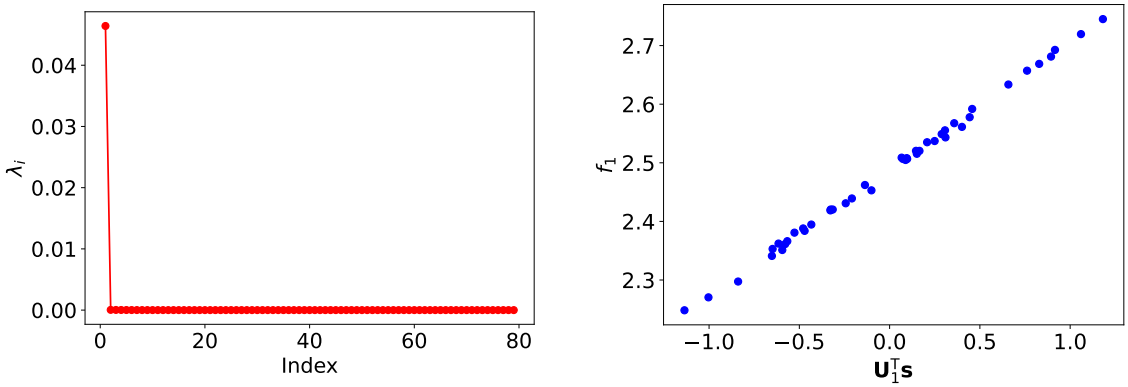


Figure 4.3: Hypervolume as a function of the iteration for the Fonseca-Fleming problem with the reference point $\mathbf{r} = (1.01, 1.01)$. (a) $N = 2$, (b) $N = 5$, (c) $N = 10$, (d) $N = 20$, (e) $N = 50$ and (f) $N = 100$. The red dashed and green solid lines are the MOBO algorithms respectively with and without dimension reduction whereas the black dash-dotted line corresponds to the HV computed from 200 points linearly spaced on the line defined in Eq. 4.30.



(a) (b)

Figure 4.4: (a) Eigenvalues obtained with the AS method with gradients for f_1 defined in Eq. 4.34. (b) f_1 as a function of the reduced variable.

Numerical set-up

We use an in-house solver [22] that is based on the immersed boundary method of Taira and Colonius [108]. The domain is rectangular and is included in $[-14.28, 38.32]$ in the streamwise direction and in $[-22.3, 22.3]$ in the cross-flow direction. 288 cells were used in the streamwise direction and 256 in the cross-flow direction. The mesh is refined around the cylinder where the mesh size is $\Delta x = 0.04$. Far from the cylinder, the grid is stretched until the mesh size is $\Delta x = 0.25$. A cylinder of unit diameter is centered at the origin $(0, 0)$ is represented by 79 Lagrangian markers.

Values of the drag coefficient C_d and the length of the recirculation zone starting from the aft of the cylinder are compared with the literature in Table 4.1. It is observed that even if the length of the recirculation zone is slightly higher than the other studies, the results are in good agreement with them.

We apply the AS method with gradients to the first objective function f_1 described in 4.2.2. 44 initial points were drawn from a Latin Hypercube Sampling. The eigenvalues, shown in Fig. 4.4a, suggest that the dependency of f_1 can be reduced to a single variable since a large gap can be observed between λ_1 and λ_2. Its dimension reduction is represented in Fig. 4.4b as

$$f_1(\mathbf{s}) = g_1(\mathbf{U}_1^\mathsf{T}\mathbf{s}) \tag{4.35}$$

where $\mathbf{U}_1$ is the eigenvector associated to the highest eigenvalue.

Regarding the second objective function f_2, a straightforward dimension reduction that can be written as a one-dimensional function is

$$f_2(\mathbf{s}) = g_2(h_2(\mathbf{s})) = h_2(\mathbf{s}) = \mathbf{s}\mathbf{s}^\mathsf{T}. \tag{4.36}$$

A design space of dimension 2 is then built with $\mathbf{U}_1^\mathsf{T}\mathbf{s}$ as the first coordinate and $\mathbf{s}\mathbf{s}^\mathsf{T}$ as the second coordinate. The boundaries of the first dimension were directly obtained with $h_{1,low} = -|\mathbf{U}^\mathsf{T}|\mathbf{1}$ and $h_{1,up} = |\mathbf{U}^\mathsf{T}|\mathbf{1}$, where $\mathbf{1}$ is a column vector where each entry is equal to one. For the dimension reduction of f_2, the boundaries of $h_2(\mathbf{s})$ are straightforward $h_{2,low} = 0$ and $h_{2,up} = 79$ (when $\forall i, s^{(i)} = \pm 1$).

The Algorithm 4 was then used to solve the above optimization problem. The number of iterations is 100 and the results were compared with those obtained

$f_1(\mathbf{s})$	$f_2(\mathbf{s})$	C_d	C_l	a/D	L_d/D
1.21	77.2	1.10	-6.06×10^{-3}	-	-
1.33	41.8	1.15	-4.24×10^{-4}	-	-
1.45	30.5	1.21	-3.06×10^{-5}	-	-
1.61	20.9	1.27	-8.73×10^{-5}	-	-
1.98	6.5	1.41	-5.93×10^{-5}	0.03	0.85
2.21	2.1	1.49	-9.28×10^{-5}	0.03	1.52
2.28	1.0	1.51	-3.48×10^{-5}	0.03	1.71

Table 4.2: Objective functions $f_1(\mathbf{s})$, $f_2(\mathbf{s})$, drag coefficient C_d, 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 for some of the Pareto solutions for $\mathrm{Re} = 40$.

using the MOBO algorithm without dimension reduction (Algorithm 3). No additional gradient evaluations were performed during the optimization process. In other words, the dimension reduction h_1 and its boundaries $h_{1,up}$ and $h_{1,low}$ were not changed after its computation with the DOE.

Results

Some Pareto solutions are depicted in Table 4.2. The drag coefficient value, the lift coefficient and the length of the recirculation zone are represented. A minimum drag reduction of 4% was achieved for $f_2(\mathbf{s}) = 1.0$, whereas we were able to reach a 30% drag reduction for $f_2(\mathbf{s}) = 77.2$. For all these cases, the lift coefficient is close to zero. Finally, for low amplitude actuation, we can observe a recirculation zone as with $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.98, 6.5)$. This recirculation zone is increased as the total actuation amplitude is decreased (see $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (2.21, 2.1)$ and $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (2.28, 1.0)$).

We represent in Fig. 4.5a the optimal velocity profiles of these Pareto front solutions with the optimal velocity profile found by optimizing f_1 with adjoint methods. The solution at $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.21, 77.2)$ is very close to the one obtained with the adjoint solution. We should also note that on the front of the cylinder, there is a small zone between $\theta = 3\pi/4$ and $\theta = 5\pi/4$ where the actuation is set in the opposite way of the free stream direction. This was already observed in Fig. 2.3. This means that setting the actuators in that way could participate to the drag reduction. The remaining optimal solutions are close to the ones found in Fig. 2.3, even if a two bumps form can be observed for some solutions (e.g. $(f_1(\mathbf{s}) = 1.45, f_2(\mathbf{s})) = 30.5)$).

The streamwise velocity and streamlines are also depicted in Fig. 4.5 with $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.21, 77.2)$ in Fig. 4.5b, $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.33, 41.8)$ in Fig. 4.5c, $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.45, 30.5)$ in Fig. 4.5d, $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.61, 20.9)$ in Fig. 4.5e, $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (1.98, 6.5)$ in Fig. 4.5f, $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (2.21, 2.1)$ in Fig. 4.5g and $(f_1(\mathbf{s}), f_2(\mathbf{s})) = (2.28, 1.0)$ in Fig. 4.5h. When we decrease the actuation, the flow behind the cylinder will slow down until to form a recirculation bubble with two counter-rotating vortices, resembling the characteristics of the uncontrolled flow. The size of these vortices increase when the actuation is lowered (Fig. 4.5f, 4.5g and 4.5h).

The Pareto front is depicted in Fig. 4.6a. A total of 42 Pareto solutions were found with the dimension reduction against 11 for the MOBO algorithm without

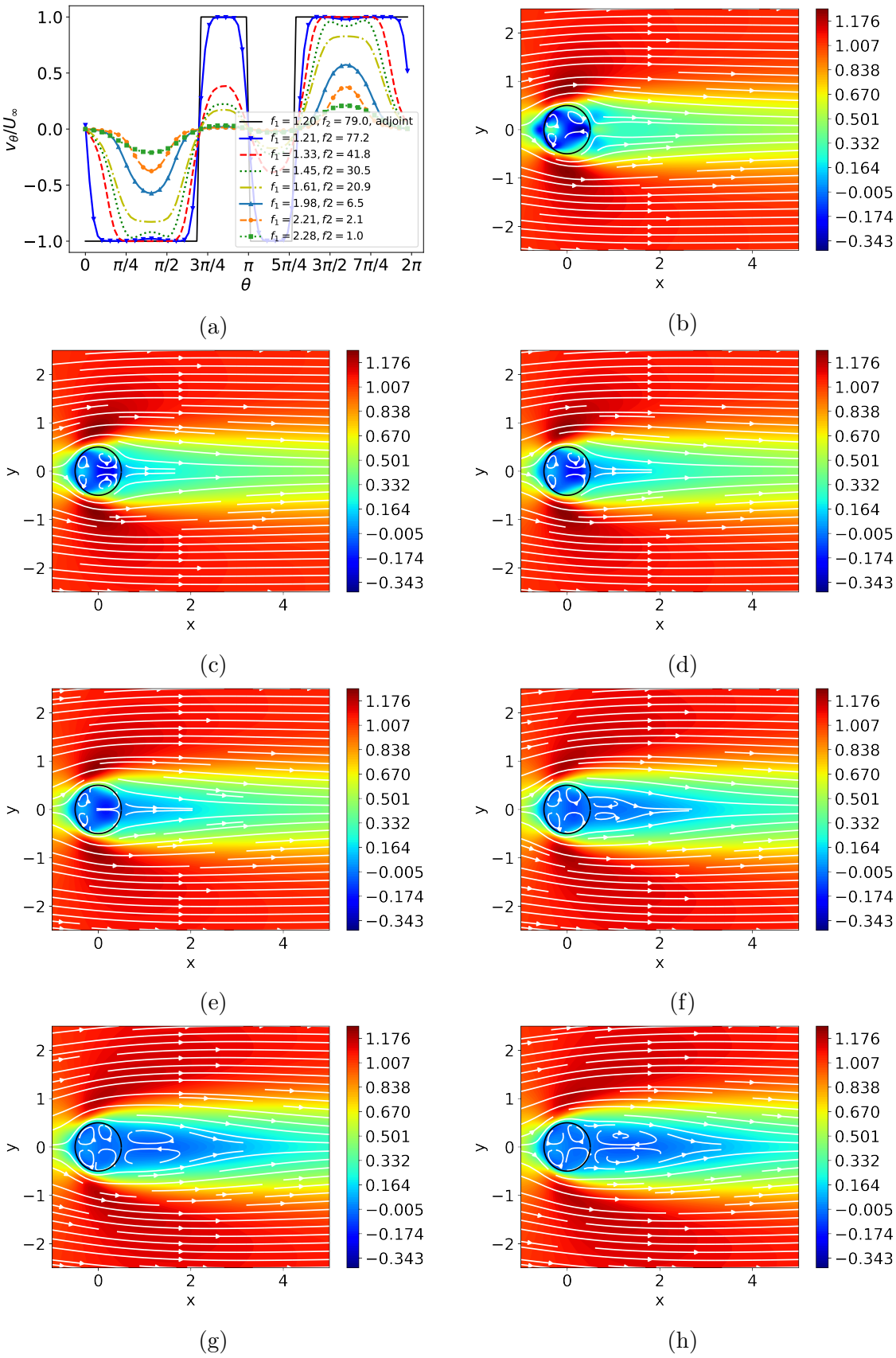
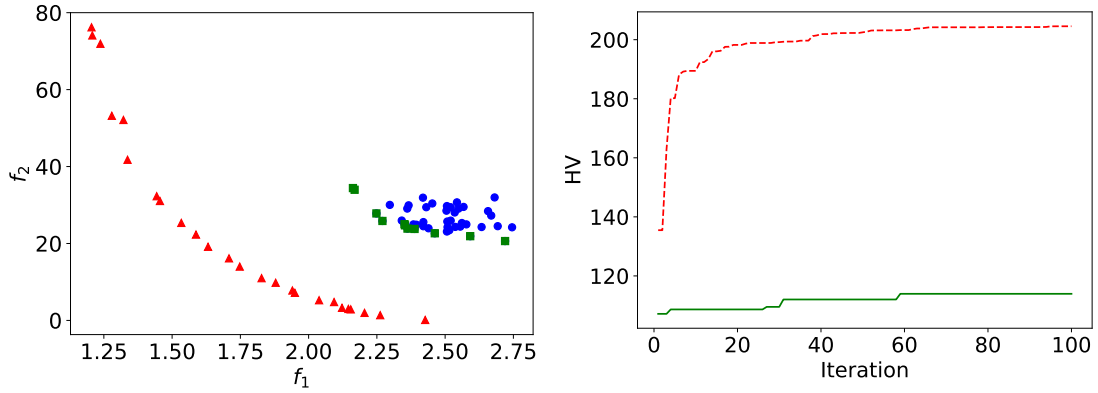


Figure 4.5: Fluid results at Re = 40. (a) Optimal velocity profiles. Streamwise velocity for (b) $f_1 = 1.21, f_2 = 77.2$ (c) $f_1 = 1.33, f_2 = 41.8$, (d) $f_1 = 1.45, f_2 = 30.5$ (e) $f_1 = 1.61, f_2 = 20.9$, (f) $f_1 = 1.98, f_2 = 6.5$, (g) $f_1 = 2.21, f_2 = 2.1$, and (h) $f_1 = 2.28, f_2 = 1.0$. Streamlines and its directions are indicated by the white lines and arrows.



(a) (b)

Figure 4.6: (a) Pareto fronts found with the MOBO algorithm with dimension reduction (red triangles) and the MOBO algorithm without dimension reduction (green squares) for the cylinder at Re = 40 and the objective functions defined in Eq. 4.34. Blue dots represent the initial DOE. (b) Hypervolume as a function of the iteration of the algorithm. The red dashed and green solid lines are the MOBO algorithms respectively with and without dimension reduction.

dimension reduction. None of the Pareto solutions found with the MOBO algorithm with dimension reduction are dominated by the ones found with the MOBO algorithm without dimension reduction. The Pareto front obtained by the former method is also more diverse than for the latter one. This is confirmed by the HV, as computed from the reference point $\mathbf{r} = (4.08, 79.79)$ (calculated with Eq. 4.5) in Fig. 4.6b. Indeed, the HV found with dimension reduction is higher at each iteration than the algorithm without dimension reduction.

4.3.3 NACA0012 profile at $Re = 1000$

Problem description

We consider now an external flow around a bi-dimensional NACA0012 profile at Re = 1000 with an angle of attack $AoA = 10°$ and a chord length $c = 1$. We set $N = 10, 20, 40$ or 80 points around the profile at a regular interval Δe with e being the airfoil profile arc length. Again, for each of these points e_i, we consider a tangential velocity $v_{\theta,i}/U_\infty = \mathbf{s}^{(i)}/4$ where $\mathbf{s} \in [-1, 1]^N$. Then, we perform a linear interpolation around the airfoil profile and for each Lagrangian marker j on the airfoil, we have

$$v_\theta(e_j) = \sum_{i=1}^{N} v_{\theta,i} \Lambda \left(\frac{e_j - e_i}{\Delta e} \right), \qquad \text{with } \Lambda(x) = \max\left(1 - |x|, 0\right), \tag{4.37}$$

where e_j (respectively e_i) is the arc length between the nose and the Lagrangian marker j (respectively the point i where we set the velocity $v_{\theta,i}$).

Finally, as with the cylinder example, a g_{filter} function is applied to v_θ to get smoother velocity profiles. For each Lagrangian marker j, the final velocity is $v_{\theta,j} = g^{(j)}_{filter}(v_\theta(e_j))$ where $g^{(j)}_{filter}$ is the j^{th} element of the g_{filter} function:

Mesh	$\overline{C_\mathrm{d}}$	$\overline{C_\mathrm{l}}$	St	CPU time (min)
Mesh 1	0.167	0.436	0.887	565
Mesh 2	0.166	0.434	0.883	663
Mesh 3	0.166	0.434	0.883	330
Mesh 4	**0.166**	**0.434**	**0.883**	**223**
Mesh 5	0.168	0.439	0.889	2971
Ref. [53]	0.165	0.417	0.876	-
Ref. [84]	0.165	0.420	0.862	-

Table 4.3: Comparison of the mean coefficient $\overline{C_\mathrm{d}}$, the mean lift coefficient $\overline{C_\mathrm{l}}$ and the Strouhal number St for the different meshes described in A.4. In the present study, all the quantities are extracted after $U_\infty t/c = 50$. Results are compared with the literature. In both references, the mean drag and lift coefficients were not explicitly mentioned and were thus extracted from the plots with WebPlotDigitizer [97]. Computations were performed on a 48-node cluster, each node with 2 CPUs Intel Xeon E5 2670 at 2.6 GHz. Since these CPUs are octo core, 16 cores are available on each node. Each simulation was run using all the cores on one node. The mesh 4 (in bold) is chosen for the optimization process.

$$g_{filter}(v_\theta(\mathbf{e})) = \mathbf{G}^5 v_\theta(\mathbf{e}), \tag{4.38}$$

where $\mathbf{e}$ contains the arc length of all the 552 Lagrangian markers and $\mathbf{G}$ is consequently a 552×552 sparse matrix:

$$G = \begin{pmatrix} 0.5 & 0.25 & 0 & \dots & 0.25 \\ 0.25 & 0.5 & 0.25 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.25 & 0 & \dots & 0.25 & 0.5 \end{pmatrix}. \tag{4.39}$$

In this case, our aim is to minimize the drag coefficient and maximize the lift coefficient. Both functions are subject to a penalty term proportional to the actuation cost. Thus, we consider the two following objective functions:

$$\begin{aligned} f_1(\mathbf{s}) &= \frac{1}{\Delta T}\int_T^{T+\Delta T} C_\mathrm{d}(\mathbf{s},t)\,\mathrm{d}t \;\; + \alpha_d \sum_{i=0}^{N} \frac{\mathbf{s}^\mathsf{T}\mathbf{s}}{N}, \\ f_2(\mathbf{s}) &= -\frac{1}{\Delta T}\int_T^{T+\Delta T} C_\mathrm{l}(\mathbf{s},t)\,\mathrm{d}t \;\; + \alpha_l \sum_{i=0} N\frac{\mathbf{s}^\mathsf{T}\mathbf{s}}{N}, \end{aligned} \tag{4.40}$$

where C_d is the drag coefficient and C_l the lift coefficient. α_d and α_l were chosen such that the first right hand-side term was roughly equivalent to the second one. Thus, we set $\alpha_d = 0.5$ and $\alpha_l = 1.35$.

Numerical set-up

Five different meshes were created and are described in A.4. The mean drag coefficient $\overline{C_\mathrm{d}}$, the mean lift coefficient $\overline{C_\mathrm{l}}$ and the Strouhal number St were extracted and compared in order to select a mesh for the optimization process. As can be seen in

Table 4.3, all the quantities examined are fairly similar for the five meshes. For this reason, mesh 4 was chosen for its computational efficiency. The results are slightly higher than in the literature but are relatively close to for this mesh and we do not exceed a relative error of 4.1% in the worst case.

The airfoil profile is described by 552 Lagrangian markers. We first build the airfoil at $AoA = 0°$ with the leading edge located at $x/c = 0$ and the trailing edge at $x/c = 1$. Then, a rotation is performed around the center $(0.5, 0)$ in order to obtain $AoA = 10°$.

The simulations were initially run for $tU_\infty/D = 100$ to obtain an statistically steady regime. The actuation was then introduced starting from this time and run an additional $\Delta TU_\infty/c = 20$ time unit.

24, 30, 37 and 44 initial designs were extracted from a Latin Hypercube Sampling for respectively $N = 10, 20, 40, 80$. Both objective functions f_1 and f_2 were reduced fitting a quadratic model as described in 4.2.2. The reduced design space is thus two-dimensional. Again, the boundaries of the dimension reduction f_j were directly obtained as with the Fonseca-Fleming problem (Eq. 4.31).

The Algorithm 4 was then launched for 100 iterations and compared with the Algoritm 3. For the j functions, a_j, b_j, c_j, $h_{j,low}$ and $h_{j,up}$ were computed at each iteration as it does not require significant additional computational cost.

Results

Some of the optimal tangential velocity $v_\theta(\mathbf{e})$ found for the different N considered with the MOBO algorithm with dimension reduction as a function of x/c are shown in Fig. 4.7. In this figure, the light blue and red velocity profiles are respectively associated with the solutions with the lowest drag and highest lift coefficients found by the algorithm. As can be seen for all N, the main differences between the optimal Pareto solutions are located at the trailing edge of the airfoil. Indeed, when the tangential velocity is positive, the drag and the lift are decreased. On the contrary, when the tangential velocity is negative, both the drag and lift are raised. Note that as we increase the number of design parameters N, the velocity profile concentrates its spatial support at the trailing edge.

Compared to the uncontrolled case, with $N = 10$, we have in the best cases $f_1 = 0.163$ (equivalent to approximately a 2% reduction over the uncontrolled case), and $f_2 = -0.484$ (corresponding to a 11% diminution). When we increase N, these reductions are further increased and with $N = 80$, we obtain in the best cases $f_1 = 0.161$ (almost a 3% reduction) and $f_2 = -0.656$ (equivalent to a 51% decrease).

To investigate long-time averaged quantities from the optimal designs, the simulations were run again from the initial snapshot without control during $\Delta TU_\infty/c = 100$ and for $N = 80$.

Fig. 4.8 displays the drag and lift coefficients of the optimal solutions found in Fig. 4.7d as a function of the time. Only the solution with a positive tangential velocity (blue line) has a lower drag coefficient than the uncontrolled case (Fig. 4.8a). Averaged from $tU_\infty/c = 50$ to $tU_\infty/c = 100$, the lowest drag coefficient found is equal to $\overline{C_\mathrm{d}} = 0.149$ (approximately equivalent to a 10% reduction over the uncontrolled case), whereas for the same time-window average, the highest lift coefficient is $\overline{C_\mathrm{l}} = 0.707$ (corresponding to a 63% augmentation over the uncontrolled case). Also, note that as we decrease the tangential velocity at the tail, the lift and drag oscillations increase.

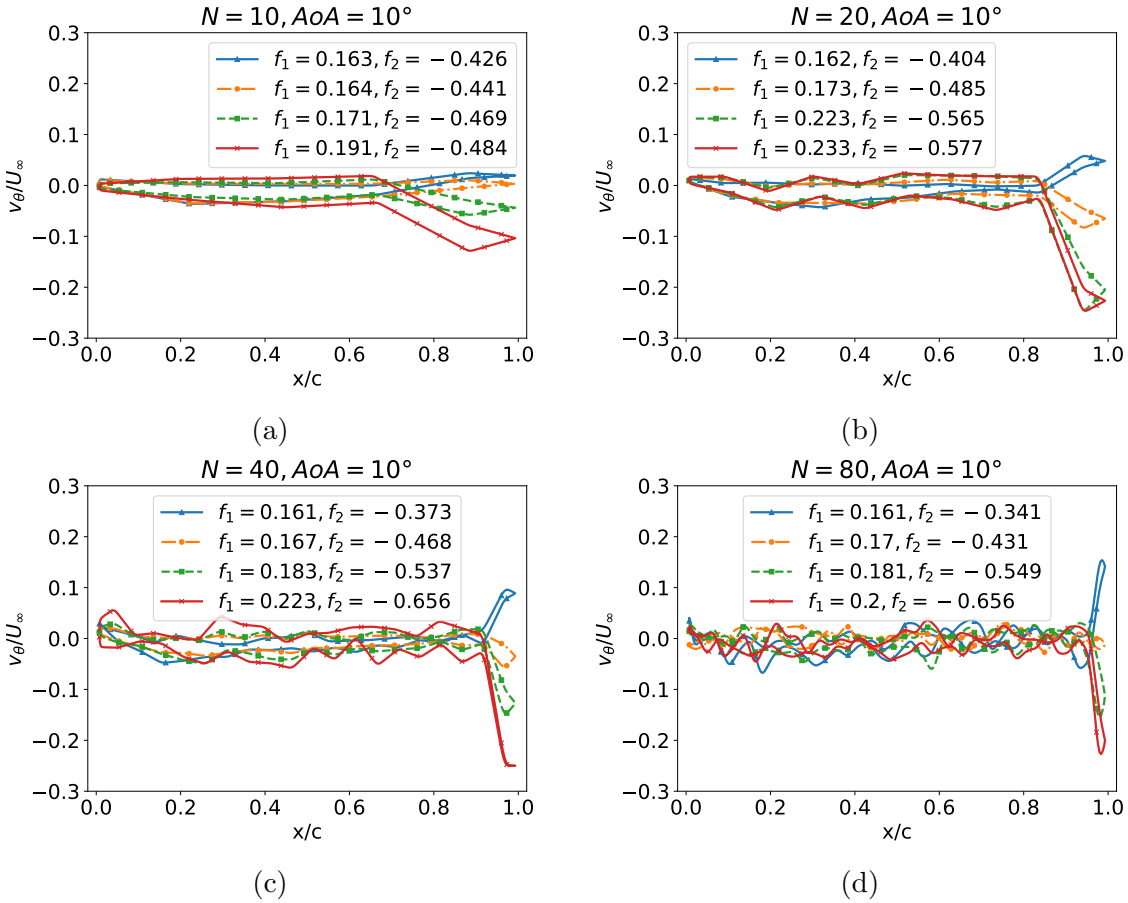


Figure 4.7: Some optimal tangential velocity profiles around the NACA 0012 airfoil as a function of the horizontal position for $AoA = 10°$. a) $N = 10$, b) $N = 20$, c) $N = 40$ and d) $N = 80$.

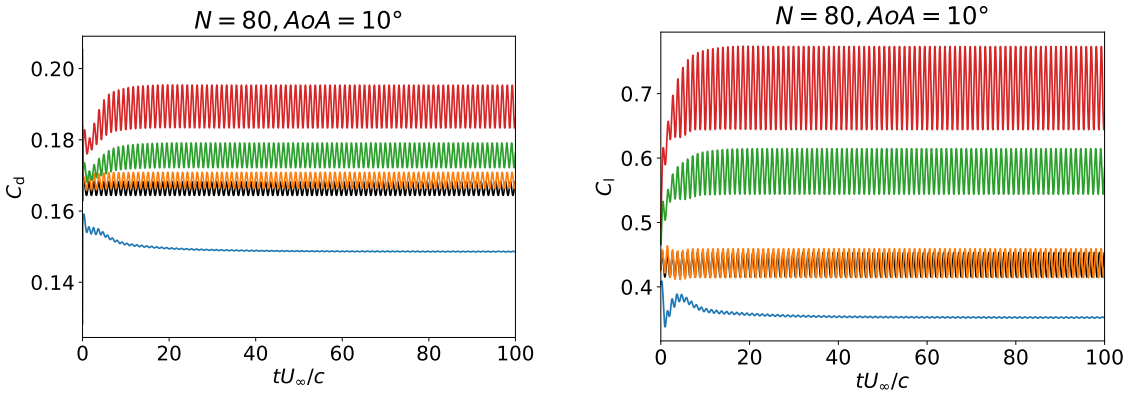


Figure 4.8: Coefficients as functions of the time for the optimal velocity profiles and the uncontrolled case for $N = 80$. a) Drag coefficient, b) lift coefficient. The lines use the same color code as the one used in Fig. 4.7 and are thus associated to the corresponding optimal tangential velocity profiles. The black line is the uncontrolled case.

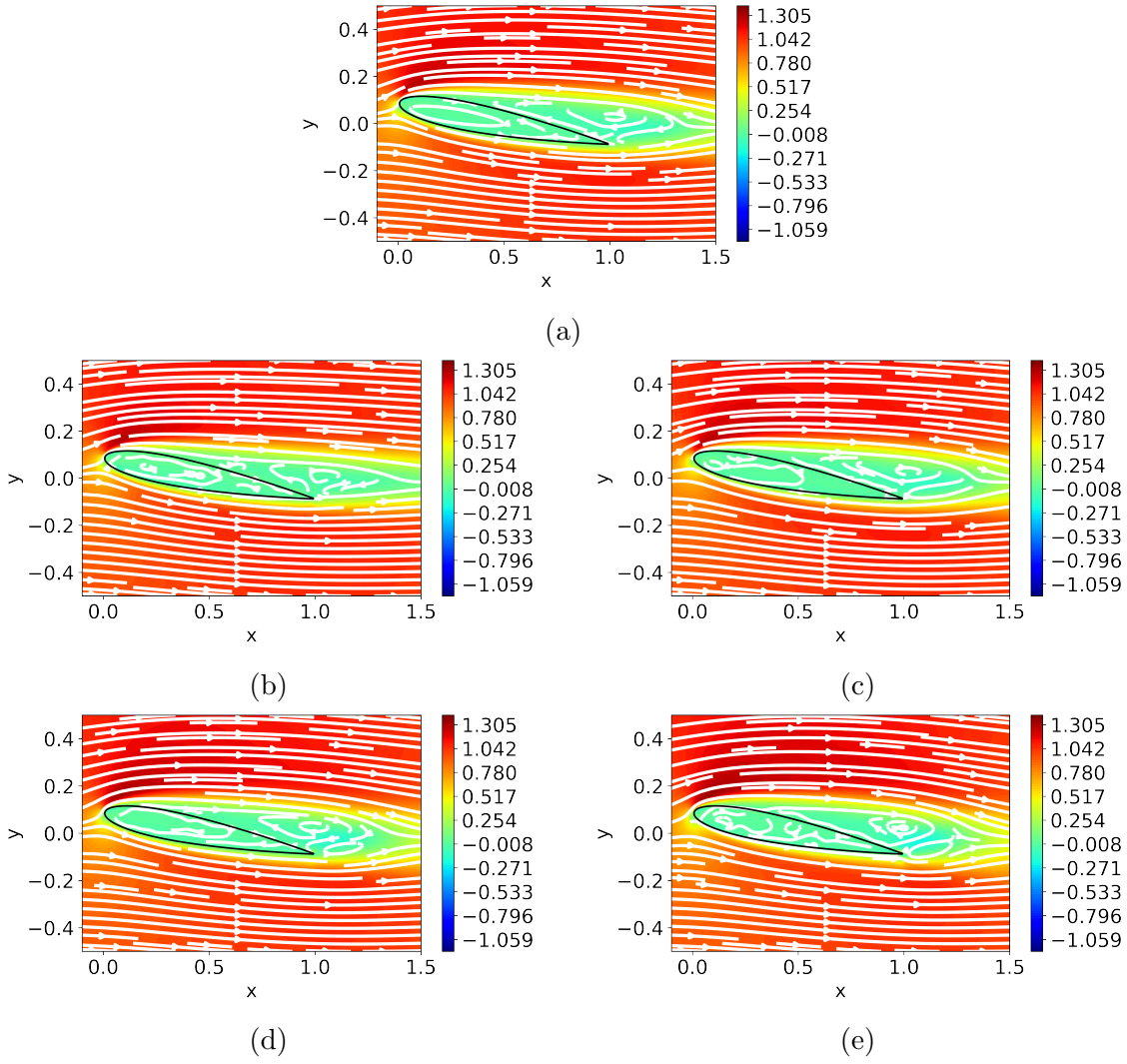


Figure 4.9: Averaged streamwise velocity with streamlines (white lines) and its directions (white arrows) for $N = 80$. a) Uncontrolled case, b), c), d) and e) are respectively the blue, yellow, green and red Pareto optimal solutions represented in Fig. 4.7d.

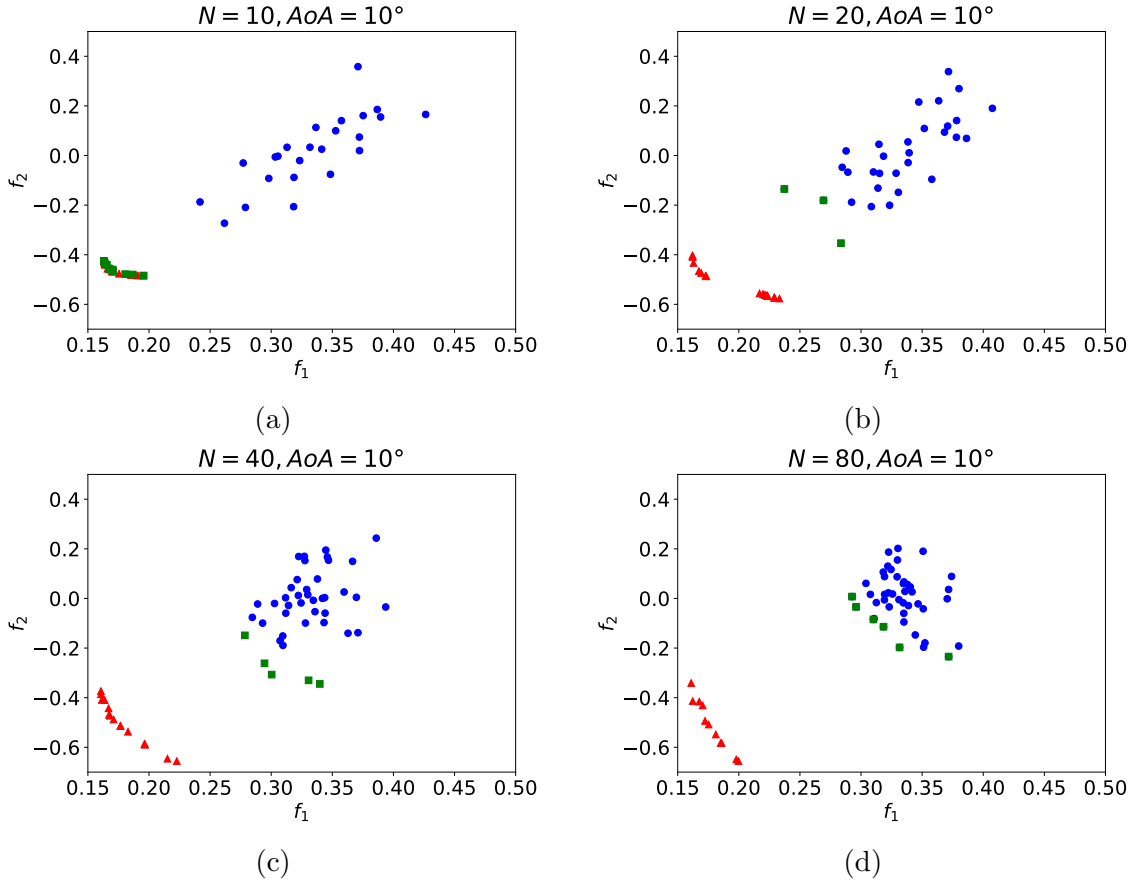


Figure 4.10: Pareto fronts found with the MOBO algorithm with dimension reduction (red triangles) and the MOBO algorithm without dimension reduction (green squares) for the objective functions defined in Eq. 4.40. Blue dots represent the initial DOE. a) $N = 10$, b) $N = 20$, c) $N = 40$, and d) $N = 80$.

The averaged streamwise velocity and the streamlines associated with the uncontrolled cases for various optimal solutions are represented in Fig. 4.9. For each case, a snapshot was stored each $0.3tU_\infty/c$ time units during the $100tU_\infty/c$ additional time units. All the stored snapshots were then averaged in order to compute the averaged streamwise velocity. When $AoA = 10°$, as mentioned in [53], two counter-rotating vortices can be observed for the uncontrolled case on the end of the suction side due to the boundary layer detachment (Fig. 4.9a). These vortices are still present for the optimal solutions (Fig. 4.9b, 4.9c, 4.9d, 4.9e). As highlighted in the optimal velocity profiles, the main difference between the optimal solutions is located at the trailing edge of the airfoil. Indeed, when a positive tangential velocity is set in this zone, on the lower surface, the pressure on the trailing edge decreases and thus both the lift and drag also diminish. On the contrary, when a negative velocity is set there, the fluid velocity decreases and the pressure is raised, resulting into a higher lift and drag, as can be seen in Fig. 4.9e.

The Pareto fronts for $N = 10, 20, 40, 80$ at the end of the optimization processes are displayed in Fig. 4.10. We find 11, 19, 16, 11 Pareto optimal solutions with the MOBO algorithm with dimension reduction and 18, 3, 5, 6 Pareto optimal solutions with the MOBO algorithm without dimension reduction for respectively $N = 10, 20, 40, 80$ (Fig. 4.10a, 4.10b, 4.10c, 4.10d). Thus, when $N > 10$, the num-

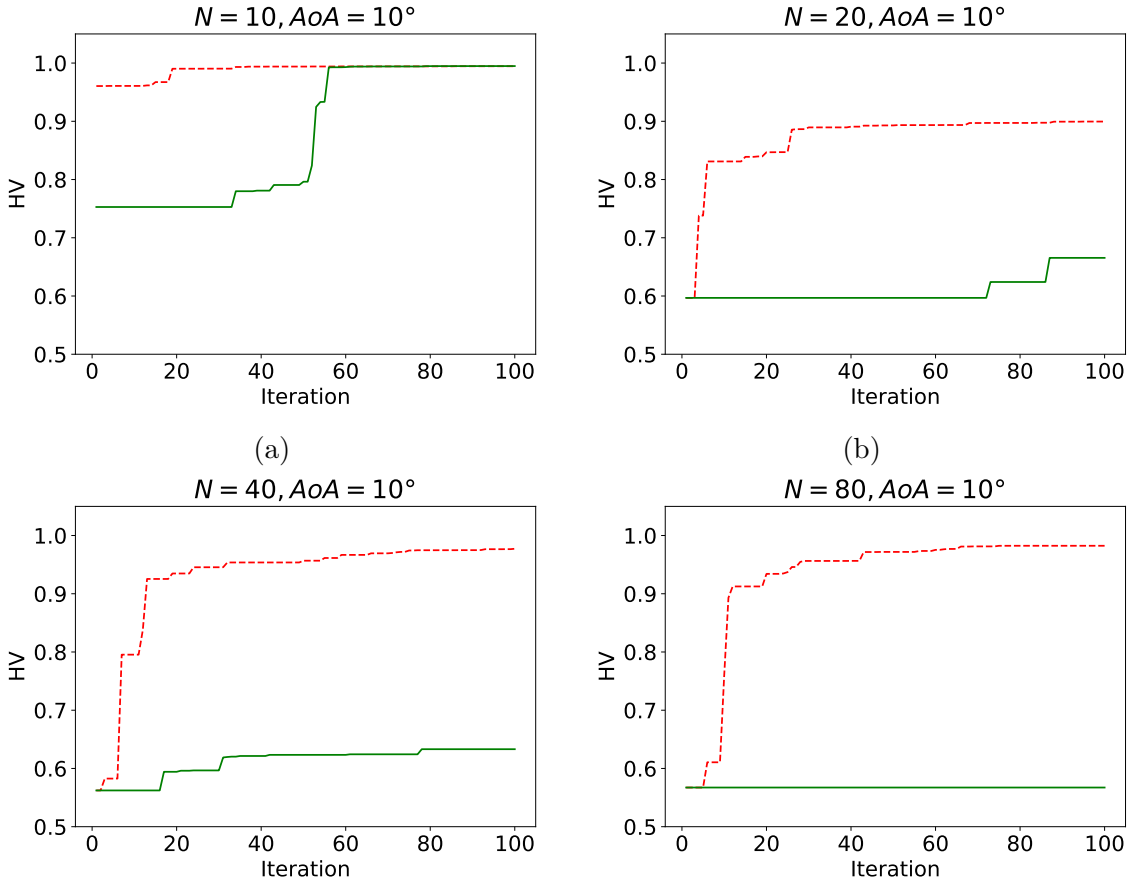


Figure 4.11: Hypervolume as a function of the iteration of the algorithm for the NACA problem. The red dashed and green solid lines are the MOBO algorithm respectively with and without dimension reduction. a) $N = 10$, b) $N = 20$, c) $N = 40$, and d) $N = 80$.

ber of Pareto optimal solutions found was higher with the MOBO algorithm with dimension reduction than without. When $N = 10$, the Pareto fronts were similar for both algorithms (Fig. 4.10a). However, as we increase N, the performance gap between the two algorithms also increase. Indeed, with $N = 20, 40, 80$ the algorithm with dimension reduction find better (or at least as good) Pareto optimal solutions than for $N = 10$. This is not the case of the algorithm without dimension reduction where no significant improvement over the DOE can be observed. All the Pareto solutions found by the algorithm are dominated by the ones obtained with the MOBO algorithm with dimension reduction.

The HV of the two algorithms as a function of the iteration for $N = 10, 20, 40, 80$ is displayed in Fig. 4.11. In each case, the reference point is computed with Eq. 4.5. As can be seen, even if for $N = 10$, more Pareto optimal solutions were obtained with the algorithm without dimension reduction, the HV in the end is similar to the one obtained with dimension reduction (Fig. 4.11a). Even it the final HV is higher with the algorithm without dimension reduction, the HV difference with the algorithm with dimension reduction does not exceed 0.03%. Moreover, we can note that the algorithm without dimension reduction requires a significant number of function evaluations to reach a higher HV than the one obtained with the algorithm

with dimension reduction. With $N = 20$ (Fig. 4.11b), $N = 40$ (Fig. 4.11b) and $N = 80$ (Fig. 4.11d), the algorithm without dimension reduction is unable to reach an HV value similar to the one obtained with the algorithm with dimension reduction before the end of the optimization budget. Also, for all the experiments performed, we observe a rapid increase and stabilization of the HV for the algorithm without dimension reduction. Indeed, the difference between the HV at the end of the optimization budget and the HV after 30 optimization iterations is lower than 5%.

4.4 Conclusions

In this chapter, a method that combines Multi-Objective Bayesian Optimization based on the ParEGO algorithm and dimension reduction was presented.

This method was first demonstrated on the Fonseca-Fleming test case. The dimension reduction was performed through a quadratic model. Results showed that the algorithm was able to find Pareto solutions close to the true Pareto front up to 100 design parameters. At each iteration, The hypervolume found by the algorithm with dimension reducion was also higher than for the algorithm without dimension reduction for all the dimensions considered ($N = 2, 5, 10, 20, 50, 100$).

We then applied the multi-objective Bayesian Optimization with dimension reduction algorithm to the two-dimensional cylinder case at $\mathrm{Re} = 40$. The design parameters were the amplitudes of 79 tangential actuators and the objective functions were the drag plus a penalization term consisting of the sum of the square of the amplitudes. The dimension reduction of the drag was performed through Active Subspaces estimated with gradients. The optimal solutions found were close to the ones obtained in Chapter 2 for the two-dimensional cylinder at $\mathrm{Re} = 500$. A Pareto solution found was similar to the one obtained with gradient-based optimization on the drag. None of the Pareto solutions found with the multi-objective Bayesian Optimization with dimension reduction were dominated by the ones found with the multi-objective Bayesian Optimization without dimension reduction. Also, more solutions were found with the former algorithm than with the latter one.

Finally, we applied this algorithm to the drag reduction in the flow around a two-dimensional NACA0012 profile at $\mathrm{Re} = 1000$ with an angle of attack $AoA = 10°$. The design parameters were $N = 10, 20, 40, 80$ equidistant tangential actuators set on the airfoil and the objective functions were the time-averaged drag and lift coefficients. A penalty term was added to both functions in order to penalize the cost of the actuation. The dimension reduction was performed through a quadratic model. Again, the developed multi-objective optimization algorithm showed better performances than the one without dimension reduction. As we increased the dimension of the design space, better Pareto solutions were found using the multi-objective Bayesian Optimization with dimension reduction algorithm.

In the future, this method could be extended in several ways. A first step could be to use another dimension reduction technique such as variational autoenconders [55] with a Gaussian Process latent variable model (GPVLM) to take into account the uncertainty in the input as in [104]. Additionally, acquisition criteria such as the Euclidean Expected Improvement (EEI) [45] or the expected hypervolume criterion (EHVI) [18] could be implemented. According to Zuhal [127], these methods showed better performances than the ParEGO algorithm. Finally, the performance of the method on cases with more than two objective functions and where the objective

functions can be reduced to design variables with a dimension greater than one should be investigated.

Chapter 5

Conclusions

5.1 Summary and conclusions

In this thesis, Bayesian Optimization was investigated for Computational Fluid Dynamics applications. The fundamental goals of this thesis were to demonstrate the Bayesian Optimization efficiency on numerical simulations that can typically arise in fluid mechanics, compare all the possible Gaussian Process models with multi-fidelity and/or derivative information for modelling or optimization and develop a numerical method laying on Bayesian Optimization to tackle multi-objective optimization problems in a high-dimensional space.

Firstly, Bayesian Optimization was applied to the canonical case of the cylinder at Re = 500. Tangential velocities were set on the cylinder wall at different locations. The goal was to minimize the root mean square of the sum of the drag coefficient and a penalty term proportional to the kinetic energy associated with the tangential actuation. Bayesian Optimization was considered both in serial and parallel and compared against other derivative-free methods such as CMA-ES, Particle Swarm Optimization, Nelder-Mead and Explorative Gradient Method. Results showed on that case that Bayesian Optimization in serial or in parallel was more efficient than other competitive algorithms. Whereas serial Bayesian Optimization was more efficient than all the other algorithms in terms of function evaluations, the parallel Bayesian Optimization performed the best in terms of the number of iterations. The influence of the Bayesian Optimization parameters were also studied. No influence of the size of the Design of Experiments, kernel or optimizer were noticed. The most important variables in the performance of the algorithm were the acquisition functions and the number of design parameters. Bayesian Optimization behaved well with the curse of dimension in that example since less than the double of function evaluations were required when the number of design parameters was more than doubled. The optimal solutions found by Bayesian Optimization indicated that the most important area of the cylinder where the momentum should be set is around the boundary layer detachment zone. Finally, the same method and parametrization were applied to the drag cylinder minimization of a three-dimensional cylinder at Re = 3900. A penalty term proportional to the kinetic energy associated with the actuators was again added to the objective function. With 7 design parameters, the optimum design was found in 36 iterations (with 5 initial samples). The most important location to reduce the drag was around the boundary layer detachment zone. A 23 % drag reduction was obtained. Results also showed that, compared

to the uncontrolled case that oscillated between a mode L and a mode H respectively associated with low and high drag coefficients, the actuation set could fix permanently the cylinder on a mode close to the mode L.

Secondly, the efficiency of Bayesian Optimization was investigated when various sources of information were available. Examples of sources of information include the gradients of the objective function according to the design parameters or the multi-fidelity model. In total, six possible Gaussian Process models produced by different combinations of multi-fidelity and gradients observations were studied. These six models were tested for modelling and optimization purposes on three different test cases: two benchmark objective functions and one Computational Fluid Dynamics case. For the latter, a tangential velocity profile similar to a wrapped normal distribution was set around a cylinder at Re = 200. The design parameters were the amplitude, the standard deviation and the angle on the cylinder where the maximum amplitude is reached. The objective function was the sum between the root mean square drag coefficient and a penalty term proportional to the kinetic energy set by the actuator. The normalized root mean square error and the normalized inference regret were investigated onto the three objective functions with different initialization costs, various gradient costs and different configuration of the Design of Experiments for the multi-fidelity models. Then for each model, a Bayesian Optimization algorithm was run for each gradient cost considered and the minimum obtained at each objective function observation was compared between the different models. On the three test cases investigated, it was generally observed that adding gradients on the high-fidelity objective function was especially useful for optimization, even when the gradient cost was twice the cost of the objective function. It was also efficient for modelling when enough samples could be set in the Design of Experiments. Adding gradient information on the low-fidelity objective function also generally gave better results than without for both modelling and optimization. The multi-fidelity model with gradient information only on the low-fidelity objective function was generally the fastest to decrease the objective function.

Thirdly, the multi-objective Bayesian Optimization in high dimensions was investigated. Two strategies are possible with Bayesian Optimization in high dimensions: adding the derivative information in the Gaussian Process model or reduce the dimension of the design space. Since adding the gradients in the model is difficult in high dimensions due to the cost of the Gaussian Process that raises cubically with the number of observations, the dimension reduction option was chosen. A method combining multi-objective Bayesian Optimization and dimension reduction was developed. A quadratic dimension reduction technique was also created to tackle non-linear dimension reduction problems. The multi-objective Bayesian Optimization algorithm was then applied to the Fonseca-Fleming benchmark optimization problem. Results showed that, in contrast to the case without dimension reduction, the developed algorithm was still able to find solutions close to the true Pareto front with a dimensional space up to 100 design parameters and for a total budget inferior to 150 function evaluations. The second case was the cylinder at Re = 40. 79 tangential actuators were set on the cylinder surface. The two objective functions were the drag coefficient and an objective function proportional to the kinetic energy set by the actuators. The multi-objective Bayesian Optimization with dimension reduction was again compared with the multi-objective Bayesian Optimization without dimension reduction. Results showed that the developed method found a

more diverse and efficient Pareto front than without using a dimension reduction method. The optimal velocity profiles found when mapping to the initial design space were coherent and a solution close to the one obtained with the adjoint techniques was found. The method was then deployed to the NACA0012 profile at $\mathrm{Re} = 1000$ and with an angle of attack of $AoA = 10°$. The objective functions were the time-averaged drag coefficient and the negative time averaged lift coefficient. Both objective functions were subject to a penalty term in order to limit the design parameters and highlight the critical zones of the airfoil. The design parameters were the tangential velocity amplitudes set at equally space points. The developed algorithm was compared with the multi-objective Bayesian Optimization without dimension reduction for $N = 10, 20, 40$ and 80 design parameters. The former method was able to find better Pareto solutions than the classical multi-objective Bayesian Optimization method, with $N = 20, 40$ and 80. Results also showed that the most critical area contributing to the drag reduction or lift increase is located at the trailing edge of the airfoil, on the pressure side. A streamwise velocity contributes to decrease the drag as less pressure is applied on the lower side of the airfoil whereas a negative streamwise velocity blocks the streamlines under the profile and increases the lift.

These findings suggest that BO and some of its variants investigated in this thesis are competitive tools to tackle optimization problems of practical interest in fluid systems. However, precaution must be exercised as their performance heavily depend on the sampling process, the choice of the surrogate model, the computational cost associated with each source of information and the acquisition function. Face to the intensive research efforts and new alternatives that have been proposed recently, careful assessment is required before they can routinely be used in flow optimization. The results shown in this work form an educated sneak peek of the complex landscape to be explored.

5.2 Suggestions for future work

Bayesian Optimization could easily be applied to other LES applications where the gradients are not accessible. An interesting idea is to keep on investigating on the drag cylinder minimization at $\mathrm{Re} = 3900$ through a spanwise control [49, 81]. The optimization of a time-dependent control system can also be considered.

Secondly, the performances of the different possible Gaussian Processes models with gradient information and/or multi-fidelity for global modelling and optimization deserve further research efforts. Indeed, guidelines on how to choose *a priori* the model according to the cost of each information and to the available budget should be examined with other cases to confirm the conclusions suggested by our results. The non-linear multi-fidelity model of [91] can also be investigated with gradient information as done in Chapter 3 to compare the performances of this model with the linear multi-fidelity formulation. Existing methods to alleviate the cost of building the models with gradients should also be combined with multi-fidelity models. A cheap-to-evaluate acquisition function who relies on the Value of Information (VOI) would also be an improvement for the multi-fidelity and single-fidelity models with derivative information. This acquisition function could take into account the cost of the derivative information and decide dynamically at each iteration if obtaining the gradients at a design point is more valuable than performing objective function

observations at the same computational cost.

Finally, the algorithm developed in Chapter 4 of this thesis should be assessed on problems of higher complexity and/or more objective functions and compared with other techniques to determine the most efficient algorithm to tackle high-dimensional multi-objective optimization problems.

Appendix A

Appendix

A.1 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 2.3. 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.1. 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.1: 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.2, 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 compu-

tational 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. [126]	-	-	0.23	1.463	0.837	-	-	-	-
Ref. [121]	-	-	-	1.28	0.622	-	-	-	-
Ref. [3]	-	-	0.235	1.518	0.876	-	-	-	-
Ref. [70]	-	-	0.225	1.440	0.818	-	-	-	-

Table A.2: 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 2.3 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.

A.2 Validation of the three-dimensional case

We present here the details of the simulation for the three-dimensional cylinder at $\mathrm{Re} = 3900$. 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 $\mathrm{Ma} = 0.2$, the Prandtl number is $\mathrm{Pr} = 0.71$ 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. (2.23), 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 A.3.

Case	$f_{vs}D/U_\infty$	$\overline{C}_\mathrm{d}$	L_d/D	$-\overline{C}_\mathrm{pb}$
Ref. [63]	0.215	1.015	1.36	0.935
Ref. [5]	0.215	1.016	1.372	0.941
Ref. [51]	0.21	1.04	1.35	0.94
Ref. [23]	0.209	0.978	1.64	0.85
Ref. [74]	0.218	1.0	1.35	-
Ref. [76]	0.206	0.99	-	0.86
Ref. [89]	0.208	-	1.51	-
Present study	**0.208**	**1.027**	**1.51**	**0.895**

Table A.3: Comparison of the three-dimensional long-time averaged uncontrolled case at $\mathrm{Re} = 3900$ 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.

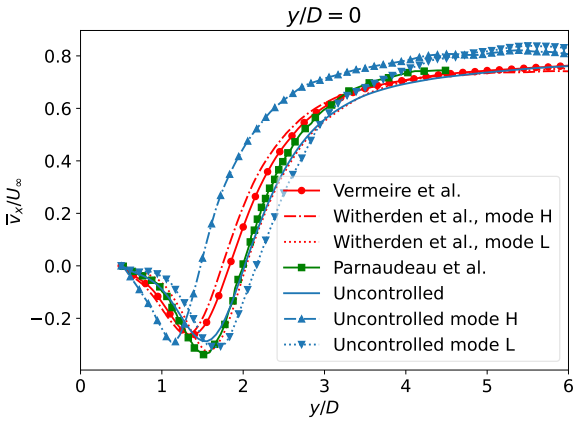


Figure A.1: Three-dimensional cylinder at $\mathrm{Re} = 3900$. Time-averaged streamwise velocity profiles. Comparison with the long time-averaged quantities of Vermeire *et al.* [115], with the modes H and L of Witherden *et al.* [119] and the experimental results of Parnaudeau *et al.* [89].

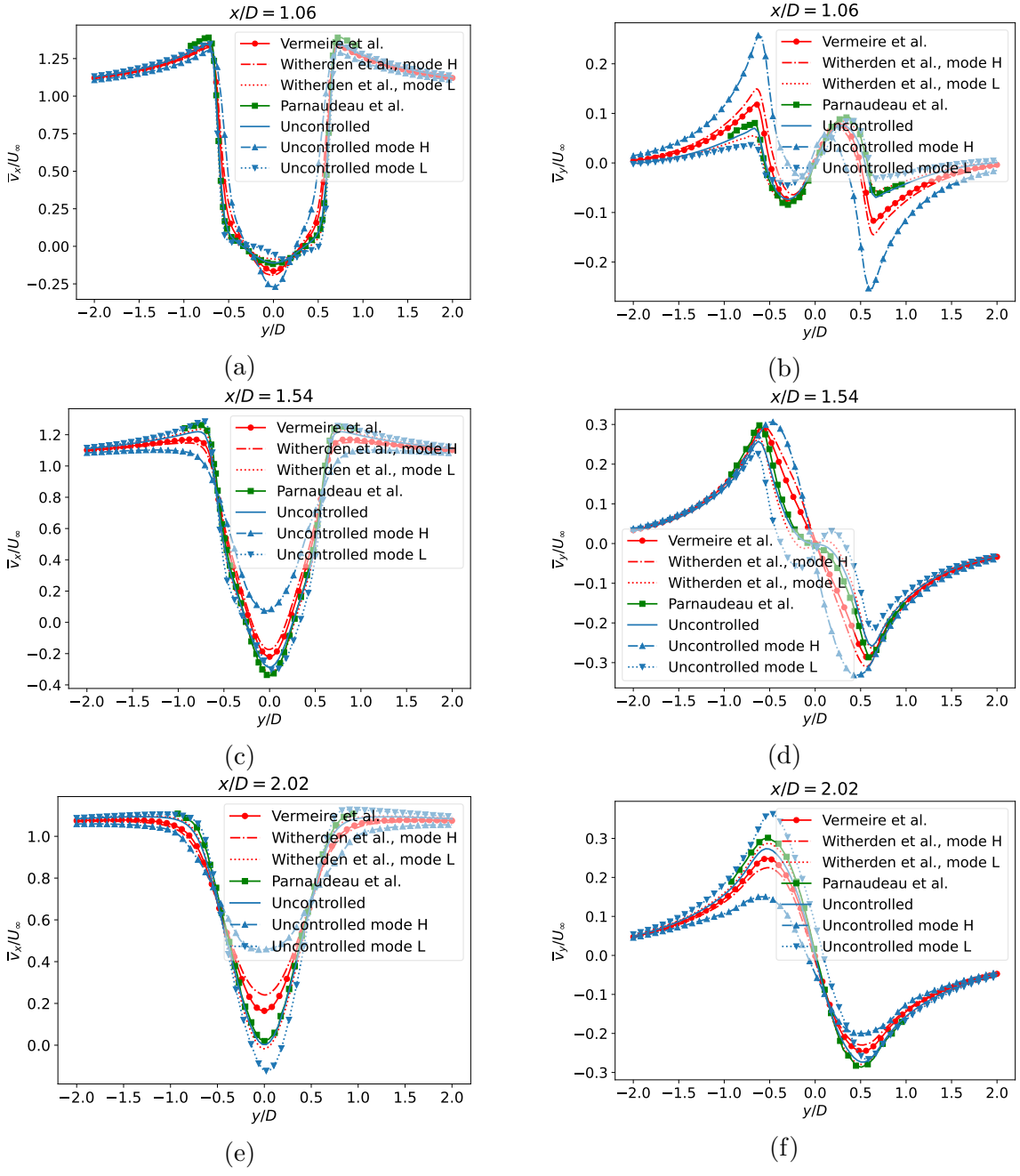


Figure A.2: Three-dimensional cylinder at Re = 3900. Time-averaged streamwise (left) and cross-flow (right) velocity profiles. Top row: $x/D = 1.06$, middle row: $x/D = 1.54$, bottom row: $x/D = 2.02$. Comparison with the long time-averaged quantities of Vermeire *et al.* [115], with the modes H and L of Witherden *et al.* [119] and the experimental results of Parnaudeau *et al.* [89].

A.3 NACA 0012 meshes

In this section, we present the meshes used for the validation of the NACA0012 simulations with an angle of attack of $AoA = 10°$ at Re = 1000. The main characteristics of the meshes are presented in Table A.4.

Mesh	x_u/c	x_d/c	y_t/c	N_{x_u}	N_{x_d}	N_{y_t}	$\Delta x_0/c$	$U_\infty \Delta t/c$
Mesh 1	-10.53	10.31	9.51	85	395	224	0.0037	0.0005
Mesh 2	-14.34	18.16	21.75	87	450	235	0.0037	0.0005
Mesh 3	-14.34	18.16	21.75	87	450	235	0.0037	0.001
Mesh 4	-14.34	18.16	21.75	87	450	235	0.0037	0.0015
Mesh 5	-17.68	37.19	26.65	114	820	610	0.002	0.0005

Table A.4: Meshes used for the validation of the NACA 0012 profile at $AoA = 10°$. x_u/c, x_d/c and y_t/c are respectively the upstream, downstream and top boundaries; N_{x_u}, N_{x_d} and N_{y_t} are the number of points used in the upstream, downstream and top direction; $\Delta x_0/c$ is the mesh size around the NACA profile and $U_\infty \Delta t/c$ is the non-dimensional time. For all the meshes, final meshes sizes of $\Delta x_d/c = 0.14$, $\Delta x_u/c = 1.85$ and $\Delta y_t/c = 1.23$ are respectively used in the upstream, downstream and top direction. The bottom boundary y_b/c is the symmetry of the mesh in the top direction according to the axis $x = 0$.

Bibliography

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

[2] C. Audet, J. E. Denni Jr., D. Moore, A. Booker, and P. Frank. A surrogate-model-based method for constrained optimization. In *8th symposium on multidisciplinary analysis and optimization*, page 4891, 2000.

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

[4] A. Belov, L. Martinelli, and A. Jameson. A new implicit algorithm with multigrid for unsteady incompressible flow calculations. In *33rd Aerospace sciences meeting and exhibit*, page 49, 1995.

[5] 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):1281–1302, 1998.

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

[7] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

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

[9] A. Carpentier and R. Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Artificial Intelligence and Statistics*, pages 190–198. PMLR, 2012.

[10] P. Catalano, M. Wang, G. Iaccarino, and I. F. Sbalzarini. Optimization of cylinder flow control via actuators with zero net mass flux. *Center for Turbulence Research, Proceedings of the Summer Program*, pages 297–303, 2002.

[11] M. Cavazzuti. *Optimization Methods: From Theory to Design*. Springer-Verlag Berlin Heidelberg, 2013.

[12] B. Chen, R. Castro, and A. Krause. Joint optimization and variable selection of high-dimensional gaussian processes. *arXiv preprint arXiv:1206.6396*, 2012.

[13] P. G. Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015.

[14] M. Coutanceau and R. Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. *Journal of Fluid Mechanics*, 79(2):231–256, 1977.

[15] D. D. Cox and S. John. A statistical method for global optimization. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE, 1992.

[16] S. Dennis and G.-Z. Chang. Numerical solutions for steady flow past a circular cylinder at reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(3):471–489, 1970.

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

[18] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

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

[20] C. M. Fonseca and P. J. Fleming. Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 45–52. IET, 1995.

[21] A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering Design via Surrogate Modelling: a practical guide*. John Wiley & Sons Ltd., 2008.

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

[23] J. Franke and 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):1191–1206, 2002.

[24] P. Frazier, W. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

[25] P. I. Frazier. A Tutorial on Bayesian Optimization. *arXiv preprint arXiv:1807.02811*, 2018.

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

[27] F. Giannetti and P. Luchini. Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581:167–197, 2007.

[28] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000.

[29] J. González, Z. Dai, P. Hennig, and N. D. Lawrence. Batch Bayesian Optimization via Local Penalization. In *Artificial Intelligence and Statistics*, pages 648–657, 2016.

[30] Z. J. Grey and P. G. Constantine. Active subspaces of airfoil shape parameterizations. *AIAA Journal*, 56(5):2003–2017, 2018.

[31] Z.-H. Han, S. Görtz, and R. Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and technology*, 25(1):177–189, 2013.

[32] Z.-H. Han, Y. Zhang, C.-X. Song, and K.-S. Zhang. Weighted gradient-enhanced kriging for high-dimensional surrogate modeling and design optimization. *AIAA Journal*, 55(12):4330–4346, 2017.

[33] N. Hansen. The CMA Evolution Strategy: A Tutorial. *arXiv preprint ArXiv:1604.00772*, 2016.

[34] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

[35] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

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

[37] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of global optimization*, 34(3):441–466, 2006.

[38] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pages 507–523. Springer, 2011.

[39] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. In *18th AIAA computational fluid dynamics conference*, page 4079, 2007.

[40] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

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

[42] W. Ji, J. Wang, O. Zahm, Y. M. Marzouk, B. Yang, Z. Ren, and C. K. Law. Shared low-dimensional subspaces for propagating kinetic uncertainty to multiple outputs. *Combustion and Flame*, 190:146–157, 2018.

[43] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black - Box Functions. *Journal of Global Optimization*, 13:455–492, 1998.

[44] Jones, Donald R. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):39, 2001.

[45] A. J. Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.

[46] C. A. Kennedy, M. H. Carpenter, and M. R. Lewis. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. *Applied Numerical Mathematics*, 35(3):177–219, 2000.

[47] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[48] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[49] J. Kim and H. Choi. Distributed forcing of flow over a circular cylinder. *Physics of Fluids*, 17(3):033103, 2005.

[50] J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

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

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

[53] D. F. Kurtulus. On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at Re= 1000. *International journal of micro air vehicles*, 7(3):301–326, 2015.

[54] 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):97–106, 1964.

[55] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. In *International conference on machine learning*, pages 1945–1954. PMLR, 2017.

[56] M.-C. Lai and C. S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of computational Physics*, 160(2):705–719, 2000.

[57] R. Lam. *Scaling Bayesian Optimization for Engineering Design: Lookahead Approaches and Multifidelity Dimension Reduction*. PhD thesis, Massachussets Institute of Technology, 2018.

[58] R. Lam, D. L. Allaire, and K. E. Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143, 2015.

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

[60] M. Lamboni, H. Monod, and D. Makowski. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliability Engineering & System Safety*, 96(4):450–459, 2011.

[61] A. Larroque, M. Fosas de Pando, and L. Lafuente. Cylinder drag minimization through wall actuation: A bayesian optimization approach. *Computers & Fluids*, 240:105370, 2022.

[62] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16, 2003.

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

[64] Y. Li, W. Cui, Q. Jia, Q. Li, Z. Yang, M. Morzyński, and B. R. Noack. Explorative gradient method for active drag reduction of the fluidic pinball and slanted Ahmed body. *arXiv preprint arXiv:1905.12036*, 2020.

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

[66] Y. Ling, S. Ghosh, I. Asher, J. Kristensen, K. Ryan, and L. Wang. An intelligent sampling framework for multi-objective optimization in high dimensional design space. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0912, 2018.

[67] M. N. Linnick and H. F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204(1):157–192, 2005.

[68] J. L. Lions. *Optimal control of systems governed by partial differential equations*. Springer, 1971.

[69] C. Liu, X. Zheng, and C. Sung. Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational physics*, 139(1):35–57, 1998.

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

[71] D. J. Lizotte. *Practical bayesian optimization*. University of Alberta, 2008.

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

[73] T. W. Lukaczyk, P. Constantine, F. Palacios, and J. J. Alonso. Active subspaces for shape optimization. In *10th AIAA multidisciplinary design optimization conference*, page 1171, 2014.

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

[75] O. A. Mahfoze, S. Laizet, and A. Wynn. Bayesian optimisation of intermittent wall blowing for drag reduction of a spatially evolving turbulent boundary layer. *Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona,Spain, July 9-13, 2018*, pages 1–17, 2018.

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

[77] X. Mao, H. M. Blackburn, and S. J. Sherwin. Nonlinear optimal suppression of vortex shedding from a circular cylinder. *Journal of Fluid Mechanics*, 144(4):744–763, 2015.

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

[79] A. Marco, F. Berkenkamp, P. Hennig, A. P. Schoellig, A. Krause, S. Schaal, and S. Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1557–1563. IEEE, 2017.

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

[81] P. Meliga, E. Boujo, M. Meldi, and 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*, 72:123–134, 2018.

[82] T. Mengistu and W. Ghaly. Single and multipoint shape optimization of gas turbine blade cascades. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1652–1662, 2004.

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

[84] S. Mittal and T. E. Tezduyar. Massively parallel finite element computation of incompressible flows involving fluid-body interactions. *Computer Methods in Applied Mechanics and Engineering*, 112(1-4):253–282, 1994.

[85] J. Mockus, V. Tiesis, and A. Zilinskas. *Toward global optimization, volume 2, chapter bayesian methods for seeking the extremum.* Elsevier, 1978.

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

[87] A. Paleyes, M. Pullin, M. Mahsereci, C. McCollum, N. Lawrence, and J. González. Emulation of physical processes with emukit. In *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*, 2019.

[88] C. Park, R. T. Haftka, and N. H. Kim. Remarks on multi-fidelity surrogates. *Structural and Multidisciplinary Optimization*, 55(3):1029–1050, 2017.

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

[90] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[91] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751, 2017.

[92] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[93] M. Poloczek, J. Wang, and P. Frazier. Multi-information source optimization. *Advances in neural information processing systems*, 30, 2017.

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

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

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

[97] A. Rohatgi. Webplotdigitizer: Version 4.5, 2021.

[98] A. Roshko. *On the development of turbulent wakes from vortex streets*. National Advisory Committee for Aeronautics, 1953.

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

[100] A. Safari, K. H. Hajikolaei, H. Lemu, and 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*, volume 49712, page V02CT45A013. American Society of Mechanical Engineers, 2016.

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

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

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

[104] E. Siivola, A. Paleyes, J. González, and A. Vehtari. Good practices for bayesian optimization of high dimensional structured spaces. *Applied AI Letters*, 2(2):e24, 2021.

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

[106] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[107] J. Svenson and T. Santner. Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models. *Computational Statistics and Data Analysis*, 94:250–264, 2016.

[108] K. Taira and T. Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.

[109] S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi, and M. Karasuyama. Multi-fidelity bayesian optimization with max-value entropy search and its parallelization. In *International Conference on Machine Learning*, pages 9334–9345. PMLR, 2020.

[110] C. Talnikar, P. Blonigan, J. Bodart, and Q. Wang. Parallel optimization for large eddy simulations. In *Center for Turbulence Research Proceedings of the Summer Program 2014*, oct 2014.

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

[112] The GPyOpt authors. GPyOpt: A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

[113] D. J. Tritton. Experiments on the flow past a circular cylinder at low reynolds numbers. *Journal of Fluid Mechanics*, 6(4):547–567, 1959.

[114] S. Ulaganathan, I. Couckuyt, F. Ferranti, E. Laermans, and T. Dhaene. Performance study of multi-fidelity gradient enhanced kriging. *Structural and Multidisciplinary Optimization*, 51(5):1017–1033, 2015.

[115] B. C. Vermeire, F. D. Witherden, and 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*, 334:497–521, 2017.

[116] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[117] Z. Wang and S. Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR, 2017.

[118] F. D. Witherden, A. M. Farrington, and 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):3028–3040, 2014.

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

[120] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

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

[122] J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier. Bayesian optimization with gradients. *Advances in neural information processing systems*, 30, 2017.

[123] W. Yamazaki and D. J. Mavriplis. Derivative-enhanced variable fidelity surrogate modeling for aerodynamic functions. *AIAA Journal*, 51(1):126–137, 2013.

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

[125] O. Zahm, P. G. Constantine, C. Prieur, and Y. M. Marzouk. Gradient-based dimension reduction of multivariate vector-valued functions. *SIAM Journal on Scientific Computing*, 42(1):A534–A558, 2020.

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

[127] L. R. Zuhal, C. Amalinadhi, Y. B. Dwianto, P. S. Palar, and K. Shimoyama. Benchmarking multi-objective bayesian global optimization strategies for aerodynamic design. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0914, 2018.